

---

UTILIZAÇÃO DE REDES DE PETRI COLORIDA E TEMPORIZADA PARA A MODELAGEM COMPUTACIONAL E SIMULAÇÃO DE UMA SOLUÇÃO DE INTEGRAÇÃO VOLTADA À ÁREA DE PUBLICIDADE E PROPAGANDA

---

FRANCINE FREDDO

UNIVERSIDADE REGIONAL DO NOROESTE DO  
ESTADO DO RIO GRANDE DO SUL

DISSERTAÇÃO DE MESTRADO

ORIENTADOR:

DR. SANDRO SAWICKI

COORIENTADOR:

DR. RAFAEL ZANCAN FRANTZ



**Applied**  
Computing  
Research Group

ABRIL, 2017

First published in February 2012 by  
Applied Computing Research Group - GCA  
Department of Exact Sciences and Engineering  
Rua Lulu Ilgenfritz, 480 - São Geraldo  
Ijuí, 98700-000, Brazil.

Copyright © MMXII Applied Computing Research Group  
<http://www.gca.unijui.edu.br>  
[gca@unijui.edu.br](mailto:gca@unijui.edu.br)

In keeping with the traditional purpose of furthering science, education and research, it is the policy of the publisher, whenever possible, to permit non-commercial use and redistribution of the information contained in the documents whose copyright they own. You however are *not allowed* to take money for the distribution or use of these results except for a nominal charge for photocopying, sending copies, or whichever means you use redistribute them. The results in this document have been tested carefully, but they are not guaranteed for any particular purpose. The publisher or the holder of the copyright do not offer any warranties or representations, nor do they accept any liabilities with respect to them.

**Financiamento:** Bolsa de mestrado concedida pela CAPES.

# Universidade Regional do Noroeste do Estado do Rio Grande do Sul

A Comissão Examinadora, abaixo assinada, \_\_\_\_\_ a dissertação intitulada "Utilização de Redes de Petri Colorida e Temporizada para a Modelagem Computacional e Simulação de uma Solução de Integração voltada à área de publicidade e propaganda:", elaborada por Francine Freddo, como requisito parcial para a obtenção do título de Mestre em Modelagem Matemática.

---

Dr. Sandro Sawicki  
UNIJUÍ  
(Orientador)

---

Dr. Rafael Zancan Frantz  
UNIJUÍ  
(Co-orientador)

---

Dra. Luciana de Oliveira Rech  
UFSC

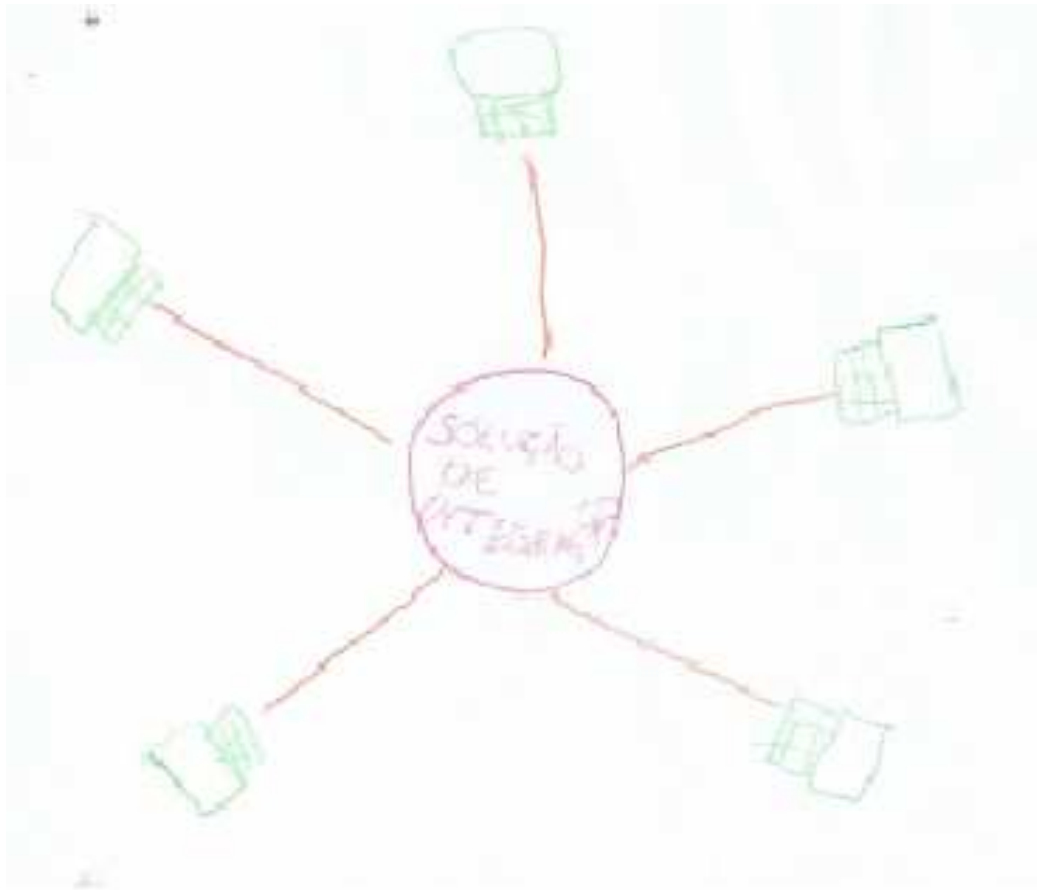
---

Dr. Manuel Osório Binelo  
UNIJUÍ

---

Dra. Fabricia Roos-Frantz  
UNIJUÍ

Ijuí, \_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_.



Integração de Aplicações por Ana Lara, Bancos de idade.

Dedico este trabalho à minha família.

---

# Conteúdo

---

<b>Agradecimentos</b> .....	<b>v</b>
<b>Resumo</b> .....	<b>vi</b>
<b>Abstract</b> .....	<b>vii</b>
<b>1 Introdução</b> .....	<b>1</b>
1.1 Contexto da Pesquisa .....	1
1.2 Motivação .....	3
1.3 Objetivos .....	4
1.3.1 Geral .....	4
1.3.2 Específicos .....	4
1.4 Metodologia .....	5
1.5 Resumo das Contribuições .....	5
1.6 Estrutura dessa Dissertação .....	6
<b>2 Referencial Teórico</b> .....	<b>7</b>
2.1 Tecnologia Guaraná .....	7
2.1.1 Linguagem de Domínio Específico .....	8
2.1.2 Notação Gráfica .....	8
2.2 Simulação de Eventos Discretos .....	12
2.2.1 Modelos de Simulação .....	14
2.3 Redes de Petri .....	17
2.3.1 Tipos de Rede de Petri .....	19
2.4 Resumo do Capítulo .....	23
<b>3 Trabalhos Relacionados</b> .....	<b>25</b>
3.1 Modelagem e simulação com Redes de Petri Colorida .....	25

3.2	Modelagem e simulação com Redes de Petri Colorida Hierárquicas	27
3.3	Modelagem e simulação com Redes de Petri Estocástica	28
3.4	Simulação de soluções de integração com outros formalismos matemáticos	29
3.5	Resumo do Capítulo	30
<b>4</b>	<b>Estudo de Caso</b>	<b>31</b>
4.1	Problema de Integração	31
4.2	Ecosistema de Software	32
4.3	Modelo Conceitual de Integração	33
4.4	Resumo do Capítulo	35
<b>5</b>	<b>Modelo de Simulação Proposto</b>	<b>36</b>
5.1	Modelagem	36
5.1.1	Ferramenta de Simulação	38
5.1.2	Modelo de Simulação usando Redes de Petri	38
5.1.3	Formulação Matemática	43
5.1.4	Variáveis Observadas	46
5.2	Experimentação	46
5.3	Resultados e Discussão	50
5.4	Resumo do Capítulo	53
<b>6</b>	<b>Verificação Formal do Modelo</b>	<b>59</b>
6.1	Definição de Verificação e Validação	59
6.2	Técnicas de Verificação e Validação	60
6.3	Verificação do Modelo de Simulação	62
6.4	Resumo do Capítulo	63
<b>7</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>65</b>
	<b>Bibliografia</b>	<b>68</b>

---

## Índice de figuras

---

2.1	Processo	9
2.2	A) Porta de entrada B) Porta de saída C) Porta de solicitação D) Porta de resposta	10
2.3	Slot	10
2.4	Tarefas no Guaraná [36]	11
2.5	Estudos de Sistemas. Kelton e Law [25].	14
2.6	Classificação dos modelos.	15
2.7	Rede de Petri	18
2.8	Rede de Petri Colorida [16]	19
2.9	Jantar dos filósofos modelado em rede colorida [29]	20
2.10	Rede de Petri temporizada determinística.	22
4.1	Exemplo de um totem utilizado pela empresa.	32
4.2	Modelo conceitual da solução de integração.	34
5.1	Equivalência entre Guaraná DSL e Redes de Petri	37
5.2	Declarações no CPN Tools	38
5.3	Modelo de Simulação em Redes de Petri	39
5.4	Modelo de Simulação em Redes de Petri	40
5.5	Modelo de Simulação em Redes de Petri	41
5.6	<b>A:</b> Porta de entrada em Guaraná <b>B:</b> Porta de entrada em Redes de Petri	41
5.7	<b>A:</b> Porta de saída em Guaraná <b>B:</b> Porta de saída em Redes de Petri	42
5.8	<b>A:</b> Porta de solicitação em Guaraná <b>B:</b> Porta de solicitação em Redes de Petri	42
5.9	<b>A:</b> Tarefa Filtro em Guaraná <b>B:</b> Tarefa Filtro em Redes de Petri	43
5.10	<b>A:</b> Tarefa Replicador em Guaraná <b>B:</b> Tarefa Replicador em Redes de Petri	43



5.11	<b>A:</b> Tarefa Tradutor em Guaraná <b>B:</b> Tarefa Tradutor em Redes de Petri	44
5.12	<b>A:</b> Tarefa Correlacionador em Guaraná <b>B:</b> Tarefa Correlacionador em Redes de Petri	44
5.13	<b>A:</b> Tarefa Enriquecedor de Conteúdo em Guaraná <b>B:</b> Tarefa Enriquecedor de Conteúdo em Redes de Petri	45
5.14	<b>A:</b> Tarefa Fusão em Guaraná <b>B:</b> Tarefa Fusão em Redes de Petri	45
5.15	<b>A:</b> Tarefa Montador em Guaraná <b>B:</b> Tarefa Montador em Redes de Petri	46
5.16	Tabela dos cenários de cada experimento	47
5.17	Tabela dos cenários do experimento de cada prioridade	48
5.18	Controle de lançamento de <i>tokens</i>	49
5.19	Tempo médio de permanência das mensagens nos <i>slots</i> (1000 <i>tokens</i> )	50
5.20	Tempo médio de permanência das mensagens nos <i>slots</i> (5000 <i>tokens</i> )	51
5.21	Tempo médio de permanência das mensagens nos <i>slots</i> (10000 <i>tokens</i> )	52
5.22	Acúmulo máximo de mensagem nos <i>slots</i> (1000 <i>tokens</i> )	53
5.23	Acúmulo máximo de mensagem nos <i>slots</i> (5000 e 10000 <i>tokens</i> )	54
5.24	Acúmulo máximo de mensagem nos <i>slots</i> - Correlacionador de tempo 20.	55
5.25	Tempo médio de permanência das mensagens nos <i>slots</i> - Correlacionador de tempo 20.	56
5.26	Tempo médio de permanência das mensagens nos <i>slots</i> - Correlacionador de tempo 20	57
5.27	Acúmulo das mensagens nos <i>slots</i> - Correlacionador de tempo 20	58
6.1	Tabela de verificação do modelo formal de simulação - Correlacionador de tempo 16	63
6.2	Tabela de verificação do modelo formal de simulação - Correlacionador de tempo 16	63

---

# Agradecimentos

---

*A persistência é o caminho do êxito.*

*Charles Chaplin, Ator e diretor inglês (1889-1977)*

**A**gradeço primeiramente a Deus pela vida e inspiração. A toda a minha família, que das mais variadas formas sempre me apoiaram e ajudaram. Ao meu namorado Cristian, pela paciência e compreensão pelos momentos ausentes. Aos professores do Grupo de Computação Aplicada, Dr. Rafael Z. Frantz, Dra Fabricia Roos-Frantz e em especial ao meu orientador Dr. Sandro Sawicki, pelas orientações e conhecimentos transmitidos. Foram de extrema importância para meu crescimento como pesquisadora. Aos demais professores do Programa de Pós-Graduação em Modelagem Matemática. A todos os colegas de mestrado. Principalmente aos colegas do Grupo de Computação Aplicada, em especial ao Ivan Kühne. Enfim, meu sincero agradecimento a todos que de uma maneira ou outra contribuíram para que esta caminhada fosse um sucesso.

---

# Resumo

---

*O início de todas as coisas é pequeno.*

*Marcus T. Cicero, Filósofo Romano (106 AC - 43 AC)*

**C**om o avanço tecnológico, as empresas que possuem aplicações para gerenciar seus processos de negócios necessitam, normalmente, trocar informações entre seus diferentes tipos de sistemas. Muitas vezes estas aplicações foram adquiridas por diferentes fornecedores em diferentes épocas, ocasionando dificuldades para trabalharem de forma única e sincronizada. Neste contexto, surge a área de Integração de Aplicações Empresariais que visa proporcionar metodologias, técnicas e ferramentas para a concepção e a implementação de soluções de integração. Para tanto, quando há uma grande demanda de recursos computacionais, esta solução pode apresentar problemas, sendo o mais comum o acúmulo de mensagens, o que ocasiona gargalos de desempenho. Para a elaboração de uma solução de integração algumas fases precisam ser seguidas: levantamento de requisitos, análise, projeto, implementação e testes. Atualmente, os gargalos de performance são encontrados após a implementação e teste do sistema. Entretanto, isso demanda tempo e custo. Esta pesquisa propõe a análise do comportamento e identificação de possíveis gargalos de desempenho da solução de integração, com base na identificação do tempo de permanência das mensagens no sistema e no acúmulo de mensagens nas filas (*slots*) considerando diferentes prioridades de processamento, ainda na fase de projeto. Para isso desenvolveu-se um modelo formal de simulação a partir de um modelo conceitual usando Redes de Petri Colorida e Temporizada. A solução de integração analisada insere-se na área de publicidade e propaganda e foi desenvolvida pela tecnologia de integração Guaraná. O modelo proposto será validado por meio de técnicas formais de verificação presentes na literatura.

**Palavras-chave:** Integração de Aplicações Empresariais; Redes de Petri; modelo; simulação.

---

# Abstract

---

*The beginnings of all things are small.*

*Marcus T. Cicero, Roman philosopher (106 BC - 43 BC)*

**W**ith the technological advance, the companies that own applications to manage its business processes need, normally, to exchange informations among its different kinds of systems. Many times, these applications were acquired by different providers in different periods, causing difficulties to work in unic and synchronized way. In this context, the area of Enterprise Application Integration emerges with the aim of providing methodologies, technics and tools to the planning and implementation of integration solutions. For this purpose, when there is a huge demand of computational resources, this solution can present some problems, and the most common is the message accumulation, resulting in performance lacks. For the elaboration of an integration solution, some stages must be followed: requirement surveying, analysis, project, implementation and tests. Nowadays, performance lacks are found after the system implementation and test. However, there is a time and cost demand. This research proposes the behaviour analysis and the identification of possible performance lacks of the integration solution based on the identification of the messages lenght of stay in the system and on the message accumulation in the lines (slots) considering different processing priorities, still in the project stage, by means of the development of a simulation formal model as of a conceptual model using Colorful and Timed Petri Nets. The solution integration analysed is insert in the Publicity and Advertising area and was developed by Guaranã; integration technology. The model proposed will be validated by means of verification formal techniques existing in literature.

**Keywords:** Enterprise Application Integration; Petri Net; model; simulation.

---

# Capítulo 1

## Introdução

---

Aprenda com o ontem, viva para o hoje, esperança para o amanhã.  
O importante é não parar de questionar.

Albert Einstein, físico alemão (1879 - 1955)

### 1.1 Contexto da Pesquisa

No contexto econômico e industrial atual, é necessário que muitos sistemas empresariais sejam reprojatados para atender a demanda do mercado, tanto em razão de mudanças em requisitos funcionais, como pela evolução tecnológica. O grande desafio, nesse contexto, é conciliar a criação de novos sistemas empresariais com as tecnologias atuais, ao mesmo tempo fazendo com que os sistemas antigos se mantenham operantes em novas plataformas ou trabalhem em conjunto. Não existiriam problemas desse tipo se os sistemas fossem totalmente independentes um do outro, mas isso não é verdade. A grande maioria dos casos, cada novo sistema depende das informações já existentes.

Neste contexto, como é possível integrar várias aplicações para que elas trabalhem em conjunto e possam trocar informações?

A partir dessa necessidade surgiu um campo de pesquisa conhecido, atualmente, como Integração de Aplicações Empresariais. Segundo Linthicum [28] o termo EAI do inglês *Enterprise Application Integration* "é o compartilhamento irrestrito de dados e processos de negócios entre quaisquer

aplicações conectadas e fontes de dados na empresa."O Processo de Negócio refere-se ao conjunto de atividades que produzem um serviço específico para seus clientes.

EAI é ainda o termo formal que contempla a integração de aplicações corporativas e um conjunto de ferramentas e tecnologias. As aplicações que a empresa foi adquirindo ao longo do tempo e que dispõe para dar suporte aos seus processos de negócios são denominadas ecossistema de *software* [17]. Como a dependência das corporações em relação à tecnologia tem crescido e se tornado mais complexa, a necessidade por um método de integração de aplicações em um único arsenal de processos de negócios tem sido a prioridade [8].

Uma solução de integração tem o objetivo de "orquestrar"um conjunto de aplicações para mantê-las harmonizadas ou oportunizar novas funcionalidades a partir daquelas existentes. Pode-se perceber que uma solução de integração é composta por processos lógicos de integração e de comunicação, que conectam processos ou aplicações do ecossistema a solução de integração [17].

A integração de aplicações permite o compartilhamento de informações dentro da mesma organização ou com parceiros. Isto gera vantagem competitiva. A maior parte das organizações utilizam vários tipos e "gerações"de sistemas desenvolvidos ao longo dos anos. Estes sistemas possuem valor nas empresas, mas o seu valor agregado pode significar pouco se estes não puderem "conversar"com outros sistemas. A necessidade de integração destes sistemas têm se intensificado com a popularidade de pacotes tais como SAP, Peoplesoft ou Baan [8].

Qualquer organização tem de se preocupar com o compartilhamento de informações entre distintas aplicações. Estas, podem ser escritas em linguagens diferentes, baseadas em plataformas diferentes, e com algumas suposições diferentes sobre como a empresa opera. Amarrar tais aplicações requer uma grande quantidade de conhecimentos e compreensão de como vincular aplicações em conjunto, tanto a nível empresarial como a nível técnico. Para isso é necessário saber sobre como cada aplicação funciona.

Uma solução de integração visa organizar a comunicação entre as diferentes aplicações. Para a sua elaboração é necessário seguir algumas etapas tais como levantamento de requisitos, análise, projeto, implementação e testes, no qual as duas últimas possuem um custo elevado. Mesmo seguindo todas essas fases, ainda pode ocorrer erros ou gargalos de performance em situações de grande demanda, que são detectados, somente na fase de implementação

e testes. Com o intuito de diminuir custos, percebe-se a importância de se verificar esses erros e gargalos ainda na fase de projeto, por meio de modelos conceituais de solução e modelos de simulação.

Os modelos conceituais podem ser construídos a partir de uma DSL (*Domain-Specific Languages*, em português Linguagem de Domínio Específico) que é uma linguagem própria para interpretação da solução de integração, dentre elas destacam-se a *Apache Camel* [23], *Mule ESB* [12], *Spring Integration* [14] e *Guaraná* [19]. Os modelos de simulação podem ser representados por técnicas matemáticas, tais como, Teoria das Filas, Cadeias de Markov, Redes de Petri. Esta dissertação aborda as Redes de Petri como formalismo para a elaboração do modelo de simulação e a Tecnologia *Guaraná* para o modelo conceitual.

