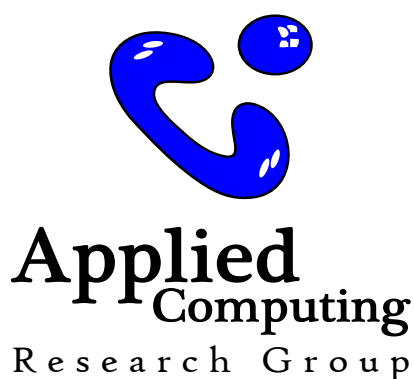

MODELO DE SIMULAÇÃO COMPUTACIONAL PARA ANÁLISE DA QUALIDADE DE SERVIÇO DE UMA SOLUÇÃO DE INTEGRAÇÃO DE RESERVA DE VIAGENS UTILIZANDO REDES DE PETRI TEMPORIZADAS

FRANCISCO DA SILVEIRA
UNIVERSIDADE REGIONAL DO NOROESTE DO
ESTADO DO RIO GRANDE DO SUL

DISSERTAÇÃO DE MESTRADO

ORIENTADOR:
DR. SANDRO SAWICKI
COORIENTADORA:
DRA. FABRICIA ROOS-FRANTZ



JUNHO, 2017

First published in June 2017 by
Applied Computing Research Group - GCA
Department of Exact Sciences and Engineering
Rua Lulu Ilgenfritz, 480 - São Geraldo
Ijuí, 98700-000, Brazil.

Copyright © MMXVII Applied Computing Research Group
<http://www.gca.unijui.edu.br>
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.

Universidade Regional do Noroeste do Estado do Rio Grande do Sul

A Comissão Examinadora, abaixo assinada, _____ a dissertação intitulada “Modelo de Simulação Computacional para Análise da Qualidade de Serviço de uma Solução de Integração de Reserva de Viagens utilizando Redes de Petri Temporizadas: ”, elaborada por Francisco da Silveira, como requisito parcial para a obtenção do título de Mestre em Modelagem Matemática.

Dr. Sandro Sawicki
UNIJUÍ
(Orientador)

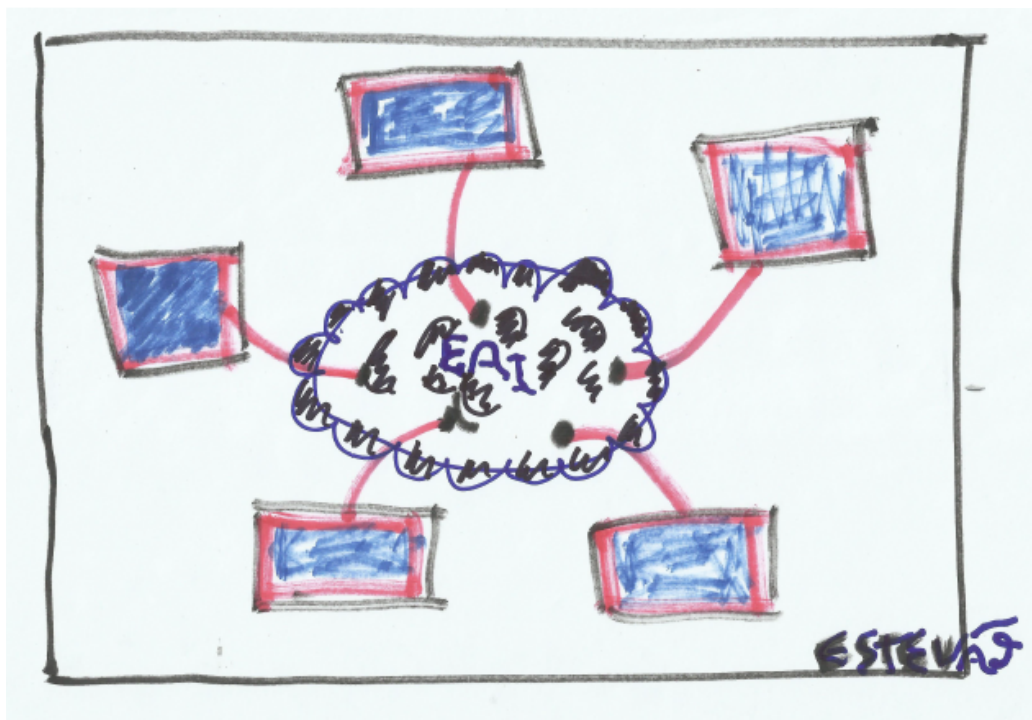
Dra. Fabricia Roos-Frantz
UNIJUÍ
(Co-orientadora)

Dra. Airam Sausen
UNIJUÍ

Dr. Rafael Zancan Frantz
UNIJUÍ

Dr. Vinicius Maran
UFSM

Ijuí, ____ de _____ de _____.



Integração de Aplicações por Estevão, 9 anos de idade.

Dedico este trabalho a Jesus e a família.

Conteúdo

Agradecimentos	vii
Resumo	ix
Abstract	xi
1 Introdução	1
1.1 Contexto da Pesquisa	1
1.2 Motivação	2
1.3 Objetivos	3
1.3.1 Geral	3
1.3.2 Específicos	3
1.4 Metodologia	3
2 Referencial Teórico	5
2.1 Integração de Aplicações	6
2.1.1 Estilos de Integração	7
2.1.2 Topologias de Integração	12
2.1.3 Plataformas de Integração	14
2.2 Redes de Petri	15
2.2.1 Formalismo Algébrico das Redes de Petri	16
2.2.2 Representação Gráfica	17
2.2.3 Redes de Petri Coloridas	18
2.2.4 Redes de Petri Temporizadas	19
2.2.5 Semântica de Disparo de Transição	20
2.3 Sistemas, Modelos e Simulação	21
2.3.1 Sistemas	21
2.3.2 Modelos de Sistemas	22

2.4	Simulação Computacional	24
2.5	Plataformas de Simulação de redes de Petri Temporizadas	25
2.6	Resumo do Capítulo	26
3	Trabalhos Relacionados	27
3.1	Trabalhos relacionados envolvendo Redes de Petri Temporizadas	27
3.2	Resumo do Capítulo	30
4	Estudo de Caso	33
4.1	Ecosistema de Software	33
4.2	Modelo Conceitual de Integração	34
5	Modelo de Simulação Proposto	37
5.1	Modelo em rede de Petri Temporizadas da Solução de Integração no Guaraná DSL	37
5.1.1	Formulação Matemática	39
5.2	Experimentação	41
5.2.1	Modelagem do Sistema e a Plataforma CPN Tools	41
5.2.2	Modelo de Distribuição de Entrada	47
5.2.3	Qualidade de Serviço (QoS)	48
5.2.4	Caracterização dos Cenários	48
5.2.5	Função Probabilidade	50
5.3	Resultados Experimentais	53
5.4	Conjectura sobre os Resultados	58
5.5	Resumo do Capítulo	60
6	Verificação do Modelo de Simulação	63
6.1	Técnicas de Verificação	63
6.2	Verificação do Modelo de Simulação	65
6.3	Resumo do Capítulo	67
7	Conclusões e Trabalhos Futuros	69
7.1	Resumo das Contribuições	70
.1	ANEXOS	71
	Bibliografia	81

Índice de figuras

2.1	Representação abstrata de uma solução de integração.	7
2.2	Representação abstrata da Transferência de Arquivos.	8
2.3	Representação abstrata de um Banco de Dados compartilhado.	9
2.4	Representação abstrata de uma Chamada de Procedimento Remoto.	10
2.5	Representação abstrata do Sistema de Mensagens.	11
2.6	Representação abstrata da topologia Point-to-point.	13
2.7	Representação abstrata da topologia Hub-and-spoke.	13
2.8	Representação abstrata da topologia <i>Enterprise Service Bus</i>	14
2.9	Exemplo de uma rede de Petri em sua representação gráfica.	18
2.10	Rede de Petri Colorida.	19
2.11	Rede de Petri temporizada determinística.	21
2.12	Fluxograma das definições de um Sistema Law e Kelton [29].	22
2.13	Fluxograma do processo de Modelagem de um Sistema Law e Kelton [29].	23
4.1	Solução de Integração desenvolvida pela tecnologia Guaraná. Frantz [21]	34
5.1	Notação gráfica do domínio do Guaraná DSL Frantz et al. [20].	38
5.2	Equivalência entre elementos do Guaraná DSL e Redes de Petri Temporizadas Fabricia Roos-Frantz [17].	38
5.3	Equivalência entre mudanças de estado do Guaraná DSL e redes de Petri Temporizadas Fabricia Roos-Frantz [17].	39
5.4	Elementos que constituem a solução de integração da Agência de Viagens [21].	40
5.5	Modelo em Rede de Petri Temporizada da solução de integração no Guaraná DSL.	42
5.6	Modelo em rede de Petri Temporizadas transcrito na linguagem do CPN Tools.	43

- 5.7 Declaração dos lugares, *tokens*, variáveis e funções na linguagem do CPN Tools. 44
- 5.8 Caminho que um *token* percorre desde a entrada no lugar Sistema de Viagens (TS) até o lugar Email (MS) no CPN Tools. 45
- 5.9 Gráfico de quantidade de sequências por conjunto de S_n , sendo $n = \{5,6,\dots,24,25\}$ 54
- 5.10 Gráfico do Total de *tokens* armazenados no lugar Email após as simulações com 10.000, 50.000 e 100.000 mil disparos em relação aos valores de tempos possíveis para esse lugar. 55
- 5.11 Gráfico do Total de *tokens* armazenados no lugar Fatura após as simulações com 10.000, 50.000 e 10.0000 mil disparos em relação aos valores de tempos possíveis para esse lugar. 56
- 5.12 Gráfico do Total de *tokens* armazenados no lugar Voos após as simulações com 10.000, 50.000 e 100.000 mil disparos em relação aos valores de tempos possíveis para esse lugar. 57
- 5.13 Gráfico do Total de *tokens* armazenados no lugar Hotel após as simulações com 10.000, 50.000 e 100.000 mil disparos em relação aos valores de tempos possíveis para esse lugar. 58
- 5.14 Gráfico geral do Total de *tokens* armazenados nos lugares Email, Fatura, Voos e Hotel, após as simulações com 10.000, 50.000 e 100.000 mil disparos em relação aos valores de tempos possíveis para o modelo de simulação. 59
- 5.15 Resultado da simulação demonstrando a quantidade de *tokens* armazenados no lugar Email (MS) após 10.000 disparos e 25 repetições. 60
- 5.16 Resultado da simulação demonstrando a quantidade de *tokens* armazenados no lugar Email (MS) após 50.000 disparos e 25 repetições. 61
- 5.17 Resultado da simulação demonstrando a quantidade de *tokens* armazenados no lugar Email (MS) após 100.000 disparos e 25 repetições. 61

- 7.1 Imagem das opções de funções de distribuição aleatórias disponíveis no CPN Tools [2]. 71
- 2 Resultado da simulação demonstrando a quantidade de *tokens* armazenados no lugar Fatura (IS) após 10.000 disparos e 25 repetições. 72
- 3 Resultado da simulação demonstrando a quantidade de *tokens* armazenados no lugar Fatura (IS) após 50.000 disparos e 25 repetições. 72

- 4 Resultado da simulação demonstrando a quantidade de *tokens* armazenados no lugar Fatura (IS) após 100.000 disparos e 25 repetições. [73](#)
- 5 Resultado da simulação demonstrando a quantidade de *tokens* armazenados no lugar Voos (V) após 10.000 disparos e 25 repetições. [74](#)
- 6 Resultado da simulação demonstrando a quantidade de *tokens* armazenados no lugar Voos (V) após 50.000 disparos e 25 repetições. [75](#)
- 7 Resultado da simulação demonstrando a quantidade de *tokens* armazenados no lugar Voos (V) após 100.000 disparos e 25 repetições. [76](#)
- 8 Resultado da simulação demonstrando a quantidade de *tokens* armazenados no lugar Hotel (H) após 10.000 disparos e 25 repetições. [77](#)
- 9 Resultado da simulação demonstrando a quantidade de *tokens* armazenados no lugar Hotel (H) após 50.000 disparos e 25 repetições. [78](#)
- 10 Resultado da simulação demonstrando a quantidade de *tokens* armazenados no lugar Hotel (H) após 100.000 disparos e 25 repetições. [79](#)

Agradecimentos

A persistência é o caminho do êxito.

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

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

Resumo

O início de todas as coisas é pequeno.

Marcus T. Cícero, filósofo Romano (106 AC - 43 AC)

O mercado dinâmico e altamente competitivo faz com que as empresas se esforcem na busca pela qualidade, condição fundamental para uma organização que visa expandir seus negócios. Com isso, a Tecnologia de Informação, tornou-se um importante diferencial, pois auxilia no gerenciamento do negócio, agiliza e facilita a tomada de decisão. A medida que a empresa cresce, são adquiridas novas aplicações para atender novas demandas de negócios, muitas dessas aplicações nem sempre têm a mesma linguagem e plataforma de operação gerando, por sua vez, um emaranhado de aplicações distintas com fins específicos e sem comunicação umas com as outras. Neste cenário, surge a área de Integração de Aplicações Empresariais, termo do inglês *Enterprise Application Integration (EAI)*, que busca proporcionar metodologias, técnicas e ferramentas para a concepção e a implementação de soluções de integração. Uma solução de integração tem o objetivo de orquestrar um conjunto de aplicações para mantê-las sincronizadas ou proporcionar novas funcionalidades que possam ser construídas a partir daquelas já existentes. Este trabalho tem por objetivo desenvolver um modelo formal de simulação utilizando redes de Petri Temporizadas a partir de um modelo conceitual de reservas de viagem. O modelo conceitual utilizado como caso de estudo foi projetado utilizando a tecnologia Guaraná. Este trabalho busca encontrar o tempo mínimo e máximo que uma mensagem leva para ser processada na solução de integração antes que ela seja implementada e implantada. Com isso, é possível identificar níveis de qualidade de serviço sem que haja a necessidade de implementar a solução, baseando-se apenas, na estrutura atual do modelo conceitual. O modelo formal desenvolvido foi simulado em cenários críticos de operação, utilizando cargas de

entrada diferentes usando a ferramenta CPN Tools. Neste trabalho são utilizadas técnicas formais de verificação presentes na literatura para demonstrar a equivalência entre o modelo conceitual e o modelo de simulação.

Palavras-chave: Integração de Aplicações Empresariais. Redes de Petri Temporizadas. Qualidade de Serviço. Modelo de Simulação Computacional.

Abstract

The beginnings of all things are small.

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

The dynamic and highly competitive market makes companies strive for quality, a fundamental condition for an organization that aims to expand its business. With this, the Information Technology, has become an important differential, as it assists in the management of the business, streamlines and facilitates the decision making. As the company grows, new applications are acquired to meet new business demands; many of these applications do not always have the same language and operating platform, generating, in turn, a tangle of distinct applications for specific purposes and without communication with each other. In this scenario, the area of Enterprise Application Integration (EAI) that seeks to provide methodologies, techniques and tools for the design and implementation of integration solutions. An integration solution is intended to orchestrate a set of applications to keep them synchronized or provide new features that can be built from those already in place. This work aims to develop a formal simulation model using Petri nets Timed from a conceptual model of travel reservations. The conceptual model used as case study was designed using GuaranÃ; technology. This work seeks to find the minimum and maximum time that a message takes to be processed in the integration solution before it is implemented and deployed. With this, it is possible to identify levels of quality of service without the need to implement the solution, based only on the current structure of the conceptual model. The formal model developed was simulated in critical operating scenarios using different input loads using the CPN Tools. In this work formal verification techniques presents in the literature are used to demonstrate the equivalence between the conceptual model and the simulation model.

Keywords: Enterprise Application Integration. Timed Petri Nets. Quality Service. Computational Simulation Model.

Capítulo 1

Introdução

Penso logo existo.

René Descartes

1.1 Contexto da Pesquisa

As atividades empresariais são continuamente ampliadas pelas tecnologias de informação disponíveis. Atualmente as aplicações empresariais são essenciais nos processos de negócios das corporações. As aplicações realizam a manipulação das informações da empresa, disponibilizando aos proprietários e interessados operabilidade entre os diversos sistemas de negócio, o que as torna imprescindíveis na tomada de decisões de uma corporação. No mercado de negócios existem diversos tipos de aplicações para atender a demanda de recursos humanos, setor financeiro, engenharia de produção entre outros. Estas aplicações são construídas em diferentes plataformas, com linguagens variadas e para inúmeras funções Frantz [21]. As aplicações que operam dentro da infraestrutura das corporações e que formam um sistema heterogêneo tornam necessária a criação de maneiras que oportunizem a integração entre essas diversas aplicações. Neste cenário, é normal que as empresas ao longo de sua formação, contenham no ecossistema de *software*, aplicações que não cooperam entre si. Esta cooperação, interação, ou interoperabilidade é o que se denomina na comunidade empresarial de intergração de aplicações. No momento em que é criada uma integração entre essas aplicações há interoperabilidade no ecossistema de *software*. A Integração de Aplicações Empresariais, do termo em inglês *Enterprise Applications Integration (EAI)*, é uma área de estudo que fornece ferramentas e métodos para

a integração de aplicações empresariais Hohpe e Woolf [25]. Há um crescimento significativo da procura por EAI, aumentando a demanda e os investimentos das empresa neste mercado. Nesta área de integração de aplicações existem algumas plataformas operacionais que oferecem um ambiente de programação com uma Linguagem de Domínio Específico, termo do inglês *Domain Specific Language (DSL)*. Esta linguagem é o principal meio de integração para as aplicações. As principais plataformas de implementação de soluções de integração são o Guaraná DSL Frantz et al. [20], Apache Camel Ibsen e Anstey [27], Mule Soft Dossot et al. [14] e Spring Integration Fisher et al. [18]. Uma solução de integração pode ser representada em alto nível, por um modelo conceitual, podendo ser interpretada como um sistema dinâmico de eventos discretos. Este trabalho busca criar um modelo de simulação em redes de Petri Temporizadas de uma solução de integração, elaborada no Guaraná DSL Frantz [21], de uma Agência de Viagens, para ainda na fase de projeto, poder analisar se há possíveis gargalos de desempenho da solução de integração. Um gargalo de desempenho é definido pela diminuição do fluxo de dados em pontos críticos do sistema. Estes pontos críticos são determinados pela estrutura do modelo. Com isso, pretende-se apresentar a Qualidade de Serviço (QoS) do modelo, através de simulações da rede de Petri Temporizada na ferramenta *CPN Tools cpn* [2].

1.2 Motivação

A maioria das corporações investe bastante em aplicações heterogêneas para o gerenciamento dos processos de negócios. Essas aplicações são adquiridas sem considerar sua possível integração. Os ecossistemas de *softwares* das empresas são formados por um aglomerado de aplicações que não conversam entre si. Esta situação potencializa um grande mercado de desenvolvimento de soluções de integração empresariais para gerar interoperabilidade entre as diversas aplicações Hohpe e Woolf [26]. Mas, o desenvolvimento de uma solução de integração está dividido em etapas de levantamento de requisitos, análise, projeto, implementação e testes, no qual os dois últimos possuem um custo elevado Pressman [34]. Apesar de uma solução de integração seguir todos os padrões e técnicas de implementação, podem ocorrer erros em sua estrutura. Além disso, podem apresentar gargalos de performance nos seus componentes em situações de grande demanda. Sendo assim, pode-se utilizar modelos computacionais e matemáticos, e técnicas de simulação de eventos discretos visando analisar o comportamento e encontrar possíveis gargalos de performance que podem surgir quando uma solução de integração de aplicação é submetida a um cenário crítico de funcionamento. Assim, a presente pesquisa tem como motivação a possibilidade

de analisar o comportamento e identificar os gargalos de performance, ainda na fase de projeto, por meio dos modelos conceituais da solução.

1.3 Objetivos

Os objetivos são apresentados como Objetivo Geral e Objetivos Específicos.

1.3.1 Geral

Desenvolver um modelo formal de simulação com o objetivo de identificar o tempo que uma mensagem leva para percorrer as tarefas dentro de uma solução de integração, com vistas a encontrar a Qualidade de Serviço (QoS) oferecida pelo modelo, ainda na fase de projeto.

1.3.2 Específicos

- Pesquisar trabalhos que utilizam redes de Petri Temporizadas para o desenvolvimento de modelos de simulação.
- Pesquisar ferramentas de simulação que suportam redes de Petri Temporizadas e escolher a mais adequada para o escopo do problema.
- Identificar as equivalências entre os elementos da tecnologia Guaraná e redes de Petri Temporizadas.
- Desenvolver o modelo de simulação usando redes de Petri Temporizadas.
- Verificar a equivalência entre o modelo desenvolvido em RdP e o modelo conceitual usando técnicas de verificação existentes na literatura.
- Simular e analisar o modelo de simulação desenvolvido.

1.4 Metodologia

Esta pesquisa se desenvolveu a partir da leitura de livros, artigos científicos, dissertações de mestrado, teses de doutorado. Pesquisas bibliográficas de

materiais que envolvem soluções de integração de aplicações, redes de Petri Temporizadas e sistemas. Associado a isto, foram estudados os tipos de Redes de Petri aplicados no contexto de Integração de Aplicações Empresariais. Baseado na Rede de Petri escolhida foi criado um modelo formal de simulação equivalente ao modelo conceitual proposto no estudo de caso. Foram definidos os cenários de simulação similares a um processo real de funcionamento de uma solução de integração. Os resultados experimentais foram tabulados, sendo que, para cada cenário foram executadas 25 repetições, seguindo a lei dos Grandes Números. As simulações computacionais, foram realizadas por meio da ferramenta CPN Tools, que melhor se aplica ao tipo de Rede de Petri escolhida, ou seja, Rede de Petri Temporizada. Finalmente, a validação da equivalência do modelo formal de simulação em Rede de Petri com o modelo conceitual foi realizado por meio de técnicas de verificação formal largamente encontradas na literatura. Os resultados obtidos na simulação foram utilizados nas técnicas de verificação, buscando avaliar a precisão do modelo criado.

Capítulo 2

Referencial Teórico

*Think like a wise man,
but communicate in plain language.*

William B. Yeats, Irish dramatist & poet (1865-1939)

A partir do referencial teórico são apresentados os conceitos básicos que foram usados para a construção do modelo de simulação. O estudo dos trabalhos desenvolvidos em EAI, Guaraná DSL, redes de Petri Temporizadas e CPN Tools é descrito neste capítulo, conduzindo à definições importantes na produção do modelo de simulação que envolvem o estudo de caso. A Seção §2.1 é descrito o tema integração de aplicações evidenciando os seus principais aspectos teóricos, bem como o crescimento na área de tecnologia da informação devido a grande aplicabilidade nos negócios empresariais. A integração de aplicações fornece métodos que viabilizam a interoperabilidade no ecossistema de *softwares* das corporações. Estas corporações são constituídas por diversos tipos de aplicações e muitas funcionam em plataformas diferentes. Neste capítulo é realizado um estudo sobre a plataforma do Guaraná DSL, que é uma das ferramenta de implementação de uma solução de integração e que fornece o modelo conceitual do estudo de caso. O modelo conceitual da solução de integração de uma Agência de Viagens é peça central desta pesquisa. A Seção seguinte §2.2 discorre sobre o tema das redes de Petri e os principais tópicos das extensões, que seriam as redes de Petri Coloridas e Temporizadas. Esta Seção aborda os conceitos de equação de estado, regras de disparo, formação gráfica, formalismo algébrico e apresenta ferramentas de simulação

das redes de Petri, principalmente da ferramenta CPN Tools. Com a plataforma do CPN Tools é possível modelar diversos tipos de sistemas em redes de Petri coloridas e temporizadas, oferecendo ambientes de programação capazes de descreverem praticamente todas as sintaxes e condições lógicas de modelos representados em redes de Petri Coloridas e/ou Temporizadas. A Seção §2.3 discorre sobre os conceitos de sistema, modelo e simulação. Nesta seção há descrições das etapas de construção e interpretação de modelos de sistemas dinâmicos ou estáticos, abordando as características de simulação destes modelos e enfatizando a importância de caracterizar-se corretamente os elementos que formam um modelo de simulação.

2.1 Integração de Aplicações

Atualmente os processos de negócios empresariais, em sua maioria, são operados por aplicações. Esta tecnologia aumenta a qualidade dos serviços prestados entre as diversas corporações e seus clientes. A medida que o mundo se moderniza e muda, as aplicações seguem esse mesmo padrão e se atualizam em termos de linguagem e funcionalidade, tornando o ambiente de *software* das empresas cada vez mais diversificado. Segundo Hohpe e Woolf [26] as empresas são tipicamente compostas por centenas ou milhares de aplicações que são construídas por customização, isto é, construídas especificamente para um determinado serviço, adquiridas de terceiros, ou fazem parte de um sistema legado, isto é, são herança de outros sistemas ou uma combinação destes, operando em vários níveis de plataformas com sistemas operacionais diferentes. Neste cenário surgem as EAI (*Enterprise Application Integration*) com ferramentas e métodos de integração para orquestrar esses ecossistemas de *softwares* através de soluções de integração baseadas em conceitos da engenharia de *software*. As EAI são compostas basicamente por sistemas de *softwares* que trocam informações e/ou funcionalidades entre si. Cada aplicação pode ter a sua própria linguagem, que é traduzida para uma linguagem universal, um formato intermediário como XML e por um formato de comunicação a ser utilizado durante a troca de informações entre os sistemas Chaves et al. [7]. O objetivo de uma solução de integração é manter dados sincronizados entre diferentes aplicações ou desenvolver novas funcionalidades sobre as já existentes. Estes conceitos formam uma estrutura de aplicabilidade da EAI David [10] que dão suporte nas infraestruturas típicas das corporações que operam por meio de redes, conforme é ilustrado na Figura §2.1. A integração empresarial, por definição, opera com múltiplas aplicações rodando em várias plataformas em locais diferentes. Esta situação torna complexo o desenvolvimento eficaz das EAI,

segundo [26] a integração de uma empresa não é tarefa fácil. A integração da empresa exige uma mudança significativa na política corporativa. O negócio de aplicações geralmente se concentra em uma área funcional específica, como *Customer Relationship Management* (CRM), faturamento, finanças, etc. Hohpe e Woolf [26].

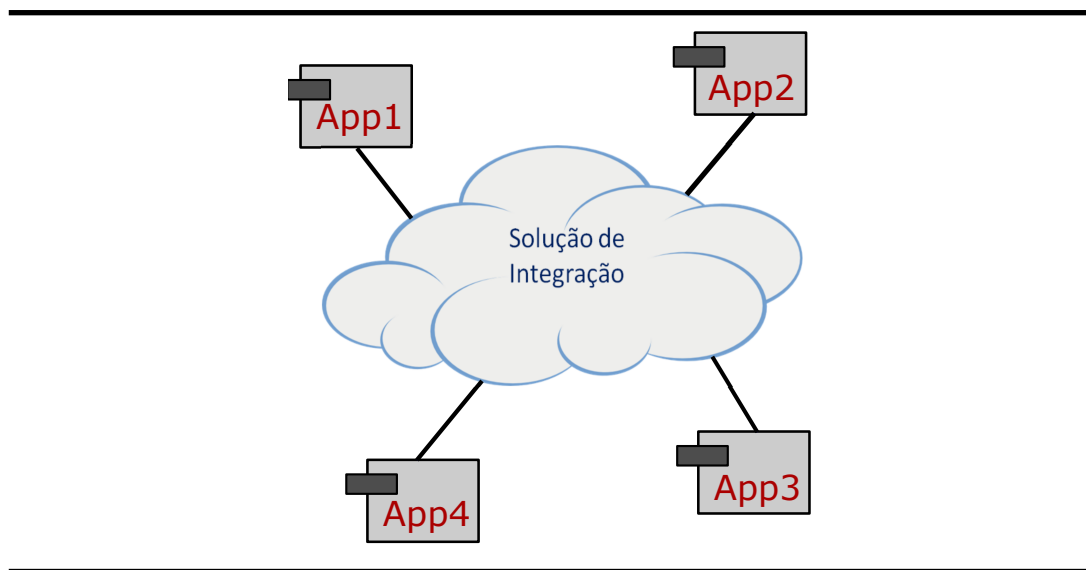


Figura 2.1: Representação abstrata de uma solução de integração.

