

DIEGO HIROKI UEDA  
JOSÉ CAETANO DE BARROS JUNIOR

**SISTEMA COMPUTACIONAL PARA MODELAGEM E  
SIMULAÇÃO DISTRIBUÍDAS DE MODELOS EM REDE DE PETRI**

Monografia apresentada à Escola  
Politécnica da Universidade de São  
Paulo para Conclusão do Curso.

São Paulo  
2010

DIEGO HIROKI UEDA  
JOSÉ CAETANO DE BARROS JUNIOR

**SISTEMA COMPUTACIONAL PARA MODELAGEM E  
SIMULAÇÃO DISTRIBUÍDAS DE MODELOS EM REDE DE PETRI**

Monografia apresentada à Escola  
Politécnica da Universidade de São  
Paulo para Conclusão do Curso.

Curso de Graduação:  
Engenharia Mecatrônica

Orientador:  
Prof. Dr. Fabrício Junqueira

São Paulo  
2010

Aos meus pais e minha irmã.

*Diego Hiroki Ueda*

Aos meus colegas, familiares,  
amigos e camaradas.

*José Caetano de Barros Junior*

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DA  
ENGENHARIA MECÂNICA/NAVAL DA ESCOLA POLITÉCNICA (EPMN) –  
USP.

**Ueda, Diego Hiroki**

**Sistema computacional para modelagem e simulação distri -  
buídas de modelos em rede de Petri / D.H. Ueda, J.C. de Barros  
Junior. -- São Paulo, 2010.**

**83 p.**

**Trabalho de Formatura - Escola Politécnica da Universidade  
de São Paulo. Departamento de Engenharia Mecatrônica e de  
Sistemas Mecânicos.**

**1. Simulação distribuída 2. Simulação de sistemas 3. Redes  
de Petri 4. Redes de computadores I. Barros Junior, José  
Caetano de II.Universidade de São Paulo. Escola Politécnica.  
Departamento de Engenharia Mecatrônica e de Sistemas  
Mecânicos III. t.**

## **Agradecimentos**

Ao nosso orientador Prof. Dr. Fabrício Junqueira pela sua constante orientação e pelas inúmeras sugestões que colaboraram com a execução deste trabalho.

Aos professores da Escola Politécnica da USP, que nos forneceram conhecimentos e contribuíram para a nossa formação.

A toda a minha família pelo carinho e apoio desde sempre.

Aos meus amigos e colegas de graduação pela companhia e amizade ao longo destes anos.

A todos aqueles que contribuíram de forma direta ou indiretamente na produção deste trabalho.

(Diego)

Aos meus familiares que me apoiaram nesse período, aos meus colegas de turma pela ajuda que me forneceram, e a todos que acreditaram e investiram suas esperanças em mim.

(José Caetano)

## Sumário

<b>LISTA DE FIGURAS.....</b>	<b>II</b>
<b>LISTA DE TABELAS.....</b>	<b>V</b>
<b>RESUMO.....</b>	<b>VI</b>
<b>ABSTRACT.....</b>	<b>VII</b>
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1 OBJETIVO.....	2
<b>2 CONCEITOS DE FERRAMENTAS .....</b>	<b>3</b>
2.1 UML (UNIFIED MODELING LANGUAGE) .....	3
2.1.1 <i>Diagrama de classes</i> .....	4
2.1.2 <i>Diagrama de casos de uso</i> .....	6
2.1.3 <i>Diagrama de sequência</i> .....	6
2.2 REDE DE PETRI .....	7
2.2.1 <i>Estrutura de uma Rede de Petri</i> .....	8
2.2.2 <i>Grafo de Rede de Petri</i> .....	8
2.2.3 ‘ <i>Marcação</i> ’ de uma Rede de Petri.....	9
2.2.4 <i>Regras para execução da Rede de Petri</i> .....	10
2.2.5 <i>Tipos de Redes de Petri</i> .....	11
2.2.6 <i>Linguagem PNML</i> .....	12
2.3 SIMULAÇÃO DISTRIBUÍDA .....	13
2.3.1 <i>Algoritmos de sincronização</i> .....	14
2.3.2 <i>Protocolo Token Ring</i> .....	15
2.3.3 <i>Interface entre modelos</i> .....	16
2.4 VISUAL STUDIO .....	18
<b>3 PROJETO DO AMBIENTE DE MODELAGEM E SIMULAÇÃO.....</b>	<b>19</b>
3.1 ESPECIFICAÇÃO DOS REQUISITOS.....	19
3.2 MODELAGEM UML.....	21
3.2.1 <i>Diagrama de Casos de uso</i> .....	21
3.2.2 <i>Diagrama de Classes</i> .....	22
3.2.3 <i>Diagrama de Sequência</i> .....	23
<b>4 ESTRUTURA DO AMBIENTE DE MODELAGEM E SIMULAÇÃO.....</b>	<b>30</b>
4.1 LINGUAGEM PARA DESCRIÇÃO DA RDP .....	30
4.2 INTERFACE DE USUÁRIO .....	30
4.3 SERVIDOR/CLIENTE .....	35
4.4 MECANISMO PARA O GERENCIAMENTO DA SIMULAÇÃO DISTRIBUÍDA .....	35
<b>5 TESTES DE SIMULAÇÃO.....</b>	<b>39</b>
5.1 SIMULAÇÃO LOCAL OU ISOLADA .....	39
5.2 SIMULAÇÃO DISTRIBUÍDA ENTRE DUAS ESTAÇÕES.....	43
5.3 SIMULAÇÃO DISTRIBUÍDA ENTRE TRÊS ESTAÇÕES .....	56
<b>6 CONCLUSÕES .....</b>	<b>74</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>75</b>

## Lista de Figuras

FIGURA 2.1 - DIAGRAMAS DEFINIDOS PELA UML ADAPTADO DE OMG (2007) .....	4
FIGURA 2.2 - EXEMPLO DE REPRESENTAÇÃO DE UMA CLASSE ADAPTADO DE (BOOCH, ET AL., 2005) .....	5
FIGURA 2.3 - ILUSTRAÇÃO DAS FORMAS DE ASSOCIAÇÕES (JUNQUEIRA, 2006) .....	6
FIGURA 2.4 - EXEMPLO DE UM DIAGRAMA DE CASOS DE USO.....	7
FIGURA 2.5 - EXEMPLO DE UM DIAGRAMA DE SEQUÊNCIA.....	7
FIGURA 2.6 - EXEMPLO DE GRAFO DE REDE DE PETRI ADAPTADO DE COSTA (1995).....	10
FIGURA 2.7 - (A) EXEMPLO DE UM MODELO EM RDP (B) DESCRIÇÃO EM PNML (FREITAS; SILVA, 2006) .....	13
FIGURA 2.8 - EXEMPLO DE FUNCIONAMENTO DE UMA REDE TOKEN RING (JUNQUEIRA, 2006) .....	16
FIGURA 3.1 - CLASSIFICAÇÃO DOS REQUISITOS NÃO-FUNCIONAIS ADAPTADO DE (SOMMERVILLE, 2006) .....	21
FIGURA 3.2 - CASOS DE USO DO AMBIENTE DE MODELAGEM E SIMULAÇÃO .....	21
FIGURA 3.3 - CLASSES, INTERFACES E DELEGATES.....	22
FIGURA 3.4 – ENVIAR DADOS PELA REDE .....	25
FIGURA 3.5 - RECEBER DADOS EM UMA REDE .....	26
FIGURA 3.6 - REALIZAR A CONEXÃO A UM SERVIDOR.....	27
FIGURA 3.7 - ABRIR PROJETO.....	28
FIGURA 3.8 - SALVAR PROJETO .....	29
FIGURA 4.1 - INTERFACE DE USUÁRIO COM EXEMPLO DE RDP .....	31
FIGURA 4.2 - FUNÇÕES À DIREITA DA INTERFACE DE USUÁRIO. HABILITAÇÃO DE NOVAS FUNÇÕES AO SELECIONAR: (A) UM 'LUGAR', (B) UMA 'TRANSIÇÃO' E (C) UM 'ARCO' .....	33
FIGURA 4.3 - COMANDOS PRESENTES NO MENU "ARQUIVO" .....	34
FIGURA 4.4 - COMANDOS PRESENTES NO MENU "SIMULAÇÃO" .....	34
FIGURA 4.5 - COMANDOS PRESENTES NO MENU "COMUNICAÇÃO" .....	34
FIGURA 4.6 - JANELA PARA CONECTAR-SE COM OUTRA ESTAÇÃO .....	35
FIGURA 5.1 - MODELO A SER SIMULADO .....	39
FIGURA 5.2 - MENSAGENS EXIBIDAS APÓS A PREPARAÇÃO DO MODELO .....	40
FIGURA 5.3 - EVOLUÇÃO DA RDP APÓS O DISPARO DA 'TRANSIÇÃO' FUNDIDA .....	41
FIGURA 5.4 - MENSAGEM APÓS O DISPARO DA 'TRANSIÇÃO' .....	41
FIGURA 5.5 - EVOLUÇÃO DA RDP APÓS O SEGUNDO DISPARO DA 'TRANSIÇÃO' FUNDIDA .....	41
FIGURA 5.6 - MENSAGEM APÓS O SEGUNDO DISPARO DA 'TRANSIÇÃO' FUNDIDA.....	41
FIGURA 5.7 - DISPARO DA 'TRANSIÇÃO' TEMPORIZADA .....	42
FIGURA 5.8 - MENSAGENS EXIBIDAS APÓS O DISPARO DA 'TRANSIÇÃO' TEMPORIZADA.....	42
FIGURA 5.9 – SEGUNDO DISPARO DA 'TRANSIÇÃO' TEMPORIZADA.....	42
FIGURA 5.10 - MENSAGENS EXIBIDAS APÓS O SEGUNDO DISPARO DA 'TRANSIÇÃO' TEMPORIZADA.....	43
FIGURA 5.11 - AMBOS OS MODELOS EM DEADLOCK INDICAM O FIM DA SIMULAÇÃO.....	43
FIGURA 5.12 - MODELO A SER SIMULADO .....	43
FIGURA 5.13 – ANEL LÓGICO ENTRE DUAS ESTAÇÕES .....	44
FIGURA 5.14 - CONFIGURAÇÃO DE CONEXÃO DA "ESTAÇÃO A" .....	44
FIGURA 5.15 - CONFIGURAÇÃO DE CONEXÃO DA "ESTAÇÃO B" .....	45
FIGURA 5.16 - MODELO A SER SIMULADO NA "ESTAÇÃO A" .....	46
FIGURA 5.17 - MENSAGENS EXIBIDAS APÓS O PREPARO DA RDP NA "ESTAÇÃO A" .....	46
FIGURA 5.18 - MODELO A SER SIMULADO NA "ESTAÇÃO B" .....	47
FIGURA 5.19 - MENSAGENS EXIBIDAS APÓS O PREPARO DA RDP NA "ESTAÇÃO B" .....	47
FIGURA 5.20 - DISPARO DA 'TRANSIÇÃO' FUNDIDA NA "ESTAÇÃO A" .....	48
FIGURA 5.21 - MENSAGENS EXIBIDAS NA "ESTAÇÃO A" APÓS O DISPARO DA 'TRANSIÇÃO' FUNDIDA .....	48
FIGURA 5.22 - DISPARO DA 'TRANSIÇÃO' FUNDIDA NA "ESTAÇÃO B".....	49
FIGURA 5.23 - MENSAGENS EXIBIDAS NA "ESTAÇÃO B" APÓS O DISPARO DA 'TRANSIÇÃO' FUNDIDA .....	49
FIGURA 5.24 - SEGUNDO DISPARO DA 'TRANSIÇÃO' FUNDIDA NA "ESTAÇÃO A" .....	50
FIGURA 5.25 - MENSAGENS EXIBIDAS NA "ESTAÇÃO A" APÓS O SEGUNDO DISPARO DA 'TRANSIÇÃO' FUNDIDA.....	50
FIGURA 5.26 - SEGUNDO DISPARO DA 'TRANSIÇÃO' FUNDIDA NA "ESTAÇÃO B" .....	51
FIGURA 5.27 - MENSAGENS EXIBIDAS NA "ESTAÇÃO B" APÓS O SEGUNDO DISPARO DA 'TRANSIÇÃO' FUNDIDA.....	51

FIGURA 5.28 - NÃO HOUVE DISPARO DE 'TRANSIÇÕES' NA "ESTAÇÃO A", MAS A VARIÁVEL TEMPO ALTEROU-SE PARA 3 .....	52
FIGURA 5.29 - MENSAGENS EXIBