

## 5

### O Modelo de simulação

#### 5.1. A Seleção do Ambiente de Desenvolvimento

Dois modelos, cada um representando um dos regimes de atendimento em estudo, foram desenvolvidos integralmente em VBA Excel.

A escolha do VBA em ambiente EXCEL deveu-se a fatores diversos:

- (a) o suporte de ferramental estatístico, disponível no EXCEL;
- (b) a versatilidade do VBA como linguagem de programação e sua ampla disponibilização - fornecido junto ao pacote Microsoft Office, amplamente utilizado;
- (c) com a utilização de linguagem de programação genérica, obtém-se a flexibilização desejada na criação de variáveis de saída, nem sempre facilmente modeladas em softwares específicos de simulação;
- (d) por fim, e o mais representativo - a natureza combinatória do problema em estudo não se mostrou adequado à representação/modelagem em software específico de simulação, após efetuados testes exaustivos neste sentido (software ARENA versão STUDENT). Exige uma grande quantidade de iterações (proporcionados por *loops*) em diversas etapas do processo, que inviabilizaram sua modelagem adequadamente, nesta versão STUDENT.

A princípio, desenvolveu-se o primeiro modelo (**PEPS**) como um híbrido entre o VBA-Excel e o software de simulação ARENA.

O aplicativo VBA produzia previamente:

- (a) o processo de embaralhamento (*shuffling*) das carretas, de forma a produzir a aleatoriedade necessária ao processo de chegadas;

(b) os processos de (b1) embaralhamento (*shuffling*) dos contêineres e (b2) seu subsequente empilhamento lógico (seis pilhas, cada qual com altura de cinco contêineres), necessário para fornecer a aleatoriedade na montagem das pilhas;

(c) toda a movimentação lógica da estrutura de pilhas, simulando a operação do *reach stacker* - a geração da informação dos consequentes acessos mínimos a cada um dos contêineres, fruto desta movimentação.

Tais dados eram então transmitidos ao aplicativo ARENA, que simulava as chegadas de carretas uma-a-uma, na seqüência estabelecida pelo aplicativo VBA-Excel, processando-as tantas vezes quanto estipuladas previamente, gerando seus relatórios genéricos.

O aplicativo ARENA executava, portanto, as seguintes funções:

- geração de chegadas segundo modelo Poisson,
- leitura dos parâmetros de cada chegada (anteriormente pré-processadas pelo modelo VBA-Excel), a saber – identificação da carreta (contêiner) e quantidade de movimentações para acessar o contêiner correspondente,
- processamento das movimentações, segundo distribuição estatística específica  $N(2,5 ; 0,1667)$  (definida anteriormente)
- um gerador de relatórios, apresentando variáveis de desempenho não-específicas, que pouco contribuiriam para o melhor entendimento do experimento em estudo (detalhamento das variáveis de desempenho desejadas será fornecido no item seguinte 5.2),

A freqüência de troca de arquivos entre os softwares ARENA e VBA-Excel foi identificada previamente como um problema para o processamento do modelo MOVIMENTAÇÃO MÍNIMA. Especificamente em relação ao processo decisório envolvido neste modelo - que apresenta um maior nível de complexidade na modelagem, o desenvolvimento em linguagem de programação genérica (no caso, o VBA-Excel) tornou-se necessário. Este processo envolve entidades/variáveis diversas, que devem ser registradas e atualizadas permanentemente, a cada

movimentação de contêineres, além de decisões correspondentes a tais atualizações. Estão envolvidos, em linhas gerais, a identificação das carretas em fila, a cada atualização do relógio de simulação com:

- seus tempos de espera individuais atualizados,
- suas movimentações mínimas de acesso, apuradas individualmente, após cada novo atendimento,
- a decisão em relação ao contêiner selecionado a ser retirado, com base na mínima dentre as mínimas movimentações dos contêineres presentes na fila, àquele instante,
- as trocas de arquivos entre os softwares VBA-Excel e ARENA, para incluir as tomadas de decisão, a cada atendimento, sobre o contêiner a ser retirado.

A construção de modelo com este grau de complexidade e controle ultrapassaria facilmente a capacidade de processamento do ARENA versão STUDENT, já praticamente atingido no modelo anterior (**PEPS em VBA-Excel+ARENA**), acima apresentado, de concepção bastante mais simples que este último.

Optou-se então pelo desenvolvimento dos modelos de simulação dos dois regimes em estudo, integralmente, em VBA-EXCEL, abandonando-se o aplicativo regime PEPS desenvolvido em VBA-Excel + ARENA.

O Apêndice 1 apresenta em detalhe este modelo, plenamente funcional.

## **5.2. A Estrutura Excel, Base dos Modelos de Simulação**

Todo aplicativo desenvolvido em linguagem VBA-Excel tem suas bases de dados em planilhas Excel.

Os modelos dos regimes constam de dois aplicativos em um arquivo Excel, com três planilhas. A primeira relativa ao modelo **MOVIMENTAÇÃO MÍNIMA**, a segunda ao **PEPS** e a terceira, de consolidação de resultados dos dois regimes.

As duas primeiras planilhas de modelo possuem os seguintes módulos (arquivos), conforme abaixo.

### 5.2.1. A Geração de eventos estatísticos

Neste módulo são realizadas as seguintes tarefas:

- (1) O embaralhamento (*shuffling*) de contêineres,
- (2) O embaralhamento (*shuffling*) de carretas,
- (3) A geração estatística de taxas de atendimento, segundo item 4.1 acima (maior detalhamento no Apêndice 2),
- (4) A geração estatística de tempos entre chegadas, segundo especificado no item 4.2 acima (maior detalhamento no Apêndice 2).

### 5.2.2. A Movimentação lógica das estruturas de pilhas

Esta estrutura consta dos seguintes componentes:

**ESTRUTURA ORIGINAL** – resultado de embaralhamento e empilhamento da estrutura de contêineres originalmente gerado ao início de cada nova replicação.

**ESTRUTURA MODIFICADA** – resultante das modificações (montagens e desmontagens) sofridas na ESTRUTURA ORIGINAL, a partir da retirada inicial e das subseqüentes retiradas/recebimentos de contêineres de/para sua estrutura.

**ESTRUTURA BUFFER** – resultante das modificações (montagens e desmontagens) realizadas na ESTRUTURA BUFFER, a partir do recebimento inicial e das subseqüentes retiradas/recebimentos de contêineres de/para sua estrutura.

A importância deste módulo está associada integralmente ao processo de depuração do modelo, para que se possa visualizar se as movimentações de contêineres estão ocorrendo tal como programado em código fonte.

O Quadro 5.2.2.A apresenta sua estrutura, após duas movimentações de contêineres, o que permite um acompanhamento lógico das movimentações realizadas, de forma a proporcionar um melhor entendimento do processo de montagem/desmontagem de pilhas.

O modelo apresentado é o **PEPS**, de mais fácil compreensão, por não envolver critério de decisão envolvido na ordem de atendimento à fila.

**Quadro 5.2.2.A**

Carreta Nº	Contêineres a serem Movimentados	Altura das Pilhas Originais (após a movimentação)						Altura das Pilhas Originais (antes da movimentação)						Altura das Pilhas Buffer (após a movimentação)						Altura das Pilhas Buffer (antes da movimentação)						altura original	altura buffer	altura original	altura					
	Pela Direita	Pela Esquerda	Lado Acesso	Movimentação	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6						
10																																		
11																																		
12	22	17	12	ESQ	12	0	0	3	5	5	5	5	5	5	5	5	1	5	5	0	0	0	0	0	0	0	0	0	0	0	1	0	3	4
13	27	5	11	DIR	5	0	2	5	5	5	5	0	0	3	5	5	1	5	0	0	0	0	0	1	5	5	0	0	0	0	1	3	1	

Observa-se, neste Quadro 5.2.2.A, inicialmente na célula A12 da planilha, a chegada da carreta 22 (que corresponde ao contêiner de mesmo número), gerado através do módulo descrito no item 5.2.1 acima.

Este contêiner está presente na JANELA ORIGINAL, na posição (pilha 3, altura 4), conforme pode ser visualizado na planilha (cujas informações

encontram-se devidamente registradas pelo programa nas células AD12, AE12, AF12 e AG12).

Todo o contêiner pode ser movimentado pelos dois lados – direito ou esquerdo, sendo de interesse obviamente o acesso pelo lado que proporciona a movimentação mínima. Visualiza-se que, pelo lado direito são necessárias dezessete movimentações para obtê-lo, enquanto pelo esquerdo são necessárias apenas doze (registro nas células B12, C12). Escolhe-se então o acesso pelo lado esquerdo, desfazendo-se a **JANELA ORIGINAL** e montando-se os onze contêineres não desejados na **ÁREA BUFFER** em ordem inversa ao do desmonte efetuado, ficando assim com as seguintes estruturas **ORIGINAL MODIFICADA** e **BUFFER** intermediárias apresentada no Quadro 5.2.2.B.

**Quadro 5.2.2.B**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
1	27	14	6	18	13	15	5				0	0	0	18	13	15	5						0	8	10					5
2	3	5	22	19	26	20	4	AL TURA			0	0	0	19	26	20	4	AL TURA					0	28	24					4
3	11	7	1	21	23	2	3	AL TURA			0	0	1	21	23	2	3	AL TURA					0	7	11					3
4	24	28	4	29	9	30	2	AL TURA			0	0	4	29	9	30	2	AL TURA					0	5	3					2
5	10	8	25	12	17	16	1	AL TURA			0	0	25	12	17	16	1	AL TURA					6	14	27					1
6	1	2	3	4	5	6					1	2	3	4	5	6						1	2	3	4	5	6			
7	PILHA Nº										PILHA Nº													PILHA Nº						
8	JANELA ORIGINAL							EXECUTAR			JANELA MODIFICADA													AREA BUFFER						

Este Quadro transitório foi criado apenas para fins didáticos, vez que o Quadro 5.2.2.A acima representa a migração total de contêineres após os dois acessos, mascarando a informação parcial, ora apresentada.

Observe-se que se movimentam contêineres indesejados do lado desejado da estrutura em que este se encontra para o lado correspondente da estrutura de transferência, o que proporciona o mínimo deslocamento do *reach stacker*. Se a movimentação é feita a partir da estrutura original pelo lado esquerdo, o

preenchimento da estrutura *buffer* é realizado pelo mesmo lado, a menos que o lado para o qual seja transferido se torne cheio, invertendo-se assim o lado.

Voltando ao quadro 5.2.2.A, a próxima movimentação gerada corresponde à carreta vinte e sete (célula A13). Sua localização pode ser avaliada através do Quadro 5.2.2.B, como presente na *ÁREA BUFFER* pilha 3, altura 1.

O acesso a este contêiner pode ser efetuado pelo lado esquerdo com onze movimentações ou pelo lado direito com cinco apenas. Seleciona-se o lado direito com quatro movimentações improdutivas relativamente aos contêineres 10, 24, 11 e 3 e uma produtiva, do contêiner 27.

Observe-se que o lado direito da pilha original modificada para o qual se deve transferir os contêineres indesejados encontra-se cheio. Assim sendo, transferem-se estes, excepcionalmente, para o lado esquerdo desta, obtendo as estruturas finais após duas movimentações apresentadas em 5.2.2.A.

### 5.2.3. A Estrutura do arquivo de movimentação de contêineres

Os seguintes campos são gerados a cada retirada de contêiner da estrutura em que se encontra:

- N° da Carreta que chegou (e conseqüentemente o contêiner correspondente a ser retirado da estrutura na qual se encontra);
- Quantidade de contêineres a serem movimentados na estrutura à qual pertence:
  - pela direita
  - pela esquerda
  - a identificação do lado de acesso mínimo
  - a quantidade mínima de acessos
- Altura das pilhas **ORIGINAL** e **BUFFER**: as seis de cada uma das duas estruturas. Informação necessária para garantir a adequada movimentação e o correto empilhamento de contêineres nas estruturas **ORIGINAL** e **BUFFER**.
- Localização do contêiner desejado:

- se está presente na estrutura **ORIGINAL** ou **BUFFER**,
- em qual pilha,
- a que altura.

O quadro abaixo apresenta a estrutura integral deste módulo para os 30 contêineres movimentados.

**Quadro 5.2.3.A**

Curteira	Nr.	Lado Acesso	Movimentação	Altura das Pilhas Originais (após a movimentação)						Altura das Pilhas Originais (antes da movimentação)						Altura das Pilhas Buffer (após a movimentação)						Altura das Pilhas Buffer (antes da movimentação)						original	buffer	coluna	altura	
				1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6					
12	14	5	DIR	5	5	5	5	4	0	5	5	5	5	5	4	0	0	0	4	0	0	0	0	0	0	0	1	0	5	5		
13	10	16	DIR	7	0	3	5	5	4	0	5	5	5	5	4	0	0	0	5	5	0	0	0	0	4	0	0	1	0	2	4	
14	30	0	DIR	0	0	3	5	5	4	0	0	3	5	5	4	0	0	0	5	5	0	0	0	0	5	5	0	0	0	2	4	
15	5	6	DIR	6	0	3	5	3	0	0	0	3	5	5	4	0	0	0	5	5	5	0	0	0	5	5	0	0	1	0	4	4
16	15	7	DIR	7	0	0	1	3	0	0	0	3	5	3	0	0	1	5	5	5	5	0	0	0	5	5	5	0	1	0	3	2
17	27	12	DIR	8	0	3	5	3	0	0	0	0	1	3	0	0	0	3	5	5	5	0	1	5	5	5	5	0	1	3	4	
18	21	11	DIR	3	0	0	5	3	0	0	0	3	5	3	0	0	0	5	5	5	5	0	0	0	3	5	5	0	1	0	2	1
19	9	3	DIR	3	0	0	5	0	0	0	0	0	5	3	0	0	0	5	5	5	2	0	0	0	5	5	5	0	1	0	4	1
20	26	15	DIR	3	0	2	5	0	0	0	0	0	5	0	0	0	0	2	5	5	2	0	0	0	5	5	5	2	0	1	3	3
21	8	3	DIR	3	0	2	5	2	0	0	0	2	5	0	0	0	0	2	5	4	0	0	0	2	5	5	2	0	1	5	5	5
22	28	8	DIR	1	0	1	5	2	0	0	0	2	5	2	0	0	0	2	5	4	0	0	0	2	5	4	0	1	0	2	2	
23	16	3	DIR	3	0	1	5	4	0	0	0	1	5	2	0	0	0	2	5	1	0	0	0	2	5	4	0	0	1	5	2	
24	23	8	DIR	5	0	0	1	4	0	0	0	1	5	4	0	0	0	1	5	5	1	0	0	0	2	5	1	0	1	0	3	2
25	24	9	DIR	4	0	0	4	4	0	0	0	0	1	4	0	0	0	2	5	1	0	0	0	1	5	5	1	0	0	1	3	3
26	7	7	DIR	1	0	0	4	4	0	0	0	0	4	4	0	0	0	1	5	1	0	0	0	2	5	1	0	0	1	3	2	
27	3	3	DIR	3	0	1	5	4	0	0	0	0	4	4	0	0	0	0	3	1	0	0	0	1	5	1	0	0	1	4	4	
28	22	2	DIR	2	0	1	5	2	0	0	0	1	5	4	0	0	0	0	3	2	0	0	0	0	3	1	0	1	0	4	3	
29	19	4	DIR	2	0	2	5	2	0	0	0	1	5	2	0	0	0	0	1	2	0	0	0	0	3	2	0	0	1	4	2	
30	6	6	DIR	6	0	0	1	2	0	0	0	2	5	2	0	0	0	1	5	2	0	0	0	0	1	2	0	1	0	3	2	
31	2	2	DIR	2	0	0	1	0	0	0	0	0	1	2	0	0	0	0	1	5	3	0	0	0	1	5	2	0	1	0	4	1
32	25	4	DIR	2	0	0	2	0	0	0	0	0	1	0	0	0	0	0	4	3	0	0	0	1	5	3	0	0	1	4	5	
33	20	4	DIR	1	0	0	2	0	0	0	0	0	2	0	0	0	0	0	3	3	0	0	0	0	4	3	0	0	1	4	4	
34	4	2	DIR	2	0	0	0	0	0	0	0	0	2	0	0	0	0	0	4	3	0	0	0	0	3	3	0	1	0	3	1	
35	17	6	DIR	3	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	4	3	0	0	1	4	2	
36	29	2	DIR	2	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	2	3	0	0	0	1	3	0	1	0	3	1	
37	12	1	DIR	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	2	3	0	0	1	5	3	
38	13	3	DIR	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	2	2	0	0	1	4	2	
39	18	2	DIR	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	2	0	0	1	5	1	
40	11	1	DIR	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	4	1	
41	1	1	DIR	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	4	1	



#### 5.2.4. A Fila de Transferência de Contêineres de movimentações improdutivas

Esta estrutura é responsável pela desmontagem de contêineres identificados em movimentações improdutivas, após tomada a decisão do lado de acesso à estrutura do contêiner desejado, e subsequente montagem destes, na estrutura diversa da qual foi retirado. É representada por uma coluna de 15 células em cada uma das pastas dos modelos.

#### 5.2.5. A última pilha ativa

Visa identificar a posição dos últimos contêineres externos (pelos lados direito e esquerdo) nas estruturas de pilhas **ORIGINAL** e **BUFFER**, a cada iteração. Essa informação é necessária para garantir que as pilhas sejam preenchidas até a quinta altura, sempre a partir do último contêiner presente na estrutura que recebe as movimentações improdutivas pelo lado acessado, sem que espaços sejam deixados entre pilhas na montagem.

Este conjunto de informações, gerado a cada nova chegada de carreta (e retirada de contêiner) permite uma adequada depuração em cada um dos modelos, garantindo assim que estes executem as tarefas a que se propõem, tal como descritas no item 5.2.2 acima.

### 5.3. Visão Geral do Modelo de Simulação

O Quadro 5.3.A abaixo representa a tela principal do aplicativo desenvolvido, automatizado, integrados os regimes desejados – **PEPS** e **MOVIMENTAÇÃO MÍNIMA**.

Quadro 5.3.A

SIMULAÇÃO DE REGRAS DE SEQUENCIAMENTO : PEPS (FIFO) X SPT

The interface is divided into several sections:

- PILHAS ORIGINAL**, **PILHAS ORIGINAL EM PROCESSAMENTO**, and **PILHAS BUFFER**: Each section contains a table with 6 columns labeled 'NÚMERO DA PILHA' (1-6) and 5 rows labeled 'A', 'L', 'T', 'U', 'R', 'A'.
- ORDEM DE CHEGADA DE CARRETAS**: A list of 30 numbered slots (01-30) for arrival order.
- ORDEM DE ATENDIMENTO DE CARRETAS**: A list of 30 numbered slots (01-30) for service order.
- PARÂMETROS DE ENTRADA**:
  - Nº Replicações**: Radio buttons for 100, 200, 300, 400, 500.
  - Regime de Atendimento**: Radio buttons for **PEPS (FIFO)** and **MÍNIMA MOVIMENTAÇÃO (SPT)**.
  - E[exponencial] (min/chegada)**: Radio buttons for 3,0, 3,5, 4,0.
  - Tempo médio de Serviço (movim/contêiner)**: Radio buttons for 2,2, 2,5, 2,8.
- VARIÁVEIS DE DESEMPENHO DO PROCESSO (tempos em minutos)**:
  - TEMPO TOTAL DE SIMULAÇÃO**: Input field.
  - QUANTIDADE DE MOVIMENTAÇÕES MÉDIA P/ACESSAR CONTÊINER**: Input field.
  - QUANTIDADE MÉDIA DE CARRETAS COM TEMPO DE ESPERA EM FILA**: Radio buttons for ranges: 0 < T < 30, 30 < T < 60, 60 < T < 90, 90 < T < 120, T > 120.
  - TEMPO MÉDIO ATENDIMENTO**: Input field.
  - TEMPO MÉDIO ESPERA EM FILA**: Input field.
  - TEMPO MÉDIO OCIOSIDADE**: Input field.
  - TAMANHO MÉDIO DA FILA**: Input field.
- PROCESSAR**: A blue button at the bottom.

São parâmetros de entrada deste aplicativo:

- o regime de atendimento,
- a quantidade de replicações desejada,
- $E[\text{exponencial}] = \text{tempo médio entre chegadas} = 1/\lambda$  e
- o tempo de serviço médio para movimentar um contêiner ( $\mu$  da distribuição  $N(\mu, \sigma)$ ).

Como pode ser observado nesse quadro, apresenta-se a estrutura padrão de pilhas em estudo (estrutura lógica de seis pilhas, com cinco contêineres em cada uma). As estruturas se encontram vazias inicialmente, sendo preenchidas sistematicamente, a cada iteração, após o fornecimento dos parâmetros de entrada.

Efetuada o processamento completo dos trinta contêineres, apura-se o desempenho do regime pré-definido através das seguintes variáveis:

- (1) o **tempo total de simulação**, variável-chave, passível de identificar se a janela estipulada de 2 (duas) horas é suficiente para garantir o atendimento às 30 (trinta) carretas (correspondentes aos 30 contêineres pré-segregados em noite anterior);
- (2) a **distribuição, por faixa de intervalos de tempo de 30 (trinta) minutos, da quantidade média de carretas em fila de espera**. Esta seria uma informação complementar necessária ao tempo **médio de espera em fila** e **tamanho médio da fila**, insuficientes *de per si*, para fornecer um quadro mais abrangente da composição da fila de espera e do tempo efetivamente dispendido em espera. Busca-se, em última análise, identificar níveis de satisfação de clientes associados ao tempo de espera em fila;
- (3) a **quantidade de movimentações média para acessar um contêiner** e o **tempo médio de atendimento a uma carreta**, como medidas clássicas de eficiência/desempenho da regra de seqüenciamento em estudo;
- (4) Finalmente, o **tempo médio de ociosidade do equipamento**, medida de desempenho do equipamento de movimentação de contêineres, praticamente relacionada à identificação de períodos de não-chegadas de carretas, associados à não-existência de fila de espera.

Com este leque de variáveis de saída, pretende-se obter uma avaliação abrangente do desempenho relativo dos regimes e cenários desejados.

## 5.4. Premissas operacionais utilizadas na modelagem

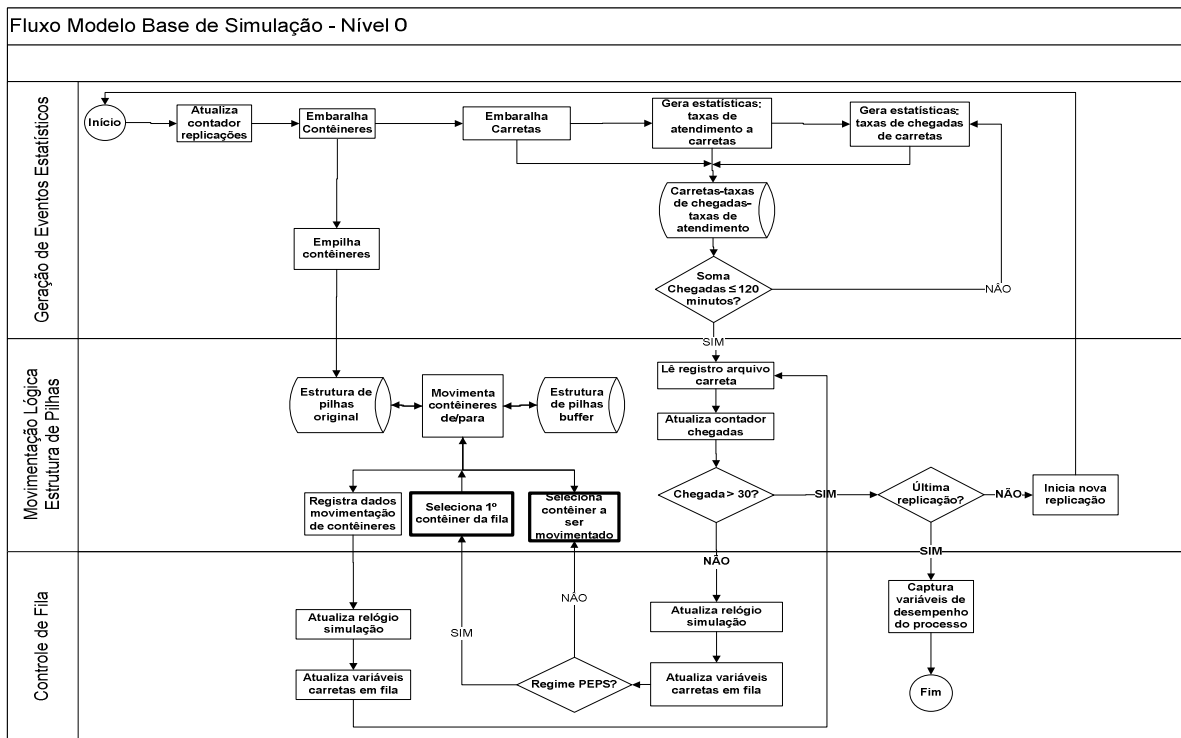
Algumas premissas operacionais utilizadas na modelagem e dignas de registro são:

- Toda pilha deve ser preenchida até a quinta altura, antes de se preencherem as pilhas seguintes;
- Esgotado o espaço lógico disponível de preenchimento do lado da estrutura para onde os contêineres foram transferidos – **ORIGINAL** ou **BUFFER**, passa-se a preencher o lado oposto da mesma estrutura;

Pretende-se assim reproduzir as regras de operação do *reach stacker* em ambiente de programação.

## 5.5. Fluxo do modelo base de simulação para os dois regimes - PEPS e MOVIMENTAÇÃO MÍNIMA

O diagrama abaixo apresenta o fluxo do modelo de simulação, nível 0, para ambos os regimes.



## 5.6. Modelo lógico do regime PEPS (1º a entrar, 1º a sair)

Em linhas gerais, apresenta-se o modelo lógico do regime PEPS.

1. é gerada a chegada de carreta,
2. atualiza-se o relógio de simulação,
3. atende-se a carreta,
4. atualiza-se novamente o relógio,
5. identifica-se se há formação de fila de carretas em espera e suas identidades,
  - caso haja, atualiza-se seus tempos de espera em relação a suas chegadas, atende-se ao primeiro da fila e atualiza-se novamente o relógio e os tempos dos presentes na fila
  - se não há, mensura-se o tempo de ociosidade até a chegada de nova carreta, com sua chegada atualiza-se o relógio, atende-se a esta e atualiza-se novamente o relógio
6. Retorna-se a (5) efetuando estas operações até esgotar a totalidade dos pares carreta-contêiner a serem atendidos.

## 5.7. Modelo lógico do regime MOVIMENTAÇÃO MÍNIMA

No regime **MOVIMENTAÇÃO MÍNIMA**, também em linhas gerais, tem-se:

1. é gerada a chegada de carreta,
2. atualiza-se o relógio de simulação,
3. atende-se a carreta,
4. atualiza-se novamente o relógio,
5. identifica-se se há formação de fila de carretas em espera e suas identidades,
  - caso haja, atualiza-se seus tempos de espera em relação a suas chegadas, procede-se à seleção do contêiner de menor movimentação

dentre todos estes e seu atendimento. A carreta correspondente a este contêiner será a selecionada (próxima a ser atendida) então, pois representa o contêiner que gera o mínimo de movimentações, atualiza-se novamente o relógio e os tempos dos presentes na fila

- se não há, mensura-se o tempo de ociosidade até a chegada de nova carreta, com sua chegada atualiza-se o relógio, atende-se a esta e atualiza-se novamente o relógio
6. Retorna-se a (5) efetuando estas operações até esgotar a totalidade dos pares carreta-contêiner a serem atendidos.

Importante frisar que o processo de atualização dos diversos status de cada carreta, além da recém-atendida, é necessário para garantir a correta mensuração das variáveis de interesse.

O mecanismo de avanço de tempo, como pode ser observado acima, dá-se por evento, tal qual proposto por Law e Kelton (1991).