2.1.1 Estilos de Integração

Observando esta situação os engenheiros de *softwares* que se deparam em uma situação na qual há necessidade de integração devem optar por critérios de integração para as aplicações pertencentes ao ecossistema de *software*. Existem vários critérios de integração, mas Hohpe e Wolf catalogaram os quatro principais critérios de Integração que são: transferência de arquivos, banco de dados compartilhados, chamada de procedimento remoto e mensagens. Estes critérios são estilos de integração e segundo Hohpe e Woolf [26] um produto de integração ou EAI, pode empregar uma combinação de estilos, incluídos na implementação do produto. Estes estilos são descritos a seguir:

A transferência de arquivos é um estilo de integração que realiza a troca de arquivos de uma aplicação "app1" para outra aplicação "app2" transformando os dados dos arquivos em uma linguagem comum, conforme é ilustrado na Figura §2.2. Segundo Ermagan et al. [15], a integração mais simples seria aquela que de alguma forma trabalha com aplicações que utilizam

arquivos, pois são um mecanismo de armazenamento universal, construídos em qualquer sistema operacional da empresa e disponíveis a partir de qualquer linguagem empresarial. A transferência de arquivos utiliza um mecanismo de transferência de dados comum que pode ser aplicado por uma variedade de línguas e plataformas. Cada aplicação produziria informações através de arquivos que outras aplicações precisariam consumir e que possam compartilhar. Os integradores devem assumir a responsabilidade de transformar os arquivos em diferentes formatos e produzir os ficheiros, em intervalos regulares de acordo com a natureza do negócio. Uma decisão importante com o estilo de transferência de arquivos é o formato a utilizar. A linguagem mais usada por uma solução de integração que utiliza este estilo é o XML (*eXtensible Markup Language*).



Figura 2.2: Representação abstrata da Transferência de Arquivos.

Segundo Hohpe e Woolf [24] a grande vantagem dos arquivos é que os integradores não precisam de nenhum conhecimento do código de uma aplicação, as próprias aplicações fornecem o arquivo. O conteúdo e o formato do arquivo são negociados com os integradores que lidam com as transformações exigidas para outras aplicações, ou deixam as aplicações que consomem decidir como elas querem manipular e ler o arquivo. Como resultado estas diferentes aplicações se tornam dissociadas uma da outra. Cada pedido pode fazer mudanças internas livremente, sem afetar outras aplicações, proporcionando que os dados nos arquivos se mantenham no mesmo formato.

O banco de dados compartilhados é um estilo de integração onde as aplicações existentes em um ecossistema de *software* são formatadas de modo que cada elemento pertencente ao sistema pode acessar um mesmo centro de informações conforme ilustra a Figura §2.3. As empresas envolvidas nesta estrutura de conexões podem consultar livremente um recipiente contendo um conjunto de informações exclusivas, úteis da corporação de trabalho, mesmo as de linguagem de programação diferente, pois realizando uma operação de pesquisa à dados existentes em todas as aplicações neste estilo conforme Bradác [4], há um banco de dados conectado às aplicações do sistema. Cada

aplicação pode utilizar livremente das informações disponíveis no sistema.

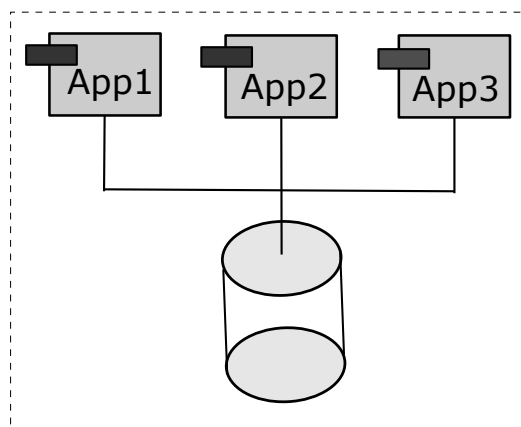


Figura 2.3: Representação abstrata de um Banco de Dados compartilhado.

Considerando uma empresa que tem várias aplicações que foram construídas de forma independente, com diferentes linguagens e plataformas a empresa precisa de informações que sejam compartilhadas de forma rápida e consistente. Conforme Hohpe e Woolf [26], a transferência de arquivo permite que as aplicações possam compartilhar dados, mas é possível que muitos desses dados estejam desatualizados. Muitas aplicações que usam um banco de dados compartilhado para lerem e modificarem dados comuns, podem frequentemente sofrer com problemas de gargalos de desempenho e até mesmo ocorrer o bloqueio de outras aplicações por inconsistência nos dados. Conforme Chaves et al. [7], deve se estudar bem o sistema da empresa para se integrar com banco de dados compartilhados. Uma de suas limitações são os pacotes externos, pois muitas vezes trabalham com um esquema diferente do que a empresa utiliza. Mesmo que haja algum espaço para adaptação, é provável que seja muito mais limitado do que os integradores gostariam. Este problema se estende à integração depois da sua construção. Quando as aplicações são distribuídas em vários computadores, o banco de dados deve ser distribuído, para que cada aplicação possa acessar o banco de dados localmente.

A chamada de procedimento remoto é um estilo de integração de aplicações que necessita invocar funcionalidades ou dados de outras aplicações. Conforme Hohpe e Woolf [26], cada aplicação pode alterar o formato dos seus dados internos sem afetar todas as outras aplicações. O estilo de transferência de arquivos e banco de dados compartilhado habilitam as aplicações

para compartilharem seus dados, o que é uma parte importante de integração de aplicações. Mas apenas a partilha de dados muitas vezes não é suficiente. Um dos mais poderosos mecanismos estruturantes no *design* de aplicações é o de encapsulamento, no qual módulos protegem seus dados através de uma interface chamada de função. Deste modo, eles podem interceptar as alterações nos dados que acontecem simultaneamente à troca dessas informações, para realizarem várias ações que elas precisam fazer Hohpe e Woolf [24]. Quando os dados são alterados no banco de dados compartilhado, estes fornecem uma grande estrutura de dados não encapsulados, o que torna muito mais difícil de realizar a troca de dados com segurança Hohpe e Woolf [26]. A transferência de arquivos permite que uma aplicação possa reagir às mudanças nos dados à medida que processa o arquivo, mas o processo pode estar atrasado. O fato de que a base de dados compartilhados tem dados não encapsulados também faz com que seja mais difícil manter uma família de aplicações integradas Bradác [4]. O que é necessário para segurar esta integração atualizada é um mecanismo em que uma aplicação realiza a chamada de uma função em outra aplicação, passando os dados que precisam ser compartilhados e invocando a função que informa ao receptor da aplicação como processar os dados. Esse mecanismo é a chamada de procedimento remoto, conforme ilustra a Figura §2.4.

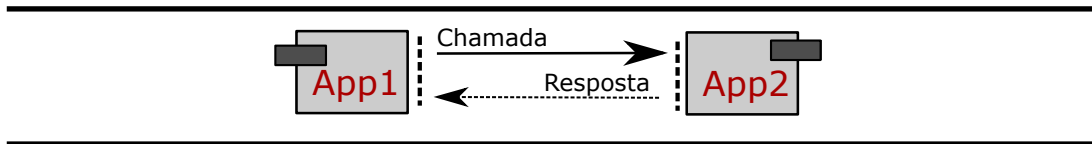


Figura 2.4: Representação abstrata de uma Chamada de Procedimento Remoto.

Segundo [26] há uma série de abordagens que aplicam chamadas de procedimento remoto: *CORBA*, *COM.NET Remoting*, *Java RMI* etc. Estas abordagens variam quanto ao número de sistemas de apoio e sua facilidade de uso. Este estilo de integração utiliza frequentemente ambientes com recursos adicionais, tais como transações. O mais utilizado atualmente são as *Web Services*, que utilizam padrões como *SOAP* e *XML*. Uma característica das *Web Services* é que elas trabalham facilmente com *HTTP*, que são fáceis de se obter através de *firewalls*. O fato é que as *Web Services* usam métodos que envolvem os dados tornando-os mais fáceis de lidar com a dissonância semântica [26]. As aplicações podem fornecer múltiplas interfaces com os mesmos dados, permitindo que alguns clientes possam ver e escolher os serviços.

O estilo de integração a mensagens é fundamentalmente uma boa opção para os problemas de sistemas distribuídos. As mensagens funcionam de modo assíncrono, isto é, não exigem que as aplicações funcionem especificamente num dado momento para garantir que a mensagem chegue à outra aplicação. O envio de uma mensagem não requer que ambos os sistemas estejam prontos ao mesmo tempo para troca de dados. Além disso, pensar sobre a comunicação de forma assíncrona força os desenvolvedores a reconhecerem que trabalhar com uma aplicação de modo remoto torna o processo mais lento.

Mensagens podem ser transformadas em trânsito sem que o remetente ou o destinatário saiba sobre essa transformação. Na verdade, essa dissociação entre os sistemas permite aos integradores transmitir mensagens para múltiplos receptores. A transformação significa que as aplicações separadas podem ter diferentes modelos conceituais. Através do envio de pequenas mensagens com frequência se permite que as aplicações colaborem no compartilhamento de dados [24]. Se um processo precisa ser realizado uma vez, é recebida uma reivindicação de seguro que pode ser feita imediatamente com uma mensagem, enquanto uma única afirmação retorna. Informações podem ser solicitadas e uma resposta construída rapidamente enquanto tal colaboração não iria ser tão rápida com a chamada de procedimento remoto, pois o chamador não precisa parar enquanto a mensagem está sendo processada e ao ser devolvida a resposta.

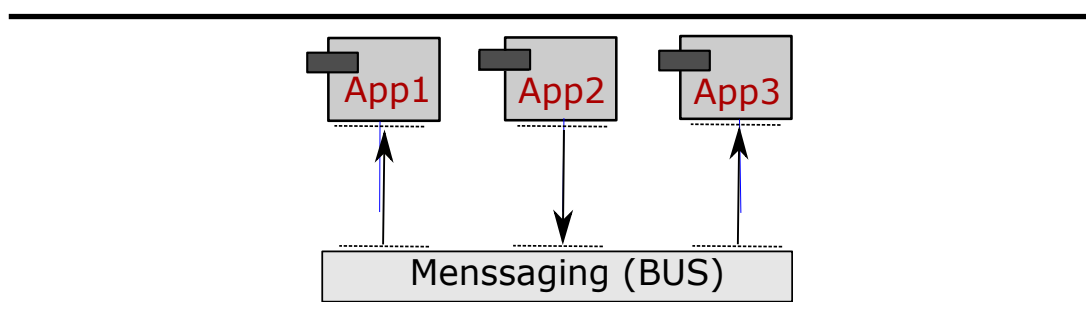


Figura 2.5: Representação abstrata do Sistema de Mensagens.

A alta frequência de mensagens reduz muitos dos problemas de inconsistência que os integradores de transferência de arquivos enfrentam, mas não totalmente. Ainda acontecem problemas de atraso com os sistemas por não haver atualização simultânea. A capacidade de transformar mensagens tem a vantagem de permitir que as aplicações sejam dissociadas

umas das outras, diferente do que ocorre na chamada de procedimento remoto e na transferência de arquivos. Mas essa independência significa que os integradores muitas vezes são obrigados a escrever um aglomerado de código a mais para compensar a dissociação entre as aplicações. Uma vez decidida o uso da integração pelo sistema de mensagens, há uma série de novas questões a serem consideradas e práticas que se devem empregar para esse estilo de integração Hohpe e Woolf [26].

2.1.2 Topologias de Integração

As topologias de integração são formatos de interconectividade entre sistemas que configuram o modo de comunicação a partir de critérios de eficiência. Os tipos são: *Point-to-Point*, *Hub* e *Enterprise Service Bus*. Estas topologias funcionam a partir da ideia de *middleware*.

O *middleware* é qualquer tipo de *software* que permite a comunicação entre dois ou mais sistemas. De fato o *middleware* pode ser visto como um canal no sistema de operação entre aplicações, que permite a comunicação entre dois processos ou pode ser um completo sistema de troca de mensagens com vários mecanismos integrados para dar garantia e fiabilidade de entrega de mensagens. O *middleware* orquestra a troca de informações entre aplicações e faz essa comunicação através dos Protocolos de Informações (PI).

O tipo *Point-to-Point* é pioneiro e acompanhou as primeiras integrações, onde dois sistemas "A" e "B" eram ligados diretamente havendo uma transferência de dados de via única. Se o canal tem múltiplos receptores, apenas um deles pode consumir com êxito uma mensagem particular. Se houver múltiplos receptores tentando consumir uma única mensagem, o canal assegura que apenas um deles vai ter êxito conforme ilustra a Figura §2.6, de modo que não há coordenação entre os receptores Linthicum [30]. O canal ainda pode ter vários receptores para consumir várias mensagens ao mesmo tempo mas, apenas um único receptor consome mensagem num dado momento, isto é, a comunicação é síncrona.

O tipo *Hub* consiste na centralização da lógica de comunicação entre as aplicações. Tradicionalmente chamado de *Hub* faz o papel de gerenciar os estilos de integração usados para o sistema. Todas as aplicações periféricas podem, assim, comunicar-se com todas as outras, transmitindo a estas seus dados, e recebendo através do orquestramento do nó central, conforme ilustra a Figura §2.7. A falha de uma linha de transmissão ligando qualquer nó periférico para o nó central irá resultar no isolamento desse nó

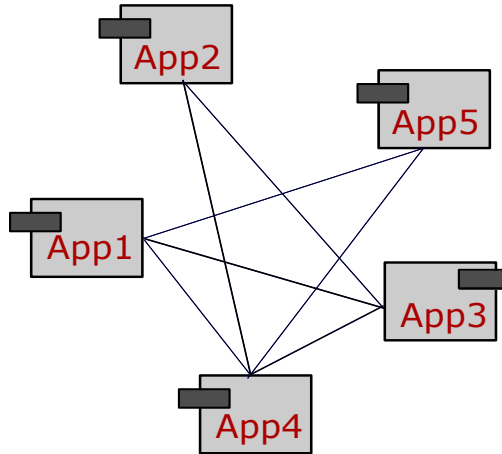


Figura 2.6: Representação abstrata da topologia Point-to-point.

periférico de todos os outros, mas os nós periféricos restantes não serão afetados. No entanto, a desvantagem é que a falha do nó central fará com que todos os nós periféricos falhem também Linthicum [30].

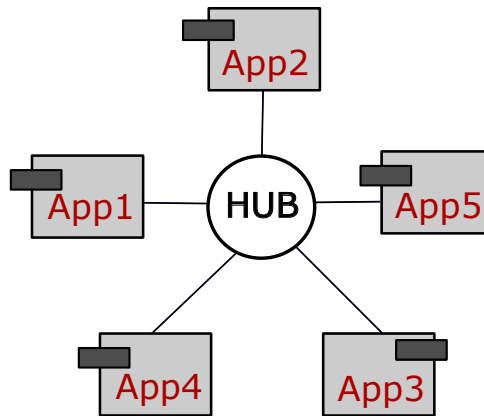


Figura 2.7: Representação abstrata da topologia Hub-and-spoke.

Na intenção de solucionar problemas causados por um *Hub* intermediador, outro modelo de EAI surgiu: o *Bus*. Embora ainda usado como um componente de encaminhamento central, para passar mensagens de sistema para sistema, a arquitetura de barramento procura diminuir a carga de funci-

onalidade colocada sobre um único componente de distribuição por algumas das tarefas de integração de outras partes da rede Linthicum [30]. A medida que EAI baseados em *Bus* evoluíram, foram identificadas uma série de outras funcionalidades necessárias, tais como a segurança de processamento de transações, manipulação de erro etc. Em vez de exigir embutir esses recursos para a lógica de integração central, como teria sido exigido por uma arquitetura de correção, a arquitetura de barramento permite que estas funções sejam locadas em componentes separados conforme ilustra a Figura §2.8.

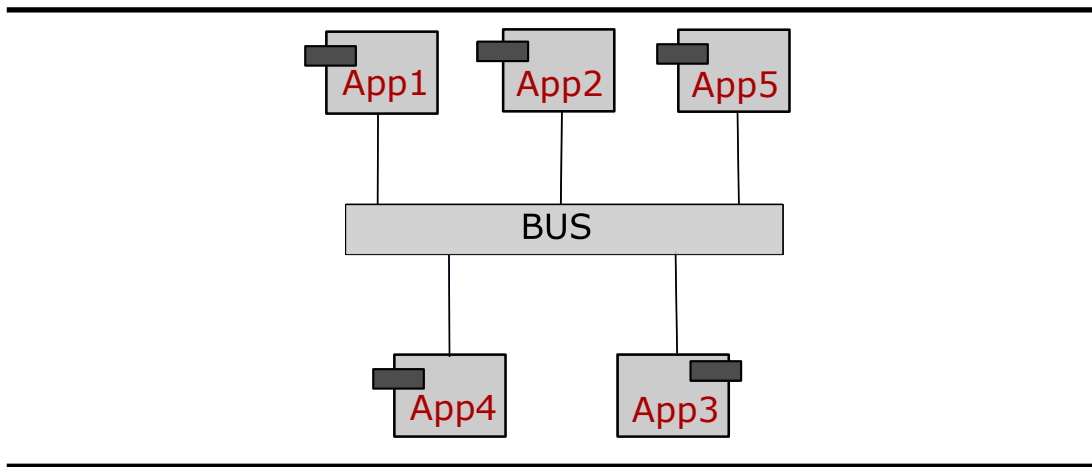


Figura 2.8: Representação abstrata da topologia Enterprise Service Bus.

O resultado final, são soluções leves feitas à medida da integração com maior confiabilidade. As soluções são completamente abstraídas da camada de aplicações e seguem um padrão consistente. Podem ser projetadas e configuradas com código adicional mínimo, implicando em não exigir modificação dos sistemas que precisam ser integrados. Esta versão do modelo EAI baseado no sistema de barramento, veio a ser conhecido como o *Enterprise Service Bus*, ou *ESB*.

2.1.3 Plataformas de Integração

Os modelos conceituais para uma solução de integração são construídos em relação aos sistemas empresariais e tentam descrever os processos reais de troca de informações que necessitam haver entre essas aplicações. As plataformas proporcionam uma Linguagem de Domínio Específico chamada DSL para se desenvolver soluções de integração empresariais. Há nas empresas ou na Nuvem *softwares* motores que executam essas soluções de integração. Os

exemplos de plataforma são o Guaraná DSL Frantz et al. [20], Apache Camel Ibsen e Anstey [27], Mule Sof Dossot et al. [14] e Spring Integration Fisher et al. [18]. Nestas plataformas pode se transcrever modelos conceituais através de uma Linguagem de Domínio Específico e realizar verificações e monitoramento como por exemplo: da memória usada, mensagens sendo processadas, tempo de execução e gargalos de desempenho da solução de integração.

2.1.3.1 Tecnologia Guaraná

A tecnologia Guaraná DSL Frantz et al. [20], é uma plataforma utilizada para projetar soluções de integração de aplicações empresariais em um alto nível de abstração, de forma que os engenheiros de *software* possam manter o foco na criação de modelos para a solução do problema, sem precisar se preocupar com detalhes técnicos de sua implementação. Dentre os principais recursos dessa tecnologia estão uma Linguagem de Domínio Específico (do inglês *Domain-Specific Language* - DSL) e um motor de execução que permite a implementação e execução da solução de integração. O modelo conceitual deste trabalho foi desenvolvido por meio desta tecnologia.

2.2 Redes de Petri

As redes de Petri são uma ferramenta de modelagem conceitual. Conforme Penha [33], foram inventadas em 1962 por Carl Adam Petri como um formalismo para descrever a concorrência e sincronização dinâmica em sistemas distribuídos. Redes de Petri são grafos compostos por dois tipos de nós, lugares e transições. Os lugares são representados por círculos e transições por retângulos, esses nós são conectados por arcos. Lugares podem conter fichas ou *tokens*, representados por pontos dentro do lugar, e a quantidade de fichas em um lugar é chamado de marcação. Um arco de entrada conecta um lugar para uma transição e um arco de saída conecta uma transição para um lugar. Quando uma transição é acionada, os *tokens* nos lugares ligados pelos arcos de entrada são removidos e então são adicionados aos lugares ligados pelos arcos de saída. Uma transição está ativa e pode ser disparada se a quantidade de *tokens* nos lugares satisfazem o peso determinado pelos arcos de entrada. O peso dos arcos de saída não são necessariamente o mesmo peso que os arcos de entrada. Quando uma transição dispara, a rede de Petri muda seu estado, este conceito é denominado conjunto de marcações. As marcações representam a quantidade de *tokens* em cada lugar após uma sequência finita de disparos das transições.

2.2.1 Formalismo Algébrico das Redes de Petri

As Redes de Petri provém do conceito de Grafos, e são definidas como grafos bipartidos pois, apresentam dois tipos de nós, lugares e transições. De um modo formal, as redes de Petri são definidas como:

Definição 1: Uma Rede de Petri é uma Sextupla:

$$RP = (P, T, A_r, K, W, M_0) \quad (2.1)$$

em que:

- $P = \{p_1, p_2, \dots, p_m\}$ é um conjunto finito de lugares;
- $T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições;
- $A_r \subseteq (P \times T) \cup (T \times P)$ é um conjunto arcos;
- $K : P \longrightarrow \mathbb{N} \cup \{\infty\}$ é a função capacidade;
- $W : A_r \longrightarrow \mathbb{N}^+$ é a função Ponderação (Peso);
- $M_0 : P \longrightarrow \mathbb{N}$ é a função da marcação inicial, que satisfaz:

$$\forall p \in P : M_0(p) \leq K(p). \quad (2.2)$$

Pelas Definições da rede de Petri, encontram-se as seguintes condições satisfeitas:

$$P \cap T = \text{Vazio} \quad (2.3)$$

e

$$P \cup T \neq \text{Vazio} \quad (2.4)$$

Pela condição da equação (2.3) os lugares e as transições são nós distintos, o que determina o termo bipartido. Pela condição da equação (2.4) entende-se que em uma rede de Petri existe pelo menos um lugar ou uma transição. Uma rede de Petri é formada por uma estrutura composta por lugares e transições conectados pelos arcos, e uma marcação inicial. Dessa forma, dado um conjunto de lugares com suas respectivas capacidades, um conjunto de transições e um conjunto de arcos ponderados que conectam os lugares às transições e vice-versa, tem-se uma estrutura de rede de Petri, a qual é definida como:

Definição 2: Uma estrutura de rede de Petri é uma rede sem marcação inicial, a qual é denotada por:

$$E_{rp} = (P, T, A_r, K, W) \quad (2.5)$$

2.2.2 Representação Gráfica

Conforme definido anteriormente, as redes de Petri têm uma estrutura de um grafo ponderado, direcionado e bipartido, e apresentam uma marcação inicial. Essa formalização define uma representação gráfica, em que:

- Os lugares são representados por círculos.
- As capacidades dos lugares são representadas junto ao seu respectivo lugar através da igualdade de K ao seu valor. Caso esta representação não seja especificada, o lugar apresenta capacidade infinita.
- As transições são representadas por barras ou por retângulos.
- A marcação atribuída a cada lugar um número inteiro não negativo m que determina que o lugar com marcação m tem m tokens, as quais são representadas por pequenos círculos cheios ou ponto. Matematicamente a marcação é definida através de um vetor $M = [m_1 \ m_2 \ m_3 \ \dots \ m_k]^T$ em que k é o número de lugares da rede ($P = \{p_1, p_2, \dots, p_k\}$), e m_i é o número de tokens contidos no lugar p_i .
- Todos os arcos são direcionados sempre de uma transição para um lugar e de um lugar para uma transição, nunca de transição para transição e de lugar para lugar.
- Os pesos dos arcos são representados através de números próximos aos mesmos, e se não aparece esse número (peso) num arco, então este arco tem peso 1, por definição.

Exemplo 1: na Figura §2.9 é mostrada uma rede de Petri, cujos lugares são os círculos etiquetados por p_1 , p_2 , p_3 e p_4 e as transições são as barras etiquetadas por t_1 , t_2 , e t_3 . Da definição de rede de Petri, tem-se:

- $P = \{p_1, p_2, p_3, p_4\}$;
- $T = \{t_1, t_2, t_3\}$;
- $A_r = \{(p_1, t_1), (p_1, t_2), (p_2, t_1), (p_3, t_2), (p_4, t_3), (t_1, p_3), (t_2, p_2), (t_2, p_4), (t_3, p_2)\}$;
- $K(p_2) = 4; K(p_3) = 5$

- $W(p_1, t_1) = 2; W(p_1, t_2) = 3; W(p_2, t_1) = 1; W(p_3, t_2) = 1; W(p_4, t_3) = 2;$
- $W(t_1, p_3) = 1; W(t_2, p_2) = 2; W(t_2, p_4) = 3; W(t_3, p_2) = 1;$
- $M_0 = [3 \ 1 \ 1 \ 1]^T.$

Os arcos estão definidos por pares (p_i, t_j) ou (t_j, p_i) , que determinam: arco saindo do lugar p_i para transição t_j e arco saindo da transição t_j para o lugar p_i respectivamente.

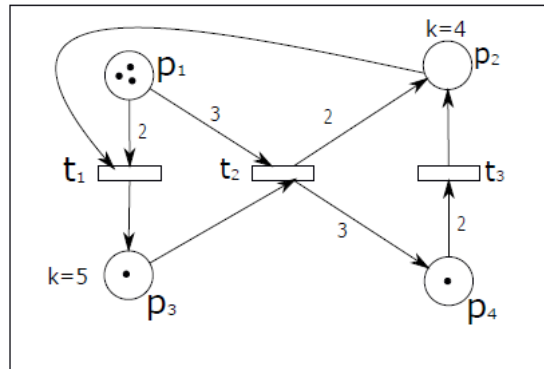


Figura 2.9: Exemplo de uma rede de Petri em sua representação gráfica.

2.2.3 Redes de Petri Coloridas

Segundo Francês [19] as redes de Petri coloridas tem o objetivo de reduzir o tamanho do modelo, sendo de grande valor para a modelagem de sistemas complexos. Este tipo de rede de Petri permite que os *tokens* sejam individualizados, através de cores atribuídas a eles; assim, diferentes processos ou recursos podem ser representados em uma mesma rede. As cores não significam apenas padrões, elas podem representar tipos de dados complexos. Usa-se a nomenclatura de colorida apenas para referenciar a possibilidade de distinção entre os *tokens*. A Figura §2.10 apresenta uma rede colorida que possui a representação original, onde são utilizadas cores para os *tokens*, os arcos são rotulados com cores (a, b, c e d).

Mesmo sendo um pouco rudimentar, a rede de Petri colorida original provê mecanismos que permitem realizar uma escolha determinística.

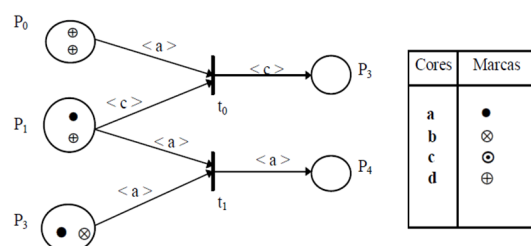


Figura 2.10: Rede de Petri Colorida.