## 1.2 Motivação

Atualmente a maioria das empresas investe na aquisição de inúmeras aplicações heterogêneas de *software* para organizar o seu processo de negócio. Esse número elevado de aplicações é comumente conhecido como ecossistema de *software* [21], o qual consiste em várias aplicações criadas por diferentes pessoas e fornecedores, desenvolvidas interna ou externamente, geralmente criadas sem considerar sua possível integração [17].

Deste modo, percebe-se a necessidade de se criar uma solução de integração, para que todas as aplicações que compõe o ecossistema de *software* consigam comunicar-se entre si. Pois uma solução de integração visa orquestrar este conjunto de aplicações para mantê-las sincronizadas e permitir que novas funcionalidades sejam construídas a partir daquelas que já existem [17].

Geralmente, o desenvolvimento de uma solução de integração está dividida em etapas de levantamento de requisitos, análise, projeto, implementação e testes, no qual as duas últimas possuem um custo elevado. Apesar de uma solução de integração seguir todos os padrões e técnicas de implementação podem ocorrer erros em sua estrutura. Além disso, podem apresentar gargalos de performance nos seus componentes em situações de grande demanda. Assim, a presente pesquisa tem como motivação a possibilidade de analisar o comportamento e identificar os gargalos de performance, ainda na fase de projeto, por meio dos modelos conceituais da solução.

Considerando o contexto apresentado, formula-se a seguinte hipótese:

Uma solução de integração de aplicações pode ser classificada como sistema estocástico, dinâmico e discreto. Sendo assim, pode-se utilizar modelos

computacionais e matemáticos, e técnicas de simulação de eventos discretos visando analisar o comportamento e encontrar possíveis gargalos de desempenho que podem surgir quando uma solução de integração de aplicação é submetida a um cenário crítico de funcionamento, ainda na fase de projeto.

## 1.3 Objetivos

Nessa seção serão apresentados o Objetivo Geral e os Específicos que norteiam esta dissertação.

### 1.3.1 Geral

*Analisar o comportamento e identificar os possíveis gargalos de desempenho da solução de integração de aplicações voltada à área de publicidade e propaganda, por meio da observação do tempo de permanência das mensagens e acúmulo nos slots, a partir do modelo conceitual, utilizando Redes de Petri como técnica matemática para o desenvolvimento de um modelo formal de simulação.*

### 1.3.2 Específicos

- Realizar uma análise temporal por meio do modelo de simulação construído com Rede de Petri.
- Utilizar *tokens* com prioridade de processamento, para observar se causará impacto no comportamento da solução de integração.
- Identificar o tempo médio de permanência dos *tokens* em todo o processo.
- Identificar o tamanho máximo dos lugares.
- Baseado em temporalidade e prioridade, expor o comportamento da solução de integração de aplicações em cenários críticos de funcionamento.
- Escolher o tipo de Rede de Petri e *software* para simulação do modelo.
- Demonstrar a equivalência entre o modelo conceitual e o modelo de simulação.



## 1.4 Metodologia

A metodologia utilizada consistiu em realizar uma revisão bibliográfica acerca da área de Integração de Aplicações Empresariais, descrevendo suas características, estilos de integração e qual a finalidade de uma solução de integração.

Associado a isto, foram estudados os tipos de Redes de Petri que foram aplicados no contexto de Integração de Aplicações Empresariais. Baseado na Rede de Petri escolhida foi criado um modelo formal de simulação equivalente ao modelo conceitual proposto no estudo de caso.

Foram definidos os cenários de simulação similares a um processo real de funcionamento de uma solução de integração. Os resultados experimentais foram tabulados, sendo que, para cada cenário foram executadas 25 repetições, seguindo a lei dos Grandes Números [20].

As simulações computacionais, foram realizadas por meio da ferramenta *CPN Tools*, que melhor se aplica ao tipo de Rede de Petri escolhida, ou seja, Rede de Petri Colorida/Temporizada.

Finalmente, a verificação da equivalência do modelo formal de simulação em Rede de Petri com o modelo conceitual foi realizado por meio de técnicas de verificação formal largamente encontradas na literatura.

Os resultados obtidos na simulação foram utilizados, também, pelas técnicas de verificação buscando avaliar a precisão do modelo criado.

## 1.5 Resumo das Contribuições

O desenvolvimento desta dissertação teve contribuições importantes:

- Com o entendimento dos elementos da tecnologia Guaraná e o estudo de Redes de Petri foi possível elaborar um modelo de simulação equivalente ao modelo conceitual de uma solução de integração de um problema da área de publicidade e propaganda e apresentá-lo no Salão do Conhecimento, na Jornada de Pesquisa, evento promovido pela UNIJUÍ, em Ijuí, RS, setembro, 2016.
- Dando continuidade ao trabalho apresentado acima, foram realizadas algumas simulações com o modelo de simulação proposto e analisado o comportamento da solução de integração, encontrando alguns gargalos, estes resultados foram apresentados na 7<sup>o</sup> MCSul (Conferência Sul em Modelagem Computacional), evento ocorrido em Rio Grande, RS, novembro, 2016.

## 1.6 Estrutura dessa Dissertação

Essa dissertação está organizada da seguinte maneira:

**Capítulo §1.** Compreende essa introdução, na qual contextualiza e motiva a pesquisa.

**Capítulo §2.** Proporciona para o leitor uma revisão da literatura técnica e científica relacionadas à pesquisa desenvolvida nessa dissertação.

**Capítulo §3.** Se introduz os trabalhos relacionados identificados ao longo dessa pesquisa.

**Capítulo §4.** No qual se apresenta o estudo de caso analisado nessa dissertação.

**Capítulo §5.** Apresenta o trabalho desenvolvido, introduz a contribuição central da pesquisa desenvolvida, seus resultados e discussões.

**Capítulo §6.** Apresenta o trabalho de validação da proposta.

**Capítulo §7.** Apresenta as conclusões da pesquisa desenvolvida nessa dissertação, juntamente com os trabalhos futuros.

---

# Capítulo 2

## Referencial Teórico

---

Estudar é polir a pedra preciosa, cultivando o espírito, purificando-o.

Confúcio, filósofo chinês (551 a.C. - 479 a.C.)

**E**ste capítulo apresenta os conceitos e termos envolvidos no contexto de uma solução de integração e da simulação de sistemas, que são fundamentais para o entendimento desta pesquisa. A Seção §2.1 trata da tecnologia Guaraná, a Seção §2.2 da Simulação de Eventos Discretos e os tipos de modelos e por fim a Seção §2.3 Redes de Petri, no que consiste e suas extensões.

### 2.1 Tecnologia Guaraná

Uma solução de integração de aplicações empresariais precisa manter os dados e as informações do ecossistema de *software* da empresa em sincronia, além de permitir que novas funcionalidades sejam desenvolvidas a partir daquelas existentes, sem que as aplicações tenham que ser modificadas [18].

Nesse contexto, a tecnologia Guaraná [19] é utilizada para projetar soluções de integração de aplicações empresariais em um alto nível de abstração, de forma que os engenheiros de *software* possam manter o foco na criação de modelos para a solução do problema, sem precisar se preocupar com detalhes técnicos de sua implementação.

Dentre os principais recursos dessa tecnologia estão uma Linguagem de Domínio Específico (do inglês *Domain-Specific Language* - DSL) e um motor de execução (do inglês *runtime system*), que permite a implementação e execução da solução de integração.

### 2.1.1 Linguagem de Domínio Específico

Para a criação dos modelos conceituais, as plataformas de integração proporcionam uma linguagem de domínio específico, conhecida também como DSL (do Inglês, *Domain-Specific Languages*). DSL, segundo Fowler [15] "são pequenas linguagens, com foco em um aspecto particular de um sistema de *software*". É uma linguagem desenvolvida especialmente para um domínio, permitindo expressar soluções de problemas de forma abstrata. Ela precisa ser pequena e fácil de entender, porém, ela possui uma expressividade limitada.

A DSL pode ser dividida em dois tipos: interna e externa. Uma DSL interna constitui-se nas diferentes formas de reescrita do código oferecidas ao programador. [15]. Já a DSL externa é uma linguagem separada, não depende de outra linguagem, elas possuem uma sintaxe própria e precisam de um interpretador para analisar o código e executar os comandos.

A tecnologia Guaraná proporciona uma linguagem de domínio específico que permite projetar soluções de integração a um alto nível de abstração utilizando uma sintaxe concreta gráfica e conceitos de modelagem intuitivos. Esta linguagem de modelagem é baseada nos padrões de integração documentados por Hohpe e Woolf [21].

Os modelos conceituais são desenvolvidos na tecnologia Guaraná utilizando uma linguagem gráfica. A transformação dos modelos em código executável é obtida por meio da Engenharia Dirigida por Modelos [41]. Dessa forma, modelos projetados na tecnologia podem ser reutilizados para gerar, automaticamente, soluções de integração para serem executadas em diferentes plataformas. A tecnologia Guaraná fornece um conjunto de transformações por meio das quais o engenheiro de *software* traduz os modelos projetados em código Java para serem executados [19].

### 2.1.2 Notação Gráfica

A notação da tecnologia Guaraná é fundamentada nos padrões de integração de aplicações documentadas por Hohpe e Woolf [21]. Essa representação possibilita projetar a estrutura interna de todos os processos de integração e suas portas de comunicação [5].

O Guaraná DSL é composto por cinco elementos: aplicações, processos, *slots*, tarefas e portas. Essa tecnologia permite a modelagem de diferentes tipos de tarefas, no qual cada uma é representada graficamente por um ícone, de acordo com a função que desempenha [19].

As tarefas se comunicam por meio de *slots* que atuam como *buffers*, os quais recebem a mensagem já processada pela tarefa e a deixa disponível para seguir o fluxo e ser processada pela tarefa em sequencia. Sendo assim, uma vez que a mensagem é processada e liberada para os *slots* seguintes, a tarefa está pronta para receber e processar outra mensagem [19].

### Processos e Portas

Os processos são blocos que agrupam um determinado conjunto de tarefas e contém portas para a comunicação com as aplicações integradas. Conforme a Figura §2.1 os processos possuem portas de entrada, saída, solicitação e resposta, pelas quais um processo recebe ou envia mensagens para as aplicações.

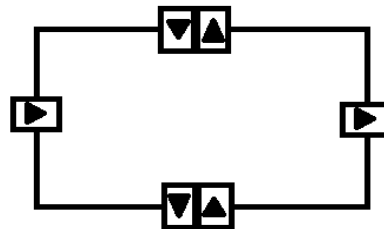


Figura 2.1: Processo

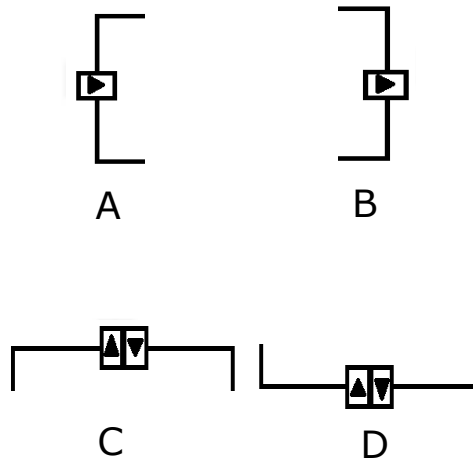
Uma porta de entrada envia uma mensagem de entrada para um *slot* e este a encaminha para uma tarefa processá-la. Já a porta de saída, lê sempre uma mensagem a partir de um *slot* e deixa disponível para o elemento seguinte no fluxo. Portas de entrada e portas de saída estão sempre conectadas umas com as outras. As portas de solicitação e resposta são aquelas que possibilitam entrada e saída de mensagens na mesma porta. Na Figura §2.2 tem-se a notação gráfica de cada uma.

### Slots

*Slots* são unidades de armazenamento temporário que conectam portas com uma ou mais tarefas. Possibilitam o processamento assíncrono das mensagens que fluem dentro de um processo, proporcionando independência entre as tarefas. São representados graficamente como mostra a Figura §2.3.

### Tarefas

As tarefas são os principais elementos que compõem os processos, pois são responsáveis pelo processamento e modificação das mensagens. A tarefa



**Figura 2.2:** A) Porta de entrada B) Porta de saída C) Porta de solicitação D) Porta de resposta



**Figura 2.3:** Slot

recebe a mensagem do *slot* de entrada, processa, de acordo com sua semântica e a encaminha para o *slot* seguinte, deixando-a disponível para a próxima tarefa. A Figura §2.4 apresenta as principais tarefas divididas em grupos conforme suas atribuições dentro do processo, sua notação gráfica e respectivos nomes.

As tarefas estão divididas em quatro grupos: roteadoras (do inglês *routers*), modificadoras (*modifiers*), transformadoras (*transformers*) e temporais (*timers*).

As roteadoras são tarefas responsáveis por direcionar uma mensagem de entrada para zero, um ou mais destinos, ou ainda multiplicar, correlacionar, direcionar e ordenar mensagens no fluxo da solução de integração. Entre essas tarefas destacam-se o *filter* (filtro) e o *replicator* (replicador). O filtro é utilizado para evitar que mensagens desnecessárias avancem no processo. Já o replicador faz cópias de uma mensagem de entrada, quando se faz necessá-






























Routers	Modifiers	Transformers	Timers
 Correlator	 Slimmer	 Translator	 Delayer
 Merger	 Context Slimmer	 Splitter	 Ticker
 Resequencer	 Content Enricher	 Aggregator	 Expire Checker
 Filter	 Context Content Enricher	 Chopper	
 Idempotent Transfer	 Header Enricher	 Assembler	
 Dispatcher	 Context Header Enricher	 Cross Builder	
 Distributor	 Header Promoter	 Custom Transformer	
 Replicator	 Header Demoter		
 Semantic Validator	 Custom Modifier		
 Custom Router			

Figura 2.4: Tarefas no Guaraná [36]

rio que está seja enviada para dois ou mais destinos, sem modificar seu conteúdo original.

As tarefas modificadoras tem a função de modificar o conteúdo original de uma mensagem de entrada. O *content enricher* (enriquecedor de conteúdo) é uma tarefa que atua sobre a mensagem, quando esta necessita de que seja acrescentado mais informações, ou seja, nesse grupo de tarefas a mensagem que entra não será a mesma na saída do processo.

O grupo de tarefas transformadoras tem a função de modificar a estrutura da mensagem. Uma das tarefas desse grupo que atua sobre a estrutura da mensagem é o *translator* (tradutor) que traduz mensagens de um formato para outro. Outra é o *aggregator* (agregador) que pode receber duas ou mais mensagens e agrupá-las em apenas uma mensagem.

As tarefas temporais são usadas quando é necessário controlar o tempo em um fluxo de integração, antecipar ou atrasar uma execução.

## 2.2 Simulação de Eventos Discretos

Simulação é um termo muito amplo, porém pode-se definir basicamente como sendo o processo de elaboração de um modelo real e a condução de experimentos com a finalidade de entender o comportamento de um sistema ou avaliar sua operação [42], ou seja, permite a geração de cenários, a partir dos quais pode-se orientar o processo de decisão, fazer análises e avaliações e propor melhorias no sistema.

A simulação tem aumentado em muitas áreas, inclusive na área de Integração de Aplicações Empresariais, auxiliando os gestores na tomada de decisão em problemas complexos e possibilitando um melhor conhecimento dos processos nas organizações [37].

A utilização da técnica de simulação tem trazido inúmeros benefícios, tais como, a previsão de resultados na execução de uma determinada ação; a redução de riscos na tomada de decisão; a identificação de problemas antes mesmo de suas ocorrências; a eliminação de procedimentos em arranjos industriais que não agregam valor a produção; a redução de custos com o emprego de recursos (mão-de-obra, energia, água e estrutura física) e a revelação da integridade e viabilidade de um determinado projeto em termos técnicos e econômicos [43].

A simulação, segundo Aalst (2015), citado em Sawicki et al. [40] é composta por onze fases, sendo elas: definição do problema, modelagem, modelo conceitual, realização, modelo executável, verificação e validação, modelo validado, experimentando, resultados da simulação, interpretação e respondendo as soluções. A definição do problema precisa considerar as metas para saber o que vai e o que não vai fazer parte da simulação. A modelagem, é onde o modelo conceitual é elaborado com as propriedades mais importantes do sistema. A fase de realização consiste em transformar o modelo conceitual em um modelo executável. O modelo executável pode ser simulado no computador, por isso necessita de uma ferramenta de simulação. O próximo passo é verificar e validar o modelo, comparando com o modelo real. Com o modelo validado, os experimentos podem ser realizados, sendo conduzidos de maneira eficiente para que os resultados sejam confiáveis. Os resultados da simulação precisam ser interpretados para permitir um entendimento sobre o problema. E, por fim, é realizado um relatório com as respostas para as questões da definição do problema no qual as propostas de soluções são criadas [40].



Um dos desafios da simulação é demonstrar que o modelo criado representa o sistema real. Segundo Silva [43], para alcançar esta meta é recomendado a observação de três das onze fases citadas acima: verificação, validação e implementação. Esses preceitos devem ser observados nas diversas fases do desenvolvimento de um modelo.

Entende-se por verificação um conjunto de ações para certificar-se de que a forma conceitual adotada na formulação do modelo foi transcrita corretamente utilizando-se das linguagens de programação ou simulação. Por outro lado, a validação é uma coletânea de ações utilizadas para analisar se um dado modelo representa de fato o sistema em estudo, podendo este procedimento ser conduzido junto com a verificação. E a implementação é um dos procedimentos decisivos na validação, a qual consiste no confronto das informações geradas pelo modelo com as obtidas no sistema real [43].

Quando se pensa em simulação, obviamente se quer simular algum sistema. Segundo Forrester (1968) citado em Chwif e Medina [9], sistema é um agrupamento de partes que operam juntas, visando um objetivo em comum. Quando um sistema é imaginado geralmente associa-se a um sistema real, ou seja, que existe fisicamente. No entanto, é possível simular também um sistema que não existe na realidade. Embora seja mais difícil o processo de validação para um sistema não existente, é comum e viável, a construção de modelos de simulação de sistemas imaginários ou hipotéticos [9].

O sistema real pode ser representado por meio de um modelo, sendo físico (protótipo) ou matemático, no qual este último pode ser verificado analiticamente ou utilizando-se a simulação. Isto pode ser representado pela Figura §2.5.

Os estudos desses sistemas podem dar-se sob diferentes formas de abordagem. A primeira seria interferindo diretamente sob rotinas operacionais, ou seja, promovendo implementações, ou alterações de procedimentos, até que sejam obtidas as condições ideais. Estas ações requerem uma tomada de decisão por meio da condução de estudos preliminares e de experiência, para que as alterações não diminuam o desempenho do sistema [43].

A segunda, refere-se a utilização de modelos que representam os sistemas reais. Os modelos podem apresentar-se como protótipos ou como modelos matemáticos, os quais podem ter soluções analíticas, como por exemplo, um modelo de regressão, ou a simulação, permitindo desta forma, reconstituir a rotina funcional de um dado sistema real [43].

A tendência de crescimento no setor de serviços nos cenários nacional e mundial aliada a complexidade geralmente atribuído a esses sistemas, tem

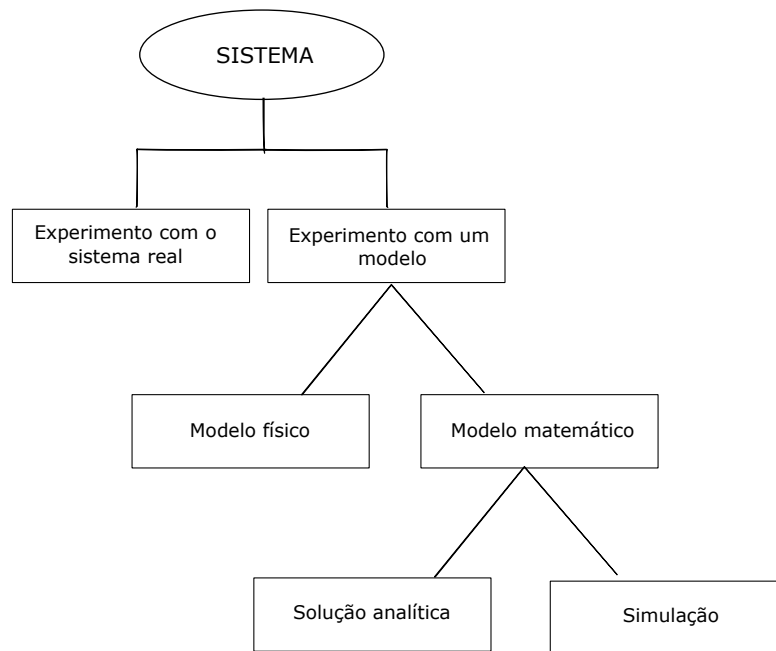


Figura 2.5: Estudos de Sistemas. Kelton e Law [25].

estimulado a realização de pesquisas sobre a aplicação de simulação no setor de serviços, porém ainda são pouco exploradas [37]. Diante disso, torna-se essencial explorar o potencial de aplicação das técnicas de simulação promovendo uma discussão do processo de modelagem em que são definidos os elementos de um sistema e como fazer o modelo para realizar a simulação.

