

INTRODUÇÃO AO SIMUL8

um guia prático

Leonardo Chwif
Afonso C. Medina
Wilson I. Pereira
Darli R. Vieira
José E. Pécora Jr



Primeira edição atualizada/Simul8 2015

Introdução ao SIMUL8

Um guia prático

Primeira Edição Atualizada

Leonardo Chwif
Afonso C. Medina
Wilson I. Pereira
Darli R. Vieira
José E. Pécora Jr.

São Paulo
2015

INTRODUÇÃO AO SIMUL8: UM GUIA PRÁTICO

1ª EDIÇÃO ATUALIZADA

Copyright © 2015 by AFONSO C. MEDINA E LEONARDO CHWIF

Capa e projeto gráfico: AFONSO C. MEDINA

Revisão: WILSON I. PEREIRA

Diagramação: AFONSO C. MEDINA

Dados Internacionais de Catalogação na Publicação (CIP) (Câmara Brasileira do Livro, SP, Brasil)

Introdução ao SIMUL8 [livro eletrônico] : um guia
prático / Leonardo Chwif...[et al.]. -- 1. ed. --
São Paulo : Ed. dos Autores, 2013.

Outros autores: Afonso C. Medina, Darli R.
Vieira, José E. Pécora Jr., Wilson I. Pereira
12 Mb ; PDF
ISBN 978-85-905978-4-1

1. Computadores - Simulação 2. Métodos de
simulação 3. Simulação de eventos discretos
4. Software de simulação I. Chwif, Leonardo.
II. Medina, Afonso C.. III. Vieira, Darli R..
IV. Pécora Junior, José E.. V. Pereira, Wilson I..

12-14907

CDD-003.3

Índices para catálogo sistemático:

1. SIMUL8 : Computadores : Simulação :
Programas : Processamento de dados 003.3

[2015]

Todos os direitos desta edição reservados a:

LEONARDO CHWIF

AFONSO C. MEDINA

livrosimulacao@livrosimulacao.eng.br

<http://www.livrosimulacao.eng.br>

Prefácio

O objetivo deste livro é introduzir o leitor no uso de uma das mais poderosas e utilizadas ferramentas de simulação: o SIMUL8.

POR QUE O SIMUL8?

O SIMUL8 começou a ser desenvolvido na década de 1990 na Universidade de Strathclyde (Escócia), com a finalidade de ensino de simulação. Devido ao seu sucesso como ferramenta de ensino de simulação, a Simul8 Corporation iniciou a comercialização do *software* como ferramenta profissional para projetos de simulação.

PÚBLICO-ALVO

Este livro foi planejado para ser usado em cursos de simulação, para o profissional de simulação autodidata que deseja utilizar o SIMUL8 em seus projetos e até mesmo para quem já o utiliza no seu cotidiano, mas não está totalmente familiarizado com as capacidades específicas do *software*. Além disso, estudantes de administração, engenharia, ciências da computação e áreas afins, que estejam cursando disciplinas de simulação de eventos discretos e que possuam atividades práticas e projetos no SIMUL8, tornam-se automaticamente público-alvo deste livro.

ORGANIZAÇÃO DO TEXTO

Ao escrever este livro, nós adotamos um estilo próximo ao de tutorial, de modo que, para o leitor, o tempo de aprendizagem dos comandos e funções sejam os menores possíveis. O livro está dividido em 8 capítulos.

No Capítulo 1, mostramos os recursos básicos do SIMUL8 por meio de um exemplo simples. No Capítulo 2 discutimos a alocação de recursos em modelos de simulação. O Capítulo 3 é dedicado ao uso de distribuições aleatórias. O Capítulo 4 mostra como podem ser representados desvios de fluxo em simulação, por meio de roteamentos. O Capítulo 5 introduz o ambiente de programação do SIMUL8 (*Visual Logic*), que permite representar situações complexas. O Capítulo 6 discute os refinamentos visuais que podem (e devem) ser feitos em modelos de simulação. O Capítulo 7 mostra o uso de diversas funcionalidades específicas do SIMUL8, que servem para representar casos não contemplados nos capítulos anteriores. O Capítulo 8 descreve casos práticos de simulação extraídos de projetos reais de consultoria. Por fim, inserimos um apêndice que ilustra o uso do *software* Stat::Fit, específico para o tratamento de variáveis aleatórias.

COMO UTILIZAR ESTE LIVRO

Para aqueles que nunca tiveram contato com algum *software* de simulação, recomendamos a leitura dos Capítulos de 1 a 4 e de 5 a 7. Profissionais que já utilizam o SIMUL8 podem, eventualmente, iniciar pelos Capítulos de 5 a 8. De qualquer forma, o livro foi concebido para ser lido na sequência, qualquer que seja o nível de conhecimento em simulação do leitor.

Leonardo Chwif

Afonso Celso Medina

Wilson Inacio Pereira

Darli Rodrigues Vieira

José Eduardo Pécora Jr.

RECURSOS DISPONÍVEIS

Acreditamos que o ideal para aproveitar este livro é acompanhar as explicações e discussões com o *software* SIMUL8. Para quem não o possui, disponibilizamos um sítio na internet para *download* e instalação de uma versão *Trial* do SIMUL8 que é funcional durante 14 dias. Você poderá baixá-lo em:

<http://www.simul8.com/ebookbr>

Adicionalmente a empresa Simulate Tecnologia de Simulação promove cursos on-line e presenciais do SIMUL8. Para maiores informações, acesse:

<http://www.simulate.com.br/treinamento.html>

Por fim, o livro tem seu próprio sítio na Internet, um lugar para você baixar material mais específico sobre Simulação de Eventos Discretos:

<http://www.livrosimulacao.eng.br>

Visite-nos!

São Paulo, Julho de 2015.

Sobre os Autores

LEONARDO CHWIF

É graduado em Engenharia Mecatrônica pela Escola Politécnica da Universidade de São Paulo, Mestre pela USP e Doutor pela USP/Brunel University (U.K) em simulação de processos. Leonardo possui inúmeros artigos em simulação publicados em periódicos e conferências nacionais e internacionais. Possui mais de 15 anos de experiência em simulação, tendo trabalhado também em grandes empresas como Mercedes-Benz e Whirlpool Corporation. Já realizou mais de 100 projetos de simulação. Juntamente com Afonso Celso Medina é autor do livro “Modelagem e Simulação de Eventos Discretos”.

AFONSO CELSO MEDINA

Engenheiro e Mestre em Engenharia Naval pela Escola Politécnica da USP, pesquisador em logística, transporte e simulação. Ex-professor da EPUSP e do Instituto Mauá de Tecnologia. Atualmente, é pesquisador no Centro de Inovação em Logística e Infraestrutura Portuária da EPUSP. Possui mais de 20 anos de experiência em projetos de Logística, Transportes e Simulação para empresas como: CSN, VALE, LLX, Codesp, Transpetro, Promon dentre outras. Juntamente com Leonardo Chwif é autor do livro “Modelagem e Simulação de Eventos Discretos”.

WILSON INACIO PEREIRA

Professor da Escola de Engenharia Mauá desde 2001, leciona disciplinas na área de Pesquisa Operacional desde 2005. Possui graduação em Engenharia Elétrica pela Escola de Engenharia Mauá e especialização em Docência no Ensino Superior pela Universidade Municipal de São Caetano do Sul.

DARLI RODRIGUES VIEIRA

Professor da Université du Québec à Trois-Rivières (UQTR – Canadá). Ex-Professor da Universidade Federal do Paraná (UFPR) e do Instituto Tecnológico de Aeronáutica (ITA). Tem sido Professor Visitante em diversas universidades estrangeiras. Autor de diversos trabalhos, entre os quais se destacam os livros recentes: *Gestão de projeto do produto baseada na metodologia Product Lifecycle Management* (Ed. Elsevier), *Auditoria Logística* (Ed. Elsevier), *Projeto de Centros de Distribuição* (Ed. Elsevier) e *Análise e Projetos de Redes Logísticas* (Ed. Saraiva).

JOSÉ EDUARDO PÉCORA JR.

Possui graduação e Mestrado em Matemática Aplicada pela Unicamp e PhD (Operations and Decision Systems) pela Universidade Laval no Canadá. Atuou como consultor em diversas empresas como: Sadia, Parmalat, CST e Aracruz. Atualmente é professor adjunto do Departamento de Administração Geral e Aplicada e do MBA em Gerência de Sistemas Logísticos da Universidade Federal do Paraná.

Sumário

Capítulo 1	Introdução.....	11
1.1.	Exemplo inicial.....	11
1.2.	Iniciando o SIMUL8	11
1.3.	Construindo o modelo.....	13
1.4.	Ajustando os parâmetros dos objetos	16
1.5.	Preparação para a simulação	21
1.5.1.	Tempos de deslocamento.....	21
1.5.2.	Ajuste da unidade de tempo	22
1.5.3.	Ajuste do tempo de simulação	23
1.6.	Executando a simulação	24
1.7.	Consulta de resultados: uma replicação	25
1.7.1.	Resultados dos pontos de entrada (<i>Start Point</i>)	25
1.7.2.	Resultados das filas de espera (<i>Queue</i>)	26
1.7.3.	Resultados das atividades (<i>Activity</i>)	27
1.7.4.	Consulta de resultados: resumo de resultados	28
1.8.	Executando uma <i>rodada de simulação</i> (várias replicações).....	29
1.9.	Dimensionando o número de replicações	31
Capítulo 2	Alocação, controle de recursos e <i>Labels</i>	34
2.1.	Introdução.....	34
2.2.	Exemplo inicial.....	34
2.3.	Prioridade de recursos (<i>Resources Priority</i>).....	39
2.4.	Turnos de trabalho (<i>Shift Work Pattern/ Availability</i>)	40
2.5.	Opções detalhadas de recursos.....	42
2.6.	<i>Pool</i> de recursos	46
2.7.	Deslocamentos de recursos.....	48
2.8.	<i>Labels</i> e <i>Resource by Labels</i>	48

2.8.1.	Criando <i>labels</i>	49
2.8.2.	<i>Resource by labels</i>	50
2.9.	Erros comuns de alocação de recursos	51
Capítulo 3	Distribuições Aleatórias no Simul8.....	53
3.1.	Introdução.....	53
3.2.	Exemplo inicial.....	53
3.2.1.	Histograma	54
3.3.	Distribuições disponíveis no SIMUL8.....	55
3.4.	Aplicação prática do exemplo inicial	56
3.4.1.	Informando as distribuições ao SIMUL8.....	56
3.4.2.	Utilizando distribuições com <i>labels</i>	58
3.4.3.	Distribuição <i>Bounded</i> : limitando os valores máximos e mínimos da distribuição	63
3.4.4.	<i>Combination</i> : criando uma nova distribuição a partir da soma de outras	64
3.4.5.	<i>Time Dependent</i> : criando distribuições que se alteram com o tempo	66
3.4.6.	<i>Time absolute</i> : distribuições "com hora marcada".....	68
3.4.7.	<i>Label Based</i> : armazenando o nome da distribuição em um <i>label</i>	69
3.4.8.	<i>External</i> : acessando distribuições externas ao SIMUL8 (Excel ou Visual Basic)	73
3.5.	Algumas ferramentas adicionais para o usuário avançado	75
3.5.1.	Simulação sem aleatoriedade.....	75
3.5.2.	Manipulando o gerador de números aleatórios do SIMUL8.....	75
3.5.3.	Trabalhando com arquivos de chegada do tipo <i>trace</i>	76
Capítulo 4	Direcionamento de Entidades	78
4.1.	Introdução.....	78
4.2.	Direcionamentos de entrada básicos: <i>Priority</i> , <i>Collect</i> e <i>Passive</i>	78
4.3.	Outras disciplinas de entrada (<i>Expired Only</i> , <i>Oldest</i> , <i>Youngest</i> , <i>Longest</i> e <i>Circulate</i>).....	81
4.4.	Disciplinas de saída: <i>Circulate</i> , <i>Uniform</i> , <i>Pervent</i> , <i>Priority</i> , <i>Label</i> , <i>Shortest Queue</i> , <i>Passive</i> e <i>Batching</i>	83
Capítulo 5	Variáveis e noções de <i>Visual Logic</i>	89
5.1.	Variáveis	89
5.2.	Tipos e criação de variáveis no SIMUL8.....	89

5.3. Noções básicas de <i>Visual Logic</i>	91
5.4. Exemplo com variáveis e <i>Visual Logic</i>	92
Capítulo 6 Refinamentos Gráficos e Animação	107
6.1. Introdução.....	107
6.2. Fundo de tela.....	108
6.3. Objetos de simulação	109
6.4. Entidades.....	114
Capítulo 7 Funcionalidades específicas.....	119
7.1. Introdução.....	119
7.2. Exemplo inicial.....	119
7.3. Formação de lotes ou bateladas	119
7.4. Criando <i>labels</i>	120
7.5. Utilizando os <i>labels</i> (<i>Label Actions</i>)	121
7.6. Disponibilidade de recursos e eficiência dos <i>Activities</i>	123
7.7. Funcionalidades de filas	127
7.8. <i>Set-up</i> (<i>Activity Change Over</i>).....	128
7.9. <i>Conveyor</i> (somente no SIMUL8 Professional)	129
Capítulo 8 Estudos de Caso.....	132
8.1. Introdução.....	132
8.2. Caso 1: movimentação de bobinas de aço pré-pintado	132
8.2.1. Fluxo do processo de produção	132
8.2.2. Descrição do sistema.....	133
8.2.1. Construção do modelo computacional.....	135
8.2.2. Executando a simulação e analisando os resultados obtidos.....	137
8.2.3. Análise final	140
8.2.4. Atividades complementares	140
8.3. Caso 2: modelagem e simulação do serviço de suporte a usuários de um ERP.....	141
8.3.1. Introdução	141
8.3.2. A empresa estudada.....	141

8.3.3.	Descrição do sistema.....	142
8.3.4.	Construção do modelo computacional.....	144
8.3.5.	Executando a simulação e analisando os resultados obtidos.....	146
8.3.6.	Atividades complementares	151
I.1.	Introdução.....	152
I.2.	Pré-processamento	152
I.3.	Entrada de dados	152
I.4.	Histograma.....	154
I.5.	Estatística descritiva	154
I.6.	Ajuste (<i>fitting</i>)	155
I.7.	Análise de correlação.....	157
I.8.	Definindo a distribuição mais adequada	158

Capítulo 1 INTRODUÇÃO

O SIMUL8 é, atualmente, um dos *softwares* de simulação com o maior número de licenças vendidas no mundo. Isto se deve, principalmente, ao seu preço de venda relativamente baixo e facilidade de uso. O desenvolvimento do SIMUL8 teve início na década de 1990, na Universidade de Strathclyde (Escócia), com a finalidade de ensino de simulação. Devido ao sucesso do programa como ferramenta de ensino de simulação, a Simul8 Corporation iniciou sua comercialização como ferramenta profissional para projetos de simulação.

Este capítulo contém um tutorial rápido de uso do SIMUL8, em que o leitor construirá um modelo básico de simulação e entrará em contato com as principais funcionalidades do programa.

1.1. Exemplo inicial

Vamos criar um modelo de simulação que represente o funcionamento de um posto bancário que possua apenas um caixa e fila única de atendimento (Figura 1.1).

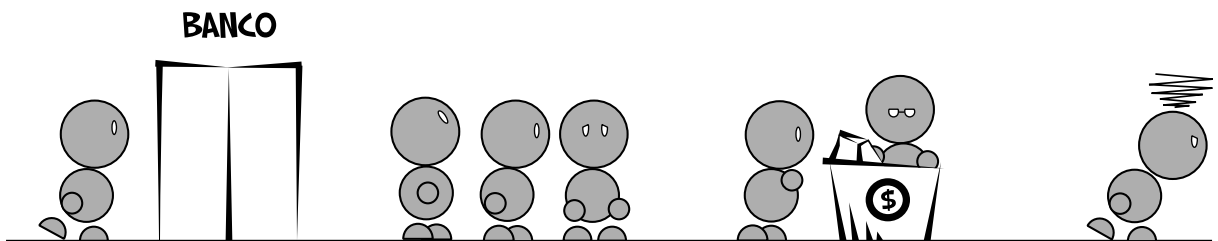


Figura 1.1 - Posto de atendimento bancário, formado por uma fila de clientes, um caixa de atendimento e um ponto de saída.

Neste posto, cada cliente é atendido pelo caixa em um tempo médio de 6 minutos e um novo cliente chega ao posto em busca de atendimento, em média, a cada 10 minutos (ou seja, o posto recebe uma média de 6 clientes por hora). Os intervalos de tempo entre as chegadas e os tempos de atendimento de cada cliente são aleatórios e seguem distribuições de probabilidade exponenciais¹. O posto funciona continuamente durante 6 horas por dia e a disciplina de atendimento é do tipo “o primeiro a chegar é o primeiro a ser atendido”: assim, não se pode “furar a fila” e não existe qualquer tipo de atendimento preferencial. Estas informações são suficientes para construirmos nosso primeiro modelo de simulação no SIMUL8.

1.2. Iniciando o SIMUL8

Abra o SIMUL8. A partir da tela inicial do programa, podemos abrir um modelo de simulação existente ou criar um novo modelo, que é o que nos interessa no momento: selecione a opção *Create New Simulation*, como mostrado na Figura 1.2. Feito isso, o SIMUL8 exibirá a tela mostrada na Figura 1.3, em que destacamos o menu de comandos, a barra de botões de acesso rápido, a janela do relógio de simulação e a janela de simulação, onde os modelos de simulação propriamente ditos são construídos.

¹ Caso não esteja familiarizado com distribuições de probabilidade, você pode consultar o Capítulo 3 deste livro ou algum material específico sobre estatística e probabilidades.

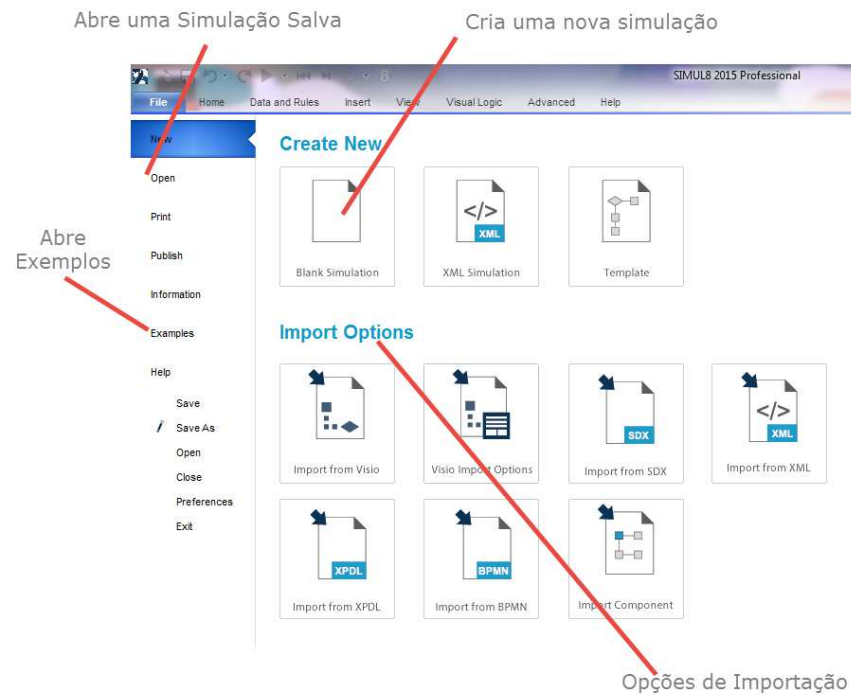


Figura 1.2 - Tela inicial do SIMUL8.

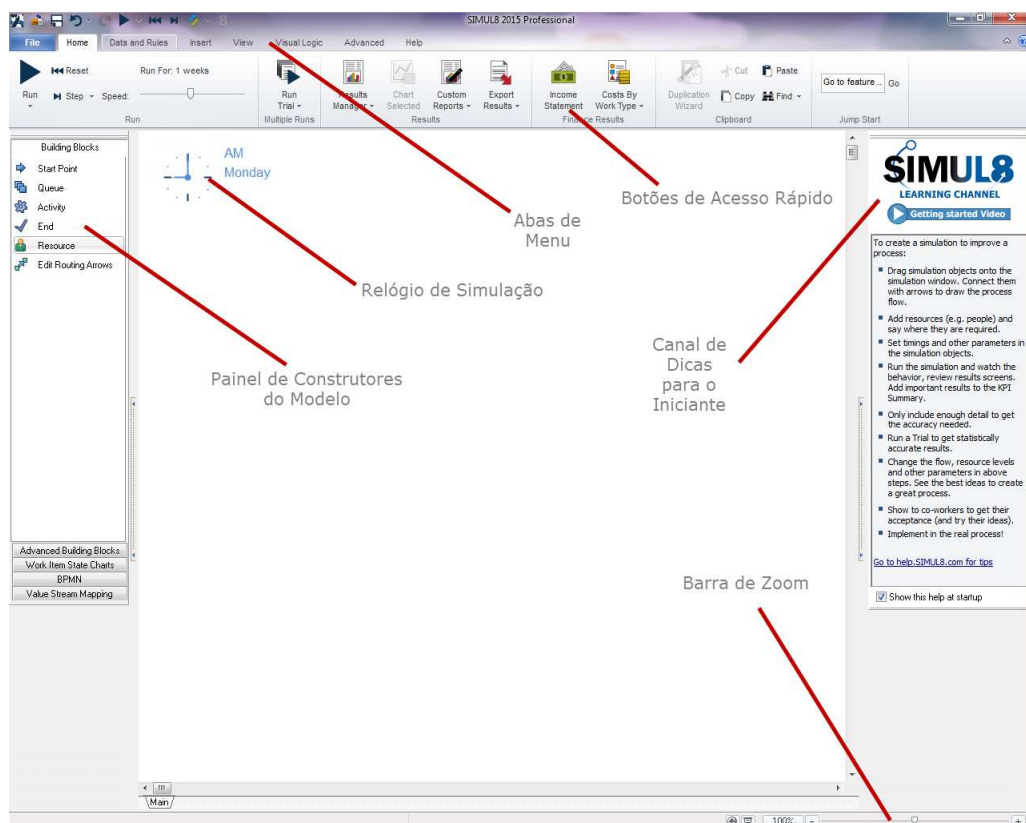


Figura 1.3 - Tela principal do SIMUL8.

1.3. Construindo o modelo

Para a construção de um modelo no SIMUL8, deve-se ter em mente alguns conceitos básicos. A maioria dos modelos podem ser construídos somente com 6 blocos construtores básicos (ou objetos de simulação):

- Entidade (*Work Item Type*);
- Chegada (*Start Point*);
- Fila (*Queue*);
- Processo ou atividade (*Activity*);
- Recurso (*Resource*);
- Saída (*End*).

A idéia geral em um modelo de simulação é a seguinte: *entidades chegam* ao sistema, aguardam em *filas* caso seja necessário, realizam *atividades* (que, por sua vez, podem depender de *recursos*) e depois *saem* do sistema.

Para inserir objetos no modelo deve-se clicar no botão do bloco construtor correspondente (no painel de bloco construtores do modelo) e, em seguida, na janela de simulação ou arrastá-los do painel de blocos construtores para a janela de simulação. Se a opção de conexão automática estiver habilitada (que é o padrão do programa), o SIMUL8 irá inserir automaticamente as ligações entre os objetos inseridos. Assim, para construir o modelo do posto bancário, faça o seguinte:

- Clique no botão  (*Start Point*) e, em seguida, na janela de simulação;
- Clique no botão  (*Queue*) e, em seguida, na janela de simulação;
- Clique no botão  (*Activity*) e, em seguida, na janela de simulação;
- Clique no botão  (*End*) e, em seguida, na janela de simulação.

Opcionalmente, pode-se arrastar e soltar os objetos do painel de blocos construtores para a janela de simulação.

A janela de simulação deverá apresentar um aspecto semelhante ao que é mostrado na Figura 1.4. Compare o resultado obtido com o desenho do exemplo na Figura 1.5: o ponto de entrada (*Start Point*) representa a entrada do sistema, ou seja, o local por onde os clientes entram no posto. Os clientes aguardam na fila (*Queue*) caso algum cliente esteja sendo atendido (*Activity*). Após o atendimento, os clientes saem imediatamente do posto bancário pelo ponto de saída (*End*).

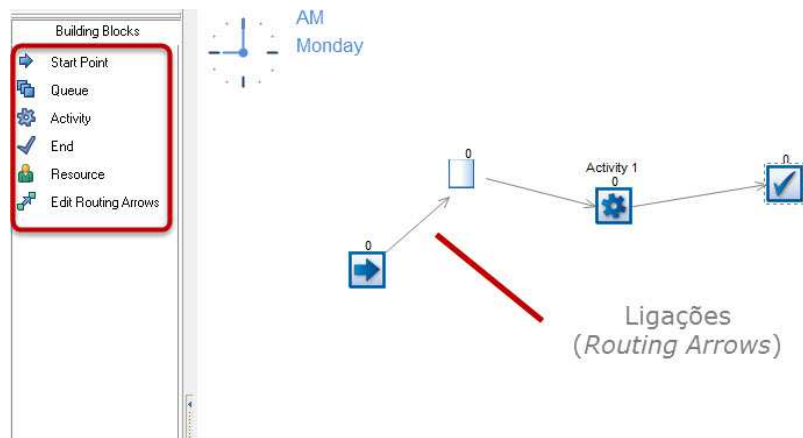


Figura 1.4 - Aspecto inicial do modelo do posto bancário.

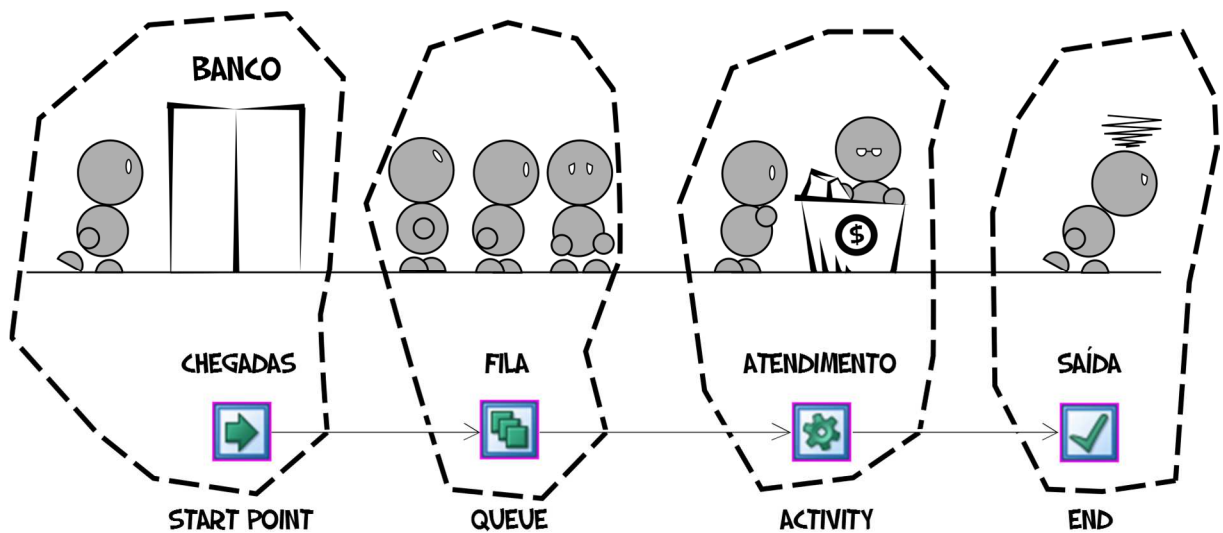


Figura 1.5 - Comparação entre as duas representações do sistema: gráfica e blocos construtores do SIMUL8.

Por questões estéticas, você poderá querer movimentar os blocos do modelo — basta clicar e arrastar o objeto desejado com o *mouse* — ou alinhá-los automaticamente: selecione os blocos e clique com o botão direito do *mouse* sobre qualquer ponto da janela de simulação (Figura 1.6). No menu de contexto que se abrirá, existem duas opções: a opção *Align*², que alinha os objetos selecionados e a opção *Space Equally*, que distribui os objetos selecionados a uma mesma distância entre si. Use-as!

² Se os objetos já estiverem alinhados, essa opção não aparecerá no menu de contexto.

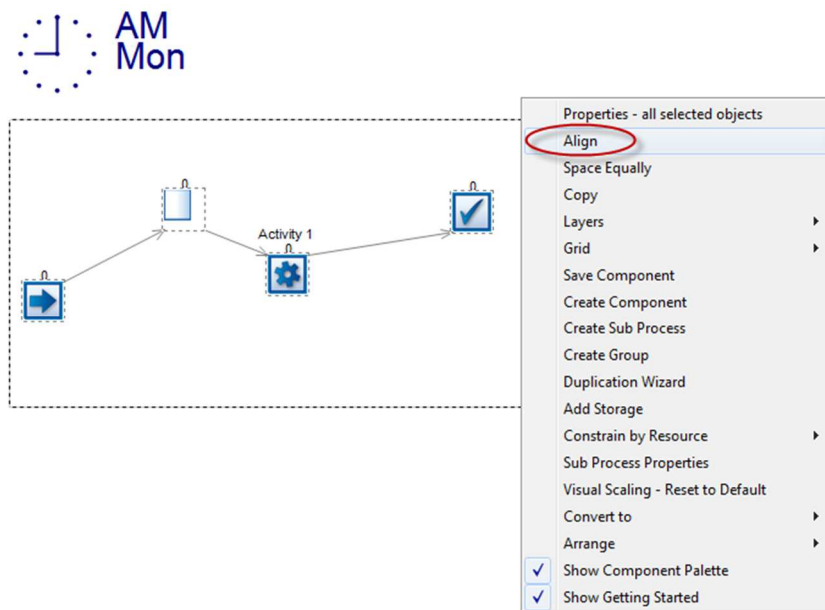


Figura 1.6 – Menu de contexto com opções de alinhamento e espaçamento de objetos selecionados.

É possível inserir as ligações que eventualmente estiverem faltando usando o *Routing Drawing Mode*: clique no botão correspondente no painel de ferramentas de conexão (Figura 1.7) para ativar o modo de edição de ligações (o cursor se transforma em uma seta apontando para a direita). Em seguida, clique nos objetos de origem e destino da ligação. Para sair do modo de edição, clique novamente no botão *Route Drawing Mode*. Para apagar uma ligação errada basta selecioná-la com o *mouse* (com o *Routing Drawing Mode* desativado) e pressionar a tecla *Delete*.

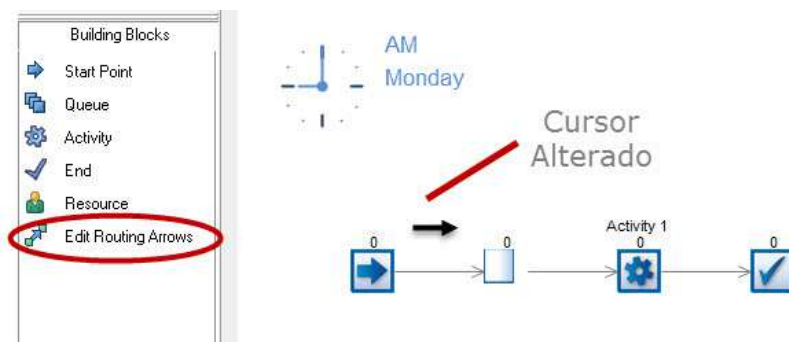


Figura 1.7 – Criação de rotas com a opção *Routing Drawing Mode*.

Outra forma de se inserir objetos e ligações no modelo é por meio de uma ferramenta chamada *Paddle* que, quando está ativada, se torna visível quando um ponto de entrada, fila ou atividade são selecionados (Figura 1.8). Clicando-se em uma das 4 abas da ferramenta é possível inserir outros objetos de simulação ligados (ou não) ao objeto original. É possível (em função da versão do SIMUL8) que você tenha que ativar a ferramenta *Paddle*, caso as abas não sejam exibidas: clique em *File* | *Preferences* | *Options* e selecione a opção *Paddle On*.

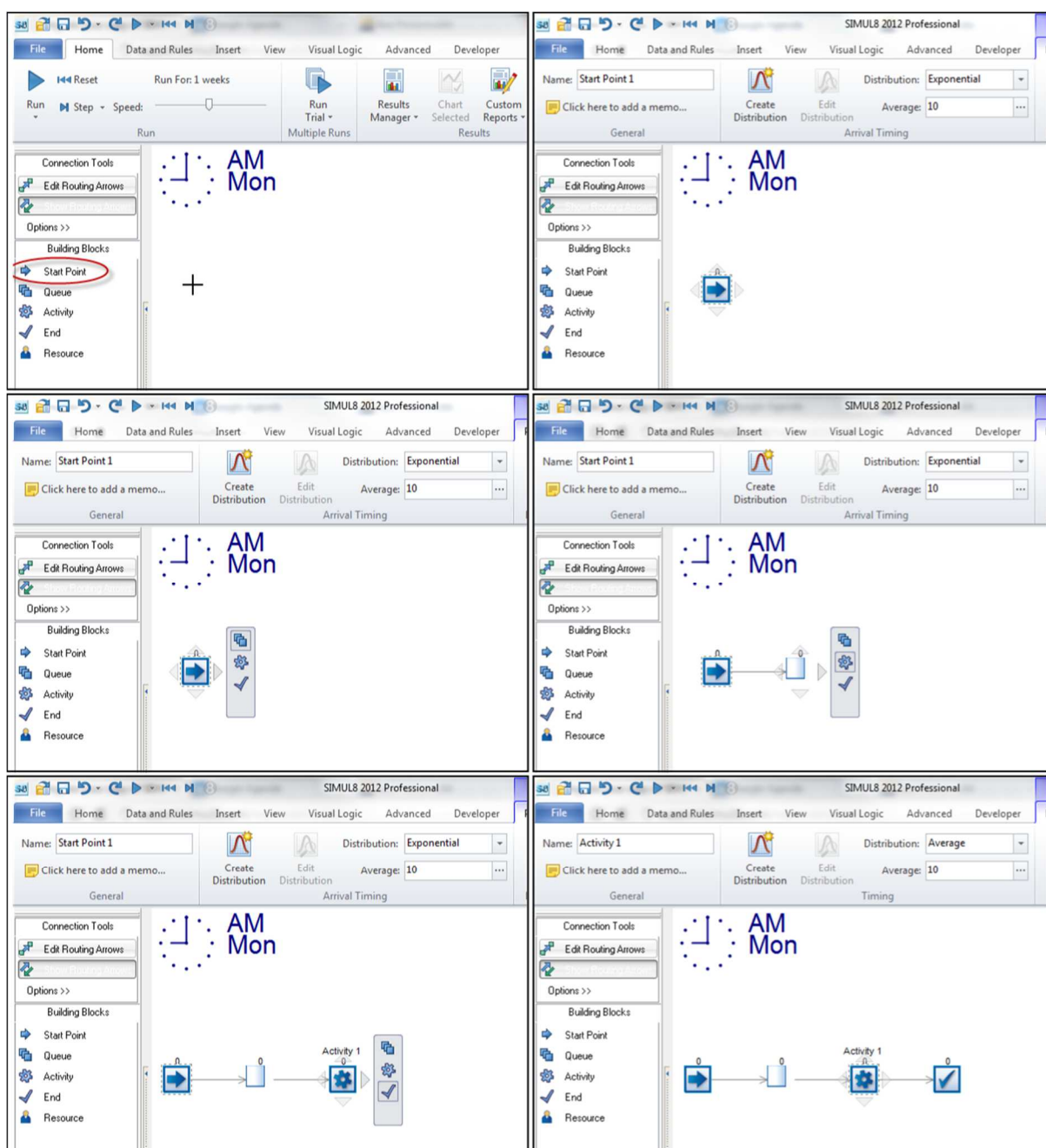


Figura 1.8 - Aspecto inicial do modelo de simulação.

1.4. Ajustando os parâmetros dos objetos

Antes de simular o modelo é preciso ajustar os processos de chegada e de atendimento dos clientes. Até a versão 2010 do SIMUL8 estes ajustes eram feitos exclusivamente na *janela de parâmetros* dos objetos correspondentes (*Start Point* e *Activity*). Estas janelas ainda estão disponíveis na versão 2015 do SIMUL8 (e nós vamos mostrá-las na sequência do texto), mas alguns ajustes podem ser feitos diretamente no menu principal do programa, como mostraremos em breve. Ao longo dos capítulos deste livro vamos usar a forma tradicional: para qualquer objeto inserido na janela de simulação, o acesso à janela de propriedades será feito dando-se um duplo-clique no objeto.

a) Abra a janela de propriedades do ponto de chegada de clientes (*Start Point*), clicando duas vezes no objeto. Altere o título do objeto de *Start Point 1* para *Chegada* e ajuste o intervalo de tempo médio entre chegadas para 10 minutos, com distribuição exponencial (Figura 1.9).

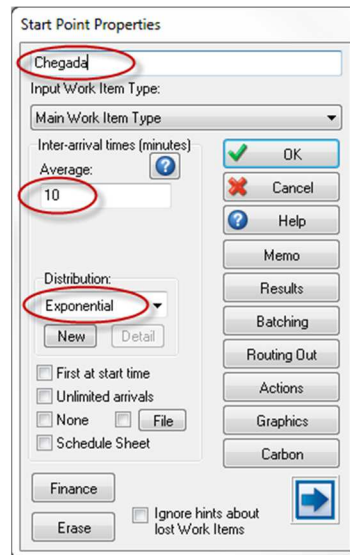


Figura 1.9 - Propriedades do *Start Point*.

O SIMUL8 mostrará uma breve explicação sobre cada botão e opção das janelas de parâmetros caso você posicione o cursor do *mouse* sobre o elemento por alguns instantes (Figura 1.10).

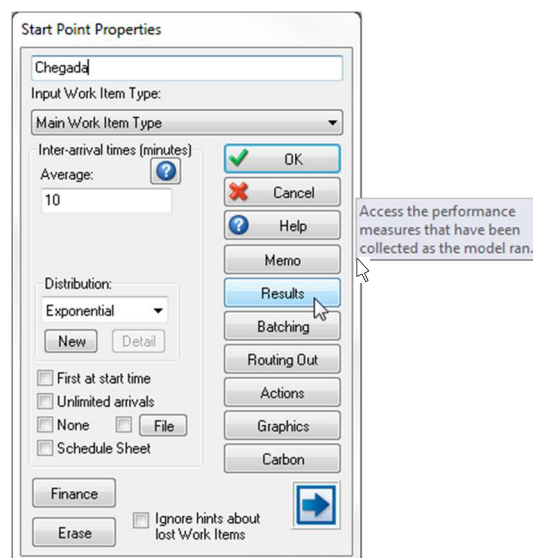


Figura 1.10 - Propriedades do *Start Point*: ajuda sobre o botão *Results*.

Ainda na janela de parâmetros do *Start Point*, clique em *Graphics* | *Title* | *Show Title on Simulation Window* para que o nome do objeto (*Chegada*) seja exibido em tela (Figura 1.11). Conclua as alterações, clicando em *OK* até fechar todas as caixas de diálogo abertas. Como um método alternativo, após a seleção do objeto, clique na aba *Graphics* e ative a opção “*Show*” no canto superior esquerdo (logo abaixo da aba “*File*”).

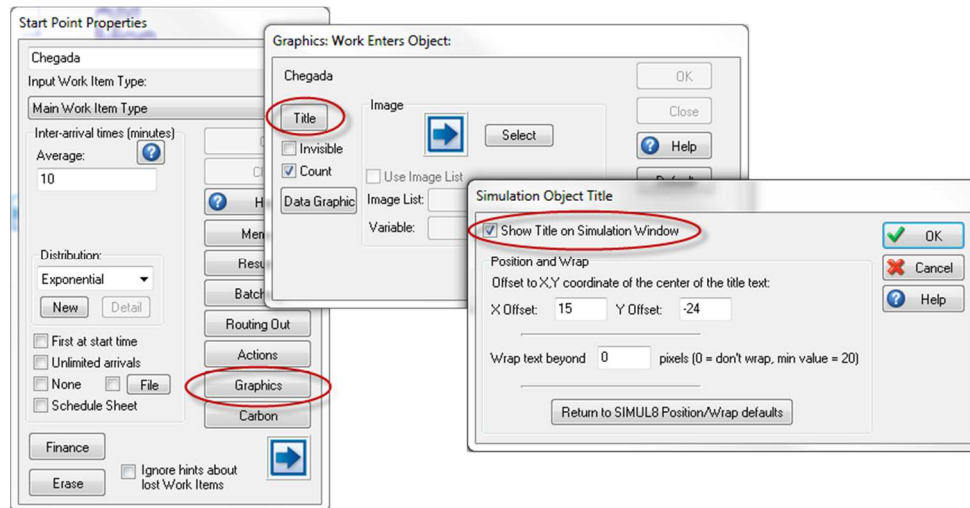


Figura 1.11 - Propriedades gráficas do *Start Point*.

O SIMUL8 cria automaticamente uma *classe de entidades* chamada *Main Work Item Type* para representar os clientes que entram e saem do sistema. Apesar de ser possível alterar o nome desta classe, ou mesmo criar outras classes, isto não é necessário na maioria das situações e não será abordado neste livro. Como discutiremos mais adiante, é mais simples e prático utilizarmos atributos (ou *labels*) para representarmos diferentes tipos de entidades.

b) Abra a janela de propriedades da fila de espera (*Queue Properties*) e altere seu nome de *Queue 1* para “Fila de espera” (Figura 1.12). Ainda nesta janela, clique no botão *Graphics* e selecione a opção de exibição *Queue* para que o SIMUL8 mostre uma fila “indiana” de clientes durante a simulação (a opção padrão é *Graphic Tank*, que mostra um “tanque” se enchendo ou esvaziando dependendo do tamanho da fila). Por fim, ainda na janela de opções gráficas, clique em *Title* | *Show Title on Simulation Window* para que o nome do objeto seja mostrado em tela. Em outra ocasião, explore as outras opções de estilo de exibição da fila (*Display Style*).

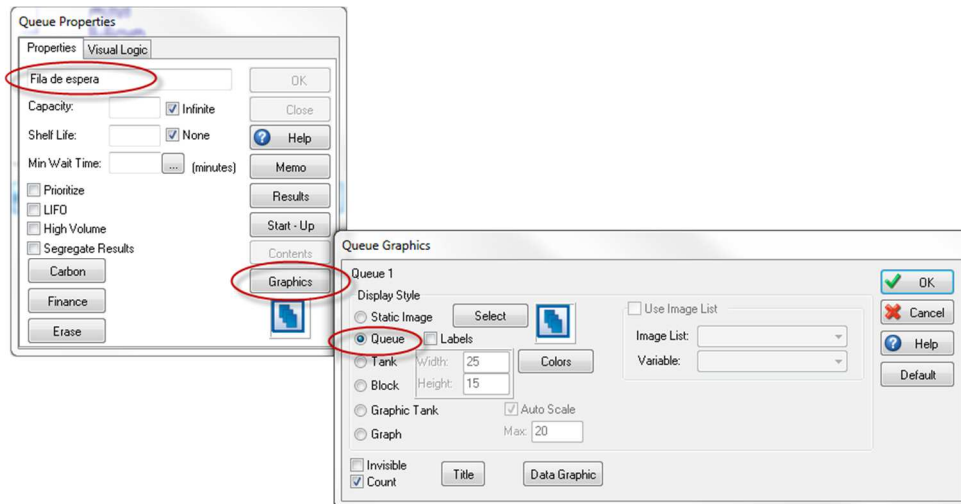


Figura 1.12 - Propriedades gerais e gráficas da fila de espera.

c) Abra a janela de propriedades da atividade (*Activity*) e altere seu título de “*Activity 1*” para “Atendimento”. Ajuste também os parâmetros do processo: distribuição exponencial, com tempo médio de 6 minutos (Figura 1.13). Nas propriedades gráficas da atividade (botão *Graphics*), verifique que a opção *Show Title* é ativada por padrão.

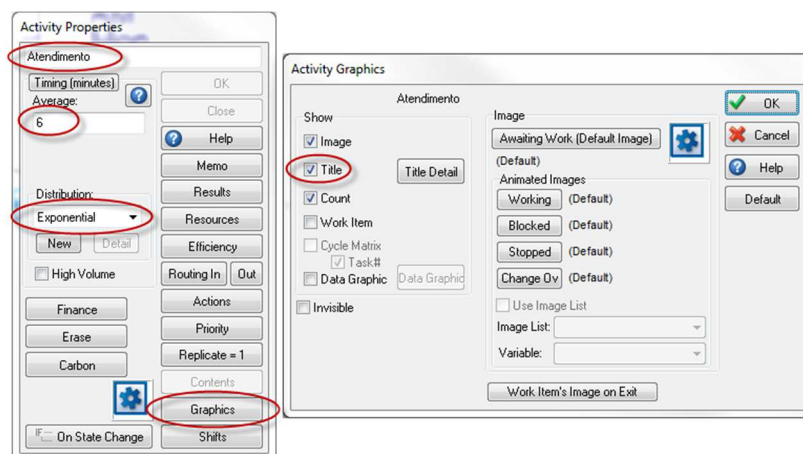


Figura 1.13 - Propriedades do atendimento.

d) Abra a janela de propriedades do ponto de saída dos clientes (*End*) e altere seu título de “*End 1*” para “Saída” (Figura 1.14). Ainda com a janela de propriedades aberta, clique em *Graphics* | *Title* | *Show Title on Simulation Window* para habilitar a exibição do nome do objeto.



Figura 1.14 - Propriedades do ponto de saída.

Um último ajuste recomendável é o da exibição permanente das linhas de ligação: clique no botão de opções do botão *Show/Hide Route Arrows* e selecione a opção *Always Show Routing Arrows* (Figura 1.15).

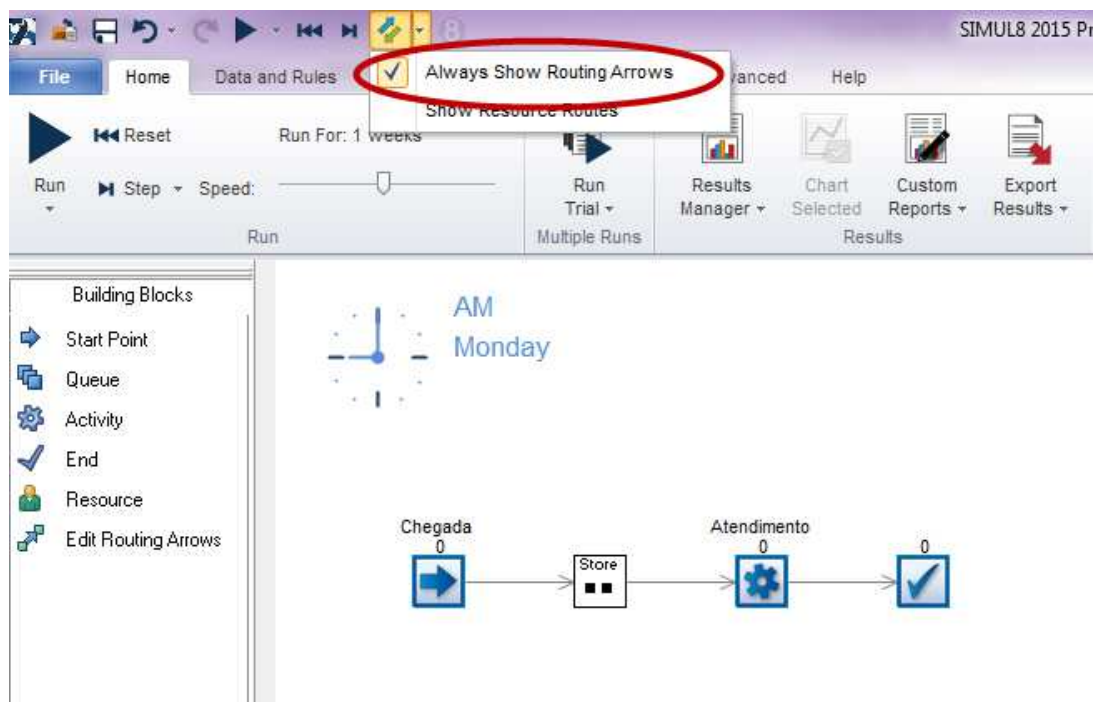


Figura 1.15 - Aspecto final do modelo do posto bancário.

Com isso, o modelo de simulação está pronto! Perceba que somente as alterações feitas nos intervalos de tempo entre chegadas e nos tempos de atendimento eram essenciais: as outras não afetam a simulação, mas facilitam o entendimento do modelo.

Antes de prosseguir, faça uma verificação rápida de quais configurações poderiam ter sido feitas diretamente pelo menu principal do SIMUL8, por exemplo, selecionando o *Start Point* (Figura 1.16). Estas opções, que são exibidas dinamicamente quando um objeto é selecionado, são úteis para os ajustes mais comuns nas propriedades de um objeto, mas o conjunto completo de opções só está disponível na janela de configurações própria.

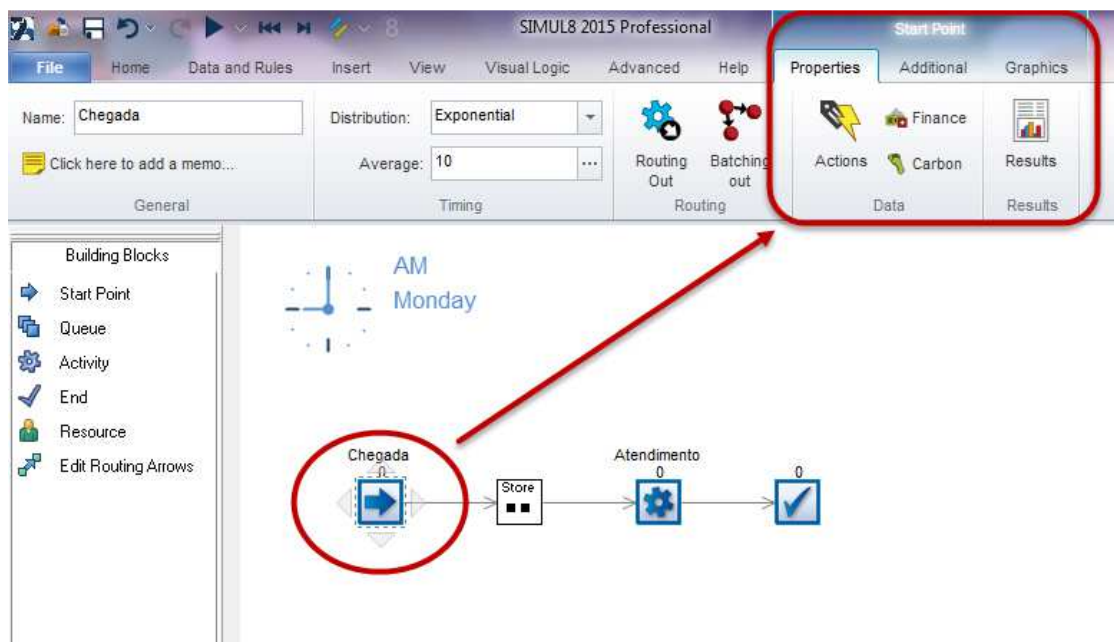


Figura 1.16 - Acesso rápido às propriedades de um objeto.

1.5. Preparação para a simulação

1.5.1. Tempos de deslocamento

Por padrão, o SIMUL8 contabiliza o tempo gasto no deslocamento de uma entidade entre os objetos do modelo de acordo com o comprimento (em *pixels*) das ligações. Como esses tempos de deslocamento geralmente são inseridos no modelo por meio de *Activities* (ou matrizes de tempo de deslocamento), este tipo de representação não é muito utilizado. Assim, antes de simular o funcionamento do modelo, clique em *File | Preferences* e, na aba *Distance*, clique no botão *Set to Zero* da seção *Travel Times* (Figura 1.17). Ao confirmar a operação, todas as ligações do modelo terão o tempo associado de percurso igual a zero.

Esta operação é muito importante porque os tempos de deslocamento não planejados podem afetar de forma significativa os resultados de uma simulação. Aliás, recomendamos que você faça isso **antes** de começar a construir qualquer modelo para não correr o risco de se esquecer!

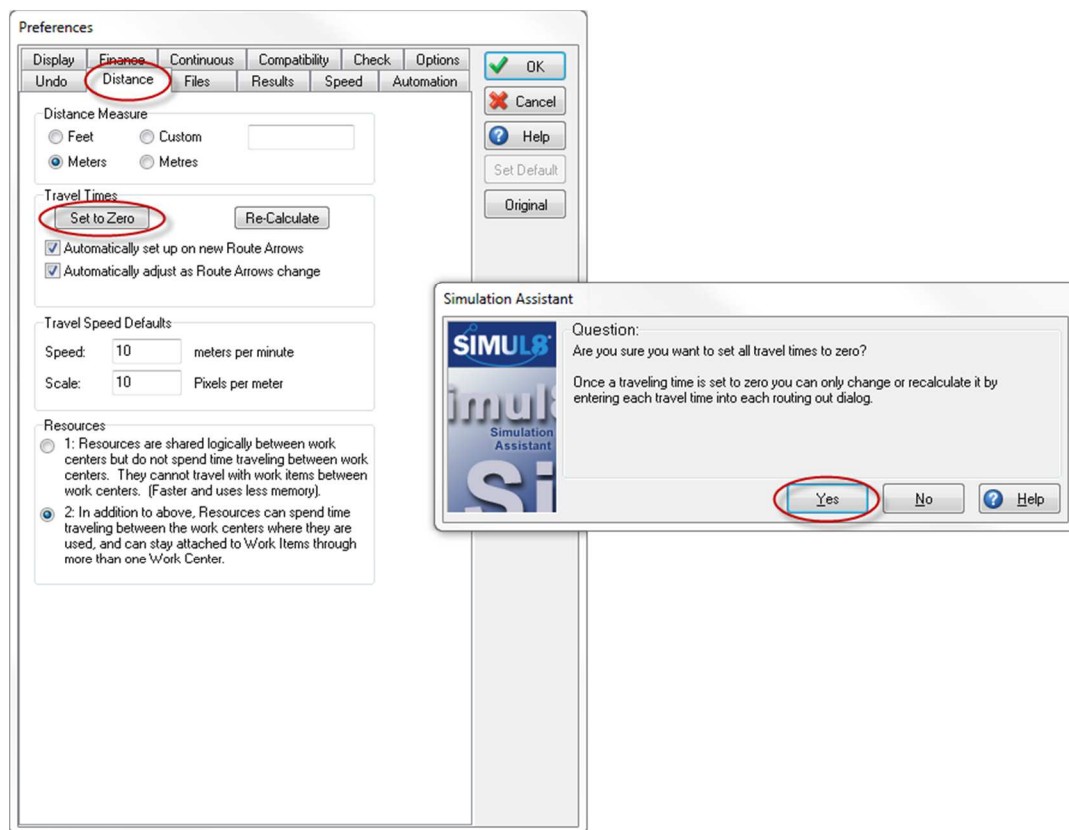


Figura 1.17 - Janela de opções de simulação: tempos de deslocamento.

1.5.2. Ajuste da unidade de tempo

Todos os objetos de um modelo de simulação no SIMUL8 utilizam uma mesma unidade de tempo. No caso do nosso modelo, a unidade mais adequada é o minuto, que (por acaso) é a unidade padrão do SIMUL8: clique em *Data and Rules* | *Properties* ou dê um duplo-clique no relógio de simulação para comprovar isso (Figura 1.18).

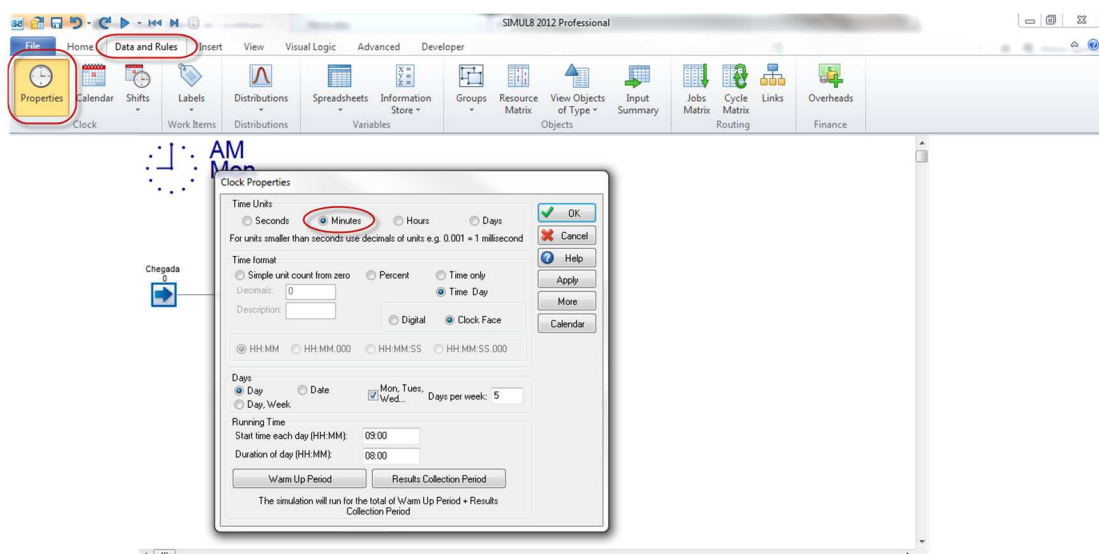
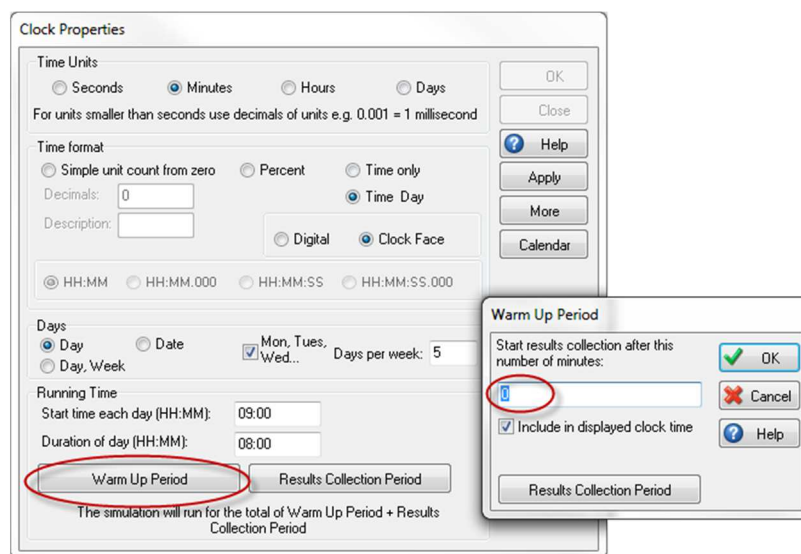


Figura 1.18 - Janela de propriedades do relógio de simulação.

1.5.3. Ajuste do tempo de simulação

Antes de executar uma simulação é preciso definir o período total de tempo a ser simulado, que é formado pela soma de dois períodos consecutivos. No primeiro, chamado de período de aquecimento (*Warm Up Period*), nenhum resultado da simulação será coletado para a compilação de estatísticas. O objetivo deste período é eliminar a influência das condições iniciais do sistema sobre os resultados da simulação³. Somente no segundo período (*Results Collection Period*) é que serão recolhidos resultados para a geração de estatísticas, como o tempo médio de espera em fila, por exemplo. No exemplo do posto bancário, por simplicidade, vamos utilizar um período de aquecimento = 0 e um período de coleta de resultados de 100 horas de operação = 6.000 minutos. Os ajustes são feitos clicando-se nos botões na parte inferior da janela de propriedades do relógio: pelo botão *Warm Up Period* (Figura 1.19) você informa o tempo de aquecimento ou *Warm up* e por meio do botão *Results Collection Period* (Figura 1.20) você informa o tempo de duração da simulação. Após ter feito os ajustes, feche a janela de propriedades, clicando no botão OK. As outras opções da janela de propriedades do relógio não são importantes neste ponto do nosso estudo e, portanto, não serão discutidas.

Figura 1.19 - Definição do tempo de aquecimento ou *Warm Up*.

³ Para maiores detalhes, consulte o livro “Modelagem e Simulação de Eventos Discretos” de Leonardo Chwif e Afonso C. Medina - Capítulo 6.

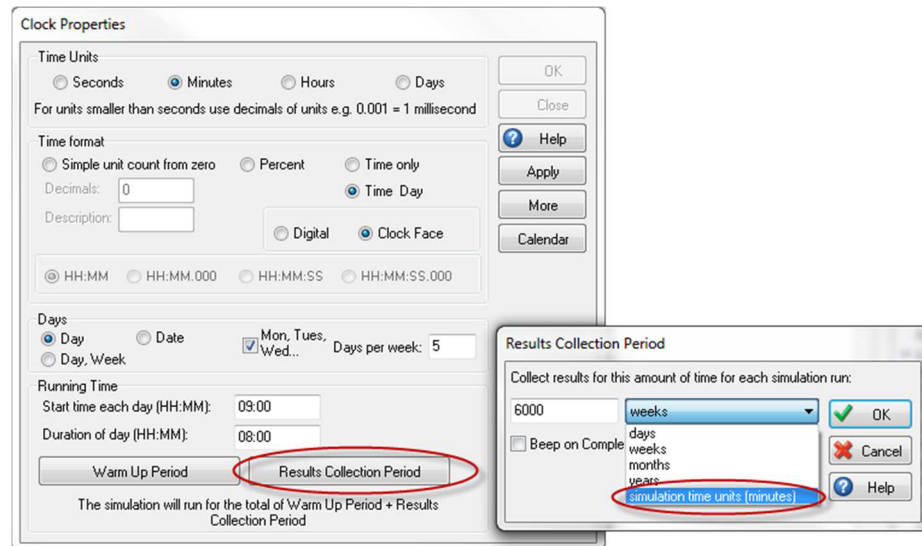


Figura 1.20 - Definição da duração da simulação ou *Results Collection Period*.

1.6. Executando a simulação

Pressione o botão *Run* para iniciar a simulação e ajuste a velocidade da animação (Figura 1.21): com o botão de controle totalmente à direita (velocidade máxima), a simulação será executada, mas a animação não será exibida; com o botão totalmente à esquerda, a simulação será encerrada⁴. Utilize um ajuste de velocidade que permita visualizar adequadamente os clientes entrando e saindo do posto bancário (não sem antes aguardar um pouco em fila).

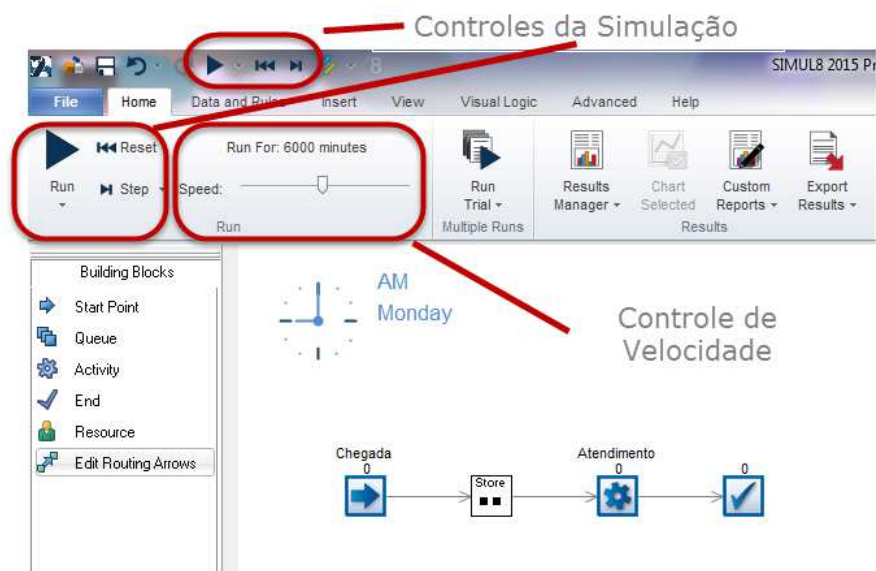


Figura 1.21 - Aspecto do modelo durante a simulação.

Durante a simulação, a janela de simulação terá um aspecto semelhante ao que é mostrado na Figura 1.22. Os números acima dos objetos indicam as quantidades de clientes que já entraram no sistema (32), que estão na fila de espera (1),

⁴ Cuidado: se você reduzir demais a velocidade de animação poderá ter que esperar muito tempo para "retomar o controle" da velocidade da simulação!

em atendimento (0) e o total de clientes que já saíram do sistema (30). Adicionalmente, a figura mostra (apenas para efeitos visuais, já que os deslocamentos acontecem instantaneamente) uma entidade em deslocamento para a saída.

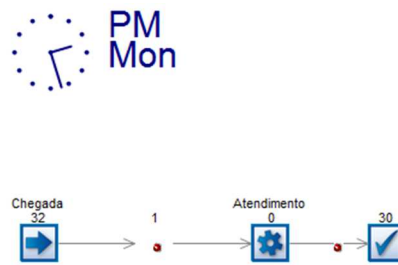


Figura 1.22 - Simulação do exemplo em andamento.

Você poderá clicar novamente no botão *Run* durante a execução da simulação para interrompê-la. Clicando novamente no botão, a simulação será retomada a partir do ponto em que parou. Você saberá que a simulação terminou quando a animação dos objetos e do relógio parar⁵. Para recolocar o sistema em suas condições iniciais, clique no botão *Reset*.

Sempre que se inicia uma simulação pelo botão *Run*, a sequência de números aleatórios utilizados pelo SIMUL8 será **a mesma** que foi utilizada na última simulação. Isto quer dizer que, se o modelo não sofrer nenhuma alteração, os resultados obtidos serão exatamente os mesmos! Na versão *Professional* do SIMUL8, para executar a simulação com uma nova sequência de números aleatórios, clique na seta abaixo do botão *Run* (vide botão de opções adicionais na Figura 1.21 e escolha a opção *Run simulation with new random numbers*. Para todas as versões do SIMUL8, o controle dos números aleatórios pode ser feito manualmente, como descrito na seção 3.5.2.

Por fim, você poderá executar uma simulação passo a passo (ou seja, parando a execução a cada evento) clicando no botão *Step*, uma opção muito útil para verificar se há erros no modelo.

1.7. Consulta de resultados: uma replicação⁶

Para consultar os resultados obtidos após a simulação basta dar um duplo-clique em algum objeto do modelo e clicar no botão *Results*: os resultados mostrados dependerão do objeto selecionado.

Importante: os valores mostrados a seguir poderão ser diferentes daqueles que você, leitor, vai obter em seu modelo devido aos números aleatórios usados em cada computador.

1.7.1. Resultados dos pontos de entrada (*Start Point*)

Os principais resultados dos pontos de entrada são: o número de clientes que chegaram ao sistema e o número de clientes “perdidos” — ou seja, que não entraram no sistema por encontrá-lo bloqueado (Figura 1.23). Por exemplo, se a fila de espera tivesse capacidade limitada, qualquer chegada que ocorresse com a “fila cheia” daria origem a um cliente “perdido”.

⁵ Na janela de ajuste do tempo de simulação é possível habilitar um *beep* que indique o final da simulação.

⁶ O conceito de replicação será discutido em detalhes na seção 1.8.

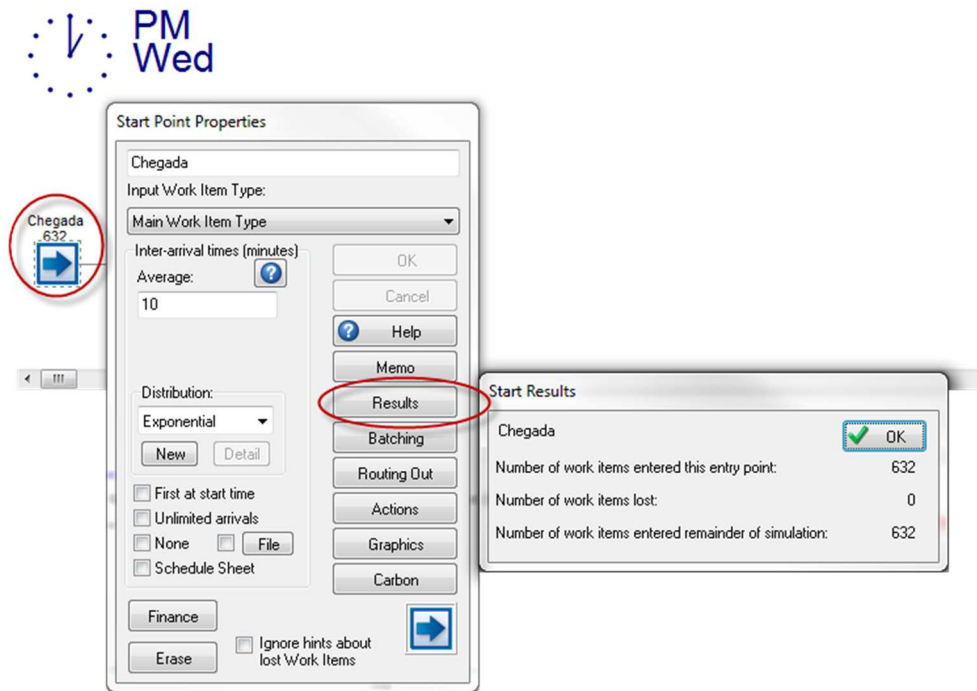


Figura 1.23 - Resultados da simulação: chegada.

