

Prof. Gerson Luiz Camillo

Notas de Aula
em
Modelagem e Simulação

Assuntos: aspectos de modelagem discreta; distribuições de probabilidades; geração de números aleatórios; aspectos em teoria de filas; experimentação com Arena (modelagem dados amostrais; modelagem DES; análise das saídas)

Universidade Federal de Santa Catarina – UFSC

Araranguá

2022/1

Lista de ilustrações

Figura 1 – Fila de check-in em aeroporto.	13
Figura 2 – Período transiente seguido pelo período transitório.	20

Sumário

1	INTRODUÇÃO	4
2	DISTRIBUIÇÕES DE PROBABILIDADE	6
2.0.1	Propriedades de memória	6
2.1	Números Aleatórios	7
3	TERMINOLOGIA BÁSICA SOBRE TEORIA DE FILAS	10
4	EXPERIMENTAÇÃO E ANÁLISE DE RESULTADOS	13
4.1	Delineamento do modelo	13
4.2	Amostragem de dados e sua análise	14
4.2.1	Análise e uso da estatística descritiva	15
4.2.2	Inferência e testes de aderência	16
4.2.2.1	Testes no Arena INPUT ANALYZER	17
4.3	Modelagem e experimentação com Arena	19
4.4	arena-output	21
4.5	Análise de Sistemas Terminais	24
4.5.0.1	Validação e verificação de modelos	28
	REFERÊNCIAS	30

1 Introdução

Este é um material complementar (notas de aula) referentes à disciplina de Modelagem e Simulação. Suplementa as aulas e o material de referência, no caso a bibliografia básica do curso (FILHO, 2008) e materiais disponíveis no Moodle do curso.

Um experimento de simulação ocorre quando o modelador ajusta parâmetros de entrada para um modelo e executa a simulação.

Tipos e métodos de simulação:

- Sistemas dinâmicos;
- Sistemas estáticos: não há evolução no tempo; usam o método de Monte Carlo para realizar muitas replicações de estado e com isso evidenciar o comportamento médio;
- Baseado em agentes;
- Modelagem de eventos discretos (DES - *Discrete Event Simulation*)

Passos e principais fases de uma simulação discreta:

- **Modelagem conceitual** para identificar as entidades, atributos, recursos e filas;
- Modelagem dos dados de entrada: coleta, análise e identificação das propriedades estatísticas (processos de aderência a distribuições de probabilidade) (**input**);
- **Implementação computacional** da modelagem através de programas e/ou linguagens adequadas ao tipo de processo (discreto vs contínuo) e respectiva *verificação*;
- Análise dos dados de saída e base nos resultados, de forma iterativa, verificar a necessidade de replicações (e seu número);
- Análise dos resultados (**output**).

A validação do modelo procura verificar a consistência das abstrações e simplificações criadas e a realidade do mundo físico. Logo, a **validação conceitual** de um modelo necessita da interação de todos os interessados e dos que trabalham na realidade física para ajustar a corretude da **lógica** e do **comportamento** do modelo.

De forma geral, os elementos de um sistema de simulação discreta (DES - *Discrete Event Simulation*) são:

- Entidades (*Entities*): elementos que seguem através do modelo (carros em estação de lavagem, clientes em fila de banco);
- Atributos (*Attributes*): características específicas de cada entidade (capacidade da estação);
- Recursos (*Resources*): elementos demandados pelas entidades (operador serviço de limpeza, máquinas, pessoal);
- Filas (*Queues*): filas de espera por recursos.

2 Distribuições de Probabilidade

Distribuições de probabilidade estatísticas. Parâmetros e propriedades principais:

- X variável aleatória;
- $f(x)$ função densidade de probabilidade de uma variável X ;
- $P(E)$ probabilidade de um evento E ;
- Não existem probabilidades negativas;
- A integração da função densidade é igual a 1;
- A probabilidade de X estar dentro de um intervalo a, b é dada pela integral definida de $f(x)$.

Propriedades cumulativas: a probabilidade cumulativa de uma variável aleatória X é: a probabilidade de observar igual ou menor que x : $P(X \leq x)$

Função densidade de probabilidade: é um parâmetro de probabilidade de distribuições contínuas, pelo seguinte motivo. Considere-se que a temperatura varie entre 10 e 40°C. Qual seria a probabilidade de se medir uma temperatura de exatamente 32,576°C? Não é uma questão válida. Então, há necessidade de trabalhar com intervalos de valores. Para este caso, se pergunta qual a probabilidade de encontrar uma temperatura igual ou menor que 32,576 graus centígrados?

2.0.1 Propriedades de memória

Antes de apresentar brevemente as características básicas de algumas distribuições de probabilidade, será definido o que seja propriedade de memória, importante para os processos de simulação discreto de filas.

Propriedade sem memória: uma distribuição de probabilidade estatística na qual a probabilidade de algum evento futuro não é afetada pela ocorrência de eventos passados. Duas distribuições de probabilidade têm essa propriedade:

- Distribuição exponencial com números reais não negativos; e
- Distribuição geométrica com inteiros não negativos.

Ambas as distribuições de probabilidade são usadas para modelar a quantidade de tempo esperada antes que algum evento ocorra, logo, importantes em modelos de filas.

Exemplo para um **sistema que não possui a propriedade de sem memória**: uma determinada marca de notebooks possui vida média 6 anos antes que apresente uma falha. Então, caso se possua um mesmo modelo que tenha 5 anos, o tempo esperado até que ele falhe é bastante curto. No entanto, se outro laptop tiver apenas 1 ano de idade, o tempo esperado será bastante longo. Neste exemplo, saber a quantidade de tempo que se passou durante a vida útil nos informa por quanto tempo ainda continuará funcionando até falhar. Assim, essa distribuição de probabilidade não teria uma propriedade de sem memória.

Exemplo para um **sistema que com propriedade de sem memória**: um proprietário de uma loja quer saber quanto tempo terá que esperar até que o próximo cliente entre. Neste exemplo, saber quando o último cliente entrou na loja não é realmente útil para prever quando o próximo cliente entrará porque cada cliente é independente e exibe um comportamento individual. Assim, essa distribuição de probabilidade teria uma propriedade sem memória. Em outras palavras, a probabilidade de algum evento futuro ocorrer não é afetada pela ocorrência de eventos passados.

Outro exemplo: suponha que uma média de 30 clientes por hora entrem em uma loja e o tempo entre as chegadas seja distribuído exponencialmente. Em média, decorrem 2 minutos entre visitas sucessivas. Suponha que 10 minutos se passaram desde a entrada do último. Como esse é um período de tempo extraordinariamente longo, parece mais provável que um cliente chegue no próximo minuto. No entanto, como a distribuição exponencial tem uma propriedade sem memória, esse não é o caso. O tempo gasto esperando o próximo cliente chegar não depende de quanto tempo passou desde que o último cliente chegou.