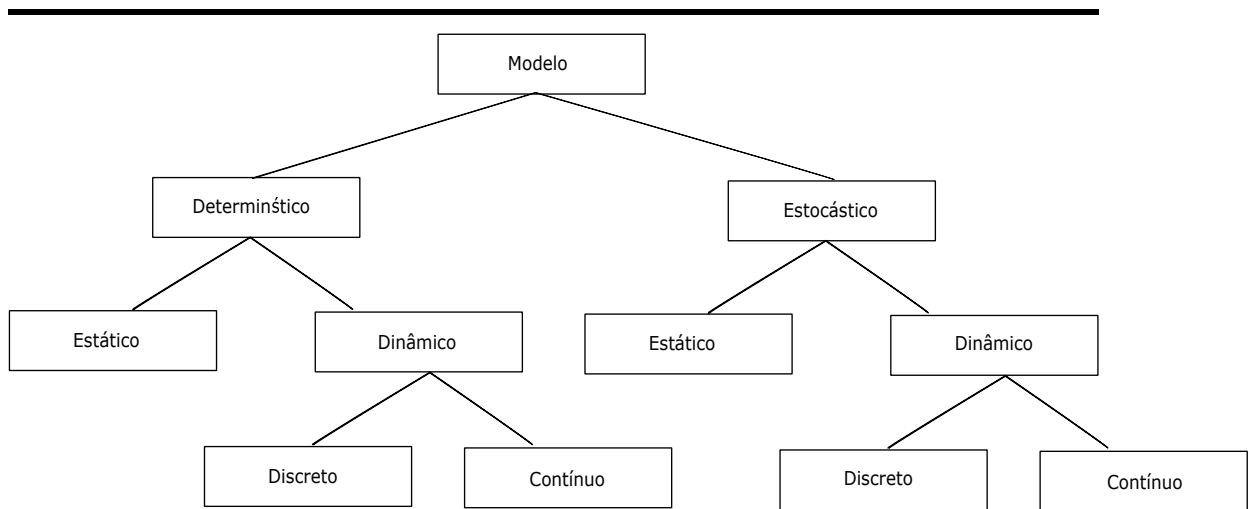
### 2.2.1 Modelos de Simulação

A simulação é um campo de pesquisa que lida com a experimentação de modelos para fazer previsões sobre o comportamento e o desempenho dos sistemas reais [40]. Um modelo é uma abstração da realidade, que se aproxima do comportamento real do sistema, mas de uma forma simplificada, pois se o modelo é mais complexo que o sistema não tem-se um modelo e sim um problema. Isso porque a principal ideia da modelagem é capturar o que de fato é importante no sistema [9].

Para construir um modelo de simulação é necessário ter conhecimento dos objetivos da simulação, ou seja, para que se quer simular, quais os requisitos. Estes podem ser traduzidos pelas medidas de desempenho do sistema, que são as variáveis de saída de um modelo de simulação [9].

Os modelos para representar os sistemas podem ser classificados em: modelos simbólicos (ou icônicos), analíticos e modelos de simulação [9]. Os **modelos simbólicos** são constituídos de símbolos gráficos (como retângulos e retas) que representam o sistema de forma estática e possuem limitações porque existe falta de informação quantitativa e dificuldades para representar detalhes. Os **modelos analíticos** basicamente podem ser reduzidos a um conjunto de equações que, ao serem resolvidas, permitem obter a solução esperada, na maioria das vezes, esses modelos são estáticos e não apresentam soluções analíticas para sistemas complexos. E o **modelo de simulação** são implementados através do auxílio de um computador, sendo o mais eficiente dos três, pois é capaz de representar sistemas que possuem um número muito grande de variáveis e com dinâmica muito complexa, caso em que o modelo analítico possui restrições [9].

A Figura 2.6 representa as classificações dos modelos de acordo com Kelton e Law [25].



**Figura 2.6:** Classificação dos modelos.

Um modelo de simulação é determinístico quando não possui nenhuma variável probabilística, e o resultado da simulação é sempre o mesmo não importando quantas vezes o modelo é executado. Já o modelo estocástico ou probabilístico utiliza uma ou mais variáveis aleatórias como entrada que conduzem a saídas aleatórias. A saída de uma simulação estocástica precisa ser tratada como uma estimativa estatística das características reais de um sistema. A grande maioria dos modelos de simulação são constituídos por

variáveis estocásticas, por isso normalmente os modelos estocásticos são mais complexos e melhor representam os sistemas reais do que os modelos determinísticos [40].

Os modelos estáticos representam um sistema em um determinado momento. Eles não se alteram ao longo do tempo, sendo assim podem ser utilizados em outros campos. No entanto, os modelos dinâmicos representam os sistemas que mudam ao longo do tempo. A maioria das simulações usa modelos dinâmicos, pois a maior parte dos sistemas mudam ao longo do tempo [40].

A simulação de modelos de eventos discretos são utilizados para modelar sistemas que mudam o seu estado em pontos discretos no tempo, a partir da ocorrência de eventos. Por outro lado, modelos contínuos são utilizados para modelar sistemas cujo estado varia continuamente no tempo [9]. Em alguns casos, é necessário construir um modelo de simulação que compreenda aspectos de eventos contínuos e discretos, esta simulação é conhecida como simulação combinada ou híbrida [40].

Ao analisar um modelo de simulação de eventos discretos pode-se distinguir três elementos básicos: entidade, atributo e atividade. Uma entidade é qualquer objeto de interesse do modelo. Um atributo é uma propriedade desta entidade e qualquer processo que pode causar mudanças no modelo é denominado atividade [9]. Analisando um modelo de simulação de manufatura, uma entidade pode ser as peças que entram no sistema, sua prioridade de produção pode ser um dos seus atributos e o seu processamento por uma máquina, uma atividade [9].

Um modelo de simulação de eventos discretos visa reproduzir as atividades de entidades envolvidas no sistema com o propósito de saber algo sobre o seu desempenho e comportamento. Com isso, é necessário saber o que é um estado de um sistema, um evento e um processo. Um estado de um sistema pode ser definido como um conjunto de variáveis utilizadas para descrever o comportamento do sistema a um determinado momento. Um evento ocorre em um momento particular no tempo e determina uma mudança no estado do sistema e um processo é uma sequência ordenada de acontecimentos e pode incluir várias atividades [40].

Portanto, a representação de uma solução de integração como um modelo de eventos discretos permite prever o futuro comportamento do sistema, possibilitando construir teorias e hipóteses baseadas em observações realizadas através dos modelos, no qual o tempo pode ser controlado, afim de entender e estudar o comportamento do sistema. Um modelo de simulação

de eventos discretos pode identificar e reproduzir as variáveis consideradas mais importantes para o desempenho do sistema e como elas interagem umas com as outras e com outros elementos. Isto facilita a compreensão do comportamento do sistema [40].

## 2.3 Redes de Petri

As Redes de Petri foram criadas por Carl Adam Petri [34] como um formalismo matemático para descrever a concorrência e sincronização dinâmica em sistemas distribuídos [2].

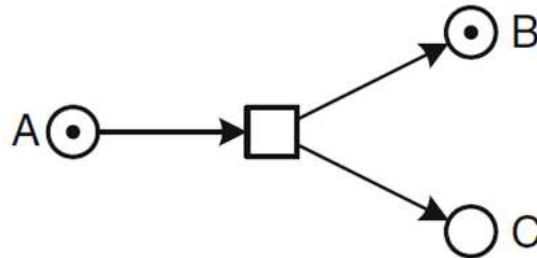
De acordo com Valette e Cardoso [44]: "A Rede de Petri é uma ferramenta gráfica e matemática que se adapta bem a um grande número de aplicações em que as noções de eventos e de evoluções simultâneas são importantes."

Dentre estas aplicações estão: avaliação de desempenho, análise e verificação formal em eventos discretos, protocolos de comunicação, sistemas de informação (organização de empresas), sistemas de transporte, logística, gerenciamento de base de dados, interface homem-máquina e multimídia [44].

Uma Rede de Petri é vista também como um tipo particular de grafo orientado que permite modelar as propriedades estáticas de um sistema de eventos discretos, constituído de dois tipos de nós: as transições (que correspondem aos eventos que caracterizam as mudanças de estado do sistema), e os lugares (que correspondem às condições que devem ser certificadas para os eventos acontecerem) interligados por arcos direcionados ponderados [33].

Sendo assim, a representação gráfica de uma Rede de Petri consiste de lugar, transição e ficha (*token*). Segundo Valette e Cardoso [44] cada elemento pode ser definido como (Figura §2.7):

- Lugar: (representado por um círculo) pode ser interpretado como uma condição, em estado parcial, uma espera, um procedimento.
- Transição: (representada por barra ou retângulo) é associada a um evento que ocorre no sistema.
- Ficha: (representado por um ponto em um lugar, *token*) é um indicador significando que a condição associada ao lugar é verificada. Pode representar um objeto, uma estrutura de dados.



**Figura 2.7:** Rede de Petri

Os lugares e as transições são conectadas por arcos (setas). Lugares podem conter *tokens* e a quantidade de *tokens* em um lugar é chamado de marcação. Um arco de entrada conecta um lugar para uma transição e um arco de saída conecta uma transição para um lugar. Quando uma transição é acionada, os *tokens* nos lugares ligados ao arco de entrada são removidos e os *tokens* são adicionados aos lugares ligados aos arcos de saída. Uma transição está ativa e pode ser emitida se a quantidade de *tokens* determinada pelos arcos de entrada existe nos respectivos lugares. A quantidade de símbolos gerados pelos arcos de saída não são necessariamente o mesmo que o removido pelos arcos de entrada [36].

Uma Rede de Petri genérica pode ser representada formalmente pela quintupla  $R = P, T, I, O, K$ , na qual  $P = p_1, p_2, \dots, p_n$  é um conjunto finito não-vazio de lugares,  $T = t_1, t_2, \dots, t_n$  é um conjunto finito não-vazio de transições.  $I : T \rightarrow P$  é um conjunto que representa o mapeamento de transições para lugares de entrada.  $O : T \rightarrow P$  é um conjunto que representa o mapeamento de transições para lugares de saída.  $K : P \rightarrow \mathbb{N}$  é o conjunto de capacidades associadas a cada lugar, podendo assumir um valor infinito [33].

A Rede de Petri é, portanto, um formalismo que permite a modelagem de sistemas dinâmicos discretos com grande poder de expressividade, permitindo representar com facilidade todas as relações de causalidade entre processos em situações de: sequencialidade, conflito, concorrência e a sincronização. Um modelo formal de especificação e controle do fluxo de informações de um sistema discreto qualquer [32].

### 2.3.1 Tipos de Rede de Petri

Nesta subseção são mostradas algumas extensões de Redes de Petri com a finalidade de aumentar a sua aplicabilidade. São discutidas as redes de Petri coloridas, hierárquica e temporizadas determinísticas.

#### Redes de Petri Colorida

Segundo Francês [16] as redes de Petri coloridas tem o objetivo de reduzir o tamanho do modelo, sendo de grande valor para a modelagem de sistemas complexos. Este tipo de rede de Petri permite que os *tokens* sejam individualizados, através de cores atribuídas a eles; assim, diferentes processos ou recursos podem ser representados em uma mesma rede. As cores não significam apenas padrões, elas podem representar tipos de dados complexos. Usa-se a nomenclatura de colorida apenas para referenciar a possibilidade de distinção entre os *tokens*. A Figura §2.8 apresenta uma rede colorida que possui a representação original, onde são utilizadas cores para os *tokens*. Nessa figura, os arcos são rotulados com cores (a, b, c e d).

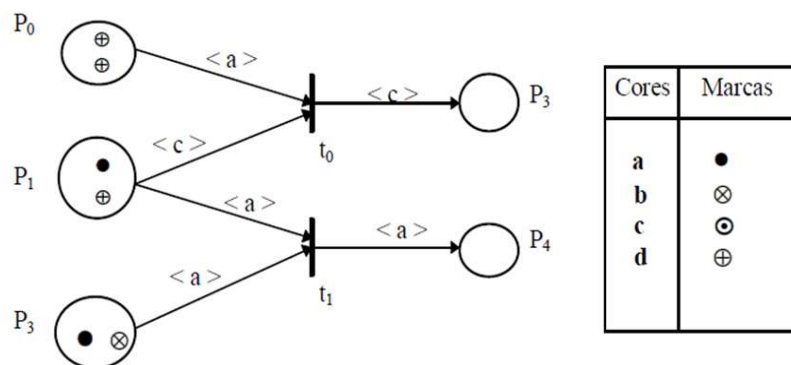


Figura 2.8: Rede de Petri Colorida [16]

Mesmo sendo um pouco rudimentar, a rede de Petri Colorida original provê mecanismos que permitem realizar uma escolha determinística. Esse poder de escolha significa uma evolução da rede colorida, porém modificações posteriores vieram dar maior adequação com relação a escolha de representações não-determinísticas [16].

De acordo com Francês [16] as redes de Petri colorida são compostas de: estrutura, declarações e inscrições. Sendo a estrutura um grafo dirigido por

dois vértices (lugares e transições), pode armazenar em cada lugar marcas de tipos diferentes e representar valores associados a tipos de dados mais complexos. Declarações compreendem a especificação dos conjuntos de cores e declarações de variáveis e as inscrições variam de acordo com o componente da rede. Os lugares possuem três tipos de inscrições: nomes, conjunto de cores e expressão de inicialização. As transições possuem dois tipos de inscrições: nomes e expressão guarda, e os arcos apenas um tipo dado pela expressão. Para diferenciar as inscrições, nomes são escritos com letras normais, cores em itálico, expressões de inicialização sublinhadas e as expressões guarda são colocadas entre colchetes.

Um problema clássico para representar uma rede de Petri colorida é conhecido como o jantar dos filósofos [29]. Esta situação consiste de três filósofos que podem estar em três estados diferentes: comendo, pensando ou com fome. Os filósofos estão a volta de uma mesa, sendo cada um deles tem a sua frente um garfo e um prato de comida. São, no entanto, necessários dois garfos para que um filósofo possa comer, ou seja, ele precisa do seu garfo e o do seu vizinho. O impasse esta em quando todos os filósofos pegarem o seu garfo (o da direita) e aguardarem a liberação do outro garfo (da esquerda). A Figura 2.9 mostra em rede de Petri colorida o jantar dos filósofos [29].

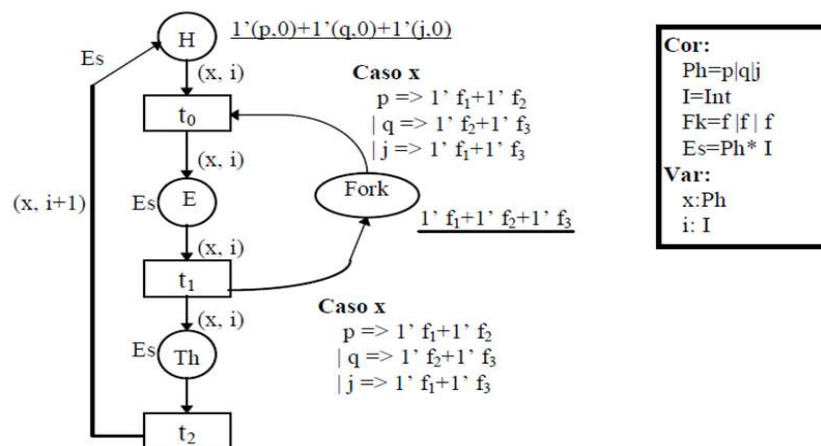


Figura 2.9: Jantar dos filósofos modelado em rede colorida [29]

Na Figura 2.9, os lugares representam os estados possíveis para os filósofos (H para "com fome", E para "comendo" e Th para "pensando") e os



recursos do sistema (no caso, os garfos representados por *Fork*). A variável  $x$  indica o filósofo que irá passar para o estado "comendo"(E) e a variável  $i$  indica o número de iterações que já ocorreram. Na primeira iteração, o lugar H possui três marcas (marcação inicial) que estão sublinhadas. Dependendo da atribuição dada à variável  $x$ , uma das três expressões é avaliada no arco que liga o lugar *Fork* à transição  $t_0$ . Assim, se, por exemplo,  $x = p$ , então a expressão  $1' f_1 + 1' f_2$  é avaliada e apenas a marcação  $1'(p, 0)$  do lugar H irá à transição  $t_0$ , significando que apenas o filósofo  $p$  irá passar para o estado "comendo". Desta forma, para cada atribuição de  $x(p, qouj)$  haverá uma situação diferente, ou seja, uma marca diferente no lugar E (estado "comendo"). A modelagem colorida, além de evitar o impasse (o que também poderia ser obtido através de uma rede não-determinística), ainda possibilita uma representação mais clara e concisa ao modelo [16].

Desta forma, percebe-se a simplicidade e facilidade da compreensão e da utilização de redes de Petri colorida para representar situações que podem ser tanto determinísticas como não-determinísticas.

### Redes de Petri Temporizadas

Segundo Valette e Cardoso [44], a rede de Petri temporizada é obtida associando-se a cada transição da rede uma duração de disparo. Sua semântica é uma noção de atraso durante a qual as flechas utilizadas para disparar a transição não são visíveis em nenhum lugar. A partir do momento em que a flecha se torna visível em um lugar, ela pode ser utilizada imediatamente por qualquer transição de saída deste lugar.

O disparo não é instantâneo, mas possui uma duração. Porém, esta associação de tempo a transição só tem sentido se a transição é interpretada como uma atividade e não como um evento instantâneo [44].

As redes de Petri temporizadas foram desenvolvidas para suprir a necessidade de escolha determinística, assim como as redes de Petri coloridas. Essas redes possibilitam a representação do comportamento dinâmico de sistemas que possuam atividades concorrentes, assíncronas e não-determinísticas, por meio da adição de tempo ao modelo [16].

O tempo pode ser associado aos componentes da rede de várias maneiras. As principais segundo Maciel et al. [29] são:

- Tempo associado aos lugares - os *tokens*, após o disparo de uma transição, só poderão estar disponíveis para disparar outra transição após um determinado tempo que está associado ao lugar.

- Tempo associado às marcas - o tempo indica quando a marca estará disponível para disparar uma transição.
- Tempo associado às transições - tem como foco as redes de Petri temporizadas determinísticas com tempos associados às transições.

Um exemplo de rede de Petri temporizada determinística é apresentado na Figura §2.10. Nela, as transições  $t_1$  e  $t_2$  possuem tempos associados diferentes, que significa que uma transição será disparada antes da outra. Assim, supondo que  $d_1 < d_2$ , então o *token* chegará primeiro no lugar  $P_1$ . De maneira determinística é possível estabelecer a ordem em que os eventos devem ocorrer [16].

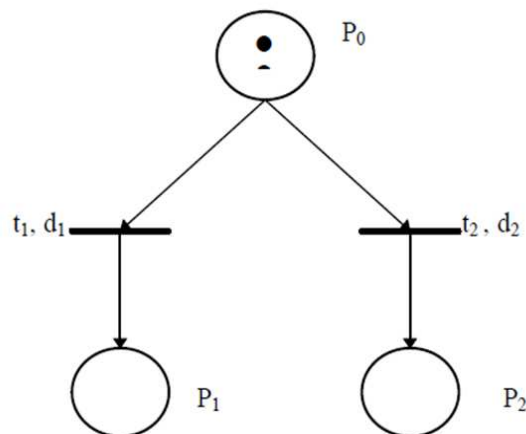


Figura 2.10: Rede de Petri temporizada determinística.