1.7.2. Resultados das filas de espera (Queue)

Existem dois grupos de informação disponíveis nas filas de espera: um sobre a quantidade de clientes e outro sobre os tempos de espera (Figura 1.24). Na seção *Number of work items in this storage*, é possível consultar o número de entidades presentes na fila no instante em que a simulação foi interrompida (*Currently*), os números mínimo, médio e máximo de entidades que estiveram na fila (*Minimum*, *Average* e *Maximum*) e o número total de entidades que entraram na fila durante a simulação (*Total Entered*).

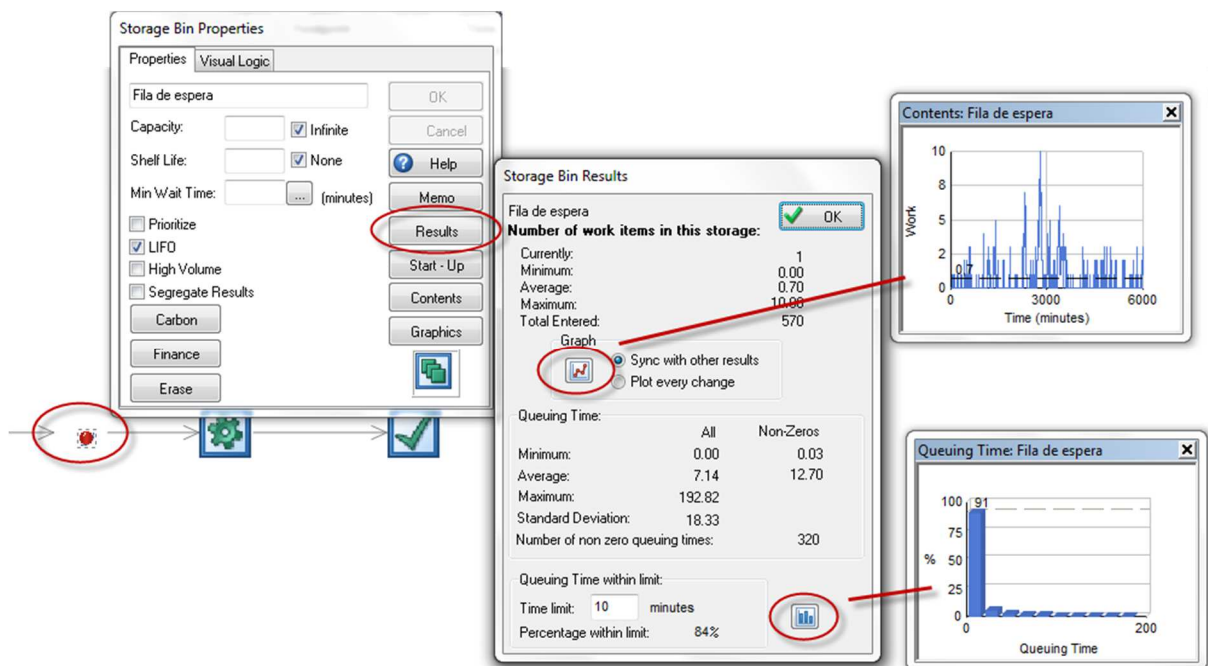


Figura 1.24 - Resultados da simulação: fila de espera.

O botão *Graph* cria um gráfico com a evolução do número de clientes em fila ao longo da simulação, com o valor médio indicado por uma linha tracejada (Figura 1.25).

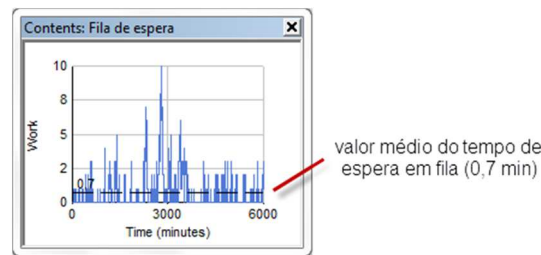


Figura 1.25 - Gráfico do número de elementos em fila ao longo do tempo.

Sobre o tempo de espera (*Queuing Time*), são fornecidos os valores mínimo, médio e máximo (*Minimum, Average e Maximum*) e o respectivo desvio padrão (*Standard Deviation*) para todas as entidades que passaram por esta fila (coluna *All*). Os tempos associados às entidades que não tiveram que aguardar na fila (porque o servidor estava livre) são indicados na coluna *Non-Zero*: o total de entidades com esta característica é dado por *Number of non-zero queuing times*. É possível, ainda, obter um histograma dos tempos de espera, clicando no botão *Histogram* (Figura 1.26).

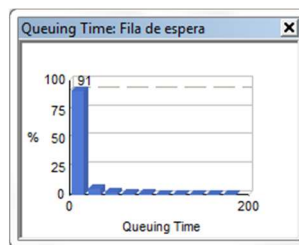


Figura 1.26 - Histograma dos tempos de espera em fila.

Por fim, é possível consultar a percentagem de entidades que permaneceram em fila por um período de tempo igual ou inferior a um limite especificado (*Percentage within limit*) - vide Figura 1.24.

1.7.3. Resultados das atividades (*Activity*)

Nas atividades, é possível consultar dois grupos de informações (Figura 1.27). O primeiro, sobre o total de entidades processadas (*Number of work items*), apresenta o número de entidades que estavam em processamento quando a simulação foi interrompida, além dos valores mínimo, médio, máximo e total de entidades processadas (*Currently in Activity, Minimum, Average, Maximum e Completed Jobs*).

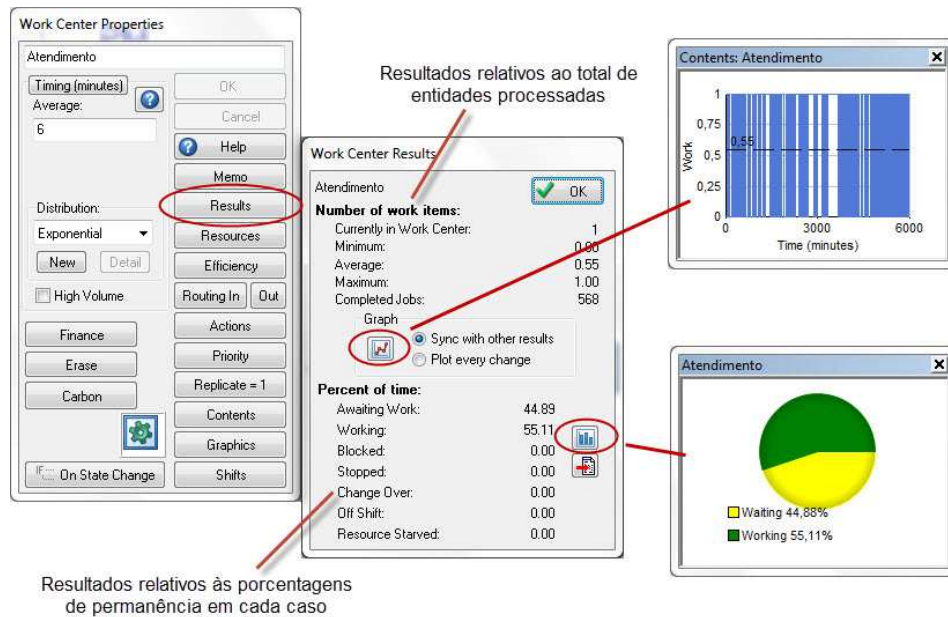


Figura 1.27 - Resultados da simulação: atendimento.

O segundo grupo de informações, sobre porcentagens de tempo (*Percent of Time*), só será apresentado quando a atividade corresponder a um único servidor (*Replicate* = 1), que é o caso do nosso exemplo. Neste conjunto de resultados são apresentadas as porcentagens de tempo em que a atividade esteve livre (*Awaiting Work*), em processamento (*Working*), bloqueada — porque a entidade processada não pôde “seguir em frente” (*Blocked*) - e parada por avarias (*Stopped*).

Por padrão, cada *Activity* no SIMUL8 representa o processo de execução de uma atividade, “sobre” uma entidade: enquanto a atividade não terminar, o objeto *Activity* não poderá processar outra entidade, que deverá aguardar em fila. No entanto, objetos *Activity* podem ser configurados para representar a execução de mais de uma atividade em paralelo, por meio de uma propriedade chamada *Replicate*. Por exemplo, com apenas um objeto, podemos representar a operação em paralelo de 15 caixas de supermercado, alimentados por uma fila única, utilizando um *Activity* com *Replicate* = 15. O próximo capítulo discutirá detalhes sobre o uso da propriedade *Replicate*.

A utilidade das demais opções não será comentada neste livro.

1.7.4. Consulta de resultados: resumo de resultados

O SIMUL8 permite a criação de um resumo de resultados personalizado, que pode ser consultado após o final da simulação por meio do botão *Results Manager* (Figura 1.29) da aba principal do programa (aba *Home*). Para configurar os resultados que serão exibidos no resumo é preciso clicar com o botão direito do *mouse* sobre os valores desejados nas janelas de resultados correspondentes (o cursor mudará de cor e de formato ao ser posicionado sobre algum resultado numérico — vide Figura 1.28). Para remover algum resultado do resumo, o procedimento é o mesmo.

Para o exemplo do posto bancário, faça as seguintes seleções:

- **Fila de espera:** tempo médio de espera e número médio de clientes em fila;
- **Atendimento:** porcentagem de tempo em atividade;
- **Saída:** tempo médio no sistema.

Execute novamente a simulação do posto, clique no botão *Results Manager* e observe os resultados, que deverão ser semelhantes aos da Figura 1.29⁷.

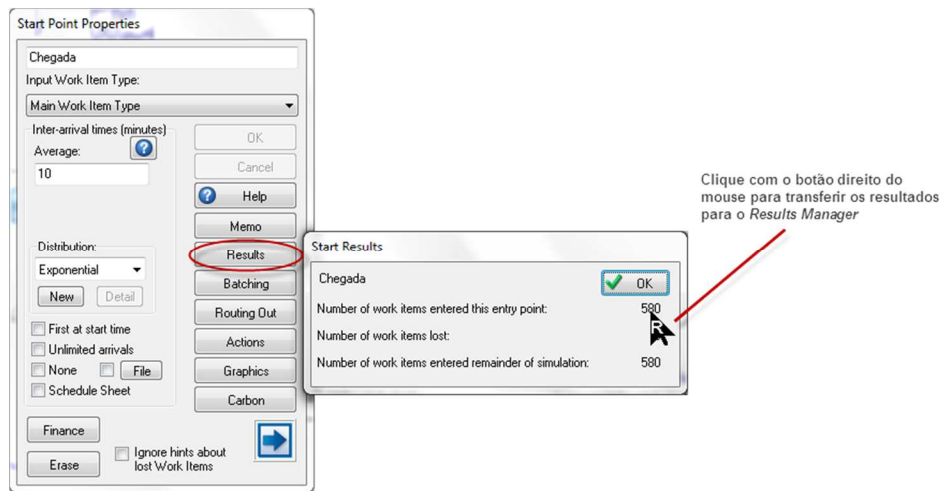


Figura 1.28 - Inclusão de resultados do número de entidades que entraram no sistema no relatório de resumo (*Results Manager*).

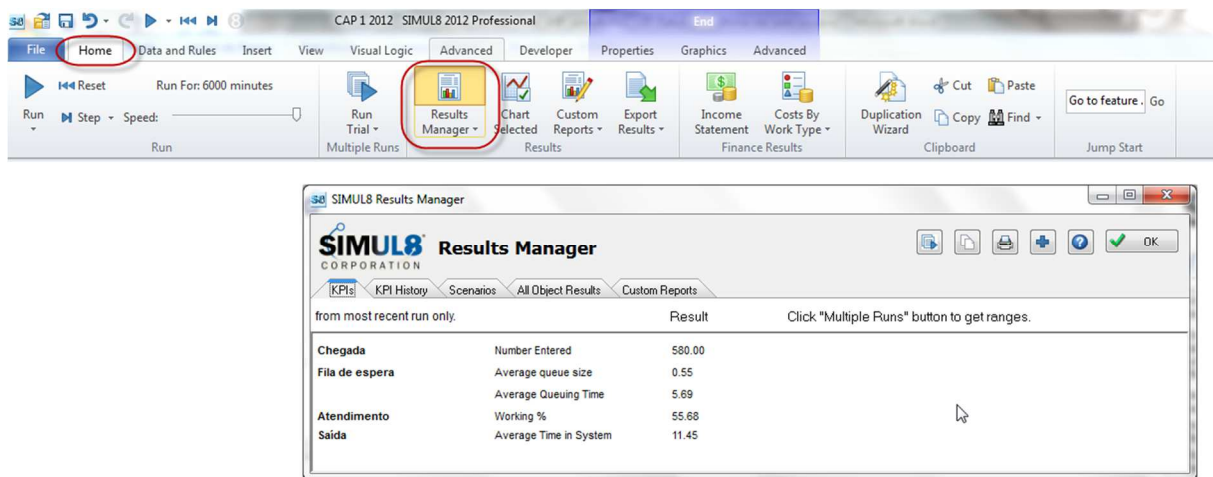


Figura 1.29 - Exemplo de resumo dos resultados de uma simulação.

1.8. Executando uma rodada de simulação (várias replicações)

Nosso modelo de simulação utiliza dados de entrada aleatórios (intervalo entre chegadas e tempo de atendimento). Portanto, cada execução da simulação, que chamaremos de *replicação do modelo*, produzirá resultados de saída também aleatórios. Assim, não podemos concluir muita coisa a partir de uma única replicação. Por exemplo, o tempo de espera em fila pode ser muito alto porque o SIMUL8 usou números aleatórios “ruins” na simulação. Quando tomamos os resultados de várias replicações, cada uma executada com diferentes valores aleatórios de entrada, estaremos executando uma *rodada de simulação*. Neste caso, os resultados possuem características importantes, como um valor médio e uma *previsão*. Para exemplificar esta ideia, vamos simular 30 dias de operação do nosso posto bancário. Inicialmente, altere o período de coleta de resultados (*Data and Rules | Properties | Results Collection Period*) para 360

⁷ O resumo de resultados do SIMUL8 Professional (*Results Manager*), mostrado na Figura 1.29, contém mais opções do que o correspondente da versão Standard (*Results Summary*).

minutos (= 6 horas). Em seguida, clique na seta abaixo do botão *Run Trial* (na aba *Home*), selecione a opção *Trial Setup* e ajuste o número de replicações da rodada para 30 (Figura 1.30).

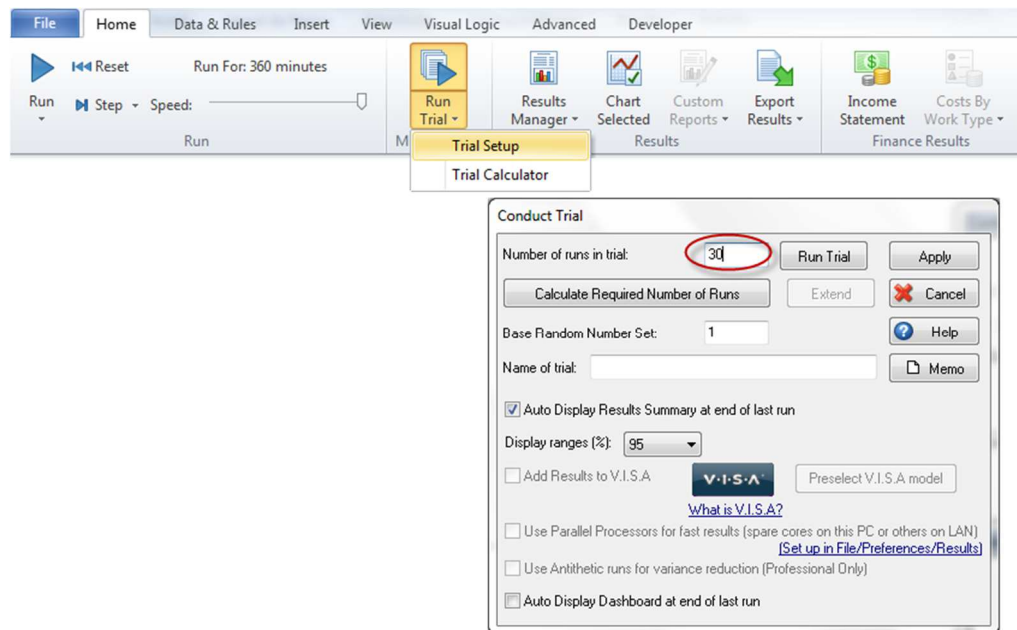


Figura 1.30 - Janela de ajuste da rodada de simulação.

Por fim, para executar as 30 replicações do modelo, clique no botão *Run Trial* como indicado na Figura 1.31 (caso você clique no botão *Run*, o SIMUL8 executará apenas 1 replicação). Ao final da rodada, o SIMUL8 exibirá automaticamente o resumo de resultados (Figura 1.32).



Figura 1.31 - Diferença entre as opções *Run* e *Run Trial*.

SIMUL8 Results Manager				
SIMUL8 CORPORATION Results Manager				
KPIs				
KPI History				
Scenarios				
All Object Results				
Custom Reports				
		Low 95% Range	Average Result	High 95% Range
Chegada	Number Lost	0.00	0.00	0.00
	Net Number Entered	34.28	36.67	39.05
Fila de espera	Number Entered	34.28	36.67	39.05
	Average queue size	0.70	1.05	1.39
Atendimento	Average Queuing Time	6.40	8.99	11.59
	Working %	55.68	60.96	66.24
Saída	Average Time in System	12.29	15.15	18.01

Figura 1.32 - Resumo de resultados de uma rodada de simulação.

Os resultados de uma rodada de simulação sempre são mostrados na forma de intervalos de confiança⁸. Por exemplo, conforme a Figura 1.32, o tempo médio de espera em fila (*Average Queueing Time*) no posto bancário é de 8,99 minutos (*Average Result*). Além disso, com 95% de confiança estatística, pode-se afirmar que o tempo médio de espera está entre 6,40 e 11,59 minutos. Quanto mais replicações forem feitas em uma rodada, mais precisos serão os resultados obtidos, ou seja, menor será o tamanho do intervalo de confiança. Experimente executar 60 replicações do modelo para comprovar isso.

1.9. Dimensionando o número de replicações

Uma forma prática de se dimensionar o número adequado de replicações de um modelo é pelo uso da ferramenta *Trial Calculator*. Por exemplo, vamos determinar quantas replicações devem ser feitas para garantir que os tempos de espera em fila e de permanência no sistema variem, no máximo, 5% em relação aos respectivos valores médios. Para isso, clique em *Home | Run Trial | Trial Calculator* e selecione as grandezas de interesse (*Key Performance Indicator — KPI*), entre as que já constam no resumo de resultados, adicionando-as à lista *KPI's to meet required precision* (Figura 1.33). Ajuste a precisão percentual (*Required precision*) — 5% para o nosso exemplo — e clique em OK.

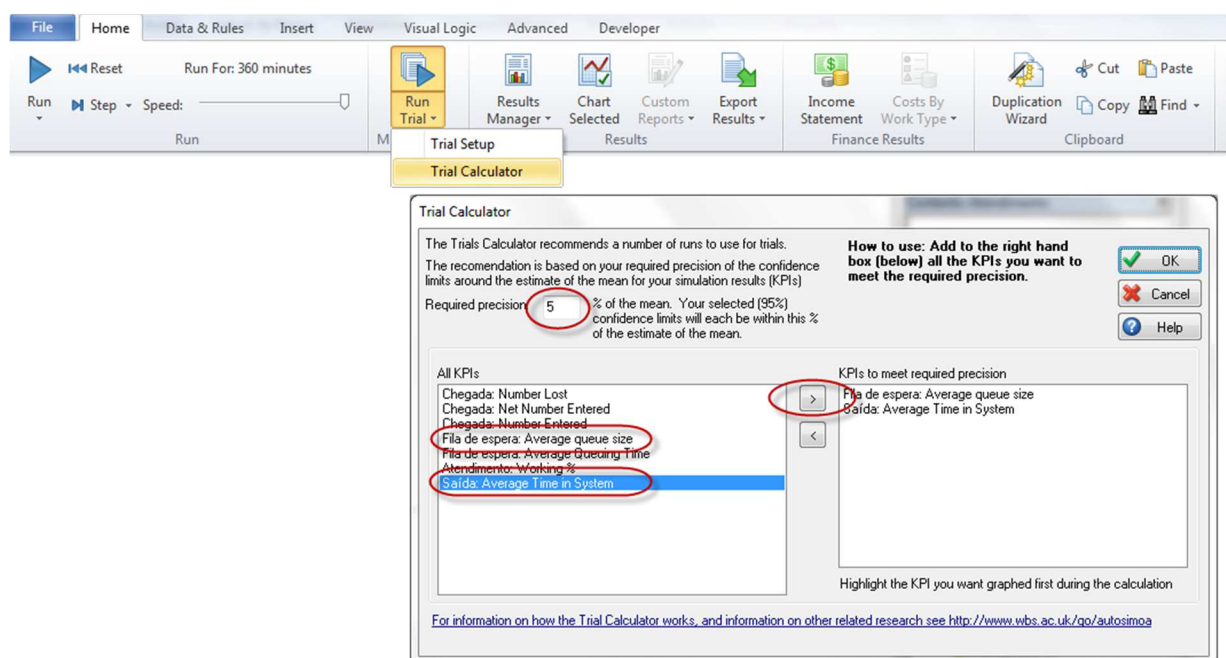


Figura 1.33 - Configuração da ferramenta *Trial Calculator*.

Durante a simulação, o SIMUL8 mostrará um gráfico com a evolução do intervalo de confiança, conforme as replicações são executadas (Figura 1.34).

⁸ Para entender o conceito de intervalos de confiança, consulte algum material de estatística ou o Capítulo 6 do livro “Modelagem e Simulação de Eventos Discretos” de Leonardo Chwif e Afonso C. Medina.

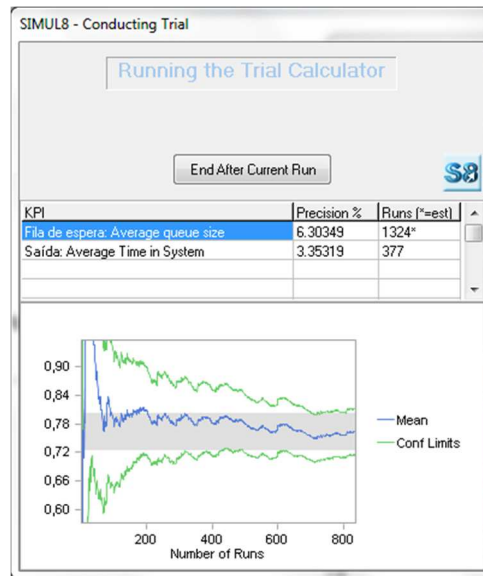


Figura 1.34 - Resultados intermediários do cálculo do número de replicações.

Ao final do processo, será indicado o número recomendado de replicações (Figura 1.35) e o resumo de resultados correspondente (Figura 1.36). Como cada grandeza selecionada impõe um limite diferente, o SIMUL8 irá adotar como número de replicações o **maior** valor recomendado (=1.294 no nosso exemplo).

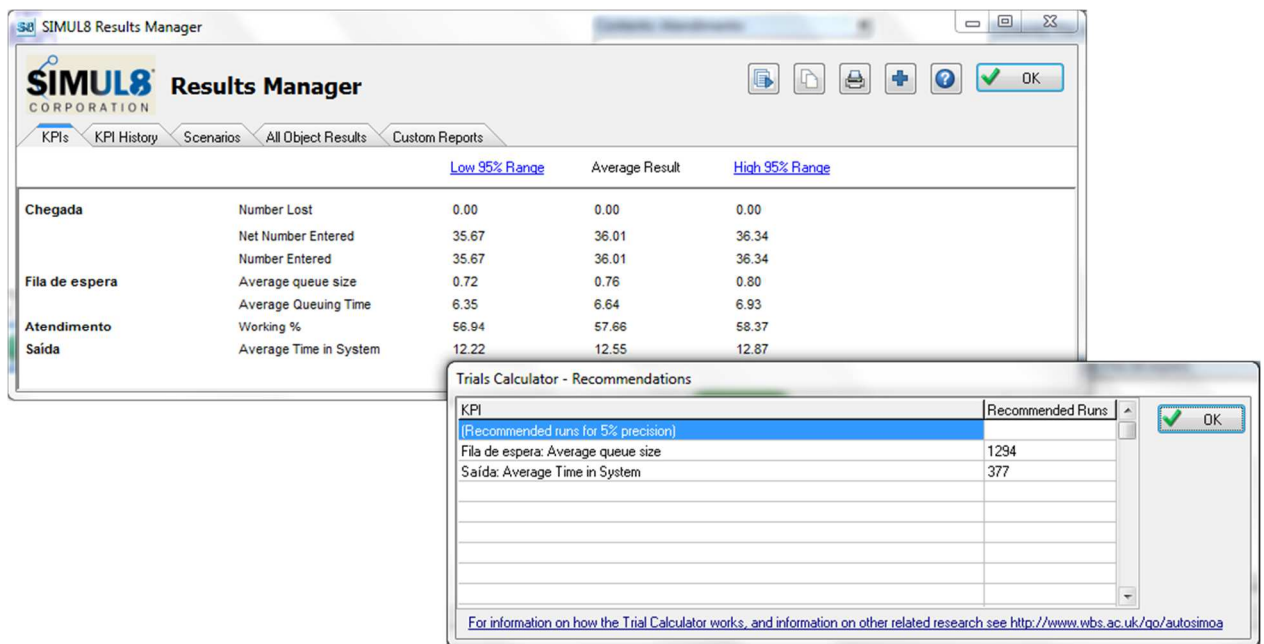


Figura 1.35 - Resultados finais do cálculo do número de replicações.

		Low 95% Range	Average Result	High 95% Range
Chegada	Number Lost	0.00	0.00	0.00
	Net Number Entered	35.67	36.01	36.34
Fila de espera	Number Entered	35.67	36.01	36.34
	Average queue size	0.72	0.76	0.80
Atendimento	Average Queuing Time	6.35	6.64	6.93
	Working %	56.94	57.66	58.37
Saída	Average Time in System	12.22	12.55	12.87

Figura 1.36 - Resultados de saída após 1.294 replicações.

Capítulo 2 ALOCAÇÃO, CONTROLE DE RECURSOS E LABELS

2.1. Introdução

Os *recursos* são entidades associadas aos objetos *Activity*, que controlam ou restringem a execução de uma atividade. Isto acontece porque, se um recurso não estiver disponível para um *Activity* que o utiliza, este *Activity* não poderá realizar sua tarefa. Na prática, isto serve para representar a disponibilidade limitada de recursos, como máquinas, pessoas, ferramentas etc.

Outro conceito importante em simulação é o de *atributo de entidade*, chamado no SIMUL8 de *label*. Neste capítulo faremos uma introdução ao uso de *labels* para explicar a sua funcionalidade na alocação de recursos —*Resource by Label*— assunto que será retomado em outros capítulos deste livro, dado seu uso recorrente nos modelos de simulação.

2.2. Exemplo inicial

Exemplo 2.1. Considere um centro de distribuição de produtos que possui duas baias para carga de caminhões e uma baia para descarga de treminhões (Figura 2.1). Ambas as operações são feitas por uma de duas empilhadeiras disponíveis. Os treminhões chegam em intervalos fixos de 8 horas, enquanto os caminhões chegam em intervalos sucessivos segundo uma distribuição exponencial, com taxa média de 3 caminhões por hora. O carregamento de um caminhão demora, em média, 30 minutos, segundo uma distribuição exponencial. A descarga de um treminhão leva, em média, 3 horas, também segundo uma distribuição exponencial. Pretende-se simular o sistema por 1200 horas para avaliar o comportamento dos tempos de espera dos caminhões e dos treminhões, além da utilização média das empilhadeiras⁹.

A seqüência do processo para os caminhões é: chega, aguarda em fila se necessário, recebe a carga e sai. Para os treminhões, o processo é: chega, aguarda em fila se necessário, descarrega e sai. Como visto no Capítulo 1, os processos de chegadas serão modelados por *Start Points*, as esperas (filas) por *Queues*, os processos de carga e descarga por *Activities* e as saídas por *Ends*. As empilhadeiras, como veremos adiante, serão modeladas por recursos (*Resources*).

Como existem duas empilhadeiras disponíveis, duas operações de carga poderão ocorrer simultaneamente. Por motivos didáticos, vamos representar a operação de carga por meio de dois *Activities* em paralelo, como ilustrado na Figura 2.2, ainda sem a representação das empilhadeiras.

⁹ Exemplo retirado de “Introdução a Simulação com o SIMUL8” de Moreira, Gomes e Silva, Secção de Urbanismo, Transportes, Vias e Sistemas Departamento de Engenharia Civil e Arquitectura, Instituto Superior Técnico, 1993.

Seria mais simples (e eficiente, do ponto de vista computacional) usar apenas um *Activity* com *Replicate* = 2 do que usar dois objetos em paralelo. Repetindo, tomamos esta decisão apenas por razões didáticas.

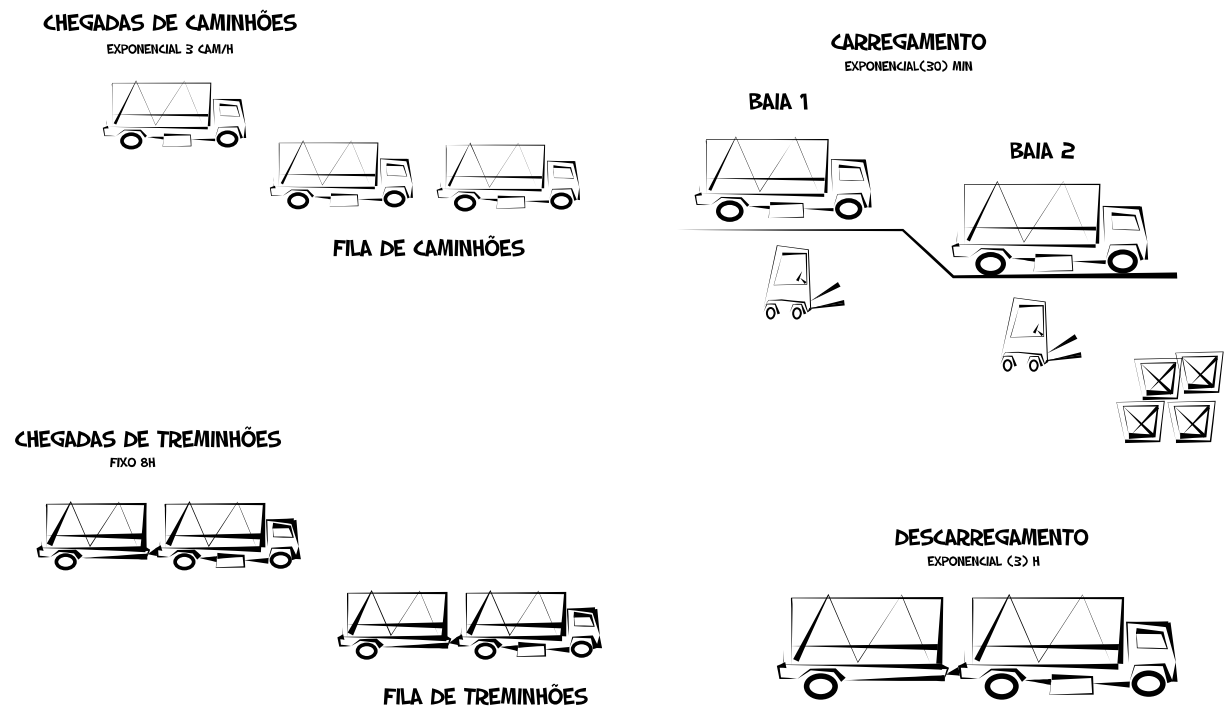


Figura 2.1 - Exemplo inicial: um centro de distribuição que recebe caminhões para serem carregados e treminhões para descarregamento.

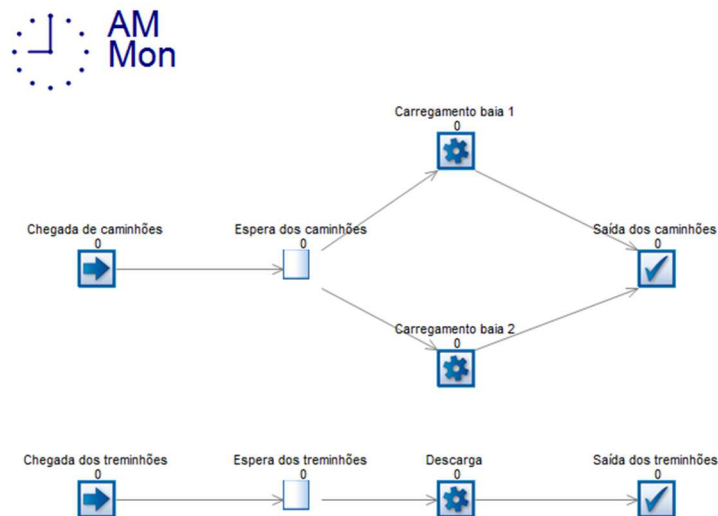


Figura 2.2 – Representação parcial do modelo do centro de distribuição.

Segundo as informações da descrição do exemplo, a duração das atividades deve ser configurada no SIMUL8 da forma indicada na Tabela 2.1 – valores em minutos.

Tabela 2.1 - Tempos entre chegadas sucessivas e de duração dos atendimentos (em minutos).

Variável	Treminhões	Caminhões
Tempo entre chegadas	Constante = 480	Exponencial, média = 20
Duração da atividade (atendimento)	Exponencial, média = 180	Exponencial, média = 30

As duas empilhadeiras são utilizadas em três atividades do processo, de modo que elas podem não estar sempre disponíveis para uso. Este fato as qualifica como *recursos*, como comentamos na introdução deste capítulo. Para adicionar o recurso “Empilhadeiras” ao modelo, clique no objeto *Resource* no painel de blocos construtores e, em seguida, em algum ponto da janela de simulação (Figura 2.3) – opcionalmente, arraste o objeto para a janela de simulação. O título do recurso, “Empilhadeiras”, pode ser digitado no canto superior da tela, no campo *Name*: do menu *Properties*, que surge enquanto o recurso estiver selecionado. Pelo mesmo menu, na opção *Number available*, você pode alterar o número de empilhadeiras disponíveis: no caso do exemplo, o valor é igual a 2.

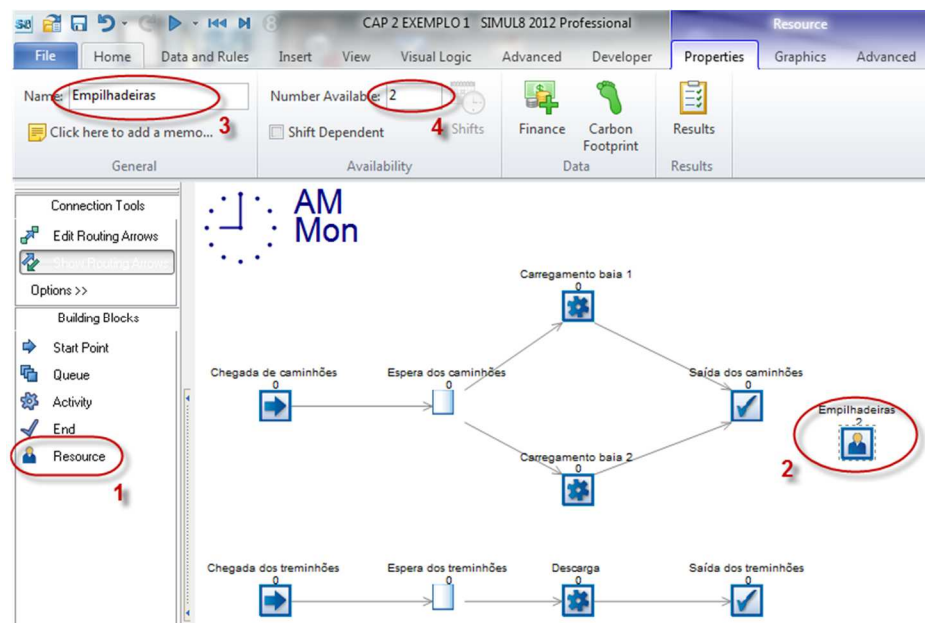


Figura 2.3 – Modelo do problema do armazém, com a inclusão de recursos.

A janela de propriedades do recurso, aberta clicando-se duas vezes no ícone do recurso, também permite trocar o nome e o número dos recursos disponíveis, como mostrado na Figura 2.4.

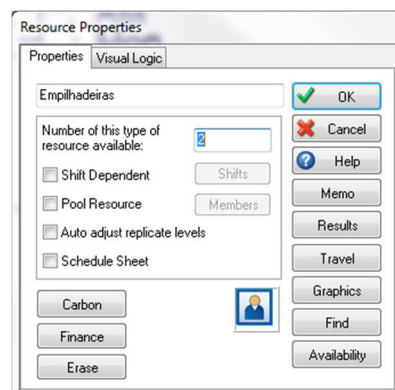


Figura 2.4 – Janela de propriedades do recurso.

Por fim, é preciso *associar* os recursos às atividades, indicando para o SIMUL8 que a atividade só pode acontecer se o recurso estiver disponível¹⁰. Se isto não for feito, o modelo de simulação vai representar uma situação absurda, como a de uma empresa que contrata um operador de máquina e compra uma máquina automática que funciona sem a presença do operador! Para fazer esta associação entre atividade e recurso, abra a janela de propriedades da atividade “Carregamento baía 1” e clique no botão *Resources*. Associe o recurso à atividade clicando no botão *Add* e selecionando o recurso desejado (Empilhadeiras), como apresentado na Figura 2.5.

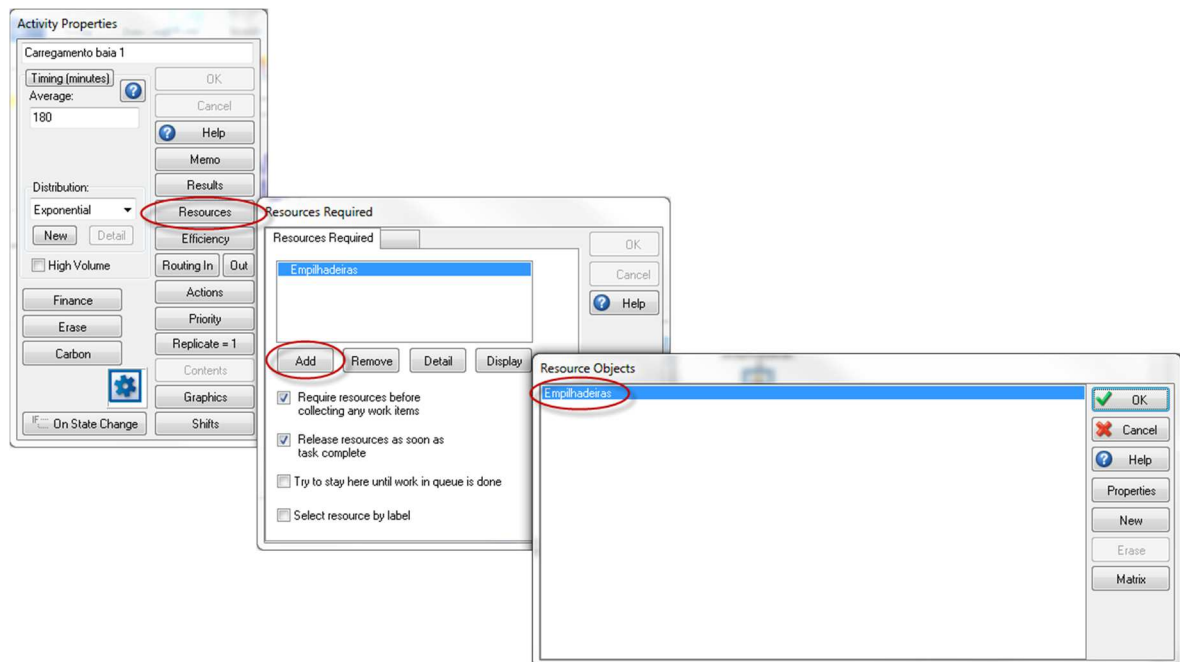


Figura 2.5 – Associação de recurso a uma atividade.

Ainda na janela *Resources Required*, clique no botão *Detail* para definir o número de recursos necessários para a execução da atividade (*Number Required: Min e Max*). Para este exemplo, não é preciso alterar os valores padrão (ambos iguais a 1), mas pedimos que você abra esta janela para discutirmos o seguinte:

¹⁰ Em nossa experiência, percebemos que muitos erros em modelos de simulação são devidos a problemas com a alocação de recursos.

Se forem definidos valores diferentes para o mínimo e o máximo de recursos exigidos, o tempo de execução da atividade será inversamente proporcional ao número de recursos que puderem ser alocados para a atividade (isto será detalhado mais adiante neste capítulo).

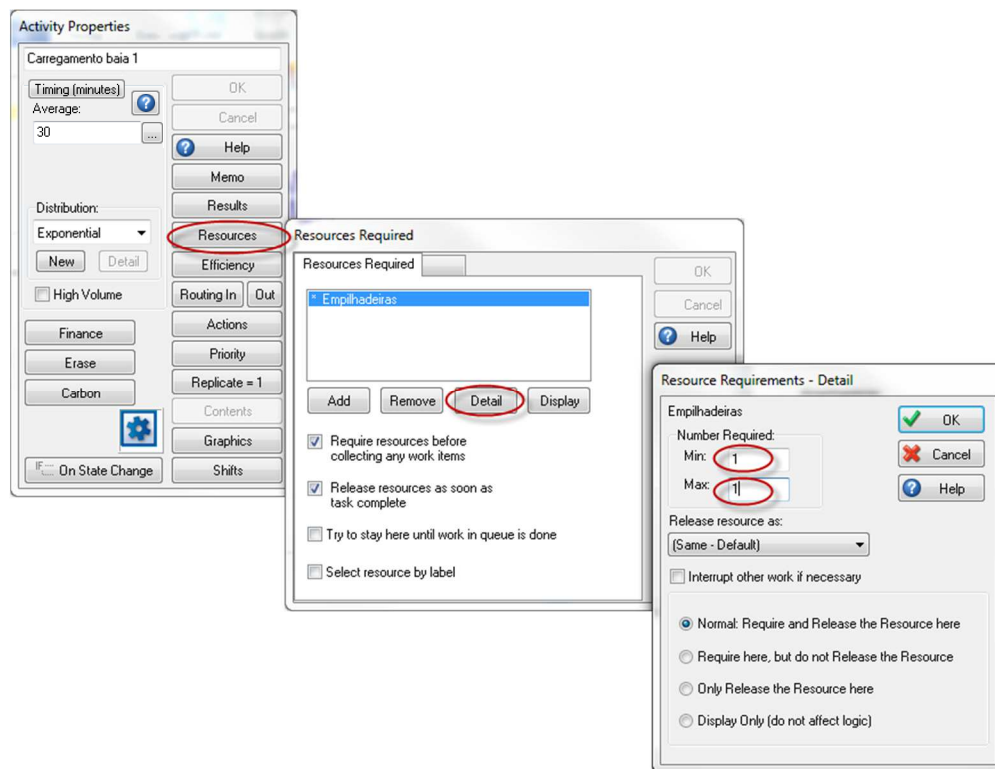


Figura 2.6 – Definição do número de recursos necessários para execução da tarefa no Activity.

Repita a associação do recurso “Empilhadeiras” com a atividade de carga na baia 2 e com a atividade de descarga. O SIMUL8 indicará visualmente a associação entre recursos e atividades se você clicar sobre um recurso e mantiver o botão do *mouse* pressionado (Figura 2.7). **obs.** A associação recurso - atividade pode ser também ser realizada de forma gráfica da mesma maneira que se liga dois objetos através de um link. Neste caso o link não será visualizado mas aparecerá a mensagem “Recurso xxx adicionado a atividade yyy”

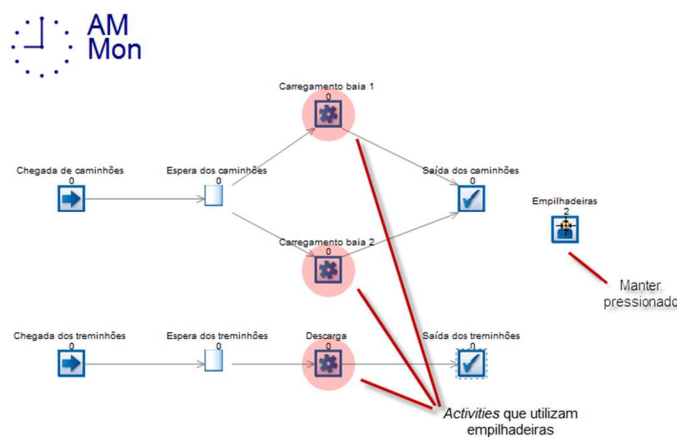
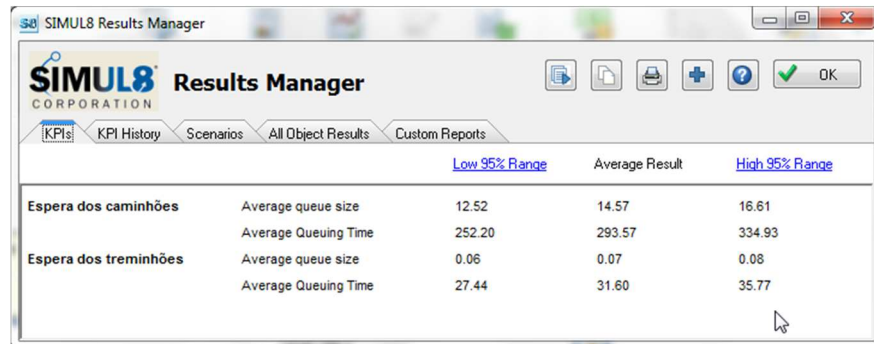


Figura 2.7 - Mantendo o mouse pressionado sobre objeto Resources o SIMUL8 realça os Activities que utilizam o recurso com bolas vermelhas.

Simule o sistema por 72.000 minutos (*Warm-up* = 10.000 minutos), fazendo uma rodada com 30 replicações (*Trials*) e analise os principais indicadores de desempenho do modelo: médias dos tamanhos e dos tempos de espera nas filas e

utilização percentual dos recursos. Para acessar os resultados de saída de um recurso ou configurar o relatório de resultados basta proceder como discutido no Capítulo 1. Lembre-se, é claro, de zerar os tempos de deslocamento do modelo (*Travel Times*) antes de executar a simulação. Compare seus resultados com os nossos, mostrados na Figura 2.8. Que conclusões podem ser obtidas sobre o funcionamento do sistema?



		Low 95% Range	Average Result	High 95% Range
Espera dos caminhões	Average queue size	12.52	14.57	16.61
	Average Queuing Time	252.20	293.57	334.93
Espera dos treminhões	Average queue size	0.06	0.07	0.08
	Average Queuing Time	27.44	31.60	35.77

Figura 2.8 – Alguns resultados do modelo.

Considere, agora, que uma das empilhadeiras pare de funcionar, de modo que apenas uma operação de carga ou descarga possa ser feita a cada instante. Para isso, no modelo, altere a quantidade empilhadeiras disponíveis para 1 (*Number of this type of resource available*). Simule novamente o modelo e veja o tamanho do “estrago” causado.

2.3. Prioridade de recursos (*Resources Priority*)

Como vimos no exemplo anterior, um recurso pode ser associado a atividades diferentes. No entanto, quando as duas atividades precisam ser executadas (porque existem entidades em suas filas), qual delas terá prioridade na alocação do recurso?

Para responder a essa pergunta, vamos retornar ao exemplo da seção anterior, assumindo que exista apenas uma empilhadeira disponível e que a prioridade de alocação do recurso tenha que ser para a operação de descarga dos treminhões. Todo *Activity* possui como valor padrão de prioridade de alocação de recursos igual a 50 (confira, clicando no botão *Priority* — Figura 2.9). Qualquer valor acima deste faz com que um *Activity* tenha maior prioridade na alocação de um recurso do que os demais *Activities* que utilizem o mesmo recurso. Assim, para tornar a descarga dos treminhões prioritária, basta alterar a prioridade do *Activity* “Descarga” para qualquer valor acima de 50 (por exemplo, 51). Com isso, se houver um caminhão e um treminhão em fila, a empilhadeira será alocada para a descarga do treminhão, devido à prioridade. Mais adiante, discutiremos como alterar dinamicamente (ou seja, durante a execução da simulação) a prioridade de alocação de recursos.

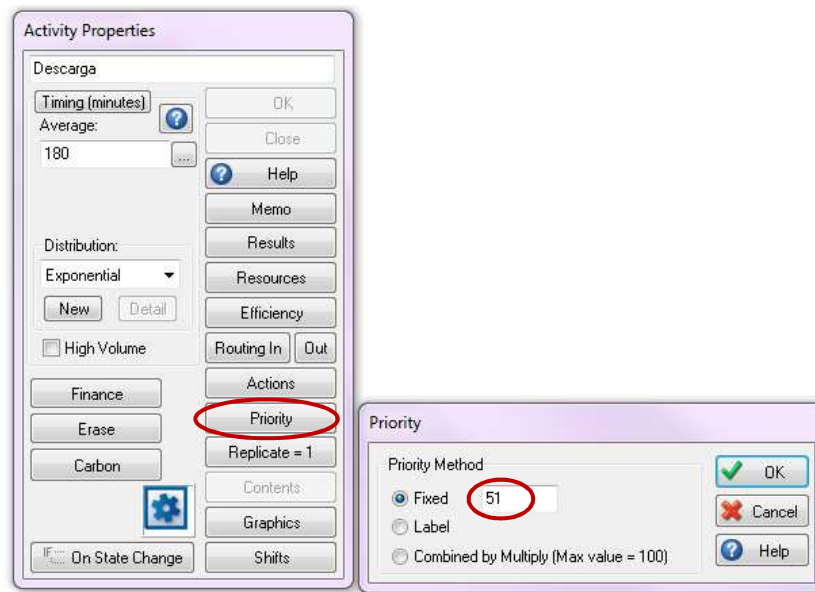


Figura 2.9 – Alteração da prioridade de quem utiliza o recurso primeiro (*Activity Priority*).

2.4. Turnos de trabalho (*Shift Work Pattern/Availability*)

Agora veremos como os recursos podem ser associados a *turnos de trabalho* e como definir em quais momentos que eles estão disponíveis, ou não. Para isso, construa um modelo como o que é representado na Figura 2.10.

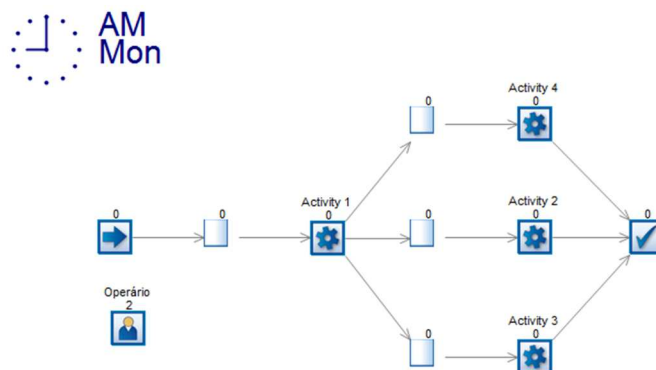


Figura 2.10 - Modelo para recursos.

Os parâmetros deste modelo estão dispostos na Tabela 2.2.

Tabela 2.2 – Dados de entrada para o modelo (valores em minutos).

Objeto	Propriedade	Valor
Chegada	Inter Arrival Times	Exponencial, média = 3 min
Activity 1	Timing	Normal(3; 2,5)
Activity 2	Timing	Normal(10; 2,5)
	Resources	Operário
Activity 3	Timing	Normal(10; 2,5)
	Resources	Operário
Activity 4	Timing	Normal(10; 2,5)
	Resources	Operário
Operário	Number of this resource	2

Note que somente os *Activities* 2, 3 e 4 possuem um recurso associado (“Operário”), cuja disponibilidade é de 2 unidades. Após criar o modelo, ajuste as configurações do tempo de simulação (*Results Collection Period*) para 2.400 minutos e considere tempo de *warm-up* igual a zero.

Como principal medida de desempenho do resumo de resultados, considere o total de peças produzidas no período total de simulação (Saída - *Work Complete*). Em nossos testes, realizando uma rodada com 10 replicações, obtivemos um valor médio de 496,65 peças¹¹.

Considere, agora, que há dois turnos de trabalho, um das 9:00 às 12:00 (turno da manhã) e outro das 12:00 às 17:00 (turno da tarde), sendo que 3 operadores estão disponíveis no primeiro turno e apenas 2 operadores estão disponíveis no segundo turno. Para representar esta situação, acesse o menu *Data and Rules* e, em seguida, *Shifts*, para definir os 2 turnos de trabalho, como ilustrado na Figura 2.11 para o caso do turno da manhã (clique no botão *New* para inserir os dados de cada turno).

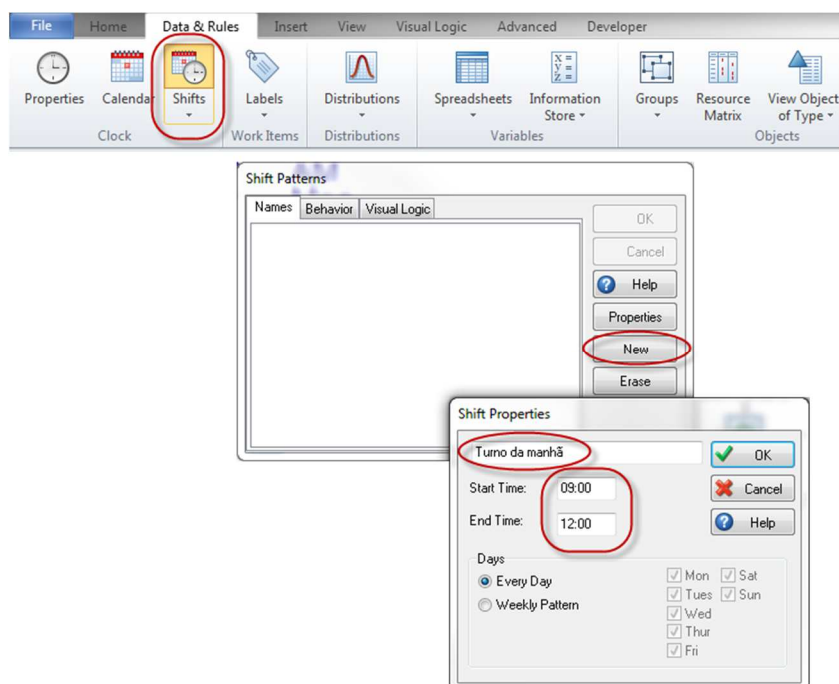


Figura 2.11 - Definições do turno da manhã de trabalho para o modelo.

¹¹ O resultado da sua simulação não precisa ser igual a este, mas não deve ser muito diferente. Se houver uma diferença significativa, confira as configurações dos objetos e da simulação. Confira, ainda, se os tempos de deslocamento do modelo foram zerados.

Por si só, os turnos nada mais são do que definições de janelas de tempo para o SIMUL8. Eles só terão alguma utilidade quando nós definirmos que existem 3 operários disponíveis no primeiro turno e 2 no segundo. Para isso, selecione as propriedades do recurso “Operário” e escolha a opção *Shift Dependent* para definir sua disponibilidade total em cada turno. A Figura 2.12 ilustra como deve ser feita esta configuração.

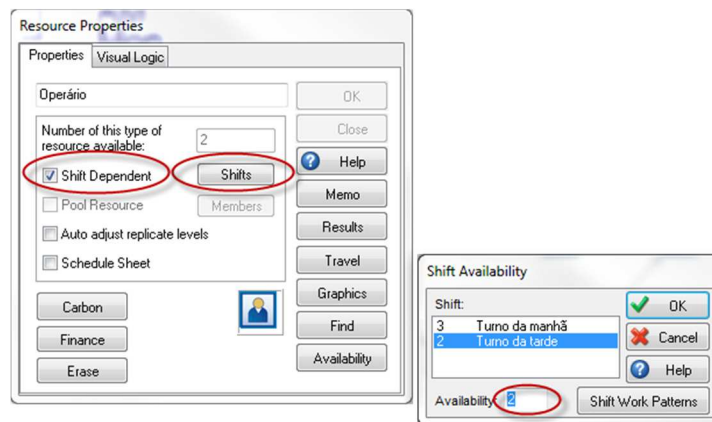


Figura 2.12 - Configuração do recurso.

Após essas modificações, obtivemos uma produção média de cerca de 557 unidades após 10 replicações do modelo, ou seja, um aumento de quase 18% na produtividade original, com a adição de um recurso no primeiro turno.

Agora, suponha que a cada turno os operadores usam 2% do seu tempo produtivo para fazer ginástica laboral, para minimizar lesões por esforço repetitivo. Para verificar o impacto desta situação na produtividade, abra a janela de propriedades do recurso “Operário”, clique na propriedade *Availability* e ajuste o valor da disponibilidade para 98%. Simulando esta nova condição, obteremos uma produção média de cerca de 552 peças, o que representa uma perda de produtividade de cerca de 0,9% em relação à situação anterior – a perda não é próxima de 2% porque os recursos não param ao mesmo tempo. No Capítulo 7 detalharemos melhor a questão de disponibilidade de recursos e de *Activities*.

2.5. Opções detalhadas de recursos

A alocação de recursos pode ser feita de formas mais complexas do que as discutidas até o momento. Estas opções podem ser acessadas na janela de configuração de recursos dos *Activities*, clicando-se no botão *Detail* (Figura 2.13).

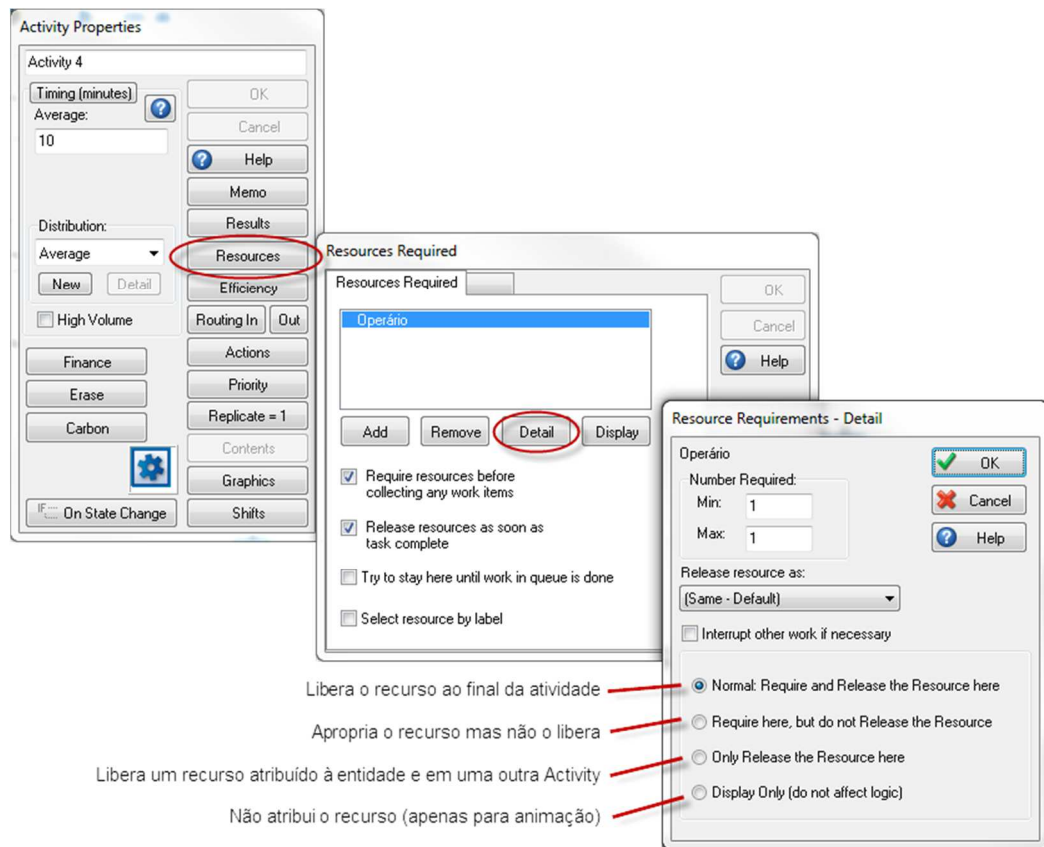


Figura 2.13 - Detalhes de alocação de recurso.

O primeiro campo da janela de detalhamento define as quantidades mínima e máxima de recursos que podem ser alocadas para a atividade. Por exemplo, se pudermos alocar de 1 a 4 recursos, o SIMUL8 considerará que o tempo configurado no *Activity* para a realização da tarefa equivale à situação em que há apenas um recurso trabalhando. Se for possível alocar 2 recursos, o tempo de execução será dividido por 2; se for possível alocar 3 recursos, o tempo de execução será dividido por 3, e assim por diante. Assim, o número mínimo de recursos alocados será a referência para que o SIMUL8 calcule o tempo de execução da atividade. A opção de liberar um recurso como sendo de outro tipo (*Release resource as*) e a opção de interrupção de outro *Activity* não serão abordadas neste livro.

Na parte inferior da caixa de diálogo, vemos a opção padrão de alocação, que é *Normal: Require and Release the Resource Here*. Isto quer dizer que o recurso será liberado pelo *Activity* assim que a atividade terminar. No entanto, em algumas ocasiões, é preciso que o recurso permaneça alocado até ser liberado em outro *Activity*, como se o recurso ficasse “preso” à entidade até poder ser liberado. Vamos ilustrar esta situação no exemplo a seguir.

Exemplo 2.2. A Oplagic é uma transportadora que possui um centro de distribuição (CD), na cidade de Uberlândia-MG. No CD, os intervalos de tempo entre chegadas sucessivas de pedidos de entregas são exponencialmente distribuídos, com média de 4 horas. Os pedidos são despachados em 3 tipos de veículos: vans, caminhões médios e caminhões pesados. A distribuição dos pedidos de carga é de 50% para as vans, 30% para os caminhões médios e 20% para os caminhões pesados. Existem 10 carregadores disponíveis para o carregamento dos veículos. Para carregar uma van são necessários 2 carregadores, enquanto os caminhões médios e pesados exigem 4 carregadores. Os tempos de carregamento, tanto das vans quanto dos caminhões médios, seguem uma distribuição normal com média de 100 minutos e desvio padrão de 30 minutos. Para carregar um caminhão pesado, os carregadores gastam um tempo que segue uma distribuição normal com média de 130 minutos e desvio padrão de 45 minutos. A frota de veículos é composta por 3 vans, 2 caminhões médios e 2 caminhões pesados. Para serem carregados, os veículos devem estar necessariamente posicionados em uma doca. O centro de distribuição possui 5 docas, sendo 2 pequenas (para vans) e 3 grandes (para caminhões médios e pesados). Os tempos de deslocamento entre a distribuidora e os clientes (ida e volta) estão uniformemente distribuídos entre 180 e 300 minutos para as vans e os caminhões médios e entre 240 minutos e 480 minutos para os caminhões pesados. O objetivo do modelo de simulação é verificar se este sistema irá

impor atrasos significativos às entregas dos pedidos. O atraso médio (tempo de permanência no sistema) deverá ser em torno de 6 horas. Considere o tempo de simulação de 1 semana (10.080 minutos). Para efetuar a divisão percentual entre vans, caminhões médios e caminhões pesados, utilize no objeto chegada a opção *Routing out percent* para inserir a porcentagem para cada tipo de veículo.

A estrutura do modelo proposto para representar esta situação é mostrada na Figura 2.14.

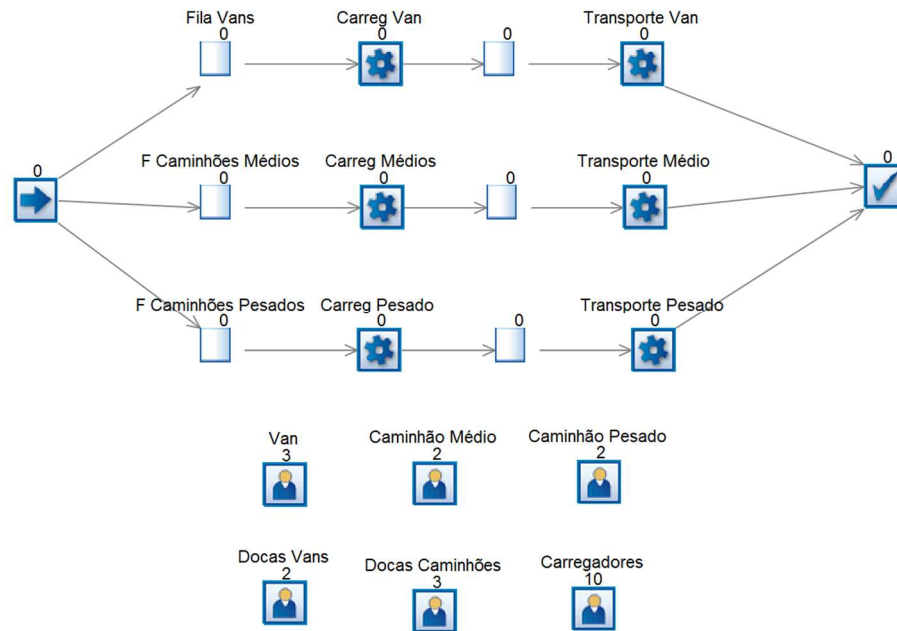


Figura 2.14 - Estrutura do modelo para a descrição fornecida

Como é possível que, em algum instante de tempo, mais de uma operação de carga ou de transporte ocorra em paralelo, é preciso ajustar a opção *Replicate* de cada *Activity*. No entanto, ao invés de fazer este ajuste manualmente, é mais simples e prático selecionar a opção *Auto adjust replicate levels* em todos os recursos, o que garante que a propriedade *Replicate* de cada *Activity* terá valor igual ao total disponível dos recursos associados (esta opção está disponível clicando nas propriedades dos recursos).

Vamos explicar passo a passo como deve ser feita a alocação dos recursos aos *Activities* “Carrega vans” e “Transporte vans” — para os demais, a alocação será feita de forma semelhante.

Para a atividade “Carrega vans” (Figura 2.15), observe que:

- Deve ser alocada uma van e uma doca específica para vans;
- A doca de vans deve possuir alocação padrão (*Normal: Require Here and Release The resource here*);
- A van, entretanto, deve possuir alocação diferente da alocação padrão. Na janela de propriedades do *Activity* “Carrega vans”, pressionando o botão *Resources*, selecionamos a opção *Require here but do not release the resource* (Figura 2.15). Assim, neste ponto do modelo, a van será alocada para realizar o carregamento, mas não será liberada (isto será indicado por um asterisco ao lado do nome do recurso na janela *Resources Required*).

