

---

# MODELAGEM E SIMULAÇÃO DE UMA SOLUÇÃO DE INTEGRAÇÃO PARA IDENTIFICAÇÃO DE GARGA- LOS DE DESEMPENHO BASEADAS EM FORMALISMO MATEMÁTICO



UMA ABORDAGEM ORIENTADA A REDES DE PETRI

---

ROBERTO SAULO CARGNIN

UNIVERSIDADE REGIONAL DO NOROESTE DO  
ESTADO DO RIO GRANDE DO SUL

DISSERTAÇÃO DE MESTRADO

ORIENTADORA:

DR<sup>a</sup> FABRICIA CARNEIRO ROOS FRANTZ

COORIENTADOR:

DR. RAFAEL ZANCAN FRANTZ



**Applied**  
Computing  
Research Group

MARÇO, 2016

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:** O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil.

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

A Comissão Examinadora, abaixo assinada, \_\_\_\_\_ a dissertação intitulada “Modelagem e Simulação de uma Solução de Integração para Identificação de Gargalos de Desempenho Baseadas em Formalismo Matemático: Uma Abordagem Orientada a Redes de Petri”, elaborada por Roberto Saulo Cargin, como requisito parcial para a obtenção do título de Mestre em Modelagem Matemática.

---

Prof<sup>ª</sup>. Dr<sup>ª</sup>. Fabricia C. Roos Frantz  
(Orientadora/UNIJUÍ)

---

Prof. Dr. Rafael Zancan Frantz  
(Co-orientador/UNIJUÍ)

---

Prof. Dr. Felipe de Souza Marques  
(UFPEL)

---

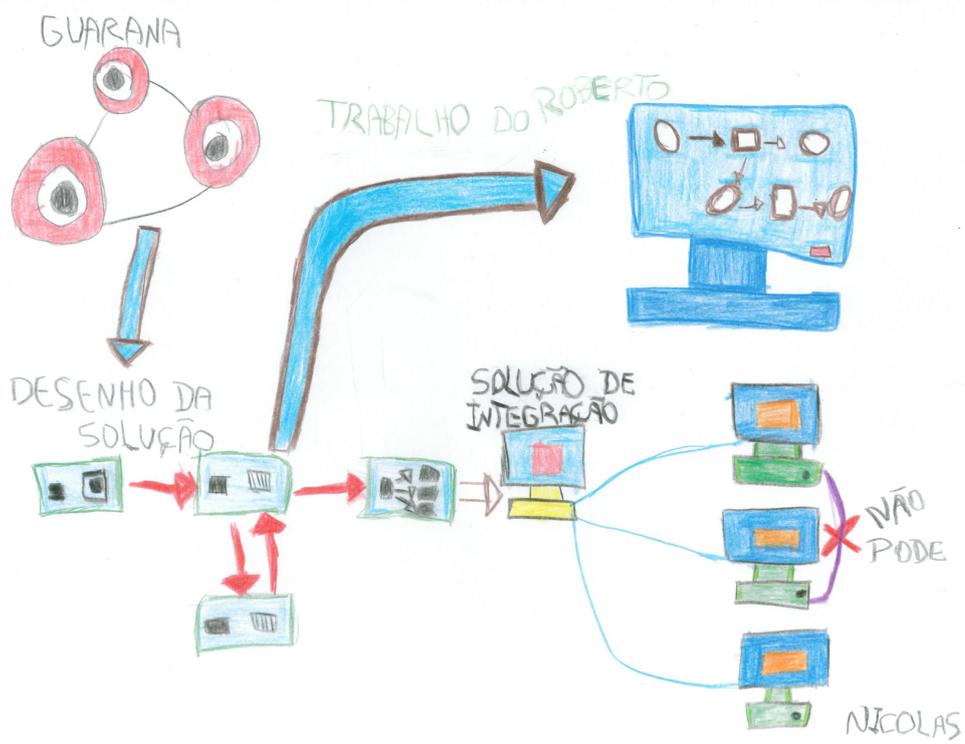
Prof. Dr. Paulo Sérgio Sausen  
(UNIJUÍ)

---

Prof. Dr. Sandro Sawicki  
(UNIJUÍ)

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





Resquisa desenvolvida por Nicolas, 6 anos de idade.



A minha mãe



---

# Conteúdo

---

<b>Agradecimentos</b> .....	<b>vii</b>
<b>Resumo</b> .....	<b>ix</b>
<b>Abstract</b> .....	<b>xi</b>
<b>1 Introdução</b> .....	<b>1</b>
1.1 Contexto da Pesquisa .....	1
1.2 Motivação .....	5
1.3 Objetivos .....	5
1.3.1 Objetivo Geral .....	5
1.3.2 Objetivos Específicos .....	6
1.4 Metodologia .....	6
1.5 Resumo das Contribuições .....	7
1.6 Estrutura dessa Dissertação .....	9
<b>2 Revisão da Literatura</b> .....	<b>11</b>
2.1 Integração de Aplicações Empresariais .....	11
2.1.1 Estilos de Integração .....	12
2.1.2 Padrões de Integração .....	13
2.1.3 Guaraná DSL .....	15
2.2 Simulação .....	21
2.2.1 Modelos .....	22
2.2.2 Classificação de Modelos .....	24
2.2.3 Modelos de Simulação .....	24
2.3 Redes de Petri .....	26
2.3.1 Notação Gráfica e Formal .....	26
2.3.2 Notação Matricial .....	28

2.3.3	Execução	30
2.3.4	Exemplo de aplicação	30
2.3.5	Redes de Petri Estocásticas	33
2.4	Validação e Verificação	34
2.4.1	Diferenciação entre Verificação e Validação	35
2.4.2	Metodologias de Verificação de Modelos de Simulação	35
2.5	Trabalhos Relacionados	38
2.6	Resumo do Capítulo	44
<b>3</b>	<b>Modelagem</b>	<b>47</b>
3.1	Equivalência entre Guaraná DSL e Redes de Petri	47
3.1.1	Representação das Tarefas em Rede de Petri	48
3.2	Estudo de Caso	49
3.2.1	Ecosistema de Software	49
3.2.2	Solução de Integração	49
3.3	Modelo de Simulação	52
3.3.1	Formulação Matemática	53
3.4	Resumo do Capítulo	58
<b>4</b>	<b>Experimento e Análise dos Resultados</b>	<b>59</b>
4.1	Experimento	59
4.1.1	Variáveis Observadas	60
4.1.2	Ferramenta de Simulação	61
4.1.3	Resultados Experimentais	64
4.2	Análise dos Resultados	65
4.3	Verificação do Modelo de Simulação	67
4.4	Resumo do Capítulo	71
<b>5</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>73</b>
5.1	Trabalhos Futuros	74
	<b>Bibliografia</b>	<b>75</b>

---

# Índice de figuras

---

1.1	Solução de Integração .....	4
2.1	Exemplo Modelo Conceitual de Solução de Integração .....	20
2.2	Possibilidade de Estudo de Sistemas .....	23
2.3	Classificação dos Modelos de Simulação .....	25
2.4	Elementos Básicos Redes de Petri, adaptada de Barros [5] .....	27
2.5	Sistema V .....	31
2.6	Estados do Sistema V .....	33
2.7	Diagrama do Processos V&V .....	36
3.1	Equivalência entre troca de estados .....	48
3.2	Modelo Conceitual de Solução de Integração Proposto por Frantz [22] .....	51
3.3	Modelo de Simulação .....	53
4.1	Interface Gráfica PIPE2 .....	62
4.2	Aba de Sensibilização das Transições Estocásticas do PIPE2 .....	63
4.3	Gráfico da Formação de Filas nos Slots .....	66
4.4	Diagrama de Aplicação da Técnica Sargent [37] .....	69



---

## *Índice de tabelas*

---

2.1	Notação Gráfica da Linguagem Específica de Domínio do Guaraná .	16
2.2	Grupos de tarefas do Guaraná .....	17
2.3	Lei de Formação Notação Matricial .....	29
3.1	Comparativo dos Elementos do Guaraná e Redes de Petri .....	48
3.2	Tarefas e seus grafos equivalentes em Redes de Petri .....	50
3.3	Elementos do Caso de Estudo e Seus Grafos Equivalentes em Rede de Petri .....	54
4.1	Variáveis Observadas .....	61
4.2	Tabela de Resultados Experimentais .....	65
4.3	Aplicação da Técnica de Verificação proposta por Sargent [37] .....	70
4.4	Aplicação da Técnica Validade de Eventos .....	71



---

# *Agradecimentos*

---

Agradeço primeiramente a Deus por tudo que conquistei.

À minha família pelo apoio oferecido.

À minha noiva Valéria por acreditar em mim nos momentos difíceis, estar sempre a meu lado e me incentivar a seguir em frente.

Aos Professores do grupo GCA Prof. Rafael Zancan Frantz , Prof. Sandro Sawicki e a Prof<sup>a</sup>. Fabricia pela paciência e maestria na transmissão do conhecimentos, que sem dúvidas servirão de inspiração na minha formação docente.

Aos demais professores do DCEEng - Departamento de Ciências Exatas e Engenharias.

Aos colegas do grupo GCA.

À Geni pela atenção, ajuda e prestatividade.

Ao CNPq pela bolsa de estudos ofertada.

Aos colegas do mestrado em modelagem matemática por toda a ajuda ao longo do curso.

A Minha gratidão à todos que de alguma forma contribuíram para eu concluir mais esta etapa.



---

## Resumo

---



Atualmente, com o apoio das aplicações de software, as empresas asseguram maior eficiência, agilidade e melhor tomada de decisão nos seus processos de negócio. Neste cenário altamente competitivo, frequentemente surgem novas necessidades de negócio, o que leva as empresas a evoluir ou criar novos processos. Adicionar novas funcionalidades ao ecossistema de software da empresa, a partir da reutilização das aplicações existentes, não é uma tarefa trivial e exige metodologias, técnicas e ferramentas adequadas para construir soluções de integração. O ecossistema de software das empresas é heterogêneo, pois geralmente suas aplicações são adquiridas de empresas terceirizadas e são desenvolvidas utilizando diferentes plataformas de desenvolvimento, o que dificulta a integração entre as aplicações, pois geralmente elas são desenvolvidas sem a preocupação da integração. A área de Integração de Aplicações Empresariais trata da integração das aplicações existentes no ecossistema de software das empresas por meio de uma solução de integração. Uma solução de integração pode ser representada por um modelo conceitual, que representa o processo de integração entre aplicações em alto nível de abstração. O Guaraná DSL é uma das tecnologias que possibilita projetar modelos conceituais de soluções de integração, utilizando uma sintaxe concreta gráfica e intuitiva. No entanto, a integração de aplicações não é uma tarefa trivial e o desenvolvimento da solução envolve além de custos, riscos como *bugs* e gargalos de performance que geralmente são observados após a implementação. Esta dissertação propõe analisar o comportamento e identificar gargalos de performance de uma solução de integração ainda na fase de projeto. Propõe-se o desenvolvimento de um modelo matemático de simulação equivalente ao modelo conceitual da solução, utilizando como base as Redes de Petri. O modelo de simulação foi verificado por meio de técnicas de verificação validadas na literatura. A principal contribuição deste trabalho foi a constatação de que é possível representar modelos conceituais de soluções de integração de aplicações por meio das Redes de Petri Estocásticas. O experimento realizado possibilitou identificar pontos de formação de filas, possíveis gargalos de desempenho e identificar comportamentos da solução de integração ainda na fase de projeto.



---

## *Abstract*

---

**N**owadays, with the support of software applications, the enterprises assure greater efficiency, agility and make better decisions regarding its business processes. In this highly competitive scenario, new business necessities frequently come up, which forces the companies to evolve or even to create new processes. Adding a new functionality to a company's software ecosystem, by reusing existent applications, is a difficult task that requires the correct methodologies, techniques and tools to build integration solutions. The companies's software ecosystem is heterogeneous, since its applications are acquired from third parties and are developed in different platforms. This heterogeneity hampers the integration between the applications, since they are generally developed without having integration in mind. The Enterprise Application Integration area deals with the integration of existent applications in the companies' software ecosystem by means of an integration solution. An integration solution may be represented by a conceptual model, which represents the process of integration between applications at a high level of abstraction. Guaranã; DSL is one of the technologies that allows the user to project conceptual integration solution models, using concrete and intuitive graphic syntax. However, the integration solution of applications is not a trivial task and the development of a solution, aside from being expensive, involve risks such as Bugs and Performance Bottlenecks, generally observed after the implementation. This research's objective is to analyze the behavior and identify Performance Bottlenecks in an integration solution at project stage, making use of the conceptual integration solution model. This research also has the objective of developing a mathematical simulation model equivalent to the conceptual model, using the Petri Nets as a base. The simulation model was verified by verification techniques validated by the available literature. The main contribution of this work was the realization that it is possible to represent conceptual models of application integration solutions through Stochastic Petri Nets. The experiment made it possible to identify points of queuing, possible performance bottlenecks and identify behaviors of integration solution still in the design phase.



---

# Capítulo 1

## Introdução

---

Nossa maior fraqueza está em desistir.  
O caminho mais certo de vencer é tentar mais uma vez.

Thomas Edison



A introdução trata da problematização da pesquisa desenvolvida e está dividida em 6 seções. A Seção §1.1 trata inicialmente de temas relacionados às aplicações empresariais, na sequência é abordada integração de aplicações, uso de soluções de integração para propor a integração de aplicações. A Seção §1.2 trata da motivação para o desenvolvimento da referida pesquisa, aborda inicialmente o problema de pesquisa e posteriormente a hipótese de solução do referido problema. A Seção §1.3 trata dos objetivos geral e específicos da pesquisa. A Seção §1.4 apresenta a metodologia de desenvolvimento do trabalho, baseada no *framework* de referencia fornecido pelo *Unified Process* (UP), está dividida na fases de Início, Elaboração, Construção e Transição. A Seção §1.5 apresenta, de forma resumida, as contribuições desta pesquisa para a modelagem de soluções de integração e por fim os trabalhos publicados referente ao tema pesquisado. A Seção §1.6 encerra esse capítulo apresentando a estrutura organizacional da dissertação.

### 1.1 Contexto da Pesquisa

Os sistemas basicamente são definidos como um conjunto de componentes que atuam em uma determinada plataforma, trabalhando em conjunto

para realizar determinadas funções em busca de objetivos comuns. Exemplos de sistemas são os carros, computadores, seres humanos, etc [15]. Um sistema de informação é um sistema que recebe, manipula e gerencia dados, com o objetivo de transformá-los em informação útil a seus usuários [40]. Os sistemas de informação proporcionam às empresas um fácil acesso às informações, suporte à tomada de decisões e um diferencial em relação aos seus concorrentes. Atualmente, as organizações dispõem de uma estrutura computacional, na maioria heterogênea, composta de pacotes de softwares comprados de empresas especializadas e de aplicações desenvolvidos na própria empresa, trabalhando de forma sistêmica.

Denomina-se ecossistema de software a estrutura de sistema informático em funcionamento de uma empresa. Esta estrutura é composta por um conjunto de aplicações, denominadas aplicações empresariais, as quais são softwares desenvolvidos para funcionalidades específicas. Estas aplicações podem ser consideradas interfaces de acesso e manipulação da informação armazenada, e têm a finalidade de dar suporte aos processos de negócios, auxiliar na organização e na solução de problemas da empresa. Podemos destacar alguns exemplos comuns de aplicações empresariais como os sistemas de faturamento automático, folha de pagamentos, marketing por E-mail, autoatendimento com suporte ao cliente, gestão de relacionamento com o cliente, Gestão de RH e muitos outros.

As aplicações empresariais são sistemas dinâmicos, o que significa que estão em constante mudança e atualização. No cenário altamente competitivo em que se encontram as empresas, frequentemente surgem novas necessidades de negócio, levando as corporações a buscar novas formas de comercializar seus produtos. Os novos processos de negócio requerem apoio do ecossistema de software em funcionamento, restando aos administradores de sistemas acompanhar a modernização ou perder espaço para a concorrência.

Adicionar uma nova funcionalidade ao ecossistema da empresa não é uma tarefa trivial. A maioria das plataformas e aplicações em funcionamento nas empresas são antigas, as quais não possuem documentação ou muitas vezes não oferecem acesso ao código. Portanto, migrar de um sistema antigo para uma plataforma mais moderna se torna inaplicável, na maioria dos casos, devido ao alto custo associado. Além disso, alterar um sistema consolidado, que até então supre as demandas da empresa, pode trazer efeitos colaterais. Qualquer inclusão ou alteração de funcionalidade no ecossistema atual deve garantir que não haja interferência entre novas e antigas aplicações, evitar conflitos de plataformas e também evitar custos muito elevados. Dispor de novas funcionalidades é tão importante quanto manter as

antigas em perfeito funcionamento. Portanto, a solução ideal é a inclusão de novas funcionalidades ao ecossistema de software reutilizando as aplicações existentes, porém, sem modificá-las.

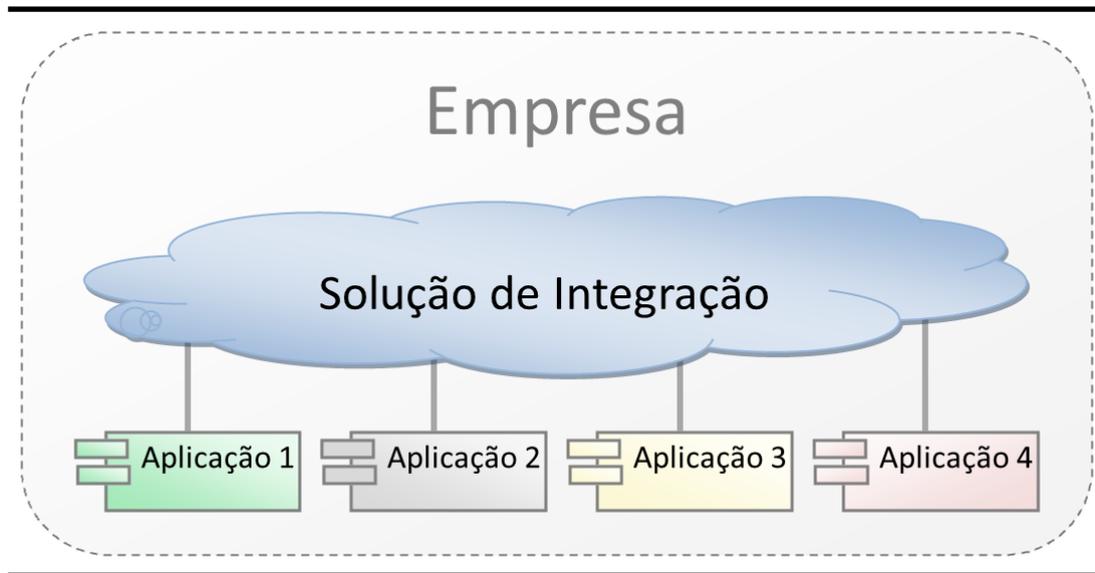
O ecossistema de software das empresas geralmente é heterogêneo, ou seja, é formado por um conjunto de aplicações diferentes, que trabalham com diferentes modelos de dados e são construídas utilizando diferentes plataformas. Dados de uma determinada aplicação podem ser usados por outras aplicações, no entanto a maioria das aplicações que compõem um ecossistema não foram projetadas para interagir umas com as outras. Uma forma de promover a comunicação entre essas aplicações é utilizar ferramentas e tecnologias projetadas para a integração das aplicações. A integração de aplicações tem como principal vantagem a utilização da base tecnológica dos sistemas já em funcionamento, sem a necessidade de fazer investimento de alto custo em novas plataformas.

Enterprise Application Integration (EAI) é uma combinação de tecnologias com a função de organizar os processos de negócio, propondo uma solução de integração para integrar aplicações sem realizar alterações nas suas estruturas de dados [29]. Na Figura §1.1 é ilustrado uma solução de integração que conecta diversas aplicações. Do ponto de vista dos negócios, a EAI representa um sistema de informação unificado, com acesso fácil a informações valiosas e decisivas, possibilitando uma varredura constante de todos os sistemas integrados em busca da informação requerida. Um olhar técnico sobre a estrutura das EAI e identificamos diferentes aplicações integradas em torno de uma plataforma de integração que promove o livre intercâmbio de informações entre as aplicações e delas com os usuários do sistema [29].

O estudo e desenvolvimento de técnicas para integrações de aplicações são extensos. A mesma heterogeneidade presente no desenvolvimento dos ecossistemas de *softwares* se repetiu nas soluções de integrações. A experiência adquirida com a integração de aplicações consolidou técnicas e metodologias [25].

A integração de aplicações pode ser realizada de várias maneiras e utilizar diversos recursos. Não é possível determinar o melhor a ser considerado para uma organização, pois cada um deles possui seus prós e contras. Assim é preciso analisar os estilos e determinar qual deles mais acrescentará benefícios à aplicação. Dentre os estilos mais utilizados destaca-se arquivo, banco de dados, chamado de procedimento remoto e sistema de mensagens [25].

Integrar aplicações significa estabelecer um mecanismo de comunicação entre elas. A forma como é realizada a comunicação, define o estilo de integração. De acordo com Hohpe e Woolf [25], o estilo de troca de mensagens é o



**Figura 1.1:** *Solução de Integração*

melhor estilo para integração por ser simples, de fácil adequação a estruturas modernas, possuir baixo acoplamento e permitir comunicação assíncrona.

Uma solução de integração é uma aplicação que funciona de forma sistêmica. Soluções de integração se tornam mais complexas à medida que se necessita integrar mais aplicações. O modelo conceitual de uma solução de integração possibilita que desenvolvedores e usuários possam compreender sua estrutura. Nele é feita uma representação abstrata dos componentes da solução e esboçada sua arquitetura. Os modelos devem utilizar linguagens bastante simples para facilitar sua compreensão, representar e discriminar as partes integrantes da solução.

Há várias tecnologias disponíveis, baseadas em mensagem, que possibilitam projetar modelos conceituais de soluções de integração. Entre elas podemos citar Camel[26], Mule, Spring e o Guaraná. Todas elas seguem os padrões de integração documentados por Hohpe e Woolf [25] e proporcionam uma linguagem de domínio específico (DSL) para a modelagem conceitual de soluções de integração [19].