Esse poder de escolha significa uma evolução da rede colorida, porém modificações posteriores vieram dar maior adequação com relação a escolha de representações não-determinísticas Francês [19].

De acordo com Francês [19] as redes de Petri coloridas são compostas de: estrutura, declarações e inscrições. Sendo a estrutura um grafo dirigido por dois vértices (lugares e transições), pode armazenar em cada lugar marcas de tipos diferentes e representar valores associados a tipos de dados mais complexos. Declarações compreendem a especificação dos conjuntos de cores e declarações de variáveis e as inscrições variam de acordo com o componente da rede. Os lugares possuem três tipos de inscrições: nomes, conjunto de cores e expressão de inicialização. As transições possuem dois tipos de inscrições: nomes e expressão guarda, e os arcos apenas um tipo dado pela expressão. Para diferenciar as inscrições, nomes são escritos com letras normais, cores em itálico, expressões de inicialização sublinhadas e as expressões guarda são colocadas entre colchetes conforme ilustra Figura 2.10.

2.2.4 Redes de Petri Temporizadas

As redes de Petri Temporizadas surgiram da necessidade de se modelar sistemas dinâmicos. Conforme o trabalho de Hamilton Fernandes de Mattias [22] existem lógicas associadas às transições de uma rede de Petri Temporizada do tipo: tempo de habilitação (TH), tempo de disparo (TD) e frequência relativa de disparo (FRD). O TH é o valor de tempo que a transição deve permanecer habilitada antes de disparar. O TD é o tempo que a transição leva para disparar. É usado para representar o tempo consumido na execução de eventos em um sistema modelado. O TD pode ser constante ou uma função, se for constante pode ser um valor inteiro, não negativo incluindo o zero. Se for uma função pode assumir valores aleatórios ou de

acordo com alguma distribuição escolhida. O parâmetro de frequência relativa de disparo (FRD) se aplica em transições que pertencem a um conjunto de conflito da rede de Petri, isto é, situações em que mais de uma transição estão aptas a disparar e há necessidade de escolha. A estrutura de uma rede de Petri Temporizada é a mesma que uma rede de Petri, o que muda realmente é que aos *tokens*, os lugares e as transições são associados valores de tempo. Seja RdPT uma rede de Petri Temporizada:

$$\text{RdPT} = (P, T, A_r, K, W, I, M_0) \quad (2.6)$$

em que:

- $P = \{p_1, p_2, \dots, p_m\}$ é um conjunto finito de lugares;
- $T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições;
- $A_r \subseteq (P \times T) \cup (T \times P)$ é um conjunto arcos;
- $K : P \rightarrow \mathbb{N} \cup \{\infty\}$ é a função capacidade;
- $W : A_r \rightarrow \mathbb{N}^+$ é a função Ponderação (Peso);
- $I : T \rightarrow (\mathfrak{R} \cup \{0\}) \times (\mathfrak{R} \cup \{0\})$, onde $I(T_j) = (d_{\min}, d_{\max})$, $d_{\max} \geq d_{\min}$;
- $M_0 : P \rightarrow \mathbb{N}$ é a função da marcação inicial, que satisfaz:

$$\forall p \in P : M_0(p) \leq K(p). \quad (2.7)$$

Uma RdPT é formada por m lugares, n transições, por um conjunto A_r de arcos, por uma função K de capacidade máxima de *tokens* nos lugares, uma função W de ponderação ou peso dos arcos, este peso pode ser constante ou uma função, uma função I do tempo sobre uma transição T_j que leva um valor de tempo pertencente a um conjunto T para duas coordenadas de delay mínimo e delay máximo e uma função marcação inicial M_0 .

2.2.5 Semântica de Disparo de Transição

Uma transição t_j é disparável se estiver habilitada segundo as regras de habilitação:

$$\forall p_i \in P : M[t_j >, M(p_i) \geq I(p_i, t_j). \quad (2.8)$$

Isto é, para todo lugar p_i pertencente ao conjunto de lugares P , a marcação do lugar p_i deve ser maior ou igual ao peso do arco que liga o lugar p_i à

transição t_j . Durante o intervalo $I(t_j)=(d_{j_{\min}}, d_{j_{\max}})$, a t_j só será disparável se: $t(t_j) \geq d_{j_{\min}}$, isto é, se o tempo da transição t_j for maior ou igual ao tempo do seu delay mínimo.

Um exemplo de rede de Petri temporizada determinística é apresentado na Figura §2.11. Nela, as transições t_1 e t_2 possuem tempos associados diferentes, que significa que uma transição será disparada antes da outra. Assim, supondo que $d_1 < d_2$, então o *token* chegará primeiro no lugar P_1 . De maneira determinística é possível estabelecer a ordem em que os eventos devem ocorrer Francês [19].

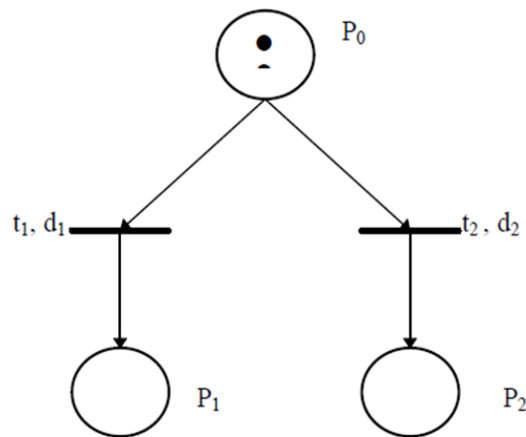


Figura 2.11: Rede de Petri temporizada determinística.

2.3 Sistemas, Modelos e Simulação

Quando se procura representar as interações entre os objetos é necessário estipular padrões de interpretação para que haja sentido no trabalho de pesquisa. Nesta Seção são descritos os conceitos que definem sistemas, modelos e simulação.

2.3.1 Sistemas

Um sistema é um agrupamento de partes que operam juntas para a realização de uma função, é um conjunto de componentes funcionais, entidades, partes ou elementos em que a integração entre esses componentes gera fluxo

de informações, fluxo de matéria, fluxo de sangue, fluxo de energia, enfim, ocorre comunicação entre os componentes de um sistema. Um sistema pode ser definido por suas características de funcionamento Law e Kelton [29]. O funcionamento de um sistema determina uma mudança de estado e esta mudança de estado faz com o sistema exerça suas funções para a troca de informações, matéria, energia e etc. Podemos definir um sistema como: dinâmico ou estático. Um sistema dinâmico muda seu estado através de um acionamento, que pode ser pelo tempo ou por eventos. Se acionado pelo tempo, pode ser contínuo ou discreto no tempo. Se acionado por eventos, estes eventos podem ser determinísticos ou estocásticos. Eventos estocásticos são processos probabilísticos, isto é, são eventos que usam variáveis aleatórias. A Figura §2.12 ilustra as definições de um sistema.

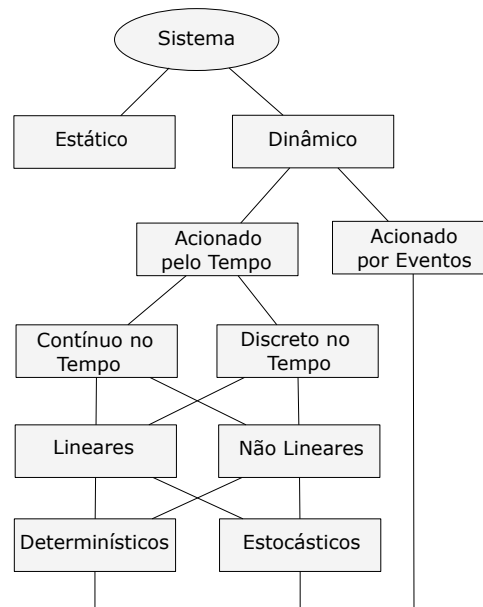


Figura 2.12: Fluxograma das definições de um Sistema Law e Kelton [29].

2.3.2 Modelos de Sistemas

Os modelos representam os sistemas de modo a demonstrar uma ou muitas partes fundamentais de suas características reais. Os modelos podem ser classificados em: **modelos simbólicos**, **analíticos** e **modelos de simulação** Chwif e Medina [9]. Os **modelos simbólicos** são constituídos de símbolos

gráficos (como retângulos e retas) que representam o sistema de forma estática e possuem limitações porque existe falta de informação quantitativa e dificuldades para representar detalhes. Os **modelos analíticos** basicamente podem ser reduzidos a um conjunto de equações que, ao serem resolvidas, permitem obter a solução esperada. São exemplos de modelos analíticos: os modelos de programação dinâmica, as cadeias de Markov e os modelos de redes de fila Chwif [8]. Embora estes modelos forneçam soluções precisas, se o sistema a ser modelado for extremamente complexo, as soluções podem se tornar complicadas e, em muitos casos, têm de ser utilizadas hipóteses simplificadoras para que se resolva o modelo analiticamente. Isto pode levar à perda da validade do modelo, visto que o mesmo não conseguiu representar, satisfatoriamente, a realidade. Os **modelos de simulação**, por sua vez, são implementados através do auxílio de um computador. Assim, como um modelo analítico pode ser representado por um conjunto de equações, um modelo de simulação pode ser representado através de uma linguagem de programação. Para tal, ao contrário dos modelos analíticos, os modelos de simulação são executados, ao invés de solucionados. As desvantagens dos modelos de simulação são: podem ser difíceis de construir e podem levar a resultados menos precisos do que os modelos analíticos Chwif [8]. A Figura 2.13 ilustra as definições de modelagem de um sistema.

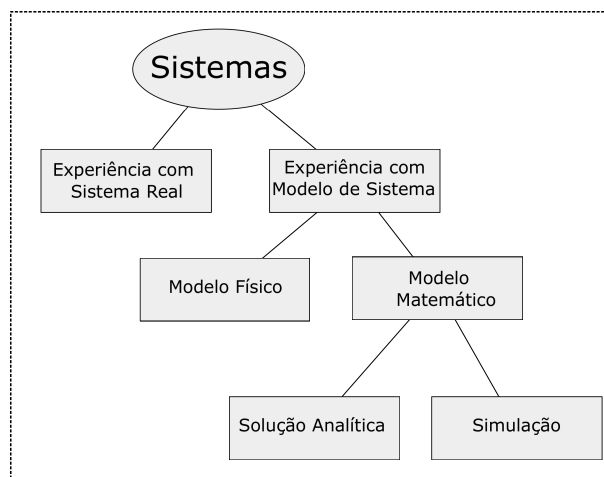


Figura 2.13: Fluxograma do processo de Modelagem de um Sistema Law e Kelton [29].

Um modelo de simulação é construído a partir dos objetivos da simulação. Estes podem ser traduzidos pelas medidas de desempenho do sistema,

que são as variáveis de saída de interesse de um modelo de simulação. Deste modo, considerando um exemplo de um modelo de simulação de uma linha de montagem, este pode ter como objetivo a produtividade e a utilização da mão-de-obra. Portanto, as medidas de desempenho são a produção horária que sai da linha e as taxas de utilização de cada recurso humano na mesma. Uma outra classificação pode ser feita em relação às variáveis com que os modelos de simulação trabalham. Os modelos de simulação que utilizam variáveis aleatórias são denominados Modelos Estocásticos, enquanto que os Modelos Determinísticos trabalham somente com variáveis do tipo não probabilísticas, e o resultado da simulação é sempre o mesmo, não importando quantas vezes se repita o teste sobre o modelo. A grande maioria dos modelos de simulação são constituídos por variáveis estocásticas.

2.4 Simulação Computacional

A simulação, segundo Aalst (2015), citado em Sawicki et al. [39] é composta por onze fases, sendo elas: definição do problema, modelagem, modelo conceitual, realização, modelo executável, verificação e validação, modelo validado, experimentando, resultados da simulação, interpretação e respondendo as soluções. A definição do problema precisa considerar as metas para saber o que vai e o que não vai fazer parte da simulação. A modelagem, é onde o modelo conceitual é elaborado com as propriedades mais importantes do sistema. A fase de realização consiste em transformar o modelo conceitual em um modelo executável. O modelo executável pode ser simulado no computador, por isso necessita de uma ferramenta de simulação. O próximo passo é verificar e validar o modelo, comparando com o modelo real. Os resultados da simulação precisam ser interpretados para permitir um entendimento sobre o problema. E, por fim, é realizado um relatório com as respostas para as questões da definição do problema no qual as propostas de soluções são criadas Sawicki et al. [39].

A simulação computacional pode ser classificada basicamente em três categorias: simulação monte carlo, simulação contínua e simulação de eventos discretos, de acordo com a taxonomia proposta por Chwif [8]. A simulação monte carlo é aquela onde um problema, notadamente não probabilístico, é solucionado através de um processo estocástico. Nesta, inexistente uma representação explícita do tempo. Já tanto a simulação contínua como a simulação de eventos discretos levam em conta o tempo. A primeira é utilizada para modelar sistemas cujo estado varia continuamente no tempo, como no exemplo de uma xícara de chá aquecendo. A simulação contínua se utiliza de equações diferenciais para o cálculo das mudanças das variáveis de estado ao

longo do tempo Chwif [8]. Por outro lado, a simulação de eventos discretos é utilizada para modelar sistemas que mudam o seu estado em pontos discretos no tempo, a partir da ocorrência de eventos, como é o caso das soluções de integração. Soluções de integração podem ser caracterizadas como sistemas estocásticos porque a taxa de chegada de seus dados de entrada não podem ser previstas e são essencialmente aleatórias, totalmente dependente das aplicações que estão sendo integradas. Modelos dinâmicos representam sistemas que alteram seu estado ao longo do tempo. O estado em uma solução de integração de mudanças como resultado do número de mensagens que está sendo processado em um tempo de execução particular, e as operações executadas nas mensagens Sandro Sawiki [37]. Modelos discretos são orientados a eventos e assim usados para modelar sistemas que mudam seu estado em momentos distintos no tempo a partir da ocorrência de eventos. Soluções de integração podem ser caracterizados como sistemas discretos pois, todos os componentes envolvidos em uma solução de integração consomem um tempo de execução específico quando ocorre um evento. Como um sistema discreto, o modelo conceitual projetado para uma solução de integração pode ser simulado.

2.5 Plataformas de Simulação de redes de Petri Temporizadas

Na simulação computacional é necessário uma plataforma compatível com os componentes das rede de Petri Temporizadas. Inicialmente o projeto analisou alguns *softwares* usados em alguns trabalhos relacionados, para aplicar as simulações do modelo em redes de Petri Temporizadas. Primeiramente foi usado o *software* PIPE pip [1]. A plataforma do PIPE mostrou-se limitada em relação as sintaxes do modelo no Guaraná DSL. Então foi selecionado a plataforma CPN Tools. O CPN Tools é uma Plataforma de livre acesso projetada para edição, simulação e análise de malhas coloridas de redes de Petri. Segundo cpn [2] a ferramenta possui verificação de sintaxe incremental e geração de código, que ocorrem enquanto uma rede de Petri está sendo construída. Um simulador rápido trata de forma eficiente redes coloridas e também de temporizadas. Os dados de mudança de estado são gerados e analisados, e um relatório de espaço é fornecido contendo os resultados das simulações da rede de Petri programada. A Ferramentas do CPN Tools são bastante completas e suportam uma grande diversidade de tipos de rede de Petri. Originalmente o CPN Tools foi desenvolvido pelo Grupo CPN na Universidade de Aarhus, de 2000 a 2010. Os principais arquitetos por trás da ferramenta são Kurt Jensen , Soren Christensen , Lars M.

Kristensen , e Michael Westergaard . A partir do outono de 2010, o CPN Tools foi transferido para o grupo AIS, Universidade de Tecnologia de Eindhoven , Holanda cpn [2].

2.6 Resumo do Capítulo

Nesse capítulo, foram introduzidos os conceitos de integração de aplicações empresariais, redes de Petri Temporizadas, sistemas, modelos e simulação. Estes conteúdos apresentaram as definições necessárias para realizar o trabalho de pesquisa. As principais definições se referem aos estilos de integração, as topologias de integração, o formalismo algébrico das redes de Petri Temporizadas, os tipos de modelo de simulação e as plataformas de simulação de redes de Petri Temporizadas. Todos estes conceitos pesquisados serão usados para construir o modelo de simulação e analisar os resultados oferecidos.

Capítulo 3

Trabalhos Relacionados

To understand is to perceive patterns.

Sir Isaiah Berlin, British social & political theorist (1909-1997)

Os trabalhos de modelagem em redes de Petri Temporizadas de sistemas discretos desenvolvidos por outros pesquisadores são fundamentais para a elaboração de novas aplicações dessa técnica. Este capítulo descreve o trabalho de autores que aplicaram em suas pesquisas conceitos importantes em redes de Petri Temporizadas que fornecem referências para a elaboração desta pesquisa.

3.1 Trabalhos relacionados envolvendo Redes de Petri Temporizadas

O trabalho apresentado por Santos et al [13] descreve um método que utiliza redes de Petri Temporizadas para modelar sistemas flexíveis de manufatura e realizar análises sobre a taxa de falhas que, teoricamente, não levam em consideração a ocorrência de falhas no fluxo da produção das indústrias automobilísticas que em sua grande maioria tem seus processos automatizados. As influências das falhas são reconhecidas e analisadas através de um modelo em redes de Petri Temporizadas em comparação com o sistema real. As redes de Petri Temporizadas são úteis para obtenção do tempo ótimo na performance do processo de produção de uma linha de montagem de veículos que são formadas por inúmeras etapas de trabalho. Estes trabalhos levam tempo para serem realizadas e o gerenciamento destas etapas necessitam de

um modelo que proporcione dados confiáveis para a melhoria dos setores envolvidos nesses processos de fabricação. Os resultados obtidos mostraram que a criação de um modelo em redes de Petri Temporizadas é imprescindível na análise da performance temporal de um sistema de manufatura.

Mattias [22] propõe uma Metodologia para análise de desempenho de Sistemas de Computação descritos através de redes de Petri Temporizadas Estendidas. Este trabalho descreve a estrutura lógica das redes de Petri Temporizadas e como podem ser aplicadas em modelos de computação distribuídos. Estes sistemas são dependentes do tempo, por essa razão é possível descrever o comportamento dinâmico através de redes de Petri Temporizadas. Os autores utilizaram, como caso de estudo, uma agência locadora de automóveis e configuram o tempo de locação utilizados pelos seus clientes. Este trabalho buscou dimensionar o número de carros necessários para atender adequadamente seus clientes.

Quental [35] propõe um modelo em redes de Petri Temporizadas do modelo da NAS Parallel Benchmarks e realiza a análise de desempenho através da ferramenta *Microsoft Office Excel* utilizando variáveis aleatórias com resultados de desvio padrão, coeficiente de variação e médias do processamento paralelo do modelo. Segundo o autor, o processamento paralelo é a capacidade de sistemas computacionais de executarem várias ações simultaneamente com o objetivo de obter a solução de um problema de maneira mais rápida. Estes sistemas exigem variáveis de tempo de resposta e de execução compatíveis com as Redes de Petri Temporizadas.

Carvalho [23] aplica o formalismo de redes de Petri Temporizadas na modelagem e simulação de linhas de produção em forma de U com trabalho multifuncional. Segundo o autor, as redes de Petri constituem uma ferramenta bastante eficiente no controle de parâmetros produtivos, o que pode resultar no aumento da produtividade dos operadores polivalentes pelo dimensionamento adequado de estoques intermediários. Os resultados obtidos neste trabalho mostraram que a existência de estoque intermediário e as quantidades relativas ao estoque estão diretamente ligadas ao tempo de execução do processo produtivo.

Gomes [6] apresenta as redes de Petri Temporizadas e Coloridas como uma ferramenta gráfica na modelagem de sistemas de produção nos quais os recursos em especial matéria primas sofrem transformações ou beneficiamentos ao longo da cadeia produtiva e apresenta um algoritmo que permite calcular para os sistemas produtivos as datas mais cedo e mais tarde sensibilização de uma transição; as datas mais cedo e mais tarde de entrada de

recursos nas posições e o caminho de mais longa duração entre a fonte e o sumidouro do sistema.

Foyo [32] aplica os conceitos de redes de Petri Temporizadas apresentando a evolução do ambiente GHENeSys que permite a implementação de processos de verificação com e sem restrição de tempo. Descreve o formalismo subjacente aos modelos, baseados em redes de Petri estendidas e condizentes com o padrão ISO/IEC. Descreve genericamente o algoritmo de geração das classes de estado enfatizando uma abordagem que não é inteiramente algébrica. Mostra como é inserido o tempo denso como parâmetro, sua complementaridade com o tempo estático e com transições instantâneas e estados e lugares que podem em princípio reter a marcação por um tempo ilimitado, e, finalmente a forma como isto foi implementado.

Guerra [11] trata essencialmente de uma extensão temporizada para Redes de Petri Orientadas a Objetos (RPOO). RPOO é um formalismo que se propõe a fazer com que várias das características das Redes de Petri e da Orientação a Objetos complementem-se e contribuam positivamente para a construção, simulação e validação de modelos. A extensão temporizada proposta promove a integração do tratamento temporal tanto em nível de Orientação de Objetos quanto em nível de redes de Petri. Como resultado, a extensão dá maior expressividade ao formalismo, no sentido de poder modelar e analisar com maior facilidade algumas classes de problemas que apresentam restrições temporais.

Salvi [36] apresenta os relacionamentos entre problemas de planejamento temporal em inteligência artificial e alcançabilidade em redes de Petri com tempo. Estas relações são dadas pela tradução de problemas de um formalismo para outro. Tais traduções proporcionam o uso dos métodos formais de análise das redes de Petri e também os sofisticados algoritmos de busca da área de planejamento automático para solucionar problemas que fazem uso informações temporais. É comparado o poder de expressividade das transições temporais e temporizadas de uma rede de Petri com uma ação durativa em PDDL, demonstrando que as duas primeiras têm poder de expressividade reduzido em relação à última. Também é apresentada uma rede de Petri temporal capaz de simular a execução de um conjunto de ações durativas descritas pela linguagem PDDL.

Machado [31] diagnosticou a cadeia logística de suprimento de madeira de uma empresa para se conhecer os principais gargalos existentes ao longo da cadeia. O modelo da cadeia logística de transporte de madeira utilizou a rede de Petri Temporal, com transições temporizadas, por trabalhar

com dois tempos que correspondem a uma duração de sensibilização. Os tempos nos elementos do ciclo operacional do transporte de madeira visou otimizar os recursos, garantir um fluxo contínuo de abastecimento e assegurar um volume adequado de entrada de madeira na fábrica. Os resultados apresentaram redução dos tempos dos ciclos de transporte de madeira em relação à situação atual da cadeia logística.

Albuquerque [3] apresenta um sistema de diagnóstico de falhas e um método de especificação de sistemas digitais. O sistema de diagnóstico é composto principalmente, por uma ferramenta de geração automática do modelo global, por um simulador de Redes de Petri e por um módulo de diagnóstico. A utilização de redes de Petri Temporizadas possibilita a modelagem de sistemas que tenham atividades assíncronas, paralelas, concorrentes, não determinísticas e temporizadas. Para obtenção dos modelos foi desenvolvido um método de especificação de sistemas digitais que parte da descrição comportamental de cada componente, através de regras operacionais. A partir das regras operacionais, são construídas as redes de Petri elementares que modelam cada componente do sistema. Os modelos obtidos são agrupados em um arquivo para formar uma biblioteca de componentes, utilizada pela ferramenta de geração automática da rede de Petri global. Esta ferramenta é parte integrante do sistema de diagnóstico de falhas e obtém o modelo global através da técnica de fusão de lugares comuns.

Aguiar [12] propõe uma metodologia *top-down* sistematizada, com instanciamento de sub-redes, utilizando técnicas de composição para manutenção de certas propriedades de interesse, visando subsidiar engenheiros no processo de construção e análise de modelos que representem Sistemas de Manufatura Flexíveis (FMS) reais. Usando redes de Petri Coloridas, Hierárquicas e Temporizadas o trabalho cria um modelo de simulação na ferramenta CPN Tools e faz uma análise de desempenho destes sistema de manufatura por simulação para identificação de propriedades como taxa de produção, gargalos de produção e relações de causa e efeito na parametrização da produção.

3.2 Resumo do Capítulo

Nesse capítulo, foram introduzidos os trabalhos relacionados com a pesquisa. Os autores que realizaram trabalhos aplicando principalmente as redes de Petri Temporizadas são apresentados e usados como referência para conduzir esta pesquisa. É importante observar os aspectos utilizados pelos autores na modelagem de diversos sistemas, e que podem ser aplicados neste

trabalho para a proposta de um modelo de simulação em redes de Petri Temporizadas.

Capítulo 4

Estudo de Caso

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

Chinese proverb

O estudo de caso utilizado nessa pesquisa se refere a uma solução de integração para o agendamento de viagens desenvolvida por meio da tecnologia Guaraná. Este sistema é representado por um modelo conceitual, o qual é composto por cinco aplicações heterogêneas, sendo elas: Sistema de Viagens, Sistema de Fatura, Servidor de Email, Sistema de Vôos e Sistema de Hotéis.

4.1 Ecossistema de Software

A solução de integração utilizada nessa pesquisa é composta por cinco aplicações heterogêneas: Sistema de Viagens, Sistema de Fatura, Servidor de Email, Sistema de Vôos e Sistema de Hotéis. Tais sistemas distintos fazem parte de um ecossistema de *software*. O sistema de viagens é uma aplicação que a Agência de Viagens utiliza para registrar informações sobre seus clientes e pedidos de reserva. O sistema de fatura é uma aplicação separada que permite que os clientes paguem suas viagens utilizando os seus cartões de crédito. O servidor de email é usado para fornecer aos clientes informações sobre suas reservas. Os voos e o hotel representam interfaces que permitem reservar voos e hotéis. Todos estes sistemas são aplicações que foram concebidas sem levar em conta a integração, assim a solução de integração deve interagir com eles por meio de sua camada de dados. Cada

reserva registrada no Sistema de Viagens contém todas as informações necessárias sobre o pagamento, o voo, o hotel e um localizador de registro que identifica exclusivamente a reserva. A solução de integração deve, periodicamente, sondar o Sistema de Viagens para novas reservas, para que voos e hotel possam ser reservados, possibilitando o cliente realizar a fatura e receber um email contendo todas informações sobre suas viagens.

4.2 Modelo Conceitual de Integração

O Modelo Conceitual utilizado nessa pesquisa, conforme Figura 4.1, representa o formato estático da solução de integração desenvolvida no Guaraná DSL o qual descreve a comunicação entre as tarefas.