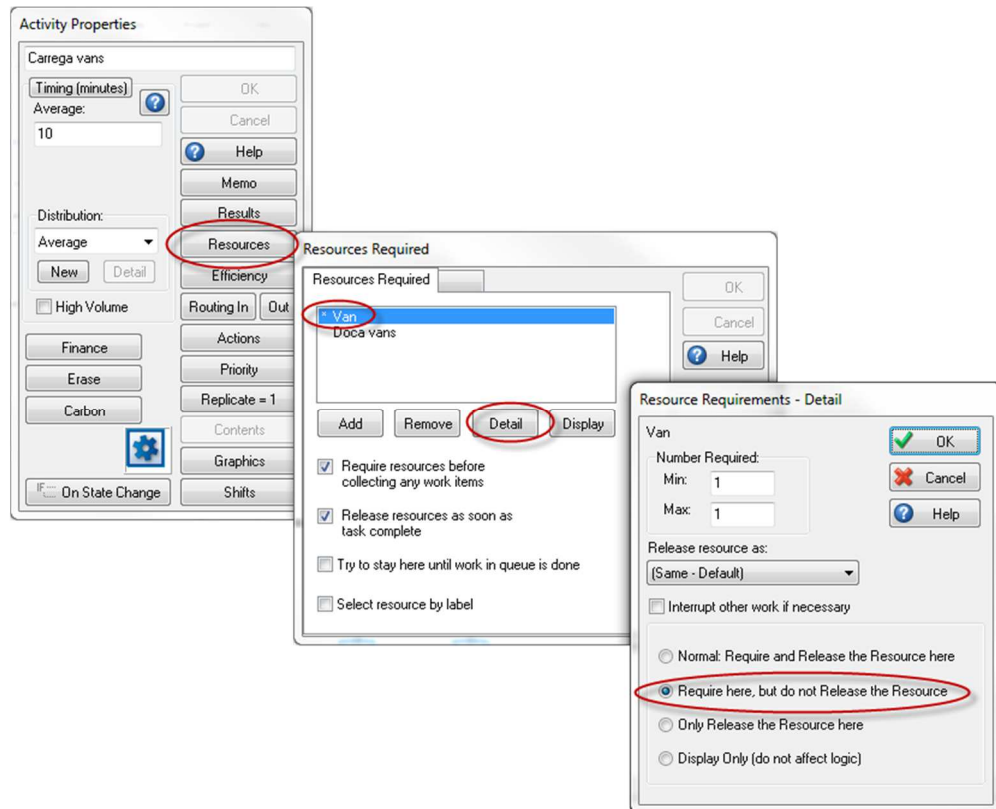


Figura 2.15 - Alocação de Recursos em Carrega Van.

A liberação da van ocorrerá somente no *Activity* “Transporte Van”, que tem as propriedades ilustradas na Figura 2.16 — a janela *Resources Required* foi acessada clicando inicialmente no *Activity* e, em seguida, na opção *Resources* no menu superior. Veja que, neste caso, foi selecionada a opção *Only release the resource here*, indicando que a van só poderá ser liberada para o próximo carregamento quando a operação de transporte acabar.

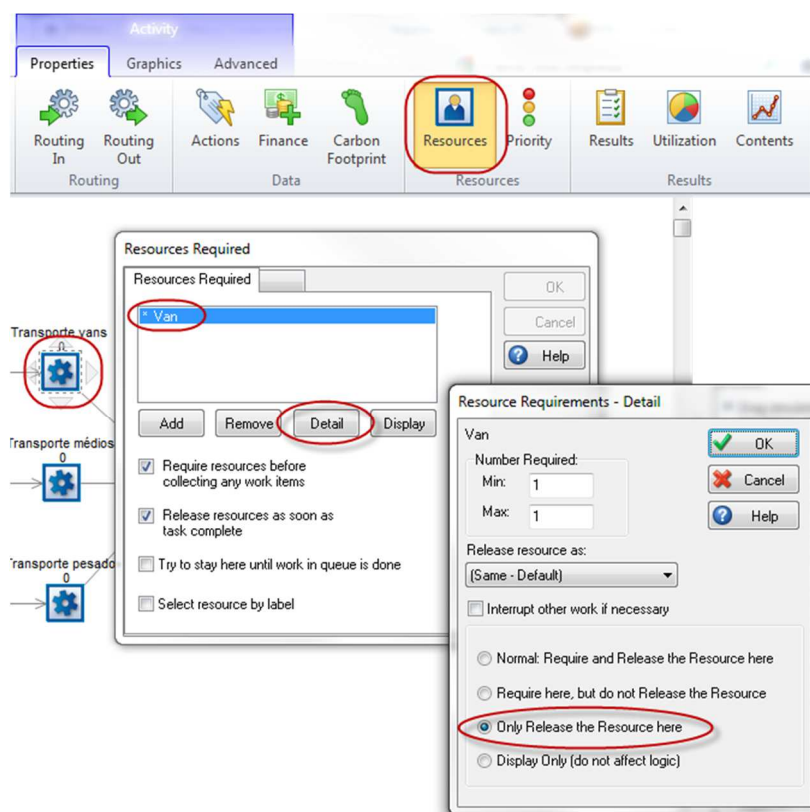


Figura 2.16 - Alocação de recursos em “Transporte Van”

As outras alocações devem ser feitas de forma semelhante. Quando a entidade passar por *Activities* intermediários (algo que *não acontece* neste exemplo), que não alocam e nem desalocam recursos, deve ser utilizada a opção *Display Only*.

2.6. Pool de recursos

Imagine a seguinte situação: um *Activity* pode utilizar um recurso A **ou** utilizar um recurso B, nesta ordem de preferência, mas não ambos. Para facilitar nossa explicação, vamos chamar nossos recursos de Huguinho e Zequinha. Neste caso, os recursos deverão ser agrupados de forma *lógica* em um conjunto denominado *pool* de recursos (*Pool resource*). Para isso, criamos um terceiro recurso, do tipo *Pool Resource*, que representa o agrupamento lógico dos demais. Este recurso é o que será associado ao funcionamento do *Activity*, ao invés dos recursos Huguinho e Zequinha¹² A Figura 2.17 ilustra a sequência de passos para definirmos o *pool* e a Figura 2.18 mostra como atribuir o *pool* criado a um *Activity*.

¹² Lembre-se que, se dois recursos são associados a um *Activity*, este só irá executar sua operação se ambos estiverem disponíveis, algo que *não queremos* que aconteça neste exemplo.

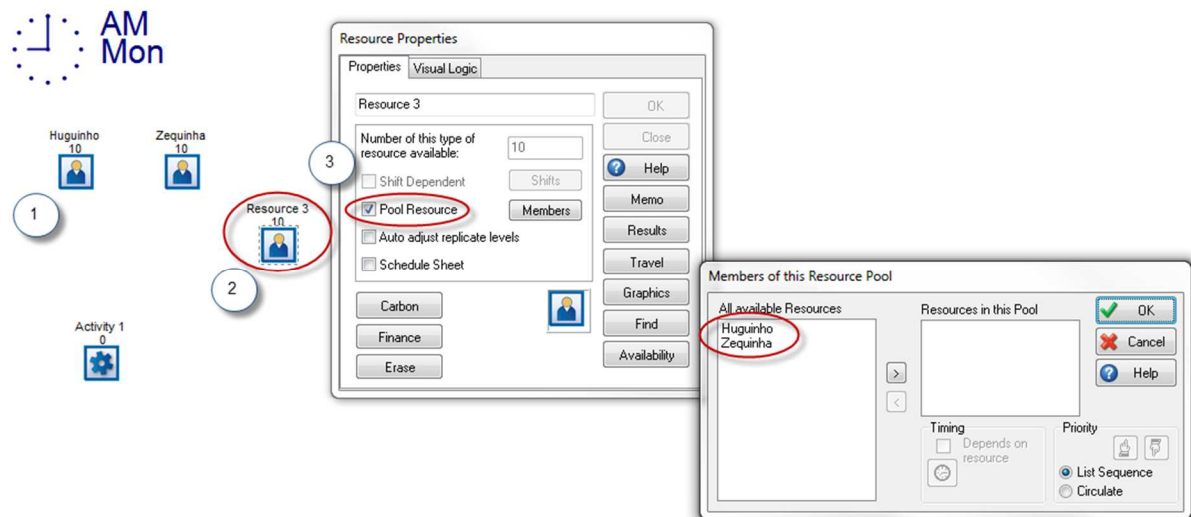


Figura 2.17 - Criação do *Pool* de recursos: 1) criar os recursos que pertencerão ao *Pool*; 2) criar um recurso que será a representação do *Pool* e 3) a partir da opção *Pool Resource*, definir os membros do *Pool*.

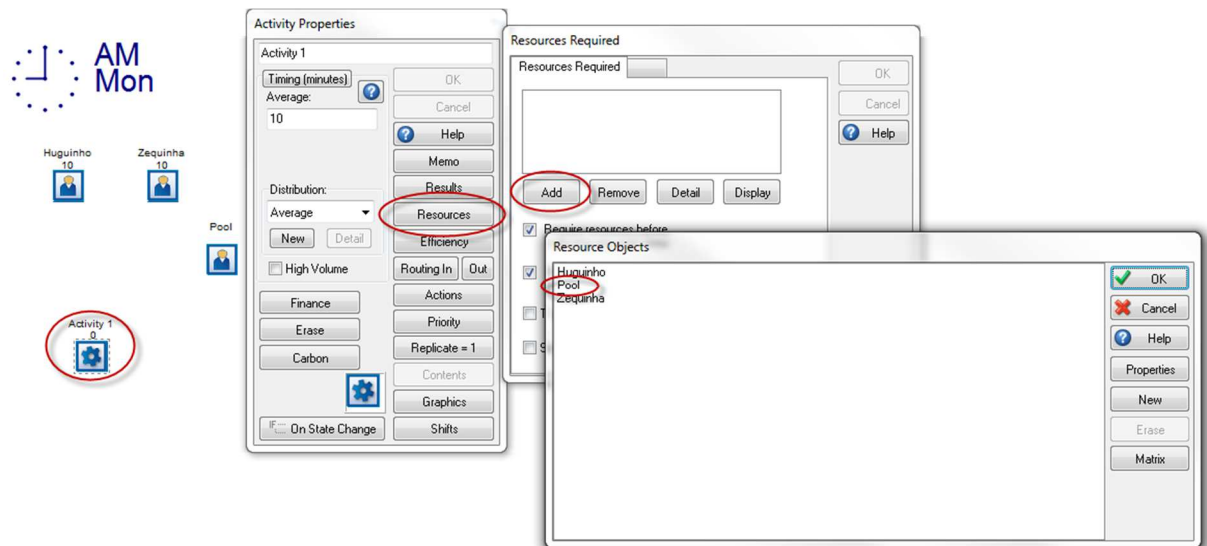


Figura 2.18 - Atribuição do *pool* ao *Activity*.

Note que a opção “*Number of this type of resource available*” do *pool* fica desabilitada e que a opção padrão para utilização dos membros é do tipo *List Sequence*. Isto significa que o recurso preferencial para a alocação do *pool* será o Huguinho (Figura 2.17, passo 3). Caso o recurso Huguinho não esteja disponível, então o *pool* tentará alocar o Zequinha para o *Activity*. Se selecionarmos a opção *Circulate*, o *pool* se encarregará de alocar alternadamente um e outro (“circulando” pela lista de recursos). A partir da versão 2014 do SIMUL8, duas funcionalidades foram adicionadas ao *pool*. A primeira, além das opções *List Sequence* e *Circulate*, foi adicionada ao *pool* a opção *Longest Available*. Neste caso o SIMUL8 irá selecionar preferencialmente o recurso do *pool* que estiver parado pelo maior tempo. Outra mudança é que se pode definir *pools* dentro de *pools*.

Os recursos originais (membros do *pool*) podem estar alocados em qualquer quantidade (por exemplo, poderíamos ter 3 recursos do tipo Huguinho e 2 do tipo Zequinha). Poderíamos, ainda, fazer com que o *Activity* tenha tempos de execução diferentes em função do recurso alocado (por exemplo, se o Huguinho for mais lento que o Zequinha), mas esta é uma funcionalidade avançada, que não será abordada neste livro.

2.7. Deslocamentos de recursos

Voltemos ao exemplo 2.1., do carregamento de caminhões e treminhões. Suponha que a empilhadeira demore 3 minutos para ir de uma doca a outra, antes de poder carregar ou descarregar um veículo. Isto pode ser representado utilizando-se a funcionalidade *Resource Travel Time*. Para acessá-la, clique no botão *Travel* nas propriedades do recurso. Em seguida, selecione os pontos de origem e destino e defina os tempos de deslocamentos associados, como ilustrado na Figura 2.19.

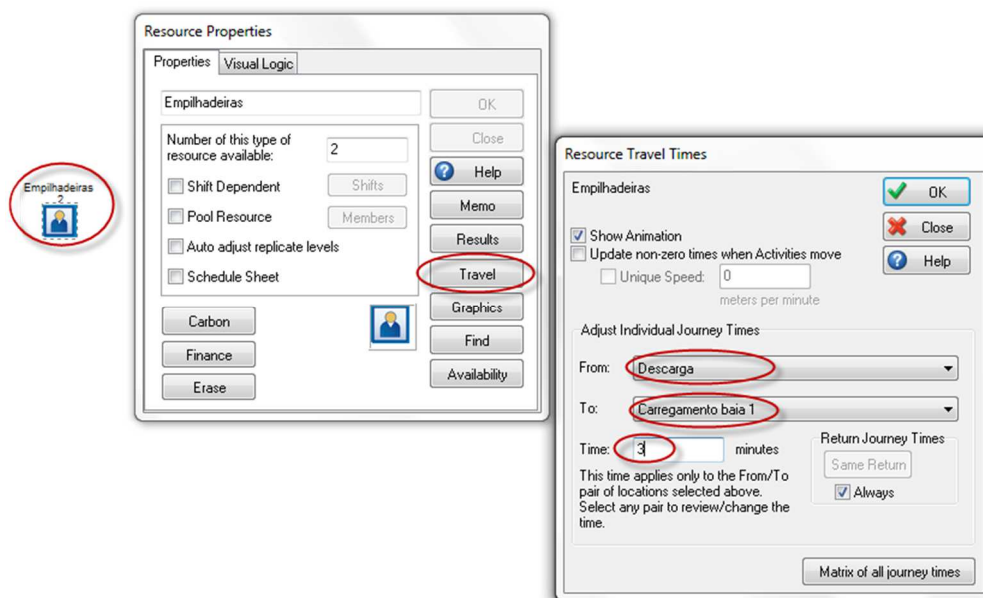


Figura 2.19 - Utilização do *Resource Travel Time*.

Esta definição também pode ser feita na *matriz de distâncias* do recurso (*Matrix of all journey Times*), lembrando que a posição com o nome do recurso se refere à posição inicial do recurso e aquela em que partirá quando estiver “em repouso” (parado). Note, ainda, que todo recurso sempre fica “parado” ao lado do *Activity* que lhe solicitou trabalho após a execução da atividade.

2.8. Labels e Resource by Labels

Vamos iniciar esta seção com uma explanação sobre atributos, denominados no SIMUL8 como *labels*. Os pontos de entrada (*Start Points*) criam entidades (*Work Items*) que se deslocam pelo sistema, sendo processadas e modificadas nos *Activities*, aguardam por processamento em filas (*Queues*), são processadas em atividades (*Activities*), transportadas por esteiras (*Conveyors*) etc. Em algumas situações pode ser necessário que cada uma dessas entidades receba e carregue

alguma *informação individual* durante a simulação. Por exemplo, em uma linha de produção, cada peça pode ser classificada segundo sua cor, tamanho, qualidade etc. Em instituições de saúde, poderíamos classificar os pacientes segundo a gravidade de sua enfermidade. Essas informações são representadas por *labels* no SIMUL8, que podem ser imaginadas como “etiquetas” presas a cada entidade, carregando informações sobre aquela entidade em particular.

2.8.1. Criando *labels*

Para criar um *label* no SIMUL8, clique em *Data and Rules* | *Labels* | *New* e defina o nome e o tipo de *label* desejado (Figura 2.20).

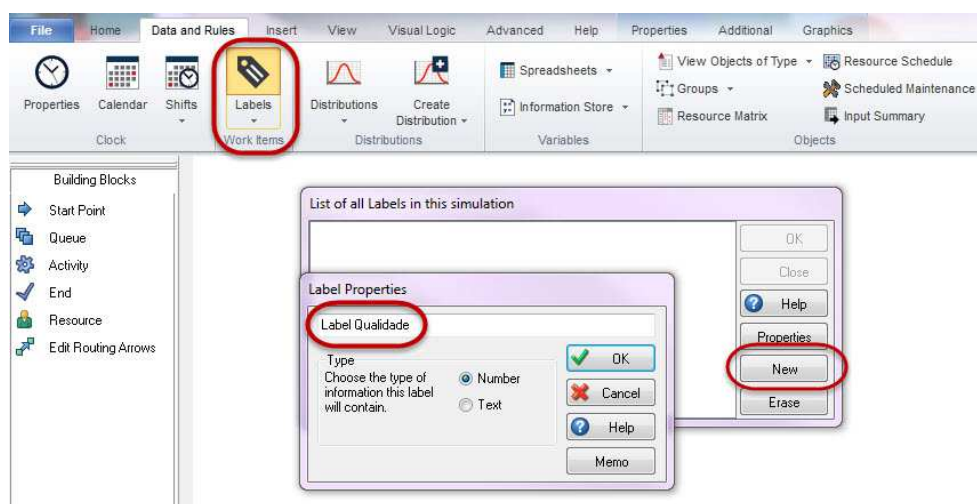


Figura 2.20 - Propriedades do *label*.

É aconselhável (mas não obrigatório) utilizar o prefixo “Label” ou “Lbl” antes do nome do *label* para facilitar a sua identificação, especialmente quando utilizarmos lógica de programação para diferenciarmos os *labels* de outros tipos de informações (Capítulo 5).

Labels podem ser do tipo numérico (*Number*) ou texto (*Text*), sendo que os últimos são menos usados que os primeiros. Na prática, quando precisamos associar informações não numéricas a uma entidade, estabelecemos uma correspondência entre as informações e números que as representem. Por exemplo, suponha que produtos sejam classificados de acordo com sua qualidade, em Qualidade A, Qualidade B e Qualidade C. Neste caso, podemos utilizar *labels* numéricos (LblQualidade), com a correspondência mostrada na Tabela 2.2.

Tabela 2.2 - Exemplo de *labels*.

Qualidade (tipo)	LblQualidade
Qualidade A	1
Qualidade B	2
Qualidade C	3

Um erro muito comum cometido por iniciantes é criar um *label* para cada tipo de qualidade, como LblQualidadeA, LblQualidadeB e LblQualidadeC, o que **não faz** qualquer sentido.

Nas próximas seções deste livro continuaremos a discutir aplicações para os *labels*. Por exemplo, no Capítulo 3 (Entrada de Distribuições Aleatórias) comentaremos como os *labels* podem ser associados a distribuições de probabilidades; no Capítulo 4 (Direcionamento de Entidades), veremos como os *labels* podem ser usados para direcionar entidades em rotas diferentes dentro de um modelo e atuar na configuração de entidades que expiram; no Capítulo 6 (Refinamentos Gráficos e Animação) utilizaremos *labels* para alterar a imagem de uma entidade ao longo da simulação; por fim, no Capítulo 7 (Funcionalidades Específicas), a utilização de *labels* será explorada em um exemplo prático. Esta introdução serviu especificamente para introduzirmos a funcionalidade chamada “*Resource by Labels*”, que será descrita a seguir.

2.8.2. Resource by labels

A funcionalidade *Resource by Labels* foi adicionada em 2008 no SIMUL8 e permite, como o nome indica, que se faça a alocação de recursos por meio de um *label*. Vamos imaginar que temos 3 recursos, Huguinho, Zequinha e Luizinho, sendo que o Huguinho só pode trabalhar em peças do tipo 1, Zequinha só pode trabalhar em peças do tipo 2 e, por fim, Luizinho, em peças do tipo 3, sempre em uma mesma máquina. Para representar esta situação, vamos criar uma estrutura semelhante à da Figura 2.21, deixando os objetos em suas configurações padrão.

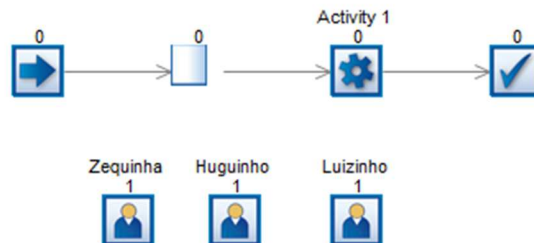


Figura 2.21 - Estrutura do modelo do exemplo.

Agora vamos criar um *label* numérico chamado “Label tipo” (Figura 2.20). Cada entidade que entrar no modelo deverá receber um *label* com o valor 1, 2 ou 3, com igual probabilidade de ocorrência (33,33%). Para isso, logo no ponto de entrada, associamos o *label* criado ao *Start Point*, como ilustrado na Figura 2.22.

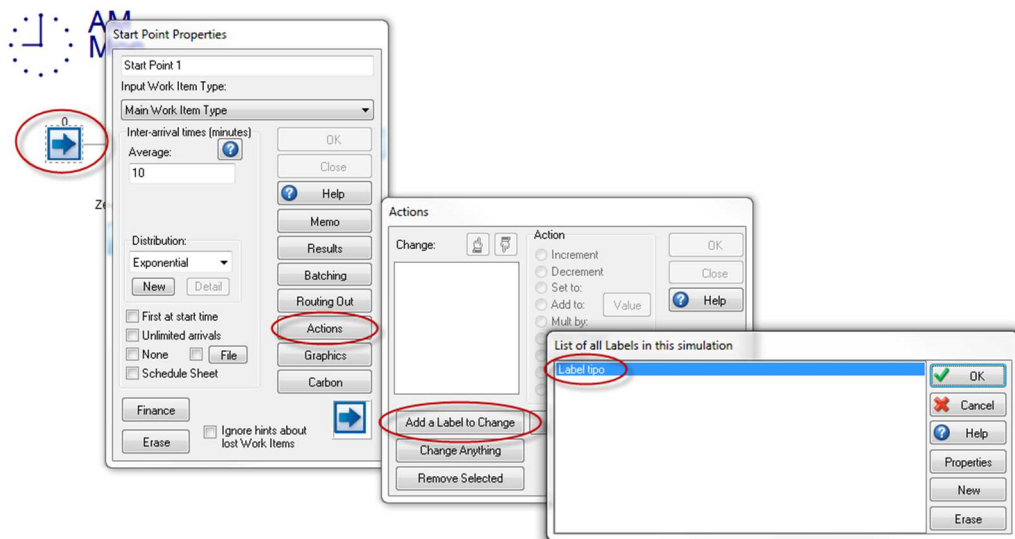


Figura 2.22 - Atribuindo um *label* para ser alterado no *Start Point*.

Em seguida, devemos atribuir ao “Label tipo” um valor segundo uma distribuição *Rounded Uniform*, de 1 até 3. Isto é possível pelo botão *Actions*, opção *Set to*, como representado na Figura 2.23.

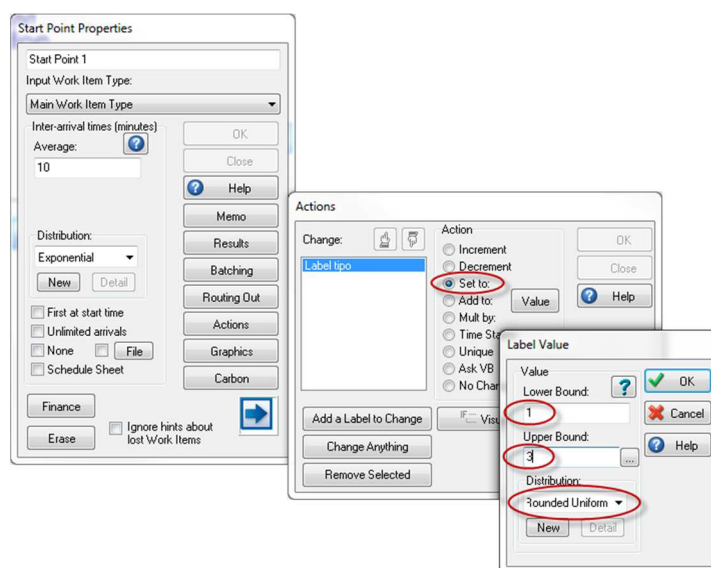


Figura 2.23 – Atribuindo uma distribuição uniforme para o label.

Em seguida, basta adicionar os recursos Huguinho, Zequinha e Luizinho ao *Activity 1*, ativando a opção *Select Resource By Label* e selecionando o *label* desejado (Figura 2.24). Assim, se a entidade tiver *label* com tipo = 1, o recurso Huguinho será alocado; se tipo = 2, o recurso Zequinha será alocado e, se tipo = 3, o recurso utilizado será o Luizinho, de acordo com a ordem definida na lista de recursos exigidos.

Importante: se a opção “*Select Resource by Label*” não tivesse sido selecionada, os três recursos seriam alocados **simultaneamente** para a realização da atividade, o que estaria errado.

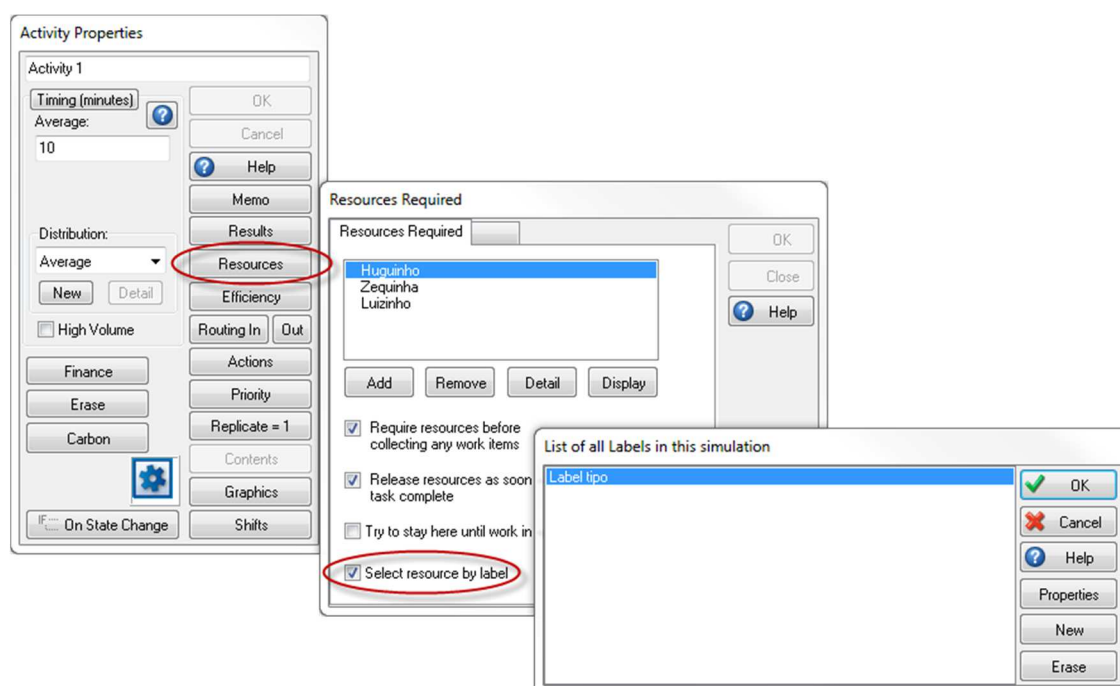


Figura 2.24 - Selecionando o recurso por meio de um label.

2.9. Erros comuns de alocação de recursos

Para finalizar este capítulo, consideramos que vale a pena destacar os erros mais comuns sobre alocação de recursos, reforçando alguns comentários já feitos anteriormente.

- Um erro comum é criar um recurso e se esquecer de alocá-lo em algum *Activity*. Para evitar isto, sugerimos, após a criação do modelo, dar um clique sobre cada recurso, sem soltar o botão do *mouse*, para que o SIMUL8 destaque com um círculo os *Activities* aos quais cada recurso está associado.
- Outro erro comum é não manter a compatibilidade entre o número de possíveis atividades em paralelo de um *Activity* (opção *Replicate*) e o número total de recursos disponíveis. Por exemplo, se você tiver um recurso com quantidade disponível = 3 e alocar este recurso a um *Activity* com *Replicate* = 1, somente um recurso será alocado ao longo da simulação (os outros dois ficarão “parados” durante toda a simulação). Para que isso não aconteça, ajuste a opção *Replicate* do *Activity* para um número maior ou igual do que o número de recursos disponíveis ou configure no recurso a opção *Auto Adjust Replicate Levels*.
- Por fim, um erro não tão incomum é adicionar dois recursos diferentes a um *Activity*, imaginando que ele utilizará ou um ou outro. Se associarmos n recursos diferentes a um *Activity*, estamos garantindo que este *Activity* só executará sua atividade se houver disponibilidade dos n recursos diferentes ao mesmo tempo (exceto quando se usa *Resource by Label*). Portanto, quando queremos que **ou** um **ou** outro recurso seja alocado, devemos utilizar um *pool* de recursos (veja a seção 2.6)

Capítulo 3 DISTRIBUIÇÕES ALEATÓRIAS NO SIMUL8

3.1. Introdução

Geralmente, simulamos o funcionamento de sistemas que são complexos demais para conseguirmos compreender totalmente sem o apoio de ferramentas computacionais – como o SIMUL8. Ao longo deste livro você deverá ser capaz de entender como é possível representar sistemas mais complexos do que o posto bancário do Capítulo 1. A complexidade de um sistema pode ser “medida” em função do número de processos, filas, recursos, entidades etc. que compõem o sistema real. Contudo, existe outro componente muito importante: a aleatoriedade dos fenômenos envolvidos. Por exemplo, um caixa bancário não atende todos os clientes gastando o mesmo tempo, nem os clientes chegam ao banco em intervalos de tempo constantes. Estes processos (atendimento e chegadas) exibem comportamentos aleatórios, em que não sabemos com certeza qual será o próximo valor em função dos valores anteriores. No entanto, podemos representar seu comportamento usando histogramas de frequências e, então, determinar qual o melhor modelo teórico capaz de representar o processo.

Parece complicado, mas a solução é muito mais simples do que o problema em si, como veremos no exemplo a seguir.

3.2. Exemplo inicial

A Figura 3.1 representa um terminal portuário que possui apenas um berço de atracação. As embarcações chegam de forma aleatória e, se encontrarem o berço livre, iniciam a manobra de atracação. Se o berço estiver ocupado, elas aguardam em fila até a sua liberação. Após atracar, a embarcação permanece no berço durante o tempo de operação propriamente dito (carga e/ou descarga da embarcação). Terminada a operação, a embarcação inicia a manobra de desatracação e, somente após o término desta, o berço será liberado para uma próxima embarcação.

CHEGADAS DE NAVIOS

EXPONENCIAL(36)
30% PANAMAX
70% SUEZMAX

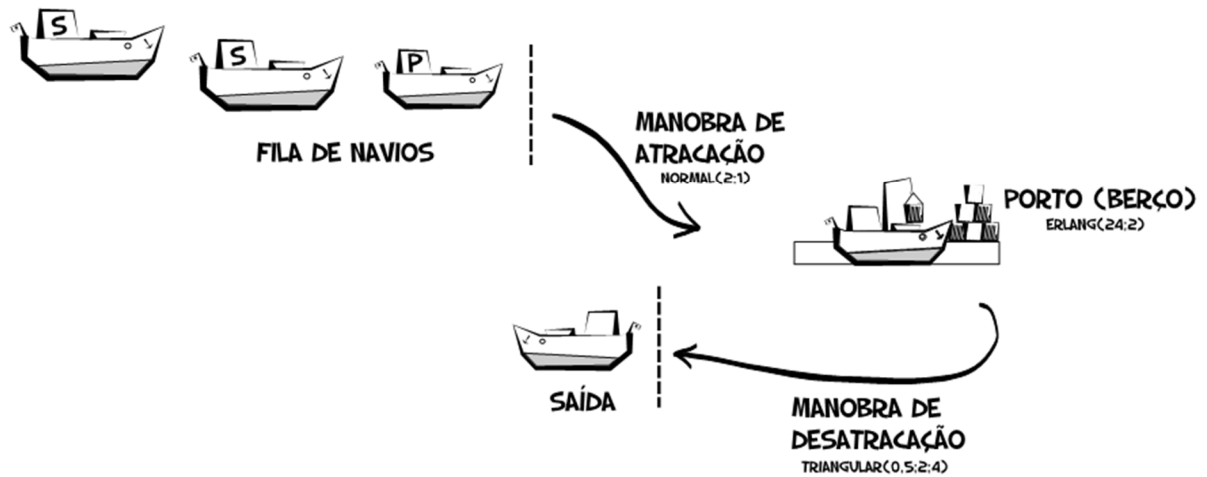


Figura 3.1 - Representação esquemática do exemplo do porto.

O centro de controle portuário construiu uma amostra dos intervalos sucessivos entre as chegadas de 100 navios e obteve o histograma mostrado na Figura 3.2.

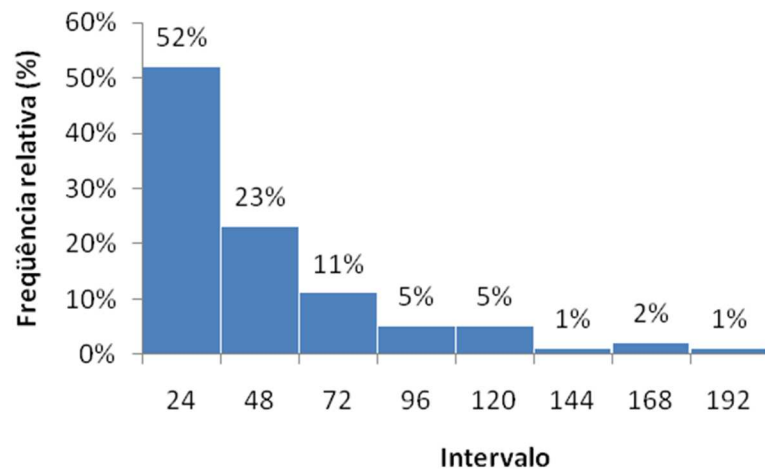


Figura 3.2 - Histograma de frequências para o intervalo de tempo (em horas) entre chegadas sucessivas.

3.2.1. Histograma

Um histograma, usualmente, apresenta no eixo vertical a frequência com que ocorre um determinado fenômeno, que é representado no eixo horizontal. No nosso exemplo (Figura 3.2), verificamos que em 52% dos casos (eixo vertical), o tempo entre chegadas de dois navios sucessivos ficou dentro do intervalo de 0 a 24 h, o que em termos de operação portuária – onde muitos tempos operacionais são contados em dias – é como dois navios chegassem praticamente juntos. Mais do que isso, podemos dizer que, e isso é importante, “a chance de que dois navios cheguem em um mesmo dia (intervalo inferior a 24 h) é de 52%”. É esta “chance” de 52% que nós queremos que o SIMUL8 utilize em seus cálculos, sorteando aleatoriamente os tempos entre chegadas de navios, de modo que se respeite o histograma informado.

Assim, o trabalho computacional será do SIMUL8, enquanto para nós fica o trabalho de informá-lo qual é o comportamento do fenômeno, seja por meio de histogramas (como o da Figura 3.2), seja por meio de distribuições estatísticas, como faremos a seguir.

O Apêndice I apresenta o Stat::Fit, um *software* bastante prático para identificação de distribuições estatísticas a partir de dados reais, um processo conhecido como “ajuste de dados” ou “identificação da distribuição teórica”. Por exemplo, seguindo passo a passo o que é apresentado no Apêndice I, você descobrirá que o conjunto de dados do intervalo entre embarcações adere a uma distribuição exponencial (como na maioria dos casos em que chegadas aleatórias e externas ao sistema estão envolvidas). Para o SIMUL8 só interessará a definição desta informação: qual distribuição teórica você deseja que ele utilize, juntamente com os parâmetros desta distribuição.

Para o exemplo do porto, já sabemos que os intervalos entre chegadas sucessivas são exponenciais. Os outros tempos envolvidos no processo são: o tempo da manobra de atracação, o tempo de operação e o tempo de desatracação. Considere que o centro de controle do porto analisou uma amostra de 100 embarcações e, utilizando o Stat::Fit, identificou as seguintes distribuições teóricas:

- Tempo de atracação: distribuição normal, com média de 2 horas e desvio padrão de 1 hora;
- Tempo de operação: distribuição Erlang de ordem 2, com média de 24 horas;
- Tempo de desatracação: nesta parte, o porto não tinha dados coletados. Consultando operadores experientes, foi sugerido que se utilizasse uma distribuição triangular com menor valor igual a 0,5 hora, maior valor igual a 4 horas e moda (valor mais frequente) igual a 2 horas.

A Figura 3.3 apresenta um possível modelo para representar este sistema. Sugerimos que você construa este modelo para poder acompanhar as discussões que faremos sobre distribuições no SIMUL8. Um detalhe de importante é que como somente um único navio pode ficar atracado por vez, a alocação do berço deve ser realizada na atividade *Manobra*

de atracação e a liberação na atividade *Manobra de desatracação*. Assim, tome os seguintes cuidados ao ocupar o recurso berço:

- Na atividade *Manobra de atracação*: ocupe o recurso *berço* com a regra de alocação *Require here but do not release the resource*;
- Na atividade *Porto*: ocupe o recurso *berço* com a regra de alocação *Display only*;
- Na atividade *Manobra de desatracação*: libere o recurso *berço* com a regra de alocação *Only release the resource here*.

Considere o tempo de simulação de 10 anos (87.600 horas).

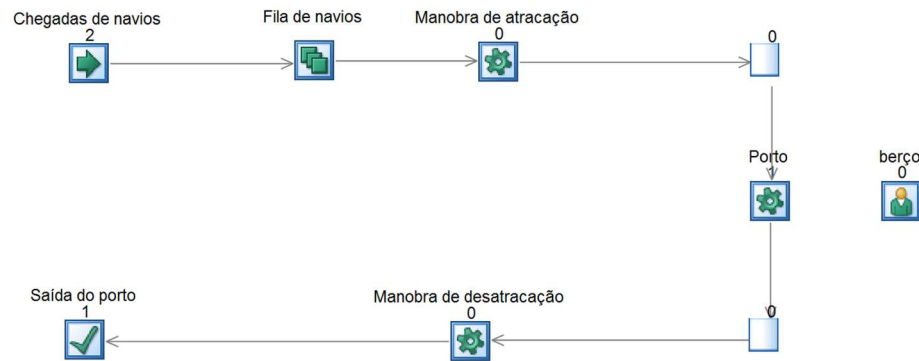


Figura 3.3 - Representação do modelo do porto no SIMUL8.

3.3. Distribuições disponíveis no SIMUL8

O SIMUL8 é capaz de trabalhar com as principais distribuições de probabilidade conhecidas, bastando que você forneça os parâmetros das distribuições para que ele se encarregue do resto. A seguir, comentaremos as principais opções disponíveis:

- *Average*: é uma distribuição normal com o coeficiente de variação pré-definido em 25%, ou seja, o desvio-padrão vale 25% do valor da média;
- *Beta*: é uma distribuição bastante flexível e seus parâmetros podem ser obtidos facilmente por meio do Stat::Fit;
- *Erlang*: a distribuição Erlang de ordem k representa uma sequência de distribuições exponenciais. Assim, para $k=1$, ela é a própria distribuição exponencial. Para $k=n$, ela representa a soma de n valores com distribuição exponencial. Ao passo que se aumenta a ordem k (tipicamente, $k \geq 10$), a distribuição se assemelha à distribuição normal. Com $k > 20$, a distribuição erlang é praticamente determinística (ou seja, fornece valores constantes). No SIMUL8, a média informada para esta distribuição é a média total, e não a média de cada distribuição exponencial;
- *Exponential*: uma distribuição que possui coeficiente de variação igual a 1 e que representa quase todos os processos de chegadas que são externas ao sistema. É assim que, geralmente, pessoas chegam a um banco, carros chegam ao pedágio, pessoas chegam a uma sala de reuniões etc;
- *Fixed*: distribuição que, na verdade, não representa aleatoriedade, já que é definida por um valor fixo, determinístico, constante.
- *Gamma*: é uma distribuição que pertence à família das distribuições exponenciais e que é pouco utilizada na prática, mas é importante, pois dela são derivadas as distribuições Beta, Weibull etc.;
- *Lognormal*: distribuição geralmente associada ao tempo de execução de tarefas. Possui como característica mais marcante a probabilidade zero para valores pequenos (em oposição à exponencial, que tem maior ocorrência para valores pequenos);

- *Normal*: distribuição simétrica, usual em fenômenos de menores variabilidades que a distribuição exponencial. Em modelos de simulação, suas caudas tendem a $-\infty$ e a $+\infty$, o que pode causar problemas graves (veja mais adiante a opção *Bounded*);
- *Pearson V*: as distribuições de Pearson são geralmente utilizadas para modelar tempos de espera, que sempre começam com um valor mínimo > 0 e são ilimitados por um valor máximo;
- *Pearson VI*: difere da anterior pois tem valor mínimo igual a 0;
- *Poisson*: distribuição discreta, geralmente utilizada para identificar o número de ocorrências de determinado evento. Esta distribuição é equivalente à exponencial: se t é o tempo médio entre chegadas consecutivas de uma distribuição exponencial, então $1/t$ é o número de ocorrências de chegadas em 1 unidade de tempo, segundo uma distribuição de Poisson;
- *Triangular*: utilizada quando conhecemos o menor valor, o maior valor e o mais provável (moda) da variável em questão;
- *Uniform*: utilizada quando qualquer valor dentro de uma faixa é igualmente provável. Esta distribuição pode ser utilizada em casos de falta de informações sobre a distribuição que rege o fenômeno, ao mesmo tempo em que se admite uma distribuição com alto coeficiente de variação;

O SIMUL8 possui dois tipos de distribuição uniforme: *Uniform* e *Rounded Uniform*. A primeira gera valores reais (fracionários), enquanto a segunda gera somente valores inteiros dentro do intervalo fornecido.

- *Weibull*: uma das distribuições mais flexíveis, pertencente à família das distribuições exponenciais. Muito comum para representar tempos relativos à quebra de equipamentos em modelos que envolvem manutenção.

Além dessas, você ainda pode construir a sua própria distribuição a partir de combinações de distribuições (*Combination*) ou de histogramas (*Probability Profile*), como veremos a diante.

3.4. Aplicação prática do exemplo inicial

3.4.1. Informando as distribuições ao SIMUL8

Como discutido no Capítulo 1, a maneira usual de se informar a distribuição desejada para o SIMUL8 é por meio da janela de propriedades do objeto em questão. Para o problema do porto, por exemplo, a distribuição do intervalo entre chegadas de navios pode ser fornecida diretamente na janela do objeto *Start Point*, como indicado na Figura 3.4.

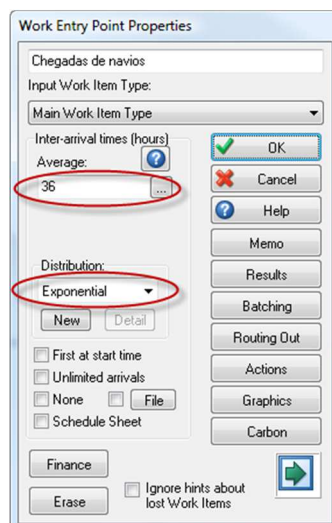


Figura 3.4 - Start Point Chegadas de Navios.

Contudo, o recomendado para melhor produtividade é inserir distribuições personalizadas: estas são criadas ao se clicar no botão *New*, logo abaixo do campo *Distribution* – vide Figura 3.4 (ou em *Data and Rules* | *Distributions*). Para o exemplo, podemos criar uma distribuição chamada “Intervalo de chegadas de navios”, como mostrado na Figura 3.5.

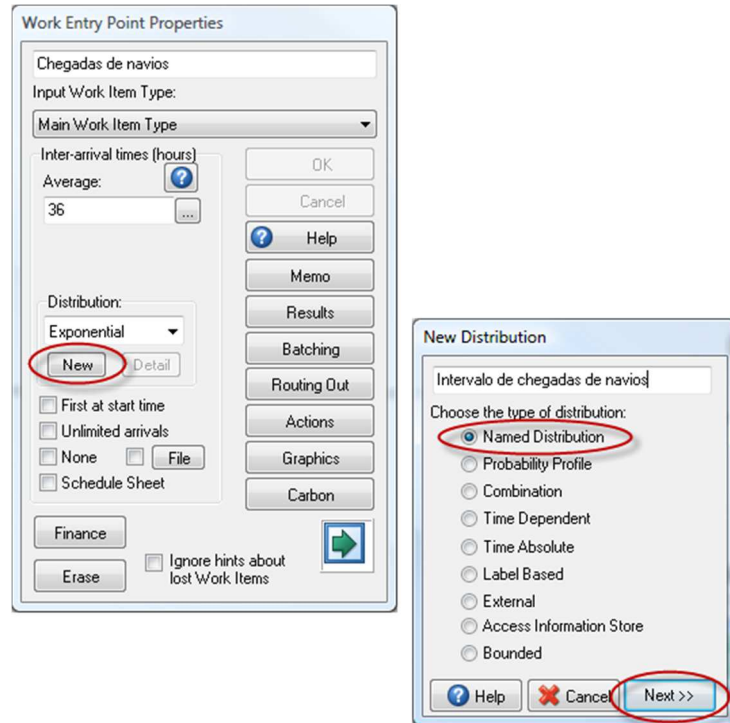


Figura 3.5 - Criação da distribuição “Chegadas de Navios”.

Como se pode notar na janela *New Distribution* (Figura 3.5), existem diversas opções para a construção de novas distribuições. Por hora, ficaremos com a primeira opção, *Named Distribution*, que permite construir uma nova distribuição a partir de alguma distribuição já existente. Clicando em *Next >>* surge a janela representada na Figura 3.6 onde informamos os parâmetros da distribuição – no caso, uma distribuição exponencial com média de 36 horas.

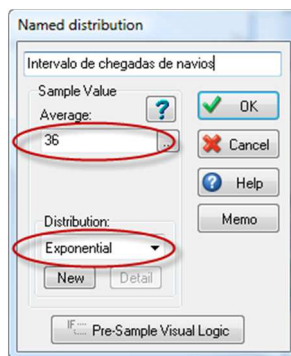


Figura 3.6 - Informando os parâmetros da distribuição exponencial para o SIMUL8.

Você pode estar se perguntando: qual foi a vantagem em se criar uma distribuição própria para representar uma distribuição exponencial, com média igual a 36 horas? Neste modelo, nenhuma! Imagine, no entanto, que estivéssemos construindo um modelo mais complexo, em que fosse necessário utilizar a mesma distribuição em vários objetos. Neste caso, ao invés de informar “Exponencial, média = 36” em cada objeto, bastaria selecionar a distribuição “Intervalo de chegadas de navios”, que estará disponível no final da lista de distribuições reconhecidas pelos objetos. Além disso, se o valor do tempo médio entre chegadas passasse a ser de 38 horas, bastaria alterar o parâmetro da distribuição, ao invés de editar individualmente todos os objetos com a distribuição em questão.

Se você definir as outras distribuições por esse método e nomeá-las como “Tempo de manobra de atracação”, “Tempo de operação no berço” e “Tempo de manobra de desatracação”, poderá encontrá-las todas disponíveis para edição em um único menu: em *Data and Rules* | *Distributions*, clicando na parte de baixo do ícone *Distributions*, será mostrada uma lista de opções como a da Figura 3.7. Caso seja clicado o próprio ícone *Distributions*, surgirá uma janela como a da Figura 3.8, onde você poderá editar (*Properties*), criar uma nova distribuição (*New*), copiar (*Copy*) ou apagar (*Erase*) alguma distribuição já criada¹³.

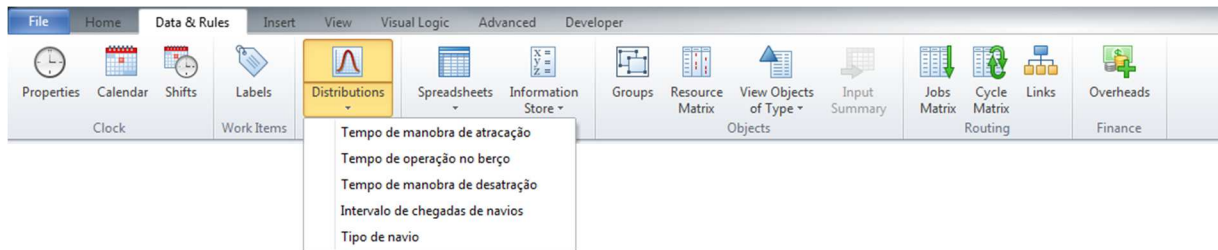


Figura 3.7 - Janela com todas as distribuições criadas no modelo até o momento.

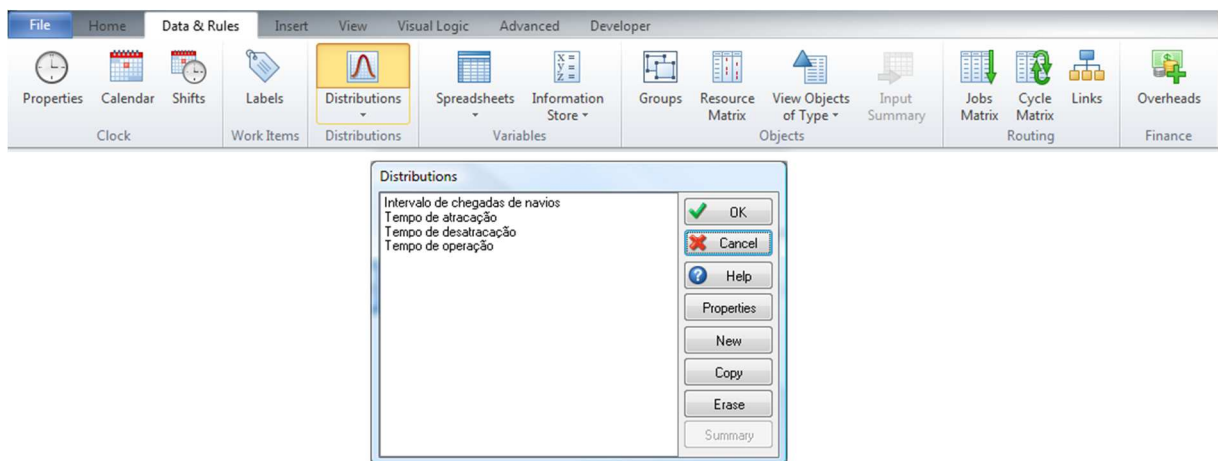


Figura 3.8 – Janela de distribuições obtida ao se clicar no ícone *Distributions* do menu *Data and Rules*.

3.4.2. Utilizando distribuições com *labels*

Considere que o modelo do terminal portuário deve contemplar, adicionalmente, o tipo de embarcação atendida. Atualmente, o porto atende dois tipos distintos de embarcações: Panamax e Suezmax. As embarcações Panamax são menores e tem tamanho compatível com a travessia do canal do Panamá, enquanto as Suezmax são maiores e de tamanho compatível com a travessia do canal de Suez (daí os nomes das embarcações). Os dados históricos mostram que 30% das embarcações que frequentam nosso porto são do tamanho de um Panamax, enquanto 70% são do tamanho de um Suezmax (isto foi indicado na Figura 3.1, mas não havia sido comentado).

Pelo colocado, deveremos nomear as entidades por um *tipo* e essa nomeação deve seguir um padrão: 30% de um tipo e 70% do outro. Como discutido parcialmente no Capítulo 2, isto pode ser feito por meio de *labels*. Acesse o menu *Data and Rules* e selecione a opção *Labels*, que abrirá uma janela com a lista de todos os *labels* utilizados atualmente no modelo (Figura 3.9).

¹³ Uma regra comum entre usuários do SIMUL8 é criar algum prefixo inicial para distinguir uma distribuição de outro objeto do modelo, o que é extremamente útil quando estamos utilizando a lógica de programação (Capítulo 5). Assim, uma sugestão é começar os nomes das distribuições com o prefixo “D_”. Por exemplo, o “D_tempo de operação no berço”, “D_tempo de atracação” etc.

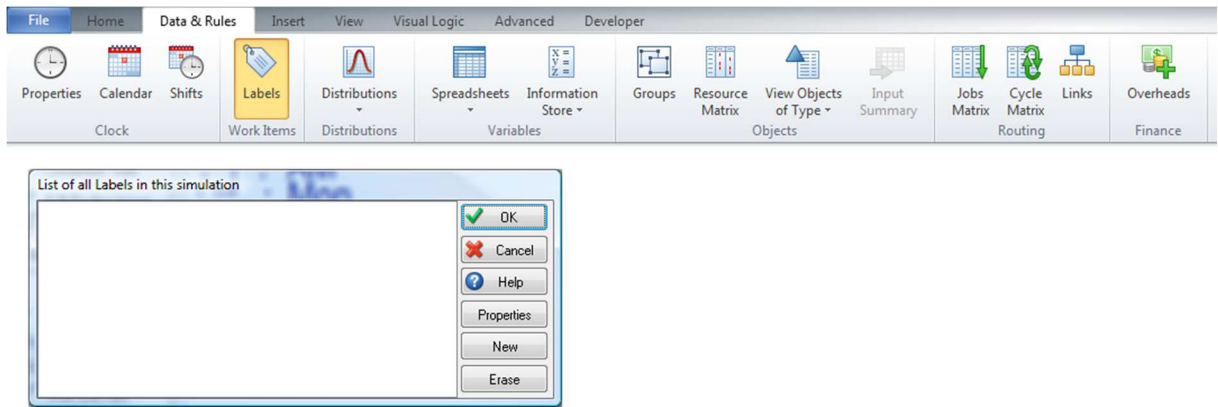


Figura 3.9 - Criação de labels a partir do menu *Data and Rules*.

Clique no botão *New* e crie um *label* numérico chamado “Tipo” (Figura 3.10).

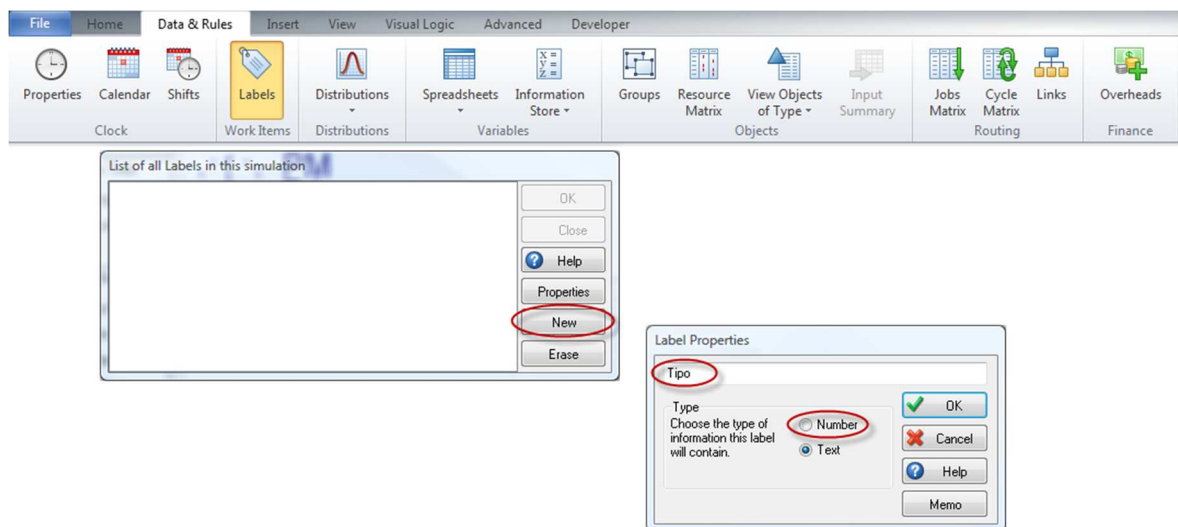


Figura 3.10 - Janela de propriedades do *label*, após de clicar no botão *New*>>.

Precisamos, agora, criar uma distribuição para representar que 30% dos navios são do tipo Panamax e 70% são do tipo Suezmax. A partir do menu *Data and Rules*, clique no ícone da opção *Distributions* e, em seguida, clique no botão *New* para criar a distribuição “Tipo de navio”. A opção selecionada deve ser *Probability Profile*, pois o que vamos definir são as probabilidades de ocorrência de cada tipo de navio (30% e 70%).

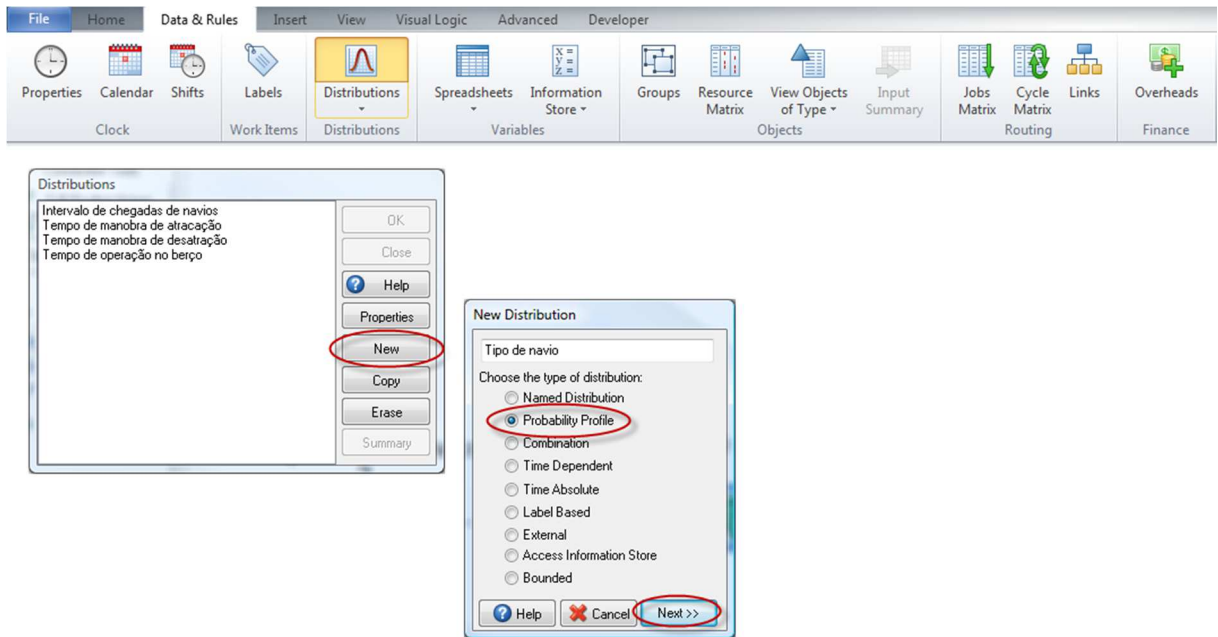


Figura 3.11 - Distribuição para o tipo de navio baseada em *Probability Profile*.

Quando clicamos no botão *Next >>* da janela *New Distribution*, obtemos uma janela (Figura 3.12) onde é possível construir distribuições quaisquer (discretas ou contínuas) arrastando-se as barras do histograma com o *mouse* ou digitando-se valores nos campos *Percent* e *Value*. Inicialmente, vamos à diversão: experimente clicar e arrastar com o *mouse* alguma barra azul do histograma. Note que, conforme você manipula a forma da distribuição, os campos *Percent* e *Value* são alterados automaticamente. No canto inferior direito, se você selecionar a opção *Continuous* a distribuição deixa de fornecer valores inteiros e passa a fornecer valores interpolados pelos respectivos valores das barras do histograma.

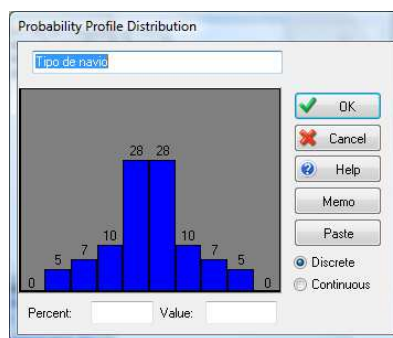


Figura 3.12 - Definição de distribuição discreta com o *Probability Profile*.

Depois da diversão, a obrigação! Clique com o *mouse* na primeira barra (a do "0") e preencha o campo *Percent* com o valor 30. Em seguida, selecione a próxima barra e preencha o campo *Percent* com o valor 70. Com isso, teremos construído, uma distribuição discreta de probabilidades que fornecerá, em 30% das vezes, o valor 1 e em 70% das vezes o valor 2 (Figura 3.13).

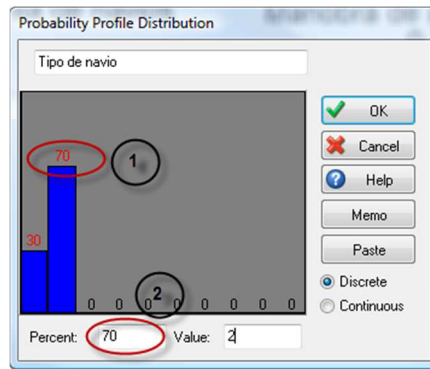


Figura 3.14 - *Probability Profile* para o tipo de navio.

Apague as outras barras, utilizando a opção *Delete* com o botão direito do *mouse*, como mostrado na Figura 3.15. Outra possibilidade para definir uma distribuição discreta é criar a tabela de probabilidade *versus* valores no Excel e, em seguida, copiar e colar os valores no SIMUL8 clicando no botão *Paste*, que aparece na lateral da janela *Probability Profile Distribution* (Figura 3.15). Esta opção é muito útil quando o número de barras ou classes for grande.

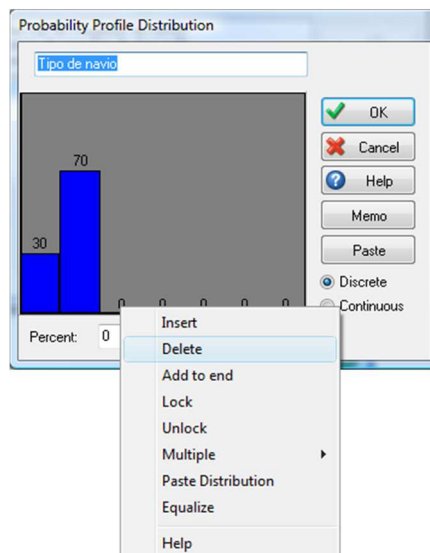


Figura 3.15 – Selecione a barrinha que deseja eliminar e a seguir, clicando com o botão direito do *mouse*, surge um novo menu suspenso com diversas opções. Selecione a opção *Delete* para apagar a barra sem uso.

Agora que criamos o *label* “Tipo” e uma distribuição que irá atribuir um valor (1 ou 2) para este *label*, falta juntarmos as peças. Isso é feito pelo botão *Actions*, encontrado nas janelas *Start Point* ou *Activity*.

A Figura 3.16 ilustra a sequência a ser usada neste exemplo: no *Start Point* “Chegadas de navios”, ao selecionar o botão *Actions*, abriremos uma janela que permite associar um *label* criado previamente (“Tipo”) a cada entidade que entra no modelo (botão *Add Label to Change*). Nesta janela de ações, como queremos que um valor seja atribuído ao *label*, devemos selecionar a opção *Set to*: seguida pela seleção da distribuição “Tipo de navio”, como mostrado na Figura 3.17.

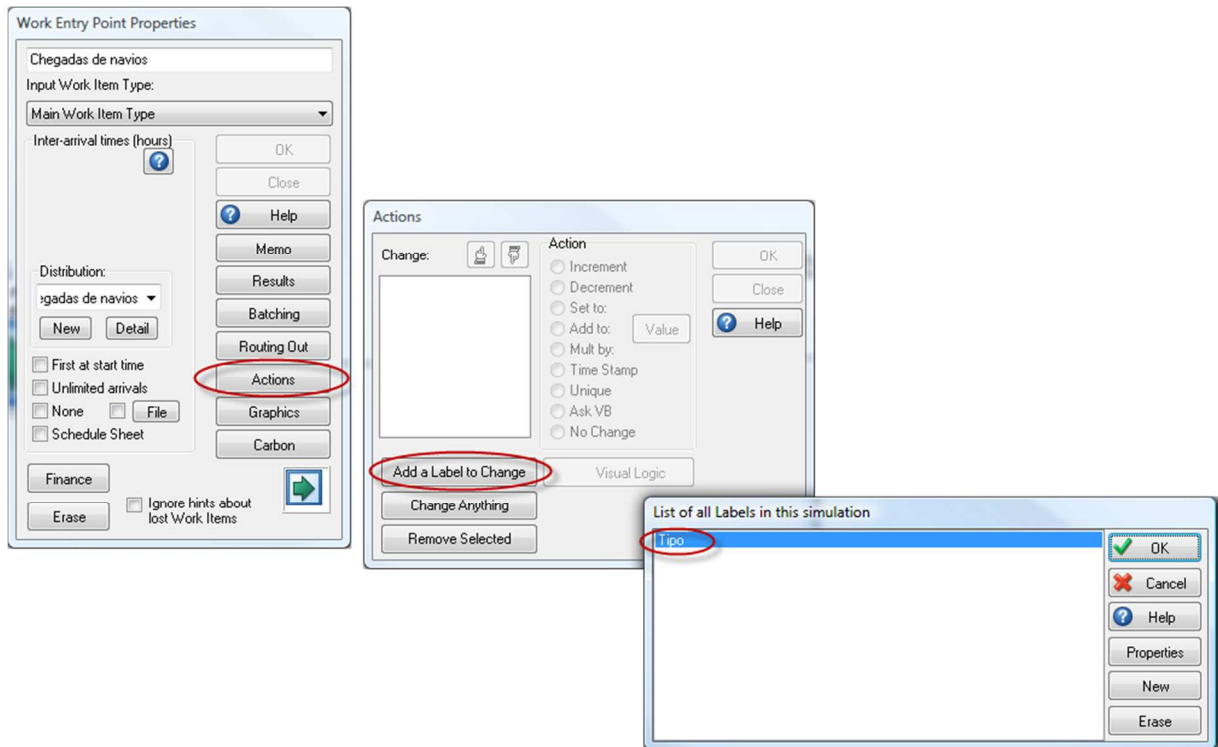


Figura 3.16 – Sequência de comandos para associação do label “Tipo” às entidades do modelo.

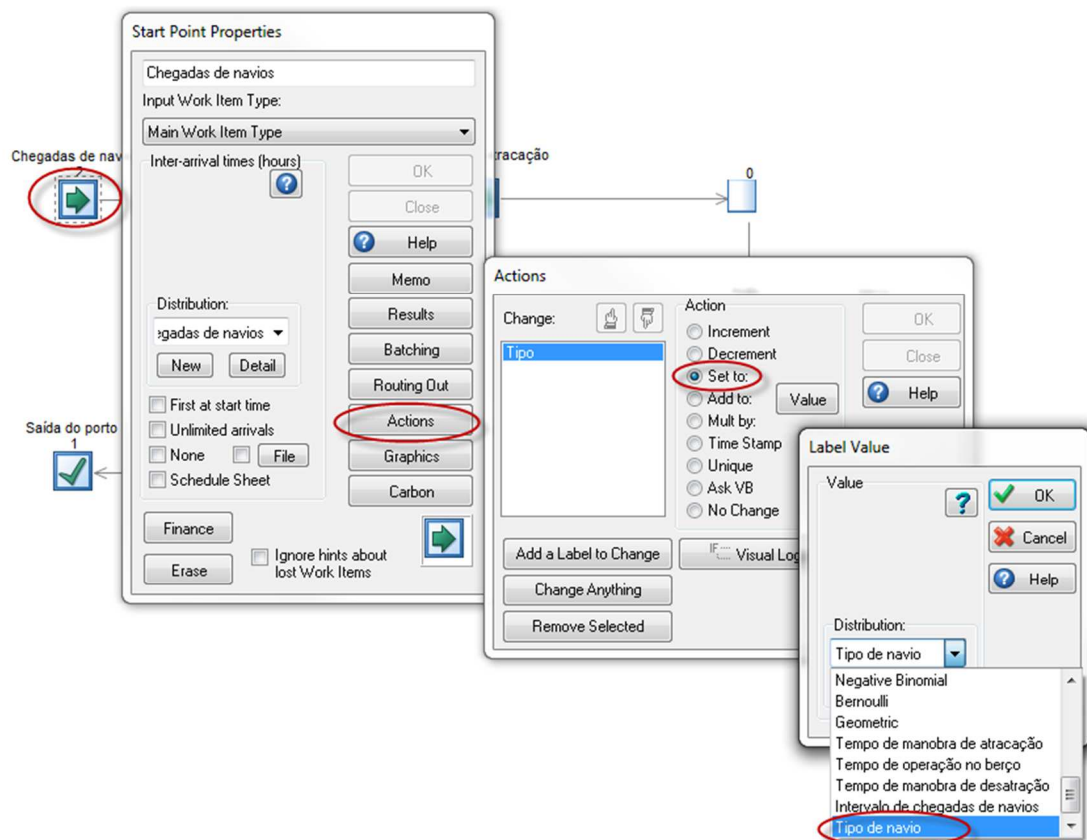


Figura 3.17 – Atribuição de uma distribuição (ou um valor) ao label que representa o tipo do navio.

Vale a pena observar que um valor numérico armazenado em um *label* também pode ser usado como tempo de execução de um *Activity* (parâmetro *Timing*).

3.4.3. Distribuição *Bounded*: limitando os valores máximos e mínimos da distribuição

O tempo de duração da atracação possui distribuição normal com média de 2 horas de duração e desvio padrão de 1 hora. Esta distribuição pode ser visualizada na Figura 3.18.

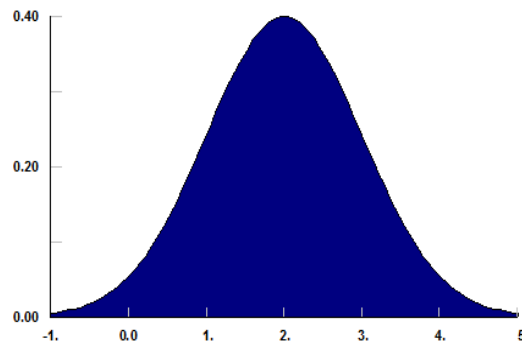


Figura 3.18 – Distribuição normal com média = 2 e desvio padrão = 1.

Repare na cauda para a esquerda da distribuição e onde situa o valor zero. Se você for um observador atento, deve ter notado que parte da curva contém valores *negativos* para o tempo de atracação. A menos que estejamos simulando um sistema com capacidade de voltar no tempo, é possível que o SIMUL8 sorteie um navio com tempo de atracação de -0,2 horas!

É nesse ponto que entram as distribuições do tipo *Bounded*, que impõem limites (*bounds*) inferiores e superiores para as distribuições teóricas envolvidas. Faça isso para o nosso exemplo, limitando o tempo de atracação mínimo em 0,5 horas e o máximo em 4 horas. Para isso, você terá que criar uma nova distribuição, chamada, por exemplo, “Tempo de atracação limitado”, como indicado na Figura 3.19 e na Figura 3.20.

Não se esqueça de configurar o *Activity* "Manobra de atracação" para utilizar a nova distribuição criada.

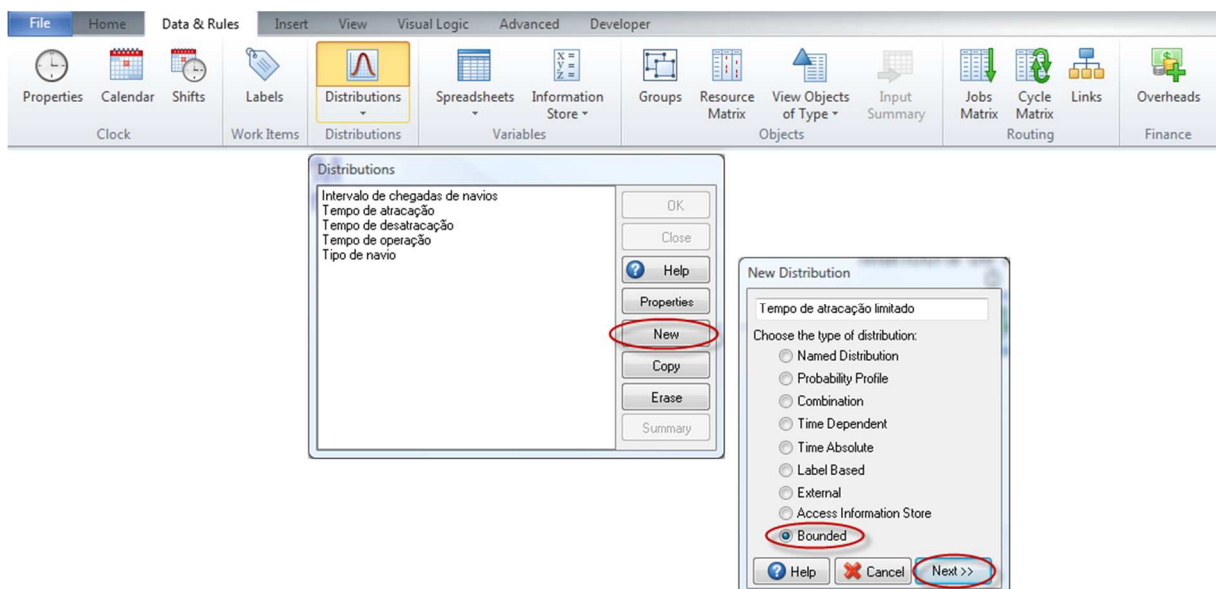


Figura 3.19 - Nova distribuição limitada (*bounded*).

Figura 3.20 - Distribuição Normal limitada entre 0,5 e 4 horas.

Sempre que você estiver trabalhando com distribuições teóricas, lembre-se de utilizar a distribuição *bounded* não só para valores negativos, mas também para valores muito altos, que você sabe que nunca ocorrerão no sistema.

3.4.4. *Combination*: criando uma nova distribuição a partir da soma de outras

O SIMUL8 permite combinar ou *somar* o efeito de duas distribuições. Por exemplo, considere que 80% das embarcações esperam um tempo adicional de 3 horas para desatracar (além do seu tempo normal de desatracação) devido à prioridade de tráfego de embarcações estabelecida pela Guarda Costeira.

Com a opção *Combination*, você pode construir uma nova distribuição que soma, em 80% dos casos, 3 horas à distribuição original de desatracação (nos 20% restantes dos casos será usado o próprio tempo de desatracação original). Para tanto, inicialmente, construa uma distribuição discreta “Porcentagem dos navios que esperam a Guarda Costeira”, utilizando a opção *Profile Distribution*. Atribua a 80% dos casos o valor 3 e a 20% dos casos o valor 0, como representado na Figura 3.21.

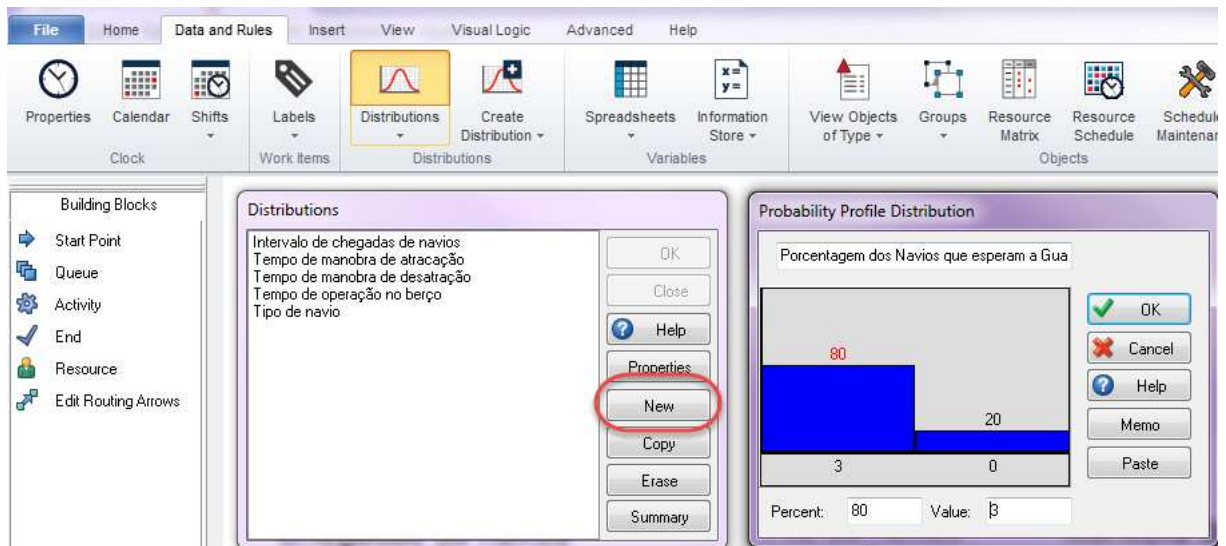


Figura 3.21 - Distribuição para o tempo adicional de desatracação devido à regra da Guarda Costeira.

Em seguida, crie uma nova distribuição “Tempo de desatracação combinado”, utilizando a opção *Combination* (Figura 3.22).

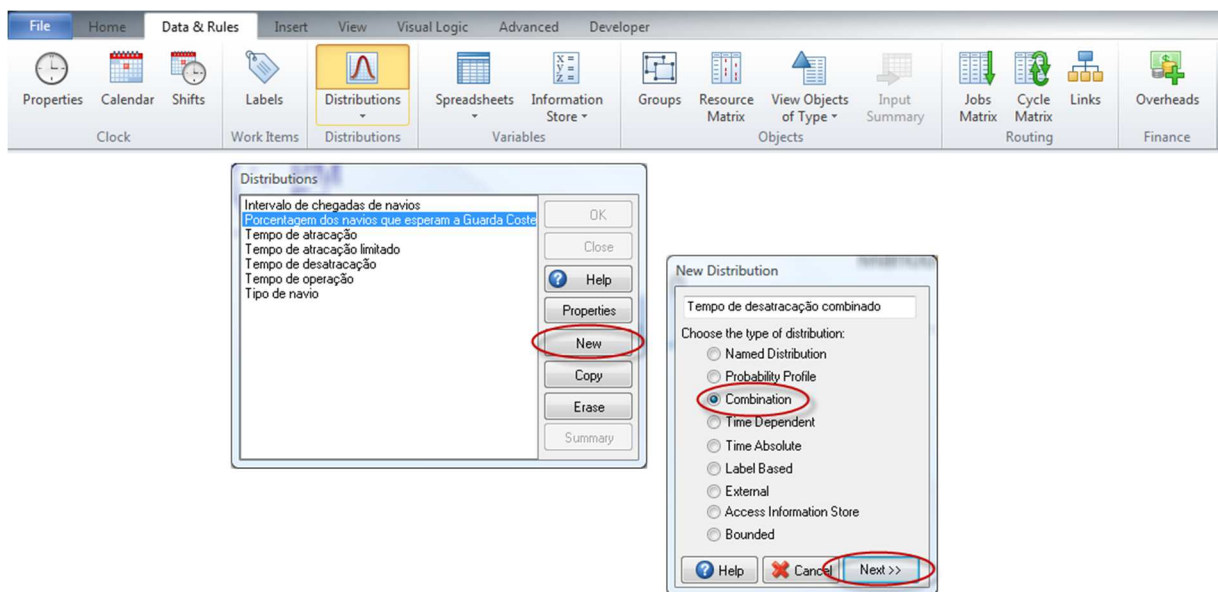


Figura 3.22 - Criando a nova distribuição para o tempo de desatracação do navio a partir da opção *Combination*.

Agora é hora de combinar os tempos. Com o botão *Add*, adicione a distribuição criada (“Porcentagem dos navios...”) e a anterior (“Tempo de manobra de desatracação”), como mostrado na Figura 3.23.

Simule o modelo, sem se esquecer de utilizar a nova distribuição no *Activity* “Manobra de desatracação”.

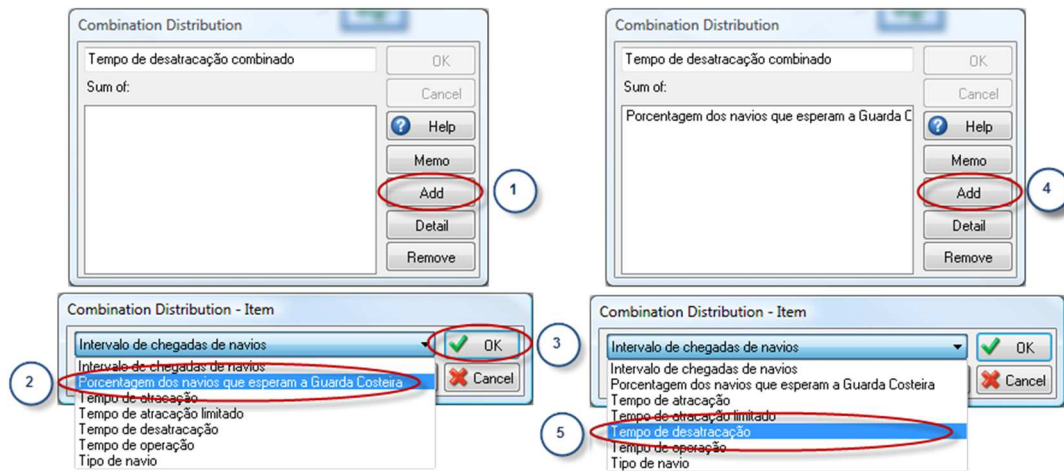


Figura 3.23 - Com o botão *Add*, adicione as duas distribuições (1) "Porcentagem dos..." e (2) "Tempo de atracação", formando uma nova distribuição que é a soma das duas.

Fique atento ao comportamento do conjunto de dados que você está utilizando. Muitas vezes, valores não usuais podem representar algum estado válido do sistema (por exemplo, em parada para manutenção ou afetado por sazonalidades) e são bons candidatos ao uso da opção *Combination*. A nossa prática sugere, sempre que possível, trabalhar com uma amostra de dados muito superior àquela necessária para identificação da distribuição estatística que melhor se ajusta aos dados.

3.4.5. *Time Dependent*: criando distribuições que se alteram com o tempo

Considere que, devido à baixa luminosidade, a manobra de atracação noturna deve ser realizada a uma velocidade menor do que a utilizada durante o dia. Assim, das 6 às 18 h, o tempo médio de atracação deve ser igual ao tempo original (2 horas), enquanto das 24 h até as 6 h do dia seguinte, o tempo médio de atracação deve ser de 4 horas. Utilizando uma distribuição do tipo *Time dependent* (Figura 3.24), podemos criar uma nova distribuição chamada "Tempo de atracação por horário", que é uma distribuição para o tempo de atracação que considera as faixas de horário.

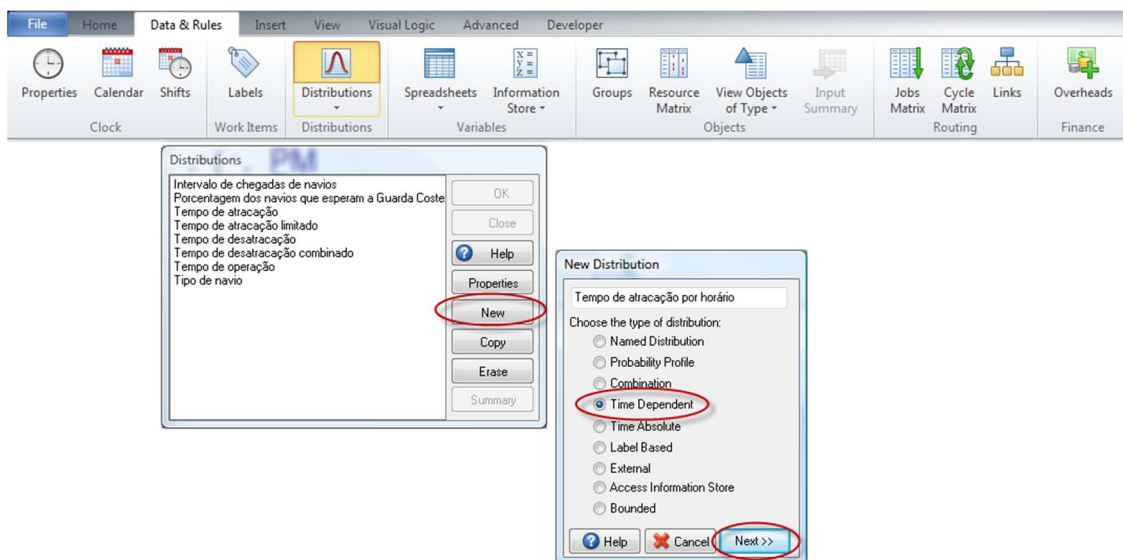


Figura 3.24 - Criação de uma nova distribuição com a opção *Time Dependent*.

Na janela *Time Dependent Distribution* (Figura 3.25), com o botão *Add*, podemos definir os intervalos de tempo para a distribuição. No caso, a distribuição criada previamente, “Tempo de manobra de atracação”, será atribuída ao intervalo que se inicia às 6 horas Figura 3.25.

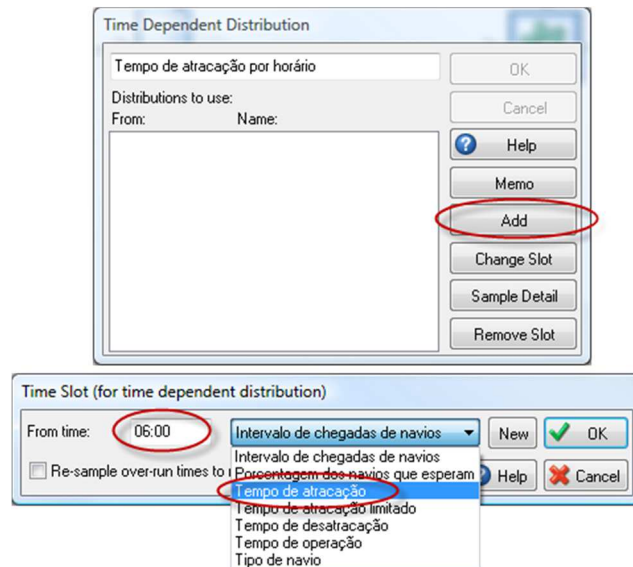


Figura 3.25 - Ao clicar o botão *Add* da Janela *Time Dependent*, temos a janela *Time Slot* para marcação do início do intervalo de tempo em que o tempo de atracação será governado pela distribuição antiga de manobra de atracação.

Antes de definirmos a distribuição válida a partir das 18:00 horas, precisamos criá-la! Para isso, basta clicarmos no botão *New* e utilizarmos a opção *Named Distribution* para criarmos a nova distribuição “Tempo de atracação noturna” (Figura 3.26).

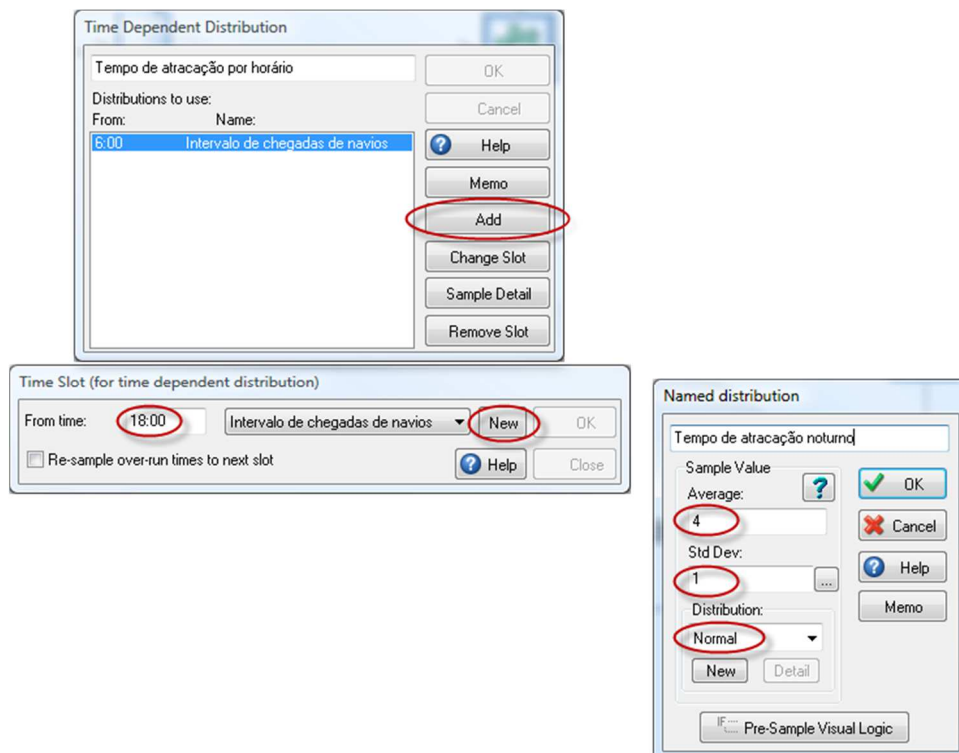


Figura 3.26 - Janela para criar uma nova distribuição, que se inicia às 18 horas e representa o tempo de atracação noturno, uma normal de média 4 e desvio padrão de 1 hora.

Finalmente, clicando no botão *Add*, podemos definir o novo intervalo, válido a partir das 18:00 horas, associando a distribuição adequada (“Tempo de atracação noturna”).

As distribuições *Time Dependent* são particularmente úteis para simular sistemas que apresentam sazonalidades ou variações temporais conhecidas. Um bom exemplo do que estamos falando é a representação do fluxo de clientes em uma padaria ao longo do dia.

3.4.6. *Time absolute*: distribuições “com hora marcada”

Distribuições do tipo *Time Absolute* são distribuições que mudam exatamente em determinada hora, ou seja, com “hora marcada”. Por exemplo, considere que, toda noite às 0:00 h, uma balsa de abastecimento atravessa a região portuária. Com a opção *Time Absolute* é possível definir a hora de ocorrência (Figura 3.27).

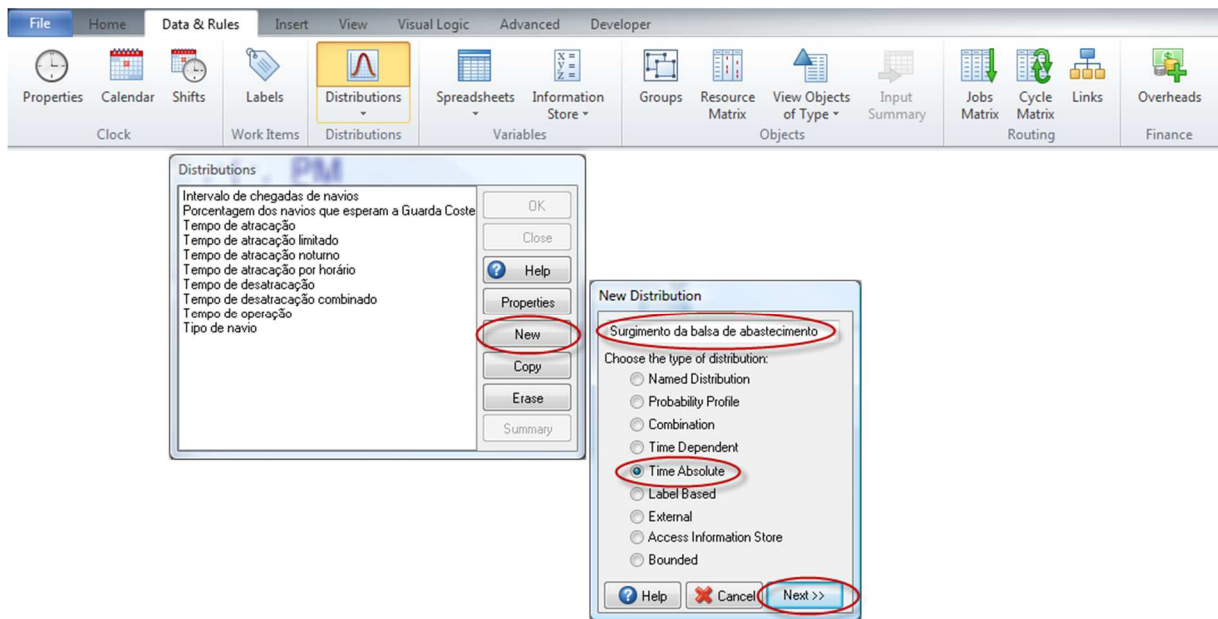


Figura 3.27 - Opção Time Dependent.

Adicionalmente, a janela *Time Absolute* permite que se defina uma distribuição estatística que será *somada* ao tempo fornecido. Seria como dizer que a balsa atravessa o canal às 12 horas, mas geralmente atrasa entre 0 e 1 hora. Isso pode ser feito adicionando-se uma distribuição uniforme com intervalo entre 0 e 1 (Figura 3.28). Com esta nova distribuição (“Surgimento da balsa de abastecimento”), convidamos o leitor a criar, como um exercício de fixação, um “bloqueio” diário nas operações de atracação e desatracação no modelo do porto.

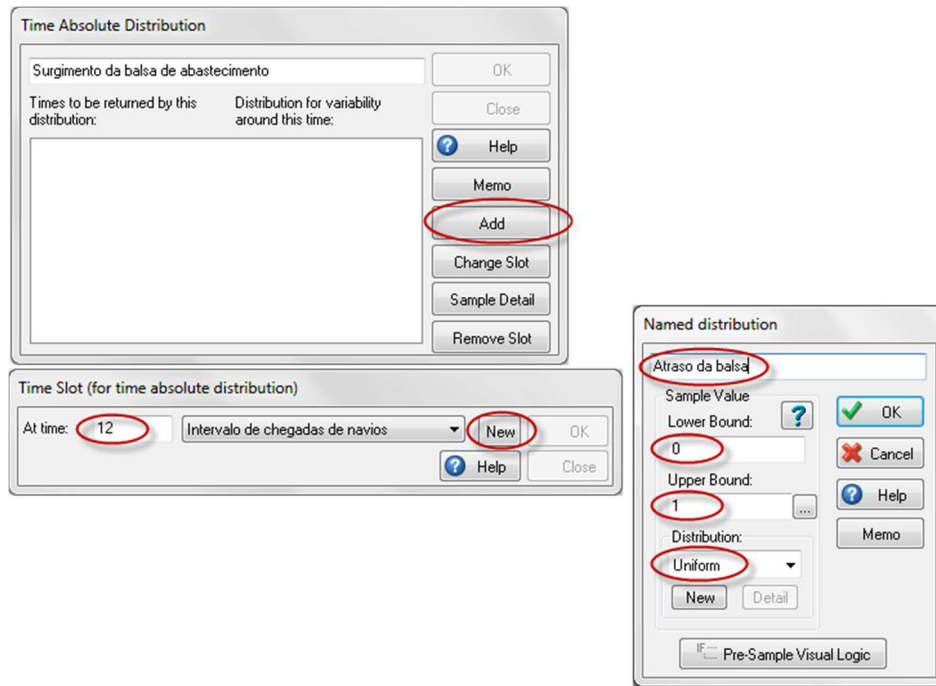


Figura 3.28 - Janela Time Slot para uma nova distribuição criada com a opção *Time Absolute*.

3.4.7. *Label Based*: armazenando o nome da distribuição em um *label*

Apesar de aparentar ser um pouco trabalhosa de se utilizar, como veremos nesta seção, a distribuição *Label Based* simplifica a representação de situações em que entidades distintas são criadas por diferentes pontos de entrada (*Start Points*), passam por um mesmo processo (*Activity*), mas possuem distribuições estatísticas próprias para o tempo de operação.

Considere que o nosso porto passou por algumas reformas e que será capaz de atender grandes embarcações, do tipo Capesize. Essas embarcações, por serem maiores do que as que o porto recebe usualmente, têm um tempo de operação também maior. Assim, o Centro de Controle Portuário informa que o tempo médio de operação das embarcações Capesize é de 48 horas, segundo uma distribuição Erlang de ordem 2 e que o tempo entre chegadas sucessivas destas embarcações é de 96 horas, segundo uma distribuição exponencial. A ideia é fazer com que a distribuição do tempo de operação seja obtida diretamente pelo *label* da entidade navio.

Inicialmente, o modelo deve ser modificado para permitir a chegada do navio Capesize. Para isso, basta acrescentar um novo *Start Point* com intervalo entre chegadas de 96 horas, segundo uma distribuição exponencial (a Figura 3.29 apresenta o modelo modificado).

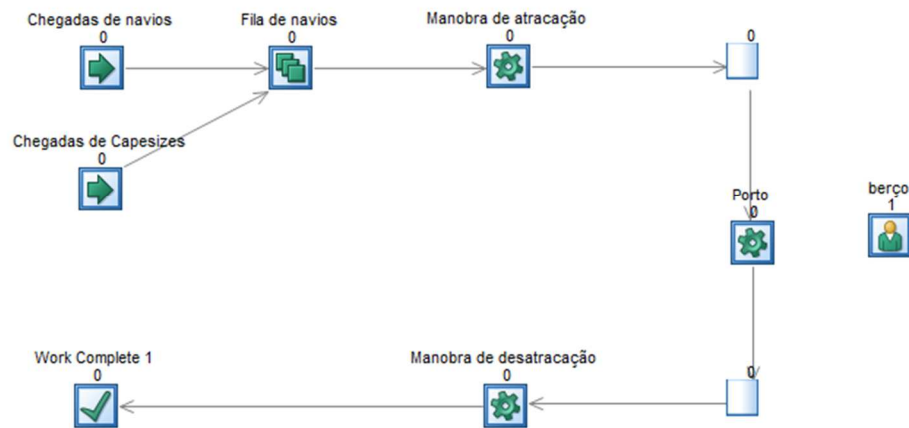


Figura 3.29 – Novo modelo, agora com um segundo *Start Point* para o surgimento das embarcações Capesize.

O que vamos fazer agora é criar duas distribuições distintas: uma para os navios Capesize e outra para as embarcações menores. A distribuição do tempo de atracação das embarcações menores — aquelas que surgem do primeiro *Start Point* — já foi criada anteriormente: trata-se da antiga “Tempo de operação”. Para as embarcações maiores, vamos criar uma nova distribuição chamada “Tempo de operação Capesize”, como representado na Figura 3.30.

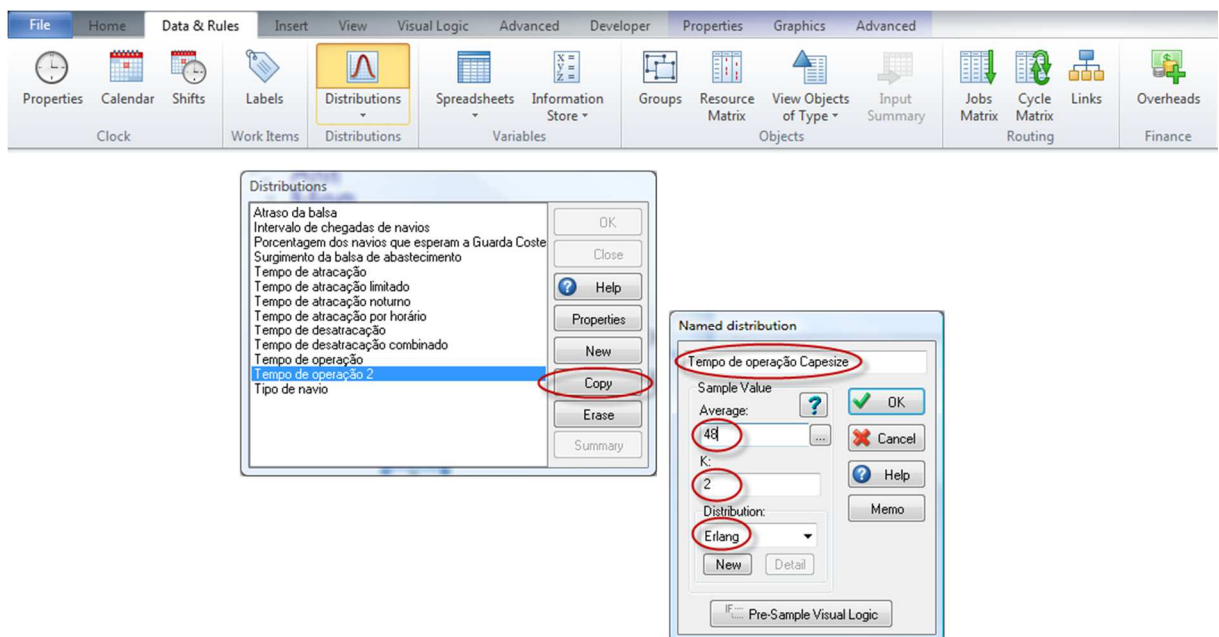


Figura 3.30 - Criação da nova distribuição “Tempo de operação Capesize”. Note, que apenas como ilustração de outra facilidade do SIMUL8, a distribuição foi criada a partir da cópia da antiga distribuição “Tempo de operação”.

Na seção 3.4.2 já havíamos criado o *label* “Tipo de navio” para caracterizar dois tipos distintos de entidade. Agora, vamos criar um *label* que irá armazenar o *nome* da distribuição do tempo de operação da embarcação. Assim, vamos criar uma nova distribuição “Tempo de operação label” a partir da opção *label Based* da caixa de diálogo *New Distribution* (Figura 3.31).

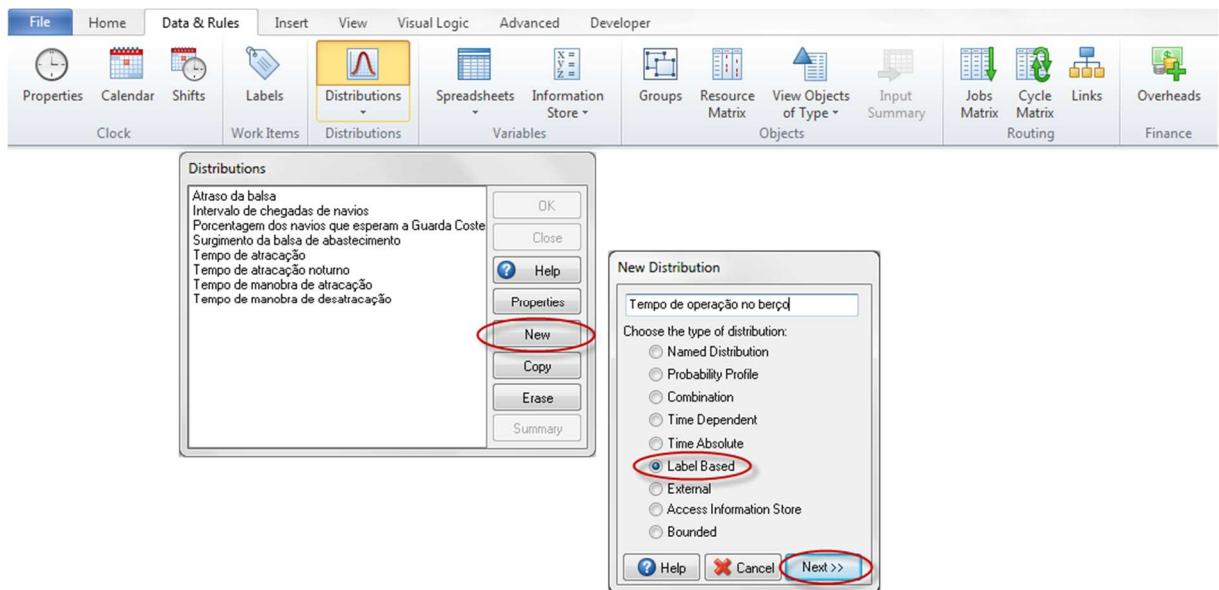


Figura 3.31 - Criando a nova distribuição para o tempo de operação no berço a partir da opção *Label Based*.

Ao se clicar no botão *Next>>* da Figura 3.31, o SIMUL8 solicita qual *label* contém o valor (um número) da distribuição ou o nome (um texto) da distribuição. Neste caso, queremos utilizar os nomes de duas distribuições — “Tempo de operação” e “Tempo de operação Capesize”. Precisamos, portanto, de um novo *label* que armazene estes nomes.

Assim, na janela *Label Based Distribution* (Figura 3.32), selecionamos o botão *Create New Label* para criarmos o *label* “Tempo de operação label”¹⁴, no formato *Text*.

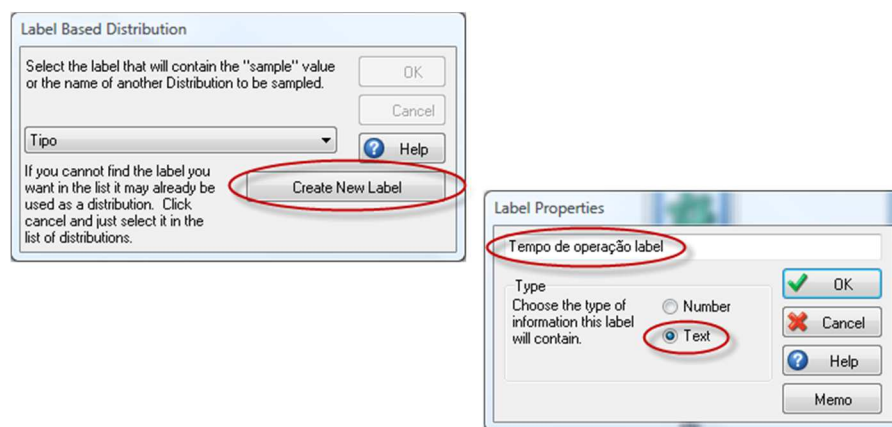


Figura 3.32 - A distribuição *Label Based* exige um *label* específico que armazena ou o valor numérico ou o nome da distribuição já criada. Assim, criamos um novo *label* no formato texto para armazenar o nome da distribuição do tempo de operação.

O último passo é atribuir ao *label* criado o nome da distribuição correta em cada um dos *Start Points*. A Figura 3.32 representa a atribuição do nome “Tempo de operação” para o *label* “Tempo de operação label” das entidades que representam as embarcações Panamax e Suezmax. Um procedimento semelhante deve ser feito no *Start Point* das embarcações Capesize, mas atribuindo o nome “Tempo de operação Capesize”.

¹⁴ O fato de o nome do *label* ser igual ao da distribuição não tem razões ocultas ou telúricas. Para qualquer nome que você escolha para o *label*, o SIMUL8 irá adotar como novo nome da distribuição, eliminando a anterior.

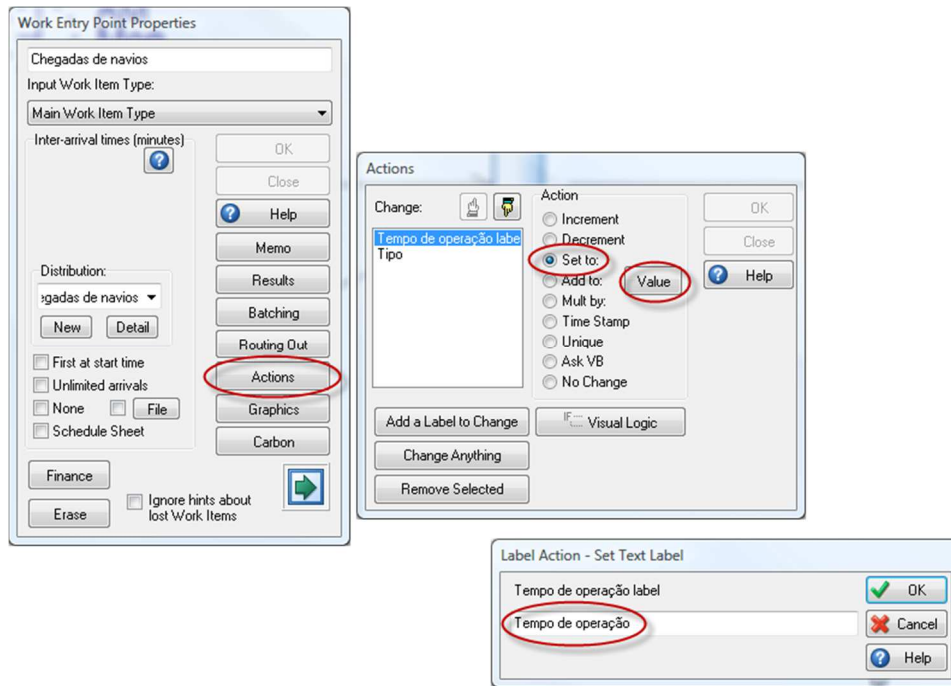


Figura 3.33 - No Start Point “Chegadas de navios”, o Label “L_distribuição do tempo de operação” recebe o valor “Tempo de operação”, o nome da distribuição respectiva.

A última etapa na criação deste exemplo é informar ao *Activity* Porto que a distribuição do tempo de operação agora é “Tempo de operação label”, como representado na Figura 3.33.

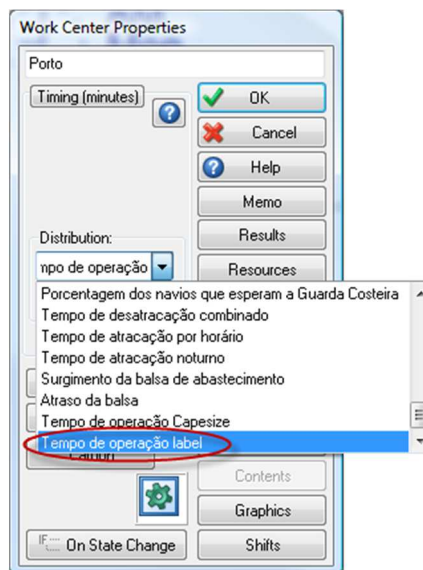


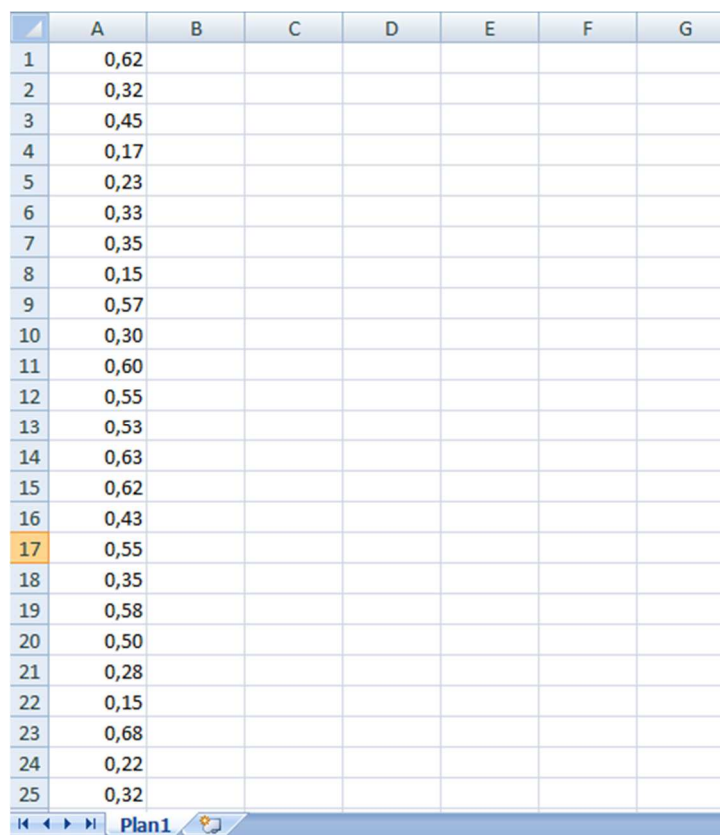
Figura 3.34 - A distribuição *Label Based* “Tempo de operação label” é atribuída ao *Activity* Porto. Assim, toda vez que uma entidade entrar no Porto, o label fornecerá qual a distribuição a ser utilizada.

Outra possibilidade é criarmos um *label* numérico para o tempo de operação e atribuir a respectiva distribuição a este *label* no *Actions* do *start point*, por exemplo. Em seguida, no *Activity* onde este valor será utilizado, deve-se configurar uma distribuição “fixa” cujo valor numérico será obtido do *label* indicado no campo “Fixed Value”.

3.4.8. *External*: acessando distribuições externas ao SIMUL8 (Excel ou Visual Basic)

Em muitas situações, desejamos simular um sistema a partir de um conjunto de dados reais, evitando a utilização de distribuições teóricas. Isso é praxe, por exemplo, quando queremos *validar* o nosso modelo, ou seja, mostrar que o modelo apresenta saídas compatíveis com os valores do sistema real, quando submetido às mesmas condições de entrada. As distribuições criadas com a opção *External* permitem que o SIMUL8 leia valores diretamente de uma planilha eletrônica em formato Excel ou definidas via Visual Basic¹⁵.

Considere, por exemplo, que a gerência do nosso porto tem dúvidas quanto ao modelo, pois existe um tempo de pré-operação não considerado: toda vez que uma embarcação atraca, ela aguarda um tempo para liberação alfandegária, decorrente do processo burocrático de apresentação de documentos da embarcação e declaração de carga. Assim, a gerência forneceu uma planilha eletrônica com os valores dos tempos, em horas, da pré-operação de 100 embarcações. Os dados estão na pasta “Plan1” da planilha mostrada na Figura 3.35, dispostos da célula A1 para baixo (A2, A3...).



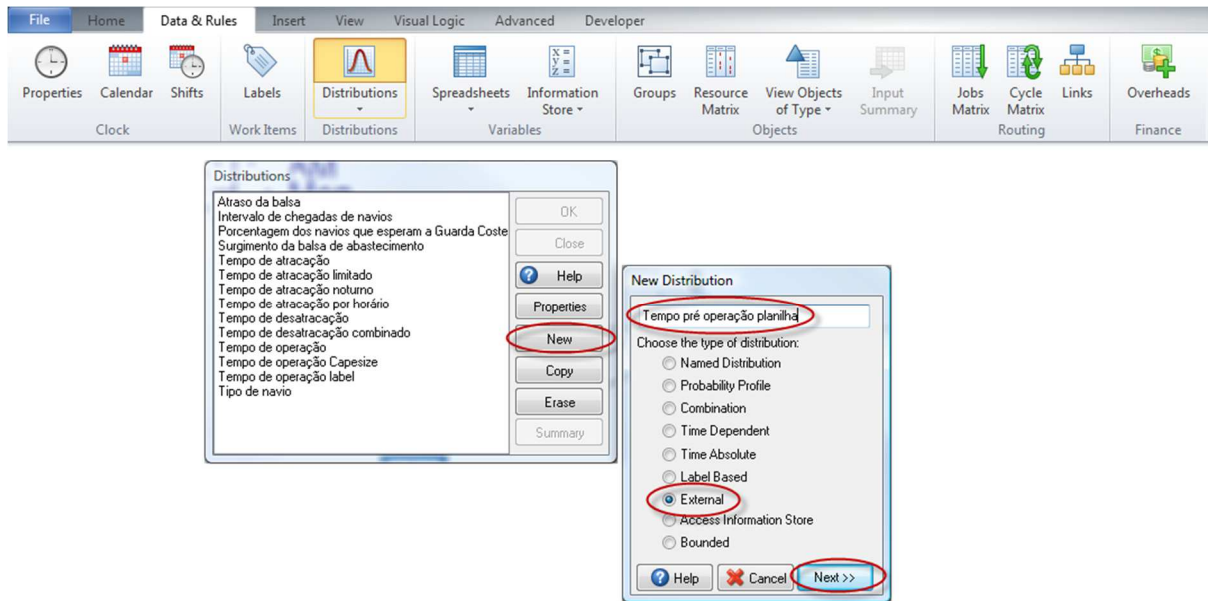
	A	B	C	D	E	F	G
1	0,62						
2	0,32						
3	0,45						
4	0,17						
5	0,23						
6	0,33						
7	0,35						
8	0,15						
9	0,57						
10	0,30						
11	0,60						
12	0,55						
13	0,53						
14	0,63						
15	0,62						
16	0,43						
17	0,55						
18	0,35						
19	0,58						
20	0,50						
21	0,28						
22	0,15						
23	0,68						
24	0,22						
25	0,32						

Figura 3.35 - Planilha com dados dos tempos de pré-operação para 100 embarcações.

Você não precisa se preocupar quanto ao separador decimal usado no Excel, pois o SIMUL8 é capaz de reconhecer tanto o ponto quanto a vírgula.

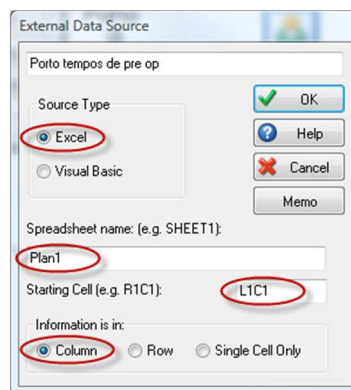
Inicialmente, vamos criar a distribuição “Tempo pré operação planilha”, do tipo *External*, como mostra a Figura 3.36.

¹⁵ Nesta seção, discutiremos apenas o acesso a planilhas de dados.

Figura 3.36 - Opção *External*.

Após clicarmos no botão *Next >>*, o SIMUL8 abrirá a janela *External Data Source* (Figura 3.37) em que são fornecidos os dados específicos da planilha. No caso, a planilha que contém os dados é “Porto tempos de pre op.xlsx” e os dados “começam” pela célula da Linha 1 e Coluna 1. Por isso, o campo *Starting Cell* está com o valor L1C1. Nesta janela, deve ser informado adicionalmente se os dados estão em uma coluna (como no nosso exemplo) ou em uma linha, o nome da Pasta **como consta no Excel** (“Plan1”) e a origem da fonte de dados (“Excel”).

O SIMUL8 vai utilizar todos os dados a partir da célula indicada até que encontre uma célula na coluna que **não contenha um número** (ou seja, que esteja vazia ou que contenha um texto).

Figura 3.37 - Janela *External Data Source*.

Se o seu Excel é o brasileiro, ou seja, a língua é o português, a referência é Linha-Coluna e não Row-Column. Por isso, a *Starting Cell* deve ser L1C1 (ou L-número da linha-C-número da coluna). Já o Excel em inglês só aceita a notação R1C1 (ou R-número da “row”-C-número da “column”).

Para que os navios aguardem o tempo de pré-operação no modelo, devemos acrescentar um *Activity* para a pré-operação (utilizando a distribuição recém-criada) entre o *Activity* da manobra de atracação e o do porto, como representado na Figura 3.38.

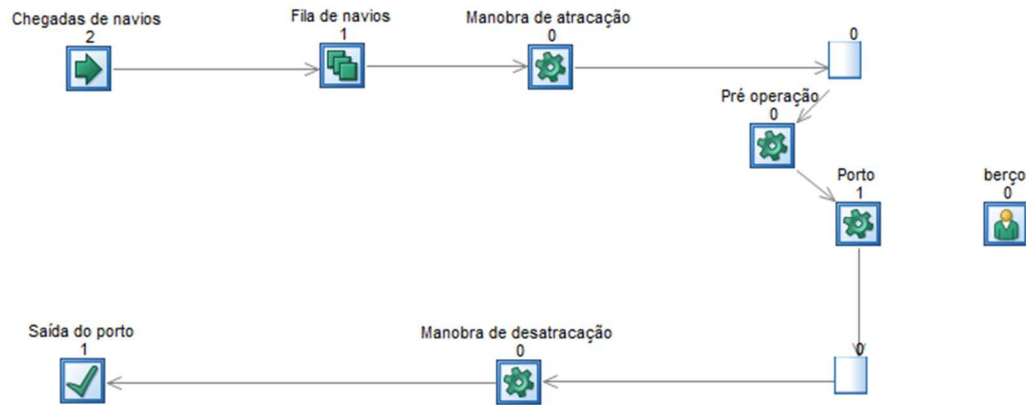


Figura 3.38 - O Activity Pré Operação utiliza a distribuição recém criada “Porto tempos de pre op”, lendo diretamente do Excel o valor do tempo de pré-operação para cada entidade (navio) que visita o Activity.

Esta é das opções mais úteis para quem trabalha com sistemas em que a validação é crítica ou com sistemas em que os fenômenos aleatórios mudam com muita rapidez – como nos modelos utilizados para a simulação de planejamento da produção. Por exemplo, uma fábrica define a cada manhã o que vai produzir dentre uma dezena de opções disponíveis. Neste caso, o melhor a fazer é simular diretamente as corridas de produção, usando dados reais.

Importante: para a execução do modelo, a planilha deve estar obrigatoriamente aberta, não importando em qual pasta de trabalho do seu computador ela esteja salva.

3.5. Algumas ferramentas adicionais para o usuário avançado

3.5.1. Simulação sem aleatoriedade

Pode não fazer sentido à primeira vista, mas imagine que você deseja testar seu modelo em busca de erros. Esse trabalho, que usualmente chamamos de **verificação**, se torna (bem) mais fácil se você não utilizar números aleatórios, mas sim, valores fixos conhecidos.

Para fazer isso no SIMUL8, basta considerar todas as distribuições como sendo do tipo *Fixed*. Se for possível, você pode até realizar alguns cálculos manuais para verificar se os dados de saída do modelo são coerentes¹⁶.

3.5.2. Manipulando o gerador de números aleatórios do SIMUL8

Um observador atento já deve ter percebido que toda vez que refaz uma simulação no SIMUL8, ele parece repetir a sequência de eventos, o que pode parecer estranho, pois, afinal, **não era para ser aleatório?**

O SIMUL8 repete sempre a mesma *semente* do gerador de números aleatórios de maneira proposital, para facilitar a verificação do modelo – e a experiência mostrará a você que é bom que seja assim. Afinal, é mais fácil corrigir e desenvolver um modelo quando você pode pausar antes de determinado erro que você já sabe, pela simulação anterior, o momento que ocorre.

¹⁶ Para um maior detalhamento sobre esta técnica, consulte a seção 5.3 (*Valores constantes ou simplificados vs. cálculos manuais*) do livro “Modelagem e Simulação de Eventos Discretos” de Leonardo Chwif e Afonso C. Medina.

Quando o modelo está verificado e livre de erros, você pode informar ao SIMUL8 que deseja novas sementes de números aleatórios a todo o momento. Isto pode ser feito facilmente utilizando-se o comando *Random Sampling* do menu *Advanced*, como mostrado na Figura 3.39. Na janela *Random Sampling Parameters*, é possível garantir o uso (ou não) de números aleatórios distintos a cada rodada pela opção *Auto change random number on EVERY restart*. Outra opção (pouco usada na prática) consiste em alterar manualmente a semente de números aleatórios pela opção *Random Stream Set Number*.

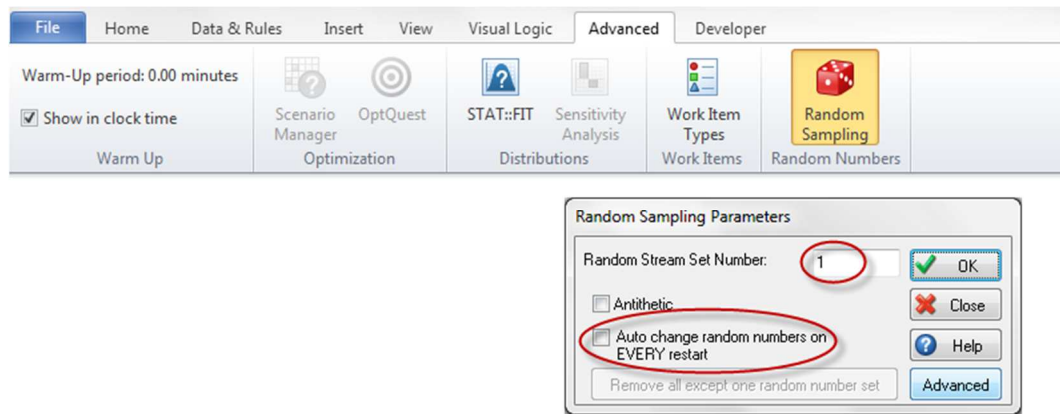


Figura 3.39 - Com a opção *Auto Change random number*, a cada rodada do simulador, novos números aleatórios são gerados. A opção *Random Stream Set Number* serve para se substituir a semente de geração de números aleatórios

3.5.3. Trabalhando com arquivos de chegada do tipo *trace*

No item 3.4.8 descrevemos como o SIMUL8 pode utilizar informações de um arquivo de dados externo. Para que o SIMUL8 gere entidades na chegada de acordo com os instantes de tempo definidos em uma fonte dados interna, basta acessar no objeto (*Start Point*) a opção *Schedule Sheet* (Figura 3.40).

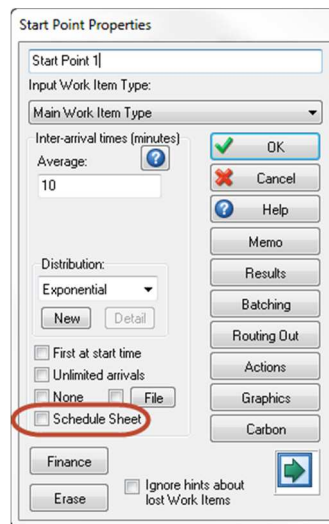


Figura 3.40 – Opção *Schedule Sheet*.

Uma vez feito isto, o SIMUL8 pedirá que se defina uma variável do tipo planilha (*Spreadsheet*) onde são especificados o horário de chegada (tanto em formato de hora como em formato de tempo absoluto do relógio de simulação) e a quantidade de entidades que devem chegar neste horário. A Figura 3.41 ilustra um exemplo que o próprio SIMUL8 define quando se cria uma nova variável nova do tipo *Spreadsheet*. No exemplo, uma entidade chega às 9:15, outra às 12:00, às 15:00 chegam 2 entidades e, quando o relógio de simulação atingir 4.200 unidades de tempo, chegará mais uma entidade, e assim sucessivamente. Caso o leitor queira explorar melhor esta funcionalidade recomendamos que veja o Capítulo 5 (Noções de Variáveis e *Visual Logic*).

Sheet: Var1

	A	B	C	D	E	F	G	H	I	J	K
1	Example Title for a schedule										
2	As many non-numeric rows at top as you need										
3	Time	Quantity	Comments								
4	9:15	1	One item arrives at quarter past nine								
5	12:00	1	Then another at noon								
6	15:00	2	Then two at 3pm								
7	420	1	You don't need to use time format - this is 16:00 (4pm) assuming the clock started at 9am								
8	1000	1	Schedule can extend to any simulation time - this arrival is on Wednesday								
9											
10											
11											

Figura 3.41 - Exemplo de *Schedule Sheet*.

Outra opção, disponível a partir da versão 2015 do SIMUL8, é por meio da funcionalidade *Day Planner*. Neste caso, pode-se definir quantas entidades devem chegar em cada dia e em cada faixa de. A Figura 3.42 ilustra esta funcionalidade.

Start Point Properties

Start Point 1

Input Work Item Type:

Main Work Item Type

Inter-arrival times (minutes)

Average: 10

Distribution: Exponential

New Detail

☐ First at start time

☐ Unlimited arrivals

☐ None ☐ File

☐ Schedule Sheet

☐ From W1 State Chart

☒ Day Planner

Finance

Erase

Ignore hints about lost Work Items

Start Point 1 - Day Planner

	Day 1	Day 2	Day 3	Day 4	Day 5
09:00	5	5	5	5	5
09:15	10	10	10	10	10
09:30	5	5	5	5	5
09:45	10	10	10	10	10
10:00	5	5	5	5	5
10:15	10	10	10	10	10
10:30	5	5	5	5	5
10:45	10	10	10	10	10
11:00	5	5	5	5	5
11:15	10	10	10	10	10

Interval: 15

Distribution: Other (defined in the Start Point)

☐ Equally spaced within interval

Days Scheduled

Add Day Copy Day

Delete Day Clear Day

☒ Repeat Schedule

Garante Chegadas Equiespaçadas

Define Tamanho do Intervalo (default 15 em 15 min)

Define tipo de Distribuição

Adiciona/ Deleta / Copia ou Limpa Dia

Figura 3.2 - Exemplo de uso da funcionalidade *Day Planner*

Capítulo 4 DIRECIONAMENTO DE ENTIDADES

4.1. Introdução

Na prática, dificilmente um processo é totalmente sequencial. Isto quer dizer que, em algumas (ou várias) partes dele, as entidades podem seguir por diferentes rotas ou vir de diferentes rotas. É este tipo de operação que abordaremos neste capítulo. Os direcionamentos no SIMUL8 são controlados nos pontos de entrada (*Start Points*) ou nos *Activities* (nunca em filas) e são ajustados nas propriedades *Routing In* e *Routing Out* destes objetos.

4.2. Direcionamentos de entrada básicos: *Priority*, *Collect* e *Passive*

Para ilustrar os diferentes tipos de direcionamentos de entradas, construa o modelo ilustrado na Figura 4.1, mantendo os valores padrão do SIMUL8 (não se esqueça de zerar os tempos de deslocamento).

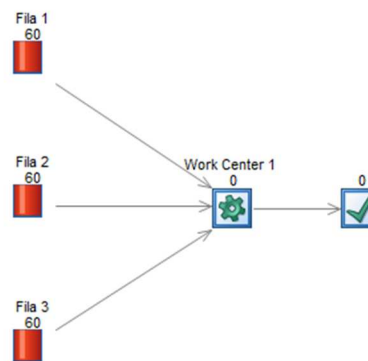


Figura 4.1 - Modelo Inicial.

Para as filas 1, 2 e 3, configure um valor inicial de 60 entidades, de modo que, no início da simulação, existam 60 entidades em cada fila. Para fazer isso, abra a janela de propriedades de cada fila, clique no botão *Start-Up* e digite o valor 60 na caixa de diálogo *Initial Conditions*.

Você pode inserir o mesmo valor inicial nas 3 filas de uma vez: selecione todas as filas com o *mouse*, dê um duplo-clique em uma delas e altere o conteúdo inicial para 60. Quando clicar no botão *OK*, o SIMUL8 perguntará se você deseja aplicar as alterações em todos os objetos selecionados - ao responder sim, você acabou de ajustar os parâmetros das três filas ao mesmo tempo! Este tipo de operação múltipla funciona com todos os objetos do SIMUL8.

Simule o modelo em uma velocidade baixa para poder ver o que está acontecendo. Observe que o *Activity* dá preferência a uma das filas e, somente quando acabam as entidades desta fila, ele irá retirar entidades de outra fila. Isto acontece porque o direcionamento de entrada padrão do SIMUL8 é do tipo *Priority*, em que é seguida uma lista de prioridades. Se você construiu seu modelo na ordem (Fila 1, Fila 2 e Fila 3) e ligou as filas ao *Activity* nesta mesma ordem (1, 2 e 3), o SIMUL8 considerará a fila 1 como a mais prioritária, seguida pelas filas 2 e 3. Para verificar esta lista de prioridades, acesse as propriedades do *Activity* e clique no botão *Routing In* (Figura 4.2). Você pode mudar a prioridade de uma fila clicando em seu nome e utilizando o botão em formato de “mãozinha” para aumentar ou diminuir a prioridade. Faça uma experiência: considere agora que a fila 3 é mais prioritária que a fila 2, que é mais prioritária que a fila 1. Rode o modelo novamente e veja o efeito desta alteração.

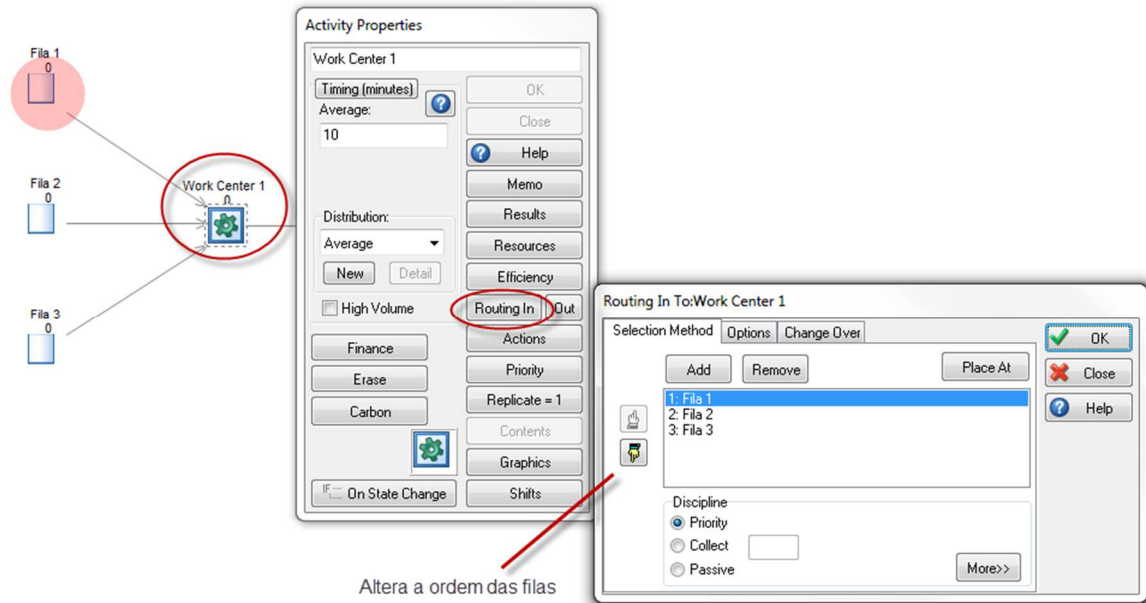


Figura 4.2 - Prioridades para o Routing In.

Observe que, ao final da simulação, todas as 180 entidades presentes inicialmente saíram do sistema (esta informação será importante nas próximas discussões).

Utilizando este mesmo modelo com 3 filas, vamos discutir o próximo tipo de direcionamento de entrada, chamado *Collect*. Nas opções de roteamento do *Activity*, clique em *Collect* e observe que aparecerão outras opções, como ilustra a Figura 4.3 (não se esqueça de clicar no botão *More>>*).

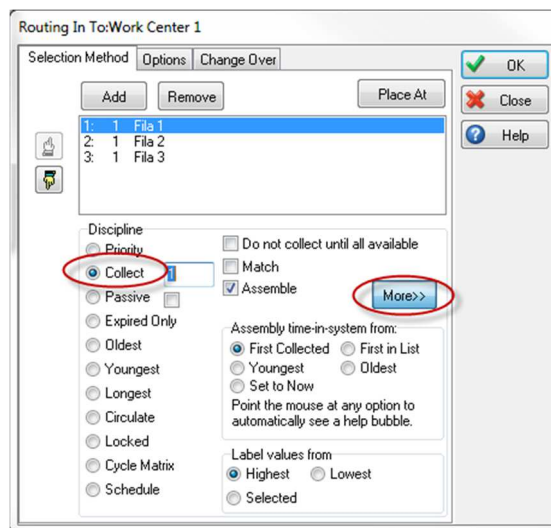


Figura 4.3 - Utilização da disciplina *Collect*.

Do que se trata a disciplina *Collect*? Imagine que o seu processo vai precisar de 2 itens que estão na fila 1, 1 item da fila 2 e 1 item da fila 3 para realizar uma *montagem* que vai unir estes itens, formando um único item na saída. Para isso, indique no campo *Collect*, o “recolhimento” de 2 entidades na fila 1, 1 entidade na fila 2, 1 entidade na fila 3, mantendo a opção *Assemble* ativada, e rode o modelo. O que ocorreu?

Primeiramente, observe que apenas 30 conjuntos foram montados. Isto aconteceu porque eram necessários 2 itens da fila 1 para formar cada conjunto e havia 60 itens nesta fila. E porque ficaram 29 itens nas filas 2 e 3? Porque quando

acabaram os itens da fila 1 o *Activity* retirou 1 item da fila 2, 1 item da fila 3, e ficou aguardando para retirar 2 itens da fila 1. Como estes itens “não chegaram”, a simulação terminou neste estado.

Agora, experimente utilizar a opção *Do not collect until all available* presente na segunda coluna da opção *Collect* (Figura 4.3). Ao executar a simulação, você perceberá que sobrarão 30 itens na fila 2 e 30 itens na fila 3, porque o *Activity* somente irá iniciar a coleta quando todos os itens necessários estiverem disponíveis.

Por fim, desative a opção *Assemble* (Figura 4.4). Antes de simular, tente calcular quantos itens sairão do modelo. Rode o modelo e confira se acertou (e se entendeu o que aconteceu).

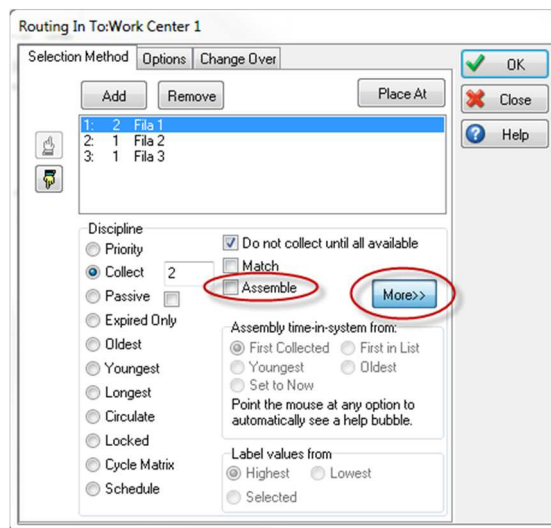


Figura 4.4 - Opção *Assemble* dentro da disciplina *Collect*.

A resposta é: neste caso irão sair 120 unidades, pois o *Activity* coleta 2 entidades da fila 1, 1 entidade da fila 2 e 1 entidade da fila 3. Como há somente 60 itens na fila 1, ele irá coletar estas 60 entidades, além de 30 entidades da fila 2 e 30 entidades da fila 3, totalizando 120 (60+30+30) unidades. Em termos temporais, o mecanismo de coleta do *Activity* é análogo ao do caso anterior, só que ele não irá “unir” os itens em conjuntos, apenas processará as entidades “ao mesmo tempo” e as liberará simultaneamente.

Vamos discutir agora outro tipo de direcionamento, chamado *Passive* (Figura 4.5). Troque a disciplina de prioridades do *Activity* para *Passive* e rode o modelo novamente. O que aconteceu?

Absolutamente nada!

As filas são objetos de simulação *passivos*, ou seja, elas não “empurram” entidades para outros objetos de simulação. Quanto você as conecta na entrada de outro objeto passivo — que não “puxa” entidades para si — nada acontece. O modo passivo é útil para situações em que se conectam dois *Activities* em série, sem que exista uma fila entre eles. Se no segundo *Activity* não for atribuída a disciplina *Passive*, ele poderá “roubar” as entidades do primeiro *Activity*. Se o segundo *Activity* estiver com a disciplina *Passive* ativada, ele processará apenas as entidades que forem despachadas pelo processo antecessor. No entanto, para evitar o uso destes artifícios, recomendamos sempre intercalar elementos ativos com passivos no modelo, ou seja, *Activities* com *Queues* (filas), mesmo sabendo que eventualmente não haverá tempo de espera na fila intermediária.

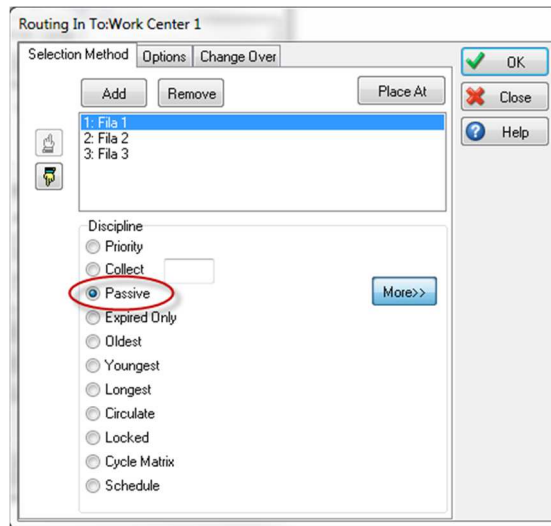


Figura 4.5 - Disciplina *Passive*.

4.3. Outras disciplinas de entrada (*Expired Only*, *Oldest*, *Youngest*, *Longest* e *Circulate*)

A lista de disciplinas de entrada apresentadas na janela *Routing In* (Figura 4.5) possui ainda as seguintes opções, na ordem da janela de comando: *Expired Only*, *Oldest*, *Youngest*, *Longest* e *Circulate*. Vamos discutir a atuação de cada uma delas.

A disciplina *Expired Only* é utilizada em situações como:

- Produtos que devem ser retirados do estoque, pois ultrapassaram a data de validade;
- Chamadas telefônicas que desistiram de esperar em uma central de atendimento após esperar muito tempo;
- Pessoas que desistiram de esperar em uma fila após esperar por muito tempo.

Basicamente, um *Activity* com a disciplina *Expired Only*, ligado a uma fila, verifica se alguma entidade da fila já ultrapassou um tempo máximo de espera (*Shelf Life*), definido nas propriedades da fila.

Você pode simular a disciplina *Expired Only* no modelo da Figura 4.1. Inicialmente, clique na fila 1 e atribua um valor para o seu *Shelf Life*, por exemplo, 5 minutos, como mostra a Figura 4.6.

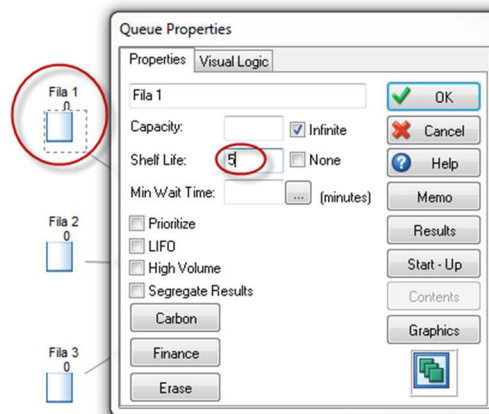


Figura 4.6 - Opção *Shelf Life* para o “tempo de prateleira” ou tempo máximo de espera em fila.

Para que as entidades que ultrapassem 5 minutos de espera sejam retiradas da fila, deve existir um *Activity* específico para isso. Assim, ao modelo original acrescente mais uma atividade que se conecta à fila 1 e à saída, como representado na Figura 4.7.

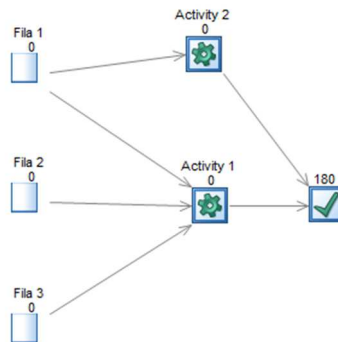


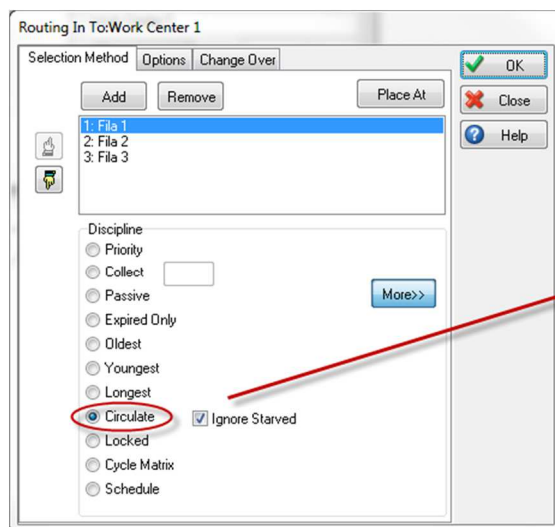
Figura 4.7 – Modelo com um segundo *Activity* para retirar as entidades que ultrapassem seu *Shelf Live*.

Agora, altere a disciplina de entrada do *Activity 2* para *Expired Only* e simule o modelo (você pode deixar o *Activity 1* com a disciplina *Collect*). Note que a atividade 1 opera normalmente, mas algumas entidades serão retiradas da fila 1 pelo *Activity 2*.

As disciplinas *Oldest*, *Youngest* e *Longest*, são fáceis de entender:

- *Oldest*: o *Activity* selecionará da fila a entidade que possuir o maior tempo de espera em fila;
- *Youngest*: o *Activity* selecionará da fila a entidade que possuir o menor tempo de espera em fila;
- *Longest*: o *Activity* selecionará entidade(s) da fila que possuir o maior número de elementos.

A disciplina *Circulate* faz com que o *Activity* selecione uma entidade por vez de cada fila, segundo a ordem estabelecida, como um carteador que distribui as cartas do baralho para os jogadores da mesa, uma carta por vez e sempre no mesmo sentido. A disciplina *Circulate* possui uma opção adicional, selecionada por padrão: *Ignore Starved* (Figura 4.8). Esta opção garante que, por exemplo, se a fila 1 não possuir mais entidades, o processo de coleta continuará nas outras filas. Caso a opção *Ignore Starved* seja desabilitada, então o *Activity* interromperá a coleta de entidades na fila que está vazia.



Se ativada, esta opção ignora a falta de entidades nas filas.

Desativada, esta opção interrompe a operação e aguarda alguma entidade disponível na fila.

Figura 4.8 - Disciplina *Circulate*.

Para simular a disciplina *Circulate* no modelo da Figura 4.1, antes de apertar o botão *Run*, altere o conteúdo inicial da fila 1 para 10.

Veja que, na situação anterior, quando acabaram as entidades da fila 1, a retirada de entidades continuou entre as filas 2 e 3. Contudo, quando a opção *Ignore Starved* foi desativada, o *Activity* leva a disciplina *Circulate* à risca, ou seja, se ele tentar retirar uma entidade da fila 1 e não conseguir, ficará esperando até poder fazer a coleta na fila 1. Assim, nenhuma entidade será coletada e sobrarão 50 entidades nas filas 2 e 3.

4.4. Disciplinas de saída: *Circulate*, *Uniform*, *Percent*, *Priority*, *Label*, *Shortest Queue*, *Passive* e *Batching*

Vamos utilizar um modelo que acompanha o SIMUL8 para exemplificar os direcionamentos de saída. Abra o modelo *routing out options.s8*, da pasta *SIMUL8\Examples\Work Center Routing Out\Routing Out Options.S8*. Este modelo é mostrado na Figura 4.9.

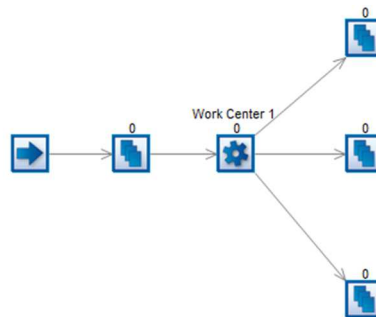


Figura 4.9 - Modelo *Routing Out Options.s8*

Acesse as propriedades do direcionamento de saída do *Activity*, clicando no botão *Routing Out* (note que o modelo usa a opção *Circulate*). De forma análoga ao que vimos anteriormente, esta disciplina de saída envia entidades alternadamente entre os destinos (no caso, filas) de saída. Rode o modelo e verifique seu funcionamento.

Agora, limite a capacidade da fila “Storage Area 2” para 20 entidades (nas propriedades da fila, altere a opção *Capacity*). Rode novamente o modelo e observe o que ocorreu. A disciplina *Circulate* irá alimentar as três filas até que a capacidade da fila “Storage Area 2” seja atingida; depois disso, a disciplina *Circulate* será obedecida apenas entre as filas “Storage Area 3” e “Storage Area 4”.

Acesse novamente a disciplina de saída do *Activity*, desative a opção (padrão) *Ignore Blocked Routes* e rode o modelo novamente. Perceba que, neste caso, quando a fila “Storage Area 2” atinge sua capacidade máxima é criada uma rota bloqueada para o *Activity*. Como a opção *Ignore Blocked Routes* está desativada, o *Activity* não ignorará a rota e tentará, a todo custo, enviar itens para esta fila. Como isto não será possível, o *Activity* não fará mais nenhum processamento e a fila anterior a ele passará a crescer de forma ilimitada até o final da simulação.

Vamos testar as outras opções. Volte a capacidade da fila “Storage Area 2” para infinito e ative novamente a opção *Ignore Blocked Routes*. Ajuste agora a disciplina de saída para *Uniform*. Neste caso, o *Activity* escolherá aleatoriamente uma fila para enviar entidades processadas. Note que há uma diferença fundamental entre as disciplinas *Circulate* e *Uniform*: enquanto *Circulate* utiliza, obrigatoriamente, uma sequência pré-definida, *Uniform* garantirá apenas que, a longo prazo, cada fila de saída receba a mesma quantidade de entidades. Rode o modelo e observe quantas entidades ficaram em cada fila ao final da simulação. Os valores são muito próximos, certo?

E se quisermos que 20% das entidades se dirijam para a fila “Storage Area 2”, 50% para a fila “Storage Area 3” e 50% para a fila “Storage Area 4”? Basta utilizar a opção *Percent*, definindo o perfil de probabilidades adequado, como mostrado na Figura 4.10.

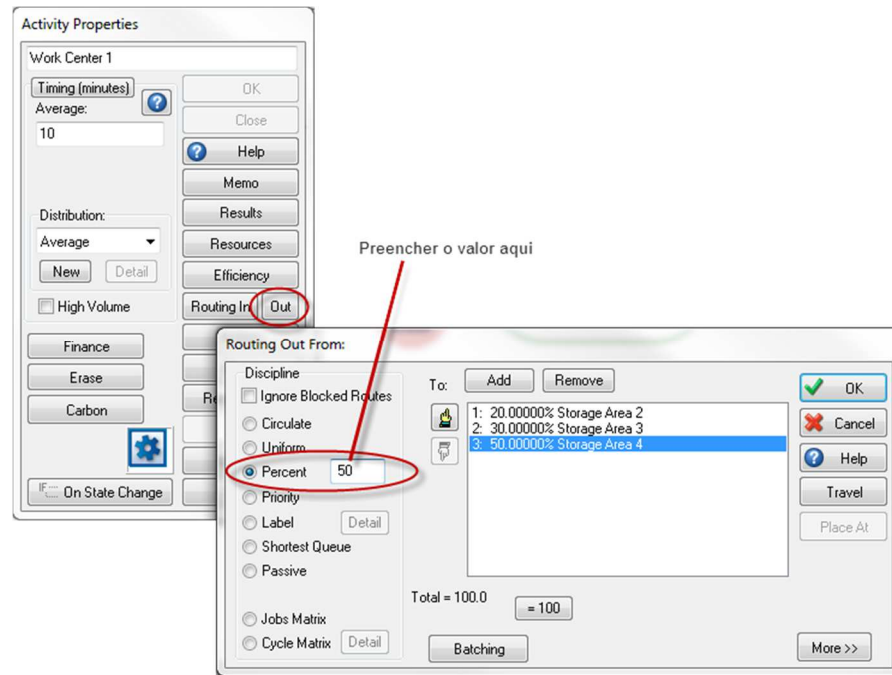


Figura 4.10 - Routing Out Percent.

Note que, neste modelo, a opção *Uniform* é equivalente à opção *Percent*, com o percentual de 33,33% para cada fila.

A próxima opção de direcionamento de saída é *Priority*, nossa conhecida. Neste caso, assim como no direcionamento de entrada, o *Activity* seguirá a prioridade definida na lista: “Storage Area 2”, “Storage Area 3” e “Storage Area 4”. Selecione esta opção no *Activity* e execute o modelo. Veja que somente a fila “Storage Area 2” receberá as entidades, por estar na primeira posição da lista de prioridades. Se esta fila tiver capacidade limitada, receberá entidades até que fique cheia; quando isso ocorrer, o *Activity* irá enviar entidades para o próximo destino da lista (“Storage Area 3”) enquanto for possível, passando para a próxima opção (“Storage Area 4”) quando necessário.

Em seguida, temos as opções *Label* e *Shortest Queue*. Vamos começar pela segunda, que é mais fácil de explicar. Imagine que você está em um supermercado com uma fila para cada um dos vários caixas. Neste caso, uma boa ideia seria escolher a fila com o menor número de pessoas, certo? É exatamente este tipo de operação que esta opção faz.

A opção *Label* é muito utilizada na prática. Para ilustrá-la, vamos utilizar o modelo *Label* “Label Routing.S8”, disponível na pasta *SIMUL8\Examples\Labels*. A Figura 4.11 ilustra esse modelo.

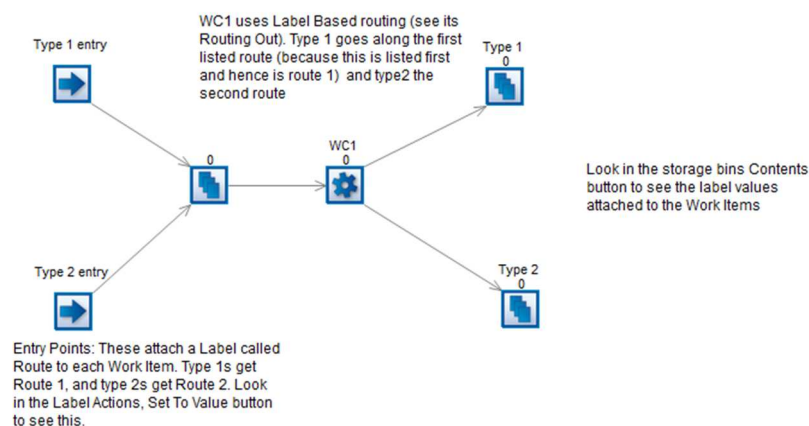


Figura 4.11 - Routing Out by Label.

Observe que há um *label* chamado “Route” que, no ponto de entrada “Type 1 Entry”, ajustado inicialmente com o valor 1, como mostra a Figura 4.12. De modo semelhante, no ponto de entrada “Type 2 Entry”, o *label* “Route” é ajustado para 2.

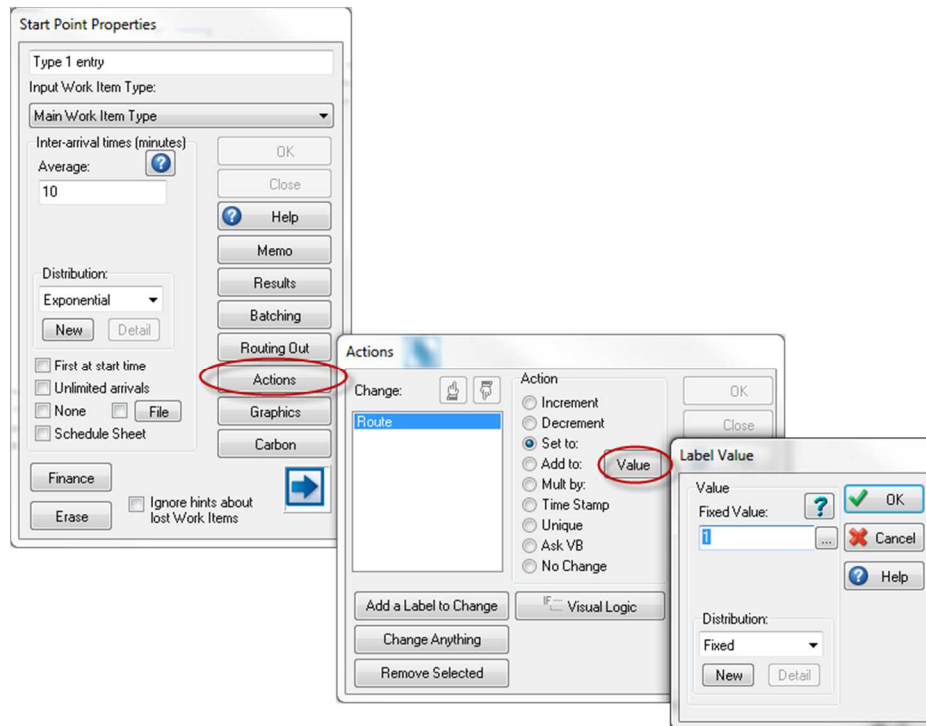


Figura 4.12 - Ajuste de valores do *label* “Route”.

No *Activity* WC1, acesse as propriedades de direcionamento de saída e observe que a opção *Label* está acionada e que o *label* “Route” foi selecionado (Figura 4.13). Desta forma, se o valor do *label* for igual a 1, então o direcionamento de saída será para o primeiro destino da rota (fila “Type 1”); se o valor do *label* for igual a 2, o direcionamento será para a fila “Type 2”. Após rodar a simulação é fácil verificar, clicando na propriedade “Contents” das filas, que cada contém, exclusivamente, entidades cujo *label* “Route” possui valor 1 ou 2.

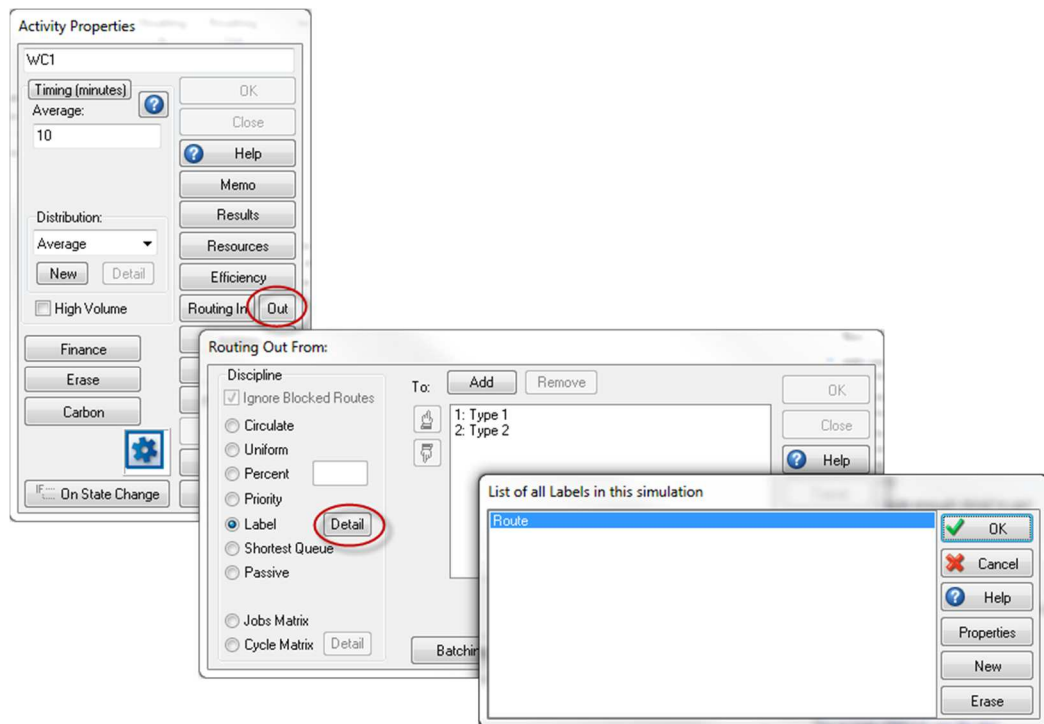


Figura 4.13 - Ajuste do direcionamento pelo *label Route*.

É possível, ainda, ajustar os valores dos *labels* por lógica de programação (como discutiremos no Capítulo 5), o que permite implementar lógicas de direcionamento complexas.

A partir da versão 2013 do SIMUL8 foi introduzida uma nova funcionalidade de roteamento denominada *By Type*. A ideia ainda é a de utilizar um *label* para roteamento, mas neste caso, é possível especificar os valores ou faixas de valores de um *label* para um roteamento específico. Suponha que tenhamos um *label* chamado *L_Rota* que pode assumir valores inteiros de 1 a 10 e dois destinos de saída (por exemplo, filas): saída 1 e saída2. Desta maneira, podemos definir que para os valores de 1 a 6 do *label* *L_rota*, a rota designada será a saída 1 e para os valores de 7 a 10 do *label*, será designada a saída 2. Deixaremos o leitor à vontade para explorar esta funcionalidade.

A opção *Passive* já foi explicada na seção anterior: se for colocada em um direcionamento de saída, então quem irá retirar a entidade do *Activity* será o *Activity* posterior. Lembre-se de que a opção *Passive* só é utilizada (e só faz sentido) quando ligamos diretamente dois *Activities*.

Por fim, vamos ilustrar a opção *Batching*. Você se lembra do roteamento de entrada *Collect* com a opção *Assemble*? Pois bem, a opção *Batching* faz o inverso do *Assemble*, ou seja, após o processamento, uma entidade recolhida pelo *Activity* é transformada em várias entidades na saída (ou seja, a entidade é “copiada”). Vamos usar, mais uma vez, um exemplo do SIMUL8 para ilustrar esta operação. Abra o arquivo *SIMUL8\Examples\Batching\Batching and Collect.s8*, mostrado na Figura 4.14.

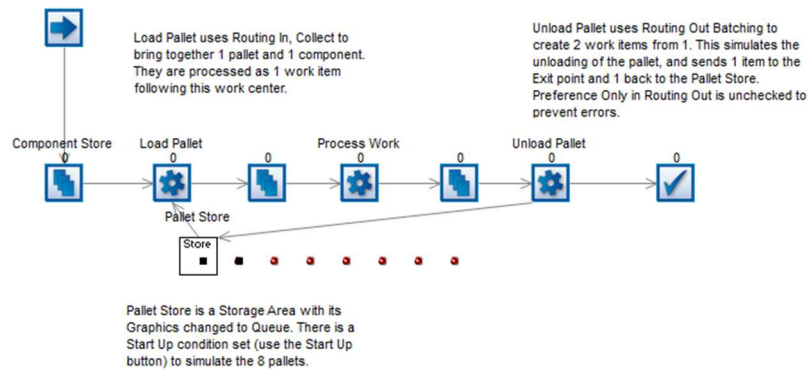


Figura 4.14 - Exemplo de *Batching* e *Collect*.

Esse modelo simula uma linha de *pallets* retornáveis, onde um número fixo de *pallets* fica circulando no sistema. O número total de *pallets* disponíveis é 8 e foi estabelecido na propriedade *Start-Up* da fila *Pallet Store*. Note que o *Activity Load Pallet* recolhe um item e um *pallet*, “unindo” os dois em uma única entidade (opção *Collect* + *Assmeble*). Contudo, após a estação *Unload*, esta entidade é separada em duas entidades (item + *pallet*). Para isso, foi utilizada a propriedade *Batching*, com valor igual a 2 no direcionamento de saída do *Activity Unload Pallet* (Figura 4.15).

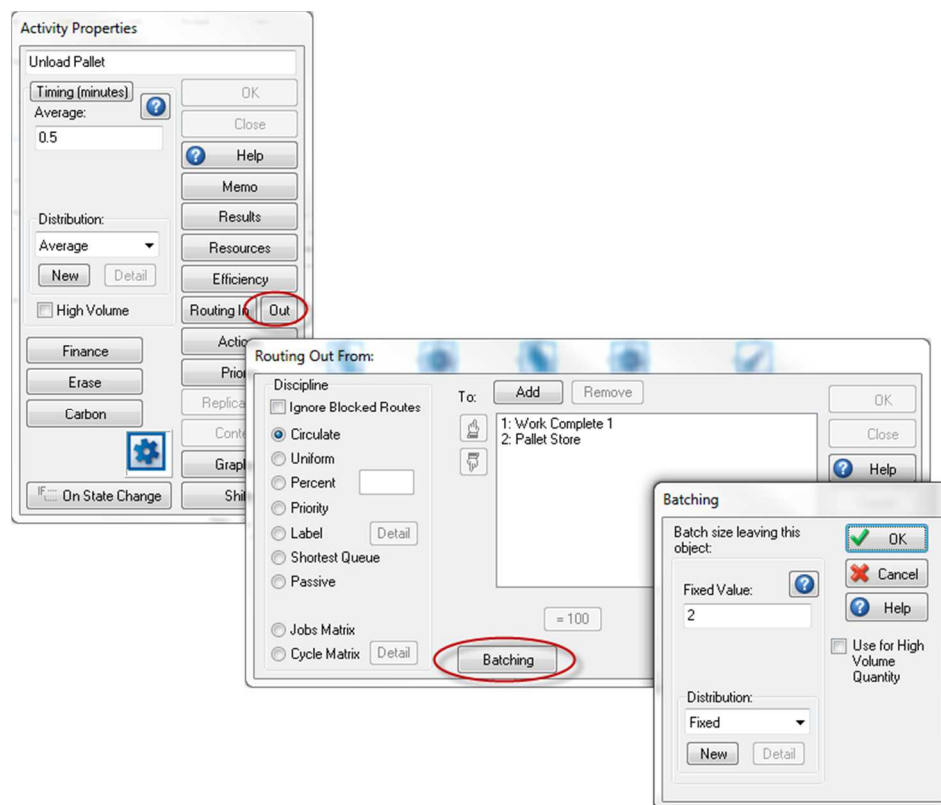


Figura 4.15 - Propriedade *Batching* do Activity.

Mas, como podemos garantir que uma entidade (item) vá para a saída e a outra (*pallet*) volte para a fila de *pallets*? Elementar, meu caro Watson!¹⁷ A disciplina de saída usada é *Circulate*: assim, cada item que sai do *Activity* vai para um dos destinos de saída, alternadamente.

É interessante notar que o valor da propriedade *Batching* também pode ser definido por uma distribuição de probabilidades, por exemplo, para representar fornadas de biscoitos, onde a quantidade varia em função de parâmetros de qualidade. Neste caso, usariamos uma distribuição do tipo *Rounded Uniform*.

Como observação final, note que os pontos de entrada de entidades (*Start Point*) possuem as mesmas opções de direcionamento de saída de um *Activity*. Logo, podemos aplicar todos os conceitos abordados nesta seção para objetos deste tipo.

¹⁷ Nota cultural: A famosa frase atribuída ao detetive ficcional britânico Sherlock Holmes, criado por Sir Arthur Conan Doyle, de fato, não aparece em nenhuma obra escrita por seu criador, mas somente nas peças de teatro e filmes da personagem.

Capítulo 5 VARIÁVEIS E NOÇÕES DE VISUAL LOGIC

5.1. Variáveis

Assim como em qualquer *linguagem de simulação*, o SIMUL8 pode trabalhar com variáveis (por exemplo: Var1, data[5,6] etc.). A principal diferença entre variáveis e *labels* (atributos) é o *escopo* da informação armazenada: enquanto os *labels* guardam atributos de uma única entidade do modelo de simulação, as variáveis possuem natureza global, podendo ser reconhecidas e acessadas durante todo o tempo de simulação. Esta diferença é ilustrada na Figura 5.1, que representa um modelo com 3 variáveis (Contagem, Sinal e Desvio) e 2 atributos (Tipo, Cor). Cada entidade deste modelo possuirá o atributo próprio de cor e tipo, mas os valores de contagem, sinal e desvio valem para todo o modelo.

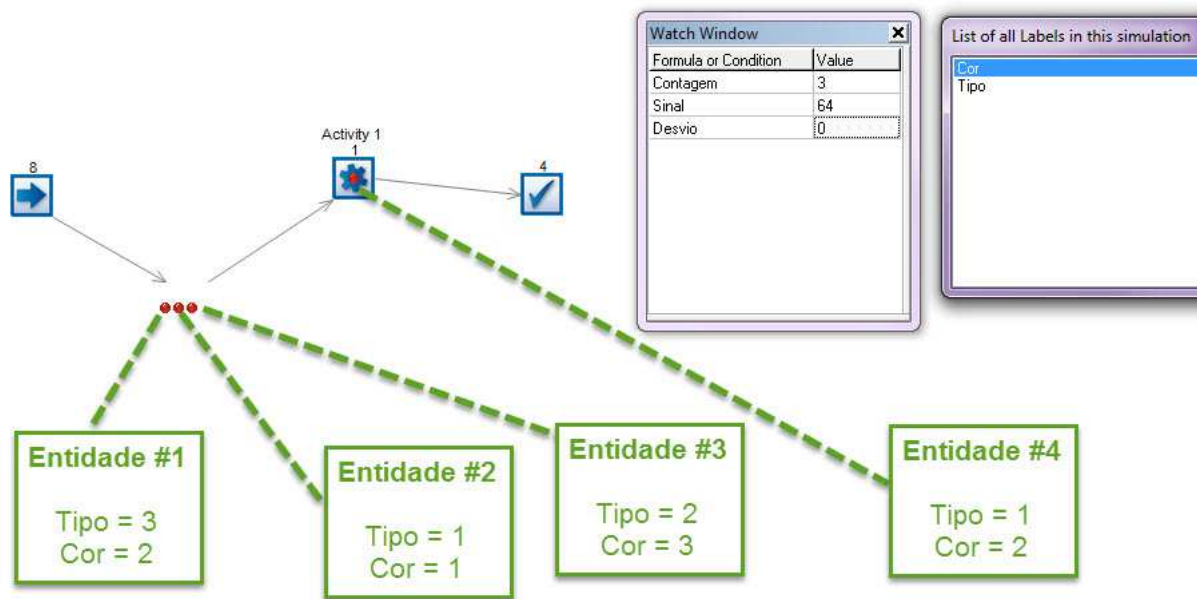


Figura 5.1 – Ilustração da diferença entre variáveis e atributos (*labels*).

Variáveis são de extrema importância em simulação, pois permitem a modelagem de situações complexas, especialmente as que envolvem tomadas de decisão com base em lógica.

5.2. Tipos e criação de variáveis no SIMUL8

Existem dois tipos de variáveis no SIMUL8: as variáveis de sistema e as variáveis criadas pelo usuário. As variáveis de sistema mais conhecidas são:

- *Graph Sync Interval*: variável que define a taxa de amostragem para a construção de gráficos no SIMUL8. Seu valor padrão é de 5 unidades de tempo;
- *Results Collection Period*: define a duração do período de simulação considerado após o período de aquecimento (*warm-up*);
- *Simulation Time*: valor atual do relógio de simulação. Esta variável pode ser lida, mas não alterada diretamente pelo usuário;
- *Warm Up Time*: define a duração do período de aquecimento do modelo.

Para visualizar qualquer variável, inclusive as variáveis de sistema, mesmo durante a simulação, abra a *Watch Window* (*Advanced | Watch Windows*) ou tecle Crtl-Alt-W e, com o botão direito do *mouse*, clique em *Insert*. A Figura 5.2 mostra a aparência desta “janela de observação”.

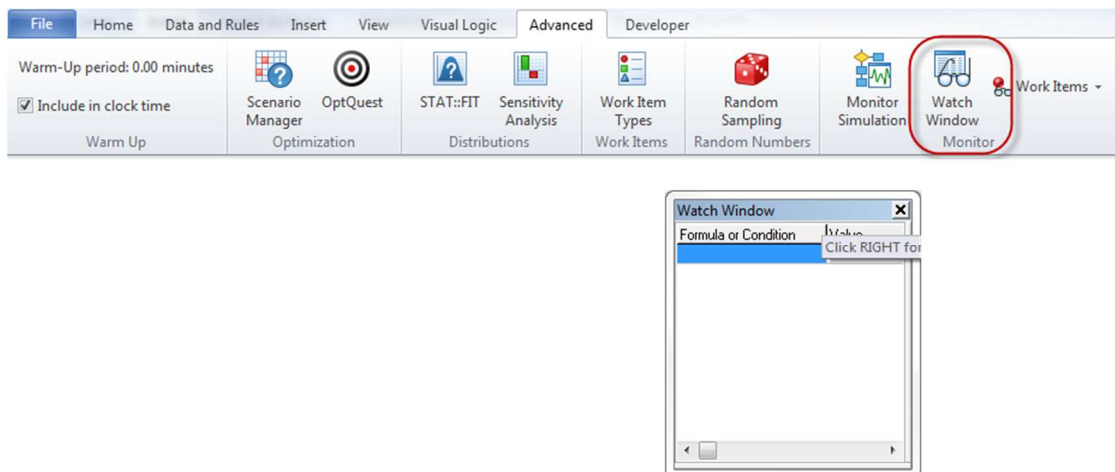


Figura 5.2 - Janela de visualização de variáveis (*Watch Window*).

Para criar uma variável definida pelo usuário (no caso, você), acesse o menu *Data and Rules* | *Information Store* ou utilize o atalho de teclado Ctrl-I. A partir da janela que surgirá, chamada *Information Store* (Figura 5.3), você poderá criar novas variáveis clicando no botão “New”.

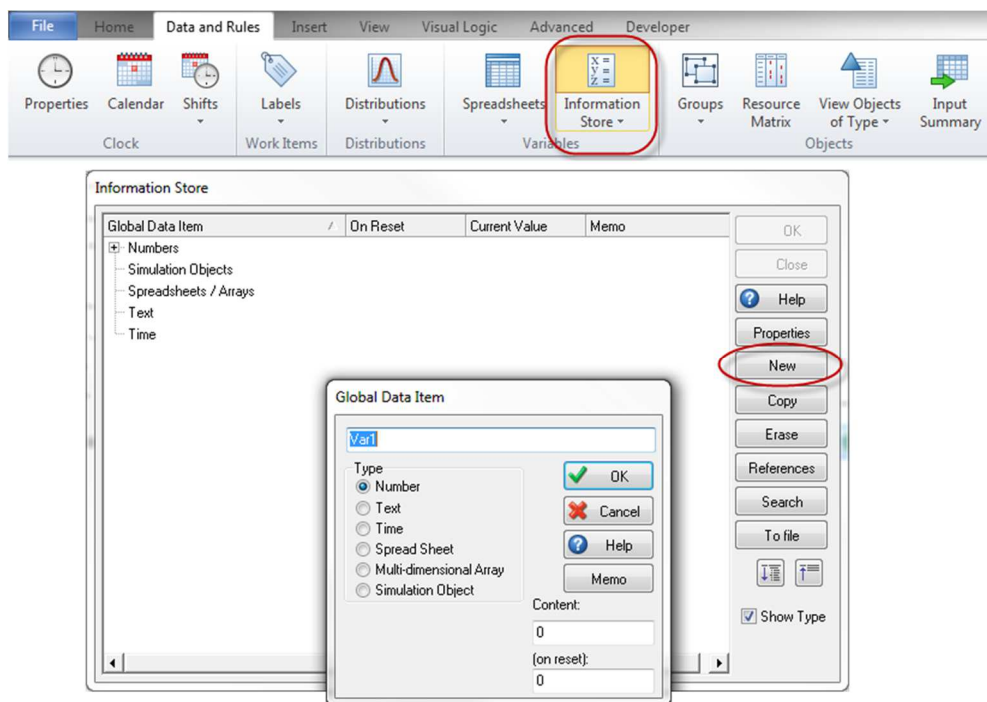


Figura 5.3 - Janela *Information Store*.

As variáveis numéricas (*number*) são as mais comuns e, como o nome indica, servem para armazenar valores numéricos inteiros ou fracionários. O valor “*On reset*” da variável indica o valor que ela possuirá quando a simulação for reiniciada (se este tipo de atribuição não for necessário, mantenha o valor padrão, que é zero). Outro tipo de variável definida pelo usuário que pode ser bastante útil é o tipo *Spread Sheet*, que se trata de uma variável bidimensional. Para visualizar ou alterar os valores armazenados neste tipo de variável, na aba *Data and Rules*, clique no botão *Spread sheets* e depois em *View*. Vai aparecer uma planilha semelhante à do Microsoft Excel®, conforme a Figura 5.4.

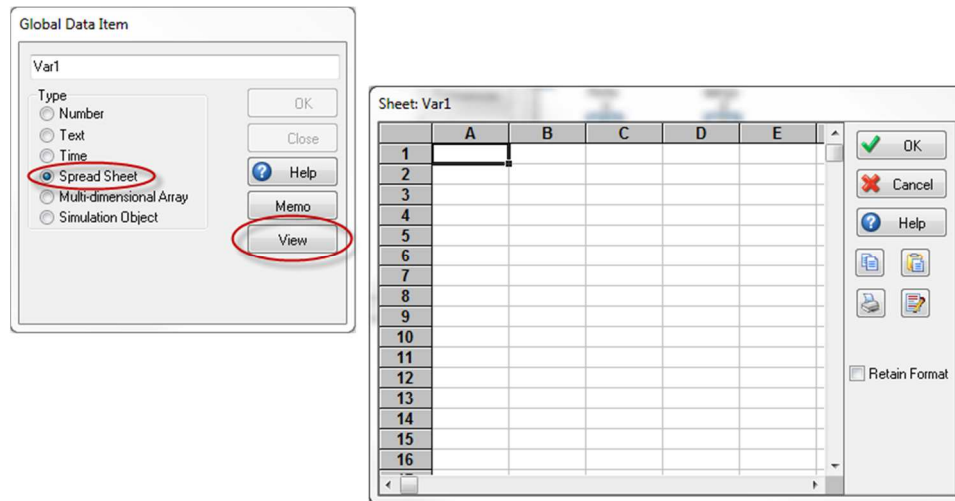


Figura 5.4 - Visualização de uma variável bidimensional (*Spread Sheet*).

5.3. Noções básicas de *Visual Logic*

A linguagem de programação do SIMUL8 é chamada de *Visual Logic*. O uso do ambiente de programação é necessário quando precisamos realizar operações lógicas complexas, que não possam ser feitas através das caixas de diálogos e funcionalidades básicas do SIMUL8. A Figura 5.5 ilustra um exemplo de código em *Visual Logic*.

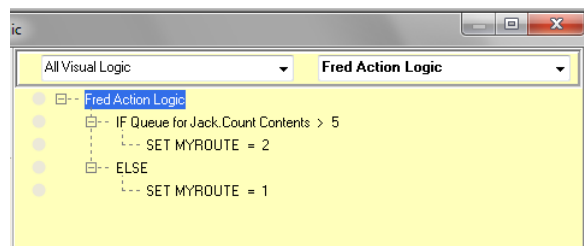


Figura 5.5 - Exemplo de bloco em *Visual Logic*.

Como o intuito desta seção é apenas discutir noções de programação no SIMUL8, vamos utilizar um dos exemplos prontos, fornecidos com o programa. Abra o arquivo *Label Routing.s8*, da pasta *SIMUL8\Examples\Visual logic*, ilustrado na Figura 5.6.

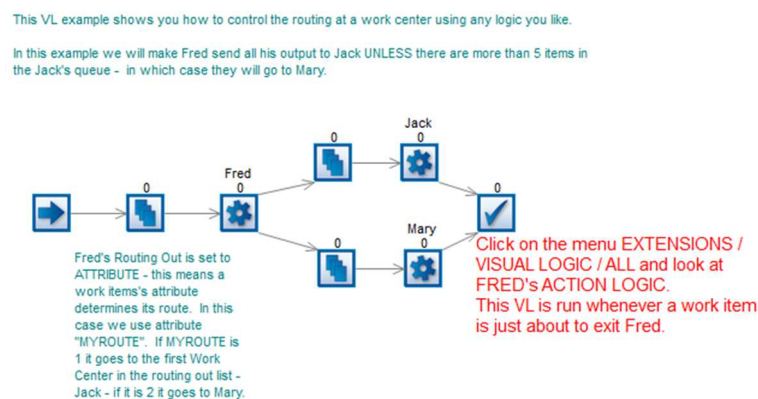


Figura 5.6 - Modelo de exemplo para *Visual Logic*.

Neste modelo, o direcionamento de saída do *Activity* "Fred" depende de um *label* chamado "MYROUTE": se o valor do *label* for igual a 1, a entidade será direcionada para o *Activity* "Jack"; se o valor for igual 2, a entidade irá para o

Activity “Mary”. O valor do *label* é ajustado por um bloco de código (ou bloco de lógica) chamado “Fred”, associado ao *Activity* “Fred”. Para acessá-lo, abra a janela de propriedades do *Activity* e clique no botão *Actions...* O botão que permite acessar o código previamente criado estará inacessível, já que não há nenhum *label* selecionado. Clique no botão *Add a Label to Change* e selecione o *label* “MYROUTE”. Agora, clique no botão “*Visual Logic?*” (que se tornou visível). A lógica que aparecerá é a que foi mostrada na Figura 5.6.

Vamos entender o que este código faz: se o número de entidades na fila antes do *Activity* “Jack” (“Queue for Jack.Count Contents”) for maior do que 5, então o *label* “MYROUTE” receberá o valor 2, indicando um direcionamento para o *Activity* “Mary”. Em caso contrário, o direcionamento será para o *Activity* “Jack”. Feche todas as caixas de diálogo abertas e simule o modelo em baixa velocidade para verificar o seu funcionamento.

Para acessar qualquer bloco de lógica de um modelo, clique em *Visual Logic* | *All* e selecione o código desejado.

É importante ressaltar que todos os blocos de lógica do SIMUL8 são locais, no sentido de que devem ser associados a objetos ou eventos específicos. Vamos esclarecer do que isso se trata:

- Alguns eventos aos quais podemos associar blocos de lógica são o de reinício do modelo (*On Reset Logic*), final da simulação (*On End Run*), início da simulação (*On Start Run*) etc. Neste caso, a inserção dos códigos é feita em *Visual Logic* | *Time Based*, no caso de eventos baseados no tempo, (como *End Run*) ou em *Visual Logic* | *Action Based*, no caso de eventos baseados em ações, como *On Start Run*. Existe, ainda, um evento especial chamado *Visual Logic Time Check* (acessível a partir do menu *Visual Logic* | *Time Logic VL*), que executa um bloco de lógica em intervalos de tempo pré-determinados. Esse tipo de evento é muito útil para quando queremos, por exemplo, gerar registros (“logs”) do estado de uma simulação;
- Também podemos associar blocos de lógica a eventos de objetos de simulação. Os casos mais comuns são: os direcionamentos de entrada ou saída de *Activities* e nas alterações de *labels* (*Label Actions*) dos pontos de entrada ou *Activities*. No entanto, há inúmeros outros pontos onde se podem associar blocos de lógica.

Os comandos mais utilizados em *Visual Logic* são:

- **SET** <var> = <valor>: atribui um valor a uma variável;
- **IF Then Else**: estrutura condicional;
- **LOOP**: bloco de repetição.

Vamos explicar algumas formas de uso de alguns destes comandos no exemplo a seguir.

5.4. Exemplo com variáveis e *Visual Logic*

Em um centro de distribuição, os pedidos chegam em intervalos médios de um pedido a cada 10 minutos, segundo uma distribuição exponencial. Caso haja produto em estoque, o pedido será atendido imediatamente, na razão de uma unidade de produto por pedido. Caso contrário, o pedido ficará em espera até a chegada do produto em estoque. A política de ressuprimento estabelece que, quando o estoque atinge 100 unidades, ou menos, o centro de distribuição deve realizar um pedido de ressuprimento de 250 unidades, que serão entregues, em média, 24 horas após a solicitação. Este *lead time* de transporte é descrito por uma distribuição *average*¹⁸.

Propõe-se simular a operação durante 7 dias, 24 horas por dia (ou seja, 10.080 minutos) com o intuito de avaliar o valor do estoque médio no período e o número máximo de clientes esperando em fila, bem como o tempo máximo

¹⁸ Como descrito no Capítulo 2, a distribuição *average* corresponde a uma distribuição normal com desvio padrão igual a 25% do valor da média.

de espera. Deseja-se, ainda, determinar o total de pedidos feitos no período e gerar um registro (“log”) do estoque de produtos, de hora em hora. Considere que o estoque inicial do modelo é de 80 unidades.

Se você chegou até este ponto do livro, não deve ter dificuldades em construir o modelo ilustrado na Figura 5.7. Note que a atividade “batching” é uma atividade *dummy* (ou seja, possui tempo de execução igual a zero, com distribuição fixa). A razão de se utilizar esta atividade será explicada na sequência.

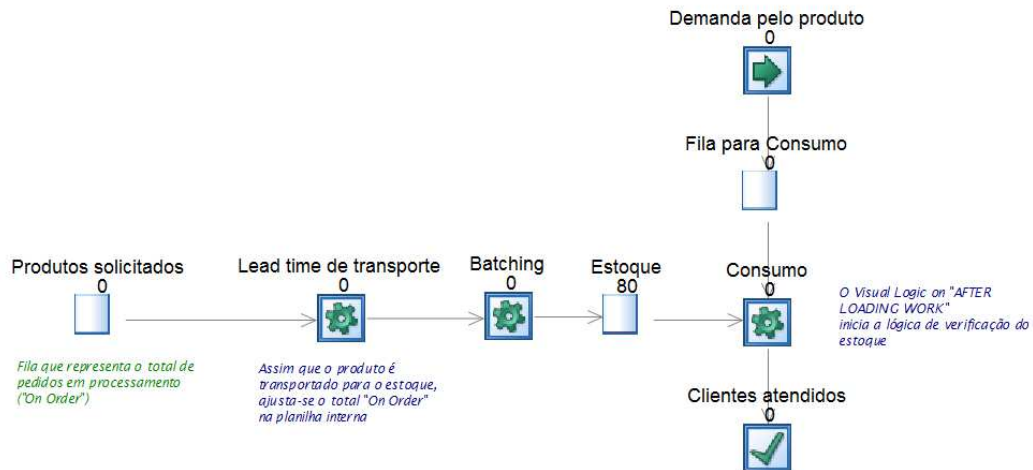


Figura 5.7 - Modelo de ressuprimento.

A partir da estrutura apresentada na Figura 5.7, devemos ajustar seus diversos parâmetros. Para o Activity “Lead Time de Transporte”, o preenchimento deve ser semelhante ao apresentado na Figura 5.8.

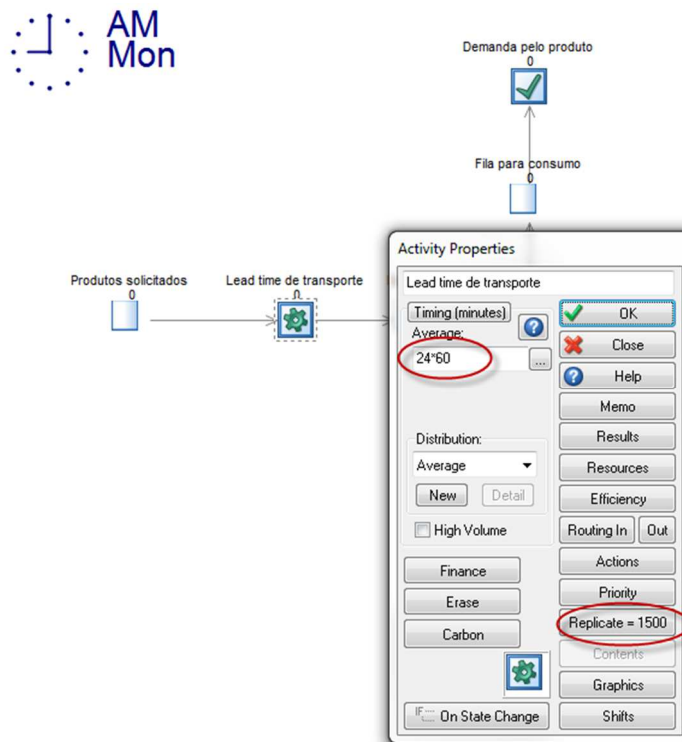


Figura 5.8 - Propriedades do Activity “Lead Time de Transporte”.

Note que a duração da atividade (ou seja, o tempo médio de transporte) é de um dia ($=24*60$ minutos) e que foi usado um valor alto (1.500) para a propriedade *Replicate*. Escolhemos um valor de *Replicate* grande pois, teoricamente, várias cargas poderiam transitar simultaneamente, dependendo dos parâmetros de ressurgimento. Caso o valor *default* seja utilizado (*replicate* = 1), se tivermos 2 cargas simultâneas uma estará na atividade Lead Time Transporte e outra ficará aguardando na fila de produtos solicitados, adicionando tempo de espera ao Lead Time Transporte o que não faz sentido.

O direcionamento de entrada do *Activity* “Consumo”, como mostra a Figura 5.9, deve ser tal que seja coletado 1 item da fila “Estoque” e 1 item da fila “Fila para Consumo”, com as opções *Do not Collect Until All Available* e *Assemble* ativadas. Assim, cada cliente/pedido que chega ao sistema retira um produto do estoque e sai. Por fim, a duração da atividade representada pelo *Activity* “Consumo” deve ser (fixa) igual a zero, pois supomos que o consumo é imediato, caso haja produto em estoque.

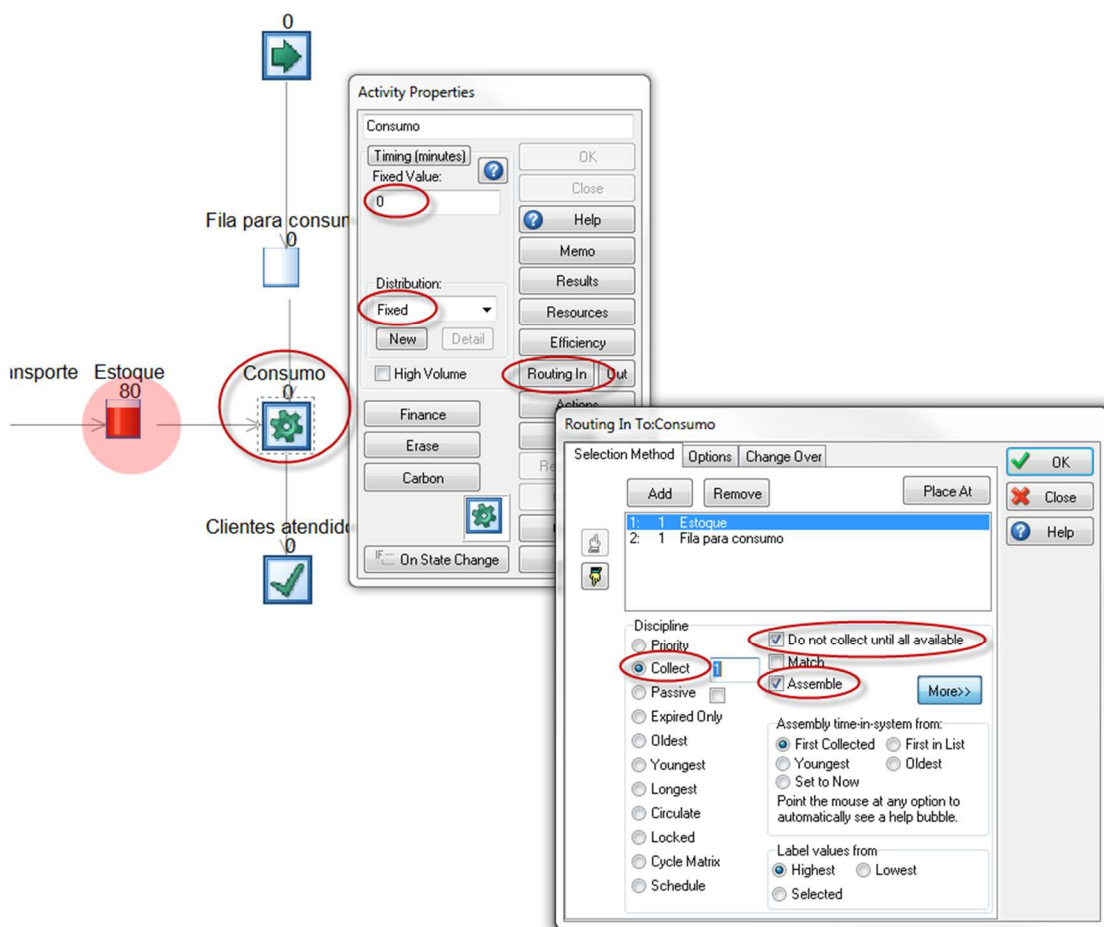


Figura 5.9 - Disciplinas de entrada do Activity "Consumo".

Agora, vamos às variáveis. Crie uma variável numérica denominada “V_total_pedidos”, que vai contabilizar quantos pedidos foram feitos. Para isso, clique em *Data and Rules* | *Information Store* e crie uma variável do tipo numérica, com o nome indicado, mantendo o valor *On Reset* padrão (zero), como representado na Figura 5.10. De modo semelhante, crie uma variável chamada “V_conta_log”, que será usada na geração do registro do estoque de produtos. O valor inicial (*On Reset*) desta variável deve ser igual a 10, pois o “log” irá ser contabilizado a partir da linha 10 da planilha de dados.

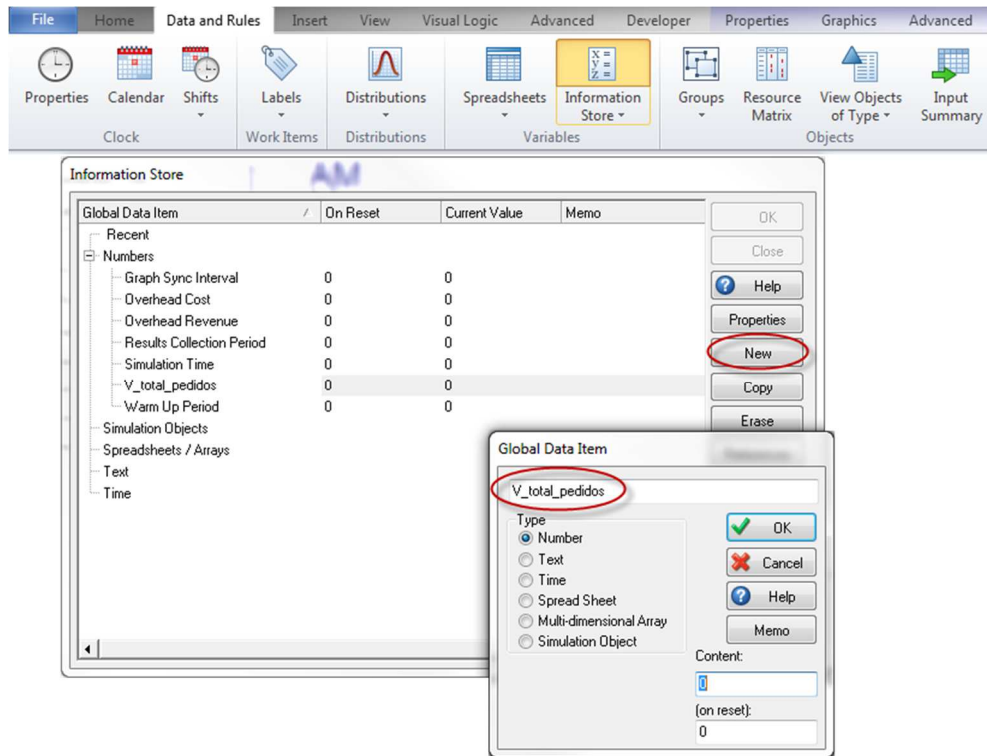


Figura 5.10 - Criação da variável “V_total_pedidos”.

Por fim, crie uma variável bidimensional (tipo *Spread Sheet*) chamada “Planilha_Config_Estoque”, preenchendo algumas células com os valores mostrados na Figura 5.11 (não se preocupe com a formatação das células). Observe que esta variável armazena 5 parâmetros, sendo que os 3 primeiros se referem ao estoque e os demais são indicadores: “Disponível”, que mede o quanto se tem de estoque no momento e “Em trânsito”, que mede o que já foi ordenado (ou seja, quantas unidades já foram pedidas ao fornecedor). Este controle é necessário porque, se não monitorarmos a quantidade em trânsito para o estoque, corremos o risco de gerar múltiplos pedidos de ressuprimento, enquanto o pedido “inicial” ainda não chegou. É evidente (para nós) que só se deve gerar um novo pedido caso a soma das quantidades em estoque e em trânsito seja menor do que 100 (ponto de pedido).

Na linha 9, colocamos um texto que formará o cabeçalho do nosso registro de estoque (“log”).

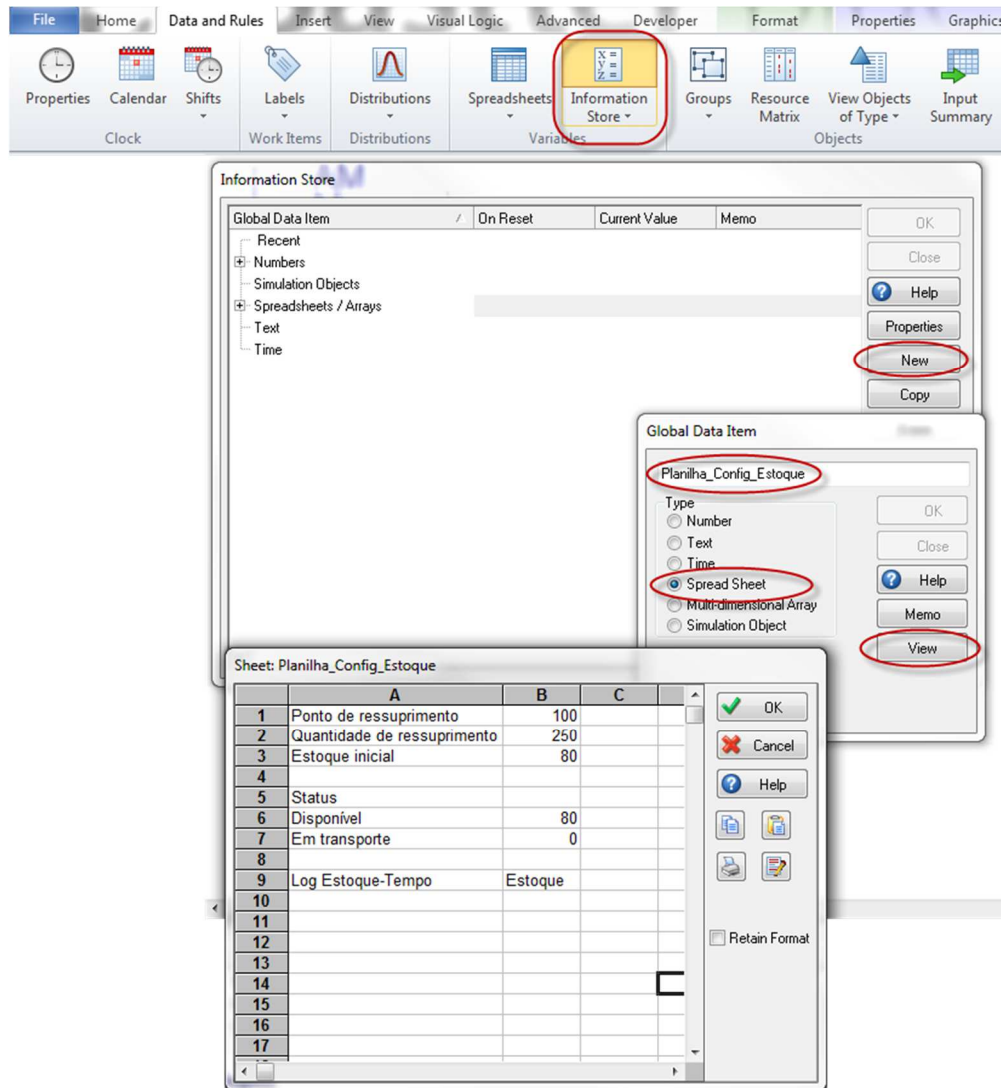


Figura 5.11 - Variável bidimensional para controle do estoque.

Como visualizaremos o valor do ponto de ressuprimento durante a simulação? Basta colocarmos na *Watch Window* a variável “Planilha_Config_Estoque[2,1]”, pois se trata da coluna 2 e linha 1 da planilha (isso mesmo: a indexação de células no SIMUL8 é feita com [<Número de Coluna>, <Número de Linha>]) Ao se clicar no campo *Formula or Condition*, abre-se automaticamente a janela do *Formula Editor* que facilita todo o processo, como mostrado na Figura 5.12.

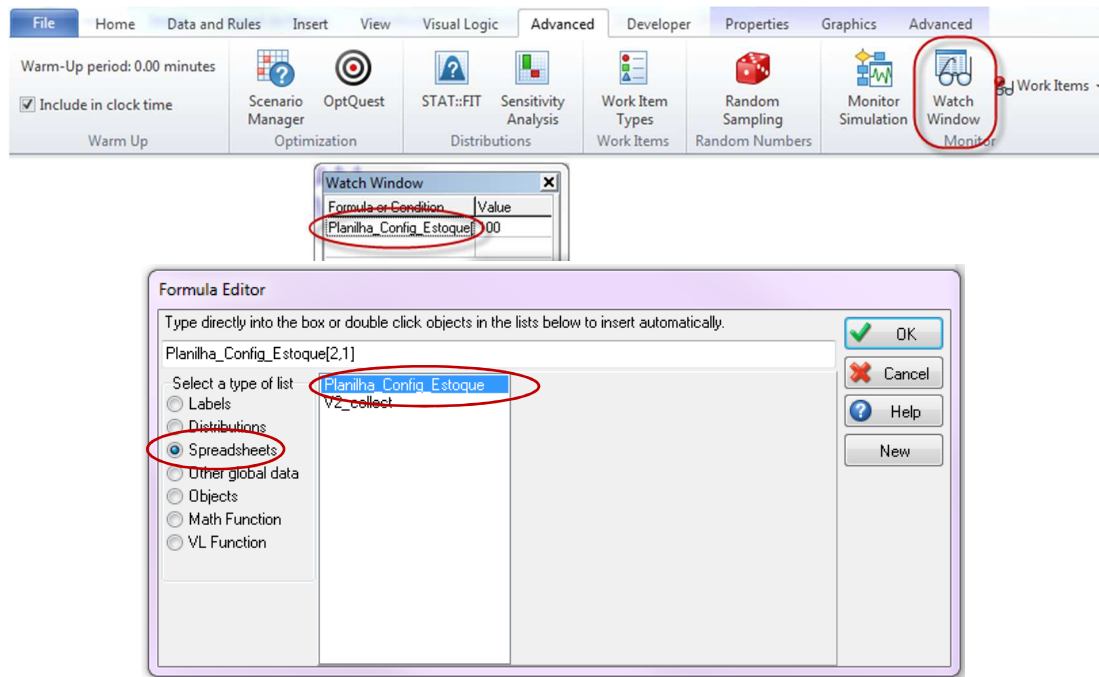


Figura 5.12 - Preenchimento da Watch Window.

Finalmente, vamos à programação. Em primeiro lugar, definiremos a lógica de inicialização, que será executada uma única vez, antes do início da simulação. Para acessar (e editar) o bloco de lógica adequado, clique em: *Visual Logic* | *Time Based* | *Before Reset* (Figura 5.13). Esta lógica será executada logo antes do comando de *Reset* o modelo.

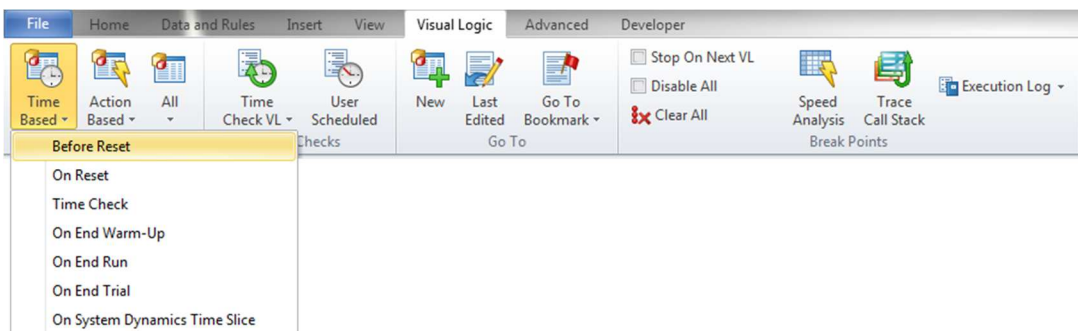


Figura 5.13 - Acesso à lógica "Before Reset".

O SIMUL8 abrirá uma janela para editarmos o código que será executado assim que o usuário clicar no botão "Reset". (Figura 5.13).

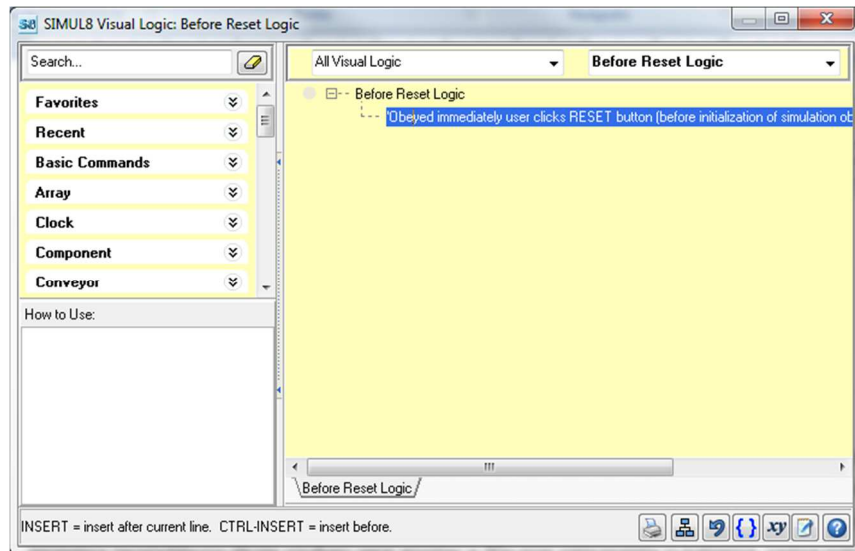


Figura 5.14 - Tela de código do Visual Logic.

Os comandos em *Visual Logic* podem ser inseridos clicando-se com o botão direito do *mouse* na região amarela da tela. Por exemplo, iremos iniciar a lógica com um comentário, como mostra a Figura 5.16.