### Redes de Petri Estocásticas

As redes de Petri não foram desenvolvidas originalmente para avaliação de desempenho apesar de toda a sua potencialidade para representar sistemas complexos. No entanto, este cenário mudou com o surgimento das redes de Petri Estocásticas, apresentadas por Molloy [30]. Esta é uma técnica capaz de, além de especificar sistemas, também apresentar uma análise probabilística.

As redes de Petri Estocásticas são definidas assumindo que todas as transições são temporizadas e que o atraso no disparo das transições é associado a uma variável aleatória distribuída exponencialmente [11].

Dada uma rede de Petri Estocástica com uma marcação que possua diversas transições habilitadas a serem disparadas, uma das transições ocorre. Quando uma transição de uma rede de Petri Estocástica é disparada, uma nova marcação pode ser gerada. Esta nova marcação pode conter transições que já se encontravam habilitadas na marcação anterior, mas não foram disparadas. Pela propriedade de ausência de memória da distribuição exponencial, pode-se assumir que a atividade associada a cada transição é recomeçada para qualquer nova marcação [11].

Segundo Narahari et al. [31] as redes de Petri Estocásticas surgiram como um modelo de performance, sendo a principal ferramenta para sistemas distribuídos e de simulação de eventos discretos, tornando-se uma ótima alternativa para a sua avaliação de desempenho.

## 2.4 Resumo do Capítulo

Nesse capítulo, foram apresentados conceitos que servirão de base para a compreensão do modelo de simulação proposto.

Foi introduzido conceitos sobre a tecnologia Guaraná que possibilita elaborar modelos conceituais de uma solução de integração. Sua interface gráfica foi baseada nos padrões de Hohpe e Woolf [21]. Sendo assim, os elementos que compõe uma solução de integração são: aplicações, processo, portas, *slots* e tarefas. O processo é um bloco que contém toda a ideia lógica da solução, as portas conectam as aplicações ao processo e enviam ou recebem mensagens, os *slots* são *buffers* por onde as mensagens ficam até que a tarefa esteja apta a executar sua função e enviar a mensagem adiante.

A simulação consiste em realizar experimentos de um sistema real, para que seja possível analisar seu comportamento, avaliar seu desempenho, e com isso propor melhorias no sistema. A simulação está sendo usada em muitas áreas, inclusive na Integração de Aplicações Empresariais, auxiliando os gestores na tomada de decisão em problemas complexos e possibilitando um melhor conhecimento dos processos nas organizações. A maioria dos sistemas são estocásticos e discretos, pois não se tem o controle do tempo.

Os sistemas podem ser modelados e simulados por meio de técnicas matemáticas, a Rede de Petri é uma delas. A Rede de Petri é um formalismo gráfico e matemático que permite representar com grande expressividade sistemas de eventos discretos. Sua representação gráfica consiste em lugar (círculo), transição (retângulo) e *token* que são marcações. Os lugares e transições são conectados por arcos (setas direcionais). Existem vários tipos de

Redes de Petri, tais como, Colorida, Temporizada, Estocástica, entre outras. Cada tipo possui características que dependendo do sistema analisado podem representá-lo melhor.

---

## Capítulo 3

# Trabalhos Relacionados

---

Tentar não significa conseguir, mas certamente quem conseguiu tentou.

Aristóteles, filósofo grego (384 a.C. - 322 a.C)

**E**ste capítulo apresenta alguns trabalhos da literatura científica que utilizam formalismos matemáticos para modelar e simular Sistemas de Eventos Discreto. Na Seção §3.1 relata trabalhos que usam Redes de Petri Colorida, a Seção §3.2 trata do uso de Redes de Petri com outros tipos, a Seção §3.3 apresenta o uso das Redes de Petri Estocásticas e por fim a Seção §3.4 trata de outros formalismos, Teoria das Filas e Cadeias de Markov.

### 3.1 Modelagem e simulação com Redes de Petri Colorida

Rodrigues [35] trata do mapeamento de estruturas LSC (*Live Sequence Chart*) em Redes de Petri Colorida. LSC é uma linguagem para modelagem de cenários. Porém, o LSC possui algumas deficiências, e por isso, optou-se pela utilização das Redes de Petri Colorida. Pois, por meio dessa rede o modelo pode ser analisado e verificado as propriedades a respeito da estrutura, bem como o comportamento dinâmico do sistema. Teve como proposta apresentar uma ferramenta de mapeamento, onde cenários do tipo LSC são mapeados em Redes de Petri equivalentes.

Portanto, este trabalho utilizou Redes de Petri Coloridas para modelar um sistema com o intuito de testá-lo desde as fases iniciais de sua implementação. Para identificar determinadas exigências e detectar erros de especificação

de projeto, assim como, erros de implementação. Prevenindo problemas em fases posteriores do projeto [35].

Borges [3] relata que atualmente os sistemas de informação tanto corporativos, como os científicos precisam gerenciar seus processos de trabalho (*wokflows*) e processos de negócio. Em comunidades científicas, seus sistemas de informações referem-se a tarefas repetitivas, processamento de dados e controle de condução de experimentos em ambientes de *hardware* e *software*. Para ambos os ambientes, tanto corporativo, como o científico, propiciam que ferramentas e formalismos existentes, ou priorizem o controle de fluxo de tarefas, ou o controle do fluxo de dados. Contudo, existem situações em que é preciso que o controle de dados e o controle de tarefas sejam atendidos simultaneamente.

Sendo assim, Borges [3] caracterizou e delimitou o controle e representação do fluxo de dados em processos de negócio e *workflows* científicos. Para isso, utilizou Redes de Petri Colorida e grafos *workflow* orientados a atores, e comparou as ferramentas CPN Tools e KEPLER que são usadas nesses dois formalismos, respectivamente. A comparação foi realizada por meio de implementações de casos práticos, e permitiu identificar a abrangência e os limites de cada abordagem (Redes de Petri Colorida e grafos orientados a atores) quanto aos aspectos de representação e controle de dados.

Lied e Sellitto [27] apresentam a Rede de Petri Colorida como ferramenta para modelagem e avaliação dos riscos ocupacionais de trabalhadores em uma linha de manufatura. Com os resultados obtidos da simulação foi avaliada a carga de trabalho individual dos trabalhadores, permitindo verificar quais as tarefas desenvolvidas devem ser estudadas para diminuição de riscos ocupacionais envolvidos. Com isto, o objetivo principal deste artigo foi analisar a situação ocupacional de trabalhadores em uma linha de montagem e eventualmente propor melhorias na distribuição de tarefas, utilizando Redes de Petri para modelar, incluindo ajustes no modelo e validação por meio de dados de campo [27].

Fahland e Gierds [13] abordam um problema aberto de geração automática de um projeto informal de *Enterprise Integration Patterns - middleware* em um modelo formal que permite verificar a concepção de *middleware* para erros funcionais, completar o projeto de *middleware* em caso de falta de funcionalidade, analisar o desempenho do sistema concebido e criar automaticamente uma aplicação em execução do *middleware*.

A contribuição desta pesquisa é fornecer para cada padrão *Enterprise Integration Patterns* um modelo formal em Redes de Petri Colorida. Usando

essa formalização, um determinado *middleware design* pode ser traduzido para um modelo em Redes de Petri Colorida, descrevendo sua funcionalidade. Para permitir que o modelo seja analisado por meio de técnicas de simulação, descobrindo possíveis problemas em estágios iniciais e ainda se necessário pode-se usar o modelo em Redes de Petri Colorida como modelo para implementação [13].

Cerioni [7] focou-se na geração e manipulação do espaço de estados de modelos em Redes de Petri de cenários de fábrica no *software CPN Tools* e quais seriam os benefícios que poderiam ser obtidos por meio da confecção de modelos formais em Redes de Petri e posterior validação por meio da análise do espaço de estados e propriedades das Redes de Petri.

O autor tratou de Sistemas Flexíveis de Manufatura que são sistemas de produção que visam garantir uma rápida resposta as mudanças de demanda do mercado e inúmeras vantagens competitivas. Este trabalho tratou do sistema de Veículos Auto Guiados que transportam peças e materiais entre as unidades produtoras que compõe [7].

Aborda também, a importância de representar cenários de fábrica e roteiros de produtos na forma de modelos, de maneira que possam ser simulados e validados antes da implementação. A contribuição do trabalho esta no âmbito do uso de modelos formais, especificamente em Redes de Petri, para identificação e fornecimento de variáveis importantes para modelos de tomada de decisão [7].

## 3.2 Modelagem e simulação com Redes de Petri Colorida Hierárquicas

Sampaio et al. [38] apresentam um método de desenvolvimento de Sistema de Diagnóstico de Falhas (SDF) utilizando Redes de Petri Colorida Hierárquicas. Esse sistema visa analisar informações advindas de um sistema específico de uma subestação e apresentar diagnóstico de falta sucinto e preciso aos operadores do sistema elétrico dessa subestação, tornando essa tarefa mais rápida, confiável e menos estressante.

No SDF proposto por Sampaio et al. [38], modelos de Rede de Petri são desenvolvidos para o sistema de proteção de cada vão da subestação, sendo que podem ser validados como modelo e implementados. Devido as características das Redes de Petri facilita a utilização desse formalismo no desenvolvimento desse sistema. Para a implementação do método Sampaio et al. [38] utilizou a ferramenta *Desing CPN* (atual *CPN Tools*).

### 3.3 Modelagem e simulação com Redes de Petri Estocástica

Roos-Frantz et al. [36] trazem um problema da área de Integração de Aplicações Empresariais, com uma solução de integração construída em um modelo conceitual utilizando a tecnologia Guaraná. Este mesmo modelo foi transcrito para um modelo de simulação equivalente em Redes de Petri Estocásticas.

A qualidade dos modelos conceituais para a integração de uma solução é essencial para garantir a integração correta. Porém, na maioria dos casos descobrir se uma solução de integração pode falhar e em que condições, é um trabalho que envolve tempo, risco e dinheiro, no qual os engenheiros de *software* precisam construir a solução completa e executá-la em um sistema, para coletar os dados, de modo que a solução de integração possa ser analisada [36].

Sendo assim, Roos-Frantz et al. [36] propõem um modelo de simulação em Redes de Petri Estocástica do modelo conceitual da solução de integração de um problema real que se baseia em um projeto para automatizar o registro de novos usuários em um repositório único do Conselho do Condado de Huelva. Com o intuito de descobrir as possíveis falhas e analisar o comportamento da solução de integração, ainda na fase de projeto, ou seja, antes da sua implementação. Para com isso, reduzir riscos, tempo e gastos desnecessários.

A escolha da Rede de Petri Estocástica deve-se a essa ter se mostrado uma ótima ferramenta de modelagem e análise de sistemas de eventos discretos e assíncronos, que se refere ao problema tratado no artigo de Ross-Frantz et al. [36].

O trabalho apresentado por Cargnin [5] utiliza Redes de Petri Estocástica para desenvolver um modelo formal de simulação para um problema voltado à gerencia de ligações telefônicas.

Baseado nos conceitos de Integração de Aplicações Empresariais foi analisado a solução de integração proposta, por meio de um modelo conceitual construído com a tecnologia Guaraná. Esse modelo conceitual consiste em automatizar a cobrança de telefonemas pessoais que tenham custos, realizados pelos funcionários, com uso de linhas telefônicas da Universidade Regional do Noroeste do Estado do Rio Grande do Sul (UNIJUÍ).



Com o objetivo de analisar o comportamento dessa solução de integração e identificar possíveis gargalos de desempenho, o modelo conceitual foi transcrito para um modelo de simulação equivalente usando Redes de Petri. Esse modelo foi simulado com a ferramenta *PiPe 2* e verificado por meio de técnicas encontradas na literatura, neste trabalho foi utilizado a técnica de Kleijnen [26] e de Sargent [39].

### 3.4 Simulação de soluções de integração com outros formalismos matemáticos

Wiesner [45] utilizou Teoria das Filas como uma nova abordagem para identificar possíveis gargalos de desempenho de uma solução de integração, utilizando um modelo conceitual, por meio de um modelo formal de simulação.

Como estudo de caso, foi tratado um problema de Integração de Aplicações Empresariais, no qual foi trabalhado com uma solução de integração projetada com a tecnologia Guaraná, para o gerenciamento de telefonemas pessoais realizados por funcionários de uma Universidade [45].

Os resultados experimentais obtidos por Wiesner [45] por meio do modelo de simulação, permitiram analisar o comportamento da solução de integração em determinadas taxas de trabalho e identificar os possíveis gargalos de desempenho.

O modelo de simulação apresentado por Wiesner [45] foi verificado e validado por meio de técnicas formais descritas na literatura. Dentre as quais para a verificação foi utilizada a técnica de rotinas de verificação que garante que o modelo está livre de erros e para a validação foi utilizada uma das mais simples e usadas técnicas de validação a da intuição de especialistas, comprovando que o modelo de simulação proposto funciona e representa a solução de integração.

Horn [22] propõe um modelo de simulação formal baseado em Cadeias de Markov para uma solução de integração representada em um modelo conceitual desenvolvido pela tecnologia Guaraná. Esta solução consiste em gerenciar os telefonemas pessoais realizados por funcionários de uma Universidade.

Com esta simulação Horn [22] buscou analisar o comportamento da solução de integração e identificar possíveis gargalos de desempenho antes da

sua implementação, ou seja, na fase de projeto. Com os resultados obtidos demonstrou-se que é possível avaliar a qualidade de uma solução de integração sem implementá-la.

Para verificar o modelo de simulação proposto, Horn [22] utilizou as técnicas sugeridas por Sargent [39] e de Freitas Filho [10] analisando a simulação para cada cenário a que o modelo foi submetido. Constatando que o modelo formal de simulação está de acordo com o modelo conceitual da solução de integração.

### 3.5 Resumo do Capítulo

Nesse capítulo, foram apresentados diversos trabalhos acerca de Redes de Petri e outros formalismos matemáticos usados na Simulação de Eventos Discretos. Relatou-se trabalhos que utilizaram Redes de Petri Coloridas, Estocásticas, Coloridas e Hierárquicas, e ainda Teoria das Filas e Cadeias de Markov.

O trabalho apresentando nesta dissertação tem como diferencial em relação aos trabalhos citados na Seção §3.1 as análises feitas acerca do modelo de simulação, sendo observadas as variáveis tempo e diferentes tipos de *tokens* (prioridades). Outra diferença deste trabalho em relação aos que estão ligados à área de Integração de Aplicações Empresariais (Seção §3.3 e Seção §3.4) está no formalismo matemático, na ferramenta de simulação e no caso de estudo. Além disso, utiliza-se prioridades nas mensagens para analisar o tempo de permanência nos *slots* e o acúmulo.

---

# Capítulo 4

## Estudo de Caso

---

*The journey of a thousand miles  
begins with a single step.*

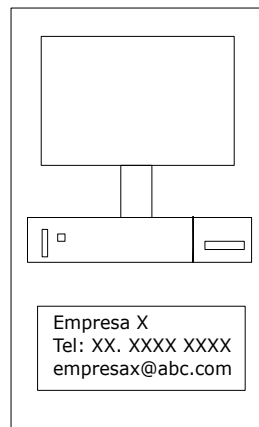
*Chinese proverb*

**E**ste capítulo apresenta o problema de integração e o modelo conceitual da solução deste problema por meio da tecnologia Guaraná.

### 4.1 Problema de Integração

O problema de integração refere-se à uma empresa real cujo ramo de atividade é voltado para a publicidade e propaganda. Seu trabalho é divulgar comerciais de vários estabelecimentos do comércio da região, utilizando estações de propagandas, chamada de "totens". Cada estação possui um monitor LCD Full HD de 20", um mini gabinete e uma impressora em seu interior, como ilustra a Figura §4.1. A empresa possui diversos totens localizados nas principais cidades do Rio Grande do Sul, onde há um intenso fluxo de pessoas.

Deste modo, a empresa precisa selecionar os comerciais que serão inseridos em cada um dos totens, pois em cada cidade existe um conjunto de estabelecimentos diferentes. Em cada local existe uma rede *wireless* em que o totem está conectado. O problema consiste em uma série de pacotes que envolvem diferentes aplicações e que trabalham individualmente, na qual a



**Figura 4.1:** Exemplo de um totem utilizado pela empresa.

empresa precisa verificar cada um deles para obter informações concretas sobre os componentes da estação. Segundo a empresa, perde-se muito tempo realizando estas verificações, tempo este que poderia ser utilizado para personalizar mais anúncios por dia ou buscar novos estabelecimentos que estejam dispostos a divulgar sua propaganda.

## 4.2 Ecossistema de Software

O ecossistema de *software* é composto por 8 aplicações, divididas em 5 pacotes. No pacote 1 é a maneira como os anúncios são recebidos pela empresa. Um estabelecimento comercial cria seu anúncio e o envia para a empresa por meio de e-mail ou de mídias digitais.

No pacote 2, o arquivo é recebido e armazenado em um banco de dados. A partir disso, inicia-se o processo de postagem do anúncio. Para isso, a empresa filtra o banco de dados buscando novos anúncios recebidos e que não foram publicados. Estes então são compartilhados nas redes sociais. Em seguida, a empresa precisa inserir estes arquivos no totem correspondente. Este processo ocorre no pacote 3, no qual a empresa utiliza um *software* de acesso remoto para se comunicar com a estação correspondente. Nela, os anúncios são inseridos em uma pasta mapeada por um *software* responsável por reproduzir vídeos. O pacote 4 representa os *software* secundários que estão executando no totem.

Estes são responsáveis por monitorar o *status* da rede, da impressora, da temperatura e do *hardware* do totem. A empresa monitora cada um des-

tes *softwares* individualmente e, caso algum deles apresentem um defeito, ela deve tomar providências para que o problema seja resolvido. O pacote 5 contém a impressora, que está instalada internamente na estação, a qual imprime um cupom de desconto a cada 30 minutos. Este cupom é uma estratégia de *marketing* para que os clientes próximos ao totem possam utilizá-lo em compras feitas nos estabelecimentos cujos anúncios estão sendo reproduzidos na estação.

O objetivo então, é de automatizar o controle de monitoramento que a empresa utiliza para verificar o comportamento de cada estação de propaganda. A atividade de monitoramento envolve cinco *softwares*: dois monitores de *hardware* (rede e temperatura), outro da impressora e dois para o vídeo. Os *softwares* de monitoramento de *hardware* verificam a temperatura dos componentes e a taxa de transmissão de dados, bem como o *status* da rede de *wireless*. O da impressora verifica o *status* da mesma, bem como seus componentes internos (guilhotina, papel). Em relação aos *softwares* de vídeo, o primeiro verifica se o monitor está ligado e o outro é responsável por reproduzir os comerciais na tela.

### 4.3 Modelo Conceitual de Integração

Diante disso, Arruda et al. [1] criou um modelo conceitual da solução de integração responsável por coordenar as atividades dos *softwares* que serão integrados, como mostra Figura §4.2, utilizando a tecnologia Guaraná. Nela, o processo de integração inicia-se pela porta de entrada P0, que verifica o banco de dados da empresa, retirando cada novo arquivo de propaganda que a empresa recebe. Este arquivo é transformado em uma mensagem e é enviado à tarefa T0, por meio do *slot* S0, que é responsável por garantir que o arquivo obtido pela porta P0 ainda não foi postado e nem inserido na estação de propaganda anteriormente, evitando duplicações de arquivos.

Assim que a mensagem sai da tarefa T0, ela entra no *slot* S1 e segue para a tarefa T1 que distribui a mensagem para as tarefas T2 e T3. A primeira realiza a tradução da mensagem para um esquema que a aplicação integrada trabalha. Feito isso, ela segue para as redes sociais onde será postada nas páginas da empresa, auxiliando na disseminação da propaganda. Já a segunda tarefa dá continuidade ao fluxo de integração gerando uma cópia da mensagem distribuída pela tarefa T1 a qual é enviada para a tarefa tradutora T4. Após traduzida, a mensagem entra na aplicação de acesso remoto que contém os dados do totem correspondente.