O objetivo de identificar os tipos de distribuição é para facilitar o cálculo da **probabilidade, valor esperado e variância** de uma amostra. As distribuições de probabilidade são necessárias para modelar processos estocásticos do mundo real sujeitos à modelagem em eventos discretos.

2.1 Números Aleatórios

A geração de números pseudo-aleatórios usando computador (determinístico) deve possuir duas características básicas:

Independência: em processos estocásticos, dois eventos são independentes se o resultado de um evento não influencia o resultado de outro. Portanto, não há uma estrutura aparente na sequência. Por exemplo: numa sequência de valores pseudo-aleatórios, o conhecimento de um conjunto de valores não revela qualquer outro valor, seja do passado, quanto do futuro (sequências ainda por gerar).

Identicamente distribuída: ou seja, a sequência deve seguir a distribuição uniforme.

Uma das medidas para quantificar a quantidade de informação ou de incerteza é a **entropia**. Igual à informação média ou, por outro lado, a incerteza média removida.

Em computação, números aleatórios são gerados por processos denominados de geradores de números pseudo-aleatórios (*pseudo-random number generators* – PRNG). Os algoritmos PRNG são inicializados por algum valor inicial aleatório (*unpredictable*) e devem possuir algum nível de entropia. Esse valor é conhecido como semente (*seed*). Ela determina a segurança dos valores gerados. O processo se resume a uma função de expansão da aleatoriedade de um conjunto de valores iniciais aleatórios em um conjunto maior de valores pseudo-aleatórios. Logo,

- O tamanho da semente é fator fundamental de segurança, pois de N bits da semente não é possível aumentar a entropia para mais de N bits usando funções determinísticas.
- O processo de obtenção da semente deve ser através de processos físicos realmente aleatórios.

Obs.: em criptografia se usam PRNG seguros, conhecidos como CSPRNG (https://en.wikipedia.org/wiki/Cryptographically-secure_pseudorandom_number_generator), que combinam aleatoriedade da semente com alto nível de entropia com técnicas de geração de bits que usam blocos de construção criptográficos. Exemplos desses blocos são as funções hash (https://en.wikipedia.org/wiki/Cryptographic_hash_function) e algoritmos de criptografia simétricos de bloco (https://en.wikipedia.org/wiki/Block_cipher).

Testes: quando não é possível determinar a qualidade de aleatoriedade do processo de geração (verdadeiramente aleatório vs PRNG), devem ser usados métodos para testar as sequências de valores (números ou bits) quanto a determinadas propriedades/estatísticas:

- Cada bit deve ter probabilidade de exatamente igual a 0,5. Se forem números reais, então devem estar distribuídos uniformemente entre 0 e 1.
- Testes de frequências simples, dupla, múltipla e de blocos. Ou seja, se quer verificar se sequências de valores não possuem padrão dentro da sequência completa.

Logo, testes não provam que uma sequência é ou não aleatória, mas que os valores possuem propriedades que passaram em avaliações probabilísticas.

Método Congruente Linear (MC): a partir de três parâmetros a , b , m (números inteiros grandes e primos) e de uma semente aleatória, a sequência gera um número de forma recorrente dentro da faixa de valores de m ($0 \leq m - 1$), conforme a seguinte fórmula:

$$x_{i+1} = (a \cdot x_i + b) \bmod(m)$$

Exemplo: dado que $a = 8523872351$, $b = 7865833$, $m = 786843$ e a semente $x_0 = 13$, será gerada uma sequência de valores entre 0 e 786443, com boas propriedades de distribuição, considerando que o uso de um valor de m primo. Para verter o conjunto de valores inteiros em números de ponto flutuante no intervalo entre 0 e 1, pode-se usar a seguinte fórmulação:

$$0 \leq R_i = x_i/m \leq 1$$

Obs.: o método de congruência linear pode gerar uma sequência identicamente distribuída, mas os valores não são independentes. Outra limitação: só gera valores dentro de uma faixa limitada, dada por m , e de forma cíclica.

Para gerar um valores que seguem uma distribuição de probabilidades a partir de uma sequência de números uniformemente distribuídos entre 0 e 1, pode ser usado método da função inversa. Por exemplo, para gerar valores y_i que seguem a distribuição exponencial de parâmetro β , considerando um conjunto de números R_i aleatórios e uniformemente distribuídos:

$$y_i = -\left(\frac{1}{\beta}\right) \cdot \ln(R_i)$$

3 Terminologia Básica sobre Teoria de Filas

Sistemas de filas compreendem processos servidores que atendem entidades, as quais esperam em áreas denominadas de filas. Têm-se, portanto, processos de chegadas e processos de atendimento. A bibliografia de referência explora esse assunto no Capítulo 8 (FILHO, 2008). Estudos de modelagem procuram identificar:

- As variáveis das entidades;
- As distribuições probabilísticas de chegada e de atendimento das entidades.

Usa-se a teoria de filas para tratar o comportamento estocástico das variáveis associadas aos processos de chegada e de atendimento (ou tempo de serviço). Observação importante: nos processos de modelagem estacionários (visto a seguir), as taxas de chegadas e atendimento relacionadas às respectivas distribuições de probabilidade, são constantes ao longo do tempo da simulação. As variáveis assumem valores estocásticos conforme a distribuição, mas sua taxa é constante.

Principais informações:

- Tempo (número) médio de espera das entidades;
- Ocupação média dos processos servidores;
- Probabilidade de formação de filas.

Modelos e Notação de Filas:

- **Processos de chegadas:** taxa média de chegada de entidades dado por λ (distribuição de probabilidades dos períodos de **tempo entre chegadas**);
- **Processos de atendimento:** taxa média de atendimento (serviço) dada por μ ¹ (distribuição de probabilidades dos períodos de **tempo de serviço**);
- **Número de servidores:** quantidade de servidores disponíveis;
- **Capacidade da fila** tamanho da fila de espera; normalmente finito em sistemas reais; em sistemas computacionais são os *buffers* de entrada;
- **Tamanho da população:** número total de entidades a serem servidas; normalmente infinito;

¹ μ também representa média populacional em estatística descritiva.

- **Disciplina de atendimento:** o processo de atendimento tem uma disciplina de serviço (critério): FIFO (First-in, First-out); LIFO (Last-in, First-out); LVF (Attribute ID: serviço ordenado por valores crescentes); HVF (Attribute ID: serviço ordenado por valores decrescentes); entre outros.

A notação de Kendall estabelece a seguinte forma: **A/S/m/B/K/SD**, ou seja, **Distribuição dos tempos das chegadas/Distribuição dos tempos de serviços/Número Servidores/Capacidade Fila/Tamanho População/Disciplina Atendimento**.

Tipos de distribuições de probabilidades:

- M Exponencial (M que deriva do inglês: *Memoryless*);
- E_k Erlang com parâmetro k ;
- H_k Hiperexponencial com parâmetro k ;
- D Determinística; e
- G Geral

Por exemplo: $M/M/1/\infty/\infty/FIFO$ informa que: (M/M) TEC e TS com distribuição exponencial, (1) uma unidade de serviço; (∞/∞) capacidade da fila e tamanho da população infinitos; e, critério de atendimento do primeiro a chegar. Resumidamente, pode ser abreviado por $M/M/1$.

Um modelo possui **condição de estabilidade** quando $\lambda \leq m \cdot \mu$, ou seja, a taxa de chegadas é menor que a taxa de atendimento versus a quantidade de servidores. Para facilitar o processo de modelagem, os problemas são considerados em **estado estacionário**, quando não dependem de uma condição inicial. Por outro lado, sistemas transitórios tratam filas com condição inicial, como por exemplo, quando um banco abre e há um acúmulo de clientes em espera.

Considerando um sistema de fila simples ($M/M/1$), as principais variáveis associadas são (conforme nomenclatura [Filho \(2008, p. 337\)](#)):

- τ tempo entre chegadas, isto é, tempo decorrido entre duas chegadas sucessivas (referenciado também como TEC);
- λ a taxa **média** de chegadas que é igual a $1/E(\tau)$;
- s tempo de serviço por cliente (referenciado também por TS);
- μ é a taxa de média de serviço do servidor que é igual a $1/E(s)$.

Obs.: uma taxa de serviço μ seguir uma distribuição exponencial, então, que é designado por β . $E(var)$ indica esperança da variável var ou a média da mesma.

Algumas medidas de avaliação da performance do modelo

- $T_{fila} = \frac{\sum \text{tempos-espera-na-fila}}{\text{num-total-clientes}}$ Tempo médio em fila
- $T_{sist} = \frac{\sum \text{tempos-no-sistema}}{\text{num-total-clientes}}$ Tempo médio dispendido no sistema
- N_{fila} Número médio de entidades em fila
- N_{sis} Número médio de entidades no sistema
- $P(n)$ Probabilidade de existirem n entidades no sistema
- $P_{fila} = \frac{\sum \text{num-clientes-que-esperam}}{\text{num-total-clientes}}$ Probabilidade de um cliente esperar na fila
- $P_{recurso} = \frac{\sum \text{tempo-livre-recurso}}{\text{tempo-total-simulacao}}$ Percentual de não ocupação do recurso (operador, atendente, etc). A ocupação é dada por: $1 - P_{recurso}$
- ρ Variável que indica o percentual de ocupação do recurso.

No Arena, as variáveis dependentes de tempo são genericamente classificadas como *Tally variables*.

A teoria de filas pode ser tratada de forma analítica ou por simulação. Quando há muitas variáveis estocásticas envolvidas, a simulação por eventos discretos pode acomodar melhor as soluções de diversos problemas.

4 Experimentação e Análise de Resultados

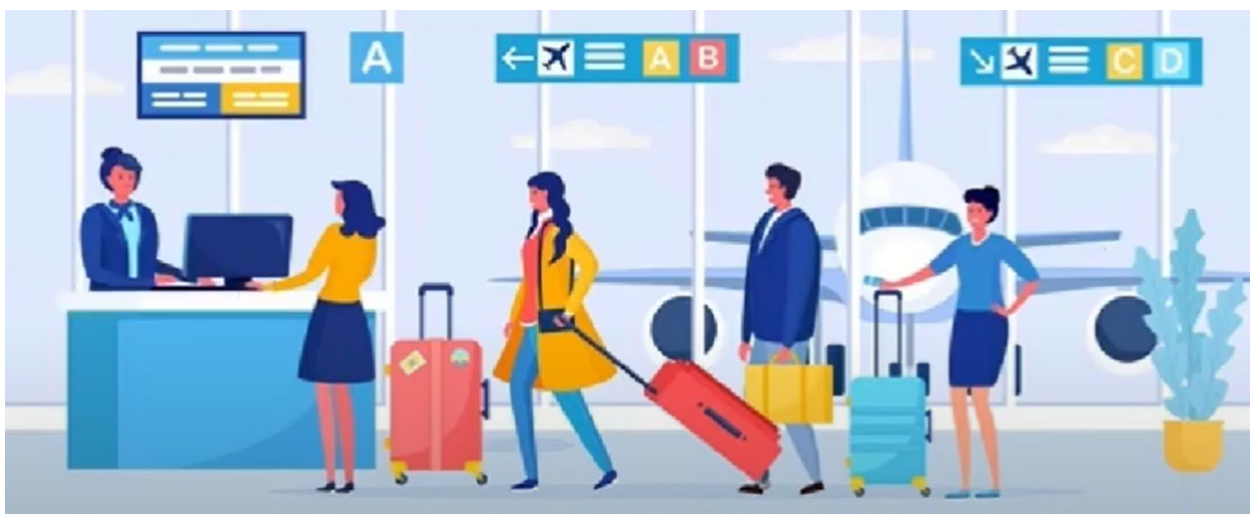
Uma simulação discreta usando computador segue uma conjunto de passos ou tarefas padrões, quais sejam:

1. Criação do modelo de um processo do mundo real, discretizando e identificando os componentes (no nível de abstração adequados);
2. Amostragem e análise dos dados de entrada do modelo;
3. Execução do modelo, identificando os tempos de execução e o número de replicações;
4. Análise dos dados das rodadas e identificação do nível de confiança dos resultados que determinará julgamentos quanto à modificação de algum aspecto da modelagem.

4.1 Delineamento do modelo

A criação de um modelo para simulação discreta (DES) envolve identificação dos elementos que compõem/determinam o escopo da Teoria das Filas aplicada a um determinado problema do mundo real. A Figura 1 apresenta uma situação do mundo real que se deseja modelar usando simulação discreta.

Figura 1 – Fila de check-in em aeroporto.



Trata-se de uma fila de check-in em aeroporto, no qual passageiros esperam para serem atendidos junto a um atendente da companhia aérea. Os seguintes elementos podem ser identificados:

- Entidades: as pessoas na fila para check-in e embarque;
- Recursos: o balcão de atendimento.

Por exemplo: numa fila de check-in de aeroporto, os passageiros são as entidades, o/a atendente é um recurso, e a fila de espera por atendimento. A fila pode ter o atributo de tamanho máximo. O recurso (balcão de atendimento) pode ser composto ou discretizado conforme segue, atendendo o **nível de abstração** desejado:

- (um recurso): o próprio atendente;
- (dois recursos): o atendente e o computador;
- (três recursos): o atendente; o computador (hardware e o sistema operacional); e, o computador (sistema de reservas da companhia).

4.2 Amostragem de dados e sua análise

Esta fase é descrita como modelagem de dados e consiste basicamente em:

- Amostragem e coleta de dados do sistema sendo modelado;
- Análise e tratamento dos dados tomando por base a estatística descritiva; e,
- Processo de inferência (especificamente, encontrar uma distribuição de probabilidade que descreva o comportamento dos dados).

Além da identificação da distribuição de probabilidade, há necessidade de encontrar/estimar os parâmetros da mesma. E, qual o propósito de encontrar uma distribuição de probabilidade? A resposta é que sistemas do mundo real apresentam características/comportamentos que, apesar de estocásticos, acabam seguindo um padrão de distribuição estatístico. A execução de um modelo em um sistema DES requer que sejam gerados dados e números aleatórios seguindo um determinado padrão e, esse é o mesmo padrão de comportamento dos dados reais.

O propósito desta etapa é identificar se os processos de chegada à fila e tempos de serviço seguem alguma distribuição estatística (uniforme, normal, Poisson, exponencial, Erlang, Gama, Weibull, Beta, etc). A importância desta informação reside no fato de que o modelo vai gerar valores aleatórios seguindo esta distribuição.

Amostragem e coleta de dados é uma fase importantíssima, pois a coleta de dados não seguindo as premissas estatísticas de amostragem aleatória, pode dificultar a identificação da distribuição e/ou apontar para uma distribuição distante da realidade. Em qualquer caso, a simulação e seus resultados estarão comprometidos e poderão gerar

decisões errôneas. O ponto principal em amostragem: deve ser **representativa** da população (processo) de interesse da modelagem. Por isso, sugere-se que se consulte bibliografia adequada que trate dos requisitos para processos de amostragem.

4.2.1 Análise e uso da estatística descritiva

Essa fase compreendem o uso da estatística descritiva para apresentar e analisar os dados. O processo se inicia com um histograma dos dados e o cálculo de medidas descritivas (ou de posição) como média, mínimo, máximo, mediana e moda, além das medidas de dispersão, como amplitude, coeficiente de assimetria, variância e desvio padrão¹.

Podem ser usados pacotes estatísticos como R (<https://cran.r-project.org/>) e também a ferramenta complementar ao Arena conhecida como INPUT ANALYZER. Esta última é própria para este processo pois o resultado (distribuição e seus parâmetros) pode ser usado como entrada para o modelo sendo executado no Arena.

O processo de inicia com a plotagem do histograma de frequências (dados discretos ou contínuos) e identificação visual da distribuição. Essa fase pode levantar suposições sobre questões acerca dos dados. Um histograma que não segue aparentemente uma distribuição de probabilidade conhecida, pode indicar que: o sistema do mundo real é caracterizado por uma grande variabilidade; os dados coletados podem caracterizar situações (processos) distintos; e, ou problemas na amostragem (como tamanho insuficiente).

Na criação de um histograma há duas formas de elaborar, considerando se tratar de dados discretos ou dados contínuos. Para o primeiro caso, são contados os números de ocorrências de cada valor e a partir dos mesmos é feita a plotagem direta. No diagrama é relacionado o valor (eixo x) pelo respectivo número de ocorrências (eixo y).

Para o caso contínuo, não vale o procedimento acima, de forma que há necessidade de dividir os valores em faixas denominadas de **classes** ou **intervalos**. O primeiro método é por meio da raiz quadrada do número total de observações (N), conforme segue:

$$K = \sqrt{N}$$

Também se pode usar a regra de **Sturges** para determinar as quantidades de classes, na qual N é a frequência total, ou seja, a quantidade total de observações, e K é a informação de número de classes:

$$K = 1 + 3,322 \times \log N$$

Por exemplo, caso o número de observações seja de 1000 valores, então o número de classes estimados é:

¹ O desvio padrão é a raiz quadrada positiva da variância, logo, enquanto S^2 é variância, S é o desvio padrão (amostral)

$$K = 1 + 3,322 \times \log N = 1 + 3,322 \times \log 1000 = 1 + 3,322 \times 3$$

$$K = 10,966$$

No caso, a escolha recai por 11 classes para dividir a amplitude dos dados (diferença entre o valor mínimo e máximo). Em qualquer caso, a aproximação é pelo inteiro superior.

A amplitude das classes é dada por:

$$amplitude = \frac{valor_max - valor_min}{num_classes}$$

Além das etapas acima, alguns pontos merecem atenção do analista. Considerando que os dados foram corretamente amostrados, ainda podem aparecer valores extremos, conhecidos como **outliers**. Esses **dados extremos não** são tratados pelo programa INPUT ANALYZER. Logo, pode ser necessário a retirada dos mesmos do rol de dados.

Outra questão trata da análise da série de dados considerando sua **evolução temporal**. Para isso, os dados devem ser amostrados em ordem. A análise da série temporal pode indicar correlações entre os próprios dados ou entre os dados e alguma função de horário/dia/etc. O exemplo desta situação indica a necessidade de configuração de regimes transitórios de operação.

4.2.2 Inferência e testes de aderência

Processo de encaixe (do inglês, **fit**) faz com que o Arena procure a ajustar a distribuição de probabilidade que melhor adere ao padrão dos dados. O programa INPUT ANALYZER (pacote do Arena) apresenta dois testes estatísticos não-paramétricos, que são baseados nas seguintes hipóteses:

- Hipótese Nula (H_0): a variável aleatória **segue** a distribuição sob hipótese com o(s) parâmetro(s) estimado(s); ou seja, os dados aderem à distribuição em teste;
- Hipótese Alternativa (H_1): a variável aleatória **não segue** a distribuição sob hipótese com o(s) parâmetro(s) estimado(s); ou seja, os dados não aderem à distribuição em teste.

Um teste estatístico (ou teste de significância) serve para verificar se os dados (amostra) fornecem evidência suficiente para aceitar como verdadeira a hipótese nula (ou hipótese de trabalho). O INPUT ANALYZER dispõe de dois testes estatísticos: o Chi Quadrado e o Kolmogorov-Smirnov. Os resultados deverão fornecer evidências suficientes para *aceitar* ou *rejeitar* H_0 , com base no nível de significância desejado.

O valor-p (*p-value*) é uma medida de significância estatística obtida a partir de dados amostrais. O valor-p é o menor nível de significância para rejeitar a hipótese nula. O teste de valor-p pode ser usado como critério da seguinte forma: se o seu valor for abaixo do valor crítico estabelecido pelo nível de significância (0,1, 0,05 ou 0,01), então há evidências para rejeitar H_0 , ou seja, a distribuição não adere aos dados coletados. O princípio é que: **se o valor p calculado for maior que o nível de significância, então isso não constitui prova de que a H_0 é verdadeira, mas que não há evidência para rejeitá-la.** Alguns valores de significância e nível de decisão:

Valor p obtido do teste	Interpretação
$0.01 \leq p\text{-value} < 0.5$	evidência forte contra H_0
$p\text{-value} \leq 0.01$	evidência moderada contra H_0
$0.05 \leq p\text{-value} < 0.1$	evidência potencial contra H_0
$0.1 < p\text{-value}$	evidência fraca ou inexistente contra H_0

Tabela 1 – Interpretação possível considerando os testes estatísticos Qui e KS para aderência dos dados a uma distribuição e conforme nível de significância requerido

Associados ao teste de hipótese tem-se a questão dos erros que se comete nas decisões baseadas no mesmo: erro tipo I (α) é quando se rejeita a hipótese nula, mas ela é verdadeira; e, erro tipo II (β) ocorre não se rejeita a hipótese nula, mas ela realmente é falsa.

4.2.2.1 Testes no Arena INPUT ANALYZER

Seguem dois exemplos para ilustrar o processo de decisão quanto à aderência de distribuições de probabilidade aos dados amostrais:

Distribuição: Normal

Expressão: NORM (100, 9)

Erro quadrático: 0.000173

Test do Chi-Quadrado

Número de intervalos = 32

Graus de liberdade = 29

Teste Estatístico = 0.00902

P-value correspondente = > 0.75

Kolmogorov-Smirnov Test (KS)

Teste Estatístico = 0.00902

P-value correspondente > 0.15

Sumário dos Dados

Número de Pontos de Dados	= 500
Valor Min nos Dados	= 62.8
Valor Max nos Dados	= 134
Média da Amostra	= 100
Desvio Padrão da Amostra	= 9.96

Sumário do Histograma

Intervalo do Histograma	= 62 to 134
Número de Intervalos	= 40

Observações:

- Graus de liberdade está associado/ajustado ao número de parâmetros estimados.
- A **aplicação do teste Qui-quadrado** requer que a amostra tenha ao menos 25 observações e pelo menos cinco classes (FILHO, 2008, p. 193).
- O teste de **K-S** não se aplica a dados de distribuições discretas, como a Poisson (FILHO, 2008, p. 200).

Portando, considerando que $\text{valor-p} \leq 0,1$ (para nível de significância de 90%), e valor de teste menor que o valor crítico de 5,99, a hipótese nula não é rejeitada, e pode-se considerar a distribuição normal para os dados. A expressão $NORM(100,9)$ pode ser usada no Arena como parâmetro de geração de aleatoriedade.

O próximo exemplo é de uma distribuição de 100 números com média amostral de 6,85 e desvio padrão de 6,6. O teste do Chi-Quadrado reportou o valor de teste igual a 1,43. Consultando a tabela, para um nível de significância de $\alpha = 0,05$ e com 3 graus de liberdade, o valor crítico é igual a 7,81². Como o valor do teste é menor que o valor crítico, este teste não rejeita H_0 . Verificando também o valor de p (p-value), vê-se que é maior que 0.05 para o nível de confiança de 95%.

Distribuição: Exponencial

Expressão: $-0.001 + EXP0(6.85)$

Erro quadrático: 0.002184

² Região de aceitação: valor do teste estatístico abaixo do valor crítico (conforme tabela própria que tem como parâmetros nível de significância e graus de liberdade).

Test do Chi-Quadrado

Número de intervalos = 5

Graus de liberdade = 3

Teste Estatístico = 1.43

P-value correspondente = 0.703

Sumário dos Dados

Número de Pontos de Dados = 100

Valor Min nos Dados = 0

Valor Max nos Dados = 43

Média da Amostra = 6.85

Desvio Padrão da Amostra = 6.6

Sumário do Histograma

Intervalo do Histograma = -0.001 to 43

Número de Intervalos = 8

O teste estatístico de Chi-Quadrado apresentou evidências para aceitar a hipótese nula e, dessa, forma confirmar a distribuição e seu parâmetro: $-0.001 + EXPO(6.85)$ (distribuição exponencial de média 6,85). No Arena, a expressão resultante que pode ser inserida é $EXPO(6.85)$, uma vez que o valor de $-0,001$ é pouco significativo e resulta da organização do histograma pelo INPUT ANALYZER.

4.3 Modelagem e experimentação com Arena

Os três principais módulos de uma simulação (Arena) são:

- CREATE: cria as entidades no modelo, seguindo um determinado parâmetro: constante, schedule ou uma expressão seguindo determinada distribuição estatística;
- PROCESS processo ou serviço prestado
- DISPOSE: módulo que retira as entidades do sistema.

Outro módulo importante, é o ASSIGN, que atribui valores aos atributos das entidades que circulam no sistema. Para uma simulação de servidor único, do tipo $M/M/1$, os parâmetros padrões são:

- Taxa de chegada: normalmente seguindo a distribuição de Poisson com média λ (número de entidades que chegam num período de tempo)

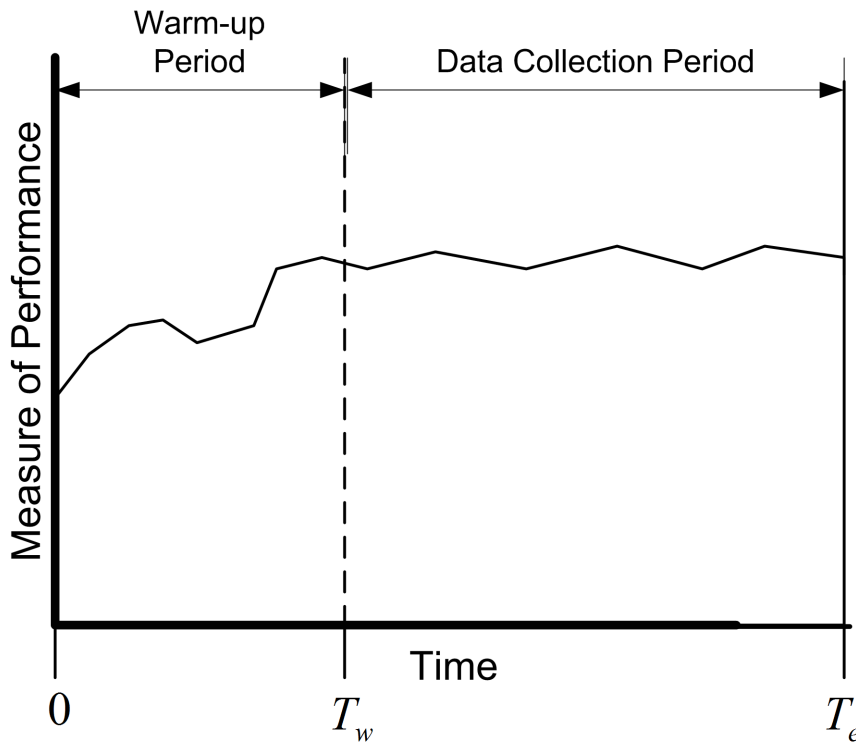
- Taxa de atendimento: normalmente seguindo a distribuição exponencial negativa com média $\beta = \frac{1}{\mu}$
- Regra de atendimento FIFO (primeiro a chegar, primeiro a ser atendido)

Obs.: a média β da distribuição exponencial é o inverso da distribuição de Poisson.

Os sistemas podem ser classificados como terminais e não-terminais. **Terminais** são caracterizados por um estado inicial vazio e a duração do período de execução (rodada) é a mesma do período de funcionamento do sistema do mundo real (sistema se reinicia). Exemplos: posto de lavagem, agência bancária, lojas, ou, qualquer sistema que não funcione durante 24 horas por dia. **Sistemas não-terminais** são aqueles em que o sistema não se reinicia, ou seja, funcionam 24 horas por dia. Exemplos: fábricas, portos e aeroportos (24h), etc. A discussão a seguir será a análise sobre sistemas terminais.

Outra característica importante e que afeta questões de performance são os períodos em que o sistema se encontra em um **estado transiente** (regime transitório) ou **estado estacionário** (permanente). Por exemplo, um posto médico que abre com uma fila de espera. A Figura 2 mostra este período transiente inicial e, após, o sistema entrando no seu estado de regime permanente.

Figura 2 – Período transiente seguido pelo período transitório.



Fonte: (ROSSETTI, 2021)

A importância em reconhecer esses períodos é que as estatísticas do funcionamento médio do sistema podem ser afetadas. Ele está presente tanto em sistemas terminais quanto não-terminais. Em sistemas não-terminais, esse período pode se tornar não significativo em relação ao processo geral quando se tem tempos de corrida longos. Já em sistemas terminais, o período transiente pode ser significativo, de forma que sistemas DES podem identificá-lo como período de *warm-up* (ou tempo de aquecimento).

Então, considerando **sistemas terminais**, o período transiente deve ser identificado e, na medida do possível, tratado e/ou descartado, conforme algumas técnicas básicas a seguir e também conforme apresentado em [Filho \(2008, p. 247\)](#):

- Simular por um tempo mais longo;
- Eliminar o período transiente;
- Inicializar o sistema já em estado estacionário, ou seja, com as variáveis em seus valores médios.

Considerando a opção de remoção do período transiente, no Arena (e outros DES) é possível especificar um período de *warm-up* no setup da rodada. Caso a opção seja uma execução mais longa, há uma regra geral que informa que o tempo de simulação deve ser dez vezes maior que o tempo transiente (*warm-up*).

Obs.: A presença de períodos transientes gera um *bias* na estimativa média (estado estacionário) das variáveis do sistema. Uma outra questão é que o período transiente pode interferir em dois momentos distintos: na amostra de dados que serão submetidos ao processo de aderência (modelagem de dados) para inferência de uma distribuição de probabilidades; e, durante a execução do modelo.

4.4 Aspectos do relatório de saída modelo Arena

Nesta seção será apresentado o processo de análise básica dos resultados de uma execução de uma modelo simples no Arena. Então, após ajustada a rodada, com definição de período³ e do número de replicações, o sistema é colocado para executar. Ao final da rodada, o Arena informa sobre a visualização do relatório de saída. A geração deste relatório só é possível após a instalação do add-on Crystal Reports (arquivo de instalação presente no diretório Crystal disponibilizado pelo instalador do Arena).

O relatório gerado pode ser visualizado em painel próprio⁴. O relatório pode ser salvo em formato pdf ou no d Crystal Reports. Na aba Relatórios (*Reports*) e no próprio

³ No caso, está se considerando sistemas terminais.

⁴ Padrão de visualização fornecido pelo Crystal Reports.

relatório, os resultados são gerados em seções, sendo as principais: *Entity*, *Process*, *Queue* e *Resources*. Caso tenham sido executadas diversas rodadas (replicações), cada replicação apresentará os valores por seção. Por exemplo, considerando um modelo $M/M/1$ de pedágio e nomeando os elementos (módulos) como segue:

- CREATE: *Chegada carros no pedagio*
- PROCESS: *atendimento no pedagio*
- DISPOSE: *passagem pelo pedagio*

Então, as variáveis serão referenciadas como a seguir (exemplos de variáveis):

- *Chegada carros no pedagio*.**NumberOut**: quantidade de entidades criadas no sistema
- *passagem pelo pedagio*.**NumberOut**: quantidade de entidades que foram atendidas e saíram do sistema
- *atendimento no pedagio*.**WIP** (Work In Process): entidades que estão em atendimento

Indicador de performance geral (do sistema): **Number Out** : (average): número de clientes atendidos na rodada pelo sistema. Será considerado como média quando a rodada tiver mais de uma replicação.

Algumas variáveis gerais:

- TNOW (System Time): variável interna do Arena que armazena o valor do relógio; usado para o cálculo de duração de tempo.
- TAVG (Time in System - average value): armazena a média de uma variável do tipo Tally (dependente do tempo).
- DAVG (Dstat ID) — average value: armazena a média de uma expressão através das diversas replicações, como, o tamanho de fila.

Seção **Entity**

Entity.**VATime** (VA: *value-added time attribute*): (Average) (Minimum Value) (Maximum Value): tempo de atendimento médio, mínimo e máximo. Armazena o tempo total acumulado nos processos (e seus delays associados). No módulo DISPOSE, os dados são sumarizados na variável VATime.

Observações:

1. Para confirmar, pode-se confrontar os valores de média, mínimo e máximo com a distribuição escolhida.

2. *Value-added* é um tempo dispendido para realizar algum processamento útil (valor agregado ao resultado final).
3. *NVATime* ou *non-value-added* é uma variável que tem a mesma função de *VATime*, só que acumula delays em processos que não adicionam valores (tempo dispendido “inutilmente”). Por exemplo, o tempo gasto para mover um produto entre dois processos industriais.

Entity.**WaitTime** (*waiting time attribute*): (Average) (Minimum Value) (Maximum Value): tempo de espera médio, mínimo e máximo. Armazena o tempo total acumulado nas filas (áreas de espera) como também nos processos e delays associados. Obs.: o valor mínimo sempre será da entidade primeira a ser atendida (mas também pode se referir para outras entidades atendidas).

EntitiesIn (Entity Type) e EntitiesOut (Entity Type): **Number In** e **Number Out**: variável que contém o número total de entidades do tipo especificado (Entity Type) que entram no sistema, ou seja, que são criadas pelo módulo CREATE. Por outro lado, **Number Out** é total de entidades que saem do sistema, retirados pelo módulo DISPOSE. Obs.: a diferença revela o número de entidades que estão em fila ou estão em processamento.

Entity.**WaitTime** (*waiting time attribute*): (Average) (Minimum Value) (Maximum Value): tempo de espera médio, mínimo e máximo. Armazena o tempo total acumulado nas filas (áreas de espera) como também nos processos e delays associados.

Entities**WIP** (Entity Type) (*number of entities in process*): (Average) (Minimum Value) (Maximum Value): tempo de espera médio, mínimo e máximo. This variable stores the total number of entities of the specified type that are currently in the system (Work In Process).

Entity.**Total Time** : (Average) (Minimum Value) (Maximum Value): tempo total médio, mínimo e máximo relativo às entidades dentro do sistema (soma dos tempos de serviço - **VATime** e tempos de espera em fila - **Wait Time**).

Seção **Queue**

Na execução básica, há duas variáveis:

Waiting Time - *Cobrança pedágio.Queue* : tempo de espera médio, mínimo e máximo na fila.

Number Waiting - *Cobrança pedágio.Queue* : número médio, mínimo e máximo de carros em espera na fila de pedágio (processo de atendimento).

Conforme nomenclatura usada anteriormente: $T_{file} = \text{Waiting Time}$ e $N_{fila} = \text{Number Waiting}$

Seção **Resource**

Instantaneous Utilization - *atendente* : percentual médio, mínimo e máximo de uso do recurso, no caso, o percentual de ocupação do atendente.

Total Number Seized - *atendente* : número total de entidades atendidas pelo processo (ou que passaram pelo pedágio)

Conforme nomenclatura usada anteriormente: $\rho = \text{Instantaneous Utilization}$

4.5 Análise de Sistemas Terminais

A análise de sistemas a partir deste momento considerará que sejam **terminais** e que estejam em **estado estacionários**, ou seja, que o período transiente tenha sido resolvido conforme algum método proposto pela bibliografia (FILHO, 2008, p. 247)). Esta é uma importante etapa do processo, no qual a análise dos resultados vai determinar duas direções: aceitação dos resultados e consequente decisão; ou, rejeição e reinício do processo. As ferramentas básicas são os testes estatísticos, mas outras abordagens podem ser usadas.

No caso de rejeição dos resultados, o **processo de modelagem** deve reiniciar em alguma fase anterior de forma iterativa. Algumas fases do processo que podem necessitar de reavaliação: amostragem e modelagem dos dados para fins de determinar a distribuição de probabilidade estatística; o design do modelo; e; o processo de experimentação (parâmetros de execução do modelo).

O processo de análise dos resultados visa medir o desempenho dos parâmetros/variáveis de saída do modelo (algumas vistas em na seção anterior de resultados). Determinar a **confiança estatística** das variáveis de saída é o processo padrão nesses casos. O objetivo é compreender o comportamento ao longo de um período de tempo (data/período e duração). Principais características/parâmetros de sistemas terminais:

- Condições iniciais e período: fixo
- Replicações: variável e independente.

Num primeiro momento, o **objetivo** é determinar *quantas replicações por rodada* para um determinado *nível de confiança*, considerando um nível de performance desejado.

Replicações consistem de várias execuções (dentro de uma única corrida) geradas por sementes aleatórias diferentes, mas com a mesma configuração (condições iniciais, período de execução, etc.). Logo, **uma replicação** é a geração de uma amostra do processo que representa a evolução do sistema de sua condição inicial para o estado final. Se um experimento necessita de várias replicações, cada uma é uma amostra diferente, apesar da mesma condição inicial e dos parâmetros do modelo. Do ponto de vista estatístico, as **replicações** devem ser aleatórias e independentes.

O motivo da necessidade de várias replicações é que a teoria estatística informa que o **intervalo de confiança** e os **testes de hipóteses** não são válidos para uma amostra simples. E as várias replicações tratam dos experimentos cujos cenários possuem parâmetros estocásticos. Com múltiplas replicações serão obtidas múltiplas amostras para melhor estimar a performance média.

Antes de seguir com a análise dos resultados, será apresentada uma revisão das nomenclaturas em estatística inferencial. Uma observação: o parâmetro de média μ também será encontrado em outro contexto, na teoria de filas, quando representará a taxa de atendimento/serviço:

Parâmetros	Estatísticas
População	Amostra
π	P
μ	\bar{X}
σ	S

Tabela 2 – Alguns parâmetros populacionais e respectivas estatísticas amostrais.

Sendo:

- π : proporção de algum atributo dentro dos elementos da população; P , se na amostra;
- μ : média (ou valor esperado) de alguma variável quantitativa nos elementos da população; \bar{X} , se na amostra;
- σ : desvio padrão de uma variável dentro os elementos da população; S , se na amostra.

Voltando aos resultados de uma rodada, com 1 ou mais replicações (n replicações). O Arena apresenta os valores médios, mínimos e máximos das variáveis mas, também valores indicadores de **confiança estatística** e de **precisão**. A confiança é expressa através da abordagem do intervalo de confiança (IC), dado por $P = 1 - \alpha$, sendo α o valor da significância estatística. Dada uma média de uma variável (por exemplo, $VATime$) como \bar{x} , o Arena também informa o valor de **Half Width** correspondente. Ele é a metade da largura em torno da média real do processo (μ), conforme a seguinte fórmula:

$$P(\bar{x} - h \leq \mu \leq \bar{x} + h) = 1 - \alpha$$

Onde, $(\bar{x} - h, \bar{x} + h)$ é definida como **precisão** e é referenciada como largura ou tamanho do intervalo. O valor de μ é o valor esperado (média populacional) do processo e \bar{x} é a média resultado do processo de modelagem⁵.

⁵ Aqui a média é um resultado que inclui: modelagem conceitual, modelagem dos dados de entrada; rodadas e suas replicações

Obs.: O nível de significância α é a probabilidade máxima permitida para cometer um erro do tipo I que é rejeitar uma hipótese (algo) que é na realidade verdadeira. Se α diminui (aumenta o intervalo IC) então diminui a chance de rejeitar H_0 quando a mesma é verdadeira. O erro tipo II é igual a β que é a chance de aceitar H_0 quando a mesma é falsa. Ocorre que diminuir α , aumenta a chance de aceitar a hipótese nula quando ela se revela falsa.

Um exemplo do significado: considere que foi encontrado o valor de 15 min como média de *Waiting Time* em uma fila de um ponto de atendimento. O sistema informou um *Half Width* = 3. Então:

$$P(15min - 3min \leq \mu \leq 15min + 3min) = (1 - 0.05) \times 100(\%)$$

$$P(12min \leq \mu \leq 18min) = 95\%$$

Pode-se afirmar com 95% de confiança que a média de atendimento do posto se encontra entre o mínimo de 12 min e o máximo de 18 min.

O valor de h é determinado considerando a formulação da distribuição t de Student para $n - 1$ graus de liberdade e significância de α . O valor de S é o desvio padrão amostral. Então:

$$h = t_{n-1, \frac{\alpha}{2}} \times \frac{S}{\sqrt{n}}$$

O valor de n é o número de amostras ou de replicações e esta fórmula pode ser trabalhada (invertida) de forma a se obter o número de replicações.