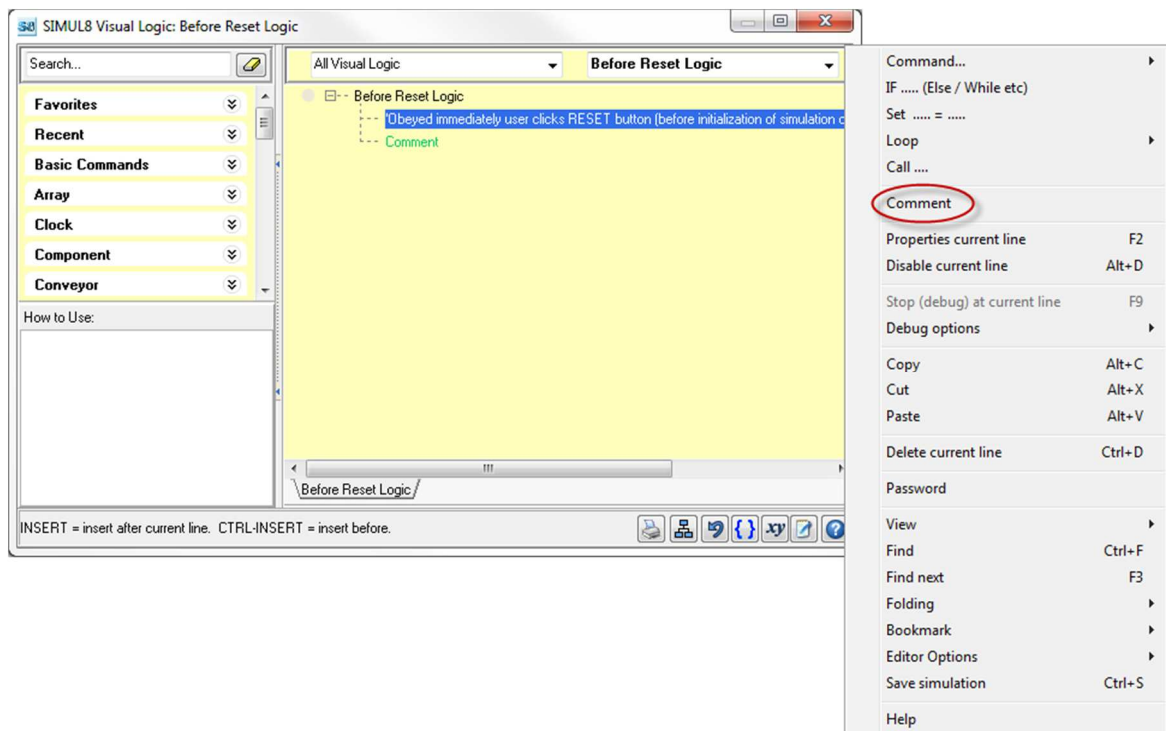


Figura 5.15 - Menu de opções do Visual Logic.

Como a primeira providência deste código será ajustar a fila que representa o estoque, insira um comentário que descreva esta operação. Aliás, é sempre uma boa prática de programação inserir comentários pertinentes ao código que está sendo desenvolvido (Figura 5.16).

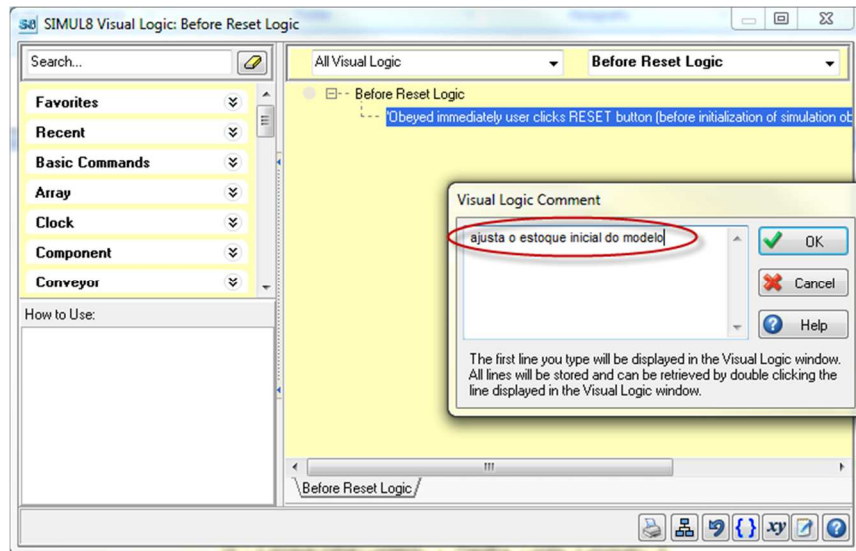
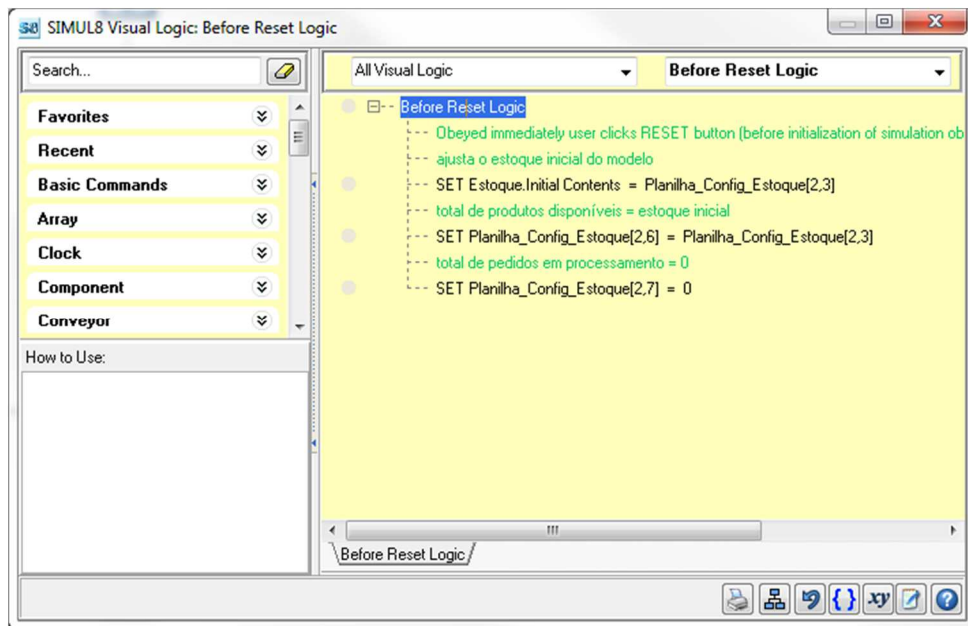


Figura 5.16 - Atribuição de comentário ao código.

Agora, construa o código mostrado na Figura 5.17.

Figura 5.17 - Lógica para o *Before Reset* do exemplo.

O comando *Set* pode ser inserido da mesma forma que o comentário (botão direito do *mouse*). Note que a primeira linha efetiva da lógica ajusta (*set*) o valor inicial da fila para o valor (80) inserido na planilha. Note, ainda, que o nome do objeto é “Estoque” e que *Initial Contents* é a propriedade que será alterada¹⁹, parâmetro facilmente obtido no *Visual Logic* a partir do *Formula Editor*, como mostrado na

¹⁹ O leitor familiarizado com Programação Orientada a Objetos deverá reconhecer a notação “Objeto.Propriedade”.

Figura 5.18. A segunda linha do código ajusta o total “Disponível” para este mesmo valor, enquanto a terceira linha efetiva de lógica atribui zero para o valor “Em transporte”, já que nenhum pedido de ressuprimento foi feito no início da simulação.

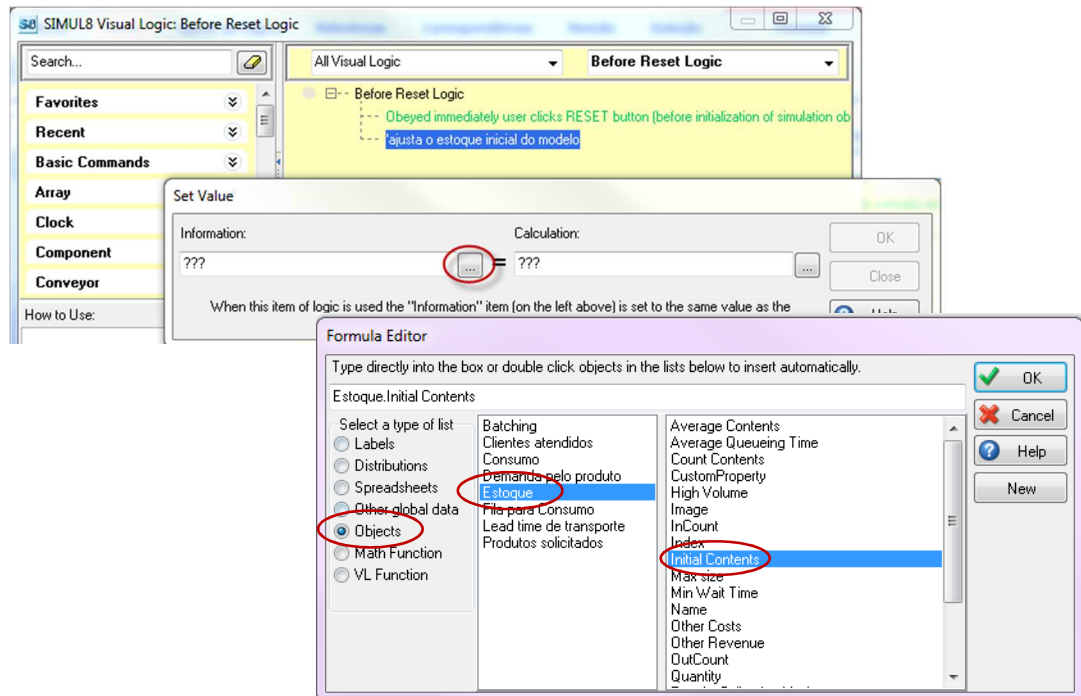


Figura 5.18 - Preenchimento do comando Set com o Formula Editor.

Vamos agora configurar o *Activity* “Batching”. Note que esta é uma atividade do tipo “dummy”, ou seja, seu tempo deve ser sempre igual a zero (distribuição *Fixed*), já que a sua função é “explodir” a carga de um “caminhão” que será criada na fila “Produtos Solicitados” (isso será explicado a seguir). O leitor pode estar se perguntando porque não se configura o *batching* na própria atividade “Lead Time Transporte”. Não se pode fazer isso porque a funcionalidade *batching* fica desabilitada quando a propriedade *Replicate* é ajustada para um valor diferente de 1 e, por isso, a necessidade da atividade “dummy”. Assim, como ilustra a Figura 5.19, o valor de *batching* deverá ser ajustado de acordo com o valor de *Planilha_Config_Estoque* [2,2].

Vamos, agora, inserir a lógica de atualização do valor *On Order*. No direcionamento de saída (*Routing Out*) do *Activity* “Batching”, clique na seção de lógica *On Exit* (Figura 5.19). Isto lhe dará acesso à lógica que o SIMUL8 executará quando alguma entidade sair do *Activity*. Neste ponto, insira a seguinte instrução:

SET Planilha_Config_Estoque[2,7] = Planilha_Config_Estoque[2,7]-1

Esta instrução atualiza o valor da variável “Em pedido”, pois assim que um produto é entregue, um item deixa de estar “Em transporte”.

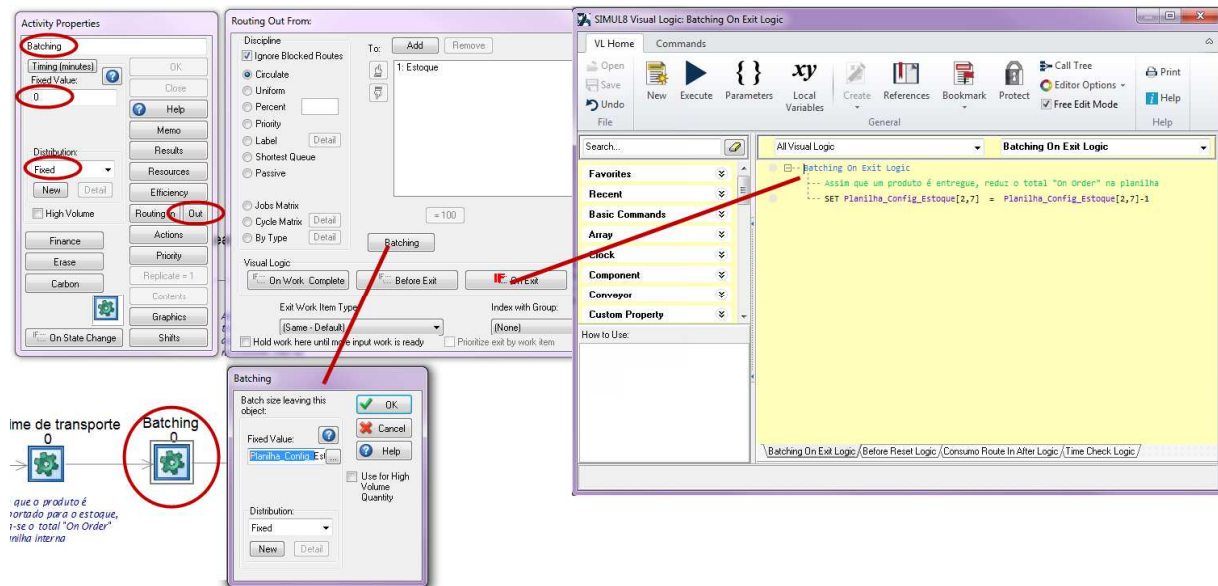


Figura 5.19 - Routing Out on Exit Logic.

Agora vamos à lógica mais complicada do modelo, que deve ser inserida no direcionamento de entrada (*Routing In*) do *Activity* “Consumo”, como indicado na Figura 5.20: clique na aba “Options”, onde está disponível o botão *After Loading Work*.

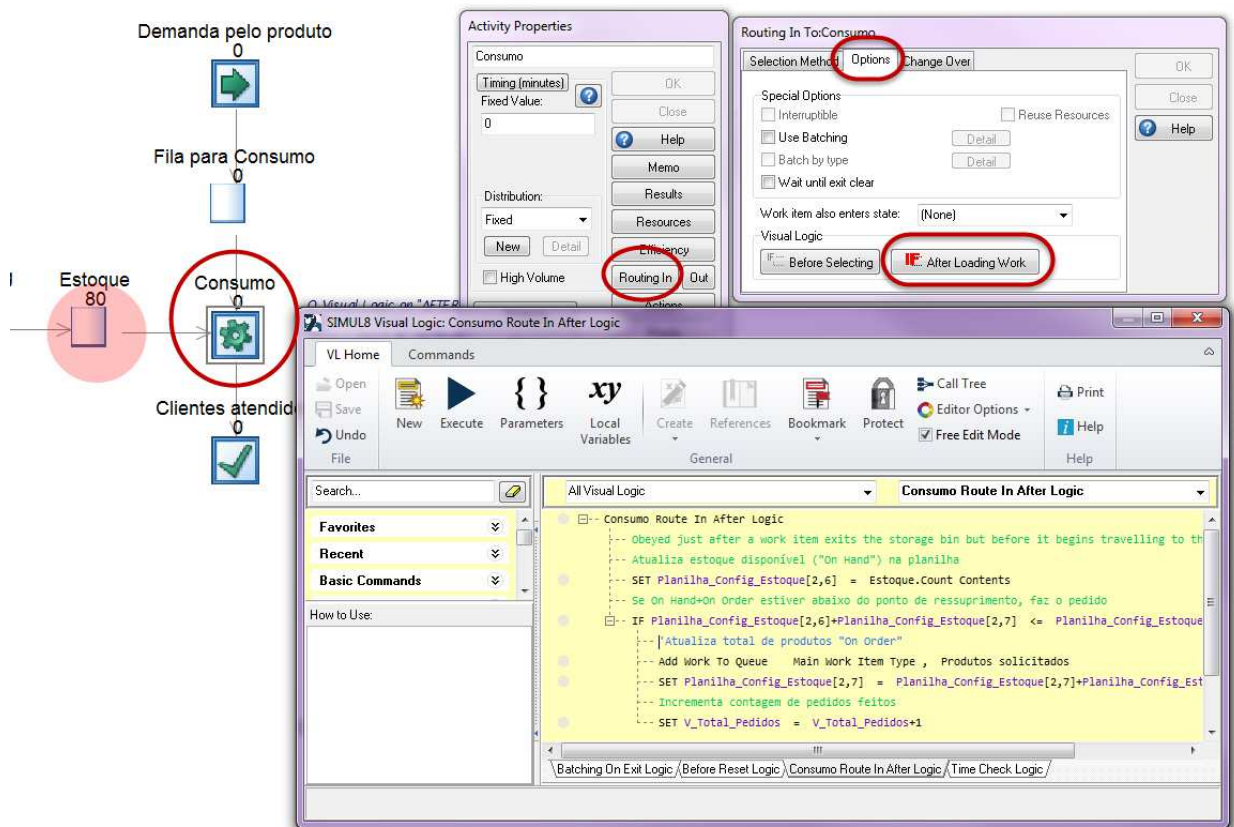


Figura 5.20 - Routing In After Loading Work.

A lógica a ser inserida neste bloco é a seguinte:

'atualiza estoque disponível na planilha

```
SET Planilha_Config_Estoque[2,6] = Estoque.Count Contents
```

'se disponível + em transporte estiver abaixo do ponto de ressuprimento, faz o pedido

```
IF Planilha_Config_Estoque[2,6]+Planilha_Config_Estoque[2,7] <=
Planilha_Config_Estoque[2,1]
```

```
    Add Work To Queue    Main Work Item Type ,  Produtos solicitados
```

'atualiza total de produtos "On Order"

```
SET Planilha_Config_Estoque[2,7]=
Planilha_Config_Estoque[2,7]+Planilha_Config_Estoque[2,2]
```

'incrementa contagem de pedidos feitos

```
SET V_Total_Pedidos = V_Total_Pedidos+1
```

Lembre-se de que o código anterior será executado sempre que uma unidade de produto for retirada do estoque, ou seja, quando um cliente for atendido. A primeira linha do código atualiza o indicador de estoque. A segunda é uma estrutura condicional: se “disponível + em transporte” estiver abaixo do ponto de ressuprimento, um novo pedido será feito. Neste caso será adicionado um item à fila “Produtos Solicitados” pelo comando *Add Work to Queue*. Em seguida, o valor *On Order* será atualizado com base no pedido e, finalmente, a última linha de lógica irá incrementar a variável que contabiliza o número de pedidos realizados.

Agora falta apenas criarmos a lógica do “log” de registro para que, a cada 60 minutos, o SIMUL8 crie uma linha na planilha, representando o tempo de simulação e o número de produtos em estoque naquele momento. Para tanto, acesse *Visual Logic | Time Check VL | Time Check Properties* (Figura 5.21).

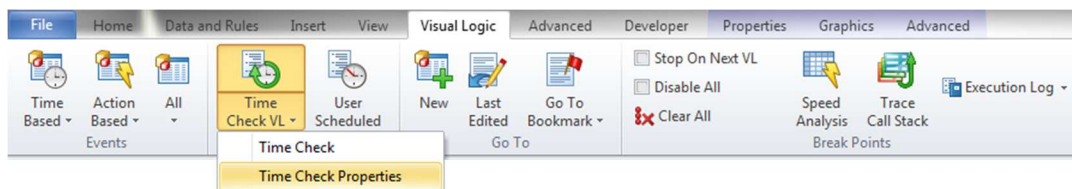


Figura 5.21 - Visual Logic Time Check.

Configure *First Time check* para 0 e *Time Interval* para 60 minutos. Isto fará com que a lógica que criaremos seja executada no instante 0 da simulação e, em seguida, a cada 60 minutos. Em seguida, abra a janela de código do *On Time Check Logic*, como mostrado na Figura 5.22.

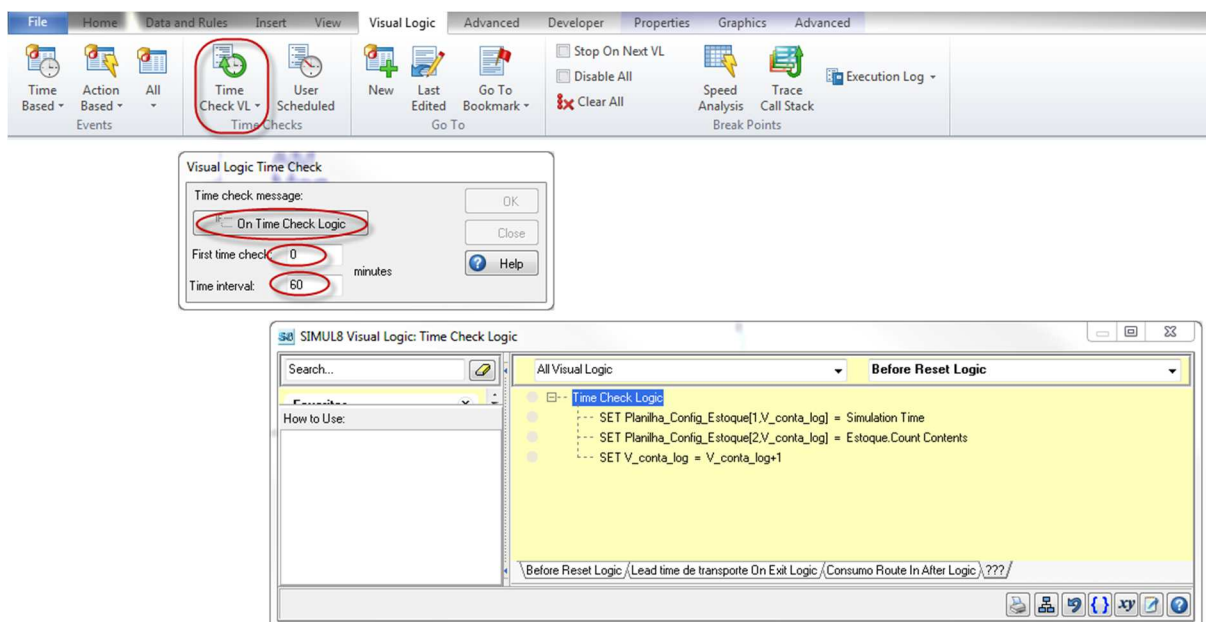


Figura 5.22 - Lógica do Time Check.

Na janela de código que se abrirá (Figura 5.22) insira as seguintes instruções:

```

SET Planilha_Config_Estoque[1,V_conta_Log] = Simulation Time

SET Planilha_Config_Estoque[2,V_conta_Log] = Estoque.Count Contents

SET V_conta_Log = V_conta_Log+1

```

As duas primeiras linhas do código gravam na planilha o tempo de simulação e quantidade atual em estoque. Note que a linha utilizada da planilha é indexada pela variável “V_conta_log” (iniciada com o valor 10), que é incrementada na terceira linha do código. Assim, a cada gravação de valor na planilha, uma nova linha será utilizada.

Para o leitor mais avançado vale a pena comentar que seria possível utilizar outra forma de modelagem equivalente à que acabamos de explicar. Neste caso apagaríamos a *Activity* “Lead Time Transporte”, conectando diretamente a fila “Produtos Solicitados” ao *Activity* Batching. Mas daí o leitor poderia se perguntar como seria representado o tempo de transporte. Bem, este seria inserido na propriedade *Min Wait Time* da fila “Produtos Solicitados”, fazendo com que toda entidade adicionada à fila espere, no mínimo, este valor de tempo (esta funcionalidade será melhor detalhada no capítulo 7). Como a atividade Batching ocorre em tempo 0, ela não causará retenção da entidade que aguardou seu tempo mínimo de processamento. Outro detalhe importante é que a lógica de atualização do estoque em transito foi inserida no evento *on Exit Logic* da atividade Batching, decrementando o estoque em transito de uma em uma unidade. Uma lógica totalmente equivalente seria inserir um decremento do valor da quantidade de ressuprimento (no caso 250) no evento *On Work Complete*, já que, neste ponto ainda não ocorreu a “explosão” da entidade em 250 unidades. Enfim, em simulação, há vários modos de se construir um modelo.

Por fim, vamos fazer com que seja exibido na tela o valor da variável “V_total_pedidos”. Para isso, clique em *Insert | Visual Data*, posicione o cursor em algum ponto da tela e selecione a variável “V_total_pedidos”. Por questões de clareza, escreva um texto explicativo à esquerda da variável (*Insert | Text Box*). Você também pode adicionar os gráficos das filas, mantendo-os em tela, como explicado no Capítulo 1. Este procedimento fará com que o modelo se assemelhe ao que é mostrado na Figura 5.23.

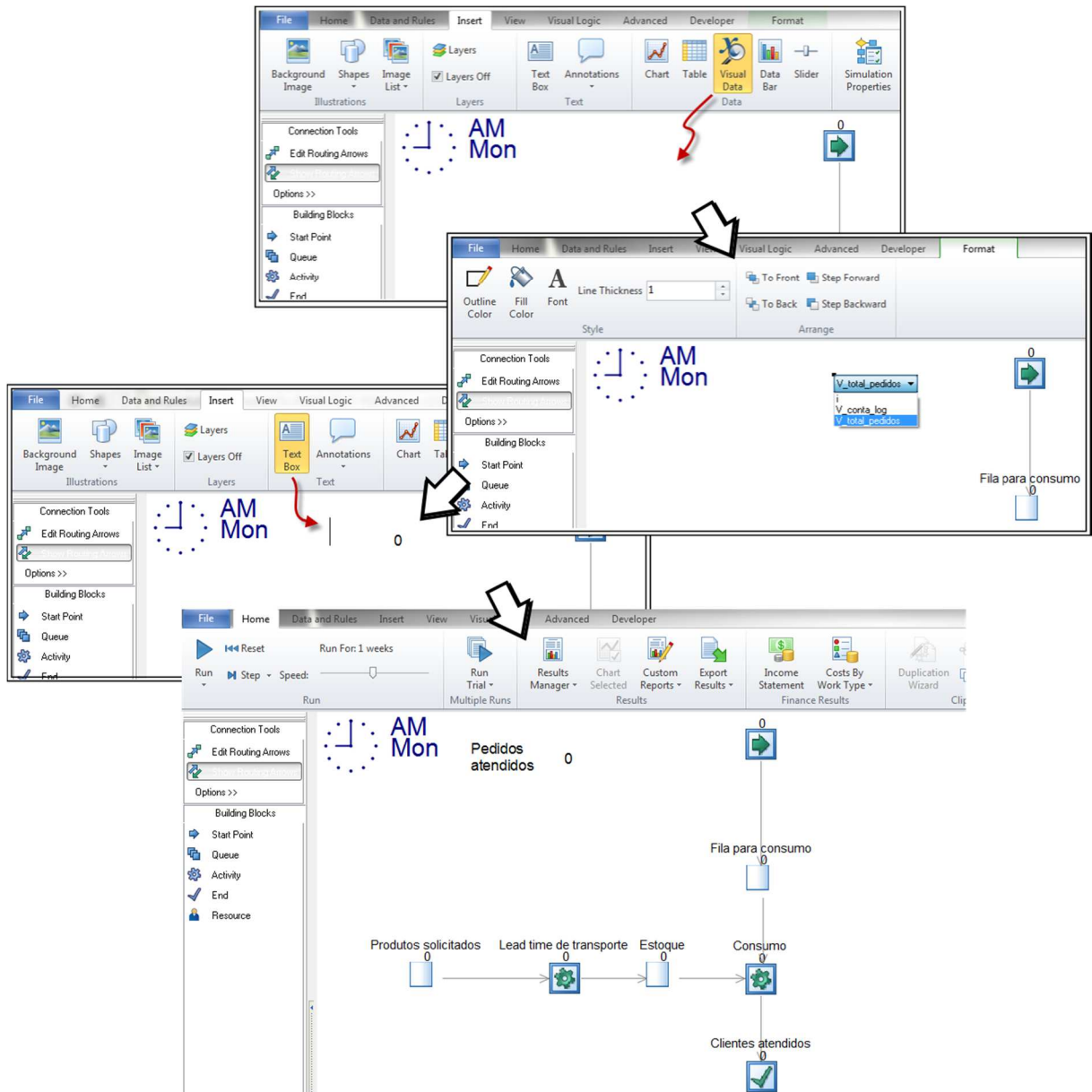


Figura 5.23 - Sequencia para colocação do *display* de tela com o valor da variável “V_total_pedidos”.

Antes de rodar o modelo, ajuste o relógio de simulação para contabilizar 24 horas por dia (início às 0:00 horas, duração de 24 horas), sete dias por semana, ou seja, 10.080 minutos (*Results Collection Period*), sem período de *warm-up*. Após a simulação, clique em *Data and Rules* | *Spreadsheets* e acesse a variável que representa a planilha contendo o “log”. A Figura 5.24 mostra um exemplo do aspecto desta planilha após a simulação.

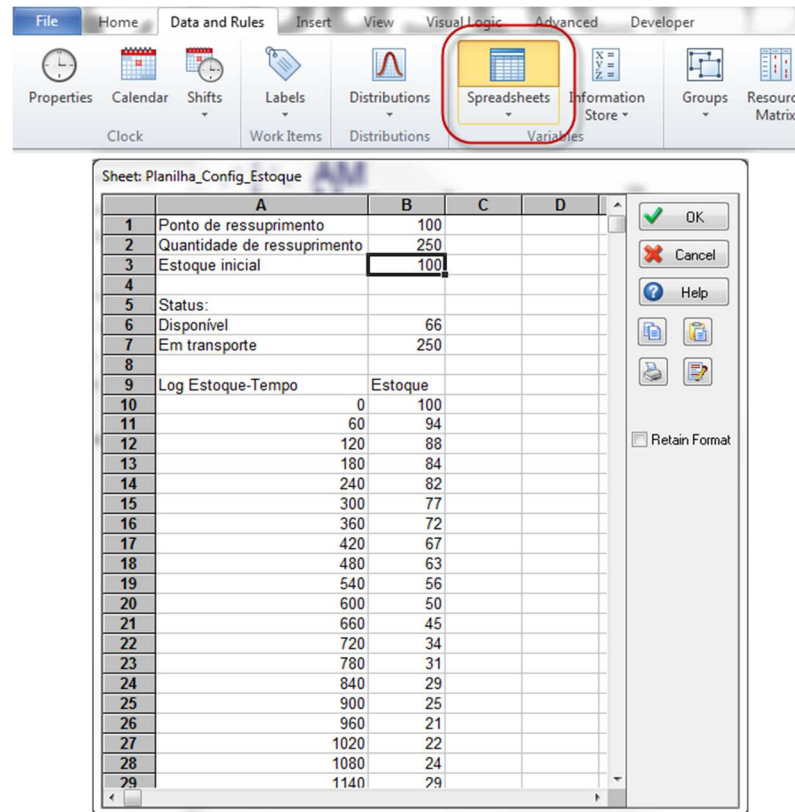


Figura 5.24 - Planilha com as saídas e o "log" da quantidade disponível em estoque a cada hora.

Com o modelo pronto (e funcionando corretamente), vamos recordar as questões inicialmente propostas: deseja-se saber o valor do estoque médio no período, o número máximo de clientes esperando em fila, o tempo máximo de espera e o total de pedidos feitos pelos clientes.

Para respondermos a essas questões devemos adicionar as variáveis relevantes ao resumo de resultados: estoque médio, número máximo e tempo máximo de espera na fila para consumo, total de pedidos de ressuprimento e número total de clientes atendidos. O procedimento para inserir dados de saída de objetos (por exemplo, tamanho médio de fila) no resumo de resultados já foi discutido no Capítulo 1. Para adicionar uma *variável* ao resumo, clique em *Data and Rules* | *Information Store*, selecione a variável de interesse (no caso "V_total_pedidos") e, com o botão direito do *mouse*, escolha a opção *Add to KPI's*. Com isso, após a simulação, o resumo de resultados deverá ser semelhante ao que é mostrado na Figura 5.25 (resultados obtidos após 100 replicações do modelo).



Figura 5.25 - Resultados do modelo.

Como mostrado na Figura 5.25, obtivemos um estoque médio em torno de 83 itens, o número máximo de clientes esperando por produto é de cerca de 90 clientes e o tempo máximo de espera está ao redor de 900 minutos. Para algumas situações estes valores poderiam ser críticos, para outras não. Por exemplo, você esperaria em uma fila por 15 horas (900minutos) para receber um eletrodoméstico? Talvez a resposta seja sim, mas se o produto fosse uma bala ou biscoito, você provavelmente desistiria. Como exercício, sugerimos que você teste outros parâmetros de entrada do modelo (estoque inicial, ponto de ressuprimento e quantidade de ressuprimento) para obter melhores desempenhos. Outra sugestão é incluir a possibilidade de desistência.

Vamos encerrar este capítulo lembrando ao leitor que há inúmeros outros comandos disponíveis em *Visual Logic*, que não serão descritos neste livro. Para ver a lista completa, abra uma janela de lógica qualquer e, com o botão direito *mouse*, selecione *Command | List All in Note Pad*. Veja que a lista de comandos é extensa, mas acreditamos que o objetivo deste capítulo tenha sido atingido ao fornecermos subsídios introdutórios à programação em *Visual Logic*.

Capítulo 6 REFINAMENTOS GRÁFICOS E ANIMAÇÃO

6.1. Introdução

Neste ponto do livro, você já deve estar familiarizado com a ideia de que o objetivo da simulação é a criação de um modelo com a finalidade de analisar o comportamento de um sistema. Em última análise, o conjunto de resultados do modelo é a peça chave para compreensão do sistema. No entanto, também há um aspecto muito importante presente na maioria dos simuladores comerciais: a animação gráfica dos resultados e processos.

O SIMUL8 já cria alguma animação automaticamente quando simulamos nossos modelos. A ideia agora é enriquecermos nossos modelos com mais elementos gráficos, tornando-o mais didático para quem “assiste” a simulação.

A bem da verdade, nem tudo é exatamente “animado”, no sentido de movimentação em tela. Boa parte do trabalho consiste na colocação de elementos gráficos ou, como o título deste capítulo sugere, *refinamentos gráficos*. Invariavelmente, contudo, utilizaremos o termo “animação” para nos referirmos a tudo que se refere à parte gráfica do modelo, animada ou não.

No SIMUL8, os recursos de animação permitem:

- Auxiliar o processo de verificação do modelo: muitos erros ou “bugs” de construção podem ser detectados graças à animação gráfica;
- Auxiliar a comunicação do projeto: clientes ou mesmo equipes de projeto terão muito mais confiança no seu modelo, caso tenham a oportunidade de verificar o seu funcionamento por meio da animação do processo. Evidentemente, o modelo também deve produzir resultados confiáveis.

Um primeiro aspecto da animação em modelos de simulação é o seu grau de detalhamento. A Figura 6.1 mostra modelos de simulação sem muitos elementos gráficos (a) e com um grau mediano de elementos (b).

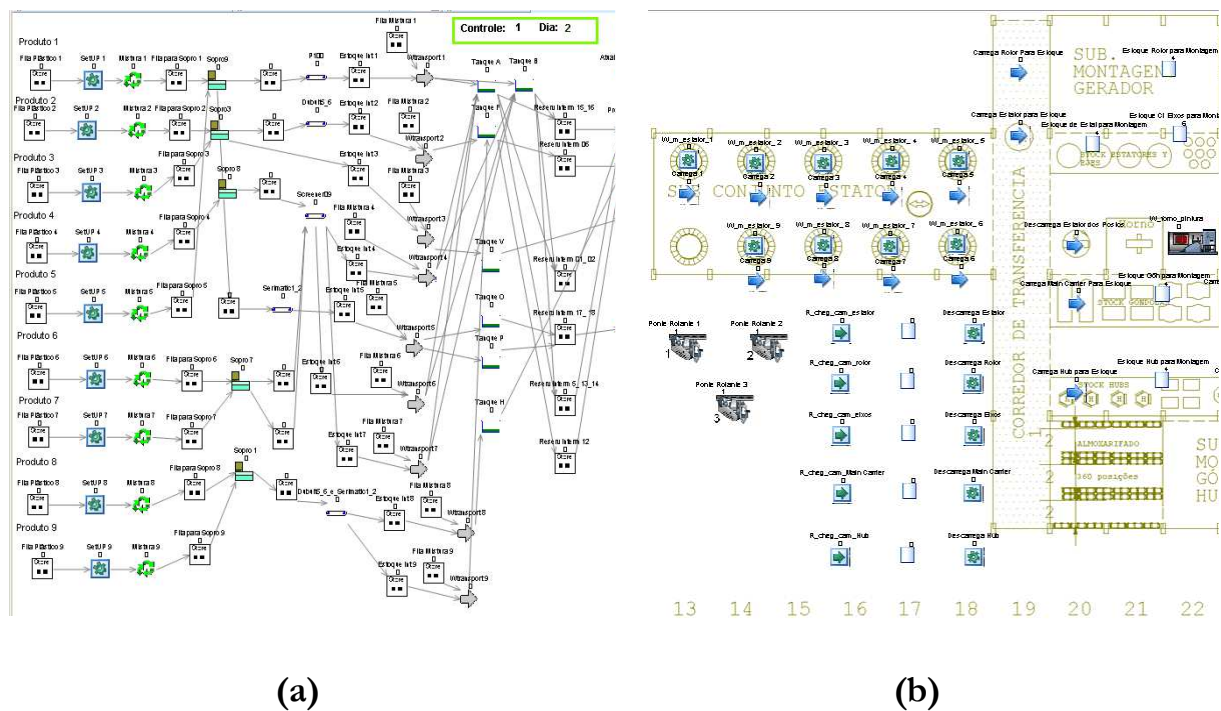


Figura 6.1 – Exemplos de diferentes níveis de refinamento gráfico e animação.

Para incrementar a animação de um modelo, podemos atuar basicamente em três níveis:

- Fundo de tela;
- Objetos de simulação;
- Entidades.

Discutiremos cada um destes níveis nas próximas seções.

6.2. Fundo de tela

O primeiro nível de melhorias gráficas trata do fundo de tela da janela de simulação. Neste caso, lidaremos apenas com elementos estáticos, ou seja, sem animação.

Vamos discutir algumas funcionalidades disponíveis para a construção de objetos de fundo:

- *Insert | Background Image*: insere uma imagem de fundo, que pode ser um arquivo *bitmap*, JPG, ou mesmo, um arquivo do AutoCad® (extensão “*.dwg”). É importante ressaltar que, uma vez que a imagem é inserida, ela só poderá ser redimensionada ou deslocada no plano de fundo: suas propriedades gráficas, como cores, por exemplo, não podem ser alteradas (o SIMUL8 não é um editor de imagens). Qualquer alteração na imagem deve ser feita previamente em um *software* específico;
- *Insert | Shapes*: insere uma forma básica (círculo, quadrado, linha, seta etc), com a opção de preenchimento (quando aplicável). Estes objetos podem ser reposicionados na tela e ter suas dimensões e outras propriedades alteradas (cor de borda, cor de fundo etc);
- *Insert | Text Box*: insere uma caixa de texto. Para esta operação o SIMUL8 exige que seja definida uma área retangular com o *mouse* (que será o tamanho da caixa de texto) antes da digitação do texto propriamente dito (não basta clicar na janela de simulação). As propriedades do texto (cor, estilo da fonte etc) podem ser alteradas após a inserção.

A Figura 6.2 ilustra estas 3 opções de objetos de fundo.

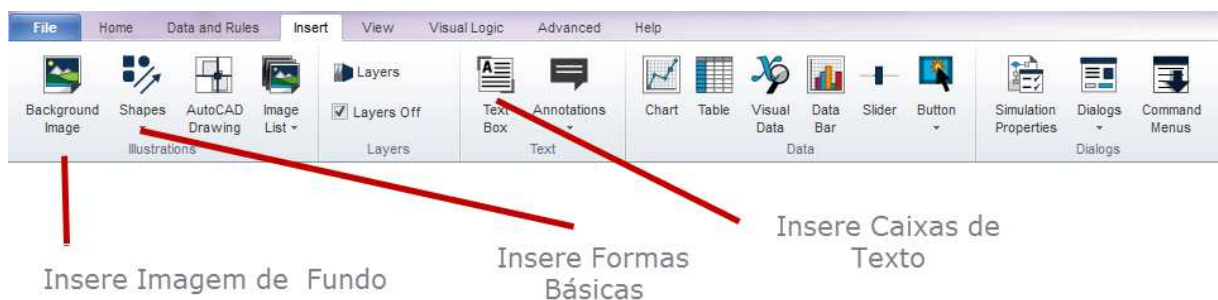


Figura 6.2 - Descrição das opções gráficas.

Como exercício, tente reproduzir o desenho mostrado na Figura 6.3 utilizando estas opções gráficas. Algumas dicas:

- Para alterar a cor do fundo (no nosso exemplo utilizamos a cor azul), clique com o botão direito do *mouse* em qualquer posição da tela e selecione a opção *Windows Properties*. Em seguida, selecione a opção *Select fill*

color e escolha a cor desejada. Para selecionar uma cor específica, a partir de seu código RGB, clique com o botão direito do *mouse* sobre a paleta de cores e selecione a opção *More Colors (Line)*²⁰;

- Observe que um dos quadrados possui preenchimento, enquanto o outro não. O mesmo vale para as circunferências;
- A figura é a capa de um livro, mas você pode escolher a imagem que quiser do seu próprio computador a partir de *Insert | Background Image*;

Seu desenho ficou parecido com o nosso ou melhor?

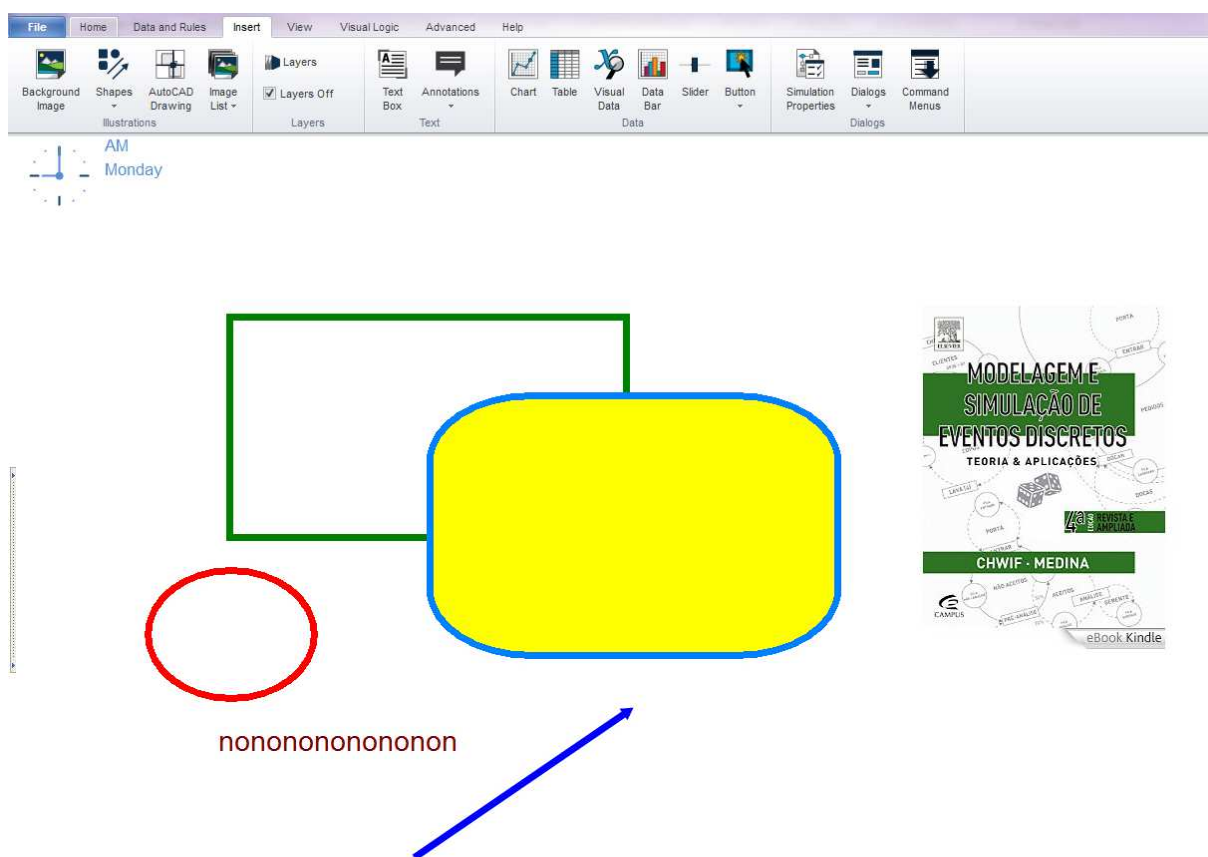


Figura 6.3 - Exemplo gráfico.

A partir da versão 2011 do SIMUL8 tornou-se possível inserir balões de textos ou anotações dinâmicas, clicando em *Insert | Annotations*. Deixaremos o leitor livre para explorar esta funcionalidade.

6.3. Objetos de simulação

Vamos tratar nesta seção da parte gráfica dos objetos de simulação (pontos de entrada, filas, atividades, pontos de saída etc). Para criar ou alterar um gráfico utilizamos o editor de imagens, disponível em *View | Image Library* ou através do atalho *Ctrl+Alt+I*. A interface do editor de imagens é mostrada na Figura 6.4

²⁰ Você pode exibir o código RGB das cores da paleta ao ativar a opção *Show full color values on status bar*. Se você não sabe do que estamos falando, não se preocupe: estamos fazendo este comentário para dar ferramentas àqueles que gostam de ter controle absoluto sobre a aparência do modelo de simulação.

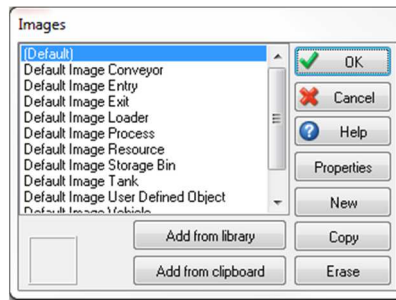


Figura 6.4 - Image Editor.

O editor de imagens permite, basicamente, três operações:

- *Add from library*: adiciona uma imagem da biblioteca de imagens do SIMUL8 ou de alguma pasta do computador. O SIMUL8 possui uma série de imagens disponíveis, de categorias como manufatura, pessoas, serviços, transportes etc²¹. Antes de criar uma imagem nova, sugerimos que você use esta opção para ganhar tempo – a menos que seja realmente necessário utilizar uma imagem personalizada;
- *Add from clipboard*: adiciona uma imagem que está na área de transferência do computador (resultado de um Ctrl+C);
- *New*: cria uma nova imagem. Neste caso, o SIMUL8 abrirá um editor de imagens, estilo “Paint”.

Não pretendemos abordar aqui as funcionalidades detalhadas desta ferramenta, já que este não é o propósito deste livro. Vale ressaltar que o tamanho da imagem padrão dos objetos do SIMUL8 é de 32×32 *pixels*. O editor de imagens pode se estender até 99×99 *pixels*, mas aconselhamos manter o formato padrão para não sobrecarregar graficamente o modelo, o que pode gerar resultados estranhos e/ou desproporcionais.

Após testar as funcionalidades do editor, tente construir uma imagem semelhante à que é mostrada na Figura 6.5 (seu tamanho é de 32×32 *pixels*). Nomeie a imagem como “Def_t”.

²¹ Desde que isto seja selecionado durante a instalação do programa.

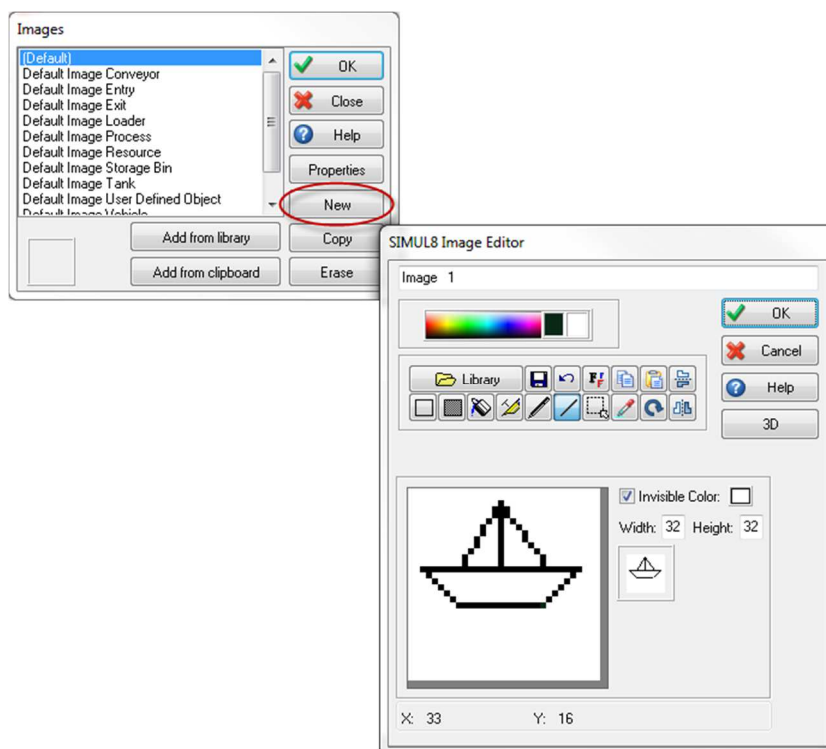


Figura 6.5 - Exemplo de uma imagem criada no editor do SIMUL8.

A imagem criada pode ser utilizada em qualquer objeto *deste modelo* de simulação, a partir das suas propriedades gráficas. Para utilizar a imagem em outros modelos é preciso salvá-la como um arquivo de imagem *bitmap*.

Uma nova e interessantíssima funcionalidade do SIMUL8 2012 (somente na versão Profissional) é a possibilidade de se importar modelos tridimensionais do *Google SketchUp*²² diretamente para o modelo de simulação e utilizar o módulo VR (*Virtual Reality*) para visualizar o modelo em 3-D. Isto é feito diretamente pelo *Image Editor* como mostra a Figura 6.6.

²² *Software* gratuito para construção e compartilhamento de modelos tridimensionais. Mais em: <http://sketchup.google.com/intl/pt-BR/>

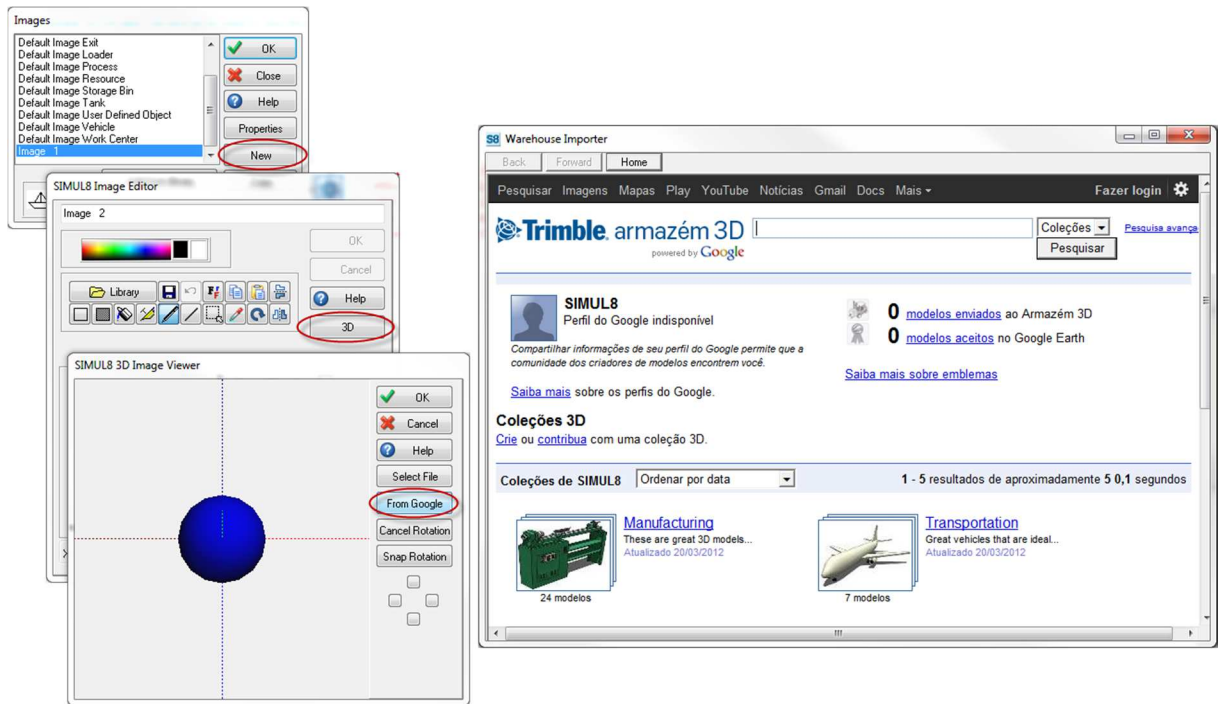


Figura 6.6 - Importação de modelos tridimensionais do Google SketchUp.

Por exemplo, nós selecionamos o modelo de um navio de transporte de gás “LNG Tanker” (Figura 6.7), mas você poderá escolher a figura que quiser. Ao final, você deve clicar no botão “Fazer download do modelo” na parte inferior da tela e confirmar ao SIMUL8 que deseja carregar o modelo para dentro da simulação (para os impacientes, o processo de importação demora alguns segundos).

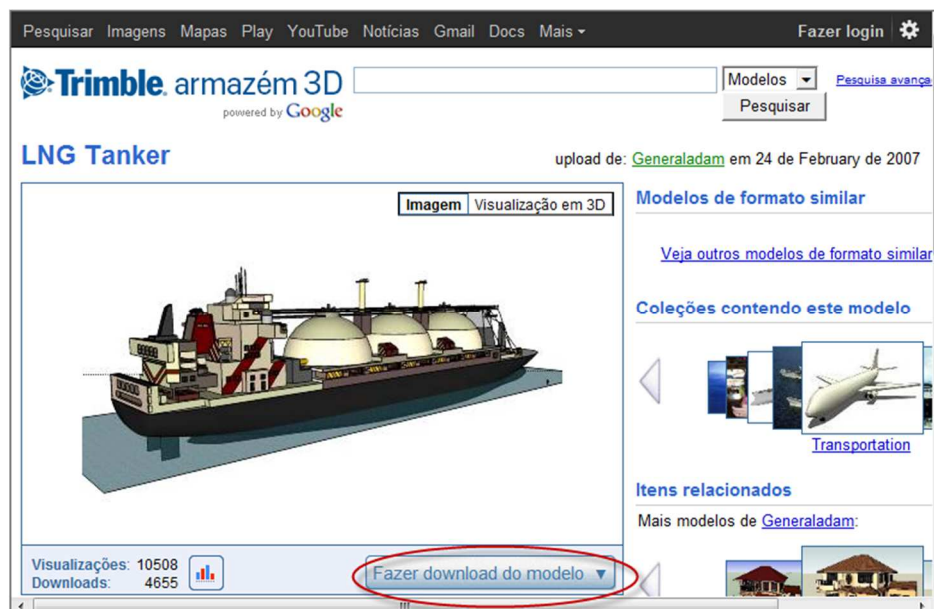


Figura 6.7 - Modelo tridimensional de um navio de transporte de gás.

Ao clicar com o botão esquerdo e movimentar o *mouse* no *Image Viewer*, você poderá posicionar a imagem na perspectiva que desejar, como mostra a Figura 6.8.

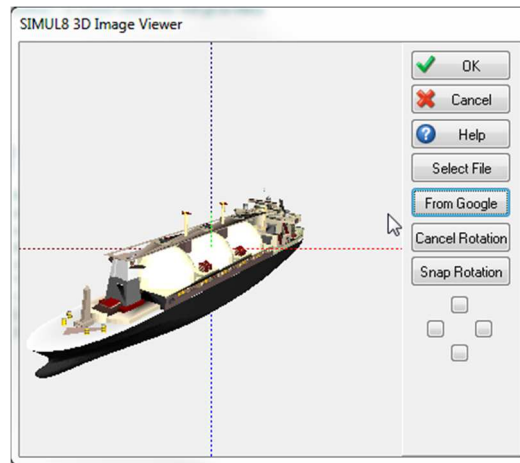


Figura 6.8 – Posicionamento da imagem com o mouse no *Image Viewer*.

Importante: a visualização em 3-D só estará disponível na versão Profissional e se for clicado o botão 3D da aba *View*.

Lembra-se do modelo do centro de distribuição do Capítulo 2? Para refrescar sua memória, mostramos novamente a estrutura no modelo na Figura 6.9.

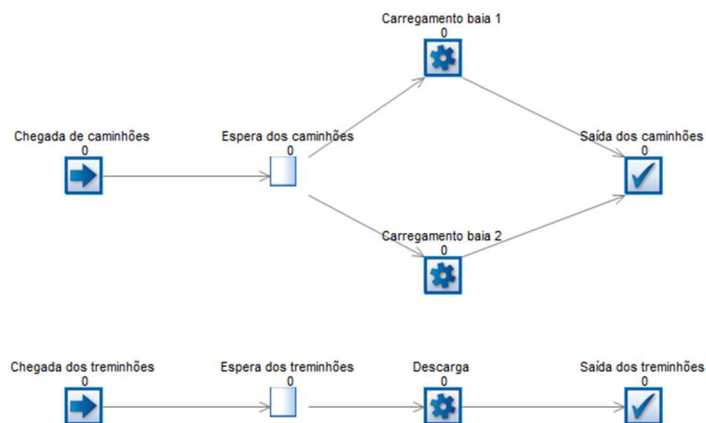


Figura 6.9 - Modelo do centro de distribuição.

Usando as técnicas de refinamento gráfico recém-discutidas, tente transformar o modelo da Figura 6.9 em outro mais “apresentável”, como mostrado no exemplo da Figura 6.10.

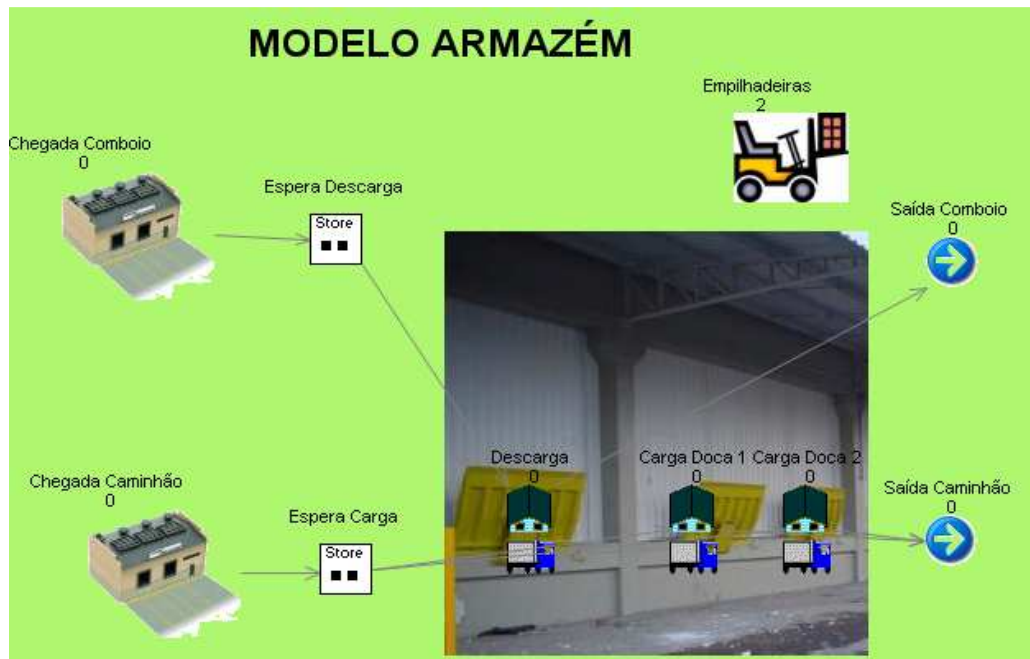


Figura 6.10 - Modelo do armazém com recursos gráficos adicionais.

Você (leitor) consegue gerar um modelo visualmente melhor que o nosso?

6.4. Entidades

No terceiro nível de refinamento gráfico, nosso foco estará nas entidades. Para associar uma imagem a uma entidade principal (*Work Item Type*), basta selecionar a opção *Advanced | Work Item Types* e clicar no botão *Image - Select*, como ilustrado na Figura 6.11. Este tipo de alteração é recomendável quando as “bolinhas vermelhas” exibidas por padrão pelo SIMUL8 não representam de forma clara as entidades reais do sistema.

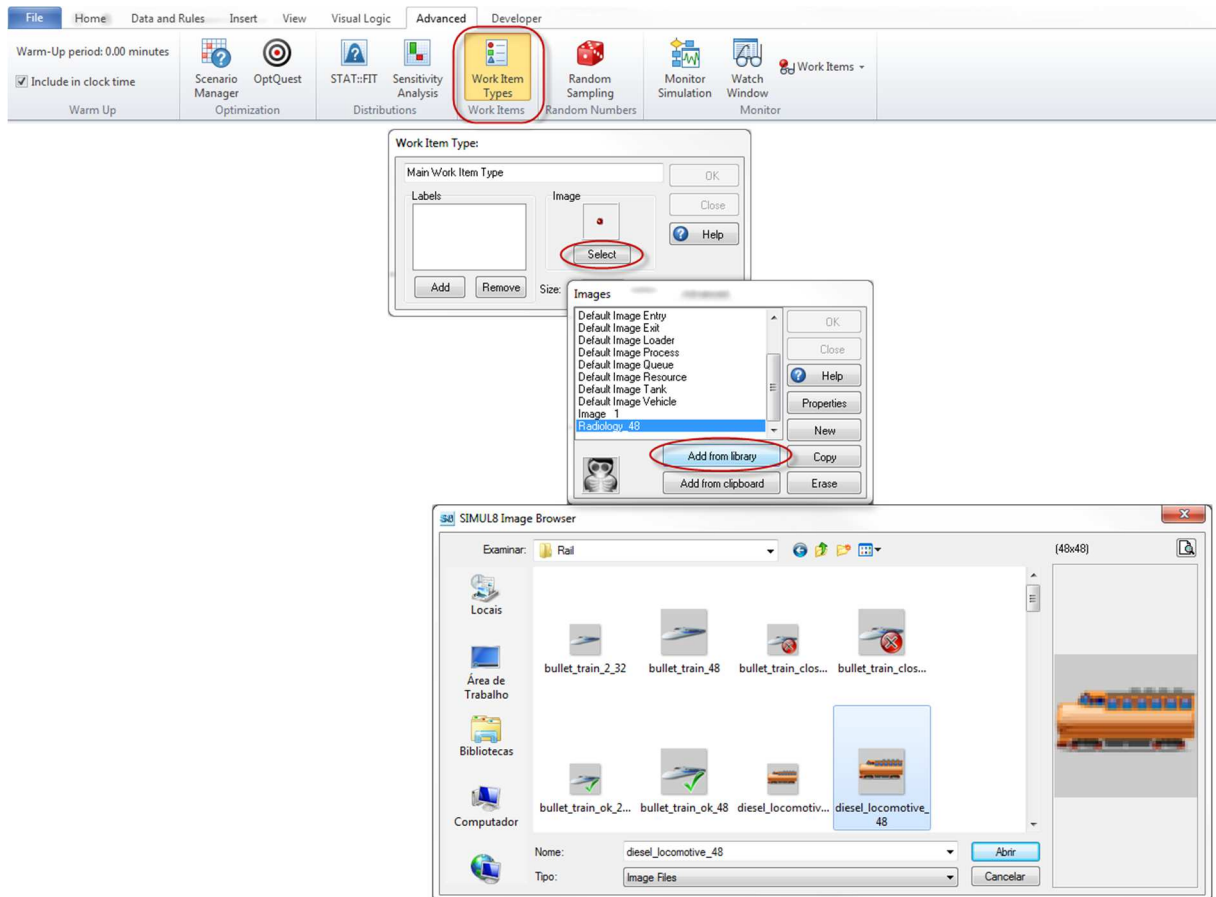


Figura 6.11 - Alteração de imagem da entidade.

Outra alteração que pode ajudar bastante a melhorar o aspecto de um modelo é a determinação de caminhos “não lineares”. Crie, por exemplo, um modelo com apenas uma chegada ligada a uma fila. Selecione a ligação entre a chegada e, em seguida, clique e arraste sobre algum ponto intermediário da ligação para adicionar um ponto de quebra. Este tipo de alteração é especialmente útil em modelos de transporte, quando queremos que a entidade siga (visualmente) um caminho pré-definido. A Figura 6.12 ilustra uma ligação com 3 pontos de quebra. Tente obter uma representação semelhante e simule o modelo para ver o efeito.

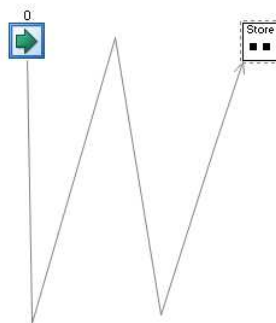


Figura 6.12 - Ligação com linhas de quebra.

Estamos considerando que, como de costume, os tempos de deslocamento do modelo foram zerados. Assim, o efeito da utilização de linhas de quebra é puramente visual. Na simulação do modelo mostrado na Figura 6.12 não haverá nenhum gasto de tempo no deslocamento das entidades entre o ponto de chegada e a fila, em termos de resultados da simulação.

Agora, abra a janela de propriedades da fila e clique no botão *Graphics*. Duas opções de visualização muito usadas são *Tank* e *Graphic Tank*, que mostram um efeito de um tanque se enchendo e esvaziando. Para o que queremos discutir no momento, selecione a opção de visualização *Queue*. Neste caso, as entidades em fila serão exibidas “uma atrás da outra”, como em uma fila indiana. Volte para a edição do modelo e ative o modo de visualização das ligações ( Show Routing Arrows). Ao selecionar a fila você verá dois pontos de manipulação, na forma de pequenos quadrados com preenchimento em preto: um no centro da fila e o outro um pouco deslocado para a esquerda. Clique neste segundo ponto e o arraste para regular a orientação e espaçamento das entidades e fila. Simule o modelo e veja o efeito da alteração (mostramos alguns exemplos na Figura 6.13).

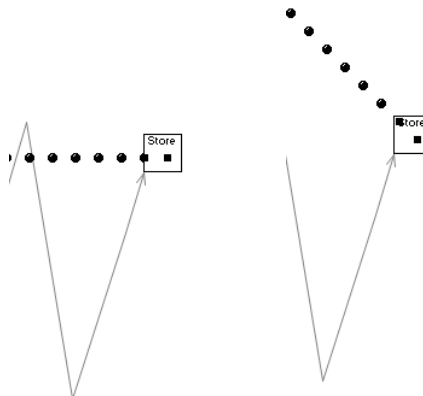


Figura 6.13 - Algumas opções de orientação e espaçamento de fila.

Agora vamos falar sobre alterações *dinâmicas* na imagem de uma entidade. Suponha que estamos construindo um modelo de uma padaria e queremos representar a entrada de massa no forno e a saída de pães assados. Para conseguir este efeito, basta acessar as propriedades gráficas do *Activity*, clicar no botão *Work Item Image on Exit* e selecionar a imagem desejada. Também é possível visualizar a entidade “dentro” do *Activity* durante seu processamento, selecionando a opção *Work Item*. Estas funcionalidades estão ilustradas na Figura 6.14²³.

²³ As imagens das empilhadeiras foram obtidas na pasta: *SIMUL8\Examples\Images\Manufacturing\Forklifta* e *Forklifib*.

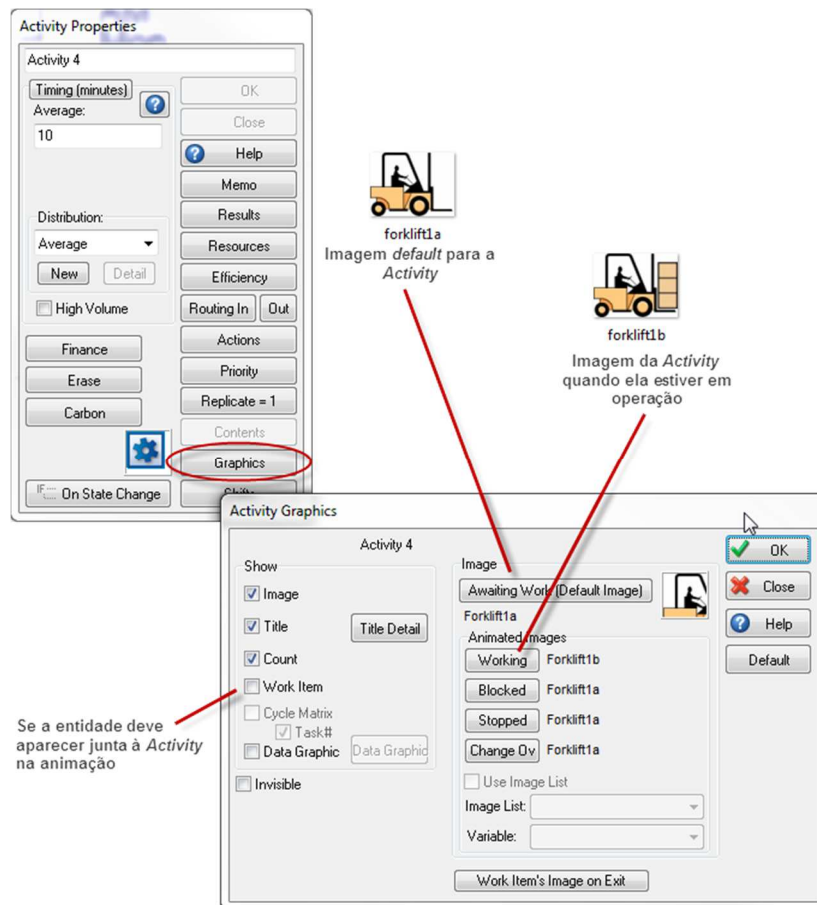


Figura 6.14 - Algumas opções de imagem de entidades em um Activity.

Agora suponha que você queira controlar a imagem da entidade através de um atributo (*label*). Há duas formas de se fazer isto. Uma é via lógica de programação (Capítulo 5), através de uma estrutura condicional IF e do comando *Set Work Item Image*. Outra forma, que não exige programação, é descrita a seguir:

- Crie um *label* numérico com o nome *image:xxxx*, onde xxxx pode ser qualquer palavra. Neste exemplo, por simplicidade, vamos criar um *label* com chamado *image:tipo*. A sintaxe “*image:*” é reservada e não pode ser alterada;
- Crie duas imagens distintas no editor de imagens (Ctrl+Alt+I) ou as importe da biblioteca de imagens (ou, ainda, de arquivos do tipo *bitmap*, extensão “.bmp”). Os nomes dessas imagens devem ser, necessariamente **tipo1** e **tipo2**. Não pode haver espaço entre a palavra “tipo” e o número que segue, pois este número é que fará a indexação automática entre o *label* e a imagem correspondente: Assim, se *label* = 1, então imagem = **tipo1**; se *label* = 2, então imagem = **tipo2**, e assim por diante (Figura 6.15);

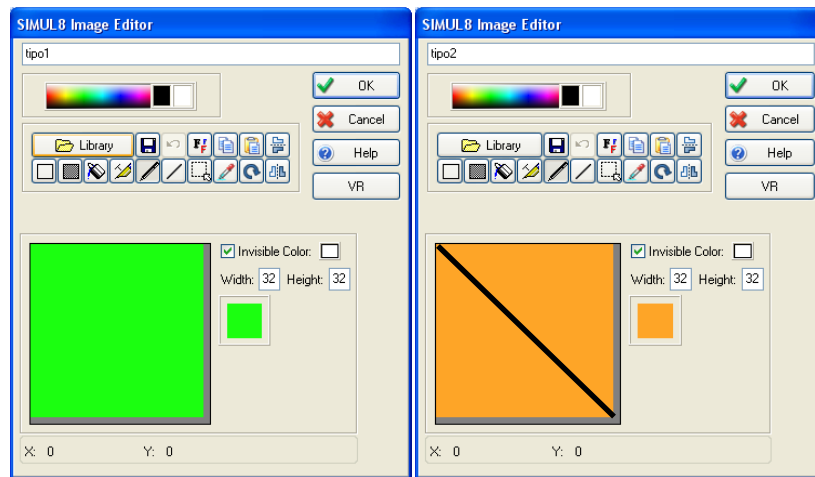


Figura 6.15 - Criação das imagens associadas aos *Labels*.

- Ajuste o valor do *label* (por exemplo, em um ponto de entrada) para uma distribuição *Rounded Uniform* entre 1 e 2, somente para efeitos de teste. Desta forma, haverá 50% de chance de que o *label* seja do tipo 1 e 50% de que seja do tipo 2. Rode o modelo. O resultado deverá ser semelhante ao da Figura 6.16.

Note que este procedimento pode ser estendido para qualquer quantidade de imagens.

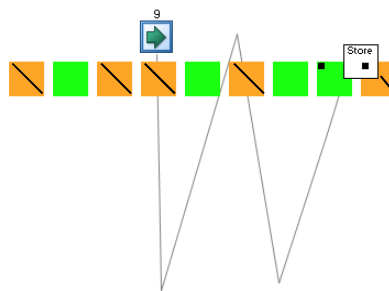


Figura 6.16 - Utilização de imagens selecionadas por *label*.

Para finalizar este capítulo, seguem algumas dicas para outras funcionalidades gráficas que julgamos interessantes:

- Para visualizar os recursos “trabalhando” nos *Activities*, ao associá-los, utilize a opção *Resources | Display | Display at Activity* (é uma das opções de propriedades gráficas dos *Activities*);
- Para que um *Activity* apresente imagens diferentes, uma para cada condição (ocioso, trabalhando, quebrado, em *set-up*), abra as propriedades gráficas do *Activity* e selecione a imagem correspondente a cada estado. Este recurso só funciona se a propriedade *Replicate* do *Activity* for igual a 1;
- Você pode visualizar gráficos na tela, dando um duplo-clique nos gráficos e selecionando a opção *Onscreen* (somente na versão Professional);
- O SIMUL8 permite a construção de gráficos de barras dinâmicos e tabelas dinâmicas. Estas funcionalidades não serão abordadas aqui, mas podem ser acessadas a partir de *Insert | Table* ou *Insert | Data Bar*;
- Imagens podem ser associadas a variáveis, isto é, dependendo do valor de uma variável a imagem de um objeto pode mudar. Esta funcionalidade se chama *Image List* e pode ser acessada clicando-se em *Insert | Image List*. Também não abordaremos esta funcionalidade, por entendermos que se trata de um recurso avançado.

Capítulo 7 FUNCIONALIDADES ESPECÍFICAS

7.1. Introdução

Este capítulo visa aprofundar alguns conceitos desenvolvidos nos capítulos anteriores, utilizando funcionalidades específicas de *Activities*, filas e recursos. Pretendemos, ainda, que o leitor se aprofunde na utilização de *labels* e apresentaremos um novo objeto de simulação: a esteira (*conveyor*).

Desenvolveremos os tópicos utilizando um exemplo de um processo de triagem e embalagens de frutas. Por razões didáticas, explicaremos os detalhes de funcionamento do modelo à medida que apresentarmos os novos conceitos.

7.2. Exemplo inicial

Considere um processo aonde frutas chegam em caixas de 100 unidades com intervalo entre chegadas sucessivas de 90 minutos, segundo uma distribuição exponencial. Estas frutas são selecionadas conforme padrões de qualidade, o que leva, em média, 30 segundos por fruta. Após a seleção, as frutas podem ter três destinos, de acordo com sua qualidade:

- Qualidade A: serão embaladas para exportação, o que leva 5 minutos por caixa com 12 frutas;
- Qualidade B: serão processadas para a fabricação de suco, em um tempo de 30 minutos por batelada de 100 frutas;
- Qualidade C: serão descartadas, ou melhor, segundo critérios de sustentabilidade, serão trituradas (o que leva 40 minutos por batelada de 50 frutas) e utilizadas como matéria-prima para a fabricação de adubo orgânico.

O modelo que representa o processo está ilustrado na Figura 7.1. Os principais detalhes sobre a construção serão explicados na próxima seção. Note que foram inseridos três textos estáticos (“Qualidade A”, “Qualidade B” e “Qualidade C”).

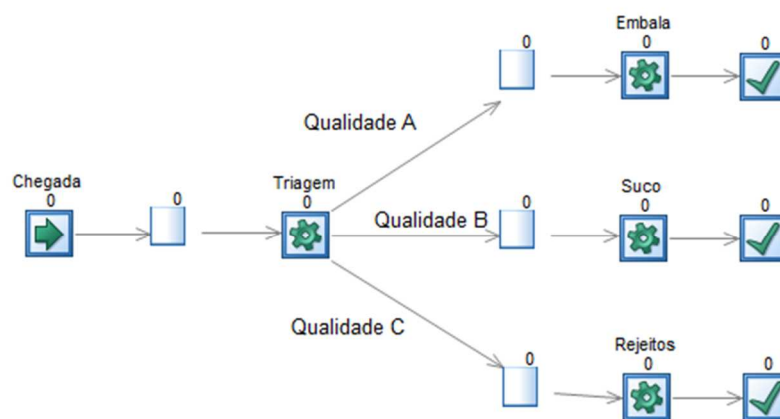


Figura 7.1 - Representação do processo de separação e processamento de frutas.

7.3. Formação de lotes ou bateladas

Para introduzir a informação sobre as bateladas (já que as frutas serão sempre processadas em lotes) utilize a disciplina de entrada (*Routing In*) *Collect* em cada *Activity*. A Figura 7.2 mostra a configuração que deve ser feita no *Activity* que representa o processo de embalagem, de 12 em 12 unidades. Na figura, note que clicamos no botão *More* para exibir

todas as opções disponíveis. Note, ainda, que selecionamos a opção *Do not collect until all available* para indicar que as frutas devem ficar na fila até que exista a quantidade necessária para compor uma caixa.

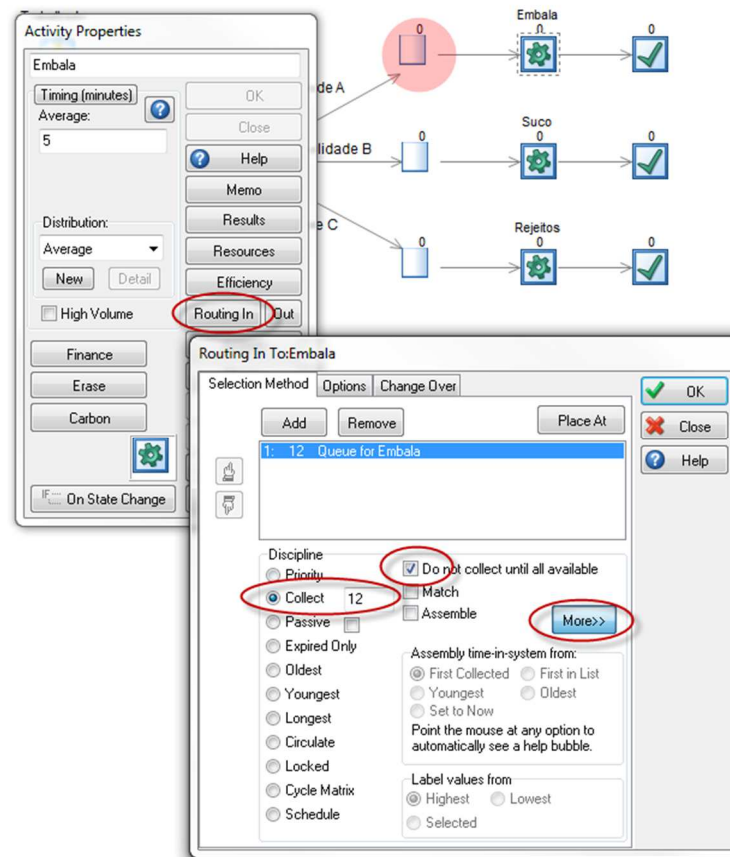


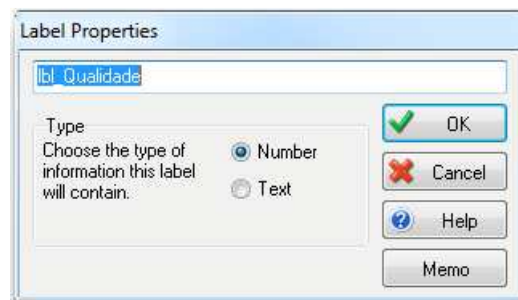
Figura 7.2 - Routing In – Bateladas.

Por fim, observe que as opções *Match* e *Assemble* deverão estar *desativadas*. Apesar de o processo gerar uma caixa a partir de 12 frutas não ativamos a opção *Assemble* porque queremos que todos os resultados sejam expressos em termos de frutas. Se tivéssemos ativado a opção *Match*, todas as frutas coletadas deveriam ser da mesma qualidade (representada por um *label*), algo que usaremos (e explicaremos) na próxima seção.

As configurações do direcionamento de entrada (*Routing In*) dos processos “Suco” e “Rejeitos” devem ser ajustadas de forma semelhante. Neste ponto, o modelo ainda não está pronto para ser executado, pois falta definir o processo de separação das frutas segundo a qualidade. Para isso usaremos *labels*, como explicado a seguir.

7.4. Criando *labels*

Como visto no Capítulo 2, *labels* (ou atributos) são informações que podem ser associadas às entidades de um modelo de simulação. Para este exemplo, vamos criar um *label* chamado “lbl_Qualidade” para representar a qualidade das frutas processadas. Para isso, basta clicar em *Data and Rules | Labels* e, na janela que será aberta, clicar no botão *New* e definir o nome e o tipo de *label* desejado (Figura 7.3 e Figura 7.4).

Figura 7.3 - Lista de *labels* utilizados na simulação.Figura 7.4 - Propriedades do *label*.

Como comentamos no Capítulo 2, mesmo quando a informação armazenada em um *label* tem uma interpretação textual (como “Qualidade A”), é preferível criar uma relação entre esta informação e um valor numérico do que utilizar um *label* de texto. Para o nosso exemplo, esta relação é descrita na Tabela 7.1.

Tabela 7.1 - Valores dos *labels* a serem usados no modelo.

Qualidade (tipo)	Lbl_Qualidade
Qualidade A	1
Qualidade B	2
Qualidade C	3

7.5. Utilizando os *labels* (*Label Actions*)

O atributo de qualidade das frutas será definido no processo de triagem. Portanto, abra a janela de propriedades do *Activity* “Triagem” e clique no botão *Actions*²⁴. Clique em *Add* e selecione o *label* “lbl_Qualidade”. Como queremos atribuir um valor para o *label*, selecione a opção *Set to* (Figura 7.5). Observe que o valor atribuído ao *label* será definido por uma distribuição de probabilidades discreta (voltaremos a discutir isso em breve).

²⁴ Este botão também pode ser encontrado em pontos de entrada, filas e esteiras (*conveyors*).

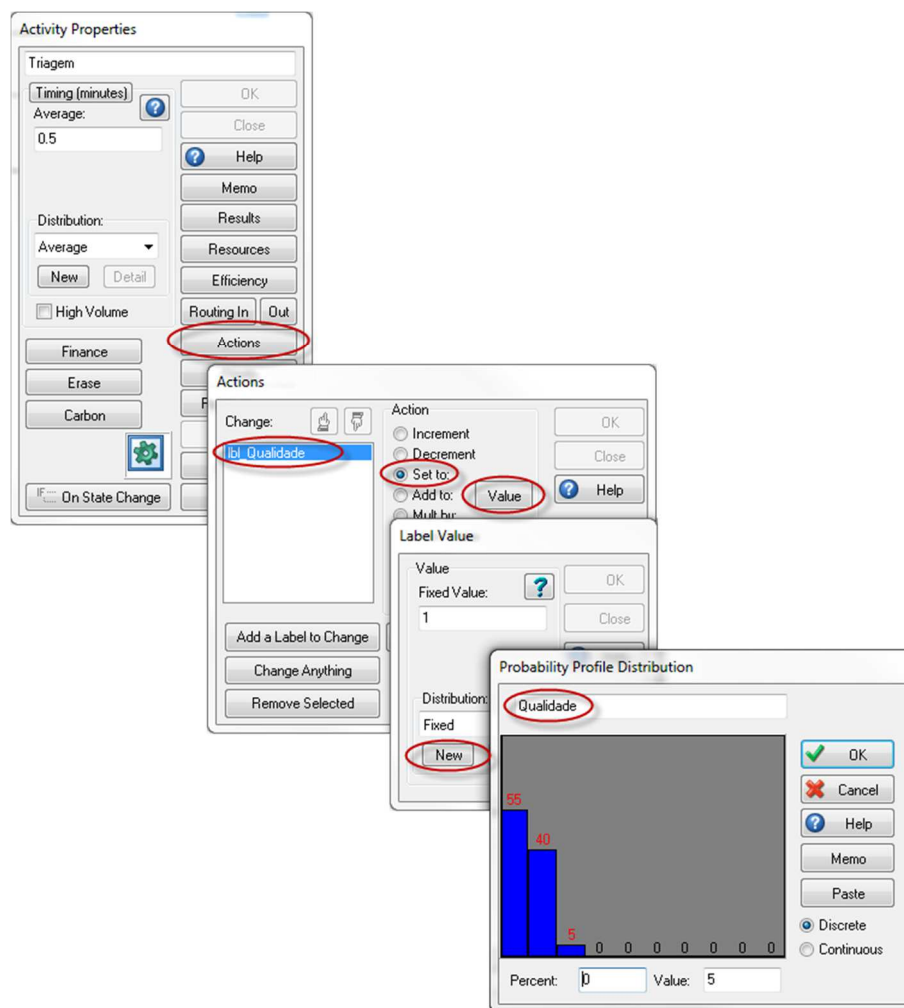


Figura 7.5 - Ações possíveis para o label "lbl_Qualidade".

A seguir, descreveremos as opções de ações possíveis.

- *Increment*: Adiciona 1 ao valor do *label*, o que é muito usado para se contar quantas vezes uma entidade passou por um ponto do modelo;
- *Decrement*: Similar ao anterior, subtraindo 1 do valor do *label*. Ação utilizada, por exemplo, para se contar quantas tarefas restam para a finalização de um processo;
- *Set to*: Utilizado para atribuir determinado valor ao *label*, a partir de uma distribuição de probabilidades ou um valor fixo. Se for utilizado em um ponto da simulação em que a entidade passa mais de uma vez, a nova atribuição sempre irá se sobrepor à anterior.
- *Add to*: Adiciona um determinado valor ao valor atual do *label*;
- *Multiply by*: Multiplica o valor atual do *label* por um valor;
- *Time Stamp*: Atribui ao *label* o valor atual do relógio de simulação. Esta ação é equivalente a atribuir ao *label* o valor da variável de sistema *Simulation Time* (o que exige o uso de *Visual Logic*);
- *Unique*: Atribui um valor numérico único para a entidade que receber o *label* (como ocorre, por exemplo, com a numeração das notas de dinheiro);
- *Ask VB*: Executa uma macro em Visual Basic para decidir o que fazer com o *label*. Esta ação deve ser evitada, pois acionar programas externos ao SIMUL8 aumenta o tempo de simulação.

Como comentamos anteriormente, utilizaremos a opção *Set to* no nosso exemplo, sendo que o valor será definido por uma distribuição chamada “Qualidade”, do tipo *Probability Profile*, com os valores dados na Tabela 7.2:

Tabela 7.2 - Distribuição de probabilidades da qualidade.

Valor	Probabilidade
1	55%
2	40%
3	5%

Os valores do *label* “lbl_Qualidade” serão usados pelo próprio *Activity* “Triagem” para programar a política de seleção das frutas. Acesse as propriedades do *Activity* “Triagem” e, nas opções de direcionamento de saída, escolha a disciplina *label*, selecionando o *label* “lbl_Qualidade”. A sua janela de propriedades deve se parecer com a que é mostrada na Figura 7.6 – preste muita atenção à ordem dos destinos de saída!

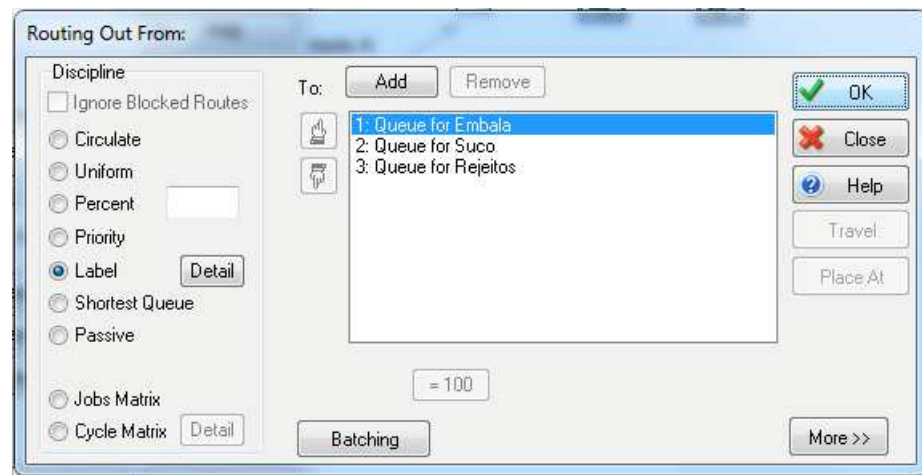


Figura 7.6 - Direcionamento de saída segundo um *label*.

Agora o modelo está pronto para ser executado pela primeira vez.

7.6. Disponibilidade de recursos e eficiência dos *Activities*

Nesta seção, mostraremos como representar situações em que os recursos não estão disponíveis durante todo o período de simulação.

Crie um recurso chamado “Trabalhador” (vide Capítulo 2) para representar a existência de 3 trabalhadores e o associe a cada um dos *Activities*. Em seguida, abra a janela de propriedades do recurso e clique no botão *Availability*. Na janela que se abrirá (Figura 7.7), ajuste os parâmetros da disponibilidade.

- Se a opção *Auto* estiver selecionada, somente duas informações são necessárias: a disponibilidade (*Availability*), em termos de porcentagem do tempo total de simulação e o tempo médio de cada parada (*Average Absence Time*). Se colocarmos um ponto de interrogação “?” neste valor, o SIMUL8 calculará automaticamente a duração de cada parada²⁵;

²⁵ A duração de cada parada seguirá uma distribuição de Erlang.

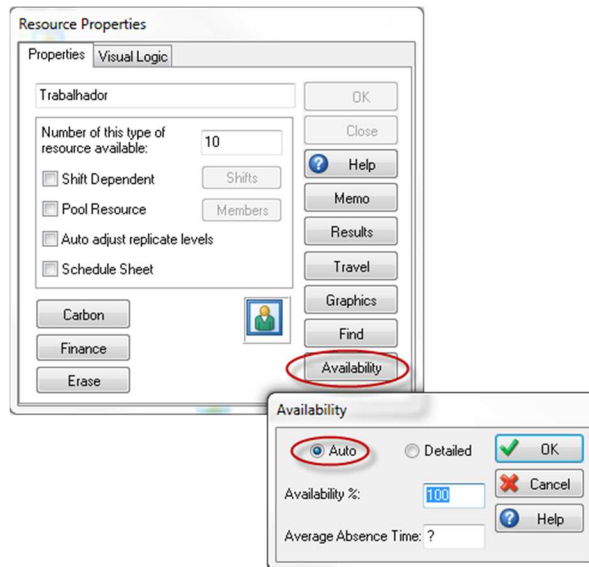


Figura 7.7 - Disponibilidade de um recurso.

- Se a opção *Detailed* for selecionada, a janela de propriedades se expande (Figura 7.8), exibindo opções para se representar mais de um tipo de parada, como por exemplo, uma parada para almoço (com duração de 1 hora) e uma parada para lanche (que dura 15 minutos). Clicando no botão *Time Between Absence*, podemos determinar de quanto em quanto tempo a parada acontece. Já o botão *Time to Return* define quanto tempo se passa até que o recurso volte à atividade.

Para o nosso exemplo, defina dois tipos de paradas. A primeira será chamada de “Parada Curta” e ocorrerá a cada 120 minutos segundo uma distribuição exponencial, com duração fixa de 10 minutos. O segundo tipo, a “Parada Longa”, representará pausas de 30 minutos aos quais os trabalhadores têm direito a cada 6 horas (360 minutos) – não queremos ter problemas com o sindicato dos separadores de frutas, certo?

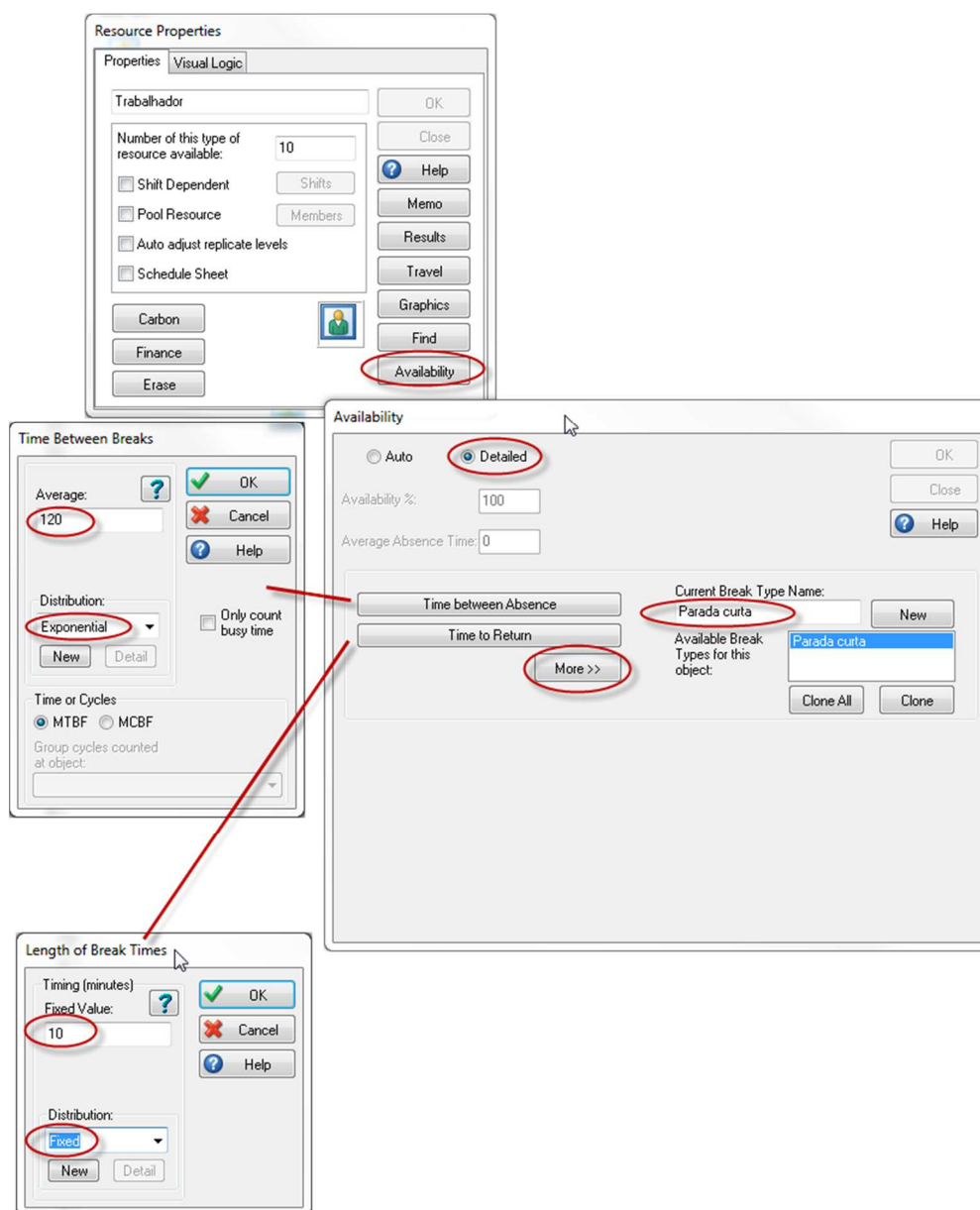


Figura 7.8 - Detalhamento da disponibilidade de um recurso.

Os objetos *Activity* podem ser programados para efetuar paradas programadas ou aleatórias. A configuração das paradas dos *Activities* funciona exatamente como a configuração da disponibilidade dos recursos, com algumas opções adicionais. A configuração das paradas é acessada pelo botão *Efficiency* da janela de propriedades do *Activity*. Como se pode observar no exemplo mostrado na Figura 7.9, a janela de parâmetros é muito semelhante à janela de disponibilidade dos recursos. A diferença está na seção *When a break-down occurs*, em que temos as seguintes opções:

- *Close feeding queues*: se ativada, esta opção faz com que a(s) fila(s) da entrada deste *Activity* não recebam mais itens até que ele volte a funcionar normalmente;
- *Stop work immediatly*: se ativada, interrompe o funcionamento do *Activity* assim que a parada ocorre (opção ativada por padrão);
- *Extend breakdown by time to empty the queue*: esta opção só estará disponível se a opção *Stop work immediatly* estiver desativada. Ela permite o processamento dos itens que já estavam na fila antes da parada acontecer.

Figura 7.9 – Parâmetros de eficiência de um Activity.

Outra diferença está na janela onde definimos o tempo de parada (*Length of Break Times*), acionada pelo botão *Time to Repair* (Figura 7.10): aqui temos a opção *Resources*, onde definimos quais recursos serão usados *durante* a parada (recursos de conserto).

Crie uma parada “Tipo Pane” (parada imediata) para o Activity “Suco”, com os seguintes parâmetros: tempo entre paradas distribuído exponencialmente com média de 200 minutos e tempo de reparo segundo uma distribuição Erlang(10, 3).

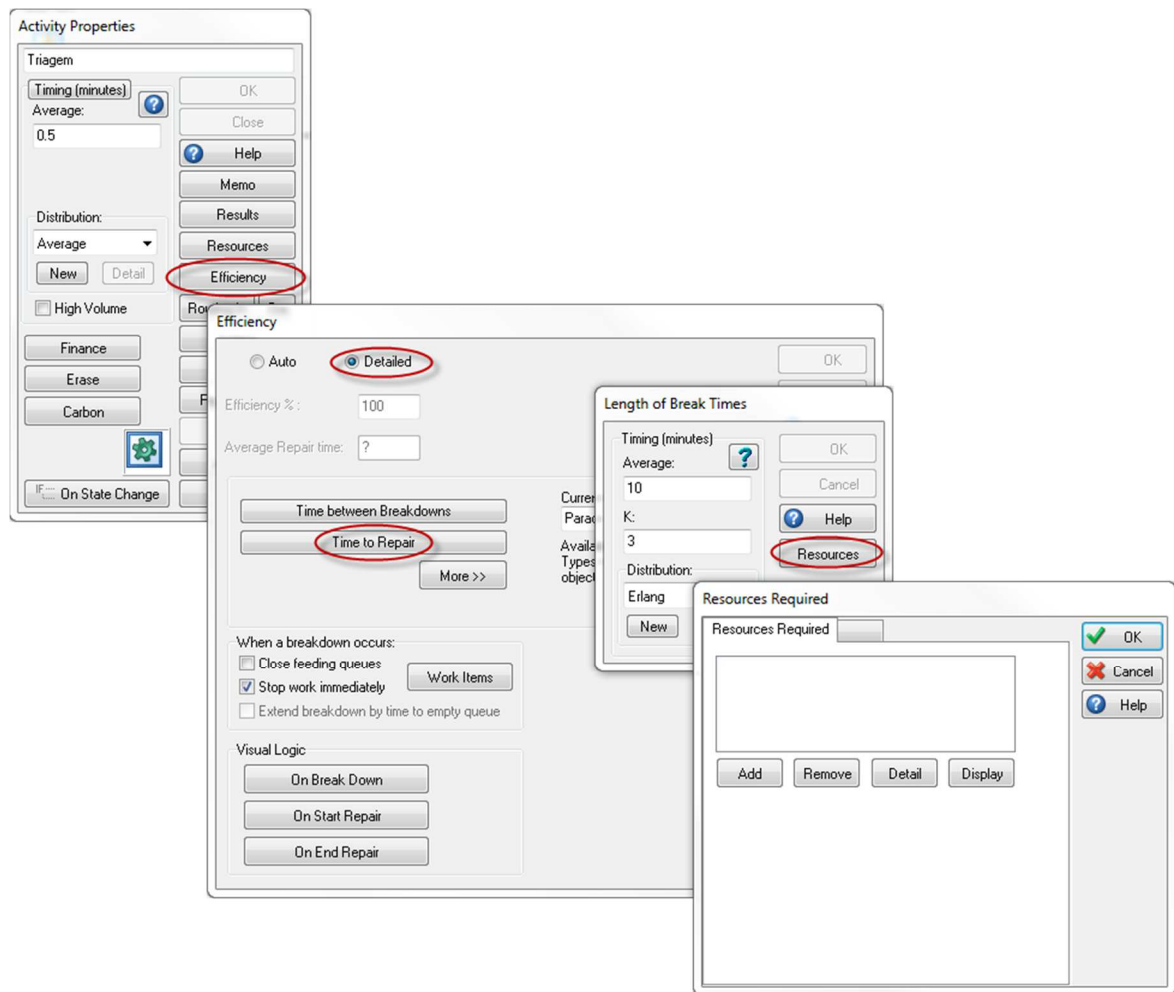


Figura 7.10 - Recursos utilizados durante a parada.

7.7. Funcionalidades de filas

As filas do SIMUL8 possuem algumas propriedades que podem ser úteis para a modelagem de situações reais. Algumas delas são: priorização de entidades, capacidade máxima (por exemplo, para quando as filas simulam a estocagem de produtos), tempo mínimo de espera, tempo máximo de espera etc. Nesta seção, vamos ilustrar essas funcionalidades em uma versão levemente modificada do nosso exemplo.

Imagine que o nosso setor de processamento das frutas comece a receber mais de um tipo de fruta, como peras e maçãs. Imagine também que as peras, por serem mais delicadas e caras do que as maçãs, têm prioridade no processo de separação e preparação do suco. Para diferenciá-las, vamos utilizar um novo *label*, chamado “lbl_fruta”, com a seguinte codificação numérica: 1 para peras e 2 para maçãs.

Vamos considerar que a cada 100 caixas de frutas que chegam ao setor, 45 são de maçãs e o restante de peras. Ajuste a seção *Label Actions* do ponto de entrada para simular isso (sim, você precisará criar uma nova distribuição).

Agora, as novidades: a Figura 7.11 mostra as propriedades da fila que antecede o *Activity* “Triagem”, com as configurações que serão usadas no nosso exemplo. Antes de alterar o seu modelo, acompanhe uma descrição de cada opção:

- *Capacity*: indica a capacidade máxima da fila. Se alguma entidade tentar entrar em uma fila cheia ela simplesmente será descartada e contabilizada como perdida (*Item lost*). A opção *Infinity* (padrão) indica que a fila tem capacidade ilimitada;

- *Shelf Life*: representa o tempo que cada item pode aguardar na fila antes de ser considerado expirado. Entidades expiradas só podem ser removidas da fila por um *Activity* com disciplina de entrada do tipo *Expired only*, como visto no Capítulo 4;
- *Min Wait Time*: tempo mínimo de espera em fila, para representar, por exemplo, um tempo de secagem de uma peça pintada, antes do próximo processamento. No nosso exemplo (Figura 7.11), nenhuma fruta ficará menos do que 15 minutos em fila;
- *Prioritize*: estabelece prioridade de atendimento segundo um *label*, do maior valor para o menor: ao entrar na fila, entidades com maior valor de *label* “furam” parte da fila. Se for desejada uma ordem de prioridade inversa, deve-se ativar a opção *Reverse Priority*;
- *LIFO*: esta opção transforma a fila em uma *pilha*, ou seja, a última entidade a entrar na fila será a primeira a sair. Esta opção só ficará disponível se a opção *Prioritize* estiver desativada – por este motivo ela não aparece na Figura 7.11;
- *Segregate Results*: caso entidades com diferentes *labels* passem por essa fila, pode ser interessante conhecermos os resultados de alguns indicadores (como tempo médio de espera em fila) para cada valor de *label* – esta opção força o SIMUL8 a fazer exatamente isso. Por exemplo, se quisermos saber os valores do tempo médio de espera das peras e das maçãs, devemos segregar os resultados pelo *label* “lbl_fruta”.

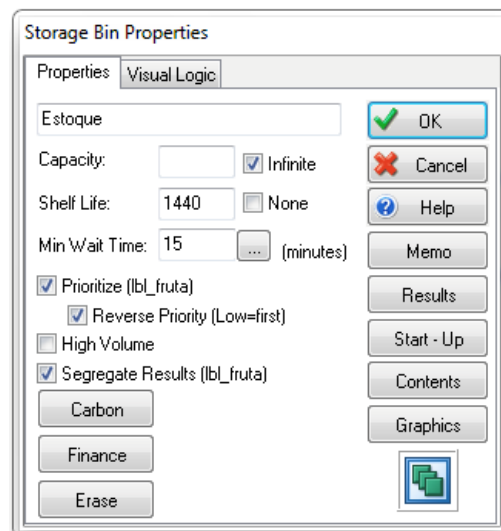


Figura 7.11 - Propriedades das filas.

Ajuste os parâmetros da fila antes da triagem como mostrado na Figura 7.11. Para as demais filas, utilize capacidade infinita e sem definição de *Shelf Life*. Para as filas que antecedem os processos de embalagem e preparação de suco, priorize as peras e faça com que os resultados sejam separados por tipo de fruta. Os resultados da fila de frutas rejeitadas também devem ser desmembrados por tipo de fruta, mas não é necessário nenhum tipo de priorização, pois todas as frutas que passarem por esta fila irão para a compostagem.

Por fim (e isso é muito importante), como nenhuma caixa ou suco pode conter tipos diferentes de frutas, ative a opção *Match* dos *Activities* “Embala” e “Suco”, selecionando o *label* “lbl_fruta”.

7.8. Set-up (Activity Change Over)

Muitas vezes, uma linha de produção produz diferentes produtos e é preciso gastar algum tempo para preparar a linha quando há troca de produtos (tempo de *set-up*). Por exemplo, imagine uma linha de produção de remédios, que produza um analgésico genérico, mas que também produza um antibiótico. Após o final da produção do lote de antibióticos é essencial que as máquinas sejam lavadas para que nenhum resíduo de antibiótico contamine os comprimidos de analgésico. O tempo gasto na lavagem das máquinas equivale ao tempo de *set-up*.

No SIMUL8, podemos introduzir tempos de *set-up* na janela de propriedades dos *Activities*, clicando em *Routing In* e selecionando a aba *Change Over*, como ilustra a Figura 7.12. Do lado esquerdo da janela encontramos os campos em que se define o tempo de *set-up*, que pode ser constante ou dado por uma distribuição de probabilidades. No lado direito da janela, temos as seguintes opções:

- *No Setup Time*: não é considerado tempo de *set-up*;
- *Every Nth Work Item*: o *set-up* ocorre a cada *N* entidades processadas;
- *When Label Changes*: o *set-up* ocorre a cada mudança de *label* das entidades processadas;
- *After Working Time*: o *set-up* ocorre após um tempo determinado.

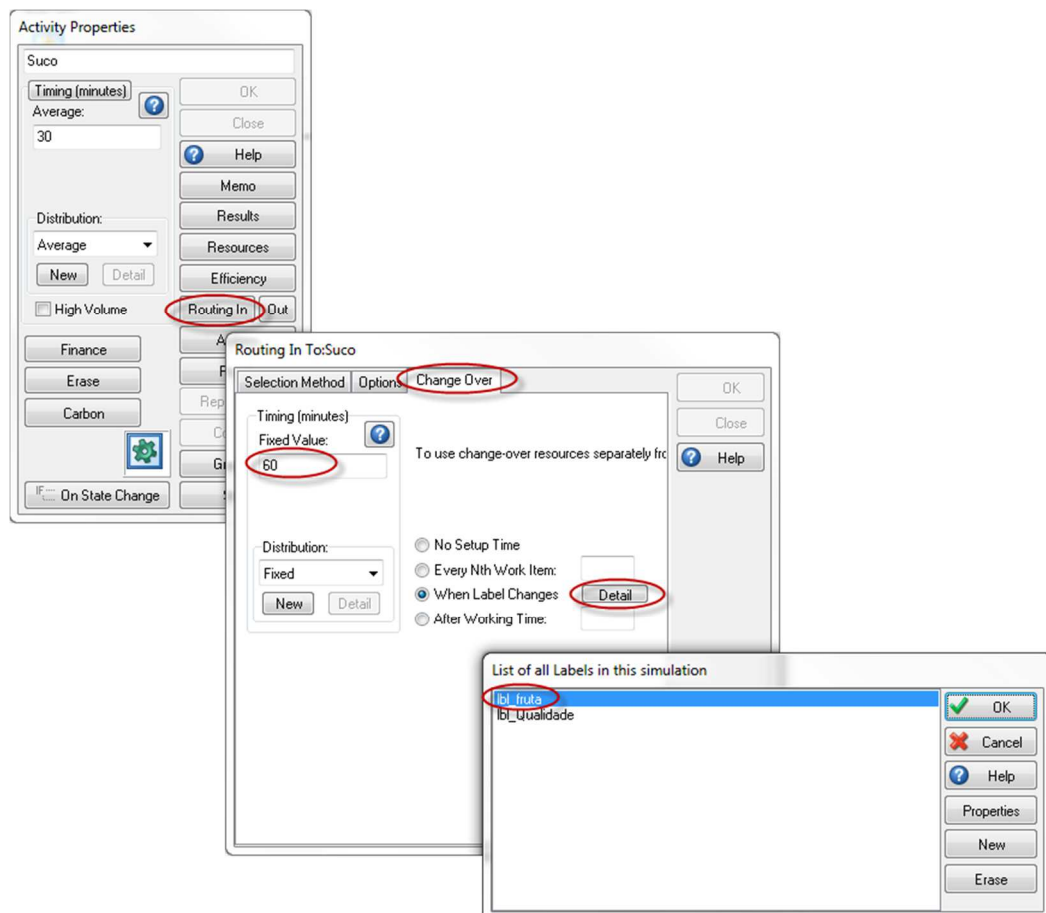


Figura 7.12 - Definição de *set-up*.

No nosso exemplo, não queremos que existam traços de maçã no suco de pera ou vice-versa. Portanto, ajuste os parâmetros de *set-up* do *Activity* “Suco” para considerar que haja uma limpeza a cada mudança de fruta (*label* “*lbl_fruta*”), com tempo total de 60 minutos.

7.9. Conveyor (somente no SIMUL8 Professional)

O *Conveyor* (ou esteira) é um objeto do SIMUL8 usado para simular o transporte de entidades (*Work Items*). A vantagem em se utilizar um *conveyor* ao invés de um *Activity* é que ele não somente computa o tempo de transporte, mas possui uma capacidade de movimentação calculada em função das suas dimensões físicas e das dimensões do item a ser movimentado. Uma vez que o *conveyor* esteja lotado, o transporte será interrompido, algo que (usualmente) acontece na realidade.

Objetos do tipo *Conveyor*  podem ser encontrados na barra de componentes *Advanced Building Blocks*, que deve estar na parte inferior da barra de componentes do SIMUL8 (Figura 7.13). A Figura 7.14 mostra as propriedades do objeto *Conveyor*. Na parte superior da janela temos os campos de nome do objeto, o comprimento e a velocidade, em função das unidades definidas como padrão nas propriedades da simulação. Como comentado anteriormente, o número máximo de entidades que o *Conveyor* pode conter depende do seu tamanho e do tamanho da entidade. O tamanho padrão das entidades é de 1 m, mas isto pode ser alterado em *Advanced | Work Items | All Work Items | Wl Type Info*. O tamanho em tela do *Conveyor* não é afetado por estas configurações, mas você pode alterar a imagem do objeto caso queira uma animação mais realista.

Se a opção *Accumulating* for ativada, as entidades transportadas continuam o seu movimento mesmo que exista um item no final da esteira bloqueando a saída, ou seja, os itens vão se acumulando até que o *Conveyor* esteja cheio. Caso esta opção esteja desativada, o *Conveyor* só se movimenta se a saída estiver liberada, como acontece nas esteiras dos caixas de supermercado.

Enquanto a opção *Accumulating* detalha as ações do *Conveyor* em relação à saída dos itens, a opção *Allow Gaps* se refere à sua entrada: se a opção *Allow Gaps* estiver ativada, estaremos simulando uma esteira em movimento contínuo, ou seja, um item pode estar separado do outro. Com a opção desativada, estaremos simulando uma esteira que só se movimenta se os itens estiverem “colados” uns aos outros.

Não discutiremos a configuração da área de retirada de itens (*Pick area length*), por se tratar de uma funcionalidade avançada.

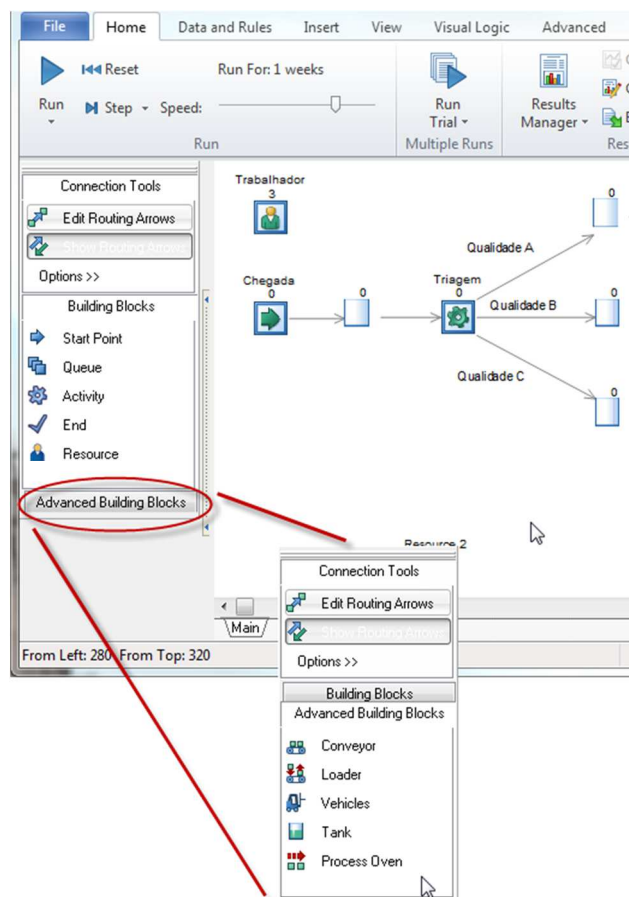


Figura 7.13 - *Advanced Building Blocks*.

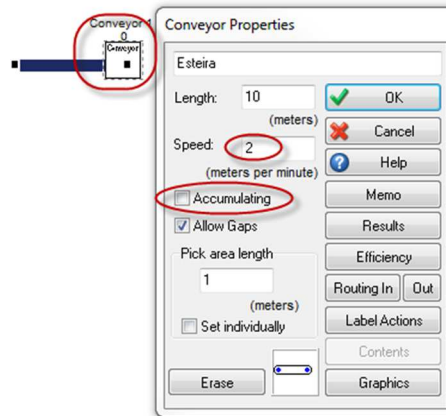


Figura 7.14 - Propriedades do *Conveyor*.

Por fim, os objetos do tipo *Conveyor* também podem estar sujeitos a paradas para manutenção, tais como os *Activities*, por meio da opção *Efficiency* (Figura 7.14).

Voltando ao nosso exemplo do processamento de frutas, imagine que as frutas de Qualidade A são transportadas por uma esteira, da triagem para a embalagem. Considere, ainda, que esta esteira tem 10 metros de comprimento e funciona a uma velocidade de 2 metros por minuto, desde que a saída esteja liberada. Altere o modelo para representar esta nova situação (Figura 7.15), considerando que cada fruta ocupe um espaço de 10 cm na esteira (a partir de *Advanced | Work Items*). Rode em modelo com uma velocidade de animação que permita visualizar o funcionamento da esteira.

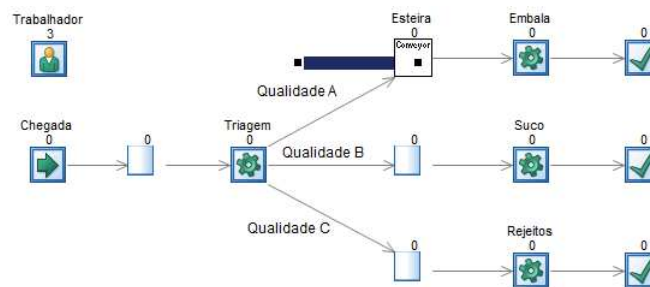


Figura 7.15 - Separação e processamento de frutas.

Quando uma ligação entre dois objetos é apagada e refeita, ela entra no final da lista de destinos de saída (*Routing Out*). Assim, se você criou o modelo com a esteira a partir do modelo discutido nas seções anteriores deste capítulo, verifique se o direcionamento das frutas após o processo triagem está sendo feito de forma correta (sequência de destinos: embalagem, fabricação de suco e descarte).

Capítulo 8 ESTUDOS DE CASO

8.1. Introdução

Neste capítulo, apresentaremos dois casos práticos baseados em projetos de consultoria realizados pelos autores. O primeiro é da área de manufatura, enquanto o segundo é da área de serviços, sobre o processo de suporte ao sistema ERP (*Enterprise Resource Planning*). Para cada caso teremos:

- Descrição do problema;
- Criação de modelo de simulação;
- Atividades complementares.

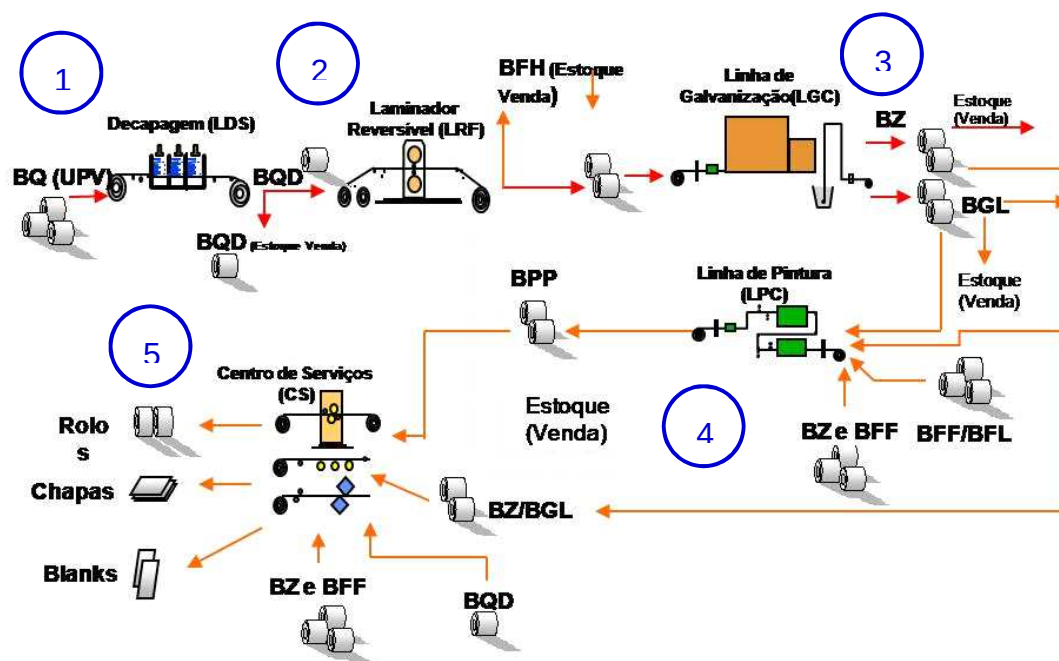
Nossa ideia para este capítulo é a seguinte: o leitor é convidado a entender o problema proposto e ter uma noção geral de como ele foi resolvido. Se desejar, crie o modelo de simulação com base na descrição geral e nos detalhes de implementação que são fornecidos. Por fim, se quiser se estender na análise do problema, pode realizar as atividades complementares, propostas como exercícios.

8.2. Caso 1: movimentação de bobinas de aço pré-pintado

Este caso se refere ao estudo de uma operação de movimentação de bobinas de aço pré-pintado em uma empresa siderúrgica. A finalidade aqui é destacar como o modelo final foi construído, analisado e como foram implantadas as possíveis soluções para o problema de gargalos na movimentação dos produtos acabados na saída de linha de pintura contínua.

8.2.1. Fluxo do processo de produção

Resumidamente, o processo de produção de aço na empresa analisada inicia na linha de decapagem, de onde as bobinas de aço são enviadas para laminação (Figura 8.1). O processo continua na galvanização, etapa onde as bobinas são revestidas. A partir deste ponto, o processo produtivo pode seguir três fluxos: a) os produtos são vendidos para o mercado nacional; b) os produtos são encaminhados para a linha de pré-pintura contínua ou; c) os produtos são encaminhados para o centro de serviços de aço, onde são executadas operações de corte. A continuação do processo através da linha de pré-pintura é a que nos interessa aqui (ponto 4 da Figura 8.1).



Legenda:

- BQ – Bobina de aço laminada a Quente
- BDQ – Bobina de aço laminada a quente Decapada
- BFH – Bobina de aço *Full Hard* (alto grau de dureza)
- BZ – Bobina de aço Zincada
- BGL – Bobina de aço Galvalume
- BFF – Bobina de aço Fina a Frio
- BFL – Bobina de aço Flandres
- LDS – Linha de Decapagem Semi Contínua
- LRF – Laminador Reversível a Frio
- LGC – Linha de Galvanização Contínua
- LPC – Linha de Pintura Contínua

Figura 8.1 - Representação macroscópica do processo produtivo da siderúrgica.

8.2.2. Descrição do sistema

A linha de pré-pintura contínua (LPC) é responsável pela produção dos produtos com maior valor agregado da unidade siderúrgica estudada, respondendo por aproximadamente 27% do volume total faturado pela unidade.

A movimentação na saída da linha é feita por uma ponte rolante com capacidade de carga de 15 toneladas. A ponte possui um elevado percentual de utilização devido às diversas movimentações necessárias para a retirada de produtos da linha de produção e seu carregamento nos caminhões despachados aos clientes.

Iniciamos o estudo elaborando um modelo abstrato do processo de movimentação de bobinas pré-pintadas, com o objetivo de visualizar a estrutura do sistema físico está estruturado. Em seguida, passamos para a etapa de coleta de dados, coletando as frequências de saída de bobinas de aço pré-pintadas da linha de produção, dadas pelo intervalo de tempo de produção de cada bobina – esta informação permite definir os intervalos entre chegadas das bobinas nas

operações subsequentes à pintura²⁶. Os valores medidos, divididos em 7 classes distintas, estão representados na Tabela 8.1 e na Figura 8.2.

Tabela 8.1 - Frequência de saída de bobinas de aço da linha de pintura em minutos (LPC).

Intervalo entre chegadas (minutos)	Percentual de ocorrência
10	1,46%
20	27,06%
30	17,73%
40	34,55%
50	5,12%
60	6,76%
70	7,32%

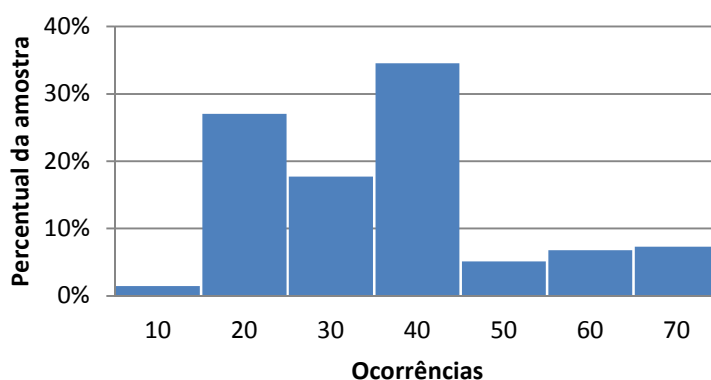


Figura 8.2 - Histograma dos dados da amostra LPC

O modelo proposto para o estudo é mostrado na Figura 8.3. O processo começa com a saída das bobinas de aço da linha de pintura, onde existe um espaço, chamado de “berço”. Pelo fato de existirem duas modas no histograma de frequências mostrado na Tabela 8.1 (ou seja, dois valores mais frequentes, nos blocos de 20 e 40 minutos), não é possível obter uma distribuição de probabilidades contínua adequada ao processo de chega. Assim, optamos por representar o processo de chegadas por meio de uma distribuição discreta (*Probability Profile*).

Uma característica importante da operação da linha é que o berço tem capacidade para receber até 3 bobinas. Assim, caso o berço esteja cheio (ou seja, com 3 bobinas), a saída de uma quarta bobina da linha é impossível. Como a operação de pré-pintura é contínua, esta limitação do berço pode causar paradas na linha.

²⁶ Na verdade, o processo apresenta maior complexidade do que o descrito neste caso. Foram seguidos todos os passos da metodologia de simulação, cujos detalhes não fazem parte do foco deste livro. Por esta razão, a descrição desta etapa do estudo está, propositadamente, simplificada.

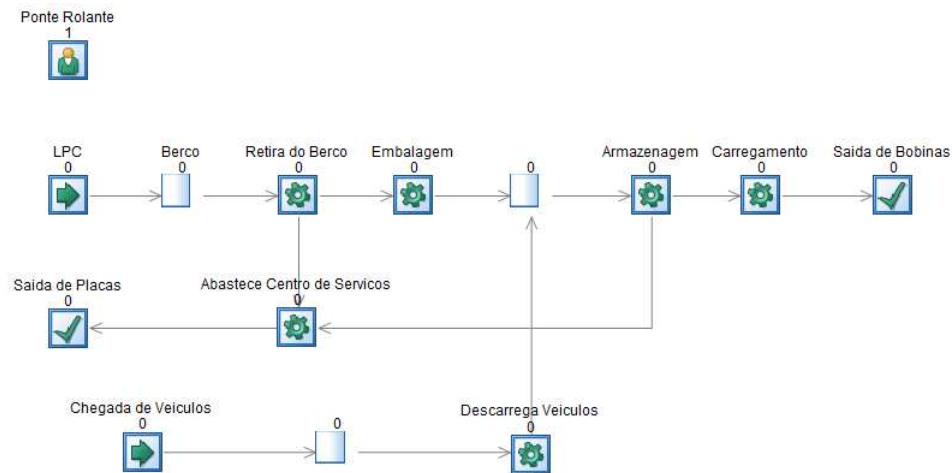


Figura 8.3 – Modelo (simplificado) da siderúrgica

Uma ponte rolante é responsável pela movimentação e manuseio das bobinas de aço desde a saída da linha de pré-pintura até o seu destino final. Assim, a ponte rolante é representada como um recurso associado a **todos** os *Activities* do modelo: Após serem retiradas do berço, 70% das bobinas são encaminhadas ao centro de serviços (onde serão cortadas em placas para serem vendidas com maior valor agregado), enquanto 30% das bobinas seguem para a área de embalagem, onde serão devidamente protegidas contra os efeitos da oxidação antes de serem enviadas para o cliente final. Note que, neste caso, a ponte rolante deve continuar alocada até o final do processo de embalagem.

Existe outro ponto de entrada de bobinas no modelo (identificado na Figura 8.3 como “Chegada de Veículos”), que representa a chegada de bobinas de uma segunda planta, via transporte rodoviário, para serem cortadas no centro de serviços. Analogamente ao que acontece após a saída da LPC, após a armazenagem, 30% das bobinas seguem para o carregamento e 70% para o centro de serviços.

A Tabela 8.2 mostra as distribuições de probabilidades usadas em cada *Activity* e ponto de entrada do modelo. O processo de carregamento não pôde ser caracterizado a partir de dados reais. Assim, optou-se por utilizar a distribuição *Average*, em concordância com os engenheiros da empresa. As distribuições matemáticas dos tempos de operação dos demais *Activities* foram obtidas a partir de dados reais, processados em um *software* de ajuste de curvas (vide Apêndice I). As ordens de prioridade dos processos na alocação da ponte rolante são: “Retira do Berço” – prioridade = 100; “Descarrega Veiculos” – prioridade = 75; demais processos – prioridade = 50 (valor padrão do SIMUL8). O processo de retirada do berço é considerado o mais crítico, devido ao seu limite de armazenamento.

Tabela 8.2 - Distribuições de probabilidade.

Processo	Distribuição
LPC	Probability Profile
Retira do Berço	Fixed(0)
Embalagem	Lognormal(3,5; 0,604)
Armazenagem	Exponencial(4,22)
Carregamento	Average(30)
Abastece Centro de Servicos	Lognormal(3,7; 1,1)
Descarga de Veiculos	Fixed(100)
Chegada de Veiculos	Exponencial(240)

8.2.1. Construção do modelo computacional

Nesta seção, discutiremos detalhes sobre a construção do modelo de simulação proposto para o estudo. Na sequência do texto, vamos considerar que o leitor que construiu o modelo já inseriu os objetos e ligações ilustradas na Figura 8.3 e garantiu que os tempos de deslocamento (*Travel times*) sejam nulos (Seção 1.5.1). Portanto, faltam “apenas” as configurações dos objetos e outros detalhes. Vamos a eles.

- a) O intervalo entre chegadas do *Start Point* chamado LPC deve respeitar a distribuição de probabilidade ilustrada na Figura 8.2 (a Seção 3.2 descreve este tipo de operação). Note que o SIMUL8 usa o ponto como separador decimal, ao invés da vírgula.
- b) Crie um *label* numérico chamado “lbl_Destino” e o atribua a cada entidade que entra no modelo, com valores que respeitem a distribuição de probabilidades ilustrada na Figura 8.4 (Dist_destino).

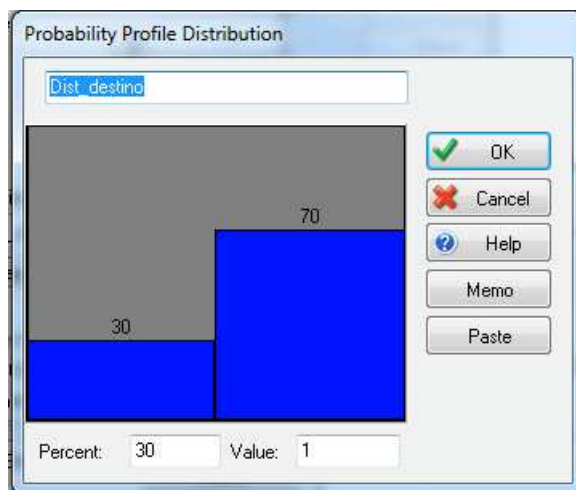


Figura 8.4 - Distribuição do *label* "lbl_Destino".

- c) O *Activity* “Retira do Berço” tem como única finalidade separar as bobinas segundo seu destino. Portanto, sua operação deve ter tempo igual a zero (com distribuição *Fixed*) e seu direcionamento de saída (*Routing Out*) deve ser do tipo *label*, como ilustrado na Figura 8.5. Observe que o primeiro destino da lista de destinos de saída (Embalagem) receberá 30% das bobinas, conforme a descrição do problema.



Figura 8.5 – Roteamento de saída do processo "Retira do Berço".

- d) Repita as operações anteriores para que as bobinas recebidas da segunda planta (*Start Point* “Chegadas de veículos”) sejam direcionadas corretamente pelo *Activity* “Armazenagem”.
- e) Note (Figura 8.3) que não existe uma fila entre os processos “Retira do Berço” e “Embalagem” e entre os processos “Retira do Berço” e “Abastece Centro de Serviços”. Isto foi feito porque o mesmo recurso (ponte rolante) irá executar os dois processos, de forma sequencial. Para que isso aconteça, o recurso ponte rolante não pode ser liberado após o termino do processo “Retira do Berço”, mas sim, após o processo “Embalagem”. Para tanto, ajuste os parâmetros dos recursos nos *Activities* “Retira do Berço” e “Embalagem” como indicado nas Figura 8.4 e Figura 8.5 (para detalhes sobre a configuração de recursos, veja a Seção 4.4).

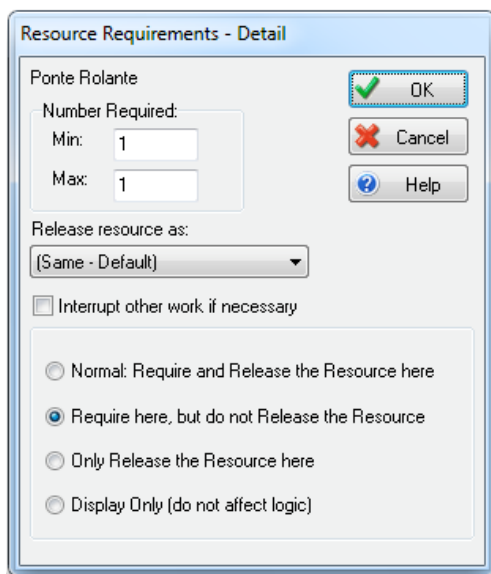


Figura 8.6 - Parâmetros do recurso no processo "Retira do Berço".

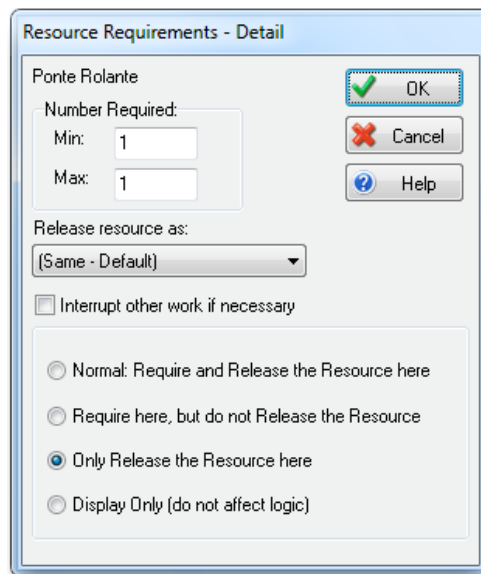


Figura 8.7- Parâmetros do recurso no processo "Embalagem".

- f) A configuração do processo de abastecimento do centro de serviços é idêntica ao de embalagem.
- g) Os processos de armazenagem e carregamento são sequenciais. Portanto, a ponte rolante só pode ser liberada ao final do carregamento – use a mesma forma de alocação discutida anteriormente.
- h) A fila que representa o berço foi considerada com capacidade infinita (configuração padrão) para que fosse possível identificar a eventual ocorrência de problemas (ou seja, mais de 3 bobinas no berço). Outra possibilidade seria considerar a fila com limite ($Capacity = 3$) e observar a perda de produtividade da LPC decorrente de paradas.

8.2.2. Executando a simulação e analisando os resultados obtidos

As rodadas de simulação foram configuradas da seguinte forma:

- Unidade-base de tempo: minutos;
- Período de aquecimento (*Warm-up period*): 10.000 minutos;
- Período de simulação (*Results collection period*): 84.600 minutos, correspondente a 2 meses de operação ininterrupta;
- Número de replicações por rodada (*Trials*): 20.

O ajuste destas configurações foi discutido nas Seções 1.5.2, 1.5.3 e 1.8. Os resultados de interesse do modelo estão apresentados na Tabela 8.3 (a Seção 1.7 discute a criação de relatórios de resultados). Para melhor fundamentar a tomada de decisão, foram considerados 3 cenários, descritos a seguir.

Cenário 1: Modelo da operação atual. A simulação deste cenário serviu para que fossem identificados os gargalos do sistema e os possíveis pontos de melhorias. Como se pode observar nos resultados da simulação (Tabela 8.3), o berço chega a acumular um valor máximo de 5,1 bobinas (o valor não é inteiro por representar a média de 20 replicações). Como o berço comporta até 3 bobinas, sob pena de uma parada na produção, este resultado é preocupante. O fato de a ocupação média do berço ser de 0,8 bobinas traz uma falsa noção de que o berço tem capacidade suficiente para manter a linha de produção ativa. Uma análise mais profunda pode ser feita através do gráfico de comportamento da fila do berço (Figura 8.8), onde podemos observar que a fila contém mais do que 3 elementos em vários momentos da simulação. Um segundo problema que se pode observar nesta simulação é o tempo de espera para o descarregamento, que pode chegar a cerca de 422 minutos (aproximadamente 7 horas!). Por fim, pode-se observar que a ponte rolante possui uma alta taxa de utilização, em torno de 87%. Para melhorar o desempenho do sistema foram consideradas

duas opções: a compra de uma nova ponte rolante ou uma modificação no processo. Estas opções serão detalhadas nos Cenários 2 e 3, descritos a seguir.

Tabela 8.3 - Resultados da simulação do Cenário 1 (modelo original).

Simulation Object	Performance Measure	-95%	Average	95%
Berço	Average queue size	0.78319	0.80028	0.81737
	Maximum queue size	4.95595	5.1	5.24405
Ponte rolante	Utilization %	85.38962	86.58768	87.78573
Fila para descarga dos veículos	Maximum queue size	3.84417	4.4	4.95583
	Maximum Queuing Time	360.05842	421.72898	483.39955
	Average queue size	0.2327	0.26174	0.29079
	Average Queuing Time	56.8662	62.28054	67.69487
Saída de bobinas	Number Completed	786.05281	802.2	818.34719
Saída de placas	Number Completed	1857.10409	1869.05	1880.99591

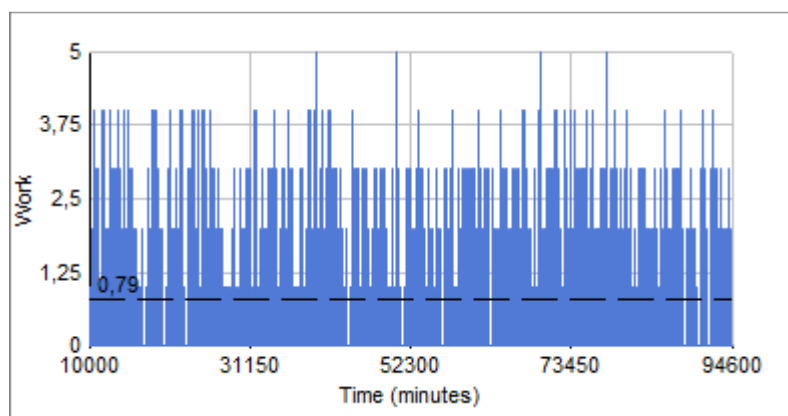


Figura 8.8 - Comportamento da fila do berço.

Cenário 2: Duplicação da capacidade de movimentação pela adição de uma ponte rolante²⁷. Analisando os dados de saída da simulação feita sob estas condições (Tabela 8.4), vemos que a ocupação máxima do berço passou a ser de 2,35 unidades, ou seja, menor do que o limite de 3 bobinas. Apesar de o tempo máximo de fila dos veículos para descarregamento diminuir para cerca de 333 minutos, o valor ainda é alto para o desempenho esperado na referida operação. Por outro lado, neste cenário, as pontes rolantes ficam em utilização somente 41,6% do tempo.

²⁷ Na prática, além da ponte rolante adicional, seria preciso contratar 5 funcionários para operar o equipamento.

Tabela 8.4 - Resultados da simulação do Cenário 2.

Simulation Object	Performance Measure	-95%	Average	95%
Berço	Average queue size	0.0406	0.04299	0.04537
	Maximum queue size	2.12097	2.35	2.57903
Ponte rolante	Utilization %	41.00308	41.58626	42.16943
Fila para descarga dos veículos	Maximum queue size	3.30564	3.8	4.29436
	Maximum Queuing Time	282.938	332.7689	382.5999
	Average queue size	0.139	0.15573	0.17246
	Average Queuing Time	33.8815	37.08295	40.2844
Saida de bobinas	Number Completed	678.1033	689.1	700.0967
Saida de placas	Number Completed	1974.348	1982.4	1990.453

Cenário 3: Eliminação do processo de descarga dos veículos para o abastecimento do centro de serviços. Na prática, esta atividade poderia ser desviada para outro ponto da empresa, cuja ponte rolante possui baixa utilização. Neste cenário, é mantida somente uma ponte rolante executando todas as atividades, exceto o processo de descarga de veículos. Os resultados obtidos para o modelo sob esta condição (Tabela 8.5) indicam que a fila máxima no berço fica entre 2,84 e 3,05 bobinas. Ou seja, ao longo de um grande número de simulações existe a chance de que a fila ultrapasse o limite de 3 bobinas, como pode ser observado no gráfico da Figura 8.9. Nesta rodada de 20 replicações, no entanto, a capacidade do berço nunca bobinas foi ultrapassada²⁸, portanto, não haveria parada da linha de pintura por conta de falta de espaço no berço.

Tabela 8.5 - Resultados da simulação do Cenário 3.

Simulation Object	Performance Measure	-95%	Average	95%
Berço	Average queue size	0.12211	0.126	0.1299
	Maximum queue size	2.84535	2.95	3.05465
Ponte rolante	Utilization %	44.19027	44.82008	45.44989
Fila para descarga dos veículos	Maximum queue size	3.30564	3.8	4.29436
	Maximum Queuing Time	281.5765	331.2911	381.0058
	Average queue size	0.13744	0.15383	0.17022
	Average Queuing Time	33.47184	36.62905	39.78627
Saida de bobinas	Number Completed	786.5785	801.85	817.1215
Saida de placas	Number Completed	1857.878	1869.7	1881.522

²⁸ Vale lembrar que o gráfico mostra somente a última das 20 replicações.

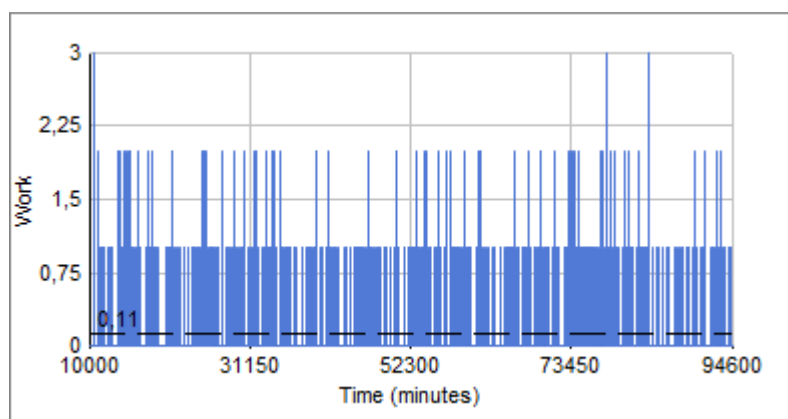


Figura 8.9 - Comportamento da fila do berço.

8.2.3. Análise final

A princípio, pode-se imaginar que o Cenário 2 é o ideal, pois a fila máxima no berço fica sempre abaixo de 3 bobinas, enquanto para o Cenário 3, eventualmente, ocorrerá um ruptura da capacidade máxima do berço. Para uma análise mais profunda da situação temos que calcular o investimento necessário para a implantação de cada cenário.

- Cenário 2: Investimento estimado de cerca de R\$ 3,5 milhões para aquisição de uma nova ponte rolante – além de 5 novos operadores, com custo anual de mão de obra estimado em R\$ 108.000,00;
- Cenário 3: Não é necessário nenhum investimento para efetuar a alteração da descarga, uma vez que já existem possuem operadores disponíveis para desempenhar esta nova atividade.

Logo, o Cenário 2 seria a *última* opção a ser adotada, até porque aumentaria a ociosidade da ponte rolante, além, é claro, do alto valor a ser investido. Assim, entendemos que o Cenário 3 é o que apresenta melhor atratividade para a empresa.

8.2.4. Atividades complementares

Para um melhor entendimento da prática de simulação, sugerimos as seguintes alterações no modelo original.

a) Considere que houve uma ampliação na linha de pintura, que fez com que a taxa de chegada de bobinas fosse triplicada. Estime a quantidade de pontes rolantes necessárias (critério: ocupação máxima do berço) e analise os demais indicadores de desempenho do modelo (vide Tabela 8.4).

- Faça a mesma análise de acordo com a hipótese do Cenário 3;
- Simule diversas configurações, como por exemplo, uma ponte rolante dedicada à retirada do berço e outras para os processos restantes, ou ainda, com todas as pontes podendo fazer todos os processos.

b) A partir do Cenário 1, crie um recurso unitário chamado “Doca”, que será utilizado tanto no carregamento quanto no descarregamento dos veículos. Como isso afeta os resultados de saída da simulação?

c) Modifique as ordens de prioridade do modelo e verifique o que acontece com o desempenho do sistema.

8.3. Caso 2: modelagem e simulação do serviço de suporte a usuários de um ERP

8.3.1. Introdução

Este caso se refere a uma aplicação de simulação na área de serviços profissionais relativos ao suporte/consultoria de uma solução de ERP (*Enterprise Resource Planning*). Este caso tem claramente os atributos de um serviço profissional clássico: baixo volume de clientes atendidos por dia; elevado grau de customização, ênfase em pessoas (e não em equipamentos), ênfase nas atividades de “*front office*” e alta intensidade de contato com os clientes. Para o propósito deste capítulo, este caso complementa os comentários desenvolvidos no anterior, orientando o leitor quanto às especificidades do processo de modelagem e simulação na área de serviços.

A literatura sobre simulação é rica em aplicações sobre manufatura e logística (o caso anterior era exatamente sobre isso), mas há poucas aplicações na área de serviços. Paradoxalmente, na atualidade, os serviços representam algo próximo de 75% do PIB dos países desenvolvidos - no caso brasileiro, já se aproxima de 70% do PIB. Portanto, serviços são cada vez mais essenciais na geração de riqueza.

Para o caso em questão, primeiramente, desenvolveu-se um mapeamento de um processo particular de suporte em alguns módulos da ferramenta mencionada para identificar gargalos e filas no sistema de informação que serve como plataforma de negócio para o atendimento de pedidos de suporte aos clientes. Em seguida, efetuou-se a modelagem e a simulação no SIMUL8, desde o momento em que o pedido de suporte é realizado, até o fim do atendimento, quando o problema é solucionado para o cliente.

Desta forma, foi possível propor melhorias em termos de recursos (humanos e tecnológicos) e de processo para reduzir os gargalos (filas) no atendimento dos pedidos de suporte.

8.3.2. A empresa estudada

Especializada na implantação, desenvolvimento e suporte dos módulos do *software* de gestão integrada, a empresa aqui estudada possui vários departamentos que atuam de forma complementar, cabendo a cada um deles a responsabilidade de um processo particular. Neste caso, vamos analisar exclusivamente o departamento que presta suporte para problemas no ERP. Desta forma, temos uma plataforma *online* em que os clientes (previamente cadastrados para acesso) solicitam suporte, gerando pedidos que são organizados em vários chamados, cada um com um número específico de controle. Além disso, o suporte é designado para equipes especializadas de cada módulo do ERP.

À semelhança das empresas que prestam serviços para grandes provedores de soluções de ERP, a empresa analisada também depende de informações e certificações para que o suporte seja feito. Desta forma, a empresa provedora da solução possui amplo controle sobre as empresas que prestam serviços sob sua chancela.

Para melhor entender a ilustração desenvolvida neste caso, vejamos resumidamente os principais módulos do sistema ERP aqui analisado:

- **MM:** é a sigla para *Material Management* (Administração de Materiais). Este módulo atua sobre o processo de compras de suprimentos, gestão e manuseio da chegada de materiais, controle de chegadas, conferência de faturas e todos os processos que versam em torno da entrada de materiais ou importação dentro da organização, através da automatização e integração com a cadeia de suprimentos, mais especificamente, com a logística de suprimentos;
- **SD:** é a sigla para *Sales and Distribution* (Venda e Distribuição). Este módulo do ERP atua sobre a logística de distribuição, ou seja, o processo de saída do produto acabado da empresa. Também integrado com os outros módulos, este acompanha as notas fiscais, pedidos de remessas, quantidade produzida e vendida e é integrado diretamente com clientes que também utilizam o ERP;

- **FI e CO:** *Financial* (Contabilidade – Financeira) e *Controlling* (Controladoria) são os módulos responsáveis pela gestão financeira e da integração financeira da empresa. Desta forma, eles estão articulados com a produção e a logística, de modo que exista consonância entre faturamento, entrada e vendas. Isto é necessário, uma vez que as transações financeiras e controladoria dependem do funcionamento desta vertente do ERP;
- **PP:** o *Production Planing* (Planejamento de Produção) é o módulo responsável pelo planejamento dos materiais que chegaram ao abastecimento e do acompanhamento da sua transformação em matéria-prima. É considerado um módulo estratégico, uma vez que a atividade final da empresa depende diretamente da estabilidade do fluxo de informações no ERP, o que é responsabilidade deste módulo.
- **BASIS:** não é um módulo do ERP, mas um suporte na estrutura do programa, que auxilia na realização de alterações e personalizações no ERP do cliente. Possui grande demanda, pois todos os outros módulos dependem deste tipo de suporte, caso queiram realizar alterações estruturais no ERP do cliente.

8.3.3. Descrição do sistema

O processo estudado é caracterizado pelo atendimento a pedidos de suporte pelos clientes. O processo mapeado não é complexo, uma vez que o atendimento segue um processo quase linear. O que dificulta o mapeamento é o desmembramento do processo do atendimento em vários módulos, cada qual com sua respectiva fila no atendimento. As etapas do processo e os módulos suportados são descritos a seguir.

- **Pedido de Suporte:** quando o cliente detecta algum problema no sistema ERP, aciona o pedido de suporte pela internet, criando um pedido diretamente para o atendimento referente ao suporte.
- **Triagem:** quando o chamado é aberto, a equipe de diagnóstico recebe as informações fornecidas pelo usuário da empresa em que o problema que ocorreu e faz uma análise para determinar para qual equipe de suporte deve ser direcionada para o pedido.
- **Análise:** a análise feita pela triagem leva em consideração a gravidade do chamado, a quantidade de informações e os consultores disponíveis para atendimento. Desta forma, o tempo de triagem se justifica através da necessidade de coleta destas informações e, às vezes, de contato com o cliente para a obtenção de informações complementares.
- **Decisão (Módulos):** após a análise, os funcionários da triagem devem direcionar o pedido para algum consultor que tenha o conhecimento necessário para a prestação do suporte.
- **MM:** quando o pedido de suporte for referente à entrada de suprimentos na empresa, com relação a algum erro dentro do sistema ERP, como pedido de mercadoria, nota fiscal de entrada, empresas cadastradas, ou qualquer outro problema. O chamado é repassado para um consultor de MM, responsável por analisar e propor uma solução, que será documentada para posterior entrega ao cliente.
- **FI:** o procedimento anterior ocorre para FI (e para CO). O consultor faz o diagnóstico referente às competências do módulo e documenta a solução.
- **SD:** igual aos itens anteriores, só que aplicado à sincronização do ERP com os processos de logística de saída (distribuição).
- **BASIS:** o BASIS também tem seus chamados atribuídos quando chegam. Por isso possui alto volume de demanda, uma vez que presta suporte estrutural para a manutenção do ERP.
- **Documentação:** quando as soluções são encontradas, o consultor precisa documentar a solução em um formulário eletrônico (como se fosse um plano de ação) e enviá-la para que o cliente corrija o problema, ou ele mesmo pode fazer a alteração remotamente, caso seja autorizado. Mesmo nestes casos, o consultor deve realizar a documentação da solução, para otimizar futuros processos.
- **Especialistas (2º nível de atendimento):** caso o problema não seja solucionado, o chamado é encaminhado para um grupo de suporte especialista, que reinicia o atendimento, seguindo o mesmo procedimento anterior, utilizando mais critério na análise do chamado.

- **Encerramento:** caso o pedido seja resolvido por um consultor normal ou pelo especialista, após a solução ser documentada e aplicada/apresentada, o chamado é fechado e o processo analisado se encerra.

Os principais dados coletados foram os tempos de atendimento dos chamados de suporte (SD, BASIS, MM e FI).

Com o processo mapeado, é necessário dispormos dos dados principais que servirão com parâmetros para o funcionamento da simulação. Os processos seguem as distribuições de probabilidade descritas na Tabela 8.6 (todos os tempos estão em minutos).

Tabela 8.6 - Distribuições de probabilidade.

Processo	Distribuição
Chegada	Exponencial(10)
Triagem	Triangular(0.5,1, 1.5)
Atendimento Basis	Normal(30, 7.5)
Atendimento SD	Normal(32, 9)
Atendimento MM	Normal(51, 15)
Atendimento FI	Normal(35,10)
Documentação (todas)	Triangular(15, 30,45)
Especialista	Triangular(30,60,90)

Alguns dados foram modelados usando o Stat::Fit (vide Apêndice I), enquanto outros, como a triagem, atendimento de especialista e preenchimento da documentação foram estimados com base em entrevistas com os consultores. Nestes casos, a experiência dos envolvidos no processo permitiu uma inferência que se aproxima o suficiente da realidade.

De forma resumida, o processo começa com a chegada de um pedido de suporte, que é classificado em um dos dois postos de triagem disponíveis, e enviado para um consultor do módulo (FI&CO, MM, SD, BASIS). Como o BASIS não é um módulo propriamente dito, qualquer consultor pode atender pedidos de suporte para o BASIS. Após o atendimento, se o consultor conseguir resolver o problema do cliente, ele deverá preencher um relatório da solução. Para efeitos de modelagem, o *mesmo* consultor deve atender o cliente e fazer o relatório. Caso o consultor não resolva o problema do cliente, o pedido de suporte será enviado a um especialista. A distribuição de suporte para o BASIS, FI, MM e SD é uniforme, ou seja, 25% para cada.

Portanto há 5 recursos diferentes: BASIS, FI, MM, SD e especialistas. Os recursos MM, SD e FI podem atender o processo BASIS. Existem 3 consultores para cada recurso (FI, MM e SD) e 2 especialistas.

O modelo de simulação proposto está ilustrado na Figura 8.10.

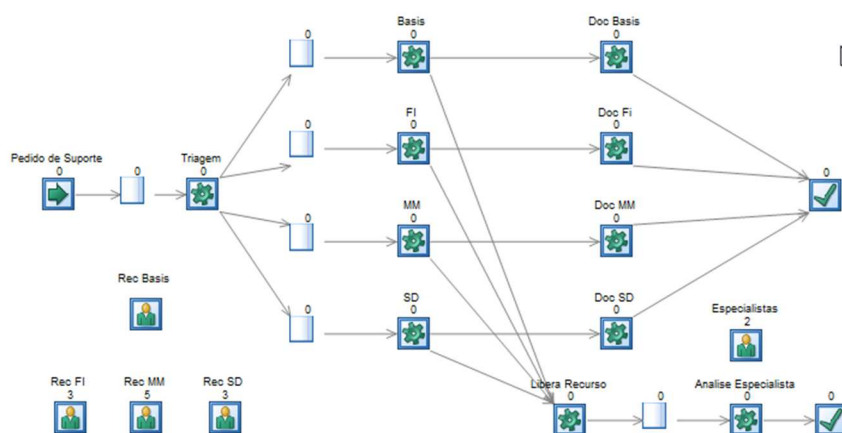


Figura 8.10 - Estrutura do modelo no SIMUL8.

Historicamente, as porcentagens de chamados que precisam passar pelos especialistas são 4%, 7%, 12% e 5%, para os módulos BASIS, FI, MM e SD, respectivamente.

Uma das medidas de desempenho mais importantes neste modelo é o nível de serviço da espera (nas filas de triagem e dos processos de atendimento – BASIS, MM, FI e SD), medido em termos da porcentagem de clientes que esperam menos de 5 minutos.

8.3.4. Construção do modelo computacional

Partindo da estrutura mostrada na Figura 8.10:

- Represente as porcentagens de chamados que devem ser encaminhados aos especialistas após a triagem (opção “*Routing Out - Percent*”).
- Crie 5 recursos (consultores dos módulos BASIS, MM, FI, SD e especialistas) e os associe aos respectivos *Activities*.
- Lembre-se que os consultores dos módulos específicos (MM, SD e FI) também têm formação para atender o módulo BASIS. Isto deve ser modelado por meio de um *pool* de recursos (*Pool Resource*) chamado “Rec Basis”, com os recursos FI, MM e SD como componentes (Figura 8.11). Este recurso é o que será associado ao *Activity* do processo BASIS.
- Como qualquer consultor pode fazer a triagem, associe o recurso “Rec Basis” ao *Activity* “Triagem”²⁹. Faça, ainda, com que este *Activity* adicione um *label* numérico (“Modulo”), com valor definido por probabilidade uniforme entre 1 e 4 para identificar o tipo de pedido de suporte (BASIS, FI, MM ou SD).

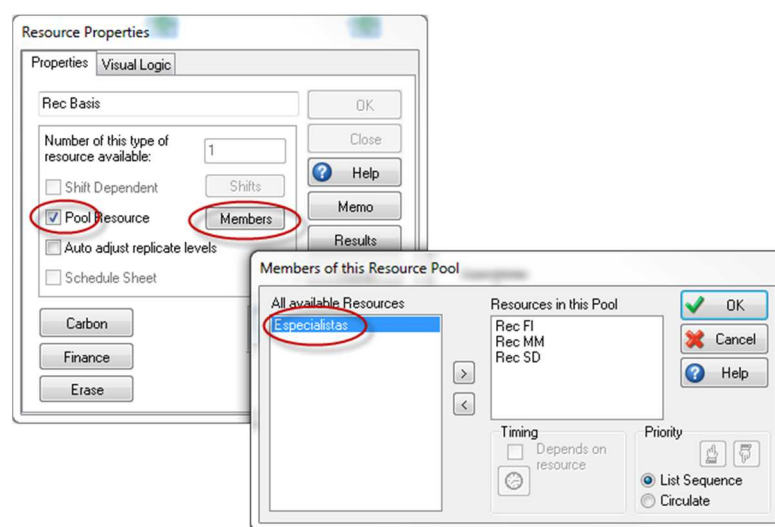


Figura 8.11 – Definição do *pool* de recursos para o atendimento dos módulos.

- Lembrando: o mesmo consultor que fez o atendimento deve fazer o relatório. Portanto, este recurso (consultor) deve ser liberado somente após o respectivo processo de documentação: a opção de alocação dos recursos nos *Activities* de atendimento (BASIS, FI, MM e SD) deve ser do tipo “*Require here but do not Release the Resource*”. Contudo, isto causa um problema na modelagem dos pedidos de suporte que vão para o especialista, pois, nestes casos, o consultor deve ser liberado. Para isso, usamos um *Activity* fantasma (tempo de operação fixo e igual a zero) chamado “Libera Recurso”, cuja única função – como o nome indica – é liberar o recurso alocado (opção de alocação: “*Only Release the Resource here*”). Como as entidades que chegam neste *Activity* “carregam” recursos diferentes, devem ser haver uma forma de

²⁹ Indiretamente, o uso do *pool* de recursos define uma ordem de preferência na alocação.

identificação para que a liberação seja feita corretamente. Assim, é necessário habilitar a opção *Select resource by Label* e indicar o nome do *label* (“Modulo”) nas opções de alocação de recursos do *Activity* (Figura 8.12).

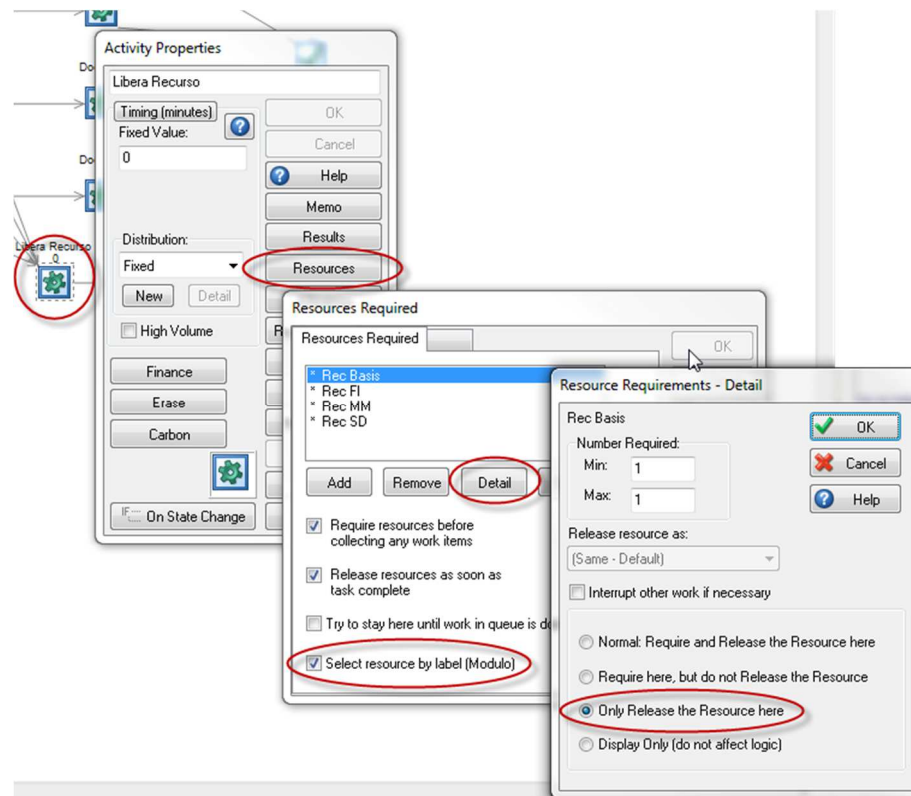


Figura 8.12 - Recursos por *label*.

f) Ajuste a propriedade *Replicate* de cada *Activity* para 20, como mostrado na Figura 8.13. Como o número de consultores é sempre menor do que este valor, o modelo funcionará adequadamente e será flexível, como discutimos na Seção 5.4. A maneira mais elegante de fazer isto, no entanto, seria utilizar a opção dos recursos *Auto Adjust Replicate Levels*. A única exceção é o *Activity* “Triagem”, que deverá ter *Replicate* = 2, já que existem apenas dois postos de triagem;

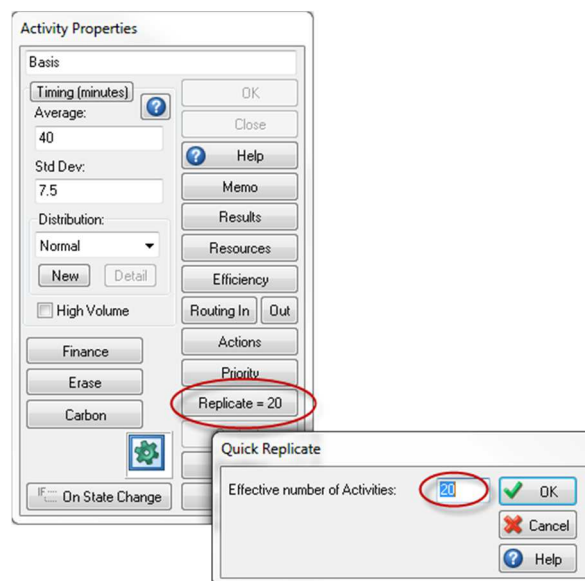


Figura 8.13 - Ajuste do parâmetro de replicação.

g) Não se esqueça de que os recursos devem ser liberados após os processos de documentação (opção “*Only Release the Resource here*”).

8.3.5. Executando a simulação e analisando os resultados obtidos

As rodadas de simulação foram configuradas da seguinte forma:

- Unidade-base de tempo: minutos;
- Período de aquecimento (*Warm-up period*): 1.000 minutos;
- Período de simulação (*Results collection period*): 43.200 minutos, correspondente a 1 mês de operação ininterrupta;
- Número de replicações por rodada (*Trials*): 20.

Assim como no caso anterior, vamos avaliar os resultados da simulação da situação atual antes de avaliar cenários de melhoria.

Cenário 1 – Cenário-base: Os resultados deste cenário estão representados na Tabela 8.7. Pode-se observar que existe uma fila na triagem, com espera média (dos clientes que esperam) de 9,65 minutos, com um pico de 48,97 minutos. Neste tipo de sistema, o cliente está muito pouco inclinado a esperar para ser atendido e exige um alto nível de serviço, uma vez que o não atendimento gera uma parada ou redução do volume e da qualidade do serviço prestado. As filas para os atendimentos para os módulos (FI, MM, SD) também apresentam um alto grau de espera, com valores médios maiores do que os da triagem. A fila para o módulo BASIS é a única que possui espera relativamente baixa: 6,40 minutos. Assim, fica evidente que há falta de recursos. Esta conclusão é comprovada pela Tabela 8.8, onde vemos que os consultores dos módulos FI, MM e SD ficam, respectivamente, 82,34%, 57,79% e 68,91% do tempo ocupados, enquanto o consultor MM fica ocioso durante cerca de 42% do tempo (utilização média de 57,79%).

Tabela 8.7 - Filas para o Cenário 1 (cenário-base).

Objeto	Indicador	95% Inferior	Média	95% Superior
Queue for Triagem	Maximum Queuing Time	43,96	48,97	53,98
	Average (non-zero) Queuing Time	9,04	9,65	10,27
	Maximum queue size	7,51	8,25	8,99
Queue for Basis	Maximum Queuing Time	28,25	31,12	33,98
	Average (non-zero) Queuing Time	5,91	6,42	6,92
	Maximum queue size	2,01	2,20	2,39
Queue for FI	Maximum Queuing Time	143,01	156,16	169,31
	Average (non-zero) Queuing Time	29,76	31,41	33,06
	Maximum queue size	6,47	6,90	7,33
Queue for MM	Maximum Queuing Time	56,59	64,93	73,28
	Average (non-zero) Queuing Time	13,70	14,49	15,29
	Maximum queue size	4,00	4,65	5,26
Queue for SD	Maximum Queuing Time	164,92	181,15	197,38
	Average (non-zero) Queuing Time	37,10	39,37	41,63
	Maximum queue size	6,55	7,20	7,85
Queue for Analise Especialista	Maximum Queuing Time	43,96	48,97	53,98
	Average (non-zero) Queuing Time	22,14	24,26	26,38
	Maximum queue size	7,51	8,25	8,99

Tabela 8.8 - Utilização dos Recursos no Cenário 1 (cenário-base).

Objeto	Indicador	95% Inferior	Média	95% Superior
Rec FI	Maximum Use	3,00	3,00	3,00
	Average Use	2,46	2,47	2,48
	Utilization %	81,86	82,34	82,82
Rec MM	Maximum Use	5,00	5,00	5,00
	Average Use	2,86	2,89	2,92
	Utilization %	57,24	57,79	58,34
Rec SD	Maximum Use	3,00	3,00	3,00
	Average Use	2,04	2,07	2,09
	Utilization %	68,05	68,91	69,78
Especialista	Maximum Use	2,00	2,00	2,00
	Average Use	0,43	0,44	0,46
	Utilization %	21,58	22,18	22,78
Rec Basis	Utilization %	67,05	67,52	67,99

Cenário 2 – Treinamento: Neste cenário, foi simulada a possibilidade de que os consultores pudessem atuar em mais de um módulo (por exemplo, após receberem treinamentos). Desta forma, dos 5 consultores originais atuantes no módulo MM, 2 foram treinados para atender o módulo FI e outros 2 foram treinados para atender o módulo SD. Com isso, a configuração de consultores ficou sendo a seguinte: 3 consultores dedicados a FI, 3 consultores dedicados a SD, 1 consultor dedicado a MM, 2 consultores que servem SD e MM e 2 consultores que servem FI e MM. A implementação deste cenário exigiu a definição de novos *pools* de recursos e a inclusão de *Activities*.

As saídas do novo modelo de simulação são mostradas na Tabela 8.9 e na Tabela 8.10. Pode-se verificar (Tabela 8.9) que os tempos de espera para as filas de triagem e do processo BASIS permanecem aproximadamente os mesmos, o que é condizente com a descrição do cenário, visto que não foi acrescentado nenhum recurso. Os valores médios associados aos tempos dos clientes que aguardam algum tempo nas filas SD e FI diminuíram e o valor correspondente da fila MM aumentou, o que também é condizente com o cenário, haja vista a realocação de recursos.

Tabela 8.9 - Filas para o Cenário 2 (treinamento).

Objeto	Indicador	95% Inferior	Média	95% Superior
Queue for Triagem	Maximum Queuing Time	44,98	49,65	54,32
	Average (non-zero) Queuing Time	9,28	9,91	10,55
	Maximum queue size	7,64	8,30	8,96
Queue for Basis	Maximum Queuing Time	25,21	28,14	31,08
	Average (non-zero) Queuing Time	4,01	4,24	4,46
	Maximum queue size	2,00	2,00	2,00
Queue for FI	Maximum Queuing Time	60,97	66,88	72,79
	Average (non-zero) Queuing Time	14,23	14,96	15,69
	Maximum queue size	4,19	4,55	4,91
Queue for MM	Maximum Queuing Time	67,99	78,51	89,03
	Average (non-zero) Queuing Time	16,41	17,46	18,51
	Maximum queue size	4,04	4,75	5,46
Queue for SD	Maximum Queuing Time	66,22	72,97	79,72
	Average (non-zero) Queuing Time	14,78	15,51	16,23
	Maximum queue size	4,08	4,55	5,02
Queue for Analise Especialista	Maximum Queuing Time	59,34	68,49	77,64
	Average (non-zero) Queuing Time	22,22	24,71	27,20
	Maximum queue size	2,32	2,60	2,88

Nota-se, ainda, (Tabela 8.10) que a utilização dos recursos está mais bem distribuída, com todos os valores (FI, MM e SD) variando, aproximadamente, entre 63% e 71%. Houve uma melhoria substancial neste segundo cenário, mas ainda existe uma espera que, geralmente, não é tolerada por este tipo de cliente, que exige altos níveis de serviço. Para tanto optou pela contratação de três novos consultores, um para cada módulo, situação ilustrada no Cenário 3.

Tabela 8.10 - Utilização dos recursos no Cenário 2 (treinamento).

Objeto	Indicador	95% Inferior	Média	95% Superior
Rec FI	Maximum Use	0,00	0,00	0,00
	Average Use	0,00	0,00	0,00
	Utilization %	69,13	69,69	70,25
Rec MM	Maximum Use	0,00	0,00	0,00
	Average Use	0,00	0,00	0,00
	Utilization %	70,96	71,43	71,91
Rec SD	Maximum Use	0,00	0,00	0,00
	Average Use	0,00	0,00	0,00
	Utilization %	62,86	63,42	63,98
Especialista	Maximum Use	2,00	2,00	2,00
	Average Use	0,43	0,44	0,46
	Utilization %	21,58	22,19	22,80
Rec Basis	Utilization %	67,06	67,52	67,99

Cenário 3 – Treinamento e Contratação: Neste cenário foi simulado o treinamento e a contratação de novos consultores. A configuração de consultores considerada foi a seguinte: 4 consultores dedicados a FI, 4 consultores dedicados a SD, 2 consultores dedicados a MM, 2 consultores que podem tanto atender SD como MM e 2 consultores que podem atender tanto FI como MM.

Segundo os resultados da simulação, mostrados na Tabela 8.11, verifica-se que as esperas médias pela triagem e pelo processo BASIS caíram praticamente pela metade em relação ao cenário-base. As filas nos processos FI, MM e SD têm valores médios um pouco acima de 10 minutos, valores melhores do que os do cenário-base, mas que ainda podem ser considerados altos. A distribuição da carga de trabalho também melhorou (vide Tabela 8.12). No entanto, ainda existem alguns pontos de pico de atendimento, como no processo MM, em que alguns clientes chegam a esperar mais de 50 minutos pelo atendimento. A ocorrência desses picos é praticamente inevitável, a menos que seja alocada uma quantidade muito grande de recursos complementares.

Tabela 8.11 - Filas para o Cenário 3 (treinamento e contratação).

Objeto	Indicador	95% Inferior	Média	95% Superior
Queue for Triagem	Maximum Queuing Time	18,65	21,76	24,86
	Average (non-zero) Queuing Time	3,55	4,19	4,82
	Maximum queue size	3,84	4,40	4,96
Queue for Basis	Maximum Queuing Time	5,74	9,66	13,59
	Average (non-zero) Queuing Time	1,84	2,40	2,97
	Maximum queue size	1,68	1,85	2,02
Queue for FI	Maximum Queuing Time	43,57	47,40	51,24
	Average (non-zero) Queuing Time	11,34	11,91	12,48
	Maximum queue size	3,32	3,75	4,18
Queue for MM	Maximum Queuing Time	49,61	53,74	57,88
	Average (non-zero) Queuing Time	12,98	13,81	14,64
	Maximum queue size	3,40	3,90	4,40
Queue for SD	Maximum Queuing Time	40,62	45,48	50,33
	Average (non-zero) Queuing Time	11,65	12,61	13,57
	Maximum queue size	2,91	3,35	3,79
Queue for Analise Especialista	Maximum Queuing Time	60,43	70,43	80,42
	Average (non-zero) Queuing Time	22,35	24,72	27,09
	Maximum queue size	2,32	2,60	2,88

Tabela 8.12 - Utilização dos recursos no Cenário 3 (treinamento e contratação).

Objeto	Indicador	95% Inferior	Média	95% Superior
Rec FI	Maximum Use	0,00	0,00	0,00
	Average Use	0,00	0,00	0,00
	Utilization %	57,33	57,83	58,33
Rec MM	Maximum Use	0,00	0,00	0,00
	Average Use	0,00	0,00	0,00
	Utilization %	57,73	58,10	58,47
Rec SD	Maximum Use	0,00	0,00	0,00
	Average Use	0,00	0,00	0,00
	Utilization %	43,43	43,97	44,51
Especialista	Maximum Use	2,00	2,00	2,00
	Average Use	0,43	0,44	0,46
	Utilization %	21,58	22,19	22,80
Rec Basis	Utilization %	52,69	53,06	53,42

Na prática, mais importante do que observar a ocorrência dos picos é determinar com qual frequência eles ocorrem. Para isso, observe a Tabela 8.13, que mostra a porcentagem dos atendimentos que tiveram uma espera de, no máximo, 5 minutos.

Tabela 8.13 - Porcentagem média de clientes atendidos em menos de 5 minutos em cada cenário.

Objeto	Cenário 1	Cenário 2	Cenário 3
Queue for Triagem	94,0	93,0	99,4
Queue for Basis	97,2	97,7	99,9
Queue for FI	46,0	80,4	92,9
Queue for MM	87,9	76,6	91,9
Queue for SD	58,9	83,9	96,5

A Tabela 8.13 mostra a evolução do nível de serviço obtido em cada cenário: os níveis para os processos de triagem e BASIS estavam dentro de limites aceitáveis desde o primeiro cenário. O módulo MM, que tinha um nível de serviço de 87,9% no cenário-base, teve uma queda no Cenário 2, em que os consultores deste módulo passaram a atender também os módulos SD e FI. No terceiro cenário, todos os processos tiveram mais de 90% dos atendimentos com espera abaixo de 5 minutos de espera. Estes valores podem ser usados como uma forma de inferir o nível de serviço prestado ao cliente, pois, como dito anteriormente, o processo real é aleatório e a formação de filas não pode ser evitada, mas sim controlada.

8.3.6. Atividades complementares

Monte e analise os cenários descritos a seguir. Antes de simular, tente deduzir o impacto das mudanças.

- Os especialistas também podem ser utilizados para atendimento aos módulos MM, FI e SD;
- Todos os consultores podem atender a todos os módulos;
- Todos os consultores têm, no mínimo, duas especialidades e alguns podem ter as três (MM, SD e FI).

Comparando cada alternativa com o cenário-base, qual parece ser a melhor opção? Assuma que os custos de treinamento não são relevantes.

Apêndice I TUTORIAL DE USO DO STAT::FIT

I.1. Introdução

O Stat::Fit é um *software* desenvolvido pela *Geer Mountain Software Corporation* para ajustar um conjunto de dados a uma distribuição estatística adequada, um processo conhecido como ajuste de curvas ou *fitting*. Uma versão educacional do *software* pode ser obtida gratuitamente no sítio de seu fabricante³⁰. A principal limitação desta versão está no total de valores que podem ser avaliados (até 50) e de distribuições utilizadas no processo de ajuste (4 distribuições contínuas e 2 distribuições discretas). As figuras mostradas neste tutorial foram criadas com a versão completa do *software*, mas principais diferenças serão destacadas, quando julgarmos necessário.

Na sequência, apresentaremos um pequeno tutorial de uso do Stat::Fit, com base no conjunto de valores dado na Tabela I.

Tabela I.1 - Conjunto de dados.

15,19	10,43	22,89	16,57	19,23
12,57	7,46	11,90	11,65	6,88
17,94	34,26	6,56	15,57	17,16
7,15	12,32	19,45	23,68	22,38
16,44	14,96	7,35	18,19	19,92

I.2. Pré-processamento

Todo conjunto de dados coletados pode conter valores discrepantes, ou *outliers*. Estes valores, quando existirem, devem ser removidos antes que o conjunto de dados seja inserido no Stat::Fit, já que o *software* não possui ferramentas para detecção de tais valores. Para mais detalhes sobre este assunto, consulte o Capítulo 2 do livro “Modelagem e Simulação de Eventos Discretos” de Leonardo Chwif e Afonso C. Medina.

I.3. Entrada de dados

Inicie o Stat::Fit. O programa deverá abrir uma janela para entrada de dados (Figura I.14). Se isso não acontecer, clique em *File* | *New* para abrir uma nova janela.

³⁰ Acesse: <http://www.mcssl.com/store/geermountainsoftwarecorp>.

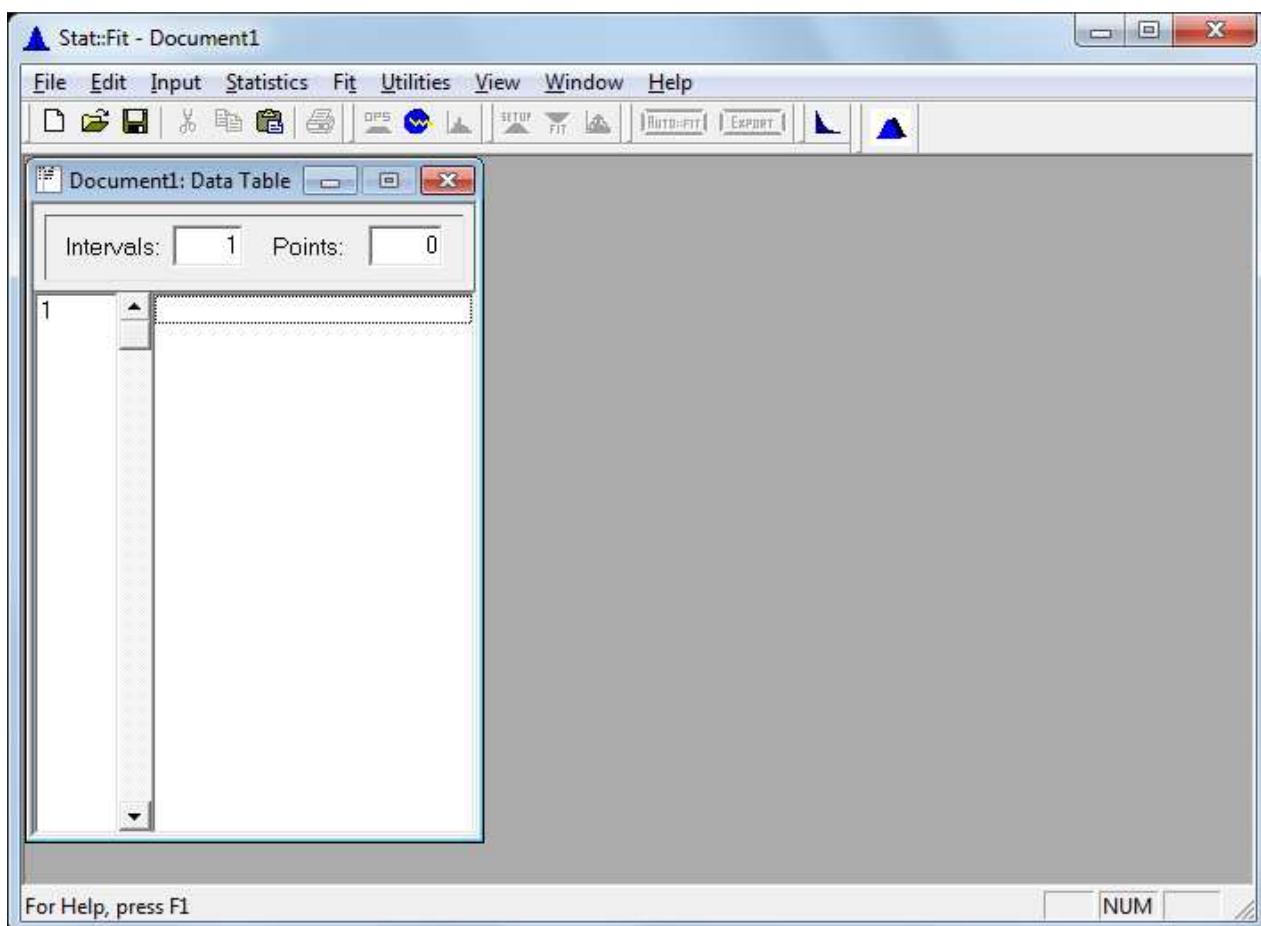


Figura I.14 - Tela inicial do Stat::Fit.

Em seguida, insira os valores dados na Tabela I.1, usando o ponto como separador decimal. A sequência de entrada dos valores deverá ser feita por colunas (vide Figura I.2). Isto é fundamental para a análise de correlação dos dados.

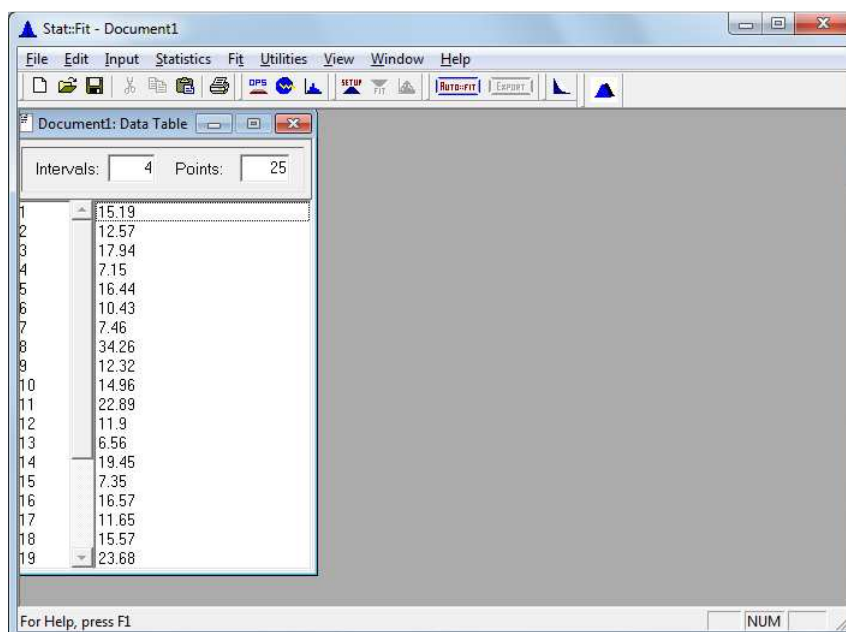


Figura I.2 - Janela de dados.

I.4. Histograma

Clique no botão *Input Graph* para gerar o histograma do conjunto de dados (Figura).

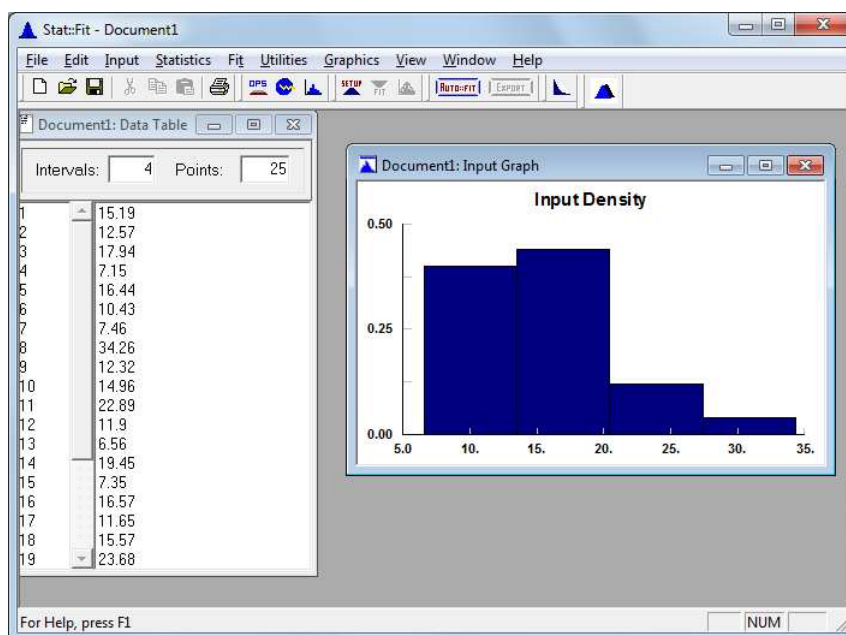


Figura I.3 – Histograma do conjunto de dados.

O número de classes do histograma é calculado automaticamente, mas pode ser alterado clicando-se em *Input | Options* (Figura I.4).

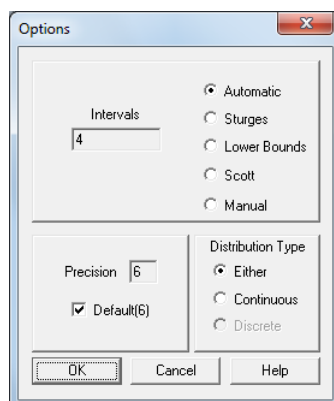


Figura I.4 - Opções do Stat::Fit.

I.5. Estatística descritiva

Clique em *Statistics | Descriptive* para obter um resumo das principais medidas de posição e dispersão do conjunto de dados. A Figura I.5 mostra os resultados obtidos para os dados da Tabela I.1. As medidas de posição (média, moda etc), de dispersão (desvio padrão, coeficiente de variação) e de forma (curtose) são calculadas de acordo com as fórmulas clássicas da Estatística.

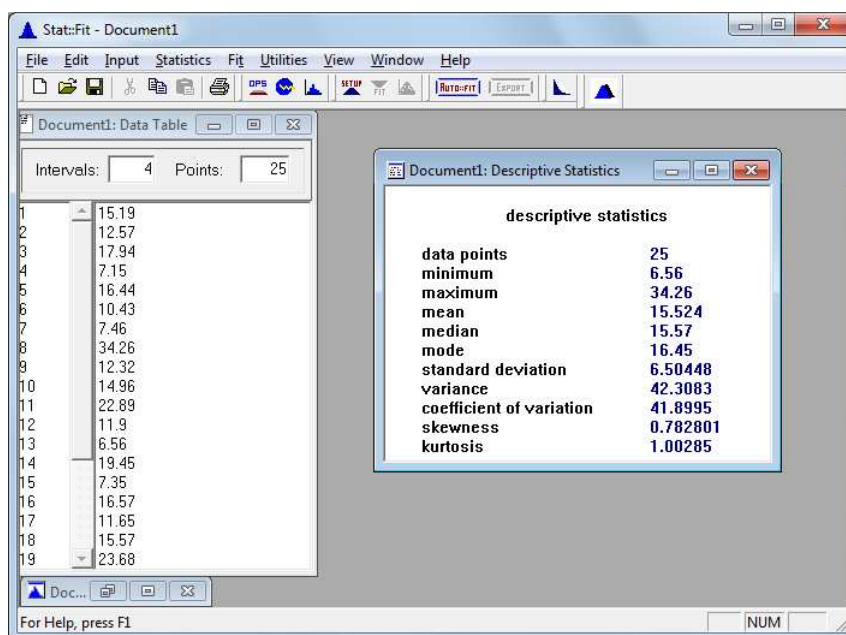


Figura I.5 - Resumo descritivo dos dados.

I.6. Ajuste (*fitting*)

Após a inserção dos dados, o Stat::Fit realiza o processo de ajuste de forma praticamente automática: clique no botão Auto::Fit e selecione a “família” de distribuições a serem testadas, em função da natureza do processo que deu origem aos dados. Por padrão, o Stat::Fit considera que o ajuste deve ser feito por distribuições com limite inferior (inteiro) dado pelo menor valor do conjunto (Figura I.6). Para este exemplo, vamos considerar que esta escolha é adequada, ou seja, que a distribuição escolhida não deve considerar valores abaixo de 6. Clique no botão OK para concluir o processo de ajuste.

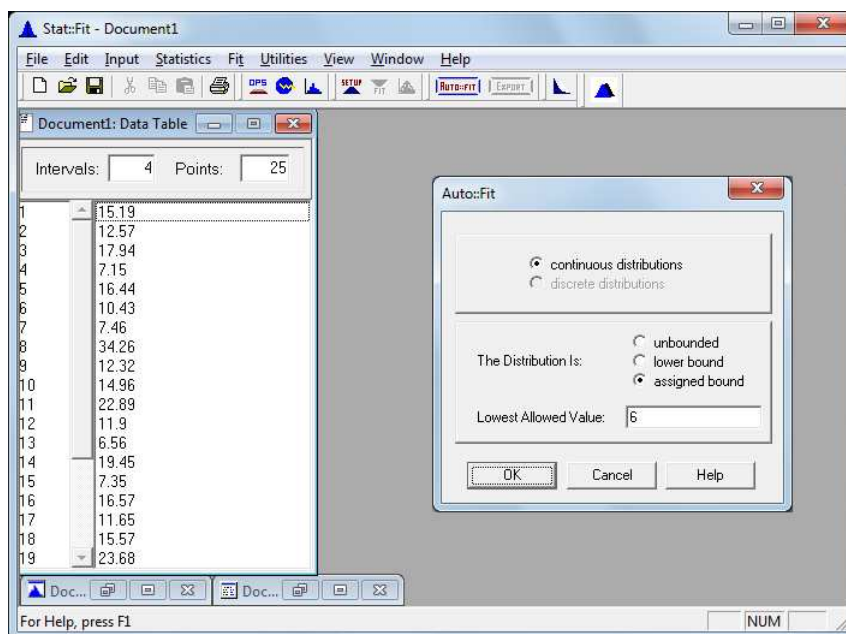


Figura I.6 - Configurações do ajuste automático.

O resultado do ajuste é mostrado em uma lista contendo as distribuições testadas (Figura I.7), em ordem decrescente de preferência. Para cada uma é dada a indicação de quais são consideradas adequadas (*Do not reject*) ou não (*Reject*) para o ajuste.

Os resultados da Figura I.7 mostram que foram testadas 13 distribuições de probabilidades, o que não acontecerá se for usada a versão estudantil do *software*, limitada a 6 distribuições. De qualquer forma, as operações descritas na sequência funcionarão da mesma forma em qualquer versão do programa.

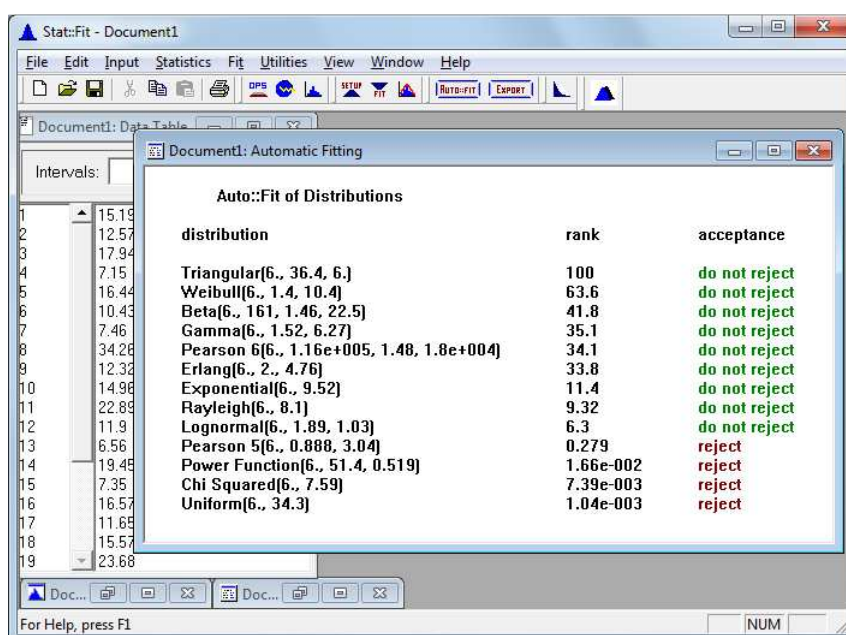


Figura I.7 - Resultados do ajuste.

Clique em uma das distribuições para desenhar o gráfico da curva correspondente sobre o histograma de frequências (Figura I.8).

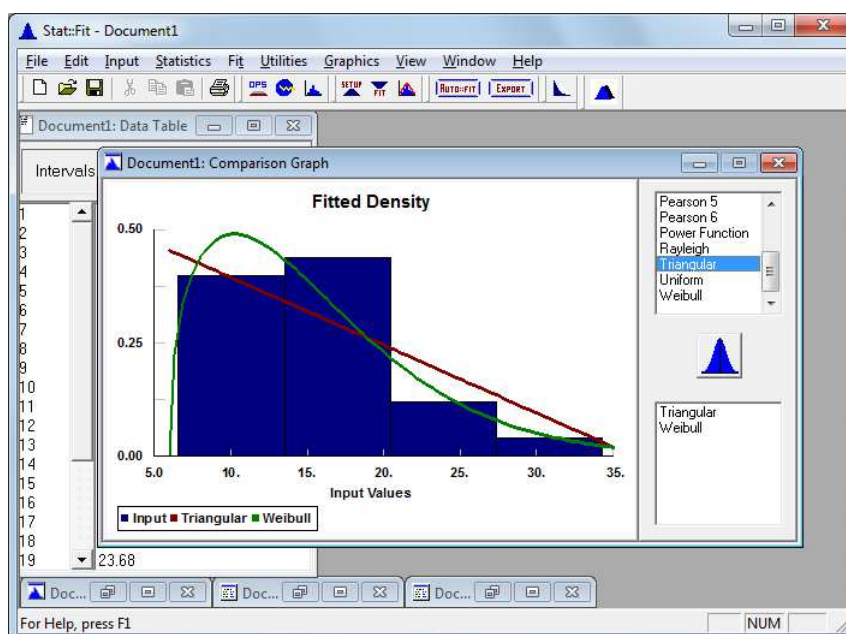


Figura I.8 - Resultado gráfico do ajuste.

Para verificar a qualidade do ajuste proporcionado pelas distribuições testadas clique em *Fit* | *Goodness of Fit*. Quanto maior o valor do *p-value* associado a uma distribuição, melhor será a qualidade do ajuste (Figura I.). De forma simplificada, valores de *p-value* maiores ou iguais a 0,1 representam ajustes adequados. Por exemplo, seria adequado considerar que o conjunto de dados da Tabela I.1 pode ser representado por uma distribuição Weibull, pois o *p-value* obtido resultou igual a 0,516.

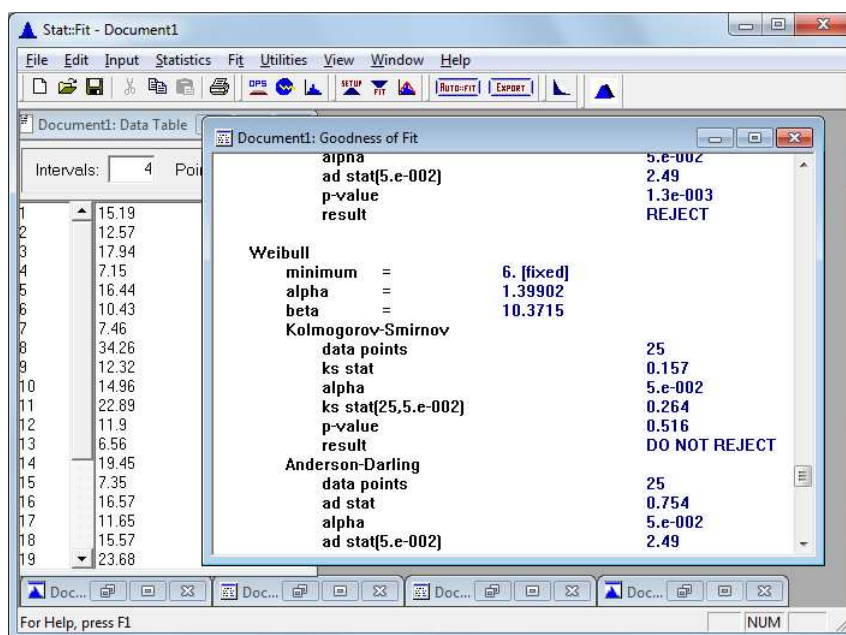


Figura I.9 - Qualidade do ajuste.

I.7. Análise de correlação

O uso de uma distribuição estatística para representar um conjunto de dados depende da *independência entre os valores*, ou seja, é preciso garantir que não exista *correlação* entre os dados. Por exemplo, se os valores medidos correspondem a um processo manual, sujeito à fadiga do operador, não se pode usar apenas uma distribuição para representar a duração do processo, já que as suas medidas de posição e dispersão (média, desvio padrão etc) variam com o tempo. Para avaliar o coeficiente de correlação dos dados do exemplo, clique em *Statistics | Independence | Scatter Plot*: o diagrama de dispersão gerado (Figura I.10) mostra que não há indícios de correlação entre os valores. Um exemplo de dados com forte correlação positiva (o dados tendem a aumentar ao longo da coleta) é ilustrado pelo diagrama da Figura I.I.11.

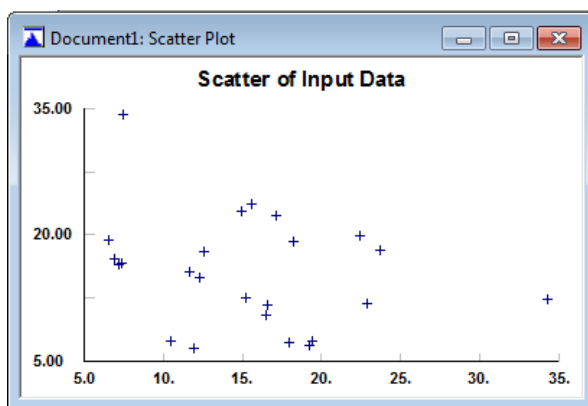


Figura I.10 - Diagrama de dispersão de dados com baixa correlação.

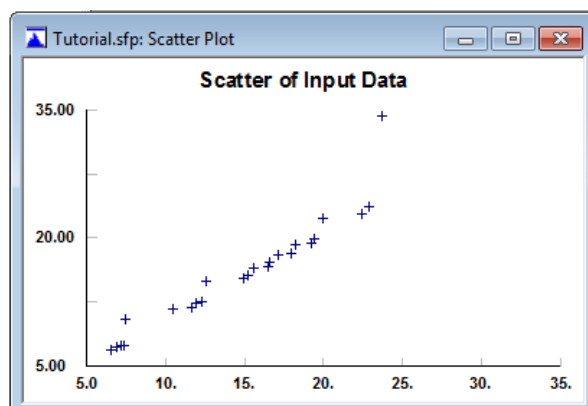


Figura I.11 - Diagrama de dispersão de dados com alta correlação.

Outra forma de se verificar se há indícios de correlação é por meio do coeficiente de correlação, que deve apresentar valores abaixo de 0,6 (quanto mais próximo de 1, maior a correlação entre os dados).

Clique em *Statistics | Independence | Autocorrelation* (Figura I.12) e observe que há poucos indícios de correlação entre o conjunto de dados deste exemplo. Como a correlação pode ser positiva ou negativa, são dados dois valores para o coeficiente de correlação (+0,216 e -0,363). Evidentemente, o maior dos valores (em módulo) é que deve ser considerado na análise.

Na prática, se houver alta correlação entre os dados, deve-se utilizar mais de uma distribuição estatística, com base em "sub-amostras" nas quais os dados possam ser considerados independentes entre si (por exemplo, antes e depois do almoço). Em um modelo do SIMUL8, cada uma destas diferentes distribuições deverá ser associada a uma distribuição *time dependent*, como discutido no Capítulo 3.

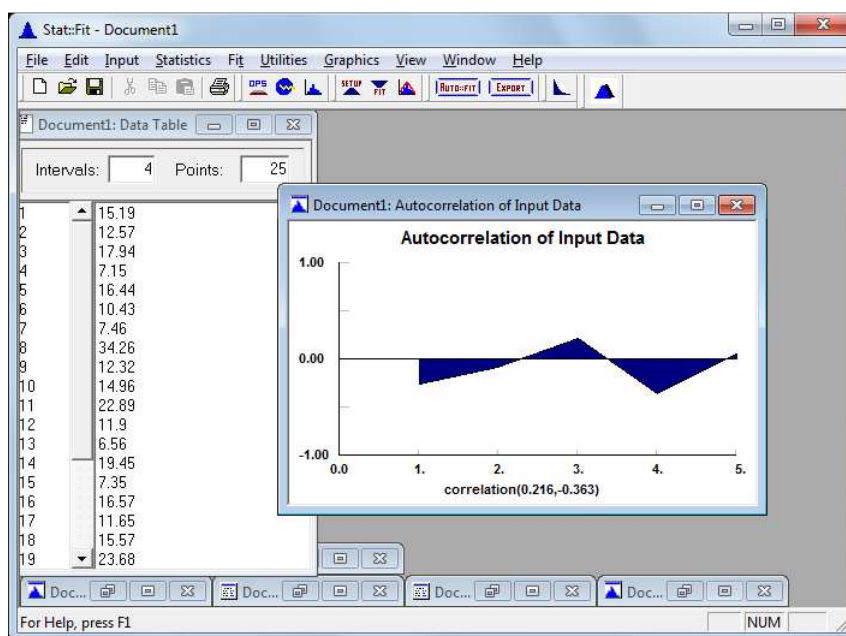


Figura I.12- Coeficiente de correlação.

I.8. Definindo a distribuição mais adequada

Com base nas análises feitas, podemos concluir que o conjunto de dados da Tabela I.1 pode ser ajustado por uma distribuição triangular (vide figura I.7). No entanto, a análise visual do histograma mostra uma tendência de cauda para a direita, que não é representada adequadamente por uma distribuição triangular. Neste ponto, o conhecimento sobre a natureza dos dados coletados é fundamental. Para o caso do exemplo que estamos estudando, devemos nos perguntar: existe a possibilidade, ainda que pequena, de ocorrerem valores muito acima da média? Se a resposta for “sim”, então a distribuição Weibull (“segunda escolha” do Stat::Fit) pode ser a mais adequada para a representação do conjunto de dados. Isto não significa, no entanto, que a distribuição triangular seja uma má escolha: do ponto de vista puramente matemático, ela seria a melhor função para modelar o conjunto de dados proposto.

Independentemente da escolha final, o uso da distribuição no SIMUL8 depende dos valores numéricos de seus parâmetros: no caso da distribuição triangular estes parâmetros são o menor valor, o valor mais provável e o valor máximo; para a distribuição Weibull, os parâmetros são: valor mínimo (quando há), parâmetro de forma e parâmetro de escala. Para obter os valores dos parâmetros em um formato adequado para uso no SIMUL8, clique em *File | Export | Export Fit* e selecione a distribuição desejada (Figura I.13). Antes mesmo da exportação, o Stat::Fit indica que, para o caso da distribuição Weibull, deve-se combinar uma distribuição fixa (ou seja, um valor constante) com uma distribuição Weibull de parâmetros 1,4 e 10,4. De fato, a “exportação” do Stat::Fit consiste apenas em copiar para a área de transferência (ou arquivo de texto) uma frase como “create a combination distribution with a fixed offset of 6. then add Weibull, 1.4, 10.4”. No SIMUL8, estas informações terão que ser inseridas manualmente.

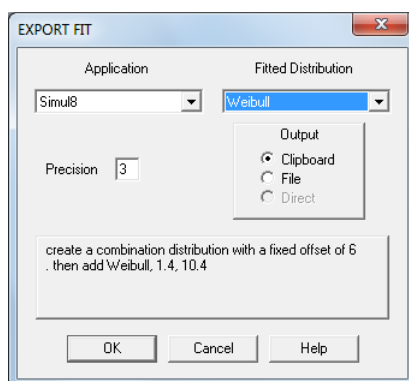


Figura I.13 - Exportação de uma distribuição.

Isto conclui este tutorial. Outras funcionalidades do Stat::Fit podem ser consultadas em sua documentação eletrônica.

SIMULATE TECNOLOGIA DE SIMULAÇÃO LTDA.

A **SIMULATE** É UMA EMPRESA JOVEM, DINÂMICA E SEMPRE ATUALIZADA COM AS MELHORES PRÁTICAS DA SIMULAÇÃO MUNDIAL.

PRODUTOS E SERVIÇOS OFERECIDOS PELA SIMULATE:

- TREINAMENTO GERENCIAL E OPERACIONAL EM SIMULAÇÃO;
- SUPORTE ESTRATÉGICO À IMPLANTAÇÃO DA SIMULAÇÃO EM EMPRESAS;
- DESENVOLVIMENTO DE MODELOS DE SIMULAÇÃO;
- DESENVOLVIMENTO, VENDA E SUPORTE DE SOFTWARES VOLTADOS À SIMULAÇÃO.

PARA MAIORES DETALHES, ENTRE EM CONTATO:



Tel.: (11) 5081-7472

simulate@simulate.com.br

VISITE NOSSO SITE NA INTERNET:

www.simulate.com.br