A tecnologia Guaraná foi criada por pesquisadores que atuam no grupo de pesquisa em computação aplicada (GCA) [2], no qual a presente pesquisa foi desenvolvida. Em função do fácil acesso à documentação desta tecnologia, optou-se por utilizá-la como objeto de estudo desta dissertação, contribuindo para sua evolução e aperfeiçoamento.

A ferramenta de desenvolvimento de modelos conceituais Guaraná não apresenta mecanismo de simulação para a análise de soluções de integração. As Redes de Petri são um formalismo matemático muito usado na modelagem e simulação de sistemas e possibilitam representar o funcionamento de um sistema e proporcionar o entendimento do funcionamento do mesmo, bem como possíveis problemas que venham a surgir. As Redes de Petri usam uma linguagem simples e de poucos elementos, possui representação matemática formal e podem ser interpretadas por ferramentas de simulação. As Redes de Petri Estocásticas são uma extensão das Redes de Petri e apresentam particularidades que podem permitir modelar características específicas das soluções de integração.

## 1.2 Motivação

Uma solução de integração pode apresentar gargalos de desempenho, o que pode comprometer seu funcionamento em situações de grande demanda. A análise do comportamento e a identificação de gargalos de desempenho de soluções de integração geralmente envolve a implementação das soluções, para posterior execução e teste frente a cenários críticos. Por demandar que as soluções sejam implementadas, tal abordagem traz consigo custos (tempo, recursos) que costumam ser elevados.

Dada a importância das aplicações para os processos de negócio e os problemas que podem ocorrer no processo de integração, a presente pesquisa motiva-se pela possibilidade de análise do comportamento e identificação de gargalos de desempenho de soluções de integração ainda na fase de projeto. Propõe-se o desenvolvimento e avaliação de um modelo de simulação construído a partir de um modelo conceitual de solução de integração. Dessa maneira seria possível conhecer o comportamento de uma solução e identificar gargalos antes de sua implementação, contribuindo para a redução do consumo de recursos nas fases de implementação e testes.

## 1.3 Objetivos

Na sequência apresenta-se o objetivo geral e os objetivos específicos da pesquisa desenvolvida.

### 1.3.1 Objetivo Geral

O objetivo Geral da pesquisa é analisar o comportamento de uma solução de integração e identificar possíveis gargalos de desempenho nesta solu-

ção, por meio da simulação do modelo conceitual utilizando o formalismo de Redes de Petri.

### 1.3.2 Objetivos Específicos

- Estudar a equivalência entre os elementos do Guaraná DSL e das Redes de Petri;
- Propor um modelo de simulação utilizando Redes de Petri Estocásticas para o modelo conceitual do caso de estudo;
- Pesquisar e identificar ferramentas de software livre para simulação de Redes de Petri Estocásticas;
- Implementar o modelo de simulação utilizando uma ferramenta de software e realizar experimentos;
- Analisar os resultados experimentais para identificar possíveis gargalos de desempenho e avaliar o comportamento da solução sob determinado cenário de funcionamento;
- Verificar o modelo de simulação proposto utilizando técnicas de verificação propostas na literatura.

## 1.4 Metodologia

A metodologia de trabalho adotada para o desenvolvimento desta pesquisa baseia-se no *framework* de referência fornecido pelo *Unified Process* (UP). Cabe ressaltar que esta escolha é motivada pela experiência que o grupo de pesquisadores do grupo de pesquisa em computação aplicada (GCA) possui em utilizá-lo em pesquisas e transferências de tecnologia, e porque o seu ciclo de vida iterativo e incremental é apropriado ao desenvolvimento de projetos que visam alcançar um grande dinamismo, incorporando, a todo o momento, o *feedback* de outros grupos de trabalho e/ou projetos relacionados, mas mantendo sob controle os riscos que podem ocorrer no dia-a-dia. Considerando-se o *framework* UP, o trabalho de pesquisa foi dividido nas seguintes fases:

**INÍCIO:** A etapa inicial ocupa-se de compreender o que será feito, identificar pontos do projeto mais relevantes, compreender os principais riscos que podem afetar o desenvolvimento do projeto e decidir sobre o processo a ser seguido. Esta fase foi concluída antes da submissão do projeto de dissertação.

**ELABORAÇÃO:** Parte dos objetivos desta fase consiste em identificar e descrever os principais blocos de trabalho e suas atividades. Este trabalho foi desenvolvido através das reuniões do tipo “*modelstorming*” e os resultados são apresentados na sequencia. Seguindo um plano de atividades e cronograma, ao decorrer da pesquisa será necessário refinar as atividades atuais, estabelecer um plano de ação concreto para os riscos que não tenham sido considerados e priorizar as atividades.

**CONSTRUÇÃO:** Esta fase irá dividir as atividades dos blocos de trabalho em pequenas etapas que sejam abordáveis de forma iterativa, buscando ciclos rápidos de análise detalhada da bibliografia, projeto da solução, prototipação e validação, a fim de que seja possível obter *feedbacks* úteis. O número de etapas e os objetivos específicos em cada uma, bem como marcos intermediários, são acordados pelos pesquisadores envolvidos em cada atividade. A forma de abordar, com êxito, esta fase é mediante a uma reunião inicial onde os professores orientadores apresentam as instruções referente a fase seguinte ao cronograma seguido e são planejadas as tarefas, reuniões periódicas para apresentação de resultados e seminários de pesquisa nos quais os mestrandos apresentam os progressos alcançados nas fases anteriores, sejam eles resultados ou conclusões sobre a bibliografia estudada. Durante esta fase, também é conveniente a participação em eventos científicos e tecnológicos relacionados com os temas de pesquisa do projeto, com o objetivo de apresentar os resultados e obter *feedback*.

**TRANSIÇÃO:** O principal objetivo desta fase é a transferência de resultados obtidos para seus usuários finais. Neste projeto estes resultados serão incorporados às demais tecnologias já desenvolvidas pelos pesquisadores do grupo GCA envolvidos no projeto, se buscará aplicá-las a projetos reais em colaboração com empresas locais no nosso estado e empresa parceira em âmbito internacional com a qual o grupo já tem histórico de colaboração. Também é objetivo desta fase obter informações que possam ser úteis para melhorar nossos métodos de trabalho e abordar melhor futuros projetos de pesquisa.

## 1.5 Resumo das Contribuições

O estudo, desenvolvimento e modelagem de soluções de integração desenvolvidas por meio da Tecnologia Guaraná faz parte de um projeto maior do grupo GCA [2], e esta pesquisa contempla a Modelagem Matemática e simulação de um modelo conceitual com intuito de avaliar seu comportamento. Desta forma, nesta sessão aborda-se de forma resumida as

contribuições que esta pesquisa trouxe para o estudo do comportamento das soluções de integração e na sequência relata-se trabalhos publicados com temas referentes à pesquisa desenvolvida.

As principais contribuições desta são listadas a seguir:

- Aplicação das Redes de Petri como técnica para modelagem e representação formal dos modelos conceituais desenvolvidos na Tecnologia Guaraná
- Desenvolvimento de um modelo de simulação equivalente ao modelo conceitual elaboradas na tecnologia Guaraná e implementação em um software de simulação para análise de comportamento e identificação de possíveis gargalos de desempenho
- Verificação do modelo de simulação proposto por meio de técnicas formais existentes na literatura.

As publicações tratam de temas bem específicos, foram desenvolvidos de acordo com o decorrer da pesquisa e contemplam os resultados encontrados e que contribuíram para os resultados finais.

- Revisão da Literatura de Simulação de Sistemas de Eventos Discretos com Foco na Integração de Aplicações: O presente trabalho relata o estudo bibliográfico sobre a aplicação das redes de Petri para elaboração de modelos de simulação de sistemas de eventos discretos. Diante da importância da integração de aplicações para as corporações, estudou-se a possibilidade de aplicar as Redes de Petri para construção de modelos de simulação de soluções de integração, baseados em seus modelos conceituais com a perspectiva de analisar seu funcionamento sem a necessidade de construí-las [7].
- Simulação de uma solução de integração com Redes de Petri estocásticas para o problema da central telefônica na UNIJUI: O presente trabalho relata a modelagem de modelos conceituais desenvolvido na tecnologia Guaraná modelados por meio de Redes de Petri e a elaboração de um modelo de simulação com base em um caso de estudo. Foi identificada a analogia de execução das duas ferramentas e de seus elementos principais com sistemas de eventos discretos. Por fim é apresentado uma proposta de um método probabilístico de estimação de métricas características à solução de integração modelada e sua contribuição para realização de simulações para que por meio desta se possa estimar seu comportamento.

## 1.6 Estrutura dessa Dissertação

Essa dissertação está organizada da seguinte maneira:

**Capítulo I.** Compreende essa introdução.

**Capítulo II: Revisão da Literatura.** Proporciona para o leitor uma revisão da literatura técnica e científica relacionadas à pesquisa desenvolvida nessa dissertação. Na Seção §2.1 apresenta-se o referencial teórico referente a integração de aplicações empresariais, na Seção §2.2 aborda-se o referencial teórico sobre simulação, na Seção §2.3 aborda-se o referencial teórico referente as Redes de Petri, na Seção §2.4 aborda-se as definições de Validação e verificação e técnicas para verificar modelos e na Seção §2.5 introduz-se os trabalhos relacionados identificados ao longo dessa pesquisa.

**Capítulo III: Modelagem** Apresenta o trabalho de modelagem do modelo conceitual de solução de integração. Na Seção §3.1 descreve a equivalência entre componentes do Guaraná e das Redes de Petri, na Seção §3.2 é apresentado o caso de estudo e na Seção §3.3 apresenta-se a descrição do modelo de simulação proposto.

**Capítulo IV: Simulação e Análise dos Resultados** Apresenta experimentos de simulação realizados no modelo formal proposto. Na Seção §4.1 apresenta-se o cenário experimental, variáveis observadas, ferramenta de simulação e resultados experimentais, na Seção §4.2 realiza-se uma análise dos resultados e na Seção §4.3 realiza-se a verificação do modelo de simulação proposto.

**Capítulo V: Conclusões e Trabalhos Futuros** Trata das conclusões, contribuições, algumas limitações encontradas e trabalhos futuros para continuação da pesquisa desenvolvida.



---

# Capítulo 2

## Revisão da Literatura

---

*A persistência é o caminho do êxito.*

*Charles Chaplin*

**O** referencial teórico aborda os temas que dão suporte a pesquisa e está dividido em 6 seções. A Seção §2.1 fala da integração de aplicações, seus estilos e padrões e por fim descreve uma ferramenta para desenvolvimento de modelos conceituais de solução de integração. A Seção §2.2 abordada a simulação e sua importância para desenvolvimento de sistemas, os modelos como representação abstrata de sistemas e sua classificação e por fim fala sobre simulação computacional. A Seção §2.3 aborda as Redes de Petri como formalismo matemático para representação de sistemas, apresenta sua notação gráfica e matricial, discussão do seu sistema de execução, um exemplo de aplicação e por fim fala-se sobre as Redes de Petri Estocásticas. A Seção §2.4 trata da importância da validação e verificação de modelos, debate-se a diferença entre validação e verificação e por fim são abordadas metodologias de verificação de modelos. A Seção §2.5 apresenta trabalhos que abordam simulação de sistemas de eventos discretos utilizando o formalismo das Redes de Petri. A Seção §2.6 encerra esse capítulo apresentando um breve resumo dos assuntos abordados.

### 2.1 Integração de Aplicações Empresariais

Integrar aplicações empresariais tem como base a informação armazenada nos aplicativos que compõem o ecossistema de software da empresa, informação esta que será a base para novos processos de negócio implementados.

Para integrar novas e antigas aplicações é necessário promover a comunicação entre elas sem interferir no seu funcionamento individual. A integração de aplicações pode ser implementada seguindo diferentes estilos de integração, como trata a Seção §2.1.1. Os estilos são diferenciados de acordo com a forma que promovem a comunicação entre as aplicações. A integração de aplicações pode ser implementada seguindo diferentes estilos de integração, como trata a seção §2.1.2. Na sequência a Seção §2.1.3 aborda a tecnologia Tecnologia Guaraná que é uma ferramenta para desenvolver modelos conceituais de solução de integração baseado em mensagem.

### 2.1.1 Estilos de Integração

A integração de aplicações pode ser realizada de varias maneiras e utilizar diversos recursos. Não é possível determinar o melhor a ser considerado para uma organização, pois cada um deles possui seus prós e contras. Assim é preciso analisar os estilos e determinar qual deles acrescentará mais benefícios à aplicação. Dentre os estilos mais utilizados destacam-se arquivo, banco de dados, chamada de procedimento remoto e sistema de mensagens [25].

- Arquivo: A transferência de arquivos ocorre quando uma das aplicações escreve um arquivo e o repassa para ser lido por outra aplicação. Esse arquivo geralmente é escrito num formato que facilita o intercâmbio de informações. Caso as aplicações usem formatos de arquivos diferentes são usados integradores para converter os arquivos para os respectivos formatos. Esse estilo de integração consegue compartilhar apenas dados, e não funcionalidades. A troca de arquivos é o recurso mais usado na integração de aplicações pela versatilidade de implantação em diferentes plataformas e linguagens. Este estilo requer uma estrutura simples e gera um baixo acoplamento entre aplicações. A desvantagem da integração por arquivos é a dificuldade de adequação a ambientes de grande tráfego, podendo ocorrer erros na criação e armazenagem dos arquivos.
- Banco de Dados: Este estilo prevê o compartilhamento de banco de dados e ocorre quando mais de uma aplicação acessa a mesma base de dados. O compartilhamento do banco de dados permite que os dados estejam atualizados de forma oportuna para todos os usuários, o que reduz os erros e aumenta a confiabilidade do sistema integrado e extingue problemas de sincronização. Este estilo de integração não necessita de comunicação direta entre as aplicações, já que cada uma delas

precisa se comunicar unicamente com o banco de dados. A desvantagem desse estilo de integração é aplicações que usam diferentes formatos de dados. Outro problema que pode ocorrer é a diminuição de desempenho da solução de integração devido ao acesso concorrente ao banco de dados, uma vez que diferentes aplicações podem ter necessidade de acessar e modificar dados ao mesmo tempo.

- Chamada de Procedimento Remoto: A Chamada de Procedimento Remoto propõe integrar serviços. Ocorre quando uma das aplicações disponibiliza algumas de suas funcionalidades por meio de uma interface conhecida como *skeleton*. Essa interface pode ser acessada por outras aplicações por meio da interface *stub*. A sincronização e alteração dos dados são realizadas conforme realização dos processos. Integrações baseadas em Chamada de Procedimento Remoto aumentam o acoplamento e a dependência entre as aplicações. A mudança do mecanismo de negócio de uma aplicação integrada por Chamada de Procedimento Remoto acarreta na alteração de todas as demais..
- Sistema de Mensagens: No sistema de mensagens o fluxo de mensagens é gerenciado por um *middleware*, ou seja, um terceiro componente independente, que trabalha como elo entre as aplicações. Com isso as aplicações podem trocar informações com diferentes aplicações, e o *middleware* é responsável pela adequação do formato das mensagens, possibilitando a comunicação entre aplicações integradas bem como com aplicações integradas pelos demais estilos de integração . A principal tarefa de um sistema de mensagem é manter uma conexão segura para mover mensagens da aplicação remetente para a receptora de uma maneira confiável. O sistema de mensagens garante a entrega da mensagem, repetindo a operação até que a mesma seja entregue. Integrações baseadas em mensagens são ideais para ambientes que exigem flexibilidade e grande fluxo de informações. Esse estilo de integração propõe integrar as aplicações sem gerar dependência entre elas, mantendo a fluidez e a coesão de cada etapa do processo.

### 2.1.2 Padrões de Integração

Integrar aplicações requer troca de informações, há a necessidade de transportar dados de uma aplicação para a seguinte. Os dados podem ser um registro de endereço que precisa ser replicado, uma chamada para um serviço remoto ou um trecho de HTML. Independentemente da carga útil, este pedaço de dados precisa ser entendido por ambas as aplicações integradas. A

necessidade dos dados serem transportados, adiciona um canal de transporte, uma via virtual que conecta aplicação emissora a receptora. Dados reunidos em um arquivo denomina-se Mensagem. Hohpe e Woolf [25] estabelece a troca de mensagem como o melhor estilo para resolver muitos problemas de integração, por ser simples, se adequar a estruturas modernas e fácil entendimento, baixo acoplamento por comunicar de forma assíncrona, o que também torna a comunicação mais confiável, já que as duas aplicações não têm de estar em execução ao mesmo tempo para trocar mensagens.

Integração de aplicações não é uma tarefa simples e, normalmente, baseia-se na experiência dos engenheiros de *software* ou dos desenvolvedores. A reutilização de técnicas parecidas para solucionar problemas semelhantes levou a formar padrões para EAI, com a função de orientar na busca da melhor solução para problemas recorrentes ao integrar aplicações empresariais. Hohpe e Woolf [25] documentaram 65 padrões de integração de aplicações baseados em mensagens. Estes padrões foram divididos nas seguintes categorias: construção de mensagem, canais de mensagens, roteamento de mensagem, transformação de mensagens, ponto final de mensagens e gerenciamento do sistema. A continuação descreve-se cada uma das categorias.

- **Construção de Mensagem:** Mensagens são recipientes de dados que fluem dentro de uma solução de integração. A estrutura de uma mensagem consiste de duas partes: cabeçalho e corpo. O cabeçalho detém as informações que identificam a mensagem e a quem os dados contidos nela pertencem. O corpo é a parte da mensagem onde sofre alterações no decorrer da solução de integração.
- **Canais de Mensagens:** Os canais são endereços lógicos necessários para armazenar mensagens que ainda não foram processadas ou já foram processadas e aguardam um novo processamento. Um canal pode ser utilizado por uma solução de integração única ou podem ser partilhados por duas ou mais soluções. Os canais são desenvolvidos para leitura e escrita de mensagens assincronamente.
- **Roteamento de Mensagem:** O roteamento de mensagens compreende-se em um mecanismo responsável pelo direcionamento das mensagens dentro da solução de integração, através da inspeção do conteúdo do corpo da mensagem e se necessário também do cabeçalho. O diferencial desse mecanismo é que ele não altera o conteúdo da mensagem, apenas toma conhecimento para fim de filtrar ou direcionar.

- Transformação de mensagens: Por haver diferença dos modelos de dados entre aplicações é necessário transformar o conteúdo de mensagens de formato para outro. Nem sempre aplicações integradas utilizam o mesmo formato de dados.
- *Endpoint* de mensagens: São implementados sem alterar a estrutura da aplicação, e tem a função de enviar e receber mensagens. Os arquivos das aplicações integradas são preservados e a implementação do *Endpoint* é externa a aplicação. Eles podem incluir interfaces que permitem competir pela leitura de dados de um aplicativo, ser seletivo na leitura dos dados ou para fornecer suporte transacional entre uma solução de integração e as aplicações integradas.
- Gerenciamento do sistema: Possibilitam a monitorização discriminada dos componentes da solução de integração, como inspecionar uma mensagem sem afetar seu processamento regular, rastrear mensagens específicas e assim por diante. Gerenciar é uma tarefa desafiadora, por que o processamento de mensagem vai de milhares a casa de milhões, são distribuídas em vários aplicativos que sofrem alterações pelas mensagens. Para dificultar o gerenciamento, a comunicação se realizada de forma assíncrona, dificultando a identificação de gargalos de desempenho, que pode estar dentro da solução de integração ou nas aplicações integradas.

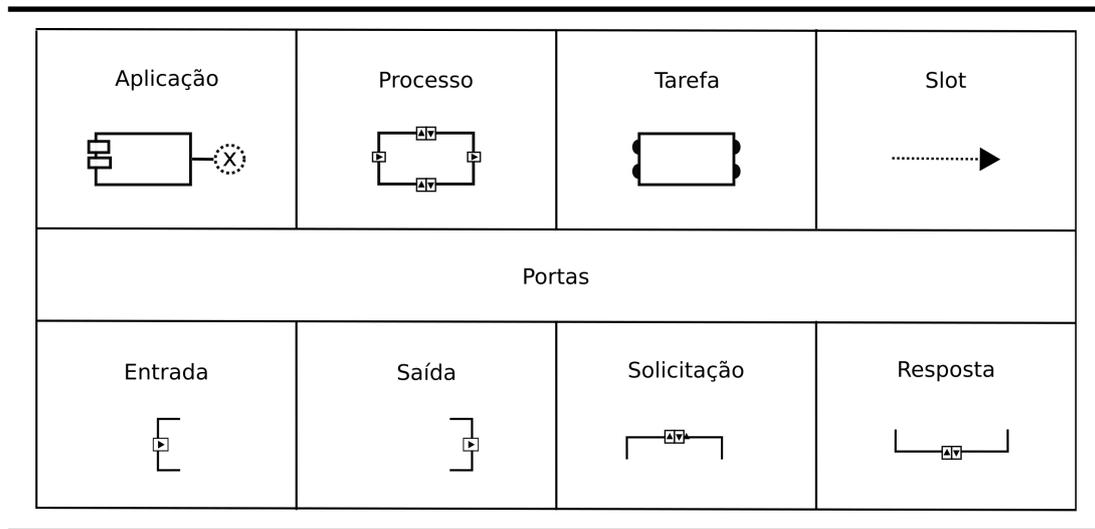
### 2.1.3 Guaraná DSL

O Guaraná DSL está entre as várias tecnologias disponíveis que possibilitam desenvolver modelos conceituais de soluções de integração de aplicações empresariais mantendo um nível elevado de abstração. Os modelos conceituais projetados com a tecnologia Guaraná são independentes de plataforma e essa autonomia possibilita que as soluções sejam implementadas em diferentes tecnologias [21].

Esta seção irá abordar a notação gráfica da linguagem Guaraná, um exemplo de solução de integração desenvolvido com esta linguagem, enfatizando a comunicação entre os elementos e suas funções, e, por último, apresentar o sistema de execução da tecnologia Guaraná.