Mas, antes, serão apresentados parâmetros do intervalo de confiança (IC). O IC representa uma estimativa intervalar (largura da estimativa) de um parâmetro populacional, como a média μ . Três fatores influenciam a largura deste intervalo (FILHO, 2008, p. 223):

- O número de replicações (número de amostragens);
- O nível de confiança dado por $100.(1 - \alpha)\%$, predefinido pelo modelador;
- Variância (S^2 ou desvio padrão amostral S) das amostras coletadas (replicações).

Como os fatores interferem no IC: o aumento do número de replicações diminui a largura do intervalo, ou seja, aumenta a precisão para encontrar a média μ . Quando o analista aumenta o nível de confiança, diminuindo a significância estatística α , está se aumentando a largura, diminuindo a precisão. Replicações com altos valores de variância S^2 levam também ao aumento da largura do IC.

A seguinte fórmula computa o número de replicações (amostras) com base no nível de confiança e precisão desejados:

$$n = \left(\frac{100 \times z \times S}{r \times \bar{x}} \right)^2$$

Sendo, n o número de replicações (amostras), z o valor de 1 desvio padrão amostral (logo $z = 1,960$), r o percentual de precisão (largura do intervalo) ou nível de confiança dado por $100 \cdot (1 - \alpha)\%$ (α é o nível de significância que pode normalmente assumir os valores de 0,1, 0,05 ou 0,01).

Mas, ao invés desta fórmula, pode-se recorrer aos resultados do Arena num processo iterativo, conforme os passos a seguir:

1. Determinar o nível de confiança desejado (90%, 95%, ou 99%). O padrão é 95% que indica nível de significância de $\alpha = 0,05$.

2. Ao executar uma rodada, determinar um valor de número de replicações (normalmente igual a 15). Essa seria a execução piloto.

3. Obter o valor de precisão (semi-intervalo $h = Half\ Width$) através do resultado do Arena para uma determinada variável objeto de análise de desempenho (o tempo médio de espera em fila: *Waiting Time*). Para esta etapa, usa-se o programa OUTPUT ANALYZER do Arena para abrir o arquivo de estatística⁶ referente à variável em análise⁷. Considerando os valores de uma simulação com 15 replicações:

Average=2.81

Standard Deviation=1.81

Half Width=1

Minimum Value=0.939

Maximum Value=7.61

Number of obs.=15

Obs.: pode-se calcular a variabilidade da variável fazendo a razão entre o desvio padrão e a média: $2,26/3,37 = 0,644$.

Calcula-se o h^* (intervalo desejado) como menor ou igual a 10% da média amostral⁸. No caso acima, seria:

$$h^* = 2,81 * 0,1 = 0,281$$

⁶ As replicações devem ser tratadas como **Lumped** (uma amostra só, com as 15 replicações).

⁷ No projeto, dentro do Arena, deve-se incluir o módulo *Output*, da aba *Input Output*. Especificam-se o arquivo .dat de saída e a expressão com a variável de interesse

⁸ É um valor determinado pelo analista

A meta seria que o intervalo obtido (h) fosse menor ou igual ao $h^* = 0,281$, considerando a média amostral de $\bar{X} = 2,81$. Como o $h = 1$ é maior que o intervalo desejado, segue-se para determinar o valor de amostra ideal.

4. Usando a seguinte fórmula, calcular a nova estimativa para n^* , ou seja, o número de replicações que satisfazem o requisito do intervalo desejado.

$$n^* = n \times \left(\frac{h}{h^*}\right)^2 = n \times \left(\frac{1}{0,281}\right)^2 = 189,967$$

5. Faz-se uma nova rodada com o novo número de replicações, no caso, com **190** e verifica-se se os resultados de *Half Width* estão compatíveis com o intervalo desejado. Observação: no programa Arena, quando do uso de várias replicações, deve-se selecionar a opção de reinicializar o sistema e as estatísticas.

4.5.1 Validação e verificação de modelos

A **validação conceitual** é um processo que tem o objetivo de indicar se o modelo está correto, ou seja, representa a realidade modelada. Três áreas que devem ser objeto de validação (Filho (2008, p. 151)):

- Simplificações e abstrações do *modelo conceitual* da realidade física.
- *Dados e amostragens* e as respectivas distribuições de probabilidade para processos estocásticos.
- Análises e decisões advindas dos *resultados* da simulação.

Algumas técnicas para validação de um modelo de simulação discreta:

- Avaliação por especialistas e testes de Turing⁹.
- Análise de sensibilidade: realizar experimentos com mudança de parâmetros e submeter a testes estatísticos.
- Validação face a face (ou validação conceitual): é apresentar o modelo aos responsáveis pelos processos do mundo real.
- Comparação com modelos anteriores de um mesmo tipo de processo.
- Duplicação de modelos através de sistemas e/ou equipes diferentes.

⁹ Teste de Turing consiste basicamente submeter a um especialista dados reais e dados simulados. Se não conseguir diferenciá-los, então a modelagem pode ser considerada validada.

A **validação operacional** trata de verificar se o modelo está produzindo resultados compatíveis com o sistema do mundo real. Há diversos níveis, iniciando com o nível qualitativo, no qual se compara os resultados com a realidade. Por exemplo, como se comporta o modelo em relação ao sistema do mundo real, quanto a filas (se há ou não), uso de recursos (se foi usado ou está ocioso), etc. No segundo nível, verificam-se valores quantitativos do modelo. Exemplo: se uma empresa fabrica em média 50 produtos por dia e o modelo reporta 100, então, ele deve ser invalidado.

E, ao final, segue-se para validação no aspecto quantitativo formal, no qual são usadas técnicas estatísticas para validação de diversos indicadores de desempenho. Existe o caso de modelos de sistema que ainda serão construídos. Neste caso, pode-se usar o processo de matriz de influências causais, que identifica a correlação entre saídas e entradas, quando se altera os valores de input.

Além da validação conceitual e operacional, há a **verificação** da implementação computacional do modelo conceitual. A engenharia de processos e de software fornece diversas técnicas para atestar/verificar se o processo de desenvolvimento está sendo executado corretamente. Exemplos: divisão de um grande modelo em módulos menores; uso de dados mais simples (e possivelmente por cálculos manuais); revisão da lógica através de animação, etc.

Referências

FILHO, P. J. d. F. *Introdução à modelagem e simulação de sistemas: om aplicações em arena*. 2. ed. ed. [S.l.]: Visual Books, 2008. 384 p. ISBN 978-857502-228-3. Citado 8 vezes nas páginas 4, 10, 11, 18, 21, 24, 26 e 28.

GUTTAG, J. V. *Introduction to Computation and Programming Using Python*. Second. [S.l.]: MIT Press, 2016. 384 p. ISBN 978-857502-228-3. Nenhuma citação no texto.

ROSSETTI, M. D. *Simulation Modeling and Arena*. 3rd and open text edition. ed. John Wiley & Sons, 2021. Disponível em: <<https://rossetti.github.io/RossettiArenaBook/>>. Acesso em: 28 maio 2022. Citado na página 20.