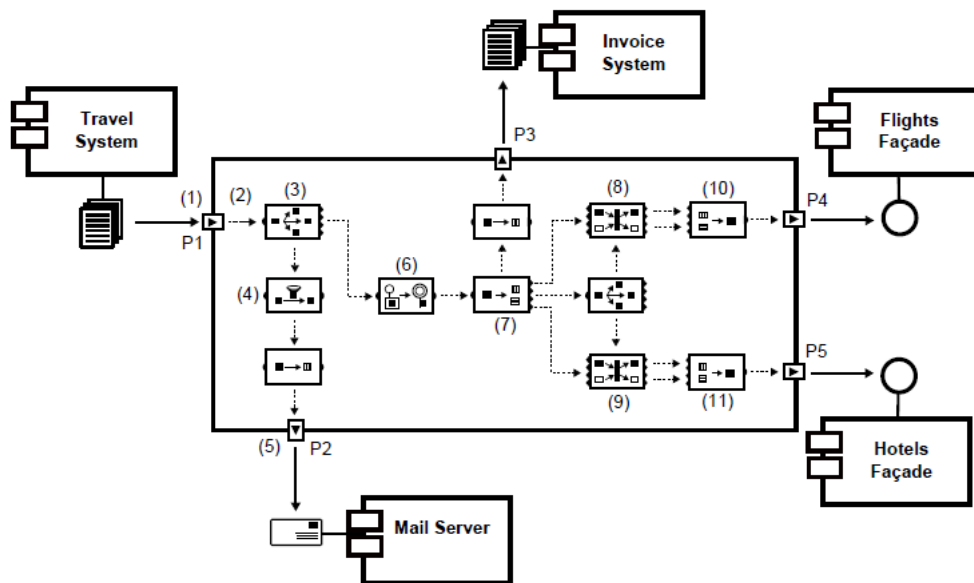


Figura 4.1: Solução de Integração desenvolvida pela tecnologia Guaraná. Frantz [21]

O fluxo de trabalho começa pela porta de entrada P1. O Sistema de Viagens utiliza esta porta, periodicamente, para encontrar novas reservas. Para cada reserva, a porta de entrada P1 insere uma mensagem para o sistema, que é, por sua vez, adicionada ao slot (2). A Tarefa (3) do Replicador

recebe mensagens do *slot* (2) e faz uma cópia que é enviada ao e-mail do cliente e a outra é usada para preparar a fatura, a reserva do hotel e do voo. A primeira cópia passa pela tarefa Filtro (4). Essa tarefa seleciona mensagens que contenham os dados necessários para serem enviadas ao cliente, o que impede a porta de saída P2 de enviar mensagens sem um destino ou endereço de e-mail corretos. As tarefas de tradução foram usadas no processo para adequar mensagens aos esquemas com as quais as aplicações de fatura e de email são integradas. A tarefa (6) separa as mensagens em duas diferentes, em uma das mensagens usa o localizador de registro, que está no cabeçalho da mensagem, para identificar a mensagem. A mensagem será identificada pelas tarefas (8) e (9) que são os correlacionadores de mensagens. A segunda cópia é separada de modo que as informações de pagamento, contidas no corpo da mensagem, vão para o Sistema de Fatura. As mensagens correlacionadas seguem para as tarefas de (10) e (11), as quais são montadas e utilizadas na reserva dos voos e o hotéis, respectivamente. Utilizando este modelo conceitual na tecnologia Guaraná é que se produz um modelo de simulação a ser experimentado nas seções seguintes.

Capítulo 5

Modelo de Simulação Proposto

Knowing is not enough; we must apply.

Willing is not enough; we must do.

Johann W. von Goethe, German Novelist (1749-1832)

Este capítulo apresenta um modelo de simulação em redes de Petri Temporizadas projetado para obtenção de resultados sobre a qualidade de serviço de uma solução de integração no Guaraná DSL. A solução gerada no Guaraná DSL integra aplicações de uma Agência de Viagens. É realizada uma correspondência entre os elementos que constituem a solução de integração com os recursos matemáticos disponíveis nas redes de Petri Temporizadas para que o modelo reflita as operações desempenhadas pela solução de integração. Ao final são realizados os testes experimentais e as análises dos resultados.

5.1 Modelo em rede de Petri Temporizadas da Solução de Integração no Guaraná DSL

As redes de Petri Temporizadas são constituídas por transições, lugares, arcos e *tokens*. Com estes elementos é possível fazer uma correspondência com os elementos gráficos do Guaraná DSL que se constitui por oito elementos conforme Figura §5.1. Analisando o modelo conceitual e reconhecendo todas as partes envolvidas no ecossistema de *software* da solução de integração da Agência de Viagens, foi possível reproduzir um modelo de simulação

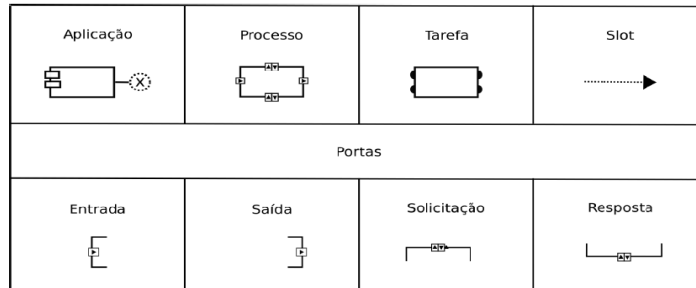


Figura 5.1: Notação gráfica do domínio do Guaraná DSL Frantz et al. [20].

em redes de Petri Temporizadas determinando as equivalências entre as funcionalidades de uma tarefa com uma transição, de um *slot* com um lugar e arcos de conexões e uma mensagem com um *token* conforme ilustra a Figura §5.2. Uma aplicação pertencente ao ecossistema da solução de integração é um recipiente de informações, logo foi classificada como sendo um lugar. Como mencionado, soluções de integração são compostas, basicamente, por



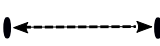



Guaraná DSL	RDP Temporizadas
 Tarefa	 Transição
 Slot	 Lugar e Arcos
 Mensagem	 Token

Figura 5.2: Equivalência entre elementos do Guaraná DSL e Redes de Petri Temporizadas Fabricia Roos-Frantz [17].

tarefas, *slots*, portas, processos e aplicações. As tarefas são funções que modificam as mensagens e fazem com que estas fluam pelo sistema. Os *slots* são os canais de conexão entre as tarefas, armazenam as mensagens processa-

das pela tarefa anterior e são ponto de armazenamento ante o processamento da tarefa posterior. As portas são conectores entre as aplicações e a solução de integração. As aplicações são *softwares* formadores e receptores de mensagens. Nas redes de Petri Temporizadas, os lugares e os arcos fazem a função correspondente aos *slots* da linguagem do Guaraná DSL. Os lugares são recipientes de armazenamento de *tokens*. Os arcos fazem a ligação entre lugares e transições e indicam o sentido do fluxo de mensagens e possuem valor unitário. As transições disparam *tokens* pelo sistema e são responsáveis pela mudança de estado. A representação da mudança de estado de um sistema em redes de Petri Temporizadas em comparação ao fluxo de mensagem no Guaraná DSL, ilustrado na Figura §5.3, evidencia a equivalência no comportamento dinâmico destes dois tipos de sistemas de eventos discretos, mostrando a passagem de uma informação, que em dois estados distintos estão em diferentes lugares. A Figura §5.4 mostra todos os elementos que

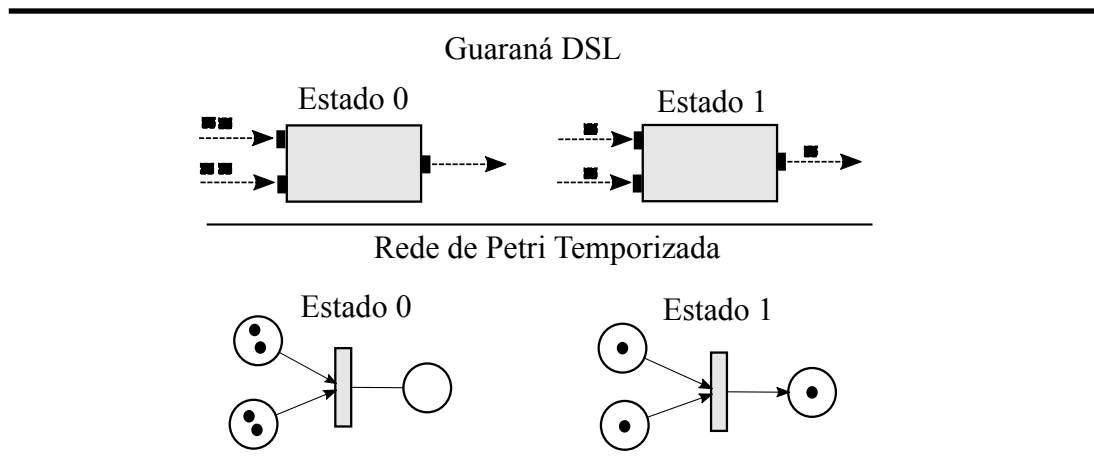


Figura 5.3: Equivalência entre mudanças de estado do Guaraná DSL e redes de Petri Temporizadas Fabricia Roos-Frantz [17].

compõem a solução de integração estudada nesse capítulo.

5.1.1 Formulação Matemática

Usando estas informações foi desenvolvido o modelo de simulação em redes de Petri Temporizadas conforme Figura §5.5, que descreve o ecossistema da solução de integração no Guaraná DSL. Seja RdPT a rede de Petri Temporizada, então:

$$\text{RdPT} = (P, T, A_r, K, W, I, M_0) \tag{5.1}$$

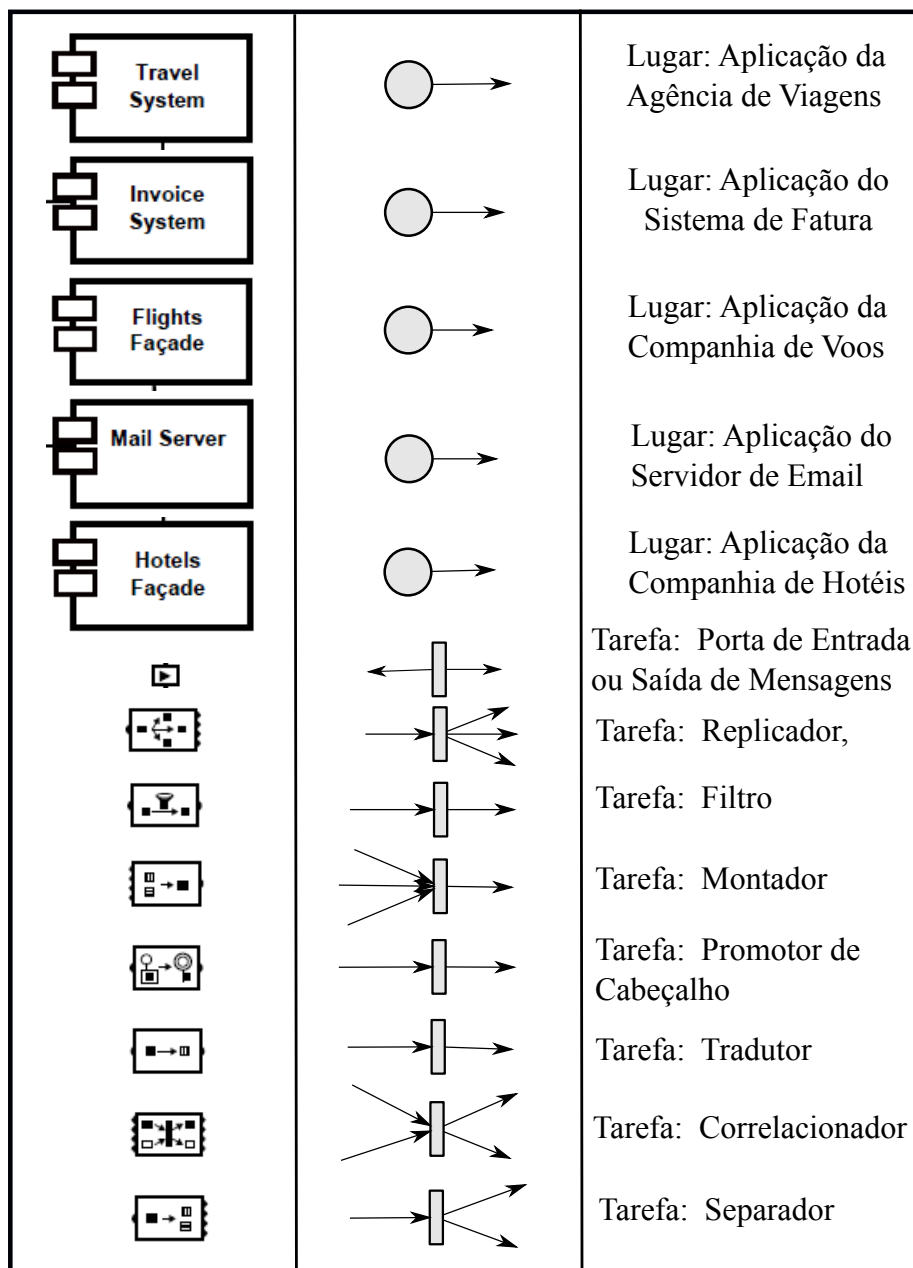


Figura 5.4: Elementos que constituem a solução de integração da Agência de Viagens [21].

$$P = \{TS, MS, IS, V, H, (2), \text{slot1}, \text{slot2}, \text{slot3}, \text{slot4}, \text{slot5}, \text{slot6}, \\ \text{slot7}, \text{slot8}, \text{slot9}, \text{slot10}, \text{slot11}, \text{slot12}, \text{slot13}, \text{slot14}, \text{slot15}, \\ \text{slot16}, \text{slot17}, \text{slot18}\}; \quad (5.2)$$

$$T = \{P1(1), (3), (4), \text{tradutor}(1), \text{tradutor}(2), \text{Replicador}, P2(5), \\ (6), (7), (8), (9), (10), (11), P3, P4, P5\}; \quad (5.3)$$

$$A_r \subseteq (P \times T) \cup (T \times P) \text{ é um conjunto arcos}; \quad (5.4)$$

$$K: P \longrightarrow \mathbb{N} \cup \{\infty\} \text{ é a função capacidade}; \quad (5.5)$$

$$W: A_r \longrightarrow \mathbb{N}^+ \text{ é a função Ponderação (Peso)}; \quad (5.6)$$

$$I: T \longrightarrow (\mathfrak{R} \cup \{0\}) \times (\mathfrak{R} \cup \{0\}), \text{ onde } I(T_j) = (1, 5); \quad (5.7)$$

$$M_0 : P \longrightarrow \mathbb{N} \text{ é a função da marcação inicial, que satisfaz:} \quad (5.8)$$

$$\forall p \in P : M_0(p) \leq K(p). \quad (5.9)$$

5.2 Experimentação

O experimento é a etapa em que são obtidas informações do comportamento do modelo de simulação. Trata-se de fazer a transcrição do modelo de simulação em redes de Petri Temporizadas para a plataforma CPN Tools, caracterizar os cenários de simulação, realizar os testes destes cenários, responder as questões relacionadas aos objetivos do trabalho utilizando as informações oriundas dos testes. Foi utilizada a ferramenta CPN Tools para realizar os experimentos com o modelo de simulação em redes de Petri temporizadas.

5.2.1 Modelagem do Sistema e a Plataforma CPN Tools

O modelo de simulação em redes de Petri Temporizadas transcrito na plataforma CPN Tools descreve o modelo formal da Figura §5.5. A Figura §5.6 mostra de uma forma geral a modelagem na linguagem do CPN Tools juntamente das sintaxes de programação que levam em consideração as características de funcionamento de cada elemento no Guaraná DSL. Quando se declara um *token* no CPN Tools, conforme mostra a Figura §5.7, deve-se determinar: o tipo do lugar (Inteiro, String, Booleano, Temporizado, Real),

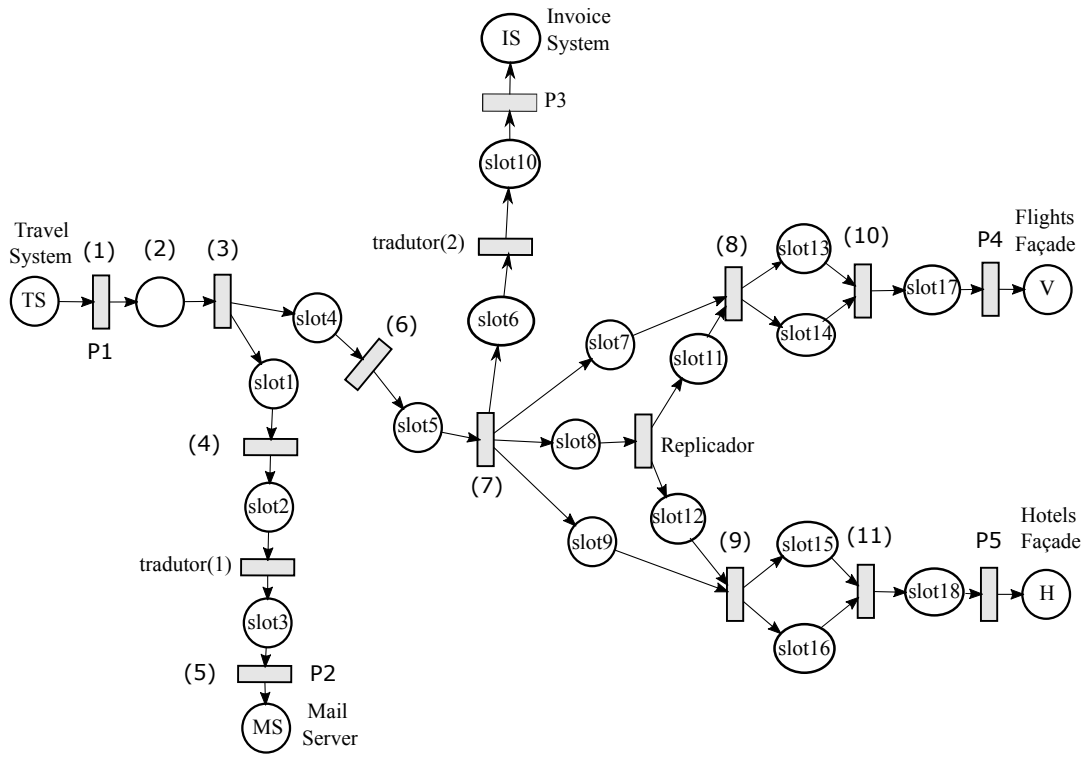


Figura 5.5: Modelo em Rede de Petri Temporizada da solução de integração no Guaraná DSL.

a cor do *token*, que está relacionada diretamente com o tipo do lugar e a variável dos arcos, que suportarão somente *tokens* do tipo declarado ao lugar. No modelo de simulação desta pesquisa foi declarado um lugar denominado *Temp*, mensagem do tipo temporizada formada por um produto cartesiano de três lugares: o lugar *mgs* temporizado, que contém *tokens* de cor *a* do tipo temporizado e arcos de variável *x*. O lugar *Tempo* que contém *tokens* de cor inteira com valores que variam de 0 até 1.000.000 (um milhão), este valor serve para suportar uma quantidade de tempo indefinida, mas com uma margem bem superior a qualquer possível valor inteiro que um *token* pode assumir em relação à 100.000 disparos, e arcos de variável *k*. O lugar *cod* contém *tokens* de cor inteira com valores que variam de 0 até 1.000.000 (um milhão) e arcos de variáveis *j* e *m*. O lugar *Temp* gera *tokens* do formato $1'(a,k,j)@+\text{delay}$. A cor de um *token* é única, representada pela letra *a* pois, o foco do trabalho é explorar exclusivamente as propriedades das redes de Petri Temporizadas. As outras coordenadas *k* e *j* são

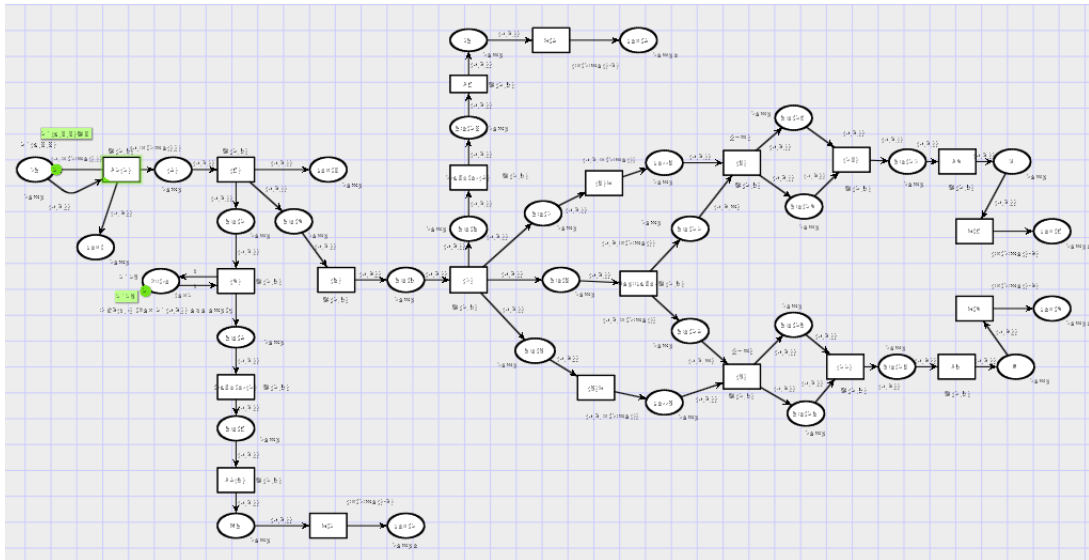


Figura 5.6: Modelo em rede de Petri Temporizadas transcrito na linguagem do CPN Tools.

auxiliares, servem para a sintaxe de programação das funcionalidades das tarefas envolvidas no modelo de simulação.

A Figura §5.8 mostra o início do modelo de simulação com o lugar TS (Sistema de Viagens) contendo um *token*. Em seguida tem-se a transição P1(1) representando a porta de entrada de mensagens do Guaraná DSL. Esta transição irá habilitar o fluxo de novos *tokens* para dentro do sistema. Os arcos de ligação $(x, intTime(), j)$ permitem a passagem de *tokens* de cor a , na coordenada x . Na coordenada k , é aplicada a função $intTime()$ que transforma o tempo atual dos *tokens* em um valor inteiro. Esta função serve para fixar na coordenada k , o tempo inicial dos *tokens* que ingressam no sistema. Após ingressarem, os *tokens* são distribuídos para dois lugares: no primeiro são armazenados no lugar *cont* onde serão contabilizados, no segundo passam para o *slot(2)* e aguardam serem disparados pela transição (3). A transição (3) representa a tarefa do Replicador de mensagens do Guaraná DSL. Quando a transição (3) dispara, envia *tokens* para três lugares: no primeiro lugar denominado *cont0*, os *tokens* são armazenados para serem contabilizados juntamente com os *tokens* que chegam ao lugar Email (MS). No segundo e terceiro lugar os *tokens* passam para os *slots* 1 e 4, conforme é ilustrado na Figura §5.8. Os *tokens* armazenados no *slot1* se destinam ao lugar

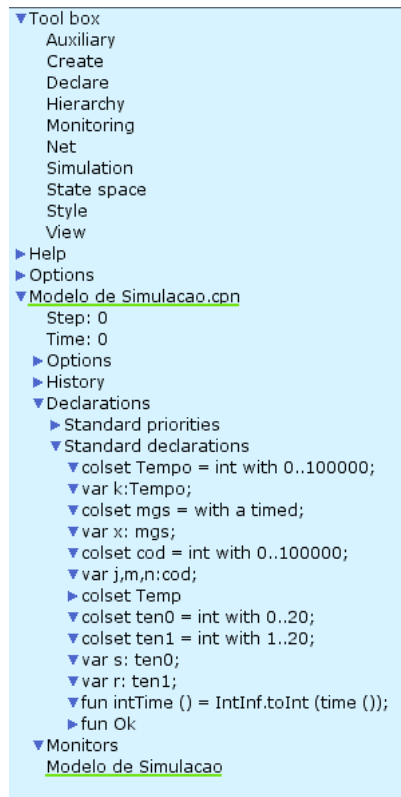


Figura 5.7: Declaração dos lugares, tokens, variáveis e funções na linguagem do CPN Tools.

Email (MS). Os *tokens* armazenados no lugar *slot4* são disparados até a transição (7). A transição (7) distribui os *tokens* para três destinos. Um *token* segue para o lugar Fatura (IS), outro para o lugar Voos (V) e o terceiro para o lugar Hotel (H). Existem quatro detalhes relacionados à configuração do modelo transcrito no CPN Tools que o diferencia do modelo formal em redes de Petri Temporizadas. O primeiro detalhe se refere a coordenada de variável *k*. O processamento de disparos das transições temporizadas no CPN Tools obedece a prioridade de que: primeiramente são habilitadas transições que estão ligadas aos lugares contendo os *tokens* de menor tempo, sendo assim, à coordenada *k* é anexado um valor inteiro, este valor é extraído do tempo inicial de todos os *tokens* que são inseridos no sistema. Isto é possível devido a função *intTime()*, que guarda o tempo atual deste *token*, transforma-o em um número inteiro e anexa na coordenada *k*. Este tempo permanece com o *token* até o lugar de chegada, para que no final, esse va-

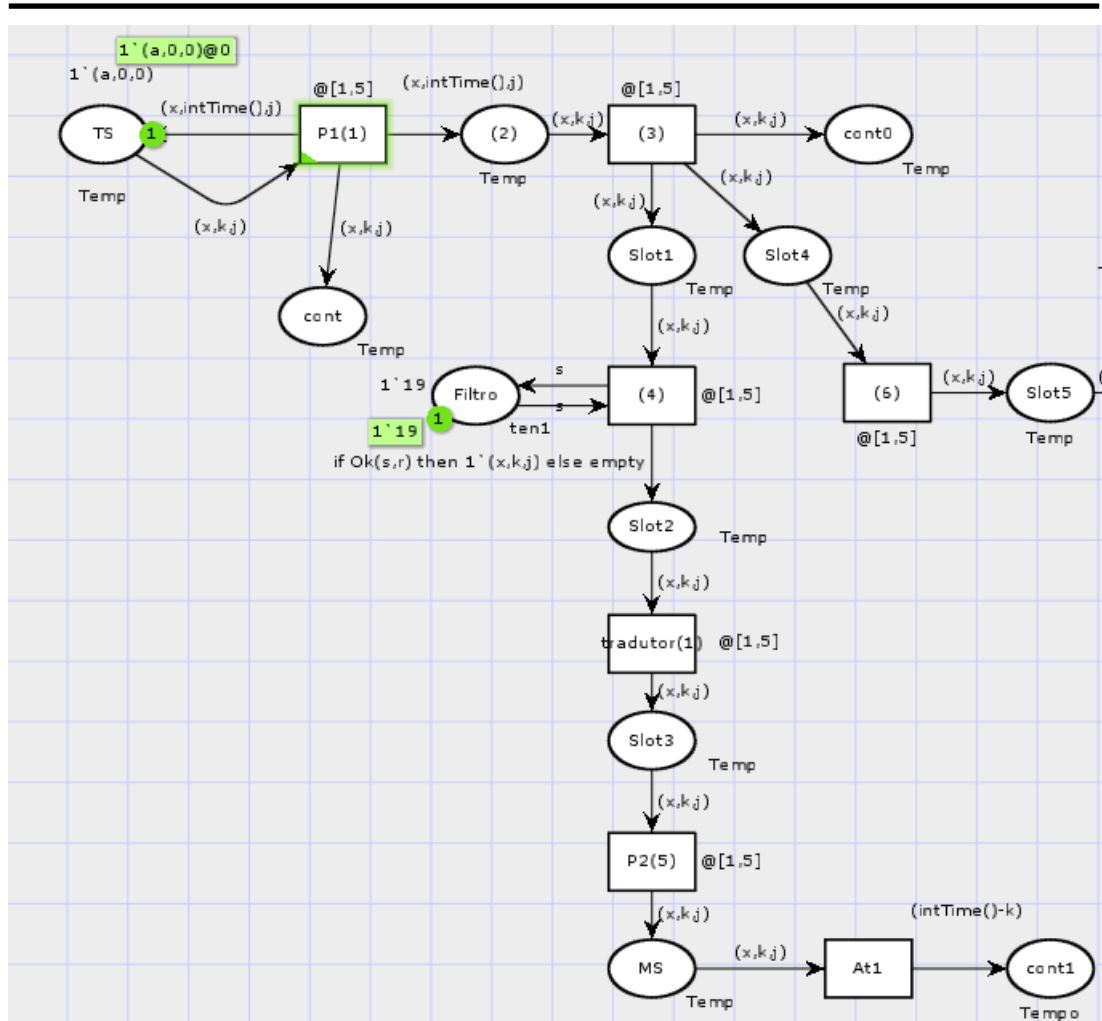


Figura 5.8: Caminho que um token percorre desde a entrada no lugar Sistema de Viagens (TS) até o lugar Email (MS) no CPN Tools.

lor possa ser subtraído do tempo acumulado durante o processo e, por essa diferença, se conheça o valor real que o *token* levou para percorrer todo caminho. Isto é necessário para que periodicamente sejam incluídos novos *tokens* no sistema e principalmente para que os demais possam ser enviados regularmente até aos lugares de chegada. A documentação encontrada em cpn [2] escreve que o tipo de tempo utilizado pelo CPN Tools é um inteiro infinito. Como os conjuntos infinitos de cores inteiras não são suportados, os

valores de tempo devem ser convertidos de valores inteiros infinitos para valores que possam ser usados no conjunto de cores, como sequências de caracteres ou inteiros. A função *intTime()* é declarada de tal modo que ela retorna o tempo atual do modelo como um número inteiro, conforme a sintaxe: *fun intTime () = IntInf.toInt (time ())*. A função *time()* retorna um inteiro infinito e a função *IntInf.toInt* é uma função da biblioteca Base SML que irá converter um valor inteiro infinito para um número inteiro.

O segundo detalhe se refere à transição (4) que representa a tarefa Filtro no Guaraná DSL. A tarefa no Guaraná DSL seleciona e habilita somente as mensagens com conteúdo correto e processa para o email do cliente. Para isso, a transição tem em sua sintaxe uma função *Ok(r,s)*. Esta função usa dois lugares auxiliares, o lugar *r* com uma quantidade de cores inteiras variando de 1 a 20, e o lugar *s* com uma quantidade de cores inteira variando de 1 a 19, fazendo com que o arco de variável *s* suporte 19 tipos de *tokens* dos 20 tipos suportados pelo lugar *r*. A função *Ok(r,s)*, de distribuição uniforme, faz com que aproximadamente 95% dos *tokens* sejam habilitados para seguir o processo até o lugar Email (MS), conforme a Figura §5.8. Esta porcentagem pode ser configurada conforme a necessidade do modelo.

O terceiro detalhe está nas tarefas (7), (8) e (9). A tarefa (7) é um separador de mensagens, enquanto que as tarefas (8) e (9) são correlacionadores. A configuração do modelo de simulação teve que se adaptar à sintaxe destas tarefas de maneira que satisfaça as condições de que uma mensagem processada no Guaraná DSL. Uma mensagem separada pela tarefa (7) segue para o Sistema de Fatura, Voos e Hotel, com informações diferenciadas em seu conteúdo, portanto as mensagens que seguem para o Hotel e para os Voos passam pelo correlacionador e somente seguirão o percurso quando habilitadas pelos correlacionadores (8) e (9). Para isso, a terceira coordenada, de variáveis *j* e *m*, são comparadas na guarda das transições (8) e (9) e somente quando os valores contidos nos *tokens* forem iguais é que serão habilitadas estas transições. A comparação é feita com valores relativos aos tempos anteriores dos *tokens*. Utilizando a função *intTime()* é possível extrair o tempo dos *tokens* e transformá-los em um número inteiro, podendo ser anexados às coordenadas *j* e *m* para realizar a comparação na guarda das transições (8) e (9). Acontece que o caminho de um *token* que passa pela transição (7) e segue para o *slot8*, passa pelo Replicador, e aguarda nos *slots* 11 e 12, é diferente do caminho do *token* que segue diretamente para os *slots* 7 e 9, aguardando a habilitação das transições (8) e (9). A transição Replicador atualiza os valores de tempo da transição (7), enquanto que os *tokens* que aguardam nos *slots* 7 e 9 não estão atualizados pela função *intTime()*. Por isso a necessidade de se colocar transições auxiliares, denominadas (8)A e (9)A, que não adicio-

nam valor de tempo aos *tokens* e que tem esse papel de atualizar os valores de tempo nos *tokens* armazenados nos *slots* 7 e 9.

O quarto detalhe aparece nas aplicações as quais chegam os *tokens* e que estão programadas com uma transição denominada atualizador e um lugar chamado contador. Esta transição serve para atualizar o tempo de chegada de um *token* qualquer e não adiciona tempo algum à este *token*. O lugar contador mostra quantos *tokens* chegam e qual valor de tempo foi associado a este *token*, permitindo conhecer de forma direta o tempo gasto para realizar o percurso desde o início até o lugar Email (MS). O arco de ligação (*intTime-k*) realiza a subtração do tempo atual do *token* com seu valor inicial. A Figura §5.8 mostra como exemplo o lugar Email (MS).

5.2.2 Modelo de Distribuição de Entrada

A inclusão de novos *tokens* no sistema se baseia na condição estabelecida pela solução de integração no Guaraná DSL. Esta condição se refere ao fato de que a aplicação *Travel System* (Sistema de Viagens) do estudo de caso, verifica periodicamente a chegada de novas mensagens. Para contemplar esta condição é necessário que o modelo de simulação tenha um Modelo de Distribuição de Entrada de *tokens*. Este modelo deve incluir periodicamente novos *tokens* no sistema gerando um fluxo bem definido. Para definir este fluxo no modelo de simulação transcrito no CPN Tools, são considerados dois aspectos que influenciam na velocidade de processamento dos *tokens* no sistema: um se refere ao fato de que no CPN Tools, uma rede de Petri Temporizada obedece uma regra de prioridade. Outro aspecto se refere a função $fun\ intTime () = IntInf.toInt (time ())$. A regra de prioridade no CPN Tools é determinística e compatível com a semântica de disparo das redes de Petri Temporizadas, conforme demonstra a equação (5.10). Esta regra serve para gerenciar os conflitos entre transições e define que se duas ou mais transições pertencentes a uma rede de Petri Temporizada estão habilitadas, então serão disparadas em sequência crescente em relação ao tempo, isto é, as transições de menor tempo são prioridade. Caso os tempos sejam iguais, então serão disparadas aleatoriamente.

$$\forall p_i \in P : M[t_j > , M(p_i) \geq I(p_i, t_j). \quad (5.10)$$

Isto é, para todo lugar p_i pertencente ao conjunto de lugares P , a marcação do lugar p_i deve ser maior ou igual ao peso do arco que liga o lugar p_i à transição t_j . Durante o intervalo $I(t_j)=(d_{j_{min}}, d_{j_{max}})$, a t_j só será disparável se: $t(t_j) \geq d_{j_{min}}$, isto é, se o tempo da transição t_j for maior ou igual ao tempo do seu delay mínimo. A função $fun\ intTime () = IntInf.toInt (time ())$ é usada no

arco de entrada e atualiza os valores de tempo dos próximos *tokens* a serem incluídos. Esta atualização juntamente com a semântica de disparo faz com que os *tokens* fluam de forma sistemática no modelo de simulação.

5.2.3 Qualidade de Serviço (QoS)

Em Tecnologia de Informação as redes são fundamentais na conexões entre os equipamentos e componentes de um sistema de computadores. A Qualidade de Serviço (QoS) determina a capacidade de troca de dados entre aplicações. Segundo [?] a Qualidade de Serviço (QoS) é um requisito das aplicações para a qual exige-se que determinados Parâmetros (atrasos, variação, perdas, ...) estejam dentro de limites bem definidos (valor mínimo, valor máximo). Aplicando esses conceitos de rede no modelo em redes de Petri Temporizadas, considerando que o sistema compreende um fluxo de dados (*tokens*), organizado por uma rede (rede de Petri Temporizada), é possível determinar o valor mínimo e máximo do tempo que uma mensagem leva para percorrer a solução de integração.

5.2.4 Caracterização dos Cenários

O serviço da Agência de Viagens é concluído quando uma mensagem chega à todas as aplicações envolvidas no ecossistema. A Qualidade de Serviço do modelo de simulação é obtida considerando o tempo que um *token* leva para percorrer todo o modelo de simulação em redes de Petri Temporizadas. Os *tokens* com tempo relativamente maior e menor formam um intervalo de valores que determinam as possibilidades de desempenho na Qualidade de Serviço que o modelo pode realizar. Os cenários de simulação do modelo em redes de Petri Temporizadas são determinados admitindo que cada transição t_i tem um valor inteiro de unidades de tempo que varia de 1 a 5, estes valores serão associados ao *token* de forma aleatória, isto é, seguindo a lógica da janela de tempo a transição habilitada sorteará um tempo inteiro entre 1 e 5 e somará este valor ao tempo do *token*, gerando o elemento $1'(a,k,j)@+delay$, no qual 1 é a quantidade do *token*, a letra *a* representa a cor ou classe, esta cor ou classe é única pois determina que todos os *tokens* sejam iguais e temporizados, isto é, os *tokens* são diferenciados somente pelo valor do tempo. A coordenada *k* é uma variável do tipo inteira e representa os valores de tempo que o *token* inicia o processo. A coordenada *j* é uma variável do tipo inteira que representa os valores de tempo que o *token* assume antes de chegar aos lugares (8) e (9). Estas transições modelam a tarefa do Correlacionador no Guaraná DSL e comparam *tokens* com mesmo valor de tempo.

O Delay é o valor de tempo inteiro, que pode ser constante ou uma função, associado a cada *token*. Quando a configuração de tempo usada está no formato de janela de tempo, cada transição do modelo tem um delay mínimo e máximo, isto é, os valores associados ao *token* podem ser qualquer um neste intervalo. Considerando uma janela de tempo [1, 5] numa transição, um *token* pode assumir equiprovavelmente qualquer valor entre o 1 e o 5, inclusive o 1 e o 5. Cada transição t_i irá responder ao sistema o tamanho de um *token* associando a este um tempo que varia aleatoriamente entre 1 e 5, assim ao chegar no final do percurso um *token* que percorreu n transições terá seu tempo entre $n \times 1$ e $n \times 5$, que são os valores de delay mínimo e máximo vezes as n transições percorridas. Nestas condições, se o tempo associado a um *token* for 1 ou 2: mensagem pequena (tamanho); tempo 3: mensagem de tamanho médio; se o tempo associado for 4 ou 5: mensagem grande (tamanho). Considerando estas informações foram realizadas simulações com três cenários: no primeiro cenário o modelo de simulação foi inicializado com um *token* do tipo $1(a,k,j)@+delay$, sendo o primeiro cenário com os seguintes parâmetros: $1(a,0,0)@0$, onde a entrada de novos *tokens* acontece em períodos variáveis, dependendo do sorteio de tempo nas transições. Foram aplicados 10.000 (dez mil) disparos de transições. No segundo cenário o modelo foi inicializado com os seguintes parâmetros: $1(a,0,0)@0$, onde a entrada de novos *tokens* acontece em períodos variáveis, dependendo do sorteio de tempo das transições. Foram aplicados 50.000 (cinquenta mil) disparos de transições. No terceiro cenário o modelo foi inicializado com os seguintes parâmetros: $1(a,0,0)@0$, onde a entrada de novos *tokens* acontece em períodos variáveis, dependendo do sorteio de tempo das transições. Foram realizados 100.000 (cem mil) disparos de transições. Os cenários foram determinados considerando o aumento gradativo do número de disparos. Em estatística, quando um experimento é repetido um grande número de vezes com os mesmos dados, seguindo a Lei dos Grande Números [5], conforme o número de repetições se incrementa, a média amostral das variáveis do experimento se aproxima, cada vez mais, da média populacional, também conhecida como média teórica ou esperança matemática. Empiricamente, para a análise de sistemas que ainda não existem, onde para obter os dados é executado um experimento artificial, a média populacional costuma ser obtida com, aproximadamente, 25 repetições. Desta forma, foram realizadas 25 repetições de cada um dos três cenários para obter resultados com precisão estatística do modelo de simulação formal proposto [40].

5.2.5 Função Probabilidade

A plataforma do CPN Tools é utilizada nesta pesquisa pois oferece muitas opções. Uma delas é a janela de tempo para simulação de redes de Petri Temporizadas. A janela de tempo de um *token* qualquer de cor a é descrito da seguinte maneira: $1'a@[t_{\min}, t_{\max}]$, sendo t_{\min} o tempo mínimo inteiro de disparo e t_{\max} o tempo máximo inteiro para disparo. A forma de sorteio de um valor inteiro entre o t_{\min} e o t_{\max} , é equiprovável devido a função P de probabilidade aplicada pelo CPN Tools ser uma função de distribuição uniforme discreta conforme a equação (5.11):

$$P(x) = \frac{x}{n(\Omega)}, \text{ tal que } n(\Omega) \neq 0. \quad (5.11)$$

onde $n(\Omega)$ é o número de elementos de um espaço amostral Ω tal que $n(\Omega) \neq 0$ e x é o número de elementos de um evento pertencente ao espaço amostral Ω . Aplicando ao modelo de simulação temos que para um evento A : sair qualquer número entre 1 e 5, inclusive os próprios 1 e 5, tem-se $x = 1$, sendo o conjunto $\Omega = \{1, 2, 3, 4, 5\}$ tal que $n(\Omega) = 5$, a probabilidade P de uma transição t_i sortear um valor de tempo entre 1 e 5 é:

$$P(1) = \frac{1}{5} \quad (5.12)$$

Considerando o fato de que na janela de tempo do CPN Tools o valor do tempo é escolhido aleatoriamente entre os valores 1 e 5, caso a escolha seja o caminho que leva ao lugar Email tem-se 5 transições, logo todas as possibilidades de tempo de um *token* podem ser calculadas. Na Figura §?? existem 5 transições no caminho que começa no Lugar Sistema de Viagens (TS) e termina no lugar Email (MS), então o tempo mínimo possível é o tempo mínimo 1 de cada transição vezes as 5 transições percorridas resultando 5, o mesmo cálculo pode ser aplicado para o tempo máximo, que acontece quando as 5 transições sorteiam o tempo 5, resultando em 25 unidades de tempo. Pode-se conhecer a probabilidade relativa dos tempos intermediários entre 5 e 25. Seja $\Omega = \{5,6,7,8,9,\dots,22,23,24,25\}$ o conjunto de todos os valores de tempo possíveis que um *token* qualquer pode assumir ao percorrer o caminho desde o início até o lugar Email (MS), contendo 5 transições, então para cada valor existe um conjunto formado por uma quantidade finita de sequências de disparos que resultam em um número pertencente ao conjunto Ω , por exemplo o conjunto de sequências de disparos para resultar o valor mínimo 5 indicado por $S5$ é dado por:

$$S5 = (1, 1, 1, 1, 1) = 1 + 1 + 1 + 1 + 1 = 5 \quad (5.13)$$

Também para o tempo máximo de 25 unidades de tempo, todas as 5 transições devem sortear o tempo máximo igual a 5, formando um conjunto de sequências S_{25} :

$$S_{25} = (5, 5, 5, 5, 5) = 5 + 5 + 5 + 5 + 5 = 25 \quad (5.14)$$

Nestes dois casos só existe uma sequência possível, mas entre os valores de mínimo e de máximo, há outros conjuntos S_n de sequências, que formam os restantes dos valores de tempo possíveis para um *token* que percorre as 5 transições. Seguindo esse padrão de cálculo pode-se formar todos os conjuntos contendo as sequências possíveis para o lugar Email (MS). O conjunto de sequências S_6 contém 1 sequência:

$$S_6 = \{(2, 1, 1, 1, 1)\} \quad (5.15)$$

O conjunto de sequências S_7 contém 2 sequências:

$$S_7 = \{(2, 2, 1, 1, 1)(3, 1, 1, 1, 1)\} \quad (5.16)$$

O conjunto de sequências S_8 contém 3 sequências:

$$S_8 = \{(2, 2, 2, 1, 1)(3, 2, 1, 1, 1)(4, 1, 1, 1, 1)\} \quad (5.17)$$

O conjunto de sequências S_9 contém 5 sequências:

$$S_9 = \{(2, 2, 2, 2, 1)(3, 2, 2, 1, 1)(4, 2, 1, 1, 1)(5, 1, 1, 1, 1)(3, 3, 1, 1, 1)\} \quad (5.18)$$

O conjunto de sequências S_{10} contém 6 sequências:

$$S_{10} = \{(2, 2, 2, 2, 2)(3, 2, 2, 2, 1)(4, 2, 2, 1, 1)(5, 2, 1, 1, 1)(3, 3, 2, 1, 1)(4, 3, 1, 1, 1)\} \quad (5.19)$$

O conjunto de sequências S_{11} contém 7 sequências:

$$S_{11} = \{(3, 2, 2, 2, 2)(4, 2, 2, 2, 1)(5, 2, 2, 1, 1)(3, 3, 3, 1, 1)(5, 3, 1, 1, 1)(3, 3, 2, 2, 1)(4, 4, 1, 1, 1)\} \quad (5.20)$$

O conjunto de sequências S_{12} contém 9 sequências:

$$S_{12} = \{(4, 2, 2, 2, 2)(5, 2, 2, 2, 1)(5, 3, 2, 1, 1)(4, 3, 3, 1, 1)(5, 4, 1, 1, 1)(4, 3, 2, 2, 1)(5, 4, 1, 1, 1)(3, 3, 3, 1, 2)(3, 3, 2, 2, 2)\} \quad (5.21)$$

O conjunto de sequências S13 contém 11 sequências:

$$S13 = \{(5, 2, 2, 2, 2)(5, 3, 2, 2, 1)(5, 4, 2, 1, 1)(4, 3, 3, 2, 1)(5, 4, 2, 1, 1) \\ (4, 3, 2, 2, 2)(5, 5, 1, 1, 1)(3, 3, 3, 2, 2)(5, 3, 3, 1, 1)(3, 3, 3, 3, 1) \\ (5, 2, 2, 2, 2)\} \quad (5.22)$$

O conjunto de sequências S14 contém 11 sequências:

$$S14 = \{(5, 3, 2, 2, 2)(5, 3, 3, 2, 1)(5, 4, 2, 2, 1)(4, 3, 3, 2, 2)(5, 5, 2, 1, 1) \\ (4, 4, 2, 2, 2)(5, 4, 3, 1, 1)(3, 3, 3, 3, 2)(4, 4, 4, 1, 1)(4, 3, 3, 3, 1) \\ (4, 4, 3, 2, 1)\} \quad (5.23)$$

O conjunto de sequências S15 contém 12 sequências:

$$S15 = \{(5, 4, 2, 2, 2)(5, 4, 3, 2, 1)(5, 5, 2, 2, 1)(5, 3, 3, 2, 2)(5, 5, 3, 1, 1) \\ (4, 4, 4, 2, 1)(5, 4, 4, 1, 1)(3, 3, 3, 3, 3)(4, 4, 3, 2, 2)(4, 4, 3, 3, 1) \\ (4, 3, 3, 3, 2)(5, 3, 3, 3, 1)\} \quad (5.24)$$

O conjunto de sequências S16 contém 11 sequências:

$$S16 = \{(5, 5, 2, 2, 2)(5, 5, 3, 2, 1)(5, 4, 3, 2, 2)(5, 3, 3, 3, 2)(5, 5, 4, 1, 1) \\ (4, 4, 4, 2, 2)(5, 4, 4, 2, 1)(4, 3, 3, 3, 3)(4, 4, 3, 3, 2)(5, 4, 3, 3, 1) \\ (4, 4, 4, 3, 1)\} \quad (5.25)$$

O conjunto de sequências S17 contém 10 sequências:

$$S17 = \{(5, 5, 3, 2, 2)(5, 5, 3, 3, 1)(5, 4, 4, 2, 2)(5, 3, 3, 3, 3)(5, 5, 5, 1, 1) \\ (4, 4, 4, 3, 2)(4, 4, 4, 4, 1)(4, 4, 3, 3, 3)(5, 4, 3, 3, 2)(5, 4, 4, 3, 1)\} \quad (5.26)$$

O conjunto de sequências S18 contém 8 sequências:

$$S18 = \{(5, 5, 4, 2, 2)(5, 5, 3, 3, 2)(5, 4, 4, 3, 2)(5, 4, 3, 3, 3)(5, 5, 5, 2, 1) \\ (4, 4, 4, 3, 3)(4, 4, 4, 4, 2)(5, 5, 4, 3, 1)\} \quad (5.27)$$

O conjunto de sequências S19 contém 8 sequências:

$$S19 = \{(5, 5, 5, 2, 2)(5, 5, 5, 3, 1)(5, 5, 4, 3, 2)(5, 4, 4, 4, 2)(5, 5, 3, 3, 3) \\ (5, 5, 4, 4, 1)(5, 4, 4, 3, 3)(4, 4, 4, 4, 3)\} \quad (5.28)$$

O conjunto de sequências S20 contém 6 sequências:

$$S20 = \{(5, 5, 5, 3, 2)(5, 5, 5, 4, 1)(5, 5, 4, 3, 3)(5, 5, 4, 4, 2)(5, 4, 4, 4, 3) \\ (4, 4, 4, 4, 4)\} \quad (5.29)$$

O conjunto de sequências S21 contém 5 sequências:

$$S21 = \{(5, 5, 5, 5, 1)(5, 5, 5, 4, 2)(5, 5, 5, 3, 3)(5, 5, 4, 4, 3)(5, 4, 4, 4, 4)\} \quad (5.30)$$

O conjunto de sequências S22 contém 3 sequências:

$$S22 = \{(5, 5, 5, 5, 2)(5, 5, 4, 4, 4)(5, 5, 5, 4, 3)\} \quad (5.31)$$

O conjunto de sequências S23 contém 2 sequências:

$$S23 = \{(5, 5, 5, 5, 3)(5, 5, 5, 4, 4)\} \quad (5.32)$$

O conjunto de sequências S24 contém 1 sequência:

$$S24 = \{(5, 5, 5, 5, 4)\} \quad (5.33)$$

Considerando esses conjuntos e a quantidade de sequências contidas em cada conjunto demonstra-se no Gráfico §5.9 que para o lugar Email há maior probabilidade de se obter *tokens* com tempos de valor igual a 15. O Gráfico §5.9 é formado por um eixo horizontal que representa os valores de tempo possíveis para o caminho Email (MS), enquanto que o eixo vertical, em cada topo há o número de sequências de cada conjunto S_n .

5.3 Resultados Experimentais

Após realizar as simulações seguindo os critérios dos cenários produziu-se resultados em relação ao padrão de valores deduzidos nesta pesquisa. Primeiramente é mostrado os valores totais de *tokens* armazenados no lugar Email (MS), pelo Gráfico §5.10, formado por um eixo horizontal que representa os valores P possíveis para o lugar Email (MS) e por um eixo vertical, que mostra valores no topo de cada coluna de tempo representando a soma total de *tokens* disparados, após as simulações nos três cenários. O Gráfico §5.10 demonstra que o maior número de *tokens* está concentrado na coluna que tem valor de tempo igual a 15. Este resultado era o mais provável

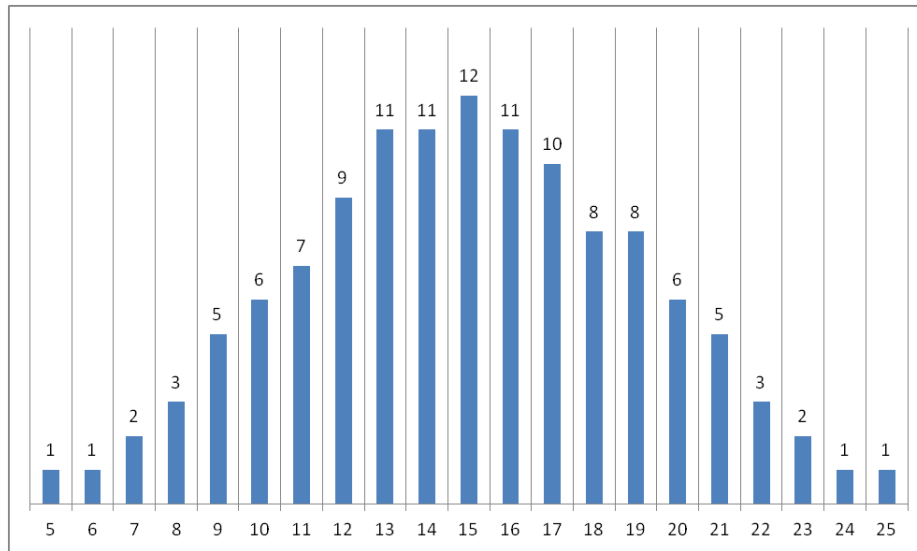


Figura 5.9: Gráfico de quantidade de seqüências por conjunto de S_n , sendo $n = \{5, 6, \dots, 24, 25\}$.

de acontecer, apesar de que os demais tempos tem uma probabilidade aproximada mas, as simulações mostraram que o padrão descrito por cada conjunto S_n de seqüências é aplicado com rigor pelo programa CPN Tools, conforme mostra o Gráfico de quantidade de seqüências §5.9, onde o valor 15 é mais provável devido ao fato de se obter 12 seqüências, o maior valor entre os demais conjuntos. Esta lógica se confirma também nos demais lugares. O Gráfico §5.11 é formado por um eixo horizontal contendo todos os valores P possíveis para este lugar e por um eixo vertical, contendo no topo de cada coluna, a soma do total de *tokens* disparados até o lugar Fatura (IS). O Gráfico §5.12 é formado por um eixo horizontal contendo todos os valores P possíveis para este lugar e por um eixo vertical, contendo no topo de cada coluna, a soma do total de *tokens* disparados até o lugar Voos (V). O Gráfico §5.13 é formado por um eixo horizontal contendo todos os valores P possíveis para este lugar e por um eixo vertical, contendo no topo de cada coluna, a soma do total de *tokens* disparados até o lugar Hotel (H). Os valores totais se referem a soma de cada coluna T oriundas das Tabelas dos resultados de simulação.

O Gráfico §5.14 é formado por um eixo horizontal que se refere à todos os possíveis valores de tempo dos quatro lugares, Email (MS), Fatura (IS), Voos (V) e Hotel (H). Estes valores vão desde o valor mínimo possível no lugar Email (MS), o valor 5, até o valor máximo possível nos lugares Voos (V) e

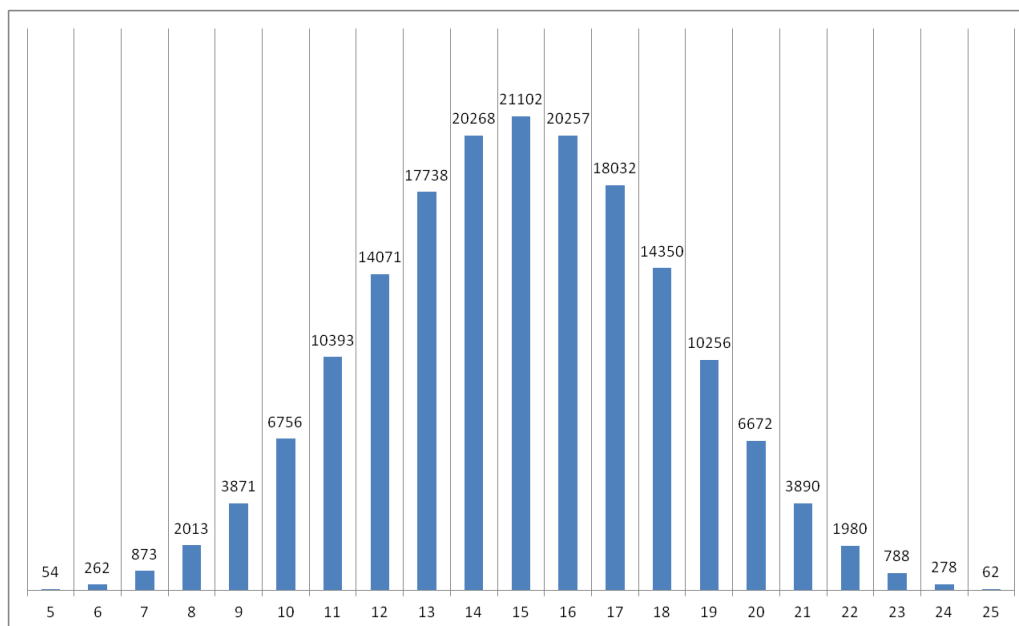


Figura 5.10: Gráfico do Total de tokens armazenados no lugar Email após as simulações com 10.000, 50.000 e 100.000 mil disparos em relação aos valores de tempos possíveis para esse lugar.

Hotel (H), o valor 40. O eixo Vertical se refere a soma do total de *tokens* disparados em todos os lugares, para cada valor de tempo possível, isto é, se considerarmos o valor de tempo 20, o valor mostrado no topo da coluna, no Gráfico §5.14 deste valor, é o resultado da soma dos *tokens* que obtiveram tempo 20 em todos os lugares. Estes valores são resultados da soma dos valores de cada coluna resultante nos Gráficos §5.10, §5.11, §5.12 e §5.13.

Como mencionado, este trabalho considera a Qualidade de Serviço (QoS) como sendo o tempo que um *token* leva para perorrer todo o modelo de simulação em redes de Petri Temporizadas. Os *tokens* com tempo relativamente maior e menor formam um intervalo de valores que determinam as possibilidades de desempenho na Qualidade de Serviço (QoS) que o modelo pode realizar. O Gráfico §5.14 é resultado da soma total de *tokens* processados no sistema. Pode-se observar que o valor de tempo que mais se repetiu durante todo processo de simulação é o valor 21, logo este valor de tempo é o mais reproduzido pelo sistema nas condições estabelecidas pelos cenários e também pela configuração do modelo de simulação. Respondendo a questão de Qualidade de Serviço (QoS) pode-se dizer que segundo os valores do Grá-

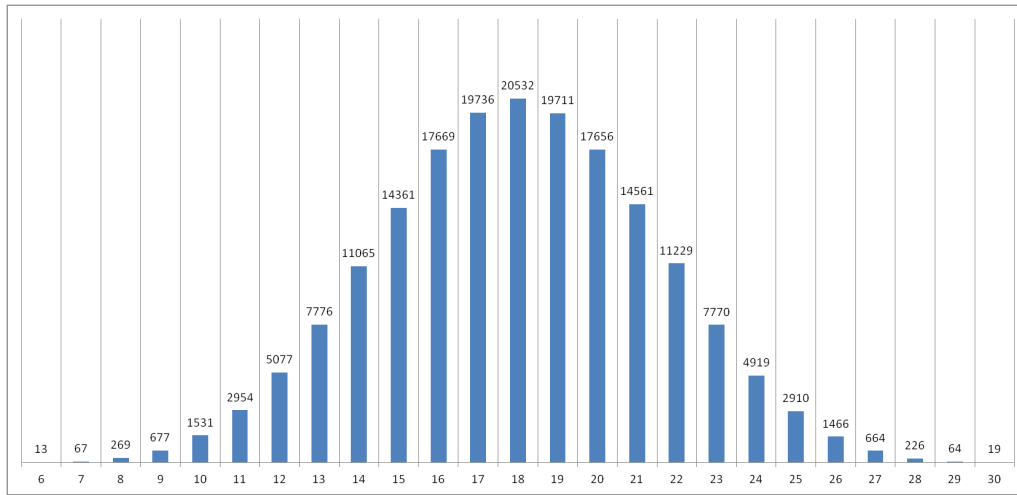


Figura 5.11: Gráfico do Total de tokens armazenados no lugar Fatura após as simulações com 10.000, 50.000 e 10.0000 mil disparos em relação aos valores de tempos possíveis para esse lugar.

fico §5.14 há uma porcentagem relativa a cada valor de tempo em relação à soma total de *tokens* observados nos lugares Email (MS), Fatura (IS), Voos (V) e Hotel (H). O total foram 723.903 mil *tokens* disparados. Exatamente 47.599 mil *tokens* tem tempo 21, o que representa 6,57% da quantia total, se ampliarmos para um intervalo maior, isto é, se considera-se os valores de tempo 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, com mais *tokens* observados e que correspondem a 31% dos 35 valores de tempo possíveis, obtem-se o número de 482.162 mil *tokens* ou 66,6% do total de *tokens*. Logo é possível afirmar que de cada 10 mensagens, 6 terão tempo dentro deste intervalo. Se, quando o tempo associado a um *token* for 1 ou 2: mensagem pequena (tamanho); tempo 3: mensagem de tamanho médio; se o tempo associado for 4 ou 5: mensagem grande (tamanho), então esses valores de tempo no intervalo $I = [15, 25]$, correspondem a uma faixa de mensagens de tamanho relativamente médio. Se usarmos o valor 3, valor de tempo médio associado por uma transição e considerar o caminho com menos transições, caminho que leva ao lugar Email (MS) com 5 transições e os caminhos com mais transições, os caminhos que chegam no lugares Voos (V) e Hotel (H) com 8 transições, multiplicarmos esses valores por 3, tem-se os valores 15 e 24. Estes dois valores, 15 e 24 juntamente com o valor 25, formam o intervalo de tempo contendo os números mais observados no Gráfico §5.14. Estes valores médios de tempo

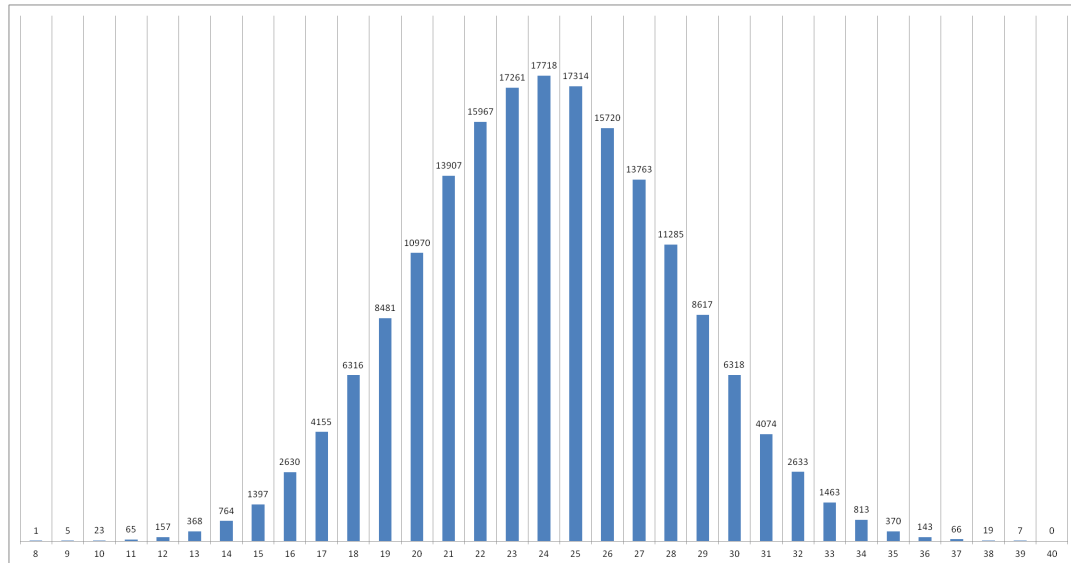


Figura 5.12: Gráfico do Total de tokens armazenados no lugar Voos após as simulações com 10.000, 50.000 e 100.000 mil disparos em relação aos valores de tempos possíveis para esse lugar.

caracterizam o tamanho de uma mensagem e são imagem da função de probabilidade. Caso queira se avaliar o sistema com probabilidades diferentes e que tendem à faixas de tempo maiores ou menores, deve se configurar esta opção. O modelo em redes de Petri Temporizadas transcrito na ferramenta CPN tools é altamente adaptável a estas situações. Analisando os resultados das Tabelas de simulação §5.15, §5.16 e §5.17 percebe-se que a porcentagem de *tokens* que chegam até o lugar Email (MS) se aproxima de 95%.

As Tabelas §5.15, §5.16 e §5.17 contém os resultados das simulações no lugar Email (MS), a primeira linha é formada pelas referências P , $R1$, $R2, \dots, R25$, T , onde P se refere a todas as possibilidades de tempo para o lugar Email (MS), variando desde o valor mínimo 5 até o valor máximo possível 25, e as colunas do Rn , com n variando de 1 a 25, se refere a cada uma das 25 repetições. Cada repetição Rn forma uma coluna de números que representa a quantidade de *tokens* observados. Os valores da coluna T se referem a soma total de *tokens*, nas 25 repetições realizadas, para cada tempo, em cada linha. A linha $T1$ se refere ao total de *tokens* processados até o lugar Email (MS), ou seja, é a soma total da cada coluna $R1$, $R2, \dots, R25$. A linha $T2$ se refere a quantidade de *tokens* armazenados no lugar *cont0* antes do Filtro. A linha TE se

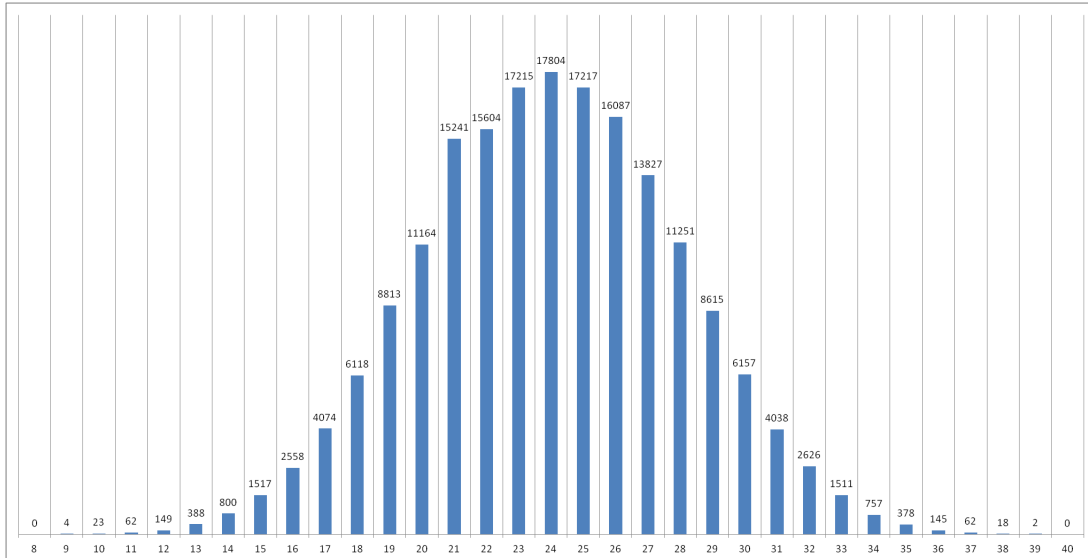


Figura 5.13: Gráfico do Total de tokens armazenados no lugar Hotel após as simulações com 10.000, 50.000 e 100.000 mil disparos em relação aos valores de tempos possíveis para esse lugar.

refere a quantidade total de entrada de *tokens* no sistema, contabilizados no lugar *cont*. A linha % se refere ao percentual aproximado de *tokens*, em cada repetição, em relação a quantidade de *tokens* que estão no lugar *cont0*, antes do Filtro, com a quantidade de *tokens* que chegam até o lugar *cont1*, isto é, a quantidade de *tokens* na linha *T1* dividida pela quantidade de *tokens* da linha *T2* vezes 100. As Tabelas §2, §3, §4, §5, §6, §7, §8, §9 e §10 contêm os valores de simulação nos lugares Fatura (IS), Voos (V) e Hotel (H). Essas Tabelas mantêm os mesmos padrões de referência das Tabelas que registram os resultados das simulações no lugar Email (MS), o único detalhe é que não possuem as linhas *T2* e % pois, essas referências são exclusivas para a análise de performance da transição (4), o Filtro, na contagem percentual dos *tokens* que são selecionados para seguirem até o lugar Email (MS).

5.4 Conjectura sobre os Resultados

Considerando o gráfico de quantidade de sequências do lugar Email (MS), com valores de tempo entre 5 e 25, constatou-se que o valor de maior probabilidade e com maior número de sequências possíveis é o valor 15. Seja o

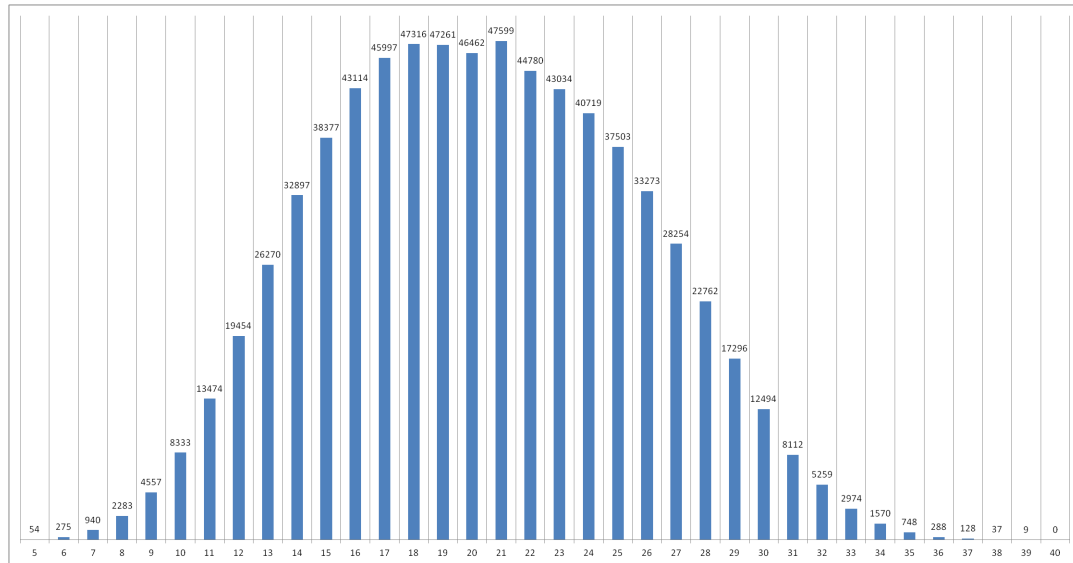


Figura 5.14: Gráfico geral do Total de tokens armazenados nos lugares Email, Fatura, Voos e Hotel, após as simulações com 10.000, 50.000 e 100.000 mil disparos em relação aos valores de tempos possíveis para o modelo de simulação.

Gráfico §5.11 da Fatura (IS), com valores de tempo entre 6 e 30, o valor que mais ocorreu foi o 18. Seja o Gráfico §5.12 dos Voos e o Gráfico §5.13 do Hotel, com valores entre 8 e 40, o valor que mais ocorreu foi o 24. Analisando estes valores, podemos conjecturar que:

Dado um caminho qualquer pertencente a uma rede de Petri Temporizada, com transições temporizadas, associando valores de tempo entre 1 e 5, inclusive o 1 e o 5 nos *tokens*, com valores aleatórios obedecendo a função de probabilidade P , o valor de maior probabilidade e que forma mais sequências é o elemento que ocupa a posição que corresponde a 60% do valor máximo.

Pois em todos os lugares o elemento que obteve maior quantidade de *tokens* é o valor que representa 60% do valor máximo possível para o caminho. No lugar Email(MS), o elemento 15 corresponde 60% de 25 que é o máximo do valor, no lugar Fatura (IS), o elemento 18 corresponde a 60% de 30, nos lugares Voos (V) e Hotel (H), o elemento 24 corresponde a 60% de 40. Então podemos conjecturar que um caminho qualquer em que o valor máximo

P	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	T
5	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	2	0	0	0	0	5
6	0	0	0	0	1	2	1	2	0	0	0	0	0	0	3	2	0	0	0	1	1	1	2	0	1	17
7	1	3	0	0	2	2	2	7	2	0	0	0	1	3	2	0	2	3	0	3	1	1	2	4	1	42
8	10	6	8	4	6	7	8	6	5	4	8	8	6	3	5	4	6	12	2	4	6	4	2	4	5	143
9	5	9	10	11	12	15	11	15	10	7	10	10	15	8	8	9	11	10	9	13	10	9	14	9	8	258
10	23	23	10	15	14	21	19	16	11	22	10	10	19	23	9	18	27	15	19	19	12	27	14	13	15	424
11	27	22	29	21	24	27	15	20	24	30	29	29	26	28	23	24	25	19	22	26	26	24	29	15	26	610
12	32	35	38	24	25	32	36	32	36	35	38	38	37	36	39	37	33	29	33	40	45	29	33	40	44	876
13	45	58	41	52	43	45	34	39	48	43	41	41	39	46	48	43	42	45	43	44	46	37	39	42	32	1076
14	51	37	45	52	55	58	47	56	48	56	45	45	63	52	58	58	57	52	52	50	51	44	62	59	51	1304
15	47	55	54	57	45	55	68	50	48	43	54	54	43	47	54	60	55	61	57	56	56	56	45	51	58	1329
16	61	59	46	56	52	43	51	46	61	58	46	46	55	53	51	52	40	55	49	58	55	49	55	67	45	1309
17	48	44	55	36	44	34	46	44	46	42	55	55	44	44	45	33	51	42	37	42	35	61	42	52	52	1129
18	29	38	35	50	33	38	40	41	33	44	35	35	33	33	26	27	28	29	33	33	34	34	27	33	37	858
19	26	18	28	21	34	28	25	21	25	20	28	28	24	27	29	32	26	21	33	21	24	23	36	19	24	641
20	20	19	16	15	22	8	20	13	17	15	16	16	11	15	20	18	16	22	21	10	12	19	12	14	17	404
21	6	9	10	7	7	12	10	13	13	5	10	10	13	10	8	6	11	10	16	8	13	9	12	9	11	248
22	2	9	3	7	8	3	5	5	8	9	3	3	9	3	3	7	7	8	3	0	4	2	4	2	5	122
23	5	1	4	2	1	1	3	1	1	4	4	4	0	4	0	1	2	2	1	5	0	6	2	0	6	60
24	0	1	1	0	1	0	2	1	0	0	1	1	1	2	0	0	0	0	0	0	0	0	2	1	0	14
25	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	1	1	0	0	0	0	0	0	5
T1	438	436	433	430	429	431	443	430	436	437	433	433	440	437	433	431	439	436	431	434	433	435	434	434	438	10864
T2	460	462	461	461	462	461	459	462	462	461	461	461	461	461	463	463	461	462	462	461	462	461	460	461	462	11533
TE	461	463	462	462	463	462	460	463	463	462	462	462	462	462	464	464	463	463	463	462	463	462	461	462	463	11559
%	95	94	93	93	94	93	96	93	94	94	93	93	95	94	93	93	93	93	93	93	94	93	94	94	94	94

Figura 5.15: Resultado da simulação demonstrando a quantidade de tokens armazenados no lugar Email (MS) após 10.000 disparos e 25 repetições.

seja múltiplo de 5, por exemplo o 55, o elemento 33, que corresponde 60% de 55, é o valor que mais terá sequências que resultam no valor 33 e consequentemente esse valor terá maior probabilidade de acontecer em detrimento dos outros valores de tempo. Outro exemplo, o valor 70, o elemento 42 corresponde à 60% de 70, logo é o valor que terá o maior número de conjuntos de sequências que resultam 42. Portanto, é mais provável que num modelo de simulação, que exista um caminho com 14 transições com janela de tempo [1, 5], em que o valor máximo do caminho seja 70, o valor de tempo 42 seja o mais observado.

5.5 Resumo do Capítulo

Nesse capítulo, foi introduzida a proposta do modelo de simulação em redes de Petri Temporizadas, fez-se a transcrição do modelo de simulação para a plataforma CPN Tools, foram realizadas as simulações e as análises sobre resultados experimentais.

P	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	T
5	1	0	1	0	2	1	0	0	0	2	0	2	5	0	0	1	2	3	0	1	1	0	0	2	24	
6	0	3	4	4	0	2	5	2	3	5	4	0	2	5	2	1	1	6	8	2	2	2	5	3	5	76
7	8	12	7	9	17	13	14	14	11	11	9	17	13	14	14	12	10	19	7	18	8	9	5	12	9	292
8	25	30	25	31	18	26	18	20	20	24	31	18	26	18	20	28	30	29	29	22	14	31	25	24	27	609
9	61	48	42	54	41	48	41	56	44	53	54	41	48	41	56	50	54	46	45	53	52	48	41	48	57	1222
10	100	73	88	63	83	80	76	88	82	71	63	83	80	76	88	90	84	105	115	87	82	81	100	95	103	2136
11	120	136	150	129	120	135	149	136	140	142	129	120	135	149	136	151	121	141	118	130	124	129	142	107	127	3316
12	167	173	171	164	157	179	201	180	155	194	164	157	179	201	180	188	155	177	178	171	164	205	158	157	183	4358
13	257	209	222	218	245	226	203	238	229	218	218	245	226	203	238	217	223	212	210	198	222	203	241	216	223	5560
14	218	231	249	248	253	227	265	246	275	265	248	253	227	265	246	263	253	226	236	274	266	244	270	257	244	6249
15	257	278	265	270	275	270	266	255	237	267	270	275	270	266	255	248	268	258	247	271	264	246	242	248	268	6536
16	248	271	228	252	263	254	264	240	243	267	252	263	254	264	240	239	242	256	263	244	259	249	268	265	240	6328
17	218	223	227	238	234	236	206	250	201	204	238	234	236	206	250	202	255	241	195	247	238	201	209	228	225	5642
18	196	178	190	186	176	186	161	152	226	157	186	176	186	161	152	191	167	187	193	176	190	203	175	195	174	4520
19	123	130	135	126	108	149	130	125	130	128	126	108	149	130	125	137	110	123	139	128	145	174	131	118	135	3262
20	81	91	92	78	75	80	93	83	95	77	78	75	80	93	83	91	108	68	80	70	70	97	74	97	80	2089
21	57	50	36	60	60	39	38	50	52	44	60	60	39	38	50	43	54	48	51	48	51	47	60	55	38	1228
22	14	30	28	21	27	19	28	24	26	30	21	27	19	28	24	23	30	29	29	23	21	30	21	27	22	621
23	11	11	10	10	9	6	10	11	14	8	10	9	6	10	11	7	4	6	7	12	13	9	9	10	7	230
24	5	4	1	4	5	3	3	7	4	5	4	5	3	3	7	4	1	1	1	4	0	1	2	5	4	86
25	1	0	0	0	0	1	0	1	2	0	0	1	0	1	1	1	0	1	0	0	0	0	0	1	0	10
T1	2168	2181	2171	2165	2168	2180	2171	2178	2189	2172	2165	2168	2180	2171	2178	2186	2171	2181	2154	2178	2186	2183	2178	2168	2174	54364
T2	2292	2290	2291	2295	2292	2290	2292	2291	2289	2291	2295	2292	2290	2292	2291	2291	2290	2296	2291	2290	2290	2291	2292	2293	2293	57289
TE	2294	2291	2293	2296	2293	2291	2293	2292	2290	2292	2296	2293	2291	2293	2292	2292	2292	2291	2297	2292	2291	2291	2292	2294	2294	57316
%	94	95	94	94	94	95	94	95	95	94	94	94	95	94	95	95	94	95	93	95	95	95	95	94	94	94

Figura 5.16: Resultado da simulação demonstrando a quantidade de tokens armazenados no lugar Email (MS) após 50.000 disparos e 25 repetições.

P	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	T
5	5	0	0	0	0	3	0	0	2	0	0	3	2	0	1	1	0	1	1	0	1	1	1	1	1	25
6	10	7	6	5	16	4	10	6	7	8	7	4	7	3	7	5	3	7	6	6	9	8	5	8	5	169
7	24	20	21	9	20	22	25	19	26	20	24	18	33	25	21	25	21	19	14	22	19	21	24	26	21	539
8	54	53	56	41	47	60	55	49	45	46	56	70	54	46	50	46	42	48	52	45	42	52	55	49	48	1261
9	83	121	80	107	79	107	92	98	103	85	95	85	117	111	80	99	85	101	96	101	106	99	66	112	83	2391
10	158	173	181	175	167	149	175	153	180	179	134	155	183	159	176	159	175	162	187	154	170	176	180	177	159	4196
11	234	282	284	273	269	276	258	239	258	267	264	236	238	237	289	276	245	236	269	245	275	247	262	243	265	6467
12	378	325	364	354	343	340	336	356	400	356	364	348	340	362	341	367	350	356	338	370	359	353	332	354	351	8837
13	449	466	449	441	440	459	404	405	391	440	455	445	436	463	478	462	468	446	470	421	456	406	463	425	464	11102
14	532	483	497	500	548	501	521	507	530	495	470	494	504	528	509	509	535	511	500	482	518	502	511	531	497	12715
15	499	518	527	523	530	545	552	541	513	516	569	535	522	546	530	504	499	514	528	516	546	555	533	507	569	13237
16	515	509	509	499	499	499	507	542	519	505	464	548	533	472	469	520	461	513	490	530	504	514	487	522	490	12620
17	504	420	441	468	457	459	397	448	459	469	466	456	437	420	419	443	454	495	441	460	419	466	478	428	457	11261
18	363	369	340	347	358	348	379	391	335	330	366	339	339	355	333	358	402	344	347	387	343	391	374	368	366	8972
19	236	248	243	238	247	250	284	240	270	250	238	232	280	247	270	241	243	280	260	260	264	266	264	254	248	6353
20	141	188	177	179	161	158	176	173	163	179	173	158	173	186	171	168	183	153	182	175	153	137	132	178	162	4179
21	88	90	103	103	104	99	98	83	82	113	103	98	90	97	109	93	110	84	81	107	96	86	105	98	94	2414
22	43	59	59	56	35	46	65	53	47	49	51	48	35	45	62	39	49	52	56	32	43	50	58	59	46	1237
23	17	25	20	22	22	14	14	19	25	18	17	24	17	27	17	22	17	16	19	27	21	23	19	20	16	498
24	10	8	9	9	6	3	5	7	8	7	11	7	5	7	7	8	5	5	7	11	11	6	7	2	7	178
25	2	1	1	3	1	2	1	3	2	1	1	5	3	3	1	2	2	5	2	0	0	2	1	1	1	47
T1	4345	4365	4367	4352	4349	4344	4355	4330	4366	4334	4328	4303	4350	4339	4342	4346	4349	4345	4346	4353	4355	4360	4358	4363	4350	108694
T2	4581	4578	4578	4579	4580	4580	4578	4583	4577	4583	4583	4587	4581	4583	4580	4582	4580	4581	4583	4579	4578	4577	4579	4578	4579	114507
TE	4582	4579	4579	4580	4581	4581	4579	4584	4578	4584	4584	4588	4582	4582	4581	4583	4581	4582	4584	4580	4579	4579	4579	4579	4580	114530
%	94	95	95	95	94	94	95	94	95	94	94	93	94	94	94	94	94	94	94	95	95	95	95	95	95	94

Figura 5.17: Resultado da simulação demonstrando a quantidade de tokens armazenados no lugar Email (MS) após 100.000 disparos e 25 repetições.

Capítulo 6

Verificação do Modelo de Simulação

Knowing is not enough; we must apply.

Willing is not enough; we must do.

Johann W. von Goethe, German Novelist (1749-1832)

Nesse capítulo é realizada a verificação do modelo de simulação proposto. Não é realizada a Validação pois por definição a validação somente é possível quando se utiliza valores do sistema real. Neste trabalho o objetivo era exatamente antecipar os possíveis cenários do sistema e simular o modelo conceitual a partir de valores sintéticos. Neste caso é possível realizar somente a verificação do modelo de simulação. Para isto se fez necessário realizar um estudo sobre as técnicas existentes na literatura e determinar quais são possíveis de aplicar nesta pesquisa. Na Seção §6.1 é apresentada as definições e as técnicas de Verificação porpostas por Kleijen [28] e Sargent [38]. Na Seção §6.2 é realizada a Verificação do modelo de simulação.

6.1 Técnicas de Verificação

Os modelos de simulação são utilizados para estudar o comportamento de sistemas reais e servem para prever a performance do sistema, em diferentes cenários de funcionamento. Na elaboração de um modelo de simulação, é necessário estar seguro de que o mesmo represente corretamente o sistema real. Isso significa estar livre de erros de sintaxe e/ou lógica e ter compatibilidade suficiente para ser utilizado como substituto do sistema, para a análise e

experimentação. Essas duas etapas são conhecidas como verificação e validação de um modelo. As técnicas de verificação tratam do desenvolvimento de programas de computadores e de procedimentos aplicáveis especialmente para modelos de simulação, ou seja, baseiam-se em métodos de correção de programas Wiesner [40]. A validação ocorre quando o desenvolvedor do modelo e pessoas conhecedoras do sistema real trabalham em conjunto para analisar e avaliar como o modelo funciona. Refere-se às técnicas utilizadas para assegurar que o modelo, apesar dos pressupostos e simplificações, representa o sistema real a um nível suficiente de precisão Wiesner [40]. Nas fases de verificação e validação, muitas vezes são encontrados erros que exigem modificações significativas no modelo. Após a correção, o modelo deve ser novamente verificado e validado. De acordo com et al. [16], deve-se notar que nenhum modelo é 100% verificado ou validado. Um modelo é uma representação do sistema, e o comportamento do modelo é, na melhor das hipóteses, uma aproximação do comportamento do sistema real.

A literatura apresenta diversas técnicas de verificação e validação de modelos de simulação quando há dados do sistema real. Trabalhos que relatam técnicas para verificação e validação de modelos sem dados reais são escassos. A modelagem proposta nesta dissertação não apresenta dados do sistema real. As técnicas aplicadas para a verificação deste modelo foram propostas por Kleijen [28] e por Sargent [38].

A técnica proposta por Kleijen [28] aborda como verificar e validar modelos de simulação através da aplicação de técnicas estatísticas, na qual o tipo de técnica aplicada depende da disponibilidade de dados sobre o sistema real. Para a verificação do modelo precisa-se de conhecimento especializado, sendo assim, a verificação se dará por meio do conhecimento de especialistas sobre o comportamento do sistema.

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

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

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

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

Existem muitas propostas de técnicas de verificação segundo Sargent [38] mas, duas dessas técnicas 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 diretamente no modelo computacional de simulação. A verificação tem que garantir que o modelo de simulação foi implementado de forma correta na ferramenta. Deste modo, a técnica propõe a comparação do número de elementos que entrou no sistema com os que saíram. Partindo da ideia de que a entrada precisa ser igual aos elementos que ficaram acumulados somados aos que chegaram ao final do processo.

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

6.2 Verificação do Modelo de Simulação

Seguindo a técnica de Kleijen 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. No entanto, no caso do modelo de simulação desta pesquisa não se observou em momento algum, considerando todos os três cenários com 10.000 mil, 50.000 mil

e 100.000 mil *tokens*, acúmulo superior a 30 *tokens* levando em conta todos os lugares, exceto é claro, nos lugares de chegada, os lugares Email (MS), Fatura (IS), Voos (V) e Hotel (H) em que é esperado que aproximadamente todos os *tokens* incluídos no sistema, sejam armazenados. Isto se deve a sintaxe de inclusão de *tokens* usada no arco de ligação com a transição P1(1), que atualiza periodicamente os tempos dos *tokens*, para que estes tempos possam ser equilibrados em relação aos demais. Logo, o pouco acúmulo nos lugares é um comportamento esperado. Portanto, não há gargalos de desempenho do modelo de simulação considerando estes cenários.

Uma solução de integração é um sistema que inclui novas mensagens periodicamente, fazendo com que os dados transitem e sejam acessados por todas as aplicações de modo assíncrono, isto é, as aplicações recebem e enviam mensagens independentemente da ordem de chegada. O modelo de simulação garante este padrão pois, há um lugar, o lugar Travel System (TS), onde são realizadas periodicamente a entrada de novos *tokens*, independentemente da chegada destes nos demais lugares. Os lugares Email (MS), Fatura (IS), Voos (V) e Hotel (H), são recipientes de *tokens* enviados pelo lugar (TS) e não fornecem novos *tokens* ao sistema. Também há o caso do correlacionador que permite o fluxo somente de mensagens com conteúdo compatível. Para este requisito, as transições (8) e (9) consideram a lógica de que, somente é permitido a passagem de novos *tokens* caso o tempo de ambos sejam iguais, compatibilizando este processo com a forma exigida pela solução de integração. A comparação entre o comportamento esperado pelos especialistas da área e o experimento desenvolvido deixa evidente que há semelhança entre comportamentos.

A proposta de Sargent [38] para verificação do modelo é aplicada no próprio modelo de simulação. É proposto um experimento com 10.000 mil, 50.000 e 100.000 mil *tokens*. Cada experimento foi executado 25 vezes, e sua intenção é descrever o fluxo de mensagem, com intuito de verificar se o modelo está implementado corretamente na ferramenta.

Analisando os resultados das Tabelas em Anexo §5.15, §5.16, §5.17, §2, §3, §4, §5, §6, §7, §8, §9 e §10, pode se observar que em relação aos *tokens* que percorrem o caminho que leva ao lugar Email (MS), a quantidade de *tokens* que não foram filtrados pela transição (4) Filtro, corresponde em média aproximadamente 94% do total que foram encaminhados. Considerando o fato de que a função $Ok(r,s)$, contida no arco posterior à transição (4) Filtro, é uma função probabilística, e que a programação buscava filtrar 95% dos *tokens*, pode se concluir que a quantidade de *tokens* que chegaram ao lugar Email (MS) mais 5% é aproximadamente igual a quantidade de *tokens* observados nos demais lugares, Fatura (IS), Voos (V) e Hotel

(H). Também, do total de *tokens* incluídos no sistema, em cada cenário, aproximadamente 99% chegaram nos lugares Email (MS), Fatura (IS), Voos (V) e Hotel (H), logo, ao final de cada simulação, em média 1% dos *tokens* permaneceram dentro do sistema. Portanto, seguindo a proposta de verificação de Sargent [38], pode se concluir que há uma equivalência significativa nos eventos esperados no modelo conceitual e no modelo de simulação.

6.3 Resumo do Capítulo

Este capítulo apresentou a diferença entre validação e verificação, trazendo algumas técnicas utilizadas para verificar modelos de simulação em Redes de Petri.

Para verificar o modelo de simulação proposto nesta dissertação foram usadas as técnicas pesquisadas por Kleijen [28] e por Sargent [38]. Na qual a primeira envolve o conhecimento de um especialista na área para prever o comportamento da solução de integração, se o modelo de simulação se portar de modo similar, o modelo está correto. A segunda técnica analisa o próprio modelo de simulação.

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

Capítulo 7

Conclusões e Trabalhos Futuros

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

William Shakespeare, British author (1564-1616)

A integração de aplicações empresariais (EAI) é uma área em crescimento. As empresas estão buscando investir cada vez mais nas opções de soluções de integração, com a finalidade de interoperabilizar o seu sistema de *software*. Mas de fato, a implementação de uma solução de integração entre aplicações heterogeneas, é uma tarefa complexa. Se torna necessário antes de qualquer implementação, ainda a fase de projeto, conseguir informações consistentes sobre o desempenho da solução de integração. Esta pesquisa propôs um modelo de simulação em redes de Petri Temporizadas para uma solução de integração de uma Agência de Viagens. O trabalho encontrou, a partir do modelo conceitual, a equivalência entre os componentes das soluções e as redes de Petri Temporizadas. A partir disso foi possível realizar simulações do modelo e conseguir resultados.

Pelos resultados de simulação, que estão ilustrados nas Tabelas [§5.15](#), [§5.16](#), [§5.17](#), [§2](#), [§3](#), [§4](#), [§5](#), [§6](#), [§7](#), [§8](#), [§9](#) e [§10](#), não se observou gargalos de desempenho mas, o modelo de simulação ofereceu informações importantes sobre a Qualidade de Serviço (QoS) do modelo de simulação. O fato de não se observar acúmulo excessivo no modelo de simulação, principalmente nos lugares que antecedem as transições (8) e (9), que representam os correlacionadores, se deve ao Modelo de Distribuição de Entrada de *tokens*. O modelo

faz com o fluxo dos *tokens* ocorra de forma constante, então as transições (8) e (9) são habilitadas em períodos mais curtos em relação às outras transições. Por mais que estejam condicionando a passagem, estas transições disparam na mesma frequência que as demais. A Qualidade de Serviço (QoS) oferecida pelo modelo demonstra que, para os cenários estipulados, existe um intervalo de tempo [15, 25], em que 66% dos *tokens* obtiveram valores dentro dessa faixa, sendo que o maior número de *tokens* obtiveram tempo igual a 21.

Logo pode se conjecturar que, dado um caminho qualquer pertencente a uma rede de Petri Temporizada, com transições temporizadas, associando valores de tempo entre 1 e 5, inclusive o 1 e o 5 nos *tokens*, com valores aleatórios obedecendo a função de probabilidade P , o valor de maior probabilidade e que forma mais sequências é o elemento que ocupa a posição que corresponde a 60% do valor máximo. Isto quer dizer que as contribuições deste trabalho se concentram no fato de ter conseguido construir um modelo em redes de Petri Temporizadas e além disso analisar a Qualidade de Serviço (QoS) oferecida. O trabalho submeteu o modelo de simulação sob cenários críticos, realizou repetidos testes e verificou que o modelo condiz com o comportamento do modelo conceitual. O trabalho pode explorar os resultados de simulação e constatar que o modelo transcrito no CPN Tools é altamente adaptável em relação à programação das transições. Por consequência é possível expandir a aplicação das funções de probabilidades para outras mais complexas que forneceriam resultados mais ajustados à realidade da solução de integração.

Para trabalhos futuros é possível aplicar funções de distribuição aleatórias, disponíveis pela ferramenta CPN Tools, conforme mostra a Figura §7.1. Estas funções podem ser exploradas e aplicadas para se obter mais faixas de valores de tempo. Isto pode trazer como resultado tempos mais específicos e próximos à solução de integração.

7.1 Resumo das Contribuições

Esta dissertação teve objetivo de analisar o comportamento de uma solução de integração antes da sua implementação, com o intuito de ajudar os engenheiros de *software* a perceber os possíveis erros e gargalos de performance, na etapa de projeto, evitando gastos financeiros e de tempo, promovendo a qualidade da solução de integração proposta, por meio de um modelo conceitual e um modelo de simulação. Nesse sentido as principais contribuições desta pesquisa são listadas abaixo:

<code>bernoulli(p:real) : int</code>	raises Bernoulli exception, if $p < 0.0$ or $p > 1.0$.
<code>binomial(n:int, p:real) : int</code>	raises Binomial exception, if $n < 1$ or $p < 0.0$ or $p > 1.0$.
<code>chisq(n:int) : real</code>	raises chisq exception, if $n < 1$.
<code>discrete (a:int, b:int) : int</code>	raises Discrete exception, if $a > b$.
<code>erlang (n:int, r:real) : real</code>	raises Erlang exception, if $n < 1$ or $r <= 0.0$.
<code>exponential(r:real) : real</code>	raises Exponential exception, if $r <= 0.0$.
<code>normal(n:real, v:real) : real</code>	raises Normal exception, if $v < 0.0$.
<code>poisson(m:real) : int</code>	raises Poisson exception, if $m <= 0.0$.
<code>student (n:int) : real</code>	raises Student exception, if $n < 1$.
<code>uniform(a:real, b:real) : real</code>	raises Uniform exception, if $a > b$.
<code>rayleigh(s:real) : real</code>	raises Rayleigh exception, if $s < 0.0$.
<code>gamma(l:real, k:real) : real</code>	raises Gamma exception, if $k <= 0.0$ or $l <= 0.0$.
<code>beta(a:real, b:real) : real</code>	raises Beta exception, if $a <= 0.0$ or $b <= 0.0$.

Figura 7.1: Imagem das opções de funções de distribuição aleatórias disponíveis no CPN Tools [2].

- Análise do comportamento e identificação de possíveis gargalos de performance em soluções de integração, por meio da simulação do seu modelo conceitual, utilizando o formalismo matemático da Rede de Petri Temporizada.
- Aplicação dos conceitos de Qualidade de Serviço (QoS) na solução de integração em redes de Petri Temporizadas.
- Desenvolvimento de um modelo de simulação equivalente ao modelo conceitual elaborado pela Tecnologia Guaraná.
- Verificação do modelo de simulação proposto por meio de técnicas formais existentes na literatura.

.1 ANEXOS

P	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	T
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
7	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	3
8	1	0	1	2	0	0	2	2	0	0	1	1	1	0	0	0	1	3	2	0	3	5	1	0	0	26
9	1	3	1	5	1	1	1	2	4	3	3	2	3	0	0	1	0	1	0	5	3	1	2	4	50	
10	4	5	3	1	3	5	3	3	5	4	5	5	3	4	3	1	3	9	5	6	4	4	7	2	99	
11	9	7	11	8	4	3	7	8	9	5	1	1	8	10	5	11	5	5	4	10	4	7	8	8	11	169
12	10	7	11	10	13	15	9	15	12	14	18	18	23	15	13	10	11	12	7	18	12	12	12	8	12	317
13	17	19	24	22	20	25	20	14	23	22	24	24	25	17	16	19	18	20	16	20	27	18	19	20	24	513
14	33	32	34	26	29	28	24	35	28	29	19	19	42	29	29	32	20	36	30	25	25	22	27	25	21	699
15	40	44	42	28	34	36	39	39	34	35	41	41	43	33	34	34	35	30	36	41	33	45	32	36	29	914
16	51	43	42	45	60	60	46	29	42	32	39	39	59	37	46	49	49	43	42	35	46	48	47	44	49	1122
17	46	46	47	48	41	56	45	63	53	73	53	53	53	45	48	39	57	57	52	57	51	48	57	51	45	1284
18	59	45	37	55	62	49	57	51	47	54	63	63	49	52	46	58	51	40	56	53	54	44	50	58	54	1307
19	48	46	56	53	45	46	46	45	51	37	46	46	53	57	47	48	65	59	48	50	44	48	45	51	51	1231
20	35	45	38	49	45	29	48	40	35	53	43	43	53	43	47	57	38	38	37	42	52	46	51	57	41	1105
21	28	42	37	33	37	33	31	35	38	41	39	39	29	28	47	32	42	30	31	39	34	32	37	34	43	891
22	31	34	21	20	23	25	31	30	28	16	24	24	25	25	21	26	18	33	29	20	33	21	23	23	35	639
23	24	18	23	23	19	17	17	24	23	16	15	15	17	27	25	17	25	16	28	15	17	23	17	17	13	491
24	8	9	15	11	9	15	9	11	10	8	8	9	15	15	10	7	13	14	12	5	12	8	13	11	266	
25	4	6	7	11	6	8	11	2	5	8	8	8	1	5	8	3	5	5	12	6	7	7	11	4	162	
26	4	2	4	4	4	4	5	3	5	2	5	5	2	7	3	9	3	3	4	2	0	8	2	2	2	94
27	2	3	2	2	1	2	1	2	3	1	1	1	3	2	1	1	4	2	0	4	1	2	1	0	3	45
28	1	0	1	1	0	0	2	1	0	0	0	0	2	1	1	1	3	1	2	1	0	0	0	1	0	19
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
T1	457	456	457	457	456	458	454	456	456	455	456	456	455	455	455	457	456	455	457	456	457	456	457	456	457	11402
TE	461	463	463	462	463	462	460	463	463	462	462	462	462	462	464	464	464	463	463	462	462	462	461	462	463	11560

Figura 2: Resultado da simulação demonstrando a quantidade de tokens armazenados no lugar Fatura (IS) após 10.000 disparos e 25 repetições.

P	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	T
6	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	4
7	0	1	1	1	1	1	1	3	2	1	1	1	1	1	3	1	0	0	1	0	0	0	0	0	1	22
8	5	6	5	4	4	2	1	4	5	2	4	4	2	1	4	5	2	3	2	4	3	4	4	3	4	87
9	6	6	14	13	8	6	7	10	10	10	13	8	6	7	10	5	7	9	13	10	8	5	9	6	5	211
10	16	29	14	12	24	21	20	12	17	23	12	24	21	20	12	22	16	28	18	29	26	18	16	18	16	484
11	39	30	35	35	35	43	39	39	33	41	35	35	43	39	39	29	52	47	34	38	37	35	30	46	44	952
12	65	69	50	58	62	88	64	71	67	67	58	62	88	64	71	63	86	59	63	54	51	64	67	65	68	1644
13	97	94	105	102	94	100	94	96	80	107	102	94	100	94	96	89	101	89	105	80	81	94	88	110	102	2394
14	134	121	125	145	129	140	129	167	141	137	145	129	140	129	167	146	130	134	138	134	132	139	168	110	147	3476
15	121	187	183	182	187	176	183	173	186	188	182	187	176	183	173	199	186	203	181	172	192	171	176	163	172	4482
16	207	205	214	219	222	209	220	227	219	224	219	222	209	220	227	213	222	242	225	236	236	222	221	210	235	5525
17	251	253	244	263	269	228	244	215	243	232	263	269	228	244	215	246	234	238	262	254	272	230	232	242	233	6104
18	262	262	262	271	267	261	256	264	257	231	271	267	261	256	264	248	251	255	253	252	261	287	257	230	274	6480
19	227	264	261	237	224	269	261	232	244	246	237	224	269	261	232	261	244	240	234	231	208	257	265	259	225	6112
20	216	236	212	217	221	198	215	231	227	224	217	221	198	215	231	219	236	196	217	225	243	207	233	221	195	5471
21	199	185	193	170	190	173	198	179	189	190	170	190	173	198	179	179	163	169	180	162	163	184	210	188	4553	
22	128	134	143	136	138	149	134	131	138	151	136	138	149	134	131	127	115	144	159	164	154	151	135	131	147	3497
23	87	97	93	94	113	97	90	106	94	96	94	113	97	90	106	106	97	103	78	97	97	97	97	111	115	2460
24	62	47	65	69	51	56	63	61	64	61	69	51	56	63	61	55	62	69	77	63	56	70	50	71	48	1520
25	42	26	32	37	25	35	40	32	38	29	37	25	35	40	32	35	39	30	34	28	35	42	30	31	41	850
26	22	18	22	16	14	21	20	19	20	12	16	14	21	20	19	16	12	20	15	21	21	19	18	21	19	456
27	7	12	12	4	7	9	5	9	5	10	4	7	9	5	9	12	8	8	7	10	8	4	8	27	6	212
28	3	4	2	3	2	3	3	5	2	3	2	3	2	3	3	5	5	2	3	1	1	4	3	1	0	69
29	0	0	0	1	0	1	0	1	1	1	2	0	1	1	1	2	4	4	1	3	2	0	0	2	0	27
30	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	2	0	0	1	7
T1	2287	2286	2288	2288	2288	2286	2288	2286	2285	2287	2288	2288	2286	2288	2286	2284	2288	2286	2289	2286	2286	2286	2286	2288	2286	57170
TE	2294	2291	2293	2296	2293	2291	2293	2292	2290	2292	2296	2292	2291	2292	2292	2292	2292	2291	2297	2292	2291	2291	2292	2294	2294	57314

Figura 3: Resultado da simulação demonstrando a quantidade de tokens armazenados no lugar Fatura (IS) após 50.000 disparos e 25 repetições.

P	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	T	
6	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	2	1	0	1	0	1	0	0	0	0	8	
7	1	0	3	2	0	2	0	1	2	1	4	1	3	1	1	0	2	4	0	3	5	1	3	1	1	42	
8	6	8	2	6	8	2	3	8	6	10	7	9	4	8	5	4	9	10	8	9	6	2	6	5	5	156	
9	20	21	14	19	12	19	17	16	17	18	23	13	15	22	17	6	20	16	15	13	13	21	15	19	15	416	
10	47	41	43	34	44	49	37	23	45	42	32	33	33	44	30	41	42	34	36	41	42	33	32	34	36	948	
11	82	77	68	78	77	83	73	91	62	62	77	64	78	80	72	70	80	78	71	66	87	58	79	61	59	1833	
12	126	121	125	135	121	124	138	109	122	105	136	143	142	119	130	135	114	112	108	120	116	115	124	143	133	3116	
13	196	224	214	193	180	200	211	200	184	219	176	178	177	184	194	206	189	201	198	210	200	199	173	193	170	4869	
14	277	252	267	257	273	268	285	280	273	258	281	310	311	258	266	301	275	267	287	252	273	281	257	304	277	6890	
15	335	355	357	365	309	370	360	386	369	354	356	346	371	363	358	352	355	359	387	360	353	385	368	354	338	8965	
16	450	468	468	417	440	454	450	485	434	458	421	467	455	418	411	442	421	438	419	427	452	432	436	424	435	11022	
17	485	501	480	505	513	538	468	471	479	495	540	481	498	490	501	464	478	490	498	496	526	496	482	479	494	12348	
18	522	504	498	525	514	463	520	477	513	505	513	520	511	508	497	506	496	524	518	528	513	515	538	491	526	12745	
19	499	472	497	501	485	450	494	524	474	500	482	492	481	488	519	505	484	530	482	512	460	520	505	502	510	12368	
20	403	453	456	449	459	426	428	383	458	433	455	458	427	473	470	425	474	459	439	461	424	450	441	436	440	11080	
21	360	366	339	390	377	360	387	369	366	347	361	352	343	372	370	378	377	331	366	371	369	323	385	365	393	9117	
22	309	288	263	269	270	313	260	283	282	277	259	289	280	300	282	296	290	276	305	281	256	292	314	289	270	7093	
23	175	166	186	186	196	197	170	197	213	209	201	181	188	192	206	180	210	194	178	191	212	220	167	208	196	4819	
24	133	123	146	114	143	136	117	125	136	130	131	113	122	137	111	141	110	121	114	122	126	125	107	118	132	3133	
25	70	80	77	73	89	72	85	83	73	89	69	78	73	68	88	66	85	65	84	60	78	53	84	80	76	1898	
26	46	34	38	36	48	36	36	38	36	32	33	30	40	27	31	40	34	38	42	32	35	30	38	38	48	916	
27	19	13	21	15	10	9	20	21	22	24	17	19	18	17	12	10	19	17	12	15	20	15	12	16	14	407	
28	9	3	10	4	3	3	10	6	4	8	4	3	4	3	4	4	8	7	5	3	7	5	7	8	6	138	
29	3	1	0	1	2	2	5	2	0	1	0	2	1	4	1	1	1	5	0	1	0	1	1	2	0	37	
30	2	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0	10
TI	4575	4572	4572	4574	4573	4577	4575	4579	4572	4578	4578	4582	4575	4576	4575	4574	4576	4574	4574	4574	4574	4573	4574	4572	4574	114374	
TE	4582	4579	4579	4580	4581	4581	4581	4584	4578	4584	4584	4588	4582	4583	4581	4583	4581	4582	4584	4580	4579	4579	4579	4579	4580	114533	

Figura 4: Resultado da simulação demonstrando a quantidade de tokens armazenados no lugar Fatura (IS) após 100.000 disparos e 25 repetições.

P	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	T
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	2
12	0	0	0	1	0	0	0	2	1	0	0	1	1	0	0	0	0	0	0	0	2	0	0	1	2	11
13	1	1	1	0	1	1	0	4	0	1	1	1	0	3	0	1	1	3	0	0	1	2	0	0	1	24
14	2	4	3	2	0	2	0	2	0	5	3	3	3	4	1	4	3	3	4	1	2	1	4	4	0	60
15	3	6	7	6	5	5	7	6	5	5	7	7	4	2	3	4	1	3	2	5	3	5	3	3	4	111
16	3	17	7	10	9	11	4	5	9	5	7	7	13	5	4	9	11	7	4	6	3	5	4	8	7	180
17	11	19	2	8	8	13	2	6	13	11	2	2	6	8	2	4	9	12	10	11	13	12	10	12	7	213
18	15	26	16	15	17	21	10	11	14	12	16	16	17	21	20	16	7	11	9	18	17	17	13	6	16	377
19	21	31	22	14	23	29	24	24	24	20	22	22	23	22	28	14	20	27	20	30	13	15	24	12	21	545
20	32	28	18	36	20	23	37	30	28	31	18	18	28	29	37	18	33	32	37	31	28	24	29	24	27	696
21	39	38	38	32	31	37	38	41	28	30	38	38	42	39	38	55	45	31	38	40	39	47	42	37	33	954
22	41	46	47	34	41	41	36	38	35	35	47	47	35	34	40	39	35	37	32	27	42	39	44	43	38	937
23	39	39	50	42	46	41	47	36	43	51	50	50	45	44	34	50	37	39	39	36	45	38	40	41	49	1071
24	45	56	46	47	47	44	48	49	48	44	46	46	41	44	43	37	42	46	43	47	39	45	37	54	42	1126
25	45	22	48	49	57	41	49	49	46	45	48	48	46	37	42	50	39	51	41	45	53	37	35	40	43	1106
26	34	25	31	39	32	41	31	39	34	46	31	31	48	44	36	34	34	33	39	33	45	39	50	36	35	920
27	43	26	37	26	34	34	36	24	40	31	37	37	31	28	35	23	42	37	36	34	24	30	35	42	44	846
28	32	24	28	34	21	19	30	31	18	24	28	28	23	17	26	26	30	31	34	36	22	22	26	23	26	659
29	31	19	23	22	24	20	15	15	27	18	23	23	23	30	20	28	23	20	24	14	25	28	16	24	21	556
30	13	11	12	18	15	12	17	20	13	20	12	12	8	15	15	13	17	8	14	12	19	23	12	15	14	360
31	11	10	9	9	9	8	12	6	11	10	9	9	8	14	10	16	8	13	14	18	12	13	15	13	10	277
32	0	2	5	5	5	9	5	6	10	6	5	5	3	3	7	7	9	5	7	8	5	3	5	10	6	141
33	1	2	3	1	5	2	3	7	4	2	3	3	3	3	12	3	5	3	5	1	2	4	6	5	3	91
34	0	3	1	4	2	1	1	2	1	2	1	1	0	2	1	1	2	2	1	1	1	2	2	1	1	36
35	1	0	0	1	1	0	0	3	1	0	0	0	0	3	0	1	1	0	1	0	0	2	0	1	1	17
36	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	6
37	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	4
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	2
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T1	454	455	454	455	456	455	453	454	454	455	454	454	452	452	454	454	454	455	456	455	455	454	453	455	452	11354
TE	461	463	462	462	463	462	460	463	463	462	462	462	462	462	464	464	464	463	463	462	462	462	461	462	463	11559

Figura 5: Resultado da simulação demonstrando a quantidade de tokens armazenados no lugar Voos (V) após 10.000 disparos e 25 repetições.