Esses dados são inseridos na cópia da mensagem que é enviada para a tarefa T5. Nela, a mensagem original e sua cópia são correlacionadas e enviadas

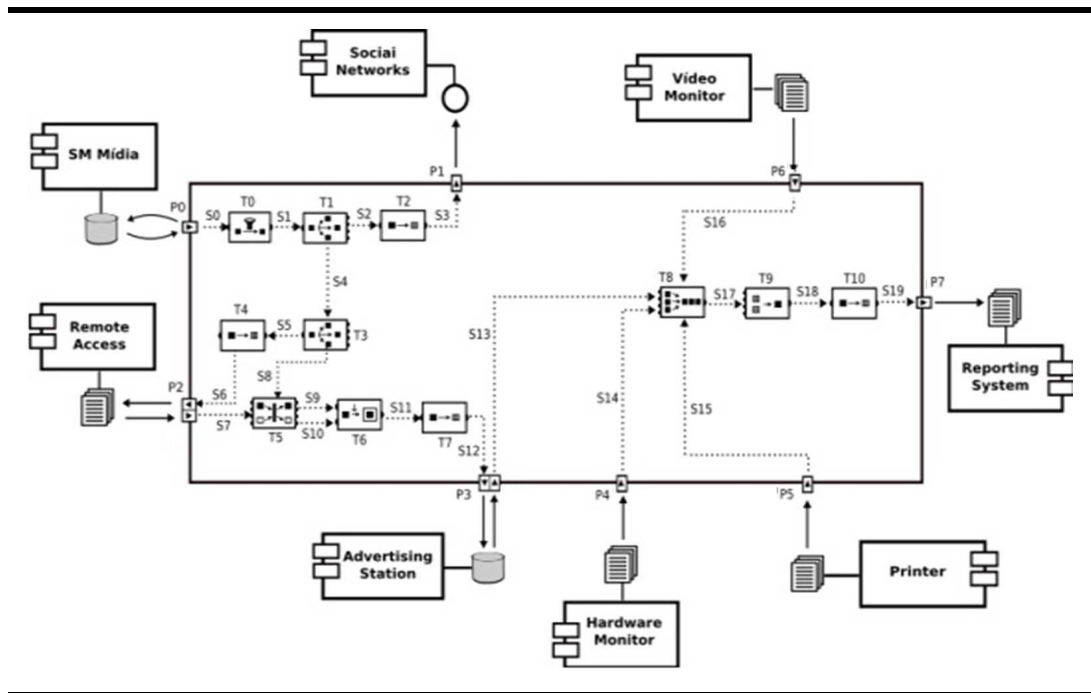


Figura 4.2: Modelo conceitual da solução de integração.

à tarefa T6 pelos slots S8 e S9. Esta tarefa adicionada no corpo da mensagem original as informações de acesso ao totem obtidas pela cópia. Com isso, o processo de integração pode acessar a estação de propaganda e inserir a mensagem que contém o arquivo de vídeo no banco de dados da mesma. Enquanto este processo ocorre, a porta de cada aplicação que monitora uma parte do totem retira as informações de seus logs e as insere dentro de mensagens.

Cada mensagem é encaminhada por meio de seu respectivo slot para a tarefa T7, que juntará todas as mensagens, colocando-as em um único slot. Este direciona as mensagens que serão as entradas para a tarefa T8 a qual criará uma nova mensagem a partir das entradas recebidas. Esta nova mensagem é traduzida e enviada para o sistema de notificação, que enviará um relatório para a empresa contendo o status de cada aplicação, componente e da própria estação. Caso a empresa identifique que houve um problema com algum componente, ela deve tomar as devidas providências para contornar o problema.

## 4.4 Resumo do Capítulo

Neste capítulo foi apresentado o estudo de caso que se refere a um problema de integração da área de publicidade e propaganda, seu ecossistema de *software* que é composto por 8 aplicações: SM Mídia, Acesso Remoto, Redes Sociais, Estação de Propaganda, Vídeo Monitor, *Hardware* Monitor, Impressora e Sistema de Notificação. Ainda foi tratado o modelo conceitual da sua solução, desenvolvida pela tecnologia Guaraná.

---

## Capítulo 5

# Modelo de Simulação Proposto

---

*Se eu vi mais longe, foi por estar sobre ombros de gigantes.*

*Isaac Newton, físico e matemático inglês (1643-1727)*

**E**ste capítulo apresenta a proposta de criação de um modelo de simulação em Redes de Petri Colorida e Temporizada a partir de um Modelo Conceitual de uma solução de integração. A Seção §5.1 mostra como foi feita a transcrição do modelo conceitual para o modelo de simulação em Redes de Petri, apresenta a ferramenta de simulação, a formulação matemática do modelo de simulação e quais as variáveis observadas no experimento. A Seção §5.2 explica como foi realizada as simulações, quais os experimentos e cenários e a Seção §5.3 trata dos resultados e discussão dos mesmos.

### 5.1 Modelagem

A modelagem computacional foi realizada com o uso de Redes de Petri Colorida e Temporizada e visa analisar o comportamento para identificar possíveis gargalos de desempenho de uma solução de integração voltada para a área de publicidade e propaganda.

Para isso, propõe-se realizar uma análise temporal utilizando *tokens* com prioridades de processamento buscando identificar durante todo o processo: (i) o tempo médio de permanência dos *tokens* nos lugares; (ii) o número máximo de *tokens* armazenados nos lugares.









Com Redes de Petri Colorida e Temporizada é possível representar *tokens* diferentes e também atribuir tempo para cada um deles. A ferramenta de simulação utilizada foi o *CPN Tools*. Esta ferramenta possui uma linguagem de programação própria para descrição de atributos dos elementos da rede. Esta linguagem fornece declarações de conjuntos coloridos (tipos), variáveis, constantes, funções e procedimentos.

Para realizar a análise temporal do modelo computacional desenvolvido foram definidos três tipos de prioridades aos *tokens*: *token* com prioridade Alta, *token* com prioridade Média e *token* com prioridade Baixa. Para cada prioridade foi associado um tempo: prioridade Alta (tempo 1), prioridade Média (tempo 2) e prioridade Baixa (tempo 3). Essas prioridades no sistema real representam diferentes tipos de mensagens de diferentes tamanhos.

Para fazer a equivalência do modelo conceitual para o modelo de simulação, utilizou-se, parcialmente, a abordagem proposta por Roos-Frantz et al. [36] que traduziu modelos conceituais de uma solução de integração, construídos com a tecnologia Guaraná para um modelo de simulação em Redes de Petri Estocástica. Contudo, a abordagem proposta neste trabalho é diferente pois envolve Redes de Petri Colorida e Temporizada para o desenvolvimento do modelo de simulação.

A Figura 5.1 mostra a equivalência entre os elementos principais da tecnologia Guaraná e Redes de Petri. A tecnologia Guaraná tem como principais elementos as tarefas, os *slots* e as mensagens que são análogos, respectivamente, as transições, os lugares e os *tokens*, principais componentes da Rede de Petri.

---

Guaraná DSL	Redes de Petri
Tarefa 	Transição 
Slot 	Lugar 
Mensagem 	Token 

---

Figura 5.1: Equivalência entre Guaraná DSL e Redes de Petri

Nas Redes de Petri, os lugares junto com os arcos, tem função semelhante ao *slot*. Os arcos fazem a ligação de um lugar para uma transição e indica o sentido do fluxo dos *tokens*, ou seja, da mensagem e não possuem valor funcional.

### 5.1.1 Ferramenta de Simulação

O *CPN Tools* é uma ferramenta de modelagem, análise e simulação de Redes de Petri Coloridas. A ferramenta foi desenvolvida na Universidade de Aarhus, Dinamarca, e é distribuído livremente para organizações não comerciais via web, no endereço <http://cpntools.org/>. Esta ferramenta tem sido utilizada em várias aplicações e projetos importantes, especialmente nas áreas de telecomunicações e sistemas de manufatura [24].

O conceito de Rede de Petri Colorida utilizadas no *CPN Tools* é complicado. Tais redes são chamadas frequentemente de Redes de Petri Coloridas generalizadas porque o tipo de *token* é descrito como um tipo de dado abstrato, como uma linguagem de programação. As Redes de Petri com restrições de tempo utilizam o conceito de tempo para representar a duração de ações, tarefas e eventos em um sistema real. O disparo de uma transição em uma Rede de Petri com restrições de tempo representa o retardo de tempo relativo à própria dinâmica de um sistema real [24].

No *CPN Tools*, a descrição de um modelo é constituída da combinação da estrutura da Rede de Petri com a linguagem de programação *CPN ML (Markup Language)*. Essa linguagem de programação especial é incluída para a descrição de atributos dos elementos da rede. Esta linguagem fornece declarações de conjuntos coloridos (tipos), variáveis, constantes, funções e procedimentos. Por exemplo, na Figura 5.2 as seguintes declarações foram utilizadas:

---

```
colset INT = int;  
var x, y : INT;
```

---

**Figura 5.2:** Declarações no *CPN Tools*

Um conjunto de cores foi definido: INT, do tipo inteiro. Assim, todos os lugares da rede possuem o conjunto de cores INT associado. Foram também definidas as variáveis *x* e *y*, as quais são do tipo INT.

### 5.1.2 Modelo de Simulação usando Redes de Petri

O modelo de simulação desenvolvido baseia-se no modelo conceitual apresentado como estudo de caso na Seção 5.4. A Figura 5.3 ilustra o modelo de simulação desenvolvido a partir do uso de Redes de Petri Coloridas e Temporizadas.

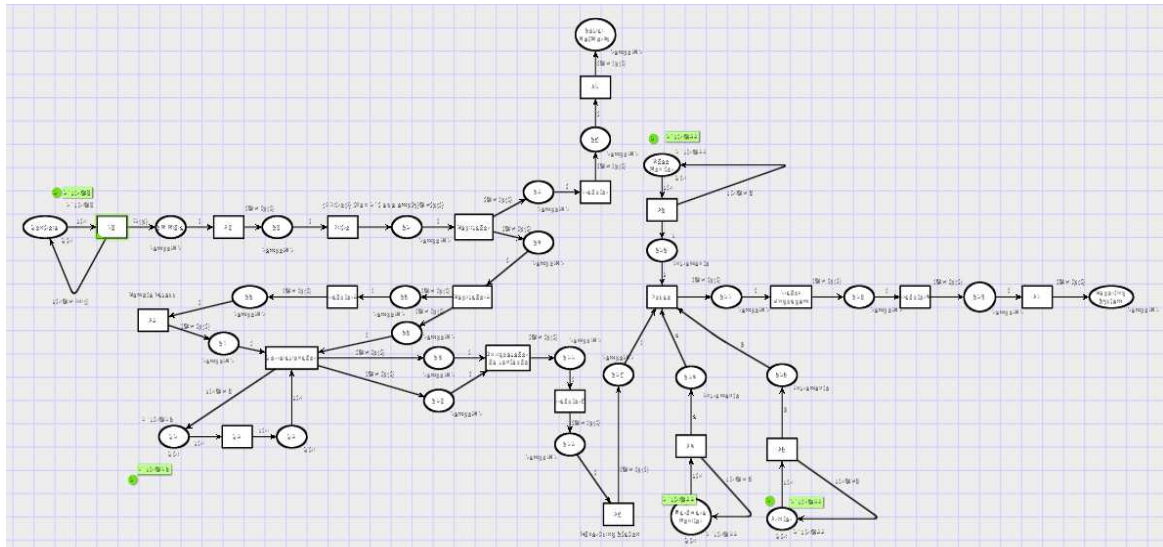


Figura 5.3: Modelo de Simulação em Redes de Petri

A Figura §5.3 foi dividida em duas partes, nas Figuras §5.4 e §5.5, para melhor visualização do modelo de simulação.

A transição P0 está modelando a porta P0 que é o início do fluxo de trabalho, no qual apresenta as mensagens que estarão sendo inseridas pela aplicação *SM Midia*. As transições P1 e P7 representam as portas de saída para as aplicações *Sociais NetWorks* e *Reporting System*, respectivamente. Já as transições P4, P5 e P6 modelam as portas de entrada das aplicações *Hardware Monitor*, *Printer* e *Video Monitor*, e as transições P2 e P3 representam portas de solicitação das aplicações *Remote Access* e *Advertising Station*.

Cada um dos elementos gerados em Redes de Petri foram analisados e estão explicados em detalhes na sequência do trabalho.

Ao realizar a transcrição das tarefas da tecnologia Guaraná para Redes de Petri sua funcionalidade é mantida. A porta de entrada P0 (Figura §5.6), tem a função de inserir mensagens na solução para serem processadas. Sua representação em Redes de Petri equivale a transição P0 que quando recebe um *token* do controlador encaminha para o lugar S0, que representa o *slot* S0, a semântica  $t@ + tp(t)$  verifica quanto tempo a mensagem (*token*) precisou para ser processada. A porta de saída (Figura §5.7, atua enviando mensagens para as aplicações integradas, seu grafo equivalente em Rede de Petri é



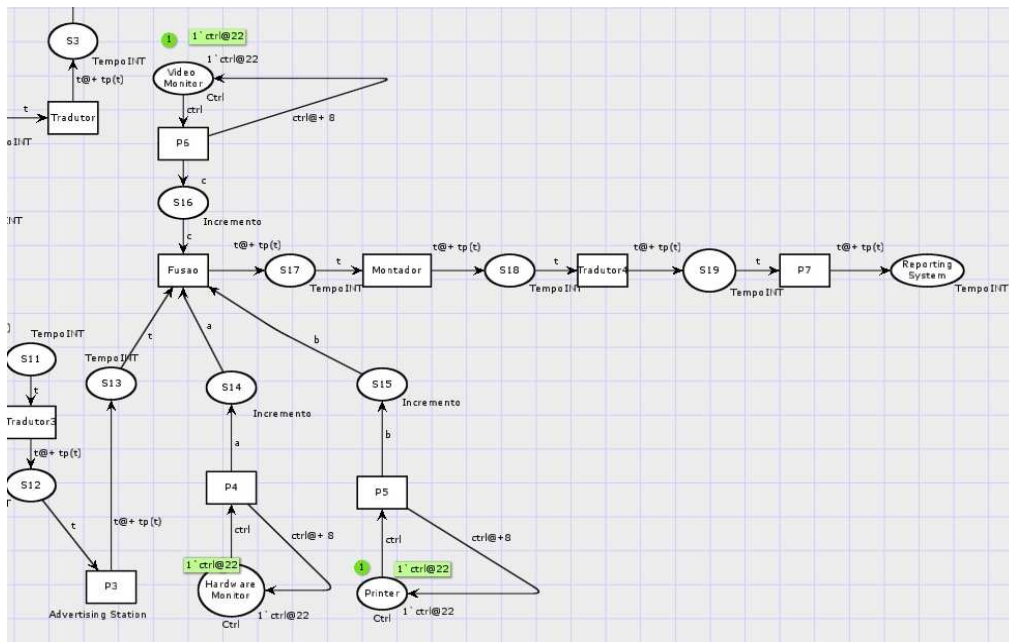


Figura 5.5: Modelo de Simulação em Redes de Petri

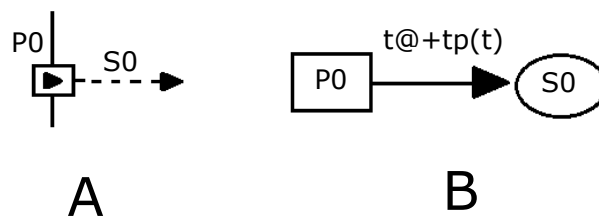


Figura 5.6: **A:** Porta de entrada em Guaraná **B:** Porta de entrada em Redes de Petri

(iffiltro(f)then1'telseempty)@tp(t), conforme Figura 5.9.

A tarefa Replicador faz cópias da mensagem original, quantas forem necessárias, sem alterar seu conteúdo, e as direciona para seguir o fluxo da solução. Em Redes de Petri ela está representada por um lugar de entrada, uma transição Replicador e dois lugares de saída, de acordo com a Figura 5.10.

A tarefa Tradutor adapta a mensagem para um formato que a aplicação que irá recebe-la a entenda. Como a mensagem é analoga ao *token*, o Tradu-

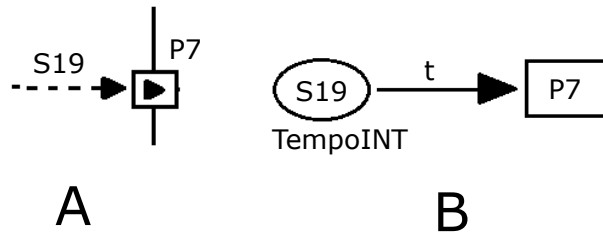


Figura 5.7: **A:** Porta de saída em Guaraná **B:** Porta de saída em Redes de Petri

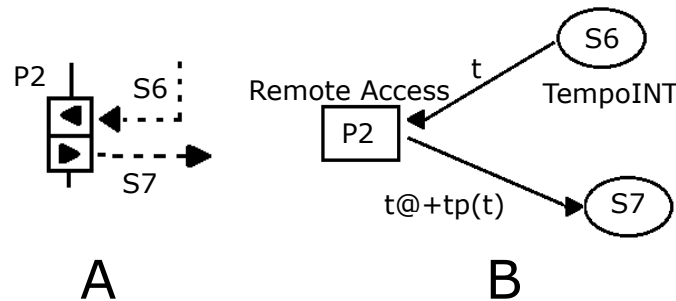


Figura 5.8: **A:** Porta de solicitação em Guaraná **B:** Porta de solicitação em Redes de Petri

tor é representado por um lugar de entrada, uma transição Tradutor e um lugar de saída, que expressam a entrada, processamento e saída da mensagem, no qual  $t$  é a variável que representa o tipo de mensagem que pode ser processada, como mostra a Figura §5.11.

O Correlacionador é uma tarefa muito importante dentro do fluxo de mensagens, pois consegue localizar mensagens correlacionadas para que possam ser processadas em conjunto. Na Figura §5.12 essa tarefa é representada pela transição Correlacionador, pelos lugares C1 e C2 e pela transição C2. Os lugares C1 e C2 e a transição C2 representam um atraso no *token* que chega na transição Correlacionador e espera até encontrar o seu *token* correspondente. Essa correspondência é feita por meio do tempo de cada *token*, quando este tempo é igual, o *token* é enviado aos lugares S9 e S10. Mantendo assim, a funcionalidade dessa tarefa, pois, no fluxo de mensagens não se encontra imediatamente a mensagem a ser correlacionada. As semânticas  $ctrl@ + 8$ ,  $ctrl@ + 16$  serão explicadas na sequência.

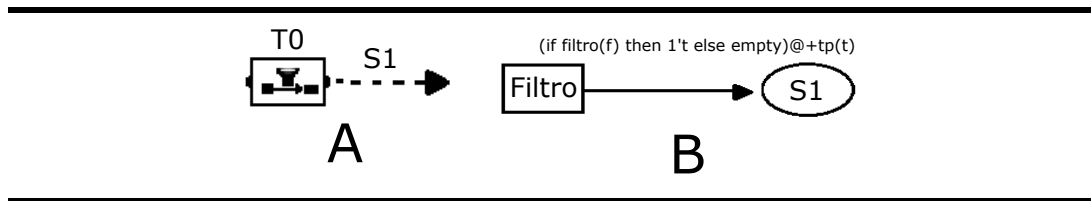


Figura 5.9: A: Tarefa Filtro em Guaraná B: Tarefa Filtro em Redes de Petri

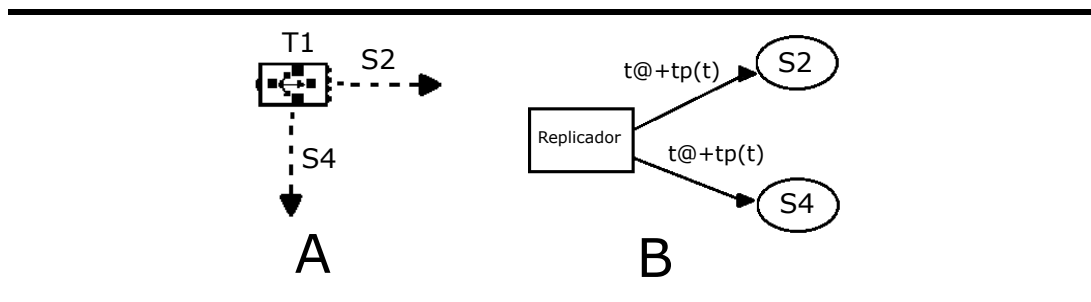


Figura 5.10: A: Tarefa Replicador em Guaraná B: Tarefa Replicador em Redes de Petri

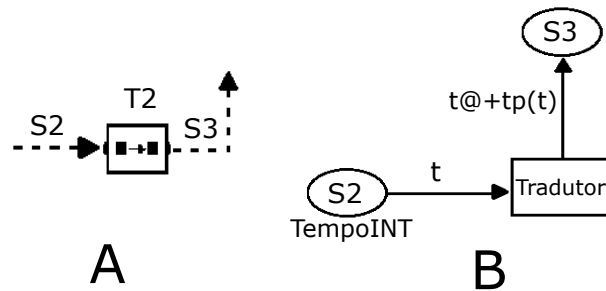
A tarefa Enriquecedor de Conteúdo recebe mensagens correlacionadas e as combina em uma única. Em Redes de Petri o grafo que a expressa essa tarefa está na Figura §5.13, que é a transição Enriquecedor de Conteúdo, que dispara quando houver ao menos um *token* em cada um dos lugares S9 e S10, e é colocado apenas um *token* no lugar S11, representando a funcionalidade dessa tarefa.

A tarefa Fusão tem como função colocar as mensagens oriundas de vários *slots* em um único, está representada pela Figura §5.14, na qual recebe mensagens dos *slots* S13, S14, S15 e S16 e as encaminha para a transição Fusão que agrupa todas no *slot* S17, as diferentes mensagens estão representadas pelas variáveis *a*, *b* e *c*.

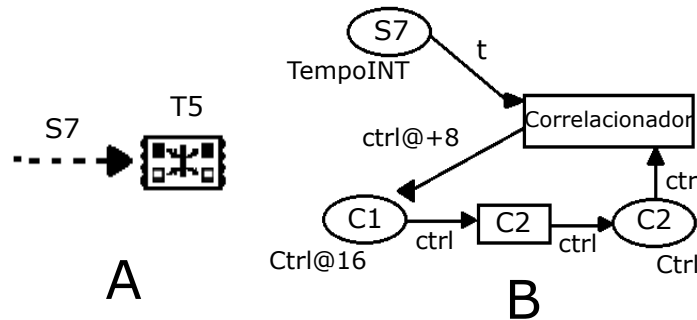
A tarefa Montador tem a função de construir uma nova mensagem a partir de duas ou mais mensagens. A Figura §5.15 representa esta tarefa.

### 5.1.3 Formulação Matemática

A representação formal para o modelo de simulação expresso nas Figuras §5.4 e §5.5 é expressa pela quintupla  $P, T, I, O, K$ .  $P$  representa o conjunto



**Figura 5.11:** *A: Tarefa Tradutor em Guarani B: Tarefa Tradutor em Redes de Petri*



**Figura 5.12:** *A: Tarefa Correlacionador em Guarani B: Tarefa Correlacionador em Redes de Petri*

de lugares conforme Equação 1.  $T$  representa o conjunto de transições de acordo com a Equação 2.  $I$  representa o mapeamento dos arcos de entrada e  $O$  os arcos de saída, expressos pelas Equações 3 e 4, respectivamente.  $K$  representa a capacidade associada que cada lugar possui. Neste modelo todos os lugares possuem a mesma capacidade associada, por isso sua representação foi suprimida.

$$P = \{S0, S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18, S19\} \text{ Equação 1}$$

$$T = \{P0, \text{Filtro}, \text{Replicador}, \text{Tradutor}, P1, \text{Replicador2}, \text{Tradutor2}, P2, \text{Correlacionador}, \text{Enriquecedor de conteudo}, \text{Tradutor3}, P3, P4, P5, P6, \text{Fusao}, \text{Montador}, \text{Tradutor4}, P7\} \text{ Equação 2}$$

$$I = \{(SMMidia, P0), (S0, \text{Filtro}), (S1, \text{Replicador}), (S2, \text{Tradutor}), (S3, P1), (S4,$$



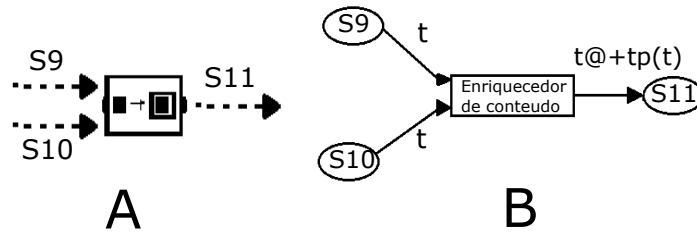


Figura 5.13: A: Tarefa Enriquecedor de Conteúdo em Guaraná B: Tarefa Enriquecedor de Conteúdo em Redes de Petri

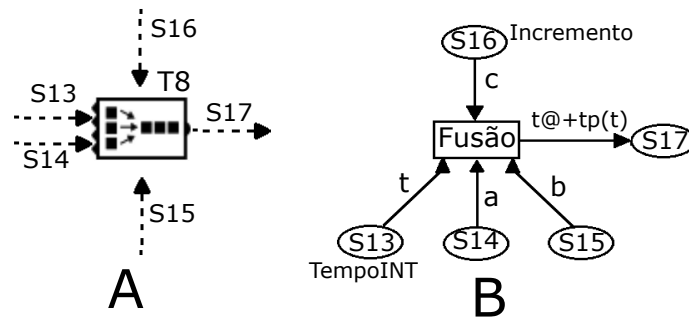


Figura 5.14: A: Tarefa Fusão em Guaraná B: Tarefa Fusão em Redes de Petri

Replicador2), (S5, Tradutor2), (S6, P2), (S7, Correlacionador), (S8, Correlacionador), (S9, Enriquecedor de conteúdo), (S10, Enriquecedor de conteúdo), (S11, Tradutor3), (S12, P3), (S13, Fusao), (S14, Fusao), (S15, Fusao), (S16, Fusao), (Video Monitor, P6), (S17, Montador), (S18, Tradutor4), (S19, P7)} Equação 3

$O = \{(P0, S0), (Filtro, S1), (Replicador, S2), (Replicador, S4), (Tradutor, S3), (P1, SocialNetworks), (Replicador2, S5), (Replicador2, S8), (Tradutor2, S6), (P2, S7), (Correlacionador, S9), (Correlacionador, S10), (Enriquecedor de conteúdo, S11), (Tradutor3, S12), (P3, S13), (P4, S14), (P5, S15), (P6, S16), (Fusao, S17), (Montador, S18), (Tradutor4, S19), (P7, Reporting System)\}$  Equação 4

O valor das taxas de disparo das transições podem ser descritas conforme a Equação 5, considerando a ordem do conjunto T. As taxas de disparo significam a porcentagem de disparo, mas são representadas e implementadas em simuladores na forma decimal. A tarefa Filtro precisa eliminar mensagens repetidas, por isso sua taxa de disparo é de 95%, ou seja, 5% das mensagens



Sendo assim, foram analisadas as variáveis de tempo, por meio da Equação 6 e o tamanho máximo do acúmulo de mensagens nos *slots*. Sabendo que, o tempo de processamento das tarefas é fixo e os *slots* não são FIFO (First In, First Out).

$$(\text{Média dos tempos} = (\sum_{\text{tempos finais}} - \sum_{\text{tempos iniciais}}) / n) \text{ Equação 6}$$

Para isso, foram definidos três experimentos, com 1000, 5000 e 10000 mensagens, que equivalem aos *tokens* no modelo de simulação em Rede de Petri. Para cada experimento foram criados três cenários, como apresentado na Tabela §5.16. Cada cenário foi simulado 25 vezes para excluir qualquer discrepância nos dados [20]. Esses cenários foram aplicados para dois tipos de Correlacionadores, um com tempo de atraso igual a 16 (baixo) e outro igual a 20 (médio). Além disso, cada prioridade (cor) foi simulada separadamente, no modelo de simulação com Correlacionador de tempo 20, de acordo com a Tabela §5.17.

	Cenário 1		Cenário 2		Cenário 3	
	<i>Tokens</i>	Tempo	<i>Tokens</i>	Tempo	<i>Tokens</i>	Tempo
Experimento 1000 <i>tokens</i>	300	1	500	1	200	1
	300	2	250	2	600	2
	400	3	250	3	200	3
Experimento 5000 <i>tokens</i>	1000	1	3000	1	1500	1
	2000	2	1000	2	2000	2
	2000	3	1000	3	1500	3
Experimento 10000 <i>tokens</i>	3000	1	5000	1	2000	1
	3000	2	2500	2	6000	2
	4000	3	2500	3	2000	3

Figura 5.16: Tabela dos cenários de cada experimento

O modelo de simulação foi implementado na ferramenta de simulação e edição de Redes de Petri *CPN Tools*. A ferramenta de simulação apresenta o tempo médio de permanência das mensagens em cada lugar e o tamanho máximo de mensagens acumuladas nos lugares. Os lugares estão representando os *slots* do modelo conceitual da solução de integração.

Para este modelo, foi criado o conjunto *TempoINT* para as cores Alta, Média e Baixa. Estas, indicam a prioridade de cada mensagem, sem considerar seu tamanho. Com a criação do conjunto *TempoINT*, foi inserida a variável *t*, a qual é um elemento deste conjunto. A declaração do conjunto *TempoINT* possui restrições de tempo, então precisa ser acompanhado pelo modificador

---

	<i>Tokens</i>	<i>Tempo</i>
<b>Experimento 1</b>	1000	1
<b>Experimento 2</b>	1000	2
<b>Experimento 3</b>	1000	3
<b>Experimento 4</b>	5000	1
<b>Experimento 5</b>	5000	2
<b>Experimento 6</b>	5000	3
<b>Experimento 7</b>	10000	1
<b>Experimento 8</b>	10000	2
<b>Experimento 9</b>	10000	3

---

**Figura 5.17:** Tabela dos cenários do experimento de cada prioridade

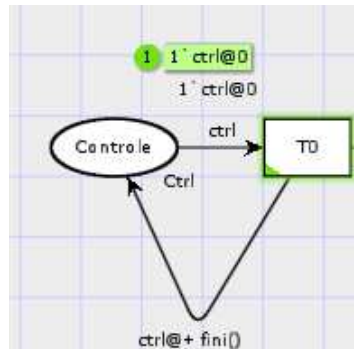
*timed*. Associar um tempo a uma cor, significa adicionar um valor de tempo a essa cor que é igual ao tempo atual do modelo mais o tempo associado a cor.

Cada cor é processada em um tempo diferente, a cor Alta é processada de um em um (unidade de tempo), a cor Média de dois em dois e a cor Baixa de três em três. Para isso foram criadas as funções  $tp2(x)$ ,  $tpx$ ,  $f2(r)$  e  $f1(r)$ .

Para que os *tokens* fossem lançados de forma aleatória no modelo, um Controle foi construído conforme a Figura §5.18. Isto significa que, a cada 1, 2 ou 3 unidades de tempo, o Controle lança um *token* para a porta de entrada do sistema de forma aleatória baseado no valor de sua prioridade. O incremento do tempo representa um intervalo em que acontecem as entradas de *tokens* no sistema. O conjunto *Ctrl* e a função  $fini()$  representam o atraso de entrada de cada *token*.

Para a tarefa Filtro foi necessário criar o conjunto  $F$  (*colset*  $F = int$  with 1..20), que é constituído de números inteiros entre 1 e 20, incluindo o 1 e o 20. Com a criação do conjunto  $F$  foi criada a variável  $f$  (*var*  $f: F$ ), a qual é um elemento deste conjunto. Posteriormente, foi criada a função filtro (*fun*  $filtro(x) = (x \leq 19)$ ), que é uma função booleana, a qual, retorna um valor verdadeiro se o valor de  $x$  for menor ou igual a 19 e falso se for igual a 20. Isso significa que 95% das mensagens irão seguir o fluxo e 5% não, funcionando como um filtro. Os números 19 e 20 representam essa porcentagem. Esses valores são somente uma estimativa para testar o modelo. O percentual de descarte da tarefa Filtro em Redes de Petri pode ser modificado.

A tarefa Correlacionador foi desenvolvida conforme a Figura §5.12 em Redes de Petri. A transição Correlacionador com o auxílio dos lugares C1 e C2 e



**Figura 5.18:** Controle de lançamento de tokens

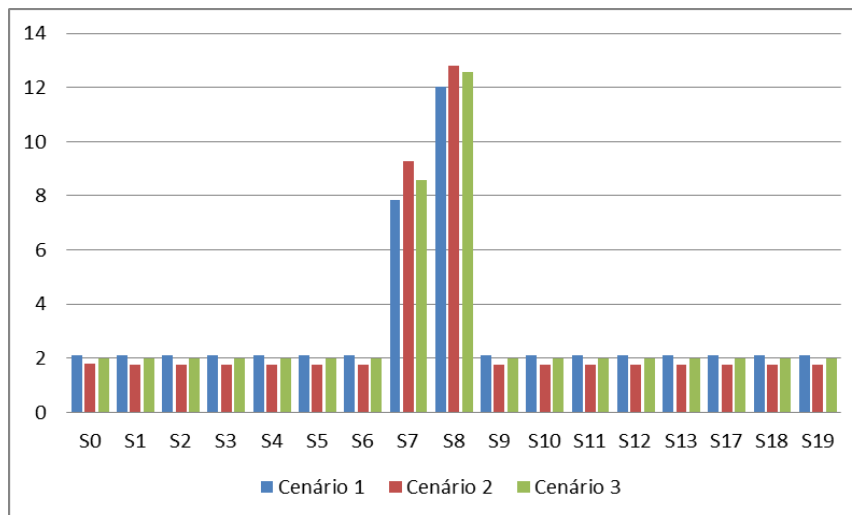
transição C2 formam o Correlacionador, que ao receber um *token* do lugar S7 aguarda 16 unidades de tempo, que significa o tempo médio que o primeiro *token* precisa para correlacionar. Este tempo foi calculado por meio do número de passos, ou seja, quantos lugares o *token* percorre até chegar ao S7 (6 passos) pelo tempo médio, 2 (média dos tempos associados aos *tokens*). Sendo realizado um ajuste de mais 4 unidades de tempo para representar de forma mais real o sistema, chegando em 16 (Ctrl@16). Para o Correlacionador de tempo de atraso 20 foi apenas acrescentado mais 4.

Para representar o cenário de cada um dos experimentos foi criado a variável  $p$ , que pertence aos conjuntos Int1, Int2 e Int3, que por meio de uma proporção representam as quantidades de *tokens* definidas para cada unidade de tempo.

Para gerar as entradas das aplicações Vídeo Monitor, Hardware Monitor e Printer, foi desenvolvido algo semelhante ao Controle da porta de entrada. Foi criado um conjunto Incremento e as variáveis  $a$ ,  $b$  e  $c$  que pertencem a este conjunto, no qual os *tokens* vão sendo inseridos aleatoriamente entre as cores Alta, Média e Baixa. Porém, o tempo é fixo em uma unidade de tempo e possui um atraso maior, de 22, para esperar a chegada do primeiro *token* do início do processo. Esse tempo foi calculado de maneira similar ao Correlacionador, no caso deste são necessários 9 passos até chegar a aplicação Advertising Station, este valor foi multiplicado pelo tempo médio de 2 e adicionado 4 unidades de tempo para representar melhor o sistema.

### 5.3 Resultados e Discussão

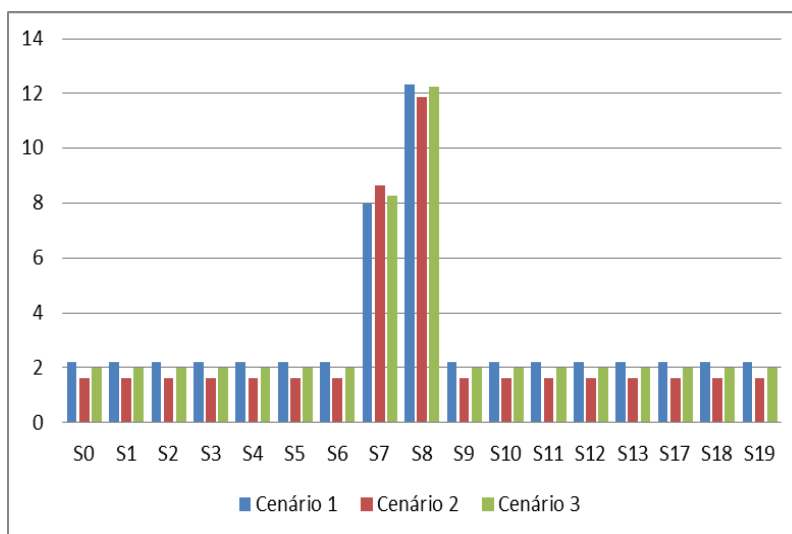
Após realizada a configuração do modelo de simulação na ferramenta *CPN Tools* foram realizados os experimentos, nesta mesma ferramenta. Inicialmente foram feitas simulações com 1000 mensagens, apresentando os resultados da variável tempo médio de permanência das mensagens nos *slots* nos três cenários. Destaca-se que foram colocados tempos de processamento diferentes nas mensagens, conforme a prioridade de cada uma, obtendo o seguinte Figura 5.19.



**Figura 5.19:** Tempo médio de permanência das mensagens nos slots (1000 tokens)

Em ambos os cenários nota-se que a mensagem leva mais tempo para ser processada nos *slots* S7 e S8, porque a mensagem precisa esperar para ser correlacionada, pois, a transição que representa a porta de solicitação P2 é assíncrona, isto significa que, faz diversas solicitações à aplicação, sem que tenha que aguardar o retorno de uma solicitação para fazer a próxima, sem seguir uma ordem, as mensagens não precisam sair na mesma ordem que entraram. Com isso, é necessário o uso da tarefa Correlacionador, que acaba provocando um atraso, porque, deve aguardar as mensagens que possuem a mesma identidade para correlacionar.

Esta mesma situação se repete no experimento com 5000 e 10000 mensagens, tendo o mesmo comportamento como pode-se perceber com a



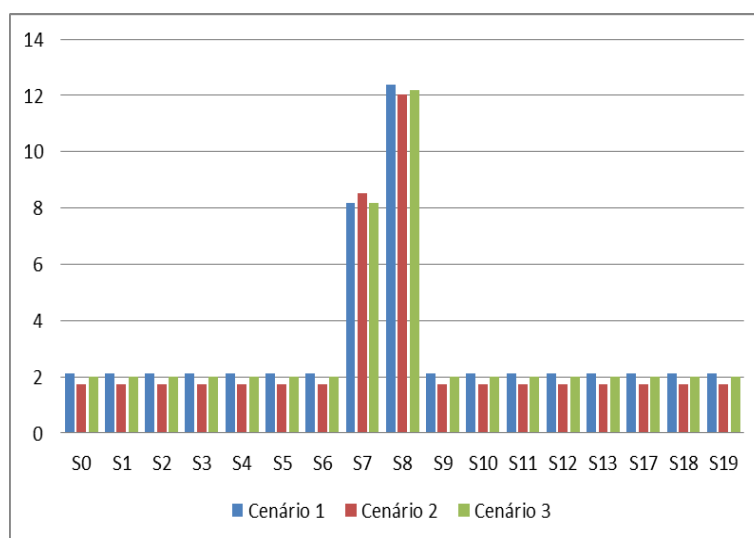
**Figura 5.20:** Tempo médio de permanência das mensagens nos slots (5000 tokens)

Figura §5.20 e a Figura §5.21 para os três cenários. É importante destacar que os *slots* S14, S15 e S16 não constam nas análises, pois as mensagens oriundas desses *slots* não necessitam de nenhum processamento, são apenas lançadas com as informações dos *status* de cada uma das aplicações em que estão conectados e postas em um único *slot*(S17) juntamente com a mensagem vinda do *slot* S13.

Quanto ao acúmulo máximo de mensagens nos *slots*, obteve-se os seguintes resultados apresentados na Figura §5.22 para 1000 mensagens e na Figura §5.23 para 5000 e 10000 mensagens, constatando que para esta última figura o comportamento de 5000 e 10000 mensagens foi o mesmo e semelhante ao de 1000 mensagens. Sendo estes acúmulos pequenos.

O Correlacionador influencia fortemente no comportamento do sistema. Percebe-se isso por meio das simulações realizadas. Com o tempo de espera igual a 16 demonstrou-se um acúmulo muito pequeno. Ao ajustar o tempo do Correlacionador para 20, sendo um tempo médio, nota-se que o acúmulo aumenta consideravelmente, em ambos os *slots* que estão ligados ao Correlacionador, *slots* S7 e S8, enquanto que nos outros *slots* analisados foram praticamente os mesmos (variando entre 1 e 2). Conforme a Figura §5.24 mostra.

Em relação ao tempo médio de processamento das mensagens tam-



**Figura 5.21:** Tempo médio de permanência das mensagens nos slots (10000 tokens)

bém houve alterações, como pode-se observar na Figura §5.25 . Sendo assim, faz sentido, pois como o tempo do Correlacionador foi alterado para 20, o tempo de médio de processamento das mensagens aumentou, gerando os gargalos de desempenho mostrados na Figura §5.24.

Com as duas simulações realizadas, Correlacionador de tempo 16 e tempo 20, confirma-se que o acúmulo será possivelmente nos *slots* S7 e S8, que estão conectados ao Correlacionador.

Os resultados das simulações feitas para cada prioridade são apresentados na Figura §5.26, que mostra o tempo médio de processamento das mensagens em cada *slot*. E na Figura §5.27, mostra os acúmulos de cada prioridade nos *slots*. Sendo que para essas prioridades não houve mudanças significativas em relação aos resultados apresentados com as prioridades simuladas todas juntas.

Sendo assim, faz sentido que o acúmulo de mensagens seja maior nos *slots* S7 e S8, já que necessitam de mais tempo para serem processadas, como constatado acima, em todas as simulações realizadas, devido a tarefa Correlacionador, que tem como semântica correlacionar mensagens, isto é, procurar mensagens com as mesmas características e juntá-las em uma única mensagem. Deste modo, esses *slots* mostram-se possíveis gargalos de desempenho desta solução de integração da área de publicidade e propaganda.



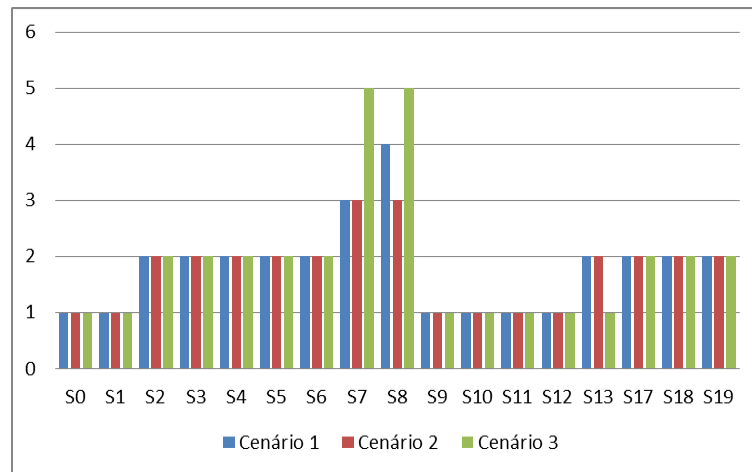


Figura 5.22: Acúmulo máximo de mensagem nos slots (1000 tokens)

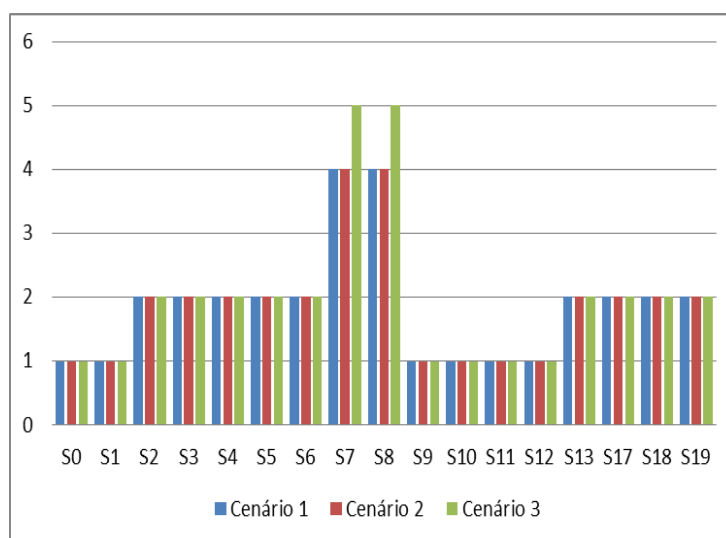
## 5.4 Resumo do Capítulo

Nesse capítulo, foi apresentado a transcrição do modelo conceitual da solução de integração de um problema da área de Publicidade e Propaganda para um modelo de simulação em Redes de Petri equivalente de acordo com a lógica observada por Roos-Frantz et al. [36]. Porém, nesse trabalho o modelo conceitual foi transcrito para Rede de Petri Colorida/Temporizada.

Foi tratado algumas características da ferramenta de edição e simulação *CPN Tools*, que foi desenvolvida principalmente para Redes de Petri Colorida, podendo também usar o tempo como variável para análise. Apresentou-se a formulação matemática da Rede de Petri Colorida/Temporizada do modelo de simulação construído, na forma de conjuntos.

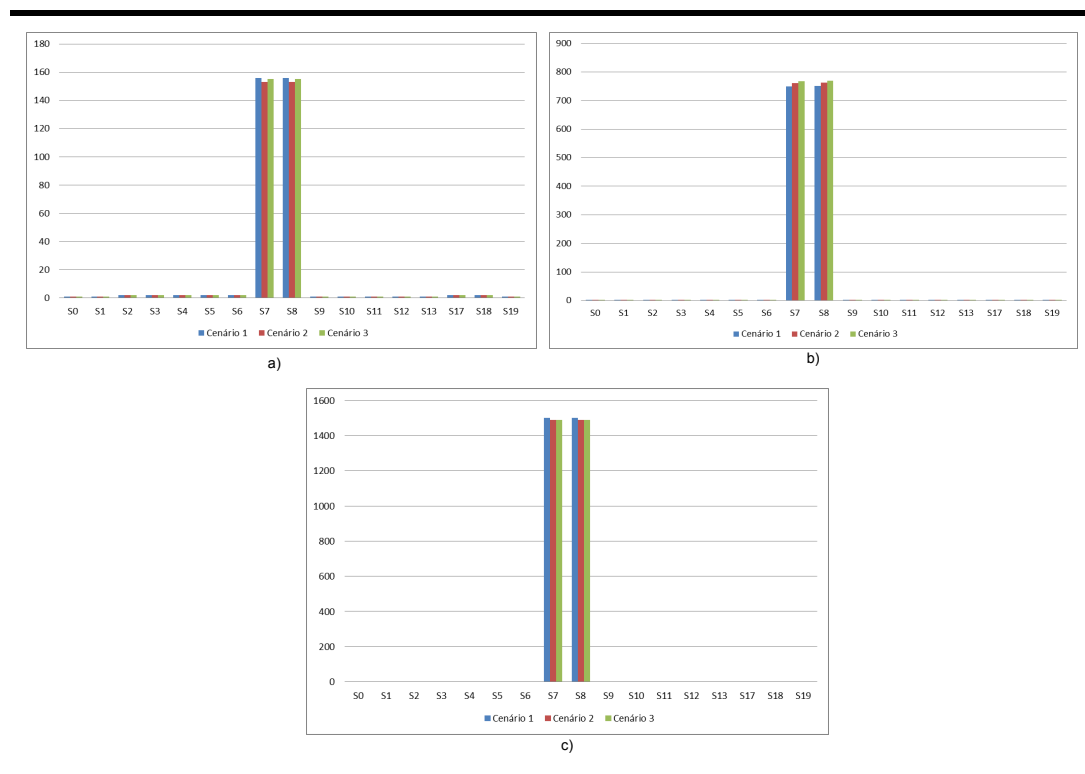
Foram apresentadas as variáveis observadas no experimento, tempo de permanência das mensagens nos lugares (*slots*), ou seja, o tempo de espera para essa ser processada e o acúmulo de mensagens nos lugares, as filas formadas nos *slots*. Mostrou-se, também, como foi feita a preparação do modelo de simulação no *CPN Tools*, para então submeter aos três experimentos e seus cenários.

Após, foi apresentado a análise dos resultados obtidos em cada um dos experimentos e seu respectivos cenários. E percebeu-se que os *slots* S7 e S8

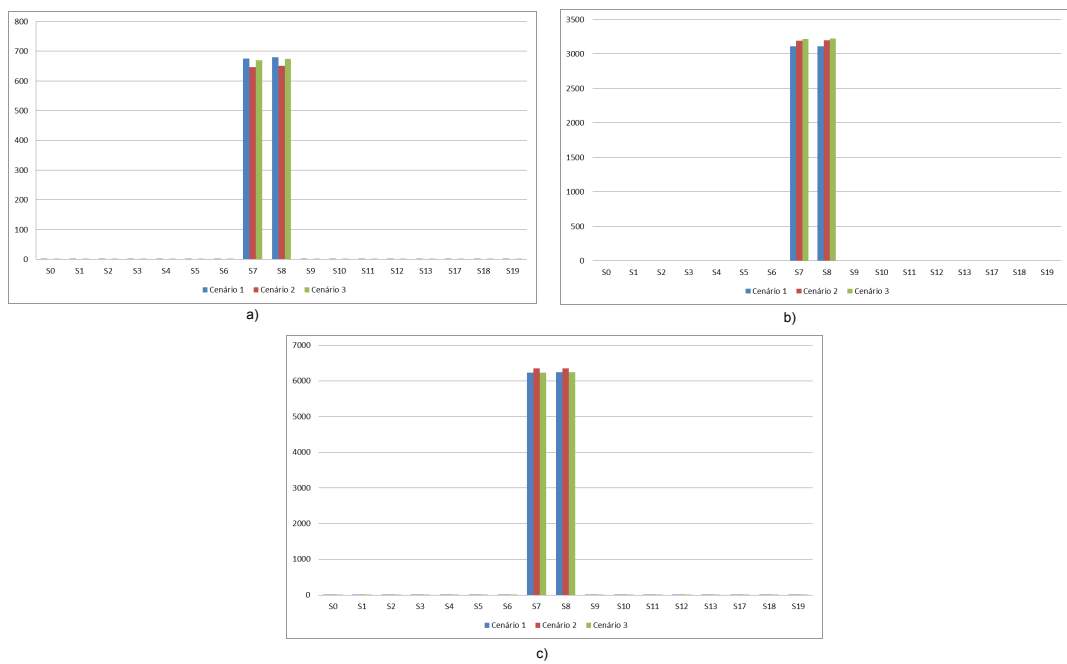


**Figura 5.23:** Acúmulo máximo de mensagem nos slots (5000 e 10000 tokens)

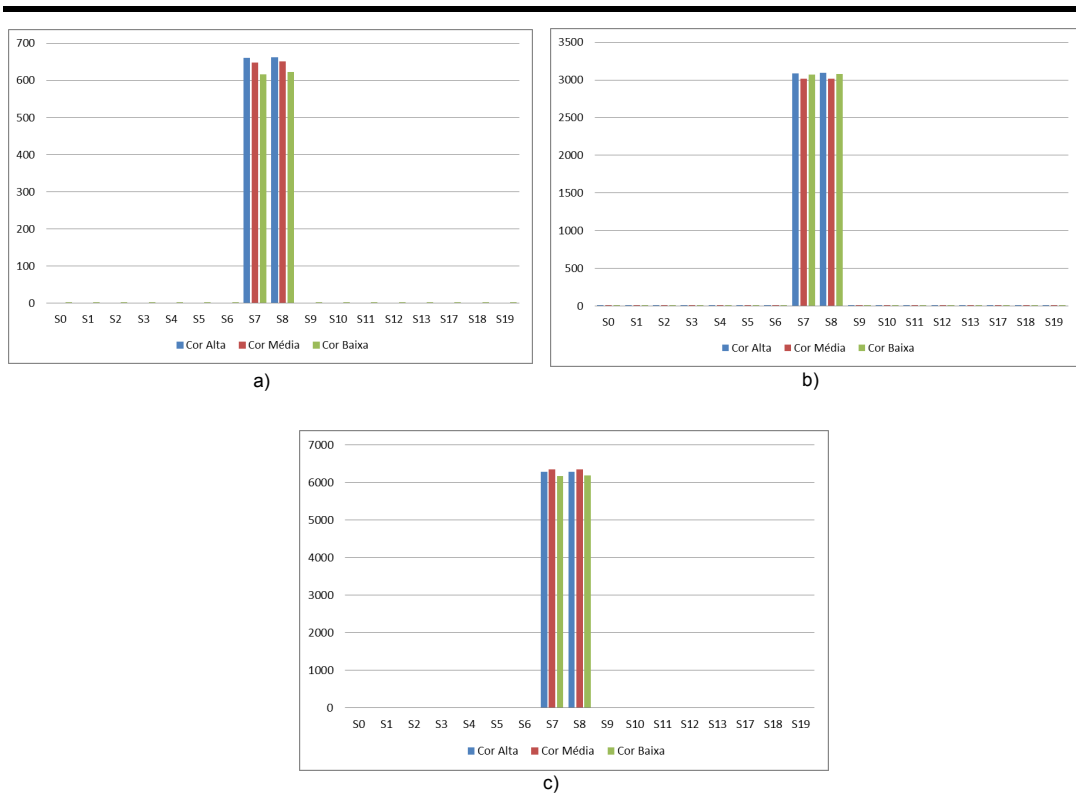
são possíveis gargalos de desempenho da solução de integração, pois entre esses slots há uma tarefa que precisa correlacionar as mensagens, o Correlacionador, demandando assim, de mais tempo que as outras tarefas.



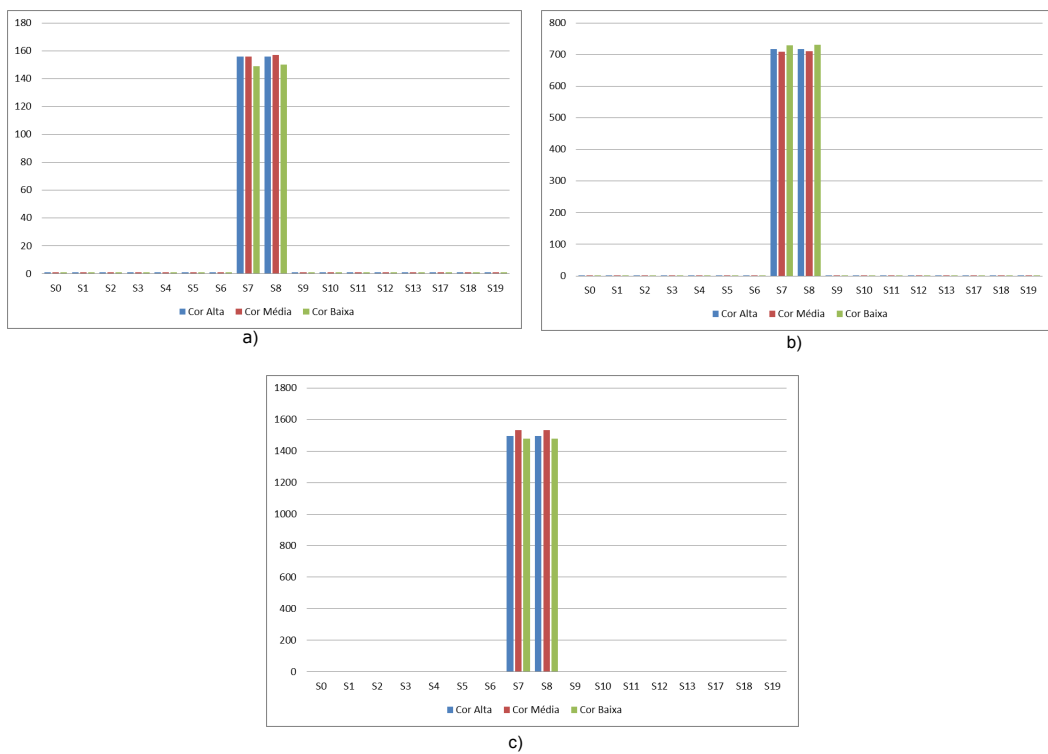
**Figura 5.24:** Acúmulo máximo de mensagem nos slots - Correlacionador de tempo 20. a) 1000 tokens b) 5000 tokens e c) 10000 tokens



**Figura 5.25:** Tempo médio de permanência das mensagens nos slots - Correlacionador de tempo 20. a) 1000 tokens b) 5000 tokens e c) 10000 tokens



**Figura 5.26:** Tempo médio de permanência das mensagens nos slots - Correlacionador de tempo 20. a) 1000 tokens b) 5000 tokens e c) 10000 tokens



**Figura 5.27:** Acúmulo das mensagens nos slots- Correlacionador de tempo 20. a) 1000 tokens b) 5000 tokens e c) 10000 tokens

---

# Capítulo 6

## Verificação Formal do Modelo

---

*Sempre que fizer algo, mesmo que ninguém venha a saber,  
faça como se o mundo estivesse olhando para você.*

*Thomas Jefferson, filósofo político (1743-1826)*

**E**sse capítulo descreve o conceito de verificação e validação na Seção §6.1, além de trazer as técnicas de verificação e validação encontradas na literatura, na Seção §6.2. E ainda, na Seção §6.3 descreve como foi realizada a verificação formal do modelo proposto neste trabalho.

### 6.1 Definição de Verificação e Validação

Um modelo de simulação expressa o funcionamento de um sistema real, no qual se analisa o seu comportamento em diferentes cenários, com o propósito de verificar se este possui erros, antes de ser implementado. Para tanto, é importante saber se este modelo de fato representa o sistema, para isto o modelo de simulação precisa ser verificado e validado.

A verificação tem a finalidade de encontrar e corrigir erros de modelagem. Refere-se às técnicas utilizadas para garantir que o modelo está correto e de acordo com qualquer pressupostos e simplificações realizados. Deste modo, a verificação consiste em assegurar que o modelo está livre de erros de implementação computacional [45].

A validação acontece quando o desenvolvedor do modelo e indivíduos conhecedores do sistema real trabalham juntos para analisar e avaliar como o

modelo funciona. Refere-se às técnicas utilizadas para garantir que o modelo representa o sistema real a um nível suficiente de precisão [6].

Carson [6] destaca que nenhum modelo é totalmente verificado e validado. Pois, um modelo formal de simulação é uma representação do sistema, e o comportamento do modelo, é no máximo, uma aproximação do comportamento do sistema real. A confirmação de que o modelo foi verificado ou validado consiste em uma série de tarefas realizadas especificadamente para verificar e validar o modelo.

De acordo com Bukh e Jain [4], um modelo de simulação, após a sua elaboração e simulação, pode estar entre quatro categorias:

- i. Não validado e não verificado;
- ii. Não validado e verificado;
- iii. Validado e não verificado;
- iv. Validado e verificado.

Para verificar e validar o modelo de simulação proposto, é necessário ter conhecimento das técnicas de verificação e validação existentes na literatura. Para analisar qual ou quais delas podem ser usadas para verificar e validar o modelo desta dissertação. Assim, a Seção §6.2 traz uma breve revisão bibliográfica sobre essas técnicas.

## 6.2 Técnicas de Verificação e Validação

A literatura apresenta diversas técnicas de verificação e validação de modelos de simulação quando há dados do sistema real. Trabalhos que relatam técnicas para verificação e validação de modelos sem dados reais são escassos. A modelagem proposta nesta dissertação não apresenta dados no sistema real. As técnicas aplicadas para a verificação e validação deste modelo foram propostas por Kleijnen [26] e por Sargent [39].

A técnica proposta por Kleijnen [26] aborda como verificar e validar modelos de simulação através da aplicação de técnicas estatísticas, na qual o tipo de técnica aplicada depende da disponibilidade de dados sobre o sistema real. Para a verificação do modelo precisa-se de conhecimento especializado, sendo assim, a verificação se dará por meio do conhecimento de especialistas sobre o comportamento do sistema.



Este conhecimento pode ocorrer de duas maneiras, pelo conhecimento quantitativo ou qualitativo. O conhecimento quantitativo de um modelo de simulação é desenvolvido quando o efeito é conhecido, não sua magnitude e a análise estatística ocorrerá diante de resultados quantitativos obtidos nas simulações. Já o conhecimento qualitativo representa as características que os especialistas do sistema modelado esperam que aconteçam.

O comportamento do sistema diante de determinadas situações é conhecido e interfere nos resultados do modelo de simulação que precisa seguir a mesma linha. Com dados numéricos obtidos das simulações é possível aplicar uma análise quantitativa e estimar com suporte estatístico a dimensão das suas estruturas.

Os resultados experimentais supostamente devem acompanhar o comportamento que os especialistas esperam, tanto do sistema como um todo, como de um determinado elemento. Por exemplo, a formação de fila em um determinado ponto do sistema ou o consumo maior de tempo para processamento de uma determinada tarefa.

A observação dos valores de entrada e saída do modelo também contribuem para uma verificação eficaz. Se a entrada e saída do modelo de simulação ultrapassam as características qualitativas, é necessário questionar se o modelo possui erros de modelagem e se as condições ou cenários estão adequados.

Sargent [39] propõe várias técnicas de verificação, sendo duas dessas adequadas para a situação de modelagem proposta, a técnica de Verificação do Modelo Computacional e a técnica de Validade de Eventos.

A técnica de Verificação do Modelo Computacional implica diretamente no modelo computacional de simulação. A verificação tem que garantir que o modelo de simulação foi implementado de forma correta na ferramenta. Deste modo, a técnica propõe a comparação do número de elementos que entrou no sistema com os que saíram. Partindo da ideia de que a entrada precisa ser igual aos elementos que ficaram acumulados somados aos que chegaram ao final do processo.

A técnica da Validade de Eventos consiste na ocorrência de eventos no modelo conceitual em comparativo a ocorrência dos mesmos no modelo de simulação. Um determinado evento que tem uma frequência de ocorrência no sistema real, para ser consolidada a verificação do modelo, é necessário que a taxa de ocorrência deste mesmo evento se aproxime quando realizado experimentos com o modelo de simulação.

### 6.3 Verificação do Modelo de Simulação

A verificação do modelo foi realizada por meio da técnica de Kleijnen [26], na qual um especialista da área determina o comportamento do sistema. Para um especialista, uma solução de integração é um sistema de mensagem, composto por unidades de armazenamento temporário denominado *slot*, no qual se espera que exista acúmulo de mensagens.

A tarefa Correlacionador (T5) é alimentada pelos *slots* S7 e S8. O *slot* S8 recebe a mensagem da tarefa Replicador (T3), já S7 depende do processamento da Porta P2. Com isso, espera-se que haja maior acúmulo de mensagens em S7 e S8, pois estes *slots* armazenam mensagens para serem correlacionadas pela tarefa T5. A característica da tarefa T5 é correlacionar as mensagens, assim, torna S7 e S8 supostos gargalos de desempenho. O experimento realizado com o modelo de simulação demonstrou que ao final da simulação gerou-se um acúmulo de *tokens* em seus lugares. Os lugares S7 e S8 acumularam um número de mensagens maior que os demais.

Seguindo a proposta de Kleijnen [26] a comparação das características esperadas com as encontradas no experimento, demonstra que o acúmulo de mensagens esperado nos *slots* se repetiu nos lugares, o que demonstra que o modelo de simulação se comportou muito próximo de sistema orientado a mensagem. A formação de fila era esperada nos *slots* que alimentam a tarefa Correlacionador (T5), neste caso os *slots* S7 e S8. De fato, o acúmulo nos lugares S7 e S8 foram maior que os demais, comportamento que era esperado para esses *slots* (S7 e S8). A comparação entre o comportamento esperado pelos especialistas da área e o experimento desenvolvido deixa evidente que existe semelhança entre o comportamento da solução de integração e o modelo de simulação.

A proposta de Sargent [39] para verificação do modelo é aplicada no próprio modelo de simulação. É proposto um experimento com 1000, 5000 e 10.000 *tokens*. Cada experimento foi executado uma vez, e sua intenção é descrever o fluxo de mensagem, com intuito de verificar se o modelo está implementado corretamente na ferramenta.

Como os *tokens* que estão antes da tarefa Filtro não foram filtrados, basta contar a quantidade de *tokens*. Como os *tokens* que estão depois da tarefa Filtro foram filtrados, é necessário contá-los e multiplicar o resultado pelo inverso de 0,95, e somar o resultados com os valores que ficaram acumulados

nos *slots* durante o processo, gerando a Tabela §6.1 para o Correlacionador com tempo de 16 e a Tabela §6.2 para o de tempo 20. Isso deve resultar em um valor aproximado a 1000, 5000 e 10.000, respectivamente, porque o filtro é probabilístico, por isso, não tem como garantir que vai dar exato a quantidade.

	Antes do Filtro	Depois do Filtro	Mais o acúmulo
<b>1000 tokens</b>	0	$938 * 1/0,95 = 987$	$987 + 3 = \mathbf{990}$
<b>5000 tokens</b>	0	$4691 * 1/0,95 = 4937$	$4937 + 4 = \mathbf{4941}$
<b>10000 tokens</b>	0	$9485 * 1/0,95 = 9984$	$9984 + 5 = \mathbf{9989}$

**Figura 6.1:** Tabela de verificação do modelo formal de simulação - Correlacionador de tempo 16

	Antes do Filtro	Depois do Filtro	Mais o acúmulo
<b>1000 tokens</b>	0	$796 * 1/0,95 = 837$	$837 + 156 = \mathbf{993}$
<b>5000 tokens</b>	0	$3999 * 1/0,95 = 4209$	$4209 + 750 = \mathbf{4959}$
<b>10000 tokens</b>	0	$8005 * 1/0,95 = 8426$	$8426 + 1501 = \mathbf{9927}$

**Figura 6.2:** Tabela de verificação do modelo formal de simulação - Correlacionador de tempo 16

Os valores utilizados para realizar a multiplicação pelo inverso de 0,95 foram os *tokens* que chegaram ao final do processo do modelo de simulação de cada experimento. Observando que o resultado dessa operação não dá exatamente 1000, 5000 e 10000, porque tem-se uma função probabilística no filtro.

## 6.4 Resumo do Capítulo

Este capítulo apresentou a diferença entre validação e verificação, trazendo seus conceitos e algumas técnicas utilizadas para validar e verificar modelos de simulação sem dados reais do sistema, bem como, a verificação de um modelo de simulação em Redes de Petri.

Para verificar o modelo de simulação proposto nesta dissertação foram usadas as técnicas pesquisadas por Kleijnen [26] e por Sargent [39]. Na qual a primeira envolve o conhecimento de um especialista na área para prever o comportamento da solução de integração, se o modelo de simulação se portar de modo similar, o modelo está correto. A segunda técnica analisa o próprio modelo de simulação.

A aplicação de uma ou mais técnicas de verificação garante que o modelo está livre de erros da implementação computacional. A utilização de uma ou mais técnicas de validação procura garantir que o modelo tenha um comportamento semelhante ao do sistema modelado. Nesse sentido, a verificação e a validação do modelo de simulação formal proposto ocasionou segurança para a realização da pesquisa desta dissertação.

---

## Capítulo 7

# Conclusões e Trabalhos Futuros

---

*If you can look into the seeds of time,  
And say which grain will grow and which will not;  
Speak then to me.*

*William Shakespeare, British author (1564-1616)*



Atualmente as empresas desenvolvem ou adquirem aplicações para gerenciar seus processos de negócios. Essas aplicações consistem no ecossistema de *software* da empresa, que geralmente não são criadas para trabalharem em conjunto, pois são heterogêneas. Para aproveitar de modo mais eficiente suas funcionalidades, faz-se necessário integrar essas aplicações, por meio de uma solução de integração. Esta solução pode ser desenvolvida por uma plataforma de integração. A tecnologia Guaraná é uma plataforma de integração que visa dar suporte na elaboração de modelos conceituais, implementar e executar soluções de integração.

Contudo, uma solução de integração pode conter erros. Para verificar se esta solução funciona, torna-se importante simulá-la antes da sua implementação, ou seja, ainda na fase de projeto. Isso evita gastos desnecessários e desperdício de tempo. A simulação é possível por meio de formalismos matemáticos, um deles são as Redes de Petri.

Esta dissertação analisou o comportamento de uma solução de integração antes da sua implementação. Com o intuito de ajudar os engenheiros de *software* a perceber os possíveis erros e gargalos de performance, na etapa de projeto, evitando gastos financeiros e de tempo, promovendo a qualidade da

solução de integração proposta, por meio de um modelo conceitual e um modelo de simulação. Nesse sentido as principais contribuições desta pesquisa são listadas abaixo:

- Contribuição 1: Análise do comportamento e identificação de possíveis gargalos de performance em soluções de integração, por meio da simulação do seu modelo conceitual, utilizando o formalismo matemático da Rede de Petri Colorida/Temporizada.
- Contribuição 2: Desenvolvimento de um modelo de simulação equivalente ao modelo conceitual elaborado pela tecnologia Guaraná.
- Contribuição 3: Verificação do modelo de simulação proposto por meio de técnicas formais existentes na literatura.

Para analisar o comportamento e identificar os possíveis gargalos de desempenho de uma solução de integração na área de publicidade e propaganda, por meio da simulação do seu modelo conceitual, utilizou-se o formalismo matemático da Rede de Petri Colorida/Temporizada. Para isso, foi preciso desenvolver um modelo de simulação equivalente ao modelo conceitual elaborado pela tecnologia Guaraná.

Para tanto foi necessário buscar conhecimentos na área de Integração de Aplicações Empresariais, tecnologia Guaraná e Redes de Petri. Também foi importante verificar na literatura quais pesquisas já haviam sido desenvolvidas utilizando Redes de Petri como formalismo matemático para elaborar modelos de simulação e quais ferramentas de simulação foram usadas para editar e simular Redes de Petri. Essa pesquisa foi essencial para a escolha do tipo de Rede de Petri e ferramenta de edição e simulação .

Com a ferramenta de simulação escolhida (*CPN Tools*) algumas dificuldades foram encontradas. Ele não é intuitivo e possui uma linguagem de programação (*CPN ML*). O conceito de Redes de Petri Colorida utilizadas no *CPN Tools* é mais complexo, porque o tipo de *token* é descrito como um tipo de dado abstrato, como uma linguagem de programação, então era preciso declará-lo na ferramenta.

Com o modelo de simulação pronto, foram realizadas as simulações e análises da solução de integração. Utilizando Redes de Petri Colorida e Temporizada foi possível verificar que a solução possivelmente terá um gargalo de desempenho nos *slots* S7 e S8 que estão conectados com a tarefa Correlacionador. Pois, foram onde apresentou-se maior acúmulo de mensagens e

também pela tarefa precisar de mais tempo para processar cada mensagem que era inserida. Este gargalo se deve também a própria semântica desta tarefa, que precisa correlacionar mensagens, de acordo com seu conteúdo.

Foi inserido valores no Correlacionador para simular uma situação real. Nesse caso, foi inserido um peso leve (16), o qual levou a um valor de gargalo (baixo). Na sequência, inseriu-se um valor médio (20) o qual acusou um gargalo muito maior. Sendo que todas as simulações, mesmo com peso leve (16) e peso médio (20) acusaram gargalos nos mesmos *slots*.

Na forma como este modelo foi elaborado e nas condições que foi submetido, o uso de prioridades não apresentou grande diferença. Os resultados foram similares. Quanto ao uso de tempo em relação ao acúmulo pode-se perceber que, no geral, quanto mais tempo de processamento da mensagem mais acúmulo.

Percebeu-se com este trabalho que Redes de Petri Colorida e Temporizada mostraram-se boas opções de formalismos matemáticos para encontrar gargalos de desempenho em uma solução de integração de um sistema de eventos discretos.

A verificação do modelo de simulação proposto foi realizada por meio de técnicas formais existentes na literatura. Foram utilizadas as técnicas de Kleijnen [26] e de Sargent [39], na qual uma envolve o conhecimento de um especialista na área sobre o comportamento do sistema e a outra verifica o modelo de simulação em si, respectivamente.

Como trabalho futuro sugere-se verificar o comportamento do modelo de simulação com intervalos de tempo entre as prioridades maiores e cenários com cargas maiores. Com pouca correlação a análise tempo/acúmulo se tornou mais difícil (até incoerente). Analisar se o filtro retirou *tokens* com tempos chave. Também, analisar o trajeto dos *tokens*/prioridades e a sua ordem de chegada nas aplicações. Propor um novo modelo que procure resolver o problema do gargalo de desempenho. Fazer um cenário no qual uma cor represente uma mensagem, para analisar a correlação do seu par exato.

---

## Bibliografia

---

- [1] G. H. S. Arruda, S. Sawicki, R. Z. Frantz e F. Roos-Frantz. *Desenvolvimento de um modelo de simulação baseado em uma solução de integração real utilizando a ferramenta prism*. *Salão do Conhecimento*, 1 (1), 2015.
- [2] F. Bause e P. S. Kritzinger. *Stochastic petri nets: An introduction to the theory*. *ACM SIGMETRICS Performance Evaluation Review*, 26(2):2–3, 1998.
- [3] G. A. P. Borges. *Fluxo de dados em redes de petri coloridas e em grafos orientados a atores*. Tese Doutoral, Universidade de São Paulo, 2008.
- [4] P. N. D. Bukh e R. Jain. *The art of computer systems performance analysis, techniques for experimental design, measurement, simulation and modeling*, 1992.
- [5] R. S. Cargnin. *Modelagem e simulação de uma solução de integração para identificação de gargalos de desempenho baseado em formalismo matemático - uma abordagem orientada a redes de petri*. Dissertação de Mestrado, Universidade Regional do Noroeste do Estado do Rio Grande do Sul, 2016.
- [6] J. S. Carson. *Model verification and validation*. Em *Simulation Conference, 2002. Proceedings of the Winter*, volume 1, páginas 52–58. IEEE, 2002.
- [7] P. R. Cerioni. *Sistema de simulação de modelos de tomada de decisão para despacho de agvs em cenários de fabrica configuráveis*. *Anais do Conic-Semesp*, 1, 2013.
- [8] L. G. Chaves, R. Baroni e M. Ferreira. *Análise da integração de aplicações (eai) no contexto de portais corporativos de médias e grandes empresas brasileiras*. *SMSI II*, 2005.



- [9] L. Chwif e A. C. Medina. *Modelagem e simulação de eventos discretos*. Afonso C. Medina, 2006.
- [10] P. J. de Freitas Filho. *Introdução à modelagem e simulação de sistemas: com aplicações em arena*. Visual Books, 2001.
- [11] A. H. C. de Sales. *Um estudo sobre redes de petri estocásticas generalizadas*. 2002.
- [12] D. Dossot, J. D’Emic e V. Romero. *Mule in action*. Manning, 2014.
- [13] D. Fahland e C. Gierds. *Analyzing and completing middleware designs for enterprise integration using coloured petri nets*. Em *International Conference on Advanced Information Systems Engineering*, páginas 400–416. Springer, 2013.
- [14] M. Fisher, J. Partner, M. Bogoevici e I. Fuld. *Spring integration in action*. Manning Publications Co., 2012.
- [15] M. Fowler. *Domain-specific languages*. Pearson Education, 2010.
- [16] C. R. L. Francês. *Introdução às redes de petri*. Laboratório de Computação Aplicada, Universidade Federal do Pará, 2003.
- [17] R. Frantz, S. Sawicki, F. Roos-Frantz, R. Corchuelo, V. Basto-Fernandes e I. Hernández. *Desafios para a implantação de soluções de integração de aplicações empresariais em provedores de computação em nuvem*. 2014.
- [18] R. Z. Frantz, R. Corchuelo e F. Roos-Frantz. *On the design of a maintainable software development kit to implement integration solutions*. *Journal of Systems and Software*, 111:89–104, 2016.
- [19] R. Z. Frantz, A. M. Reina Quintero e R. Corchuelo. *A domain-specific language to design enterprise application integration solutions*. *International Journal of Cooperative Information Systems*, 20(02):143–176, 2011.
- [20] C. M. Grinstead e J. L. Snell. *Introduction to probability*. American Mathematical Soc., 2012.
- [21] G. Hohpe e B. Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.

- [22] M. M. Horn. *Modelagem e simulação de uma solução de integração para identificação de gargalos de desempenho baseado em formalismo matemático - uma abordagem orientada à cadeias de markov*. Dissertação de Mestrado, Universidade Regional do Noroeste do Estado do Rio Grande do Sul, 2016.
- [23] C. Ibsen e J. Anstey. *Camel in action*. Manning Publications Co., 2010.
- [24] K. Jensen e L. M. Kristensen. *Coloured petri nets: modelling and validation of concurrent systems*. Springer Science & Business Media, 2009.
- [25] W. D. Kelton e A. M. Law. *Simulation modeling and analysis*. McGraw Hill Boston, 1991.
- [26] J. P. Kleijnen. *Validation of models: statistical techniques and data availability*. Em *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future-Volume 1*, páginas 647–654. ACM, 1999.
- [27] J. Lied e M. A. Sellitto. *Aplicação da modelagem por redes de petri para avaliação ocupacional de trabalhadores*. *Revista Produção Online*, 9(3), 2009.
- [28] D. S. Linthicum. *Enterprise application integration*. Addison-Wesley Professional, 2000.
- [29] P. R. Maciel, R. D. Lins e P. R. Cunha. *Introdução às redes de petri e aplicações*. UNICAMP-Instituto de Computacao, 1996.
- [30] M. K. Molloy. *Performance analysis using stochastic petri nets*. *Computers, IEEE Transactions on*, 100(9):913–917, 1982.
- [31] Y. Narahari, K. Suryanarayanan e N. S. Reddy. *Discrete event simulation of distributed systems using stochastic petri nets*. Em *TENCON'89. Fourth IEEE Region 10 International Conference*, páginas 622–625. IEEE, 1989.
- [32] R. Palomino. *Uma abordagem para a modelagem, análise e controle de sistemas de produção utilizando redes de petri*. 120 f. Tese Doutoral, Dissertação. Engenharia de Produção. Universidade Federal de Santa Catarina, 1995.
- [33] J. L. Peterson. *Petri net theory and the modeling of systems*. 1981.

- [34] C. A. Petri. *Communication mit automaten*. 1962.
- [35] C. M. d. O. Rodrigues. *Mapeando estruturas lsc em redes de petri coloridas*, 2006.
- [36] F. Roos-Frantz, M. Binelo, R. Z. Frantz, S. Sawicki e V. Basto-Fernandes. *Using petri nets to enable the simulation of application integration solutions conceptual models*. 2015.
- [37] N. Sakurada e D. I. Miyake. *Aplicação de simuladores de eventos discretos no processo de modelagem de sistemas de operações de serviços*. *Gestão & Produção*, 16(1):25–43, 2009.
- [38] R. F. Sampaio, G. C. Barroso e R. P. S. Leão. *Método de implementação de sistema de diagnóstico de falta para subestações baseado em redes de petri*. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, 16(4):417–426, 2005.
- [39] R. G. Sargent. *Verification and validation of simulation models*. Em *Proceedings of the 37th conference on Winter simulation*, páginas 130–143. winter simulation conference, 2005.
- [40] S. Sawicki, R. Z. Frantz, V. M. B. Fernandes, F. Roos-Frantz, I. Yevseyeva e R. Corchuelo. *Characterising enterprise application integration solutions as discreteevent system*. *IGI Global*, 2015.
- [41] D. C. Schmidt. *Model-driven engineering*. *COMPUTER-IEEE COMPUTER SOCIETY*, 39(2):25, 2006.
- [42] R. E. Shannon. *Systems simulation: the art and science*, volume 1. Prentice-Hall Englewood Cliffs, NJ, 1975.
- [43] L. C. Silva. *Simulação de processos*. *Universidade Federal do Espírito Santo* <http://www.agais.com/simula.htm>, 2, 2008.
- [44] J. C. R. Valette e J. Cardoso. *Redes de petri*. Editora DA UFSC. Florianópolis, 1997.
- [45] A. K. Wiesner. *Modelagem e simulação de uma solução de integração para identificação de gargalos de desempenho baseadas em formalismo matemático: uma abordagem orientada à teoria das filas*. 2016.

This document was typeset on June 13, 2017 using class `RG-BOK`  $\alpha 2.14$  for `LATEX2 $\epsilon$` . As of the time of writing this document, this class is not publicly available. Only members of [The Distributed Group \(TDG\)](#) and the [Applied Computing Research Group \(ACR\)](#) are allowed to typeset their documents using this class.