#### Notação Gráfica

A estrutura do Guaraná é fundada nos padrões de integração de aplicações, documentados por Hohpe e Woolf [25]. Utilizar essa representação de



**Tabela 2.1:** Notação Gráfica da Linguagem Específica de Domínio do Guaraná

tarefas possibilita à tecnologia Guaraná projetar a estrutura interna de todos os processos de integração e suas portas de comunicação.

O Guaraná DSL é composto por cinco elementos principais: aplicações, processos, *slots*, tarefas e portas. Na Tabela §2.1 mostra-se como cada um desses componentes são representados graficamente. O Guaraná permite a modelagem de diferentes tipos de tarefas. Cada tarefa é representada graficamente por um ícone, o qual está relacionado à função que essa tarefa desempenha.

As soluções de integração modeladas com Guaraná, geralmente, são compostas por várias tarefas, as quais podem aparecer várias vezes no mesmo modelo. A comunicação entre as tarefas é representada por meio de *slots*, que atuam como *buffers*, as quais recebem a mensagem já processada da tarefa anterior e a deixam disponível para ser processada pela tarefa seguinte. Uma vez que uma mensagem é processada e despachada para os *slots* seguintes, a tarefa está pronta para processar outra mensagem [19].

### Processos

Os processos são blocos que agrupam um determinado conjunto de tarefas e contém portas para a comunicação com as aplicações integradas. Eles possuem portas de entrada, saída, solicitação e resposta, pelas quais um processo recebe/envia informações das/para as aplicações integradas.

Routers	Modifiers	Transformesrs	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	 Set Correlation ID		
 Threader	 Set Return Address		
	 Custom Router		

Tabela 2.2: Grupos de tarefas do Guaraná

### Slots

*Slots* são unidades de armazenamento temporário que conectam portas com tarefas ou duas tarefas. Os *Slots* possibilitam o processamento assíncrono das mensagens que fluem dentro de um processo, proporcionando independência entre as tarefas.

### Tarefas

As tarefas são os principais elementos que compõem os processos, uma vez que são responsáveis pelo processamento e modificação das mensagens. A tarefa lê uma mensagem do *slot* de entrada, processa (de acordo com sua semântica) e a escreve no *slot* seguinte, deixando-a disponível para a tarefa seguinte. Este processamento de mensagens geralmente consiste de tradução, filtragem, encaminhamento, etc. Na Tabela §2.2 apresenta-se as principais tarefas subdivididas em grupos conforme suas atribuições dentro do processo, apresentando os ícones e seus respectivos nomes.

As tarefas se dividem em cinco grupos: roteadoras, modificadoras, transformadoras e temporais. Na sequência são discutidas as principais características de cada grupo de tarefas.

- Roteadoras: são 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 as principais tarefas desse grupo cita-se o *Filter* (filtro) e o *Replicator* (replicador). Filtro é utilizado para evitar que mensagens desnecessárias alcancem a tarefa seguinte, dessa forma, uma tarefa filtro recebe uma mensagem de entrada e com base em seus critérios dá permissão a mensagem para continuar ou ser removida do fluxo, podendo ser salva em um repositório de integração com a premissa de manter essa informação. Por sua vez, o replicador faz cópias de uma mensagem de entrada quando a solução requer enviá-la para dois ou mais destinos. Seu conteúdo original não é afetado, não há limite para as cópias de mensagens desde que o número de mensagens seja igual ao número de *slots* nos quais o replicador pode escrever.
- Modificadores: Esse grupo de tarefas tem a função de modificar o conteúdo original de uma mensagem de entrada. A alteração pode ser feita acrescentando informações ao conteúdo, traduzindo o conteúdo da mensagem para outro formato ou até mesmo produzir novas mensagens. Significa que a mensagem que chega não é a mesma mensagem que vai continuar no fluxo, após passar por uma tarefa modificadora. O Enriquecedor de Conteúdo atua sobre as mensagens quando for necessário adicionar mais informações para uma mensagem. Este recebe uma mensagem de entrada e busca mais informações num recurso externo baseada no conteúdo da mensagem original e acrescenta a ela.
- Transformadora: Esse grupo de tarefas tem a função de modificar a estrutura de uma mensagem, como por exemplo o *Translator* (tradutor), ou ainda construir mensagens novas a partir de outras como *Aggregator* (agregador) e o *Splitter* (Separador). O Tradutor tem a função de traduzir mensagens de um formato para outro, uma vez que é necessário quando se integra aplicativos que geralmente trabalham com diferentes formatos de mensagens. O tradutor recebe uma mensagem de entrada, a traduz para o novo formato e a envia para o próximo elemento. O Agregador é uma tarefa *stateful*, que pode receber duas ou mais mensagens de entrada e agrupa-las em apenas uma mensagem, utilizada quando tem-se diferentes mensagens com resultados individuais que podem ser combinados, ter o seu conteúdo individual reunido e ser processado como uma única mensagem para o fluxo seguinte. O Separador tem função contrária do agregador, funciona como divisor, recebendo uma mensagem de entrada e produzindo duas ou

mais mensagens de saída que serão processados individualmente.

- Temporais: Em determinadas circunstâncias é preciso "controlar o tempo" em um fluxo de integração, antecipar ou atrasar uma execução. Dentre as tarefas que possibilitam interferir no tempo estão o *Expire Checker* (Verificador de Validade) e o *Ticker* (Temporizador). Verificador de Validade é uma tarefa que se pode configurar para verificar o tempo de permanência de uma mensagem no fluxo da integração, tendo a permissão de excluí-la se houver um tempo de permanência no fluxo maior que o permitido. O temporizador é uma espécie de redutor de velocidade de fluxo, aplicado em casos que aplicativos integrados ou blocos de processos acabam ficando sobrecarregados porque recebem mais mensagens que podem processar, um temporizador na entrada de fluxo atrasa a entrada de novas mensagens.

### Portas

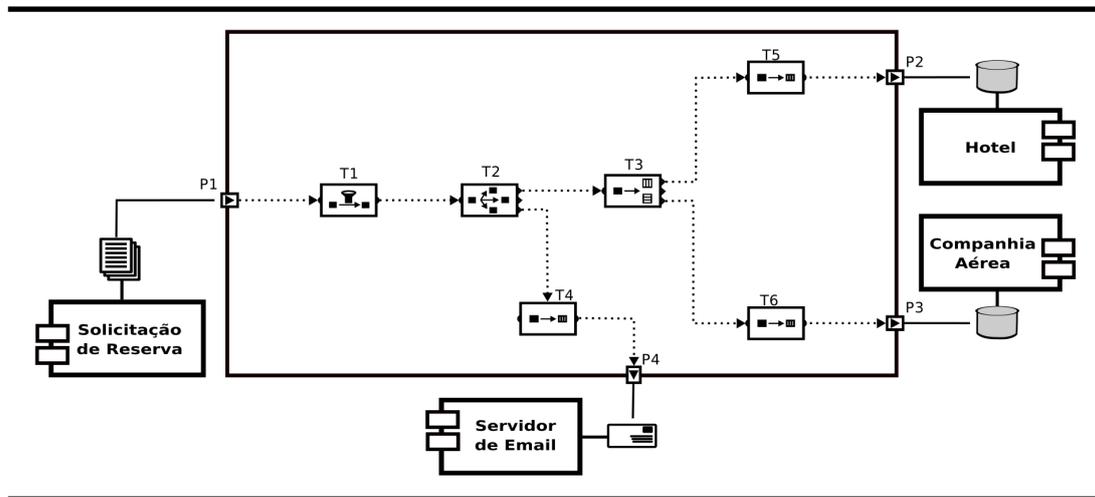
As portas estão acopladas aos processos, e possibilitam enviar e receber mensagens(s) de um para outro. Uma porta de entrada escreve uma mensagem de entrada para um *slot* interno e este a disponibiliza para a uma tarefa poder lê-la. Já a porta de saída, lê sempre uma mensagem a partir de um *slot* e deixa disponível para o elemento seguinte no fluxo. Portas de entrada e portas de saída estão sempre ligadas umas com as outras. Há variações de portas possibilitam entrada e saída acoplada a mesma, as Portas de Solicitação e as Portas de Resposta.

### Exemplo de Aplicação

Para ilustrar o uso da linguagem Guaraná na modelagem de soluções de integração, apresenta-se uma solução de integração para um problema real de integração, não trivial, que aborda problemas de reservas de passagem aéreas e de estadias em hotéis. O modelo visa automatizar o processo de reserva de passagens junto à companhia aérea e hospedagem em um hotel.

### Ecossistema de Software

O problema de integração que será apresentado foi proposto por Da Silva et al. [12] e envolve quatro aplicações. A aplicação Solicitação de Reserva é responsável por criar o arquivo de solicitação da reserva de hotel e da passagem aérea, de acordo com as especificações do cliente. A aplicação Hotel precisa receber a solicitação de reserva criada na aplicação Solicitação de Reserva, da mesma forma que Agência de Viagens, necessita receber a solicitação de reserva de passagens. Ao final é necessário que seja encaminhado um e-mail ao cliente confirmando as reservas contendo os dados das mesmas.



**Figura 2.1:** Exemplo Modelo Conceitual de Solução de Integração

### Modelo Conceitual

O modelo conceitual descrito na Figura §2.1, representa uma possível solução, modelada com o Guaraná DSL, para este problema de integração.

O processo de integração começa na porta de entrada (P1) que encaminha a mensagem para uma Filtro (T1) que verifica se o arquivo contém tanto solicitação de reserva como de hospedagem. A mensagem contendo os requisitos é encaminhada à tarefa Replicador (T2) que replicará a mensagem, encaminhando as cópias geradas para os seus respectivos destinos. Neste caso de estudo serão geradas duas cópias, a primeira encaminhada para tarefa Chopper (T3) e a outra para tarefa Tradutor (T4). A tarefa T3 têm a função de dividir a mensagem em duas, salvo que a mensagem de entrada tenha informações referentes a reserva de hotel e passagens aéreas, a tarefa divide a mensagem construindo uma mensagem contendo informações da reserva do hotel e encaminha para tarefa Tradutor (T5). A outra mensagem contendo informações da reserva de passagem aérea é encaminhada para a Tarefa Tradutor (T6).

Observando o modelo conceitual percebe-se que as tarefas T4, T5 e T6 possuem funções são semelhantes, têm a função de alterar o formato da mensagem que circula na integração para o formato da aplicação externa. A diferença entre elas é que o formato de saída de cada mensagem será de acordo com a solução integrada.

O processo prossegue com a transição T4 enviando uma mensagem para o servidor de e-mail contendo as informações de reserva de hotel e passagens aéreas para o e-mail do cliente. A tarefa T5 envia uma mensagem para a aplicação Hotel solicitando uma reserva e a tarefa T6 para a companhia aérea, e o processo está concluído.

### Modelo de Execução

O Guaraná propõe um modelo de execução com base em tarefas. O modelo com base em tarefas permite que as tarefas sejam executadas logo que haja uma mensagem à sua disposição, sem ter que esperar por todas as mensagens necessárias para iniciar o processo. No modelo de execução baseado em tarefas, há correlação de mensagens que devem ser processadas em conjunto e é feito dentro do processo por uma tarefa com esta função específica. Dessa maneira aos recursos do sistema são utilizados de forma mais eficiente [20].

As *threads* representam um recuso do *hardware* para uso das tarefas para executar suas mensagens e são responsáveis pela execução das tarefas. Tarefas aptas a disparar se agrupam em uma fila e aguardam a disponibilidade do recurso computacional para realizar seus processamentos. Uma tarefa está apta a dispara quando possuir ao menos uma mensagem em cada um de seus *Slots* de entrada

## 2.2 Simulação

A simulação é a imitação da operação de um processo do mundo real que geralmente transcorre num intervalo de tempo. A simulação tem por objetivo compreender o funcionamento, prever o desempenho e o comportamento de um sistema. A simulação é dependente de modelos que são a representação do sistema em questão, que reproduz (simula) seu funcionamento. Com a simulação possibilitou-se resolver diversos problemas sem a necessidade de interferir diretamente no objeto estudado Strang e Nguyen [43] (apud SENA [38]).

Um sistema pode ser classificado como discreto ou contínuo. Um sistema discreto apresenta mudanças bruscas em suas variáveis de estado em intervalos de tempo. O atendimento ao cliente em um banco pode ser descrito como exemplo de sistema discreto uma vez que só há alteração nas variáveis de estado (os clientes do banco) na chegada de um novo cliente ou quando é atendido e vai embora. Já um sistema contínuo relata mudanças contínuas em

suas variáveis com o decorrer do tempo. O voo de um avião pode ser descrito como sistema discreto de maneira que há uma constante e contínua mudança, por exemplo, das variáveis altitude e posição em relação ao tempo [18].

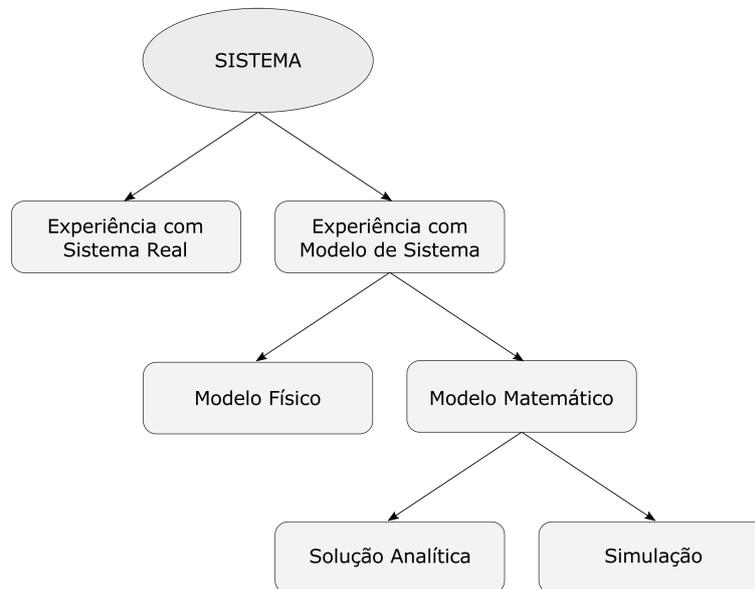
Um sistema de eventos discretos (SED) é um sistema que evolui de acordo com a sua ocorrência de eventos. Os eventos podem ser propositais, espontâneos, ou o resultado de verificação de uma condição, mudam o estado do sistema, os quais geralmente ocorrem em momentos de tempo irregulares.

A ocorrência de eventos é que causa mudanças de estado no sistema e determinam sua evolução dinâmica. Os SEDs apresentam três características básicas: a primeira é a ocorrência de eventos encadeados que determinam as tarefas que serão realizadas, a segunda é a ocorrência de eventos simultâneos que são vários eventos que podem ocorrer ao mesmo tempo e a terceira é a sincronização de atividades, que para haver a evolução do sistema é necessário que se espere o término de determinadas atividades para dar início às seguintes. Geralmente a quantidade de eventos de um SED é finita. Dessa forma ao realizar um estudo de análise e modelagem de um SED é crucial a definição dos eventos mais relevantes. A determinação dos eventos mais importantes junto com o modelo padrão de modelagem proporcionam uma ferramenta matemática para estudo do comportamento de um sistema de eventos discretos [11].

A Figura §2.2 adaptada de Kelton e Law [27] mostra as possibilidades de se realizar estudos dos sistemas. Para identificar o comportamento que um sistema em questão apresenta quando executado, observa-se seu funcionamento ou se não for possível constrói-se um modelo para representar o sistema real e obter dados. A construção de um modelo é necessária para representar o funcionamento de um sistema que por algum motivo é preciso ter conhecimento antes de ser construído. O modelo de sistema pode ser matemático e ser representado por equações matemáticas, ou físico que é um protótipo do sistema real. A complexidade do modelo matemático do sistema delimitará se é possível encontrar a sua solução analítica ou se deve optar pela simulação do mesmo.

### 2.2.1 Modelos

Um modelo tem a função de descrever a relação de componentes que integram um sistema, tendo em vista a melhor aproximação possível do comportamento real do sistema modelado, ou seja, um modelo tem por objetivo ser a representação fiel do sistema real. A finalidade de um modelo de



**Figura 2.2:** Possibilidade de Estudo de Sistemas

sistema é possibilitar o estudo de forma abstrata, com a facilidade de realizar modificações em sua estrutura e avaliar seus resultados, detectar possíveis falhas, tudo sem a necessidade de construí-lo [13, 27].

Para realizar alterações em sistemas físicos e verificar possíveis erros de execução é necessária a sua implantação, acarretando em despesas e interrupção de seu funcionamento. . Em contrapartida, o modelo de sistemas, por meio da simulação, permite obter uma prévia dos resultados gerados pelo sistema, podendo o sistema representado pelo modelo ser submetido há situações extremas de desempenho, com a contribuição de eliminar quase totalmente possíveis erros e evitar futuros transtornos. Além desses benefícios devemos considerar que em alguns casos trabalha-se com projetos de sistemas ainda em construção, ou não é possível realizar experimentos por motivos financeiros, ou então por haver risco de vida [27].

Vantagens da simulação de sistemas podem ser representadas num exemplo simples que ilustra o sistema de caixas de atendimento de um banco, com a proposta de reduzir custos com a redução do número de caixas. A redução do número de caixas sem um estudo das reações do funcionamento do sistema com essa alteração pode acarretar na formação de filas, sobrecarga dos caixas em funcionamento e a queda na qualidade dos serviços oferecidos [27, 41].

A complexidade de um sistema é que determina o modelo que melhor vai representá-lo. Em sistemas pouco complexos é possível determinar um modelo algébrico para descrevê-los e usar ferramentas matemáticas e artifícios lógicos para aproximar os resultados da realidade de seu funcionamento. A construção de um modelo que descreva a operação do sistema depende da complexidade de seus componentes, sendo um sistema simples é possível utilizar ferramentas matemáticas (álgebra, cálculo, ou probabilidade) para obter informações precisas e chegar a uma solução analítica. Por outro lado, grande parte dos sistemas são complicados demais para permitir a construção de modelos realistas para estudá-los analiticamente, sendo necessário o uso de simuladores para um melhor estudo e entendimento [27].

## 2.2.2 Classificação de Modelos

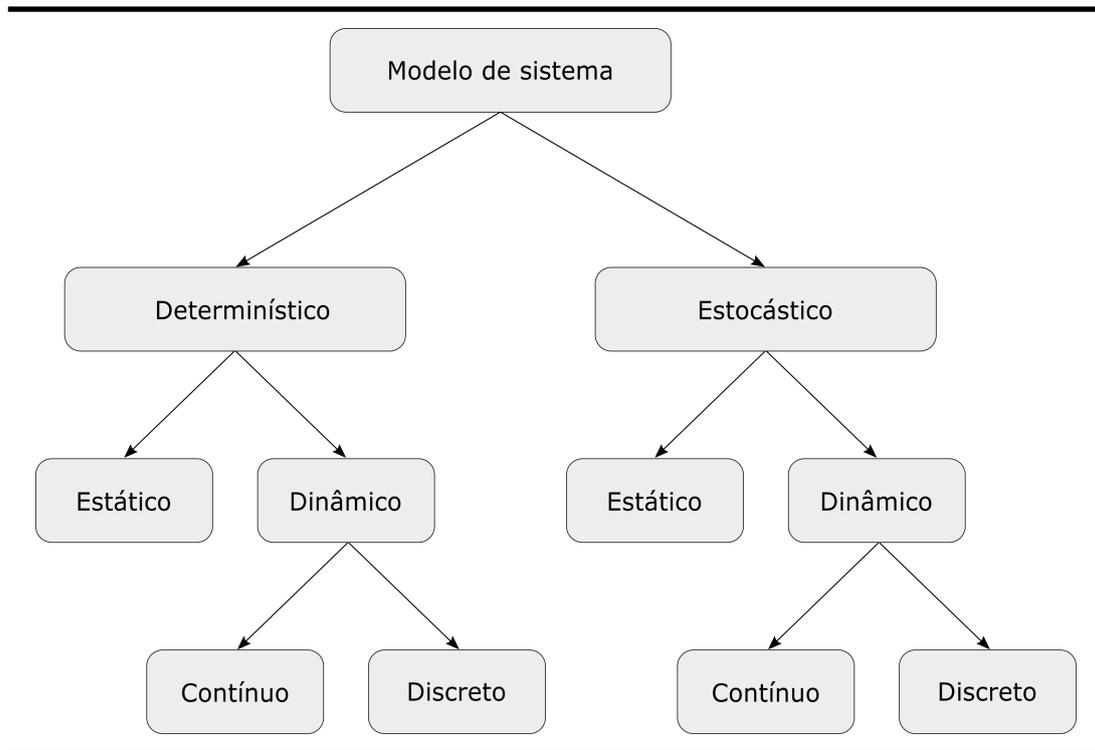
Um sistema pode ser representado por meio de modelos simbólicos (ou icônicos), analíticos e modelos de simulação [14]. Os modelos simbólicos são estruturados com símbolos gráficos (retângulos e retas), úteis para descrever a sequência de funcionamento, facilitam o entendimento e documentação do sistema, mas o representam de forma descritiva e se torna uma fraca ferramenta de análise quantitativa.

Os modelos analíticos, por outro lado descrevem um sistema através de equações ou um conjunto destas. Resolvendo essas equações chegamos à solução esperada. Os modelos analíticos fornecem informações precisas, a resolução se torna complicada se o sistema a ser modelado for extremamente complexo, em alguns casos é preciso recorrer a artifícios matemáticos para se estimar esta solução. Esses métodos podem acarretar na perda de qualidade representativa de um modelo, dispersando os resultados da realidade do sistema [27].

Os modelos de simulação apresentam como grande diferencial, segundo Strack [42], a característica de não resultar em soluções analíticas. Estes modelos não são resolvidos, mas sim "executados", ao contrário dos modelos analíticos, são técnicas ou ferramentas que possibilitam o modelo trabalhar da mesma forma do que o sistema real estudado, e resultam em dados conclusivos e previsíveis do funcionamento.

## 2.2.3 Modelos de Simulação

Os modelos de simulação são implementados através de linguagem de programação auxiliados por computadores. Alguns modelos podem apresentar alguma dificuldade de implementação devido a complexidade da



**Figura 2.3:** *Classificação dos Modelos de Simulação*

sua estrutura o que pode conduzir a resultados pouco precisos, caracterizando uma desvantagem dos modelos de simulação. Por sua vez se destaca quando trata de modelos com número grande de variáveis e que seriam impossíveis de resolver analiticamente. O desenvolvimento de um modelo de simulação leva em conta a os objetivos da simulação e as variáveis de saída do modelo [10].

A Figura 2.3, adaptada de Kelton e Law [27], apresenta a classificação dos modelos de simulação de acordo com o tratamento de suas variáveis de estado, e pelo tipo de variáveis que utilizam.

Um modelo determinístico trabalha com apenas um determinado tipo de variável e resulta em valores iguais para entradas iguais, ou seja, para uma mesma entrada teremos sempre a mesma saída. Já modelos que fazem uso de variáveis aleatórias, são denominados modelos estocásticos, que vem a ser um sinônimo de probabilístico, pode haver entradas iguais e saídas diferentes por que estes modelos ponderam fatos aleatórios que determinam a saída. Grande parte dos modelos de simulação são classificados como modelos estocásticos [10].

Um modelo pode ou não relacionar a variável tempo em sua estrutura de simulação. Modelos de sistema que não consideram o tempo são denominados modelos estáticos. Se o tempo é importante para dinâmica do sistema modelado, então é denominado modelo dinâmico. A principal característica que distingue os sistemas estáticos de dinâmicos é a representação, onde os estáticos podem ser descritos por equações algébricas, já os dinâmicos representados por equações diferenciais. No fundo todo sistema acaba por ser dinâmico, com a diferença de em alguns casos a variação do tempo não causar interferências plausíveis nos resultados [3].

Modelos dinâmicos subdividem-se em contínuo ou discreto. Um sistema dinâmico contínuo considera que um sistema evolui de forma contínua no tempo total de execução. Por sua vez os modelos dinâmicos discretos delimitam a evolução do tempo em instantes, havendo uma fragmentação do tempo de acordo com as etapas contidas em cada sistema, e avaliando o consumo de tempo de cada etapa [3].

Uma vez modelado computacionalmente é possível realizar experimentos no modelo e observar resultados imediatos. Esses resultados podem representar características importantes do sistema real [13].

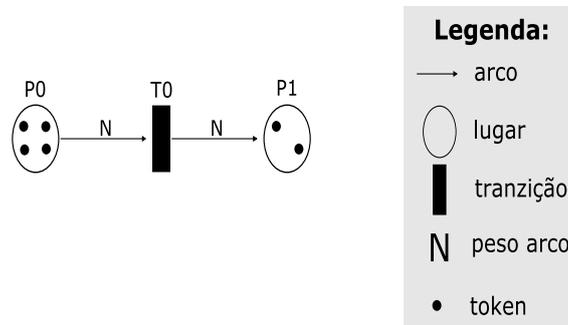
## 2.3 Redes de Petri

As Redes de Petri são uma linguagem de modelagem matemática que permite representar sistemas de eventos discretos. As ferramentas de suporte às Redes de Petri oferecem um ambiente para a modelagem, análise e simulação de sistema, possibilitando a visualização estrutural e comportamental de forma simultânea. Esta seção apresenta a notação gráfica e formal das Redes de Petri, algumas de suas extensões, uma discussão sobre o modelo de execução e apresentação de um exemplo onde são aplicadas as definições introduzidas.

### 2.3.1 Notação Gráfica e Formal

As Redes de Petri são grafos bipartidos, pois apresentam dois tipos de nós, chamados de lugares e transições, principais componentes da sua representação gráfica. Os lugares, representados por círculos ou elipses, equivalem as variáveis de estado de um sistema, já as transições, representadas por barras equivalem às ações realizadas pelo mesmo.

Os grafos das Redes de Petri são dirigidos, sempre de lugares para transições e/ou de transições para lugares. Lugares e transições são ligados por



**Figura 2.4:** Elementos Básicos Redes de Petri, adaptada de Barros [5]

arcos. Um arco liga lugar à transição, e transição a arco e são representados por setas que indicam o sentido do fluxo do sistema modelado. Os arcos de uma Redes de Petri jamais realizam conexões diretas entre transições ou lugares [5].

Na Figura §2.4, representa-se uma Redes de Petri básica, grafo contendo os elementos principais de uma Redes de Petri, onde se verifica os lugares representados por P0 (lugar 1) e P1 (lugar 2), a transição T0 e os arcos ligando lugares a transições e transições a lugares [5].

Uma Rede de Petri pode possuir *Tokens* ou marcas que se encontram nos lugares. As transições quando são ativadas consomem *Tokens* dos lugares que as alimentam e os geram nos lugares alimentados por elas. Os *Tokens* são indicadores de fluxo em uma Redes de Petri e são representados por pontos pretos dentro dos lugares. A quantidade de *Tokens* nos lugares e sua posição oscilam de acordo com o funcionamento da rede. O número de *Tokens* consumidos por uma transição é igual ao valor do peso dado ao arco de entrada. Uma Redes de Petri que apresenta *Tokens* é definida como marcada [30].

A representação formal de uma Redes de Petri R genérica esta representada nas equações §2.1 até §2.6. A Equação §2.1 apresenta uma Tupla, sendo L o conjunto dos lugares expresso na Equação §2.2, T o conjunto das Transições expresso na a Equação §2.3, A o conjunto dos arcos expresso na Equação §2.4, PA o conjunto dos pesos dos arcos expresso na a Equação §2.5 e  $M_0$  o conjunto da marcação inicial da rede expresso na a Equação §2.6.

$$R = \{L, T, A, PA, M_0\} \quad (2.1)$$

$$L = \{P_0, P_1, \dots, P_n\} \quad (2.2)$$

$$T = \{t_0, t_1, \dots, t_n\} \quad (2.3)$$

$$A = \{A_1, A_2, \dots, A_n \text{ ou} \quad (2.4)$$

$$A = \{(P_0, t_0), (t_0, P_1), (P_1, t_1), (t_0, P_0)$$

$$PA = \{PA_1, PA_2, \dots, PA_n\} \text{ ou} \quad (2.5)$$

$$PA = \{PA(P_0, t_0), PA(t_0, P_1), PA(P_1, t_1), PA(t_0, P_0), \dots, PA(t_n, P_n)\}$$

$$M_0 = \{M_0(P_0), M_0(P_1), M_0(P_2), \dots, M_0(P_n)\} \quad (2.6)$$

As Equações §2.7 e §2.8 determinam a condição de existência da Redes de Petri, onde a Equação §2.7 determina que lugares e transições são conjuntos distintos e a Equação §2.8 determina que uma Redes de Petri tem ao menos um lugar e uma transição.

$$L \cap T = \emptyset \quad (2.7)$$

$$L \cup T \neq \emptyset \quad (2.8)$$

### 2.3.2 Notação Matricial

As Redes de Petri também podem ser representadas por matrizes, conforme apresenta Costa [11]. A representação de redes por meio das matrizes é baseada basicamente no mapeamento dos arcos de entrada e saída. Pela notação matricial é possível representar uma rede simples apenas pelas matrizes dos arcos de entrada e saída e pela marcação inicial da rede. As matrizes seguem lei de formação conforme a Tabela §2.3.

A matriz  $A_e$  representa o mapeamento dos arcos de entrada, por sua vez a matriz  $A_s$  representa os arcos de saída. De acordo com a lei de formação, a matriz genérica  $A_e$  e  $A_s$  estão representadas nas Equações §2.9 e §2.10 respectivamente. As linhas representam os Lugares e as colunas as transições. A ordem das matrizes é determinada com o número de linhas igual ao número de lugares e número de colunas igual ao número de transições, onde a ordem de  $A_e$  é descrita na Equação §2.12 e de  $A_s$  é descrita na Equação §2.13. Generalizando, a ordem das matrizes dos arcos de entrada e de saída são iguais, desta forma a ordem das matrizes terá o número de linhas igual ao cardinal (número de elementos do conjunto) do conjunto dos

$N =$ Valor do peso do arco
$Ae_{(ij)} = N:$ se $\exists$ Arco de ligação direcionado do lugar $i$ para a transição $j$ ;
$Ae_{(ij)} = 0:$ se $\nexists$ Arco de ligação direcionado do lugar $i$ para a transição $j$ ;
$As_{(ij)} = N:$ se $\exists$ Arco de ligação direcionado da transição $j$ para o lugar $i$ ;
$As_{(ij)} = 0:$ se $\nexists$ Arco de ligação direcionado da transição $j$ para o lugar $i$ .

**Tabela 2.3:** Lei de Formação Notação Matricial

lugares e o número de colunas igual a cardinal do conjunto das transições.

$$A_e = [a_{ij}] = \begin{bmatrix} a_{e(ij)} & \dots & T_0 & T_1 & T_2 & \dots & T_n \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ S_0 & \dots & (S_0; T_0) & (S_0; T_1) & (S_0; T_2) & \dots & (S_0; T_n) \\ S_1 & \dots & (S_1; T_0) & (S_1; T_1) & (S_1; T_2) & \dots & (S_1; T_n) \\ S_2 & \dots & (S_2; T_0) & (S_2; T_1) & (S_2; T_2) & \dots & (S_2; T_n) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ S_n & \dots & (S_n; T_0) & (S_n; T_1) & (S_n; T_2) & \dots & (S_n; T_n) \end{bmatrix} \quad (2.9)$$

$$A_s = [a_{ij}] = \begin{bmatrix} a_{(ij)} & \dots & T_0 & T_1 & T_2 & \dots & T_n \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ S_0 & \dots & (S_0; T_0) & (S_0; T_1) & (S_0; T_2) & \dots & (S_0; T_n) \\ S_1 & \dots & (S_1; T_0) & (S_1; T_1) & (S_1; T_2) & \dots & (S_1; T_n) \\ S_2 & \dots & (S_2; T_0) & (S_2; T_1) & (S_2; T_2) & \dots & (S_2; T_n) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ S_n & \dots & (S_n; T_0) & (S_n; T_1) & (S_n; T_2) & \dots & (S_n; T_n) \end{bmatrix} \quad (2.10)$$

$$A_e = [a_{(ij)}] \quad (2.11)$$

$i = |L|$   
 $j = |T|$

$$A_s = [a_{(ij)}] \quad (2.12)$$

$$i = |L|$$

$$j = |T|$$

### 2.3.3 Execução

O disparo das transições executam as Redes de Petri. O disparo de uma transição é feito se a transição estiver apta para isto. As transições se tornam aptas para disparar quando todos seus lugares de entrada contiverem pelo menos o número de Tokens igual ao peso do arco. Transições que possuem apenas lugares de saída são uma exceção a regra e estão sempre aptas a disparar, e inserem *tokens* os seus lugares de saída. Transições que possuem apenas lugares de entrada são outra exceção, estão aptas a disparar quando tiverem *tokens* os seus lugares de entrada e não os constroem em nenhum lugar.

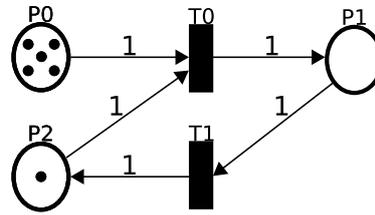
As transições consomem *tokens* nos lugares de entrada e constroem nos lugares de destino. O disparo de eventos nas Redes de Petri não prevê nem impõe uma ordem de ocorrência. Em um modelo de sistema, em algum instante, haverá mais do que uma transição habilitada a disparar, e qualquer uma pode disparar. O disparo é dividido aleatoriamente entre as transições aptas ao disparo, sem seguir uma regra de prioridade [6].

### 2.3.4 Exemplo de aplicação

Para demonstrar a execução de um modelo de sistema em Redes de Petri, nesta subseção propõe-se modelar um sistema qualquer, aplicar a notação gráfica, formal e matricial, descrever o mecanismo de execução e representar os estados do sistema.

A Figura §2.5 apresenta um sistema  $V$ , modelado em Redes de Petri, descreve um sistema qualquer que recebe solicitação de processamento de usuários para realizar uma determinada tarefa. Representando o sistema pela Tupla  $V$ ,  $P$  representa o conjunto dos lugares,  $T$  representando o conjunto das transições,  $P$  representa a marcação inicial do sistema.

Atribuindo uma semântica a cada elemento do sistema, o lugar  $P_0$  representa usuários requisitando processamento ao sistema, o lugar  $P_1$  equivale ao sistema em processamento e o lugar  $P_2$  o retorno do sistema ao estado ocioso. As transições  $T_0$  e  $T_1$ , representam, respectivamente, início e fim do processamento.



**Figura 2.5:** Sistema  $V$

O sistema  $V$  é representado pela tupla conforme a Equação §2.13, onde  $L$  representa o conjunto dos lugares descrito na Equação §2.14,  $T$  o conjunto das transições descrito na Equação §2.15,  $A$  o conjunto dos arcos de entrada ( $A_e$ ) e saída ( $A_s$ ) descrito na Equação §2.17,  $PA$  o conjunto dos pesos dos arcos descrito na Equação §2.18 e  $M_0$  o conjunto das marcações iniciais descrito na Equação §2.19.

$$V = \{L, T, A, PA, M_0\} \quad (2.13)$$

$$L = \{P_0, P_1, P_2\} \quad (2.14)$$

$$T = \{t_0, t_1\} \quad (2.15)$$

$$A = \{A_e, A_s\} \quad (2.16)$$

$$A_e = \{(P_0, t_0), (P_2, t_0), (P_1, t_1)\}$$

$$A_s = \{(t_0, P_1), (t_1, P_2)\}$$

$$(2.17)$$

$$PA = \{1, 1, 1, 1, 1\} \quad (2.18)$$

$$M_0 = \{5, 0, 1\} \quad (2.19)$$

A representação matricial do Sistema V é descrita na sequência, com ordem determinada pelas Equações §2.21 e §2.22. Dessa maneira as matrizes  $A_e$  e  $A_s$  são representadas respectivamente pelas Equações §2.22 e §2.23.

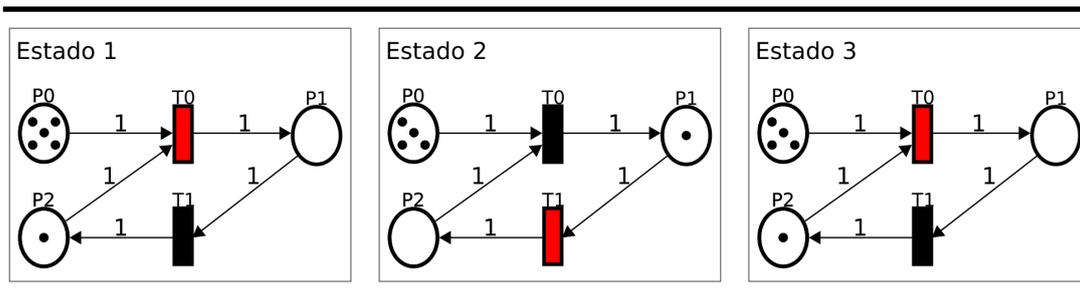
$$\begin{aligned} A_e &= [a_{(ij)}] \\ i &= |L| = 3 \\ j &= |T| = 2 \end{aligned} \quad (2.20)$$

$$\begin{aligned} A_s &= [a_{(ij)}] \\ i &= |L| = 3 \\ j &= |T| = 2 \end{aligned} \quad (2.21)$$

$$A_e = [a_{ij}] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (2.22)$$

$$A_s = [a_{ij}] = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.23)$$

A Figura §2.6 descreve os estados do sistema V e demonstra o sistema de execução do modelo. A condição que o sistema se encontra é representada pela presença do *token* no lugar P1 ou P2. A posição do *token* determina que o sistema está ocioso, aguardando solicitação de serviço. A dinâmica de execução ocorre quando há ao menos um *token* no Lugar P0, indicando que há



**Figura 2.6:** Estados do Sistema V

um usuário solicitando serviço ao sistema, e há também ao menos um *token* no lugar P2 indicando que o sistema está disponibilizando um recurso.

O Estado1 descreve o estado ocioso do sistema, a transição T0 está apta a disparar porque se tem o recurso do sistema e usuários solicitando processamento. O Estado2 indica que a transição T0 disparou e o sistema saiu do estado ocioso, foi consumido um *token* do lugar P2 e P0 e construído no lugar P1, representa início ao processamento solicitado por um usuário e a transição T1 está apta a disparar. No Estado3, o disparo da transição T2 representa a conclusão do processamento, quando o evento T1 ocorrer, o *Token* será removido do lugar P1 e colocado no lugar P2, indicando que o sistema concluiu a tarefa que lhe foi designada e volta a ficar ocioso à espera de outra solicitação para repetir o processo.

### 2.3.5 Redes de Petri Estocásticas

O formalismo das Redes de Petri possibilita descrever apenas a estrutura lógica de sistemas, pois tal não considera o tempo. Entretanto, o tempo interfere no funcionamento da maioria dos sistemas, tem um papel importante em sua estrutura, e precisa ser considerado na sua representação.

As Redes de Petri Estocásticas determinam um atraso às suas transições associando às transições probabilidade de disparo e alterar sua frequência de disparo. Define-se Redes de Petri Estocásticas assumindo como temporizadas todas as transições.

A representação algébrica das Redes de Petri Estocásticas é semelhante às Redes de Petri, descritas por conjunto de transições, lugares, arcos e marcação inicial, anexando-se conjunto das taxas de disparo de cada transição.

A representação gráfica também é semelhante, com alteração nas transições, que continuam sendo representados por retângulos, porém com interior branco para diferenciar das não estocásticas.

Imaginando um sistema, em um determinado instante terão várias tarefas aptas ao processamento, considerando que disponha de apenas uma unidade de processamento, apenas uma delas irá realizar um processamento. Um modelo de Redes de Petri Estocástica, em determinados momentos, haverá diversas transições aptas a serem disparadas, e apenas uma delas ocorrerá. Ao ocorrer o disparo há um novo estado da marcação do sistema, e haverá marcações aptas a disparar que já estavam na marcação anterior, mas não foram, configurando-se o atraso do disparo. As Redes de Petri tem a propriedade da ausência de memória, a marcação anterior não interfere na posterior, as transições são disparadas aleatoriamente de acordo com a probabilidade de disparo associada a ela.

A probabilidade ou taxa de disparo de cada transição estocástica de uma rede é determinada de acordo com sua frequência de disparo esperada. A distribuição da probabilidade de disparo é realizada de acordo com a distribuição exponencial representada pela Equação §2.24, e determina a probabilidade de disparo para cada transição apta a disparar.

$$F_{X_i}(x) = 1 - e^{-\lambda x} \quad (2.24)$$

A função apresentada na equação  $F_{X_i}(x)$  representa a probabilidade de uma transição  $x$  apta disparar naquele instante, considerando sua Taxa de Disparo  $\lambda$ , a transição é disparada aleatoriamente.

## 2.4 Validação e Verificação

Um sistema, assim como um produto, precisa ser testado para trazer à tona erros em seu desenvolvimento. Ao desenvolver um modelo de simulação para obter informações de um sistema, há uma preocupação com a veracidade dos seus resultados, os quais irão interferir na qualidade do sistema a ser desenvolvido. Nenhum modelo é 100% verificado ou validado, uma vez que todo modelo é uma representação de um sistema e seu comportamento é, na melhor das hipóteses, uma aproximação do comportamento esperado do sistema real [8, 37].

Esta seção discute inicialmente os conceitos de validação e verificação, argumentando por que a verificação é a melhor alternativa para o modelo de simulação proposto. Na sequência, apresenta-se algumas metodologias de verificação de modelos e a sua aplicação.

### 2.4.1 Diferenciação entre Verificação e Validação

Em sistemas V&V são ferramentas para testes nos modelos de simulação a fim de assegurar sua confiabilidade e remete a duas perguntas:

Verificação: O modelo está sendo desenvolvido corretamente?

Validação: O modelo representa o sistema modelado?

A diferença entre verificação e validação causa confusão. A verificação questiona se o modelo foi desenvolvido corretamente e se retorna resultados compatíveis com o sistema modelado. A validação de um modelo comprova sua precisão suficiente para representar as transformações que o sistema exerce sobre suas entradas. A validação é executada sempre que um componente do modelo é comparado com a realidade. A verificação, por sua vez questiona a maneira de desenvolvimento de um modelo de sistema ou próprio sistema e geralmente faz referência ao projeto de desenvolvimento. A verificação tem a finalidade de fundamentar a tradução de um modelo de simulação de uma linguagem para outra, durante seu ciclo de desenvolvimento, de forma precisa [31, 35].

A Figura §2.7 apresenta um diagrama simplificado do processo V&V de modelos. A proposta de modelagem das soluções de integração tem por objetivo a identificação de gargalos de antes da implementação.

A validação de um modelo de simulação parte do modelo de simulação em direção à solução de integração implementada. De acordo com a proposta de trabalho, a Validação não pode ser aplicada por não haver dados do sistema real para aferição do modelo.

A verificação é independente da implementação, e atua na tradução do modelo conceitual em modelo de simulação equivalente em Redes de Petri. Com isso é possível desenvolver um modelo de simulação que represente as funcionalidades propostas pelo modelo conceitual asseguradas por metodologias que confirmam a correspondência entre ambos.

### 2.4.2 Metodologias de Verificação de Modelos de Simulação

As modelagens de modelos conceituais propostas neste trabalho não apresentam dados do sistema real. A literatura apresenta diversos trabalhos que relatam maneiras de verificar e validar modelos de simulação quando há dados do sistema real. Trabalhos que relatam técnicas de V&V de modelos sem

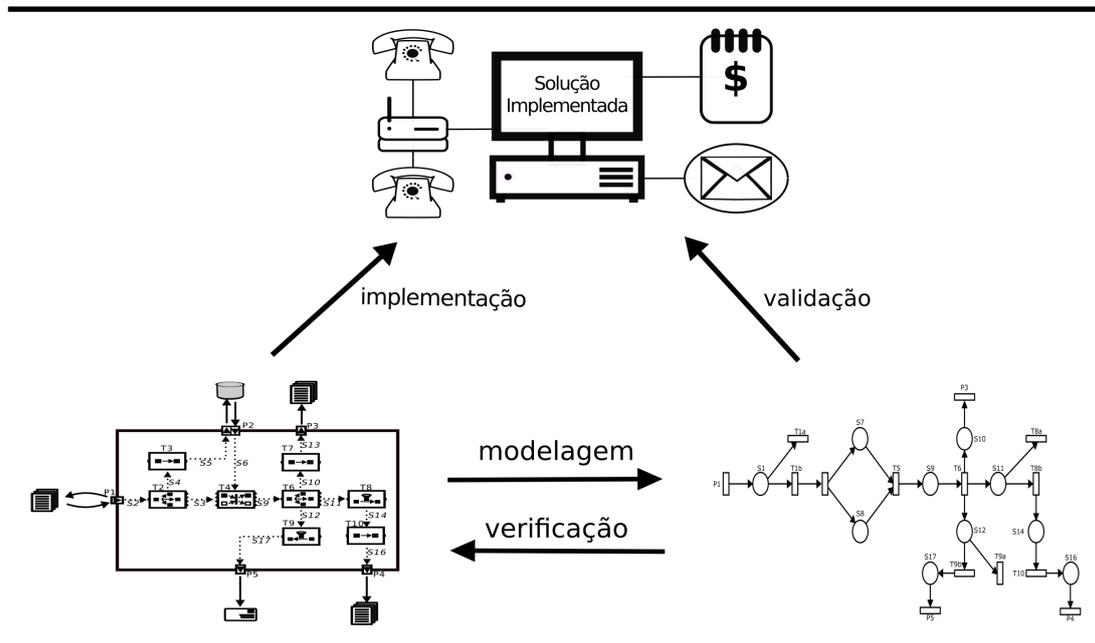


Figura 2.7: Diagrama do Processos V&V

dados reais são escassos. As técnicas aplicadas para verificação do modelo abordadas foram propostas por Kleijen [28] e por Sargent [37]. Na sequência apresentam-se as definições e proposta das técnicas citadas.

A técnica de V&V proposta por Kleijen [28] apresenta uma pesquisa sobre como validar e verificar modelos de simulação através da aplicação de técnicas estatísticas, de modo a que o tipo de técnica aplicada na verdade depende da disponibilidade de dados sobre o sistema real. Para verificação do modelo dispõe-se de conhecimento especializado e é por meio do conhecimento de especialistas no comportamento do sistema que a verificação implicará.

Um exemplo familiar do cotidiano são as filas de espera em supermercados, sabemos que em teoria o aumento da chegada de clientes por hora acarretará no aumento do tempo de espera, haverá um crescimento da fila, ao menos que se disponibilizem mais caixas para atender. O exemplo do sistema de atendimento do supermercado permite um conhecimento qualitativo. Para obter conhecimento quantitativo, um modelo de simulação é desenvolvido onde o efeito é conhecido, não sua magnitude. A análise estatística ocorrerá diante de resultados quantitativos resultantes das simulações.

O conhecimento qualitativo representa as características que os especialistas no sistema modelado esperam que ocorra. O comportamento do sistema

diante determinadas situações é conhecido, e vai interferir nos resultados dos modelos de simulação que devem seguir na mesma linha. Com dados numéricos resultantes das simulações é possível aplicar uma análise quantitativa e estimar com suporte estatístico a dimensão das suas estruturas.

Hipoteticamente os resultados experimentais devem acompanhar o comportamento que os especialistas esperam tanto do sistema como um todo como de um determinado elemento, no caso a formação de fila em determinado ponto do sistema ou consumo maior de tempo para processamento por uma determinada tarefa, por exemplo.

A observação de valores de entrada e saída do modelo também contribui para uma melhor verificação. Se a entrada saída do modelo de simulação violam as características qualitativas, é preciso questionar o modelo quanto a erros de modelagem, se as condições ou cenários estão adequadas.

Várias técnicas de verificação são propostas por Sargent [37]. Destas, duas delas são 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 no modelo computacional de simulação. O modelo de simulação é desenvolvido conforme o sistema que ele modela, e usa uma representação específica. A verificação tem que assegurar que o modelo de simulação foi implementado de forma correta na ferramenta. Desta forma a técnica propõe a comparação do número de elementos que entrou no sistema com a saída, partindo de uma premissa simples:  $\text{Entrada} = \text{Filas} + \text{Saída}$ .

Analisando a técnica sob a visão de um sistema baseado em mensagem, ela propõe realizar uma contagem das mensagens que entram no sistema. Este número de mensagens deve ser igual ao número de mensagens que saíram somados ao número de mensagens acumuladas nos *Buffers*.

A técnica de Validade Eventos implica 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 preciso que a taxa de ocorrência de determinado evento se aproxime quando realizado experimentos com o modelo de simulação.

Considerando um sistema de mensagens qualquer, diante uma situação que 12% das mensagens que adentram no sistema apresentam uma característica específica, e uma tarefa do sistema identifica mensagens que

tenham esta característica e às retira do fluxo. Da mesma forma, realizando uma modelagem do sistema e implementando-o numa ferramenta de simulação, os resultados do experimento devem apresentar que aproximadamente 12% das mensagens foram removidas do fluxo.

## 2.5 Trabalhos Relacionados

As aplicações de Redes de Petri para modelagem e simulação de sistemas de eventos discretos se expandem para diversas áreas do conhecimento devido a sua versatilidade de modelagem de situações. A sua notação gráfica permite identificar comportamentos dos sistemas modelados como concorrência, sincronização e compartilhamento de recursos. As Redes de Petri possuem várias extensões, semântica matematicamente definida o que possibilita um estudo detalhado do comportamento e propriedades de determinados sistemas, contribuem para identificação de problemas e propostas de melhorias. Na sequência aborda-se trabalhos que relatam o uso das Redes de Petri para simulação de sistemas de eventos discretos e a sua contribuição para avaliação da sua dinâmica. Os trabalhos abordados relatam o funcionamento dos sistemas modelados, características particulares de cada sistema e os resultados alcançados com a simulação.

O trabalho apresentado por Yamada et al. [44] propõe simular as etapas de funcionamento de uma indústria de beneficiamento de cana de açúcar utilizando Redes de Petri, enfatizando o sistema que abrange ações de recebimento, descarregamento e a movimentação de matéria prima até o processo de extração. A simulação desses processos visa analisar propostas de otimização e identificação de pontos vulneráveis sem interferir no funcionamento do sistema real, uma vez que o sistema produtivo sucroalcooleiro tem caráter sazonal. O sistema de recebimento de cana de açúcar e alimentação da central de moagem possui características discretas, e o tempo de execução de cada processo interfere na evolução e funcionamento correto do sistema. A elaboração do modelo parte do estudo das atividades do processo, assim como a identificação de uma entrada que representa a chegada da matéria prima transportada por caminhões, os quatro caminhos possíveis que representa os quatro tipos de descarga, encaminhamento da cana de açúcar para a central de moagem que representa a saída do sistema e o veículo é encaminhado ao setor de pesagem concluindo a operação.

A simulação do modelo proposto por Yamada et al. [44] constatou a presença de paralelismo de atividades, o qual acarretaria em uma sobrecarga na alimentação da central de moagem em situações de chegadas sucessivas de

matéria-prima. Recursos de descarregamento são compartilhados entre os setores e foi observada a necessidade do sincronismo de atividades, uma vez que é preciso uma alimentação constante de matéria-prima para beneficiamento, podendo comprometer a continuidade do processo em momentos de pouco abastecimento de cana-de-açúcar. Embora tenha denotado as variáveis do sistema, o modelo não considera suas características determinísticas e estocásticas, e o autor sugere a utilização de Redes de Petri temporizadas para distinguir cada etapa do processo e consolidar a validação do modelo.

Em seu trabalho Miyagi e Miyagi [32] realizou a modelagem de sistemas de saúde com abordagem das Redes de Petri interpretadas aplicado aos serviços do ambulatório do Hospital das Clínicas (HC) de São Paulo para caso de estudo e validação da proposta. A modelagem tem como objetivo estudar o fluxo de pacientes que procuram atendimento do ambulatório, pacientes não urgentes e, principalmente, pacientes que procuram atendimento pela primeira vez, com intuito de auxiliar a tomada de decisão pela gerência do sistema e analisar a evolução do sistema e o comportamento dos setores. O modelo foi desenvolvido pela metodologia PFS (*Production Flow Schema*) e MFG (*Mark Flow Graph*), interpretações de Redes de Petri, que evidenciam os recursos envolvidos e a movimentação dos componentes do sistema (pessoas, equipamentos, informações) onde os detalhes do sistema vão sendo gradativamente inseridos no modelo para uma melhor compreensão, considerando a evolução discreta do sistema. Foram colhidos dados na unidade de pronto atendimento do HC, identificação do fluxo de pacientes, e com base nas informações obtidas pelos funcionários do HC, o modelo foi sendo construído de acordo com as possíveis sequencias de serviços que os pacientes são submetidos, dando ênfase para pacientes que procuram o atendimento pela primeira vez.

A simulação do modelo proposto por Miyagi e Miyagi [32] considerou sugestões de alteração do fluxo de pacientes, identificou a possibilidade da eliminação de alguns processos que viriam a sobrecarregar setores do sistema e sugestão de novos processos e outros pontos do sistema. Por fim, juntamente com uma equipe multidisciplinar, o trabalho de modelagem e simulação do sistema de pronto atendimento do HC evidenciou constatações de como a teoria de Redes de Petri exige um conhecimento superficial em relação a outras técnicas de simulação, como a metodologia PFS usada na modelagem, auxiliados pelo recurso da animação do fluxo de itens do sistema proposta pelo simulador que permite a análise em determinados setores num instante qualquer e possibilita avaliar a situação.

O trabalho apresentado por Arteiro et al. [4] propôs um modelo de análise de desempenho de *middlewares* orientados a mensagens (MOMs) de-

envolvidas em Redes de Petri Estocásticas para auxiliar na investigação da capacidade desse tipo de *middleware* por meio da simulação. A simulação do modelo tem em vista a identificação da capacidade máxima de entrega de mensagens, o ponto de saturação do sistema quando exposto os ambientes de grande demanda de solicitação de serviços e o surgimento de gargalos de desempenho, mantendo um nível de processamento de dez milissegundos (10 ms). Os três sujeitos principais do sistema são os produtores (aplicativos que produzem mensagem de solicitação aos MOMs), consumidores (aplicativos que recebem mensagem dos MOMs) e os MOMs (direcionam as mensagens de clientes a consumidores). O modelo foi validado comparando dados obtidos da simulação com medições caixa-preta realizada em ambiente real utilizando MOM compatível com padrão *Java Message Service* (JMS) e, posteriormente, foi submetido situações de grande demanda de solicitações de serviço. A avaliação de desempenho do sistema avaliou valores como tamanho de fila, latência, vazão do sistema e utilização de um recurso, com fluxo de entrada de dez mensagens por segundo (10mps).

Os dados obtidos nas simulações do modelo proposto por Arteiro et al. [4] foram submetidos a um tratamento e representados graficamente para uma melhor interpretação. A simulação possibilitou identificar o crescimento da latência, vazão do sistema e fila aumenta à medida que aumentam o número de produtores. Conforme as determinações básicas do sistema real, concluem que o sistema satura em torno de 29 produtores e surgem gargalos no seu desempenho.

O trabalho apresentado por Roos-Frantz et al. [36] propõe a simulação de modelos conceituais de soluções de integração desenvolvidos na tecnologia Guaraná, modelados por Redes de Petri que possibilitou observar estatisticamente o comportamento do modelo nas etapas anteriores a sua implementação. A tecnologia Guaraná oferece um conjunto de ferramentas e metodologias que possibilita desenvolver modelos conceituais de soluções de integração utilizando uma linguagem DSL. O objetivo do trabalho é mostrar que o modelo conceitual desenvolvido no Guaraná pode ser traduzido em um modelo formal usando Redes de Petri Estocásticas e este pode ser simulado para obter informações sobre o desempenho da solução de integração, possíveis gargalos e formação de filas. Cada elemento do modelo conceitual do Guaraná é traduzido num grafo em Redes de Petri, sendo realizada a verificação do modelo da equivalência de ambos comparando o funcionamento do sistema de execução do Guaraná com o processo de execução de uma Rede de Petri. O estudo de caso é fundamentado num problema real de integração que propõe automatizar o processo de registro de novos usuários de um sistema, propõe integrar seis aplicações independentes, im-

plementadas em plataformas diferentes e em sua maioria desenvolvidas sem a preocupação com integrações futuras.

A simulação é abordada segundo Roos-Frantz et al. [36] como forma de minimizar custos e diminuir riscos de falas nos projetos de integração de aplicações com a proposta de analisar características da solução de integração numa fase anterior a seu desenvolvimento. O trabalho mostrou que os modelos conceituais constituídos com o Guaraná podem ser traduzidos em Redes de Petri Estocásticas e simulados, permitindo identificar suas características e comportamentos mediante os diferentes cenários. Os autores consideram que a abordagem sobre simulação que o trabalho propõe pode melhorar a qualidade de soluções de integração de aplicações empresariais desenvolvidas com Guaraná.

O trabalho proposto por Carvalho et al. [9] propõe um modelo de análise de desempenho de trabalho multifuncional em linhas de produção em forma de U usando o formalismo das Redes de Petri temporizadas. O formato das linhas de produção acarreta na possibilidade de mão-de-obra multifuncional, uma alternativa para empresas com pouca automação de processos, sua disposição acarreta num melhor aproveitamento da mão de obra, uma vez que possibilita aos operadores a possibilidade de desempenhar várias tarefas no mesmo ciclo produtivo deslocando-se entre postos de trabalho. A problemática abordada trata da relação de tempo de execução das tarefas nos postos de trabalhos, avaliando o tempo de cada operação que compõe a tarefa de determinado posto, a formação de estoques intermediários entre os postos de trabalho, e sua interferência no fluxo produtivo do sistema. Uma linha de produção é composta por diversos postos de trabalho, e partindo desta afirmação é desenvolvido um modelo em Redes de Petri temporizadas de um posto de trabalho genérico que representa as ações aplicadas neste posto, o recurso de serviço análogo ao operador e o estoque intermediário (estoque de produtos semiacabados). A partir deste é realizada a modelagem de uma linha de produção em forma de U com trabalho multifuncional, contendo oito postos de trabalho, sendo determinados três operadores para a linha. As Redes de Petri demonstraram que modelam facilmente as linhas de produção trabalho multifuncional.

A simulação o modelo de análise de linhas de produção em forma de U proposto por Carvalho et al. [9], considerando o tempo de execução de cada posto, sob um cenário que determina uma meta de construção de 67 peças num intervalo de tempo de 60 min. A investigação da diferença do tempo de execução de cada processo, pode resultar na formação dos estoques intermediários devido ao acúmulo de produtos semiacabados entre os postos de

trabalho e levar a ociosidade do trabalho, e por fim comprometer a meta de produção. Como alternativa de solução é determinada analiticamente carga máxima de cada estoque intermediário e a redistribuição dos operadores em nove setores. Com estas modificações foram observadas melhoras nas metas de produção nos tempos determinados.

Em seu trabalho Facchin e Sellitto [17] apresenta um modelo em Redes de Petri para medição do inventário e tempo de construção de uma manufatura em uma indústria calçadista, com objetivo de possibilitar a medição, antecipadamente, do inventário e o tempo necessário para produção de uma determinada encomenda. Conhecendo as grandezas antes da liberação do plano, um gestor pode antever e eventualmente prevenir problemas, mudando o plano. Para construção do modelo foi realizado um mapeamento dos processos, representando cada tarefa que os compõe, atribuídos os respectivos tempos de execução, coletadas por meio de medição mecânica, das transições e agrupadas em um modelo único. Para teste e refino do modelo, foi escolhido um plano já realizado, simulado em computador alimentado com a situação de carga inicial dos processos e com um plano de produção, foi informada a carga de cada processo, resultante das ordens anteriores, no momento de liberação do plano.

Os resultados obtidos por Facchin e Sellitto [17] nas simulações foram diferentes dos dados reais, sendo observado, por exemplo, o distanciamento de resultados simulados e reais e simulações que representam planos de fabricação longos. Por outro lado, a evolução discreta do sistema modelado e a sintaxe gráfica possibilita que análises sejam realizadas em todas as posições de manufatura, alternativas de gestão, podem ser simuladas e seus impactos avaliados. Verificou-se que Modificações momentâneas na estrutura da manufatura não são captadas individualmente pelo modelo de simulação, devido à simplificação de algumas atividades, negligencia algumas contingências e salienta a necessidade de trata-las de modo mais detalhado para uma possível contribuição para a resolução de problemas menos estruturados. Ao fim, realiza-se uma discussão na qual se explora como os resultados do método podem ser úteis em decisões de gestão, envolvendo o inventário admitido, restrições da manufatura. A logística interna de armazenagem e movimentação também pode ser apoiada pelo método.

O trabalho proposto por Morales Varela et al. [33] propõe um modelo em Redes de Petri para auxiliar na tomada de decisões operacionais e estruturais de uma indústria de pneus, a fim de minimizar desperdício de matéria-prima resultantes do processo de fabricação e avaliar a eficácia do processo. O modelo de simulação desenvolvido é baseado em um sistema de manufatura de

pneus em exercício na Cidade do México, um sistema que segue padrões rígidos de qualidade, e busca reduzir resíduos resultantes da produção e tornar o sistema mais sustentável. O formalismo das Redes de Petri possibilitou representar componentes, fluxos de informação, fluxo de materiais e remover os resíduos através da reconfiguração do sistema. A simulação possibilita uma visão ampla do processo, considera possíveis problemas e manutenção de máquinas. Desta forma, é possível identificar problemas que venham surgir, impactos gerados e propor melhorias tecnológicas, econômicas, sociais e políticas em seus processos de gestão e produção. O modelo é de simulação composto de transições estocásticas que representam possíveis falhas de equipamentos e peças defeituosas e transições temporizadas para representar o tempo em que os equipamentos permanecem sem operar.

As simulações realizadas por Morales Varela et al. [33] foram usadas para avaliar a eficiência do processo de produção através do indicador OEE (*Overall Equipment Effectiveness*), índice medidor de eficiência de equipamentos, simulação do modelo no *software Snoopy*, desenvolvido por pesquisadores da Universidade de Brandenburg [24]. A Validação do modelo foi realizada através de três parâmetros fundamentais (disponibilidade, desempenho e qualidade). Esses padrões são adquiridos por meio de medição do sistema real e são usados para calcular a eficiência produtiva através do indicador OEE, fornecendo os detalhes das causas da perda de produção, e é o ponto de partida para melhorar a produtividade. As Redes de Petri representaram os diferentes cenários apresentados em um sistema produtivo; identificaram as variáveis críticas, e entenderam o comportamento dos parâmetros do sistema quando suas variáveis são alteradas. Foi possível identificar pontos onde ocorrem operações que comprometem o desempenho final do sistema, com a opção de propor a implementação de métodos, ferramentas e técnicas para evitar a sinergia no sistema, dentre outros benefícios que a simulação, aliada às Redes de Petri trouxeram ao estudo deste sistema de manufatura, que podem ser usados como uma ferramenta que apoia o processo de estudo, análise e tomada de decisão estratégica para a melhoria do sistema, utilizando métodos, técnicas e ferramentas eficientes.

A grande maioria dos trabalhos utilizou do formalismo das redes para apoiar o processo de estudo, análise e tomada de decisões estratégicas para a melhoria dos sistemas modelados. O desenvolvimento dos modelos de simulação para representação do sistema na maioria foi baseada no estudo e observação do sistema real. Desta forma os trabalhos relacionados propõem comparar os resultados experimentais com os dados reais e validar o modelo proposto, agregar confiabilidade e utilizar o modelo para estimar o comportamento do sistema em diferentes situações e contribuir na sua otimi-

zação sem comprometer seu funcionamento. A pesquisa proposta propõe usar o formalismo das Redes de Petri Estocásticas para a modelagem de sistemas para propor a identificação de problemas e contribuir na melhoria do sistema modelado com o diferencial de atuar anterior a sua implementação, baseado em modelos conceituais e é proposta a verificação do modelo proposto por não possuir dados reais.

O trabalho proposto por Roos-Frantz et al. [36] apresenta o uso das Redes de Petri Estocásticas para simulação de modelos conceituais. A pesquisa proposta nesta dissertação também utilizou as Redes de Petri Estocásticas para a simulação de modelos conceituais de soluções de integração com o diferencial de simular um modelo conceitual diferente, sob cenário diferente, analisar um número maior de variáveis além de realizar a verificação do modelo de simulação proposto.

## 2.6 Resumo do Capítulo

A tecnologia Guaraná é uma tecnologia que possibilita desenvolver modelos conceituais de soluções de integração. Sua interface gráfica é fundada nos padrões documentados por Hohpe e Woolf [25]. A linguagem Guaraná DSL possui cinco elementos básicos: aplicações, processos, tarefas, *slots* e portas. A tecnologia Guaraná atua sob um sistema baseado em mensagens que determina que uma tarefa esteja apta a disparar quando houver ao menos uma mensagem em cada *slot* de entrada. Seu motor de execução propõe um sistema de execução de tarefas assíncrono.

A simulação é a imitação da operação de um processo do mundo real. O seu objetivo é compreender o funcionamento, prever o desempenho e o comportamento de um sistema. Os modelos são responsáveis por representar da forma mais fiel possível o sistema real para realizar a simulação. Os modelos podem representar um sistema de forma simbólica (ou icônico) estruturados com símbolos gráficos (retângulos e retas) com foco na descrição da mecânica de funcionamento, representar de forma analítica através de equações ou um conjunto destas e ainda representar por meio de um sistema de simulação. Em particular, os modelos de simulação podem ser representados simbolicamente, mas não possuem solução analítica, são "executados", ou seja, técnicas ou ferramentas possibilitam o modelo trabalhar da mesma forma que o sistema real estudado para obter informações do seu comportamento.

As Redes de Petri possibilitam modelar sistemas de eventos discretos, e oportunizam uma visualização de alta abstração e acompanhamento dinâmico das diferentes atividades do sistema que possuem características

predominantemente discretas. As Redes de Petri propõem uma modelagem de sistemas direta, fácil e rápida. As Redes de Petri possuem uma notação gráfica formada basicamente por lugares, arcos e transições. Também possui uma representação formal matemática que mapeia todos os elementos integrantes da rede e a conexão entre eles, resultando numa representação formal, sem ambiguidades.

A verificação e validação de um modelo são fundamentais para que ele resulte em informações precisas que não interferir no desenvolvimento do sistema real. Um Modelo se torna confiável quando é submetido a técnicas de Verificação e validação. A maioria das técnicas encontradas na literatura validam os modelos, por meio da utilização de dados reais confrontados com dados experimentais. A ausência de dados reais possibilita a verificação do modelo de simulação, que tem por objetivo assegurar a equivalência do modelo conceitual proposto com o modelo de simulação.

Os trabalhos relacionados tratam de trabalhos que envolvem a simulação de sistemas de eventos discretos modelados por Redes de Petri estocásticas. Os trabalhos abordados relatam o funcionamento dos sistemas modelados, características particulares de cada sistema e os resultados alcançados com a simulação. De maneira geral a modelagem e simulação por meio das Redes de Petri estocásticas possibilitou obter informações sobre o comportamento do sistema sob diferentes situações e identificar pontos críticos. Dessa forma foi possível testar alterações no modelo de simulação para verificar seu impacto para posteriormente transferi-las para o sistema real.



---

# Capítulo 3

## Modelagem

---

Não basta conquistar a sabedoria, é preciso usá-la.

Cícero

**E**ste capítulo apresenta a proposta de modelagem de um modelo conceitual de solução de integração de um problema de integração real desenvolvido na tecnologia Guaraná. A Seção §3.1 verifica a equivalência dos componentes do Guaraná e das Redes de Petri e propõe-se a tradução dos principais elementos do Guaraná em elementos das Redes de Petri. Posteriormente a Seção §3.2 apresenta o caso de estudo que consiste de um modelo conceitual da solução de integração. Na Seção §3.3 desenvolve-se um modelo formal de simulação em Redes de Petri equivalente ao modelo conceitual e por fim apresenta-se sua formulação matemática. A Seção §4.4 encerra o capítulo apresentando um resumo dos temas abordados.

### 3.1 Equivalência entre Guaraná DSL e Redes de Petri

A tradução de modelos conceituais de solução de integração para modelos de Redes de Petri é possível pela analogia entre seus componentes, como mostra a Tabela §3.1, e pela semelhança entre o modelo de execução do Guaraná e o das Redes de Petri. Os elementos principais dos modelos conceituais da tecnologia Guaraná são as tarefas, os *slots* e as mensagens, os quais são equivalentes, respectivamente, às transições, lugares e *tokens*, principais componentes das Redes de Petri.

Guaraná DSL	RdP
 tarefa	 transição
 slot	 lugar
 mensagem	 token

Tabela 3.1: Comparativo dos Elementos do Guaraná e Redes de Petri

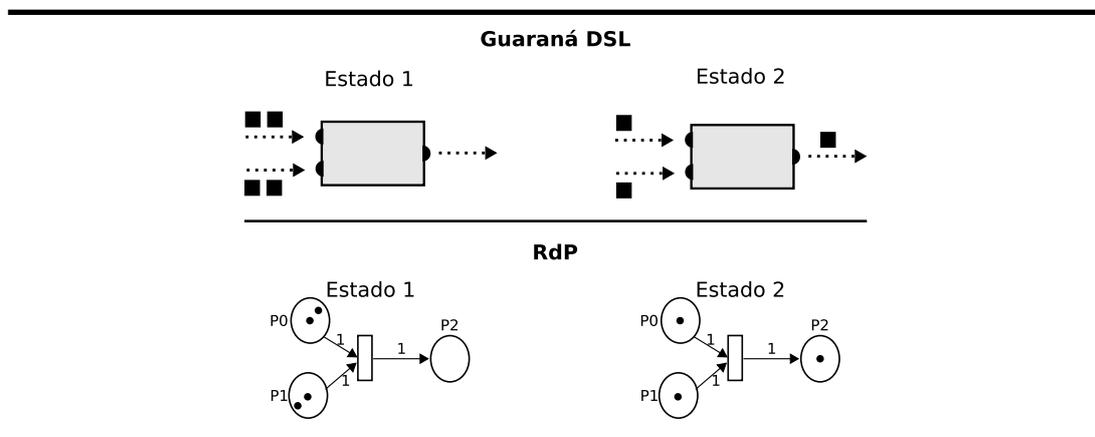


Figura 3.1: Equivalência entre troca de estados

Soluções de integração são compostas basicamente por tarefas. Os *slots* são responsáveis pela comunicação entre as tarefas, armazenam as mensagens já processadas da tarefa anterior para processamento da tarefa posterior e determinam o fluxo de mensagens. Nas Redes de Petri, os lugares junto com os arcos, têm função semelhante às tarefas e slots. Os arcos fazem a ligação entre lugares e transições e indicam o sentido do fluxo de mensagens e não possuem valor funcional. A analogia entre uma troca de estado em uma solução modelada com Guaraná e uma troca de estado em uma Redes de Petri está representada na Figura 3.1.

### 3.1.1 Representação das Tarefas em Rede de Petri

A seguir apresentam-se as principais tarefas que integram o DSL do Guaraná e seu grafo equivalente em Redes de Petri, organizadas em uma tabela, divididas de acordo com o grupo a qual pertencem.

Na Tabela §3.2 apresenta-se o grupo das tarefas Modificadoras, Roteadoras, Transformadoras e Temporizadoras, destacando sua notação gráfica no Guaraná e os grafos equivalentes em em Redes de Petri e o nome da tarefa.

## 3.2 Estudo de Caso

O Problema de integração que será objeto de estudo desta dissertação foi apresentado por Frantz [22], um caso de estudo real que consiste em uma solução de integração não trivial com base em um projeto que visa melhorar a funcionalidade da aplicação Central Telefônica na Universidade UNIJUÍ. O objetivo da solução de integração é automatizar a cobrança de telefonemas pessoais que tenham custos realizados pelos funcionários com uso de linhas telefônicas da Universidade.

### 3.2.1 Ecossistema de Software

A solução de integração proposta envolve cinco aplicações: Central Telefônica, Recursos Humanos, Folha de Pagamento, Servidor de E-mail e Notificador de Mensagens. Os aplicativos são executados em plataformas diferentes. O sistema de recursos humanos e o sistema de folha de pagamento são sistemas legados desenvolvidos na própria universidade. As demais aplicações são *softwares* comprados prontos pela instituição. Além disso, o Servidor de Email oferece uma interface de acesso para enviar e-mails quando recebe comandos da solução de integração.

A identificação do funcionário que realiza a chamada é feita por meio de uma senha pessoal que é inserida antes do início da chamada. A senha pessoal é utilizada para correlacionar as chamadas telefônicas com dados do sistema de recursos humanos e da folha de pagamento. Do sistema de recursos humanos são adquiridas as informações pessoais dos funcionários. Mensalmente o sistema de folha de pagamento calcula a remuneração de todos os funcionários, e inclui salários, bônus e deduções. As notificações são enviadas pelo servidor de Email e pelo Notificador via SMS.

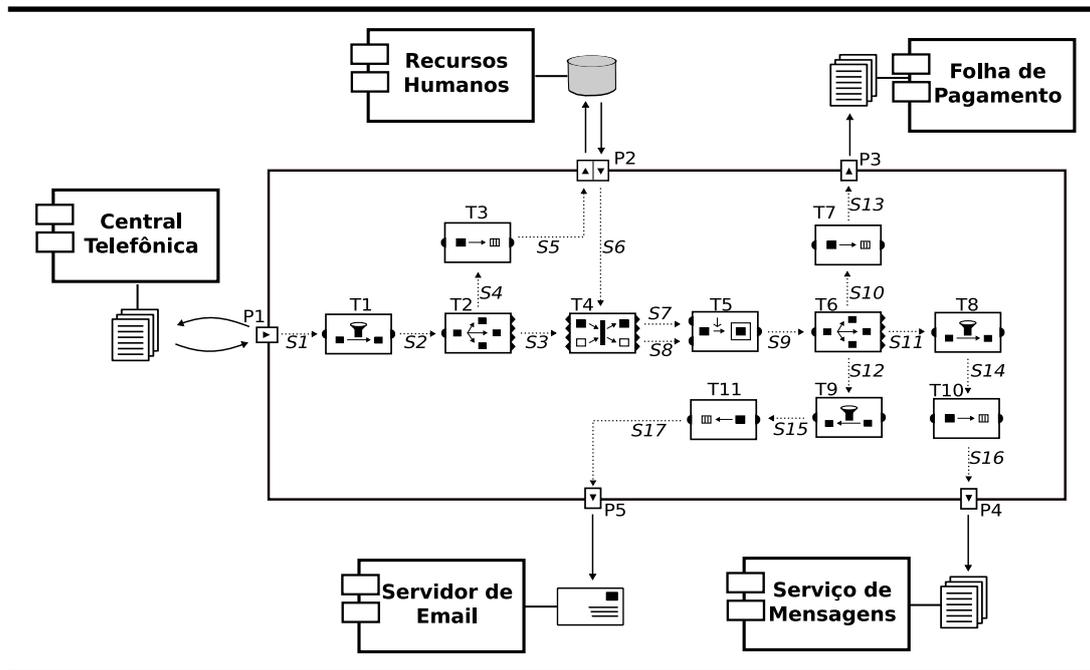
A Central Telefônica registra todos os telefonemas. Os funcionários são notificados por e-mail e por mensagem SMS, informando o valor das chamadas realizadas, de modo a torna-los cientes que a dedução aparecerá em sua folha de pagamento.

### 3.2.2 Solução de Integração

O modelo conceitual da solução apresenta um processo de orquestração exógeno, que coordena os aplicativos que compõem a solução de integra-

Grupo	Nome da Tarefa	Guaraná	RdP
Modificadoras	Slimmer		
	Context-based Slimmer		
	Content Enricher		
	Context-based Content Enricher		
	Header Enricher		
	Context-based Header Enricher		
	Header Promoter		
	Header Demoter		
	Set Correlation ID		
Roteadoras	Set Return Address		
	Correlator		
	Merger		
	Resequencer		
	Filter		
	Idempotent Transfer		
	Dispatcher		
	Distributor		
	Replicator		
Transformadoras	Semantic Validator		
	Threader		
	Translator		
	Splitter		
	Aggregator		
	Chopper		
	Assembler		
	Cross Builder		
Temporizadoras	Delayer		
	Ticker		
	Expire Checker		

Tabela 3.2: Tarefas e seus grafos equivalentes em Redes de Petri



**Figura 3.2:** Modelo Conceitual de Solução de Integração Proposto por Frantz [22]

ção, conforme está representado na Figura 3.2. A troca de mensagens realiza a comunicação entre os aplicativos através das portas. Uma porta de solicitação dá acesso direto ao banco de dados do sistema de Recursos Humanos. Para solucionar o problema da diferença de formato para troca de mensagens são usadas tarefas Tradutor para traduzir as mensagens da solução de integração em formatos que os aplicativos integrados trabalham.

O início do fluxo de trabalho da solução de integração dá-se na porta P1, que faz buscas constantes de novas chamadas na Central Telefônica. Uma nova chamada telefônica resulta em uma mensagem que é inscrita no slot S1. O S1 encaminha a mensagem para a tarefa T1 (filtro) que tem a função de retirar do fluxo mensagens geradas por ligações sem custo. Em seguida a tarefa T2 (replicador) faz duas cópias da mensagem: uma das cópias é enviada tarefa T3 (transformador) que altera o formato da mensagem e a envia para porta P2 busca informações do realizador da chamada no sistema de Recursos Humanos. A mensagem que retorna do sistema de recursos humanos é processada juntamente com a segunda cópia pela tarefa T4 (correlacionador) e encaminhada para a tarefa T5 que adiciona à mensagem as informações fornecidas. Posteriormente a nova mensagem é replicada três vezes na

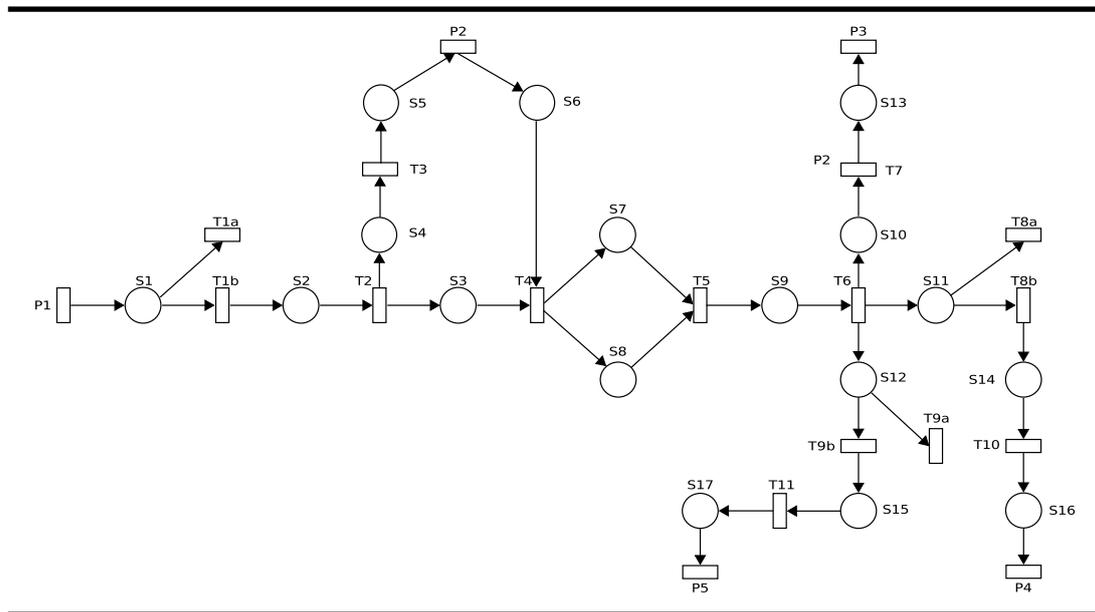
tarefa T5, e suas cópias serão encaminhadas ao sistema de Folha de Pagamento, ao Servidor de E-mail, e o Notificador de Mensagens. Uma das cópias é enviada para tarefa T7 que altera o formato da mensagem e a envia em direção à porta P3 que será responsável por enviar ordens de débito ao sistema de Folha de Pagamento. A segunda cópia é encaminhada para a tarefa T9 (filtro) que verifica se há informações referentes ao e-mail, havendo a informação a mensagem é enviada para tarefa T11 que altera o formato da mensagem e a encaminha para porta P5 para ser enviada ao servidor de e-mail. A Terceira cópia é encaminhada para a tarefa T8 (filtro) que verifica se há informações referentes ao número de celular, havendo a informação a mensagem é enviada para tarefa T10 que altera o formato da mensagem e a encaminha para porta P4 para ser enviada ao servidor de mensagens.

### 3.3 Modelo de Simulação

A Figura §3.3 apresenta o modelo de simulação construído para representar a solução de integração apresentada no caso de estudo. A transição P1 modela a porta P1 onde inicia o fluxo de trabalho, transições P3, P4 e P5 representam as portas de saída para as aplicações Folha de Pagamento, Serviço de Mensagens e Serviço de Email, respectivamente, e a porta P2 modela a porta de solicitação a aplicação de Recursos Humanos. O modelo de simulação foi desenvolvido seguindo a organização e ligação das tarefas, a disposição das portas e a direção do fluxo de mensagens.

A Tabela §3.3 apresenta as tarefas que compõem o caso de estudo, seu grafo equivalente em Redes de Petri e breve descrição da sua funcionalidade. Os nomes das tarefas foram traduzidos para facilitar o entendimento das suas funcionalidades.

A funcionalidade de cada tarefa é mantida ao realizar a transcrição das tarefas da tecnologia Guaraná para as Redes de Petri. A Porta de Entrada tem a função de inserir novas mensagens para o processamento na solução de integração, sua representação em Redes de Petri equivale à transição P1 que quando disparada insere um *token* no lugar S1, que representa o *slot* S1. A tarefa Porta de Saída atua de forma contrária a Porta de entrada, enviando mensagens para as aplicações integradas, seu grafo equivalente em Redes de Petri indica fluxo de *tokens* para fora do processo. A tarefa Transformador tem a função de adaptar a mensagem a um formato que a aplicação seguinte entenda, como as mensagens são equivalentes aos *tokens*, a tarefa é representada por um *slot* de entrada uma transição e um *slot* de saída que representam entrada, processamento e saída da tarefa, respectivamente. A



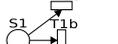
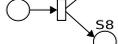
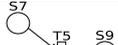
**Figura 3.3:** Modelo de Simulação

tarefa Filtro tem a função de remover mensagens do fluxo da solução de integração de acordo com a política de filtragem definida. Na Rede de Petri, a funcionalidade do Filtro foi preservada, de maneira que *tokens* são removidos do fluxo se a transição T1a disparar e seguem no fluxo se a transição T1b disparar. Porém, a política de filtragem não foi traduzida.

A tarefa Correlacionador é de grande importância dentro do fluxo de mensagens, pois pode ser usada para localizar no fluxo mensagens correlacionadas que devem ser processadas em conjunto. Esta tarefa é representada pela transição T4, que só dispara quando houver *tokens* nos lugares S3 e S6, e projeta conjuntamente *tokens* para os lugares S8 e S7. A tarefa Enriquecedor de Conteúdo recebe mensagens correlacionadas e as combina em uma única, como representa o grafo equivalente em Redes de Petri, a transição T5 dispara quando houver ao menos um *token* nos lugares S7 e S8, e é colocado apenas um *token* no lugar S9, representando a funcionalidade da tarefa.

### 3.3.1 Formulação Matemática

A representação formal para o modelo de simulação é expressa pela tupla da Equação 3.1.  $S$  representa o conjunto de lugares conforme a Equação 3.2.  $T$  representa o conjunto das transições de acordo com a Equação 3.3. A representação o mapeamento dos arcos conforme Equação 3.4 e posteriormente

Guaraná	Nome do Grafo	Funcionalidade	RdP
	Porta de Entrada	Insere mensagens na solução de integração	
	Porta de Saída	Envia mensagens a uma aplicação integrada	
	Porta de Solicitação	Solicita informações à uma aplicação integrada	
	Transformador	Adequa formato da mensagem	
	Filtro	Remove mensagens do fluxo ou permite a sua passagem	
	Correlacionador	Processa juntamente mensagens com alguma relação	
	Enriquecedor de Conteúdo	Adiciona conteúdo a uma mensagem	
	Replicador	Faz cópias idênticas de uma mensagem	

**Tabela 3.3:** Elementos do Caso de Estudo e Seus Grafos Equivalentes em Rede de Petri

subdividido nos subconjuntos  $A_e$  e  $A_s$  que representam respectivamente arcos de entrada e arcos de saída. O conjunto  $T_D$  descrito na Equação §3.5 representa o valor das taxas de disparo das transições respectivo a ordem do conjunto  $T$  na Equação §3.3. O conjunto dos valores dos arcos foi suprimido da representação devido ao fato de todos os arcos terem valor igual a um e não interferir na descrição formal do modelo.

$$MdS = (S, T, A, T_D) \quad (3.1)$$

$$S = (S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17) \quad (3.2)$$

$$T = (P1, P2, P3, P4, P5, T1a, T1b, T2, T3, T4, T5, T6, T7, T8a, T8b, T9a, T9b, T10, T11) \quad (3.3)$$

$$\begin{aligned}
A = & ((P1, S1), (S1, T1a), (S1, T1b), (T1b, S2), (S2, T2), (T2, S4), \\
& (T2, S3), (S4, T3), (S3, T4), (T3, S5), (S5, P2), (P2, S6), (S6, T4), \\
& (T4, S7), (T4, S8), (S7, T5), (S8, T5), (T5, S9), (S9, T6), (T6, S10), \\
& (T6, S11), (T6, S12), (S10, T7), (T7, S13), (S13, P3), (S11, T8a), \\
& (S11, T8b), (T8b, S14), (S14, T10), (T10, S16), (S16, P4), (S12, T9a), \\
& (S12, T9b), (T9b, S15), (S15, T11), (T11, S17), (S17, P5)) \quad (3.4)
\end{aligned}$$

$$T_D = (1, 1, 1, 1, 1, 0.05, 1, 1, 1, 1, 1, 1, 1, 0.05, 1, 0.01, 1, 1, 1) \quad (3.5)$$

A atribuição das taxas de disparos para as transições é realizada de acordo com uma estimativa do comportamento da aplicação Central Telefônica. As taxas de disparo significam a porcentagem de disparo, mas são representadas e implementadas em simuladores na forma decimal. A transição T1a faz parte da representação da funcionalidade da tarefa Filtro, e seus disparos representam ligações que não contém custo e foram removidas do fluxo. Em média 5% das ligações não tem custo. Em torno de 5% dos usuários não tem número de celular cadastrado para envio de mensagens de notificação e desta maneira atribui-se a transição T8a. Por outro lado há um número menor de usuários que não possuem um e-mail cadastrado, algo em torno de 1% que então é atribuído a transição T9a. As demais transições não têm restrições de disparo e por isso sua taxa de disparo é igual a 100%.

A notação matricial representa formalmente o modelo usando matriz dos arcos de entrada  $A_e$ , matriz dos arcos de saída  $A_s$  e o conjunto das taxas de disparo. A notação matricial faz um mapeamento dos arcos de entrada e saída e a conexão entre eles. O número de linhas é equivalente ao número de lugares e o número de colunas ao de transições. Desta forma as matrizes  $A_e$  e  $A_s$  terão ordem de  $19 \times 17$ . Os pesos dos arcos do modelo de simulação são todos iguais a 1, sendo assim representado quando houver ligação entre a transição e o lugar. A matriz que representa os arcos de entrada do modelo de simulação está representada na Equação §3.6 e a matriz





A representação formal e matricial são formas de representar o modelo gráfico de forma única, evitando mais de uma interpretação do modelo.

### 3.4 Resumo do Capítulo

Neste capítulo foi realizada a modelagem formal do modelo conceitual da solução de integração utilizando Redes de Petri Estocásticas. Para realizar a transcrição verificou-se a analogia entre seus componentes e a semelhança entre o modelo de execução do Guaraná e o das Redes de Petri. Dessa maneira foram traduzidos os principais blocos que integram o DSL do Guaraná para Redes de Petri.

A funcionalidade de cada tarefa é mantida ao realizar a transcrição da tecnologia Guaraná para as Redes de Petri, uma vez que as tarefas possuem uma semântica implementada que não pode ser traduzida. São descritas as funcionalidades das tarefas que integram o modelo conceitual e como o seu grafo equivalente em Rede de Petri mantém essa funcionalidade. Na sequência foi proposto o modelo de simulação por meio da sua formulação matemática que conta com representação gráfica e formal.

---

# Capítulo 4

## Experimento e Análise dos Resultados

---

*Sempre parece impossível até que seja feito.*

*Nelson Mandela*

**N**este capítulo é apresentado o experimento de simulação do modelo proposto. A Seção §4.1 descreve o cenário utilizado para a simulação, as variáveis observadas, a ferramenta de simulação utilizada e os resultados experimentais. Na Seção §4.2 é realizada a análise e interpretação dos resultados obtidos a partir da simulação da solução de integração modelada. A Seção §4.3 apresenta a verificação do modelo de simulação proposto. A Seção §4.4 encerra o capítulo apresentando um resumo dos temas abordados.

### 4.1 Experimento

O modelo de simulação foi implementado na ferramenta de simulação PIPE e executado 25 vezes para excluir qualquer discrepância nos dados. O cenário no qual o experimento foi realizado considera uma carga de entrada de 10000 mensagens.

A determinação do número de repetições do experimento segue a especificação de Grinstead e Snell [23], a qual determina que experimentos desta característica devem apresentar em torno de 25 repetições. As repetições tem por objetivo eliminar qualquer anormalidade nos resultados, uma

vez que se trata de um processo estocástico. A ferramenta de simulação apresenta a média de tokens acumulados em cada lugar, e o mecanismo de disparo leva em conta modelos probabilísticos.

Para se determinar a carga de entrada foram realizadas simulações com cargas de 1.000, 10.000, 20.000, 30.000, 40.000, 50.000, 60.000, 70.000, 80.000, 90.000 e 100.000 mensagens, visando analisar o acúmulo de *tokens* nos lugares. Esperava-se, com o aumento gradativo da carga de entrada, um acúmulo de *tokens* seguindo a mesma proporção. Analisando o comportamento do modelo diante as diferentes entradas, observou-se uma oscilação no acúmulo de *tokens*, e não um aumento gradativo no acúmulo como esperado. Desta forma, optou-se por analisar o comportamento do modelo diante a uma carga de entrada. A definição da melhor carga de entrada foi realizada calculando a média de *tokens* acumulados em cada lugar em relação as 11 situações e estimado um acúmulo médio para cada lugar, e estimar um comportamento médio do modelo. A Correlação de Pearson [34, 39] mede o grau da correlação entre duas variáveis ou dois conjuntos de dados, neste caso é usada para comparar qual carga de entrada apresenta acúmulo de *tokens* com correlação mais forte ao comportamento médio do modelo. A simulação que considerou carga de entrada de 10.000 mensagens obteve acúmulo de *tokens* com correlação mais forte em relação ao acúmulo do comportamento médio do sistema e por este motivo a análise de comportamento do sistema incidirá sobre a referida carga de entrada.

#### 4.1.1 Variáveis Observadas

A identificação de gargalos de desempenho em uma solução de integração é possível pela análise do comportamento dos seus componentes, que resultam em características importantes da solução e possibilita estimar possíveis problemas. Na Tabela §4.1 são apresentas algumas variáveis que podem ser extraídas dos resultados experimentais. Essas variáveis serão importantes na compreensão e análise do comportamo de uma solução de integração.

O acúmulo de *tokens* nos lugares representa a formação de filas nos *slots* da solução de integração. As filas são determinadas pela quantidade de mensagens que se agrupam nos *slots*. Sua análise possibilita uma visualização do estado geral do sistema, sendo a formação de filas muito elevadas em determinados *slots* certamente um indicador de que a tarefa esta tendo dificuldade de processar a demanda de mensagens exigidas.

O número de disparos das transições do modelo de simulação representam o número de execuções das tarefas do modelo conceitual. Os modelos

Variável	Abreviação
Tokens Acumuladas nos Lugares	FI
Número de Execuções Realizadas por Cada Transição	NE
Frequência de Disparo das Transições	TFM
Tempo Médio de Espera de Uma Transição Apta a Executar	TFT

**Tabela 4.1:** Variáveis Observadas

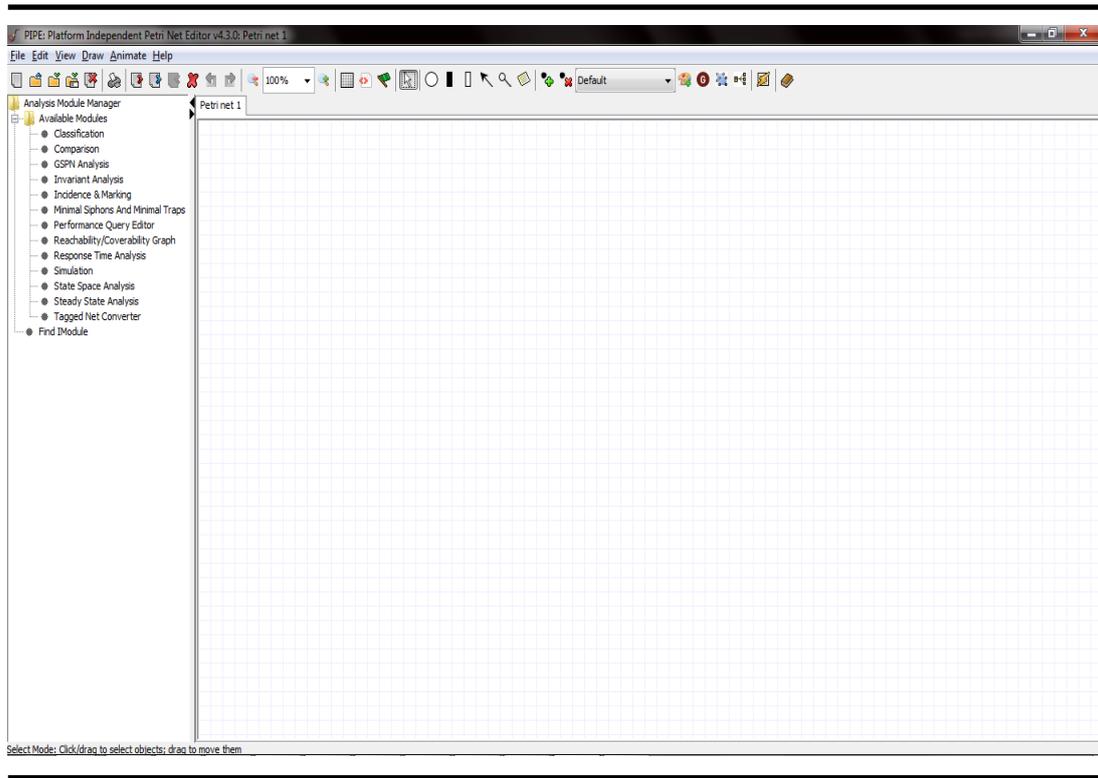
conceituais são considerados SEDs, logo o tempo interfere no seu mecanismo. O caso de estudo abordado considera apenas uma *thread* para a execução da solução, desta forma uma tarefa é executada por vez. Partindo da classificação dos modelos conceituais como SEDs, e considerando apenas uma unidade de processamento para uma solução de integração, é possível determinar quantos processamentos uma determinada tarefa realizou durante a simulação.

A frequência de disparo representa o percentual de disparo de cada tarefa em relação ao total de disparos. O tempo médio de espera de uma transição apta a disparar no modelo de simulação representa o tempo médio de espera de uma determinada tarefa para executar.

#### 4.1.2 Ferramenta de Simulação

Atualmente, existe uma grande variedade de simuladores de Redes de Petri, desenvolvidos em diferentes plataformas. A ferramenta de simulação utilizada para a análise do modelo foi o *Platform Independent Petri net Editor 2* (PIPE2) [1]. PIPE2 é uma plataforma de código aberto para criação e análise de Redes de Petri Generalizadas e Estocásticas, com interface de fácil utilização, conta com muitos módulos de análise, animação dos projetos e possibilita exportação de redes para arquivos PNG [1].

PIPE2 possui base java, desenvolvido por pesquisadores do *Imperial College de Londres*, por meio de um projeto de pesquisa nomeado "The Platform Independent Petri net Editor PIPE". O projeto teve início entre 2002 e 2003 e mantém-se em funcionamento e aperfeiçoamento com lançamento de versões atualizadas, disponibilizadas para download no endereço <http://sourceforge.net/projects/pipe2/> [16].

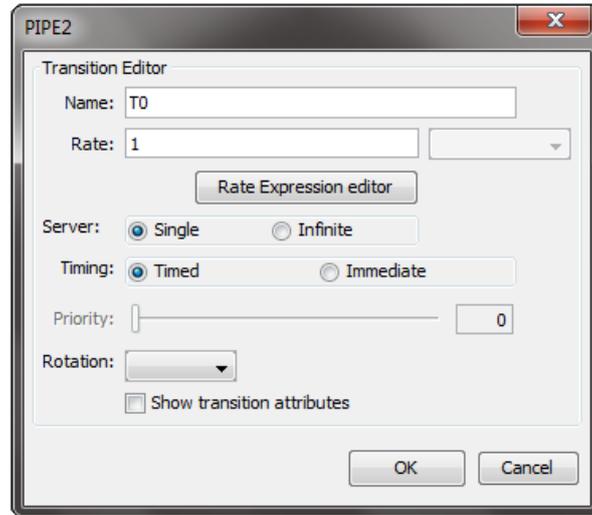


**Figura 4.1:** Interface Gráfica PIPE2

Na Figura 4.1 é ilustrada a interface gráfica do PIPE2 para a criação e edição de modelos Redes de Petri. Os modelos são criados em uma tela usando componentes a partir de uma barra de ferramentas de desenho, incluindo lugares, transições, arcos e *tokens*. Além da funcionalidade básica de design do modelo, a interface do designer oferece recursos como zoom, exportação, edição por abas e animação. O modo de animação é particularmente útil para a verificação intuitiva do comportamento dos modelos. Na aba localizada na esquerda da interface apresenta os módulos de análise das Redes de Petri.

As Redes de Petri Clássicas são representadas pelo PIPE2 como retângulos com preenchimento preto e as Redes de Petri Estocásticas como Retângulos com preenchimento branco. As taxa de disparo referente a cada transição estocástica são editadas através da aba *Transition Editor*, como apresenta a Figura 4.2, na aba *Rate* é determinada a taxa de disparo referente a transição, os marcadores *Single* e *Timed* representam, respectivamente, um único servidor para o sistema e transição temporizada.

As transições estocásticas do PIPE2 são sensibilizadas com *Rate* igual 1



**Figura 4.2:** Aba de Sensibilização das Transições Estocásticas do PIPE2

que representa a taxa porcentual, o que significa que as transições são definidas com probabilidade de disparo 100%. Desta forma, só é necessário alterar transições que tenham taxa de disparo inferior.

O módulo *Simulation* é usado para analisar o modelo de simulação proposto. O módulo de simulação de redes apresenta duas abas de edição dos parâmetros de simulação, *Firings*(disparos) e *Replications*(repetições), onde serão implementados os cenários propostos.

Como já abordado nas seções anteriores, as Redes de Petri são executadas através dos disparos de suas transições aptas. Cada disparo de uma transição equivale a um processamento realizado por uma tarefa ou componente da solução de integração. O cenário para o experimento foi pensado considerando número de mensagens de entrada, e é preciso adaptá-lo para que o simulador interprete-os como disparos. Analisando o modelo de simulação da Figura §3.3 expresso na Página 53, um *token* gerado por um disparo da transição P1, depende de outros 15 disparos para concluir todo o processo, totalizando 16 disparos, se não for filtrado. Então, para cada mensagem que se deseja gerar na entrada são necessários 16 disparos. Desta forma, o cenário proposto de 10.000 mensagens é multiplicado por 16, representando 16 disparos para concluir o processamento de cada mensagem, totalizando 160.000 disparos.

### 4.1.3 Resultados Experimentais

O modelo de simulação foi inicialmente submetido ao cenário de simulação para avaliar seu comportamento perante carga de entrada. Os resultados são expressos por meio de tabelas e gráficos juntamente com a interpretação dos mesmos.

A Tabela §4.2 mostra as informações do modelo de simulação e dados do experimento divididas em 7 colunas que apresentam o comportamento do modelo perante as condições impostas nas simulações.

Na coluna 1, são mostradas todas as transições que compõem o modelo organizadas em ordem alfabética. As linhas de 1 a 5 representam as portas do modelo conceitual, as demais as tarefas. A última linha é utilizada para apresentar a soma total dos valores de cada coluna.

Na coluna 2, estão descritas as taxas de disparos das respectivas transições. Na coluna 3 estão as probabilidades globais de disparo para cada transição, que é obtida pela razão da taxa de disparo individual com a soma das taxas de disparo. A probabilidade global possibilita calcular o valor esperado de disparos para cada transição pela multiplicação da probabilidade global pelo número total de disparos do experimento, como descrito na coluna 4. A soma dos disparos esperados é igual ao total de disparos implementados no sistema como mostra a última linha da coluna 4, com soma sendo igual a 160.000 disparos.

Na coluna 5, mostra-se o número de disparos realizados de cada transição no experimento realizado. É possível verificar que o total de disparos realizados é igual ao número de disparos total implementado no experimento. Na coluna 6, mostra-se a frequência de disparos das transições no experimento, obtida pela razão do número de disparos da transição pelo total de disparos implementado na simulação. A frequência representa o percentual de disparos da referida tarefa em relação ao número total de disparos. Na coluna 7, determina-se o atraso médio de espera de uma tarefa apta a disparar, quantificado em disparos.

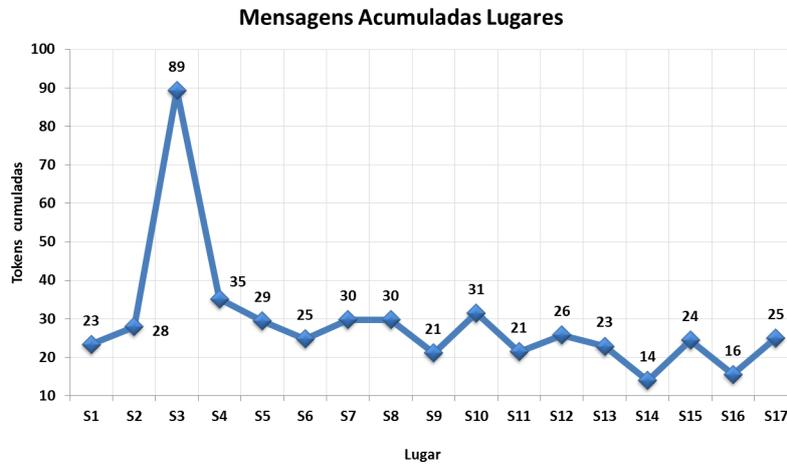
No Gráfico da Figura §4.3 é mostrado o acúmulo de *tokens* nos lugares do modelo de simulação após a simulação. O eixo y representa o número de *tokens* acumulados e o eixo x os lugares que compõem o modelo.

Transição	Taxa de Disparo	Probabilidade Global	Expectativa de Disparos	Disparos Realizados	Frequência de Disparo	Tempo Médio de Espera para Disparo
P1	1	0,062073246	9931,719	10706,40	6,69%	14,944
P2	1	0,062073246	9931,719	10052,02	6,28%	15,917
P3	1	0,062073246	9931,719	9921,91	6,20%	16,126
P4	1	0,062073246	9931,719	9472,67	5,92%	16,891
P5	1	0,062073246	9931,719	9800,76	6,13%	16,325
T1a	0,05	0,003103662	496,586	538,49	0,34%	297,127
T1b	1	0,062073246	9931,719	10144,53	6,34%	15,772
T2	1	0,062073246	9931,719	10116,62	6,32%	15,816
T3	1	0,062073246	9931,719	10081,43	6,30%	15,871
T4	1	0,062073246	9931,719	10027,23	6,27%	15,957
T5	1	0,062073246	9931,719	9997,47	6,25%	16,004
T6	1	0,062073246	9931,719	9976,12	6,24%	16,038
T7	1	0,062073246	9931,719	9944,81	6,22%	16,089
T8a	0,05	0,003103662	496,586	452,67	0,28%	353,459
T8b	1	0,062073246	9931,719	9502,21	5,94%	16,838
T9a	0,01	0,000620732	99,317	100,34	0,06%	1594,625
T9b	1	0,062073246	9931,719	9850,20	6,16%	16,243
T10	1	0,062073246	9931,719	9488,19	5,93%	16,863
T11	1	0,062073246	9931,719	9825,66	6,14%	16,284
TOTAL	16,11	1	160000	160000	100%	---

Tabela 4.2: Tabela de Resultados Experimentais

## 4.2 Análise dos Resultados

A análise dos resultados experimentais apresentou pequenas diferenças entre valores de disparos esperados e realizados, o que era esperado uma vez que seu mecanismo de seleção de disparos prevê seleção aleatória de transições. A entrada no modelo de simulação de mais *tokens* que o esperado é identificada observando o número de disparos da transição P1 a qual representa a porta de entrada do modelo conceitual. Verificando os valores de disparos das transições P3, P4 e P5 observa-se uma diferença clara entre eles e entre os valores da transição P1, o que significa que saíram menos *tokens* do modelo do que entraram e com isso houve acúmulo de *tokens* nos lugares, que por sua vez representa a formação de filas no modelo conceitual. Como já abordado em seções anteriores, a formação de filas é uma característica dos sistemas orientados a mensagens, mas uma fila pode se tor-



**Figura 4.3:** Gráfico da Formação de Filas nos Slots

nar um gargalo quando apresenta um acúmulo desproporcional ao estado geral do sistema. Para uma melhor análise da formação das filas, os lugares pertencentes ao modelo de simulação e o respectivo número de *tokens* acumulados ao final do experimento são organizados num gráfico de linhas.

Analisando o gráfico da Figura 4.3 é possível constatar que o lugar S3 mostrou acúmulo elevado, diante de um comportamento praticamente uniforme dos demais, acumulando 89 *tokens*. O lugar S3 juntamente com o lugar S6 alimenta a transição T4. O acúmulo médio no lugar S3 é em média 3,5 vezes maior que os demais e representa 18% das filas do modelo. Dessa forma, considera-se que na solução de integração o *slot* S3, nas condições do experimento, teria acúmulo de mensagens bem superior aos demais *slots*. Com isso a tarefa T4 se torna um possível gargalo.

Realizando a Correlação de Pearson [34, 39] de modo a correlacionar os disparos esperados e os disparos realizados no experimento, obtêm-se  $\rho = 0,9971$ , o que representa uma forte correlação entre os dois conjuntos de dados. Verifica-se que as transições dispararam conforme a expectativa de disparos seguindo a aleatoriedade prevista no mecanismo de disparos das redes. De acordo com o processamento assíncrono de mensagens do motor de execução da Tecnologia Guaraná é possível assegurar que haverá uma variação do número de execuções entre as tarefas de uma solução de integração e na formação de filas entre os *slots*.

A estimativa da frequência de disparo das transições foi decisiva para esti-

mar o atraso médio de uma transição apta a disparar. A espera representa o atraso das Redes de Petri Estocásticas e usa a unidade disparos. Pode-se definir que cada transição tem um tempo de espera médio entre seus disparos, usando como unidade de tempo disparos. A análise do tempo de disparo mostra que a transição T1 tem um tempo de espera menor que as demais, o que reflete no seu número de execuções maior que as demais, da mesma forma que as transições que possuem baixa taxa de disparo têm um atraso bem elevado. Em tese um *token* depende de 16 disparos para percorrer todo o modelo se não for filtrado, o que determina que cada tarefa teria atraso de 16 disparos. A média geral do atraso médio das transições do experimento, desconsiderando as transições com função de filtragem (T1a, T8a e T9a), é de 16,12 disparos. As transições com a função de filtragem são desconsideradas no cálculo da média pelo fato ter uma taxa de disparo muito baixa e atraso muito elevado e pelo fato de que é estimado o atraso médio considerando que a mensagem não é filtrada. A análise do tempo de espera das transições revela que em média as transições tem um atraso muito semelhante. Dessa forma o tempo de espera de uma tarefa, de acordo com esse experimento, não interfere na formação de filas, pois o tempo médio de espera e o tempo real de uma tarefa são muito próximos e não chegam a interferir na formação de gargalos.

### 4.3 Verificação do Modelo de Simulação

Seguindo a técnica de Kleijen [28] para verificação do modelo, a expectativa de comportamento do sistema é determinada por um especialista da área. 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 haja acúmulo de mensagens. A tarefa Correlacionador (T4) é alimentada pelos *Slots* S3 e S6. O *Slot* S3 é alimentado pela tarefa Replicador (T2), já S6 depende do processamento da Tarefa T3 e da Porta P2. Dessa forma, espera-se que haja um maior acúmulo de mensagens em S3, o qual armazena mensagens para serem correlacionadas com as mensagens de S6 pela Tarefa T4. A característica da tarefa T4 é correlacionar as mensagens, neste caso, torna S3 um suposto gargalo de desempenho. O experimento realizado com o modelo de simulação demonstrou que ao final da simulação gerou-se um acúmulo de *Tokens* nos seus lugares. O lugar S3 acumulou um número de mensagens cerca de três vezes maior que os demais. A transição T4 é alimentada pelos Lugares S3 e S6, sendo que para cada disparo que imprime um *Token* em S3, são necessários três para imprimir um em S6, sendo a taxa de disparo de T4 igual à maioria das tarefas, a tarefa T2 acaba inserindo mais *Tokens* em S3 em relação àqueles que chegam em S6.

A comparação das características esperadas com as encontradas nos experimentos, seguindo a proposta de Kleijen [28], 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 um sistema orientado a mensagem. A formação de fila mais expressiva era esperada nos Slots que alimentam a Tarefa T4, neste caso o Slot S3, o que se constatou quando houve acúmulo nos Lugares adjacentes da transição T4, neste caso o Lugar S3. De fato, o acúmulo no Lugar S3 foi em média maior que os demais, comportamento que era esperado para o Slot S3. A comparação entre o comportamento esperado pelos especialistas da área e o experimento desenvolvido deixa evidente que há semelhança entre comportamentos.

A proposta de verificação do modelo de acordo com a perspectiva da técnica de Sargent [37] é aplicada ao modelo de simulação. Para a aplicação da técnica é proposto um experimento de 16000 disparos. Para facilitar o entendimento da aplicação da técnica, o modelo de simulação foi avaliado em setores e foram colocados medidores de disparos em determinados pontos do modelo representados com linha pontilhada, como demonstra a Figura §4.4. Este 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.

Os setores do sistema estão representados por cores, e a contagem de mensagens partirá das portas de saída em direção à porta P1. Na Figura §4.4 os marcadores de disparos estão representados com linha pontilhada na cor preta e no interior o número de disparos que realizou. Os lugares apresentam em seu interior o número de mensagens que ficaram acumuladas ao final do experimento. Para melhor organização e compreensão da contagem das mensagens, estas foram organizadas na Tabela §4.3. A representação do número de disparos de uma Transição ( $T_n$ ) é determinada por  $D(T_n)$ . A formação de filas nos Lugares ( $S_n$ ) é determinada por  $F(T_n)$ .

A Tabela §4.3 é dividida em 3 colunas, a coluna 1 nomeia o setor de aplicação da técnica de acordo com sua cor, a coluna 2 mostra a comparação entre o valor do número de disparos da transição e a soma das filas e disparos das transições dos setores. A coluna 3 mostra o *checklist* que indica se o modelo foi implementado corretamente naquele setor, de acordo com a aplicação da técnica. Avaliando os resultados, verifica-se que todos os setores satisfazem a premissa de partida da técnica  $\text{Entrada} = \text{Filas} + \text{Saída}$ . A verificação parte do número de disparos da transição que antecipa (entrada) o setor com valor de disparos dispostos nos marcadores de disparos e é igualada com o valor da transição de saída do setor (ou do modelo) somada as filas formadas no

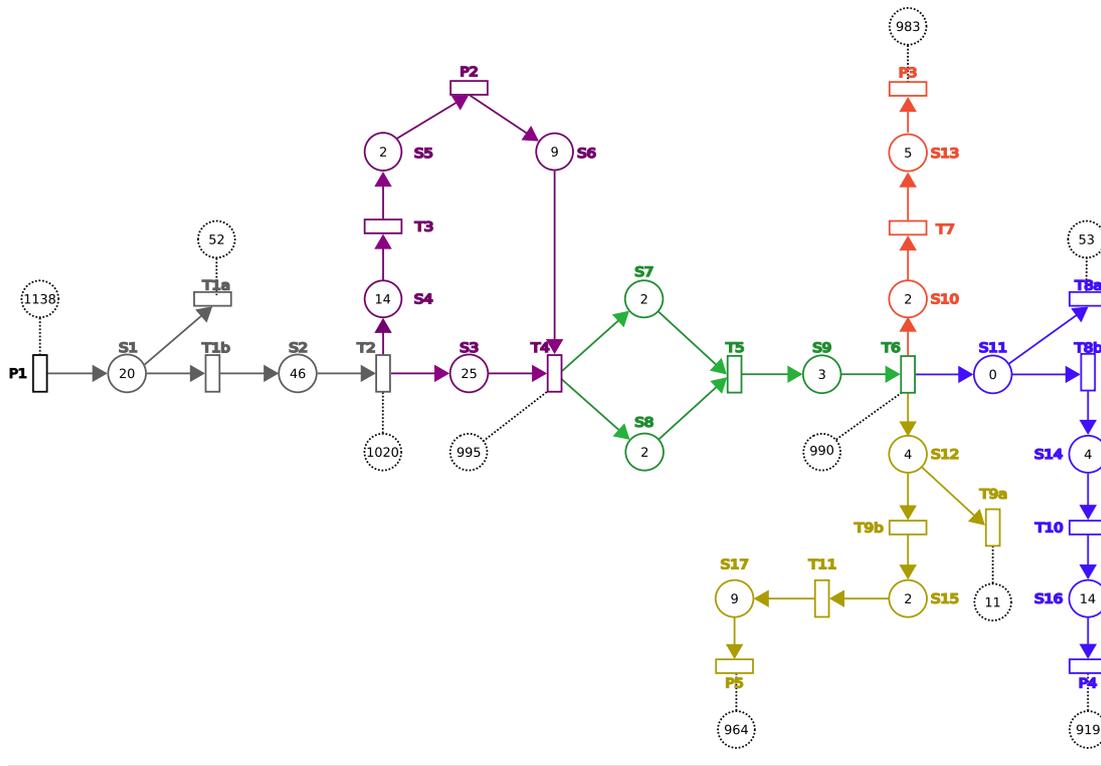


Figura 4.4: Diagrama de Aplicação da Técnica Sargent [37]

setor, havendo igualdade de valores o setor está verificado e não apresenta erros e é adicionado à coluna 3 um *checklist* positivo. desta forma conclui-se que de acordo com a técnica de Sargent [37] que o modelo de simulação foi implementado corretamente e não apresentou erros em nenhum setor.

A aplicação da técnica de validade de eventos proposta por Sargent [37] implica na ocorrência de Disparos das transições que representam o Filtro no modelo de simulação perante experimento apresentado na Seção §4.1.3. A técnica avalia se a transição disparou conforme a sua taxa de disparo. A calibragem das Transições que representam a funcionalidade de remover mensagens do fluxo foi feita de acordo com a estimativa de especialistas. Esta análise é descrita na Tabela §4.4 que demonstra a perspectiva de funcionamento da tarefa filtro do ponto de vista da Tecnologia Guaraná e do modelo de simulação em Redes de Petri. A coluna 1 identifica as Tarefas Filtro e a coluna 2 a Taxa de filtragem estimada para cada uma delas. Na coluna 3 temos a representação da tarefa Filtro em Redes de Petri, sendo as transições T1a, T8a e T9a representantes da função filtrar mensagem, e as Transições T1b, T8b

Setor	Contagem	Resultado
Azul	$D(T6)=D(P4)+F(S16)+F(S14)+F(S11)+D(T8a)$	✓
	$990=919+14+4+0+53 \Rightarrow 990=990$	
vermelho	$D(T6)=D(P3)+F(S13)+F(S10)$	✓
	$990=983+5+2 \Rightarrow 990=990$	
amarelo	$D(T6)=D(P5)+F(S17)+F(S15)+F(S12)+D(T9a)$	✓
	$990=964+9+2+4+11 \Rightarrow 990=990$	
verde	$D(T4)=D(T6)+F(S9)+(F(S7)+F(S8))/2$	✓
	$995=990+3+(2+2)/2 \Rightarrow 995=995$	
violeta	$D(T2)=D(T4)+F(S3)= D(T4)+F(S6)+F(S5)+F(S14)$	✓
	$1020=995+25=995+9+2+14 \Rightarrow 1020=1020=1020$	
cinza	$D(P1)=D(T2)+F(S2)+F(S1)+D(T1a)$	✓
	$1138=1020+46+20+52 \Rightarrow 1138=1138$	

**Tabela 4.3:** Aplicação da Técnica de Verificação proposta por Sargent [37]

e T9b representam a função não filtrar. A coluna 4 mostra a taxa de disparo estipulada para cada transição. Na coluna 5 é expresso o valor total médio de disparos ocorridos durante o experimento. O número total médio de mensagens processadas pelo Filtro é conhecido somando-se o número de disparos da tarefa com função filtrar com os disparos da tarefa não filtrar, deste total obtêm-se por uma regra de três simples com a taxa de disparo a porcentagem de mensagens filtradas no experimento, descritas na coluna 6. Comparando o número de mensagens filtradas esperado com o obtido no experimento, pode-se constatar que seus valores são muito próximos, tratando-se de processos estocásticos, o que leva a concluir que há uma equivalência significativa nos eventos esperados no modelo conceitual e no modelo de simulação.

Modelo Conceitual Tecnologia Guaraná		Modelo de Simulação Redes de Petri			
Tarefa	Taxa de Filtragem	Transição	Taxa de Disparo	Média de Disparos	Porcentagem Média Efetiva de Mensagens Filtradas
T1	0,05	T1a	0,05	538,49	0,050406159
		T1b	1	10144,53	
T8	0,05	T8a	0,05	452,67	0,04547217
		T8b	1	9502,21	
T9	0,01	T9a	0,01	100,34	0,010083875
		T9b	1	9850,2	

Tabela 4.4: Aplicação da Técnica Validade de Eventos

## 4.4 Resumo do Capítulo

Neste capítulo foi descrito o experimento realizado por meio da simulação. Um cenário experimental foi proposto considerando a carga de processamento de 10.000 mensagens recebidas pelo sistema, repetidas 25 vezes para excluir alguma discrepância nos dados. O modelo foi implementado em uma ferramenta de simulação e foram determinadas algumas variáveis para análise e identificação de possíveis comportamentos da solução de integração modelada. A ferramenta de simulação utilizada para a análise do modelo foi o *Platform Independent Petri net Editor 2* (PIPE2) que é uma plataforma de código aberto para criação e análise de Redes de Petri Generalizadas e Estocásticas. O PIPE2 conta com interface de fácil utilização, muitos módulos de análise, entre eles o módulo *Simulation* (simulação) que foi usado para analisar o modelo de simulação proposto. O módulo simulação tem como parâmetro de entrada um determinado número de entrada, sendo assim necessário à multiplicação do valor da carga de entrada no modelo conceitual por 16 totalizando 160.000 disparos.

Os resultados experimentais estão organizados em forma de tabela onde as linhas representam as transições e as colunas expressam informações como nome da referida tarefa, taxa de disparo, probabilidade global, expectativa de disparo, disparos realizados, frequência de disparo e tempo médio de espera.

Na análise dos resultados observou-se que houve formação de filas, o que é esperado quando se trata de sistema orientado a mensagens. Porém, o maior acúmulo de mensagens no *slot* S3 sugere a existência de um possível gargalo.

Por meio da correlação de Pearson foi possível identificar que haverá uma variação no número de execuções das tarefas e na formação de filas. A determinação da frequência de disparo possibilitou estimar o tempo médio de atraso das transições aptas a disparar. Com isso, nas condições experimentais estipuladas, o tempo médio que uma tarefa aguarda para processar fica muito próximo a média ideal, de modo que a diferença do tempo de atraso mínimo não interfere na formação de filas do sistema. Além disso, este capítulo apresentou uma verificação qualitativa do modelo de simulação. As técnicas utilizadas têm como base o conhecimento de especialistas da área. A verificação apresentou resultados positivos de acordo com todas as técnicas utilizadas.

---

## Capítulo 5

# Conclusões e Trabalhos Futuros

---

*Educai as crianças e não será preciso punir os homens*

*Pitágoras, 500 a.C.*



prática de integrar aplicações se torna cada vez mais frequente em empresas que buscam inovação e crescimento. Uma solução de integração é desenvolvida especificamente para integrar um determinado grupo de aplicações e propor uma nova funcionalidade. Soluções de integração podem apresentar erros de projeto. Os prejuízos resultantes de estes erros podem variar desde problemas simples até perdas significativas de negócios. A simulação de soluções de integração pode auxiliar na compreensão do comportamento das soluções, uma vez que os sistemas são mais que a soma de seus componentes. A modelagem matemática e simulação de soluções de integração de aplicações ainda é uma área pouco explorada e tem como proposta a identificação de possíveis gargalos de desempenho nas soluções de integração com base nos seus modelos conceituais. A pesquisa realizada nesta dissertação partiu de um modelo conceitual de solução de integração como caso de estudo, realizou a transcrição dos seus elementos para as Redes de Petri e propôs-se um modelo de simulação, o qual foi submetido a experimentos para obtenção de dados que foram analisados e interpretados. Para qualificação do trabalho foi proposta a verificação do modelo de simulação. Este capítulo trata na sequência das conclusões e contribuições referente a pesquisa realizada e algumas limitações encontradas.

A principal contribuição deste trabalho foi a constatação de que é possível representar modelos conceituais de soluções de integração de aplicações

por meio das Redes de Petri Estocásticas. Este resultado foi alcançado partindo da semelhança entre elementos que compõem as duas linguagens e entre seus motores de execução. A semelhança na execução possibilitou o desenvolvimento de um modelo de simulação. O experimento demonstrou que o modelo de simulação apresenta comportamento semelhante a uma solução de integração. Com isso, foi possível identificar pontos de formação de filas, possíveis gargalos de desempenho e identificar comportamentos da solução de integração ainda na fase de projeto.

A verificação teve como objetivo testar a correspondência do modelo conceitual com o modelo proposto. A aplicação de técnicas de verificação da literatura demonstrou que os resultados experimentais se aproximaram dos esperados por especialistas e também comprovaram que o modelo está implementado corretamente. Com isso, concluiu-se que é possível representar um modelo conceitual de solução de integração por meio de um modelo matemático equivalente.

A modelagem conceitual proposta nesta pesquisa apresenta as seguintes limitações: a) utilizou-se apenas as Redes de Petri Estocásticas para a modelagem da solução de integração, o que não permitiu a representação da semântica de todas as tarefas do Guaraná, a qual pode interferir no comportamento do modelo de simulação, e b) a ferramenta de simulação possui outros módulos para analisar Redes de Petri, além do simulador, os quais não foram explorados nesta dissertação.

## 5.1 Trabalhos Futuros

Como trabalho futuro sugere-se o estudo de outros módulos de análise da ferramenta PIPE, o qual poderá complementar os resultados já obtidos. Além disso, dada a grande variedade de ferramentas de simulação para Redes de Petri e as várias extensões destas para a modelagem de sistemas de eventos discretos, sugere-se como trabalho futuro o estudo de outras ferramentas de simulação, bem como a utilização de variações das redes de Petri como alternativas para a representação da semântica das tarefas do Guaraná.

O processo V&V tem a finalidade de assegurar maior confiabilidade e veracidade aos resultados dos experimentos e aos modelos propostos. Desta forma propõe-se como trabalho futuro a validação do modelo como forma de assegurar maior confiabilidade ao modelo de simulação proposto e aos resultados encontrados nesta pesquisa.

---

## Bibliografia

---

- [1] *Platform independent petri net editor 2*, 2003. URL: <http://pipe2.sourceforge.net/>.
- [2] *Grupo de pesquisa em computação aplicada*, 2013. URL: <http://www.gca.unijui.edu.br/group-gca>.
- [3] L. A. Aguirre. *Introdução à identificação de sistemas–técnicas lineares e não-lineares aplicadas a sistemas reais*. Editora UFMG, 2004.
- [4] R. D. Arteiro, F. N. Souza, N. S. Rosa e P. R. Maciel. *Utilizando redes de petri para modelagem de desempenho de middleware orientado a mensagem*, 2007.
- [5] J. P. Barros. *Cppnets: uma classe de redes de petri de alto-nível: implementação de um sistema de suporte à sua aplicação e análise*. 1996.
- [6] J. Cardoso e R. Valette. *Redes de petri*. Editora da UFSC, 1997.
- [7] R. S. Cargnin e F. Roos-Frantz. *Revisão da literatura de simulação de sistemas de eventos discretos com foco na integração de aplicações*. III SFCT, página 15, 2015.
- [8] J. S. Carson et al.. *Model verification and validation*. Em *Simulation Conference, 2002. Proceedings of the Winter*, volume 1, páginas 52–58. IEEE, 2002.
- [9] H. Carvalho et al.. *Análise de desempenho do trabalho multifuncional em linhas de produção, em forma de u pela modelagem e simulação usando redes de petri temporizadas*. XXXVII-SBPO-Simpósio Brasileiro de Pesquisa Operacional, 2005.
- [10] L. Chwif. *Redução de modelos de simulação de eventos discretos na sua concepção: uma abordagem causal*. Tese Doutoral, Universidade de São Paulo, 1999.

- [11] E. M. M. Costa. *Redes de petri e aplicações aos sistemas a eventos discretos*. Clube de Autores, 2007.
- [12] E. G. Da Silva, F. R. Frantz, R. Z. Frantz e S. Sawicki. *Desenvolvimento de uma solução de integração de aplicações para automatizar reservas de viagem*. *Salão do Conhecimento*, 1(1), 2015.
- [13] P. D. S. DA SIMULAÇÃO. *Ivan costa pereira*. Tese Doutoral, ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ, 2000.
- [14] P. J. de Freitas Filho. *Introdução à modelagem e simulação de sistemas: com aplicações em arena*. Visual Books, 2001.
- [15] D. d. P. de OLIVEIRA. *Sistemas, organização e métodos: uma abordagem gerencial*. atlas, 2002.
- [16] N. J. Dingle, W. J. Knottenbelt e T. Suto. *Pipe2: a tool for the performance evaluation of generalised stochastic petri nets*. *ACM SIGMETRICS Performance Evaluation Review*, 36(4):34–39, 2009.
- [17] T. Facchin e M. A. Sellitto. *Medição do inventário em processo e tempo de atravessamento em manufatura por modelagem em redes de petri e diagrama de resultados*. *Gestão Produção, São Carlos*, 15(2):307–321, 2008.
- [18] G. S. Fishman e L. R. Morales. *Conceptos y métodos en la simulación digital de eventos discretos*. Limusa, 1978.
- [19] R. Z. Frantz, R. Corchuelo e J. González. *Advances in a dsl for application integration*. *Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos*, 2(2), 2008.
- [20] R. Z. Frantz, R. Corchuelo, F. Roos-Frantz e S. Sawicki. *A technology proposal to realise enterprise application integration*. *International Journal of Computer Research*, 22(3):307, 2015.
- [21] 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.
- [22] R. Z. Frantz. *Enterprise application integration: an easy-to-maintain model-driven engineering approach*. Tese Doutoral, Universidad de Sevilla, 2012.

- [23] C. M. Grinstead e J. L. Snell. *Introduction to probability*. American Mathematical Soc., 2012.
- [24] M. Heiner, M. Herajy, F. Liu, C. Rohr e M. Schwarick. *Snoopy—a unifying petri net tool*. Em *Application and Theory of Petri Nets*, páginas 398–407. Springer, 2012.
- [25] G. Hohpe e B. Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.
- [26] C. Ibsen e J. Anstey. *Camel in action*. Manning Publications Co., 2010.
- [27] W. D. Kelton e A. M. Law. *Simulation modeling and analysis*. McGraw Hill Boston, 2000.
- [28] J. Kleijen. *Validation of models: statistical techniques and data availability*. Em *Simulation Conference Proceedings, 1999 Winter*, volume 1, páginas 647–654. IEEE, 1999.
- [29] D. S. Linthicum. *Enterprise application integration*. Addison-Wesley Professional, 2000.
- [30] P. R. Maciel, R. D. Lins e P. R. Cunha. *Introdução às redes de petri e aplicações*. UNICAMP-Instituto de Computacao, 1996.
- [31] J. MARTINS. *Técnicas para gerenciamento de projetos de software*. 2007.
- [32] M. M. Miyagi e P. E. Miyagi. *Modelagem e análise de serviços de saúde baseados em redes de petri interpretadas*. *Revista Produção*, 11(2), 2002.
- [33] A. Morales Varela, J. A. Rojas Ramírez, L. H. Hernández Gómez, Á. Morales González e M. Y. Jiménez Reyes. *Modelo de un sistema de producción esbelto con redes de petri para apoyar la toma de decisiones*. *Ingeniare. Revista chilena de ingeniería*, 23(2):182–195, 2015.
- [34] M. Mukaka. *A guide to appropriate use of correlation coefficient in medical research*. *Malawi Medical Journal*, 24(3):69–71, 2012.
- [35] R. Pressman. *Engenharia de software*. McGraw Hill Brasil, 2009.
- [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*. Em *17th International Conference on Enterprise Information Systems (ICEIS)*, páginas 87–97, 2015.

- [37] R. G. Sargent. *Verification and validation of simulation models*. *Journal of Simulation*, 7(1):12–24, 2013.
- [38] J. A. d. S. SENA. *Desenvolvimento de framework para análise e simulação dinâmica de sistemas elétricos de potência*. 2013.
- [39] M. Spiegel, J. Schiller e R. Srinivasan. *Teoria e prob. probabilidade e estatística*. Coleção Schaum. BOOKMAN COMPANHIA ED, 2004.
- [40] R. M. Stair, G. W. Reynolds e F. S. C. da Silva. *Princípios de sistemas de informação: uma abordagem gerencial*. 1998.
- [41] H. Stamm et al.. *Simulação industrial: uma avaliação de sua utilização no sudeste e sul do brasil*. 1998.
- [42] J. Strack. *Gpss: modelagem e simulação de sistemas*. LTC, 1984.
- [43] G. Strang e T. Nguyen. *Wavelets and filter banks*. SIAM, 1996.
- [44] M. C. Yamada, A. J. V. Porto e R. Y. Inamasu. *Aplicação dos conceitos de modelagem e de redes de petri na análise do processo produtivo da indústria sucroalcooleira*. *Revista Pesquisa Agropecuária Brasileira*. Brasília, 37(6):809–820, 2002.

This document was typeset on June 3, 2016 using class RGBOR  $\alpha$ 2.14 for L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub> . 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.