P	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	T
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	3
10	1	2	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	9
11	0	1	0	2	0	2	0	0	3	0	2	0	2	0	0	2	0	0	0	1	0	2	1	1	0	19
12	0	2	2	2	4	3	1	4	3	2	2	4	3	1	3	0	4	3	3	2	1	2	2	3	58	
13	5	2	5	6	7	5	3	3	1	1	6	7	5	3	3	10	7	6	8	6	3	2	4	4	4	116
14	6	9	10	7	12	8	3	9	13	14	7	12	8	3	9	8	12	17	1	8	8	9	8	10	6	217
15	24	25	15	17	23	17	9	17	15	21	17	23	17	9	17	22	11	23	18	11	14	20	18	13	23	439
16	33	27	36	35	35	31	30	26	31	29	35	35	31	30	26	34	48	32	32	28	29	33	26	30	34	796
17	52	46	49	52	44	51	55	47	52	56	52	44	51	55	47	41	56	64	37	52	46	55	51	41	63	1259
18	92	79	80	84	83	83	94	82	82	93	84	83	83	94	82	84	68	79	98	79	84	80	82	85	90	2107
19	107	103	121	93	107	91	99	91	99	109	93	107	91	99	91	110	106	102	105	126	104	100	128	97	102	2581
20	143	128	144	124	141	138	124	135	154	168	124	141	138	124	135	141	139	142	147	135	118	127	136	149	132	3427
21	171	169	163	176	184	172	187	195	165	168	176	184	172	187	195	182	156	170	175	177	171	178	194	191	175	4433
22	183	234	181	217	204	209	199	201	179	208	217	204	209	199	201	199	203	207	188	189	205	200	203	182	205	5026
23	213	231	211	233	204	207	214	225	195	191	233	204	207	214	225	204	218	218	238	204	226	226	204	188	232	5365
24	225	243	214	212	212	235	222	207	207	217	212	212	235	222	207	217	238	225	201	227	216	242	204	209	204	5465
25	214	219	226	245	206	226	212	220	241	224	245	206	226	212	220	219	227	208	226	228	219	180	223	218	224	5514
26	194	191	191	173	207	201	187	199	176	188	173	207	201	187	199	198	194	193	188	180	216	184	199	218	195	4839
27	170	165	180	191	186	170	161	173	187	159	191	186	170	161	173	165	167	176	161	162	184	187	177	175	185	4362
28	134	117	152	128	145	143	148	130	143	130	128	145	143	148	130	123	137	127	153	164	122	145	137	148	125	3445
29	106	91	112	94	114	106	115	112	121	105	94	114	106	115	112	107	107	114	104	109	128	103	111	119	109	2728
30	85	76	73	74	79	69	101	84	89	81	74	79	69	101	84	76	82	60	93	67	74	86	74	90	66	1986
31	60	57	48	51	39	45	45	57	58	61	51	39	45	45	57	53	48	56	44	54	40	56	38	48	45	1240
32	38	31	34	37	20	42	33	36	33	26	37	20	42	33	36	39	25	29	33	37	36	35	25	34	33	824
33	13	18	24	17	17	13	16	14	16	16	17	17	13	16	14	22	15	19	17	19	25	16	22	16	16	428
34	6	12	9	6	8	9	13	12	10	11	6	8	9	13	12	13	10	5	9	11	7	7	11	11	5	233
35	8	6	1	6	4	3	9	3	4	6	6	4	3	9	3	6	4	4	5	6	3	4	4	3	4	118
36	2	0	3	2	1	3	1	1	2	2	2	1	3	1	1	2	5	3	0	0	3	3	0	1	1	43
37	0	0	0	1	1	1	2	2	2	0	1	1	1	2	2	1	2	1	1	0	0	0	0	1	0	22
38	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	4
39	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	3
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T1	2285	2284	2285	2286	2287	2284	2286	2282	2283	2287	2286	2287	2284	2286	2282	2281	2286	2284	2286	2285	2283	2283	2285	2285	2282	57114
TE	2294	2291	2293	2296	2293	2291	2293	2292	2290	2292	2296	2293	2291	2292	2292	2292	2292	2291	2297	2292	2291	2291	2292	2294	2294	57315

Figura 6: Resultado da simulação demonstrando a quantidade de tokens armazenados no lugar Voos (V) após 50.000 disparos e 25 repetições.

P	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	T	
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	
9	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2
10	0	0	0	1	4	0	1	0	1	0	1	0	2	0	0	0	0	1	1	0	2	0	0	0	0	14	
11	2	0	1	2	1	2	0	0	1	0	4	2	3	2	2	3	1	0	4	4	0	4	0	3	3	44	
12	1	6	4	0	4	6	7	1	2	4	3	4	4	5	5	2	4	6	5	8	1	1	0	4	1	88	
13	10	9	7	6	7	13	10	9	10	10	11	5	12	11	13	10	7	8	11	11	8	6	7	9	8	228	
14	23	19	18	18	22	26	14	15	20	15	19	24	18	30	26	9	17	18	19	21	18	17	21	25	15	487	
15	39	39	35	37	31	30	41	39	35	35	31	34	43	39	34	6	32	33	36	39	35	34	24	35	31	847	
16	68	59	85	62	56	69	75	79	65	70	67	73	46	73	55	72	63	72	63	55	64	79	51	73	60	1654	
17	117	128	116	122	115	120	95	109	111	108	108	101	104	97	94	103	106	101	107	95	108	108	107	93	110	2683	
18	144	166	143	148	147	149	147	141	154	154	165	156	173	147	150	157	152	147	144	164	173	145	155	151	160	3832	
19	195	239	200	195	207	226	212	222	224	241	192	211	228	210	200	232	218	210	220	222	190	201	217	230	213	5355	
20	280	273	263	280	279	275	259	255	285	295	245	266	268	274	259	296	287	270	289	293	259	284	245	289	279	6847	
21	341	335	324	343	321	377	329	320	356	333	364	358	349	307	350	327	340	364	348	338	317	320	366	331	362	8520	
22	405	385	408	391	404	383	392	407	393	404	425	392	440	406	407	396	362	418	417	447	390	388	414	388	342	10004	
23	419	414	438	439	427	426	425	427	431	419	417	457	431	453	450	453	411	442	389	435	447	461	454	421	439	10825	
24	436	422	426	463	454	426	449	430	411	431	447	448	433	464	433	464	471	455	431	452	465	441	441	469	465	11127	
25	435	438	423	416	452	427	424	450	446	378	465	431	427	398	464	386	418	427	415	440	410	441	429	416	438	10694	
26	406	423	398	380	407	382	427	402	381	407	386	349	376	381	412	419	396	425	413	383	396	428	393	410	381	9961	
27	352	314	348	349	367	322	355	374	337	325	366	339	331	362	316	332	328	322	344	318	391	331	326	337	369	8555	
28	288	287	321	291	302	292	297	285	289	304	284	288	293	287	289	264	294	258	280	278	265	306	276	300	263	7181	
29	220	206	212	209	208	210	193	203	217	230	202	222	209	214	225	218	251	210	204	188	222	201	231	199	229	5333	
30	191	166	175	171	145	167	159	162	148	146	136	168	150	152	163	143	155	165	148	161	168	151	164	153	165	3972	
31	117	98	90	112	89	99	104	103	108	100	107	109	97	116	87	108	98	87	129	99	102	98	104	99	97	2557	
32	69	75	70	60	66	64	76	64	62	81	63	69	70	64	71	62	80	64	67	54	61	65	63	63	65	1668	
33	37	34	34	40	30	42	43	34	40	39	35	33	35	36	30	46	39	41	50	34	40	23	42	41	46	944	
34	19	19	21	27	16	25	24	21	26	25	14	21	21	29	25	15	28	21	23	18	22	21	30	15	18	544	
35	9	9	6	6	8	9	12	12	12	12	14	13	5	11	9	10	9	5	8	10	10	11	6	10	9	235	
36	7	2	2	2	3	5	1	7	2	4	5	2	6	4	2	4	4	2	4	6	4	3	5	4	94		
37	2	3	1	1	0	2	1	2	2	3	1	4	0	1	2	0	0	0	4	0	3	1	3	2	2	40	
38	0	1	1	0	1	0	1	1	1	0	0	0	0	1	1	0	0	1	1	0	2	0	0	0	0	13	
39	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	4	
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
T1	4573	4569	4570	4572	4573	4575	4572	4575	4570	4574	4577	4579	4574	4573	4574	4571	4572	4574	4572	4572	4571	4572	4572	4571	4574	114321	
TE	4582	4579	4579	4580	4581	4581	4581	4584	4578	4584	4584	4588	4582	4583	4581	4583	4581	4582	4584	4580	4579	4579	4579	4579	4580	114533	

Figura 7: Resultado da simulação demonstrando a quantidade de tokens armazenados no lugar Voos (V) após 100.000 disparos e 25 repetições.

P	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	T	
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	2
11	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2
12	0	2	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	3	0	0	1	1	0	0	13	
13	0	0	0	0	0	1	0	2	1	1	0	0	0	1	0	1	1	0	1	0	1	2	2	0	0	14	
14	2	4	2	4	4	5	2	1	2	2	2	2	2	2	1	3	1	2	2	5	3	0	2	5	4	64	
15	2	6	7	3	2	3	2	3	8	2	7	7	4	6	4	1	1	5	4	3	4	3	3	5	2	97	
16	5	4	5	11	9	16	5	7	12	7	5	5	9	6	9	8	7	8	6	10	5	7	9	3	6	184	
17	6	6	10	6	7	10	6	11	6	9	10	10	4	17	5	12	16	9	10	7	18	10	8	9	9	231	
18	8	15	8	14	11	22	20	10	20	18	8	8	15	17	13	14	10	12	11	17	21	13	12	18	11	346	
19	17	18	23	21	21	17	27	38	19	20	23	23	29	19	23	25	18	25	36	23	22	21	21	19	16	564	
20	41	32	28	20	43	18	22	31	36	32	28	28	22	24	24	30	22	25	23	33	37	27	29	20	23	698	
21	38	31	39	34	38	38	31	30	26	28	39	39	42	26	33	31	28	33	30	28	28	30	44	39	36	839	
22	31	36	31	54	43	45	32	32	43	46	31	31	34	58	45	35	36	45	40	40	38	37	36	45	35	979	
23	46	42	43	36	41	51	37	33	33	42	43	43	47	42	44	46	47	37	50	41	41	43	31	41	46	1046	
24	46	33	51	43	49	45	43	48	50	37	51	51	43	39	35	47	52	43	48	42	54	39	50	34	32	1105	
25	44	52	35	34	50	41	52	38	38	41	35	35	41	39	40	44	52	47	40	35	34	46	43	51	44	1051	
26	45	43	46	35	36	29	41	37	38	44	46	46	45	29	43	39	33	36	36	53	26	44	37	37	49	993	
27	28	43	39	39	25	33	41	30	32	39	39	39	26	34	33	34	30	36	34	42	38	27	40	44	34	879	
28	39	23	26	35	27	27	25	18	29	32	26	26	28	24	33	26	29	28	30	25	25	27	34	30	35	707	
29	22	21	20	24	17	19	23	13	16	16	20	20	28	25	27	24	29	29	20	16	20	25	17	21	27	539	
30	15	20	17	19	13	9	22	5	21	14	17	17	13	16	12	12	10	13	13	13	23	21	14	12	18	379	
31	8	8	13	8	6	16	8	4	11	11	13	13	8	7	15	10	11	4	10	9	9	13	11	10	9	245	
32	6	5	6	5	6	4	4	4	4	8	6	6	6	5	7	6	7	7	7	5	3	4	4	7	9	141	
33	3	5	5	4	2	3	4	1	4	1	5	5	3	9	2	3	8	5	7	4	0	8	3	1	5	100	
34	1	1	0	4	4	0	2	1	2	1	0	0	2	2	2	0	2	3	2	3	4	0	1	1	1	40	
35	2	3	0	2	1	3	2	2	0	1	0	0	1	2	1	1	2	1	0	1	0	1	1	1	1	29	
36	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	2	1	0	1	0	0	0	0	0	6	
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	3	
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
T1	455	453	454	455	456	456	453	455	454	455	454	454	453	450	452	455	452	454	455	455	453	453	454	454	453	11347	
TE	461	463	462	462	463	462	460	462	463	463	462	462	462	462	464	464	464	463	463	462	462	462	461	462	463	11559	

Figura 8: Resultado da simulação demonstrando a quantidade de tokens armazenados no lugar Hotel (H) após 10.000 disparos e 25 repetições.

P	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	T
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
10	0	0	0	1	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	0	0	1	0	0	0	10
11	0	2	0	3	0	0	0	2	4	0	3	0	0	0	2	0	0	2	0	0	0	0	0	2	1	21
12	1	2	3	0	0	2	3	1	2	5	0	2	3	1	0	4	1	0	1	3	2	2	2	2	1	41
13	7	4	4	5	6	7	3	7	6	8	5	6	7	3	7	4	5	13	5	4	6	7	4	2	2	137
14	11	13	16	7	9	7	18	8	7	8	7	9	7	18	8	15	11	13	13	5	11	10	8	6	14	259
15	23	18	20	21	21	20	20	13	19	22	21	21	20	20	13	17	28	23	15	25	14	19	15	14	25	487
16	30	37	28	30	33	33	28	33	27	30	30	33	33	28	33	38	35	34	31	33	28	29	41	24	35	794
17	53	46	39	46	52	59	39	51	45	41	46	52	59	39	51	56	58	40	64	56	50	38	47	60	62	1249
18	85	71	75	54	78	68	81	67	76	94	54	78	68	81	67	90	80	67	66	77	87	75	76	82	77	1874
19	107	104	99	122	124	129	105	127	116	100	122	124	129	105	127	112	110	113	119	112	108	109	138	106	94	2861
20	147	149	137	146	120	126	126	134	128	143	146	120	126	116	134	147	130	162	158	157	139	144	143	138	148	3464
21	166	180	159	170	194	170	172	179	170	174	170	194	1770	172	179	172	181	179	167	168	185	159	156	165	188	5939
22	189	200	190	190	207	179	188	182	192	205	190	207	179	188	182	191	171	177	183	183	204	187	189	161	178	4692
23	193	229	216	212	229	213	180	220	219	224	212	229	213	180	220	234	210	221	232	213	197	242	223	233	229	5423
24	199	209	223	231	215	249	230	228	205	229	231	215	249	230	228	212	233	228	235	215	222	269	227	221	215	5648
25	220	241	226	214	214	212	232	204	228	229	214	214	212	232	204	206	238	203	195	236	221	223	230	215	215	5478
26	206	192	200	235	209	201	217	205	187	186	235	209	201	217	205	178	196	209	191	201	182	177	194	200	209	5042
27	170	181	182	171	171	164	185	169	160	168	171	171	164	185	169	176	179	159	162	154	174	172	158	170	169	4254
28	171	124	164	114	133	159	148	132	163	108	114	133	159	148	132	130	120	136	136	150	144	122	145	166	147	3498
29	122	99	105	97	100	108	109	118	114	116	97	100	108	109	118	95	107	120	115	98	109	105	100	102	83	2654
30	71	73	79	87	66	69	77	85	86	84	87	66	69	77	85	68	80	64	85	89	84	61	76	100	75	1943
31	51	50	54	61	40	46	44	56	50	42	61	40	46	44	56	62	35	55	42	44	54	57	44	49	57	1240
32	32	38	29	34	26	28	37	28	31	24	34	26	28	37	28	35	37	32	40	39	29	32	28	32	33	797
33	16	11	22	22	26	25	24	17	26	28	22	26	25	24	17	25	22	16	12	14	14	23	19	16	10	502
34	10	6	9	10	9	3	12	10	8	10	10	9	3	12	10	9	8	10	10	7	12	3	12	12	7	221
35	4	3	3	3	4	4	4	4	5	7	3	4	4	4	5	2	3	6	3	5	4	6	5	4	103	
36	0	2	3	1	0	1	3	2	5	0	1	0	1	3	2	4	4	3	1	0	0	3	2	3	46	
37	0	0	0	0	0	2	1	1	1	1	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	10
38	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	3
39	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T1	2284	2284	2286	2287	2287	2284	2286	2283	2282	2287	2287	2287	2284	2286	2283	2280	2286	2285	2287	2285	2282	2283	2285	2287	2282	57119
TE	2294	2291	2293	2296	2293	2291	2293	2292	2290	2292	2296	2293	2291	2292	2292	2292	2292	2291	2297	2292	2291	2291	2292	2294	2294	57315

Figura 9: Resultado da simulação demonstrando a quantidade de tokens armazenados no lugar Hotel (H) após 50.000 disparos e 25 repetições.

P	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	T	
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	3
10	0	1	0	0	0	1	2	0	1	0	3	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	11
11	0	1	1	2	2	1	3	2	4	0	4	2	0	0	0	5	1	0	1	4	0	0	2	3	1	39	
12	5	4	7	4	1	7	3	6	2	6	0	4	6	5	4	3	2	2	4	5	3	3	4	4	1	95	
13	8	12	8	10	5	8	9	9	7	9	11	13	16	16	9	11	4	5	9	15	15	4	6	11	7	237	
14	19	16	19	17	23	20	19	19	23	20	19	24	18	17	28	17	15	17	16	22	18	15	16	15	25	477	
15	41	37	40	50	37	43	46	39	37	33	34	39	38	32	39	35	44	34	44	36	38	38	18	32	29	933	
16	63	60	55	55	79	70	76	64	67	52	62	60	60	66	54	65	58	73	64	58	76	65	62	57	59	1580	
17	96	113	101	97	91	111	112	93	103	108	106	109	109	100	110	104	85	104	111	100	102	103	110	116	100	2594	
18	161	167	162	149	164	138	156	175	162	146	116	144	159	161	168	145	154	179	147	159	127	171	151	174	163	3898	
19	212	239	221	220	212	233	220	208	225	201	236	205	207	203	188	235	215	213	219	213	235	217	208	212	191	5388	
20	295	295	284	270	261	274	245	272	187	284	260	293	305	285	259	303	278	303	290	289	278	308	306	289	289	7002	
21	334	346	331	350	345	336	331	340	326	347	329	385	314	360	322	327	331	354	338	341	329	322	363	365	297	8463	
22	425	399	388	388	410	388	380	390	390	432	393	363	422	377	397	387	436	415	362	414	399	410	377	375	416	9933	
23	441	404	425	446	428	453	444	416	412	435	434	427	438	392	427	461	427	416	448	413	414	427	432	441	445	10746	
24	392	420	426	450	437	467	495	415	461	440	455	420	461	452	425	449	424	414	466	449	464	466	406	433	464	11051	
25	394	440	432	457	426	398	403	449	437	401	431	442	437	454	440	394	420	413	439	426	426	461	430	443	395	10688	
26	410	398	407	377	378	401	367	425	389	406	451	427	380	412	439	405	402	396	354	419	389	406	403	380	431	10052	
27	329	349	352	376	381	342	329	330	308	332	347	351	343	357	362	340	361	305	368	333	360	306	383	372	378	8694	
28	317	288	287	251	293	285	305	282	299	290	269	276	261	270	291	277	278	288	295	287	267	258	274	271	287	7046	
29	238	202	221	214	236	199	213	222	225	239	215	187	213	228	217	225	224	230	211	220	229	215	206	198	195	5422	
30	152	147	145	147	143	163	144	157	161	151	166	153	151	150	148	146	171	178	155	139	167	161	141	144	155	3835	
31	105	89	102	107	95	98	118	115	111	98	113	121	95	94	103	88	101	101	89	103	102	95	113	99	98	2553	
32	58	72	70	71	67	71	84	72	61	60	67	57	77	78	73	82	65	59	60	55	60	58	75	60	76	1688	
33	43	40	41	30	29	39	39	44	30	41	32	36	31	33	39	32	40	40	38	32	38	26	36	46	34	909	
34	18	17	29	20	14	19	21	17	26	22	17	18	12	17	19	19	15	24	29	22	20	17	26	18	20	496	
35	8	8	11	8	8	5	2	9	11	16	8	10	9	13	6	10	13	8	8	10	7	13	20	10	9	246	
36	6	4	4	6	5	2	3	5	4	1	2	6	5	2	5	3	3	2	2	4	6	6	2	2	3	93	
37	2	2	1	1	2	0	4	1	1	6	0	4	4	2	1	2	3	0	3	2	2	1	1	1	3	49	
38	1	0	0	0	1	2	0	0	0	0	0	1	1	0	1	2	2	1	0	1	0	0	1	0	0	14	
39	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
T1	4573	4570	4570	4573	4573	4574	4573	4576	4570	4576	4577	4579	4572	4576	4574	4572	4572	4575	4571	4572	4571	4572	4572	4571	4573	114327	
TE	4582	4579	4579	4580	4581	4581	4581	4584	4578	4584	4584	4588	4582	4583	4581	4583	4581	4582	4584	4580	4579	4579	4579	4579	4580	114533	

Figura 10: Resultado da simulação demonstrando a quantidade de tokens armazenados no lugar Hotel (H) após 100.000 disparos e 25 repetições.

Bibliografia

- [1] *Plataforma de simulação pipe2*, 2008. URL: <http://pipe2.sourceforge.net>. Acessado em 03/06/2016.
- [2] *Cpn tools*, 2010. URL: <http://cpntools.org/>. Acessado em 10/11/2016.
- [3] L. Albuquerque, P. Maciel, T. d. Medeiros e J. Castro. *Uso de redes de petri temporizadas para o diagnostico de falhas em sistemas digitais. Simpósio de Computadores Tolerantes a Falhas*, 15(6), 1995.
- [4] P. Bradác. *Model-driven enterprise application integration*. Dissertação de Mestrado, Slovak University of Technology in Bratislava, 2011.
- [5] J. L. S. C. Martin Grinstead. *Introduction to probability*. American Mathematical Soc., 2012.
- [6] O. C. L. Carlos Augusto de Alcantara Gomes. *Modelagem dos sistemas com redes de petri coloridas: Algoritmo para cálculo das datas de entrada de fichas nas posições e sensibilização das transições. XXXVII Simpósio brasileiro de Pesquisa Operacional*, (37), 2005.
- [7] L. G. Chaves, R. Baroni e M. Ferreira. *Análise da integração de aplicações (eai) no contexto de portais corporativos de médias e grandes empresas brasileiras. SMSI II*, 2005.
- [8] L. Chwif. *Redução de modelos de simulação de eventos discretos na sua concepção: uma abordagem causal*. Tese Doutoral, Universidade de São Paulo, 1999.
- [9] L. Chwif e A. C. Medina. *Modelagem e simulação de eventos discretos*. Afonso C. Medina, 2006.
- [10] S. L. David. *Next generation application integration*, 2004.

- [11] F. V. de A. Guerra. *Modelagem de sistemas com restrições temporais em redes de petri orientadas a objetos*. Dissertação de Mestrado, Universidade Federal de Campina Grande, 2005.
- [12] M. V. de Aguiar. *Modelagem e análise de sistemas de manufatura flexível por meio de redes de petri colorida, hierárquica e temporizada*. Dissertação de Mestrado, Universidade Federal do Amazonas, 2008.
- [13] R. S. B. dos Santos. *Modelagem e análise de performance de sistemas flexíveis de manufatura baseado em redes de petri temporizadas: Estudo de caso na indústria automobilística*. Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo, 2008.
- [14] D. Dossot, J. D’Emic e V. Romero. *Mule in action*. Manning, 2014.
- [15] V. Ermagan, I. Krüger e M. Menarini. *A fault tolerance approach for enterprise applications*. Em *IEEE SCC*, páginas 63–72, 2008.
- [16] J. S. C. et al. *Model verification and validation*. *Simulation Conference, Proceedings of the Winter*, 1(1):52–58, 2002.
- [17] R. Z. F. S. S. V. B.-F. Fabricia Roos-Frantz, Manuel Binele. *Using petri nets to enable the simulation of application integration solutions conceptual models*. *ICEIS 2015 - 17th International Conference on Enterprise Information Systems*, (17), 2015.
- [18] M. Fisher, J. Partner, M. Bogoevici e I. Fuld. *Spring integration in action*. Manning Publications Co., 2012.
- [19] C. R. L. Francês. *Introdução às redes de petri*. *Laboratório de Computação Aplicada, Universidade Federal do Pará*, 2003.
- [20] 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.
- [21] R. Z. Frantz. *Enterprise application integration - an easy-to-maintain model-driven engineering approach*. Tese Doutoral, Universidad de Sevilla, 2012.
- [22] W. V. R. Hamilton Fernandes de Mattias. *Metodologia para análise de desempenho de sistemas de computação descritos através de redes de petri temporizadas estendidas*. *VIII Simpósio brasileiro de Redes de Computadores*, (8), 1990.

- [23] H. L. J. P. Hilano José Rocha de Carvalho, Andrea Ribari Yoshizawa. *Análise de desempenho do trabalho multinacional em linhas de produção em forma de U pela modelagem e simulação usando redes de petri temporizadas*. XXXVII Simpósio Brasileiro de Pesquisa Operacional, (37), 2005.
- [24] G. Hohpe e B. Woolf. *Enterprise integration patterns*. Em *9th Conference on Pattern Language of Programs*, páginas 1–9, 2002.
- [25] G. Hohpe e B. Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.
- [26] G. Hohpe e B. Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.
- [27] C. Ibsen e J. Anstey. *Camel in action*. Manning Publications Co., 2010.
- [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] A. Law e D. Kelton. *Simulation modeling and analysis*. McGraw-Hill, 1991.
- [30] D. S. Linthicum. *Enterprise application integration*. Addison-Wesley Professional, 2000.
- [31] R. R. Machado. *Avaliação do desempenho logístico do transporte rodoviário de madeira utilizando rede de petri*. Dissertação de Mestrado, Universidade Federal de Viçosa, 2006.
- [32] J. R. S. Pedro M. G. del Foyo, José A. S. P. Miralles. *Um verificador formal eficiente para sistemas de tempo real*. X Simpósio brasileiro de Automação Inteligente, (10), 2011.
- [33] D. H. e. C. O. d. F. d. e. A. P. d. S. Penha, Freitas e Martins. *Modelagem de sistemas computacionais usando redes de petri: aplicação em projeto, análise e avaliação*. Pontifícia Universidade Católica de Minas Gerais, 1(01), 2004.
- [34] R. S. Pressman. *Engenharia de software - uma abordagem profissional*. Bookman, 2009.

- [35] N. C. Quental. *Modelagem de desempenho de programas paralelos utilizando redes de petri temporizadas*. Dissertação de Mestrado, Escola Politécnica da Universidade de Pernambuco, 2006.
- [36] J. L. Salvi. *Relacionamentos temporais entre redes de petri e planejamento automático*. Dissertação de Mestrado, Universidade Federal do Paraná, 2009.
- [37] V. B. F. F. R. F. I. Y. R. C. Sandro Sawiki, Rafael Z. Frantz. *Characterising enterprise application integration solutions as discrete-event systems*. 2016.
- [38] R. G. Sargent. *Verification and validation of simulation models*. Em *Proceedings of the 37th conference on Winter simulation*, páginas 130–143. winter simulation conference, 2005.
- [39] S. Sawicki, R. Z. Frantz, V. M. B. Fernandes, F. Roos-Frantz, I. Yevseyeva e R. Corchuelo. *Characterising enterprise application integration solutions as discreteevent system*. IGI Global, páginas 2–5, 2015.
- [40] A. K. Wiesner. *Modelagem e simulação de uma solução de integração para identificação de gargalos de desempenho baseadas em formalismo matemático: uma abordagem orientada à teoria das filas*. Dissertação de Mestrado, Universidade Regional do Noroeste do Estado do Rio Grande do Sul, 2016.

This document was typeset on August 1, 2017 using class `RG-BOK` $\alpha 2.14$ for `LATEX2 ϵ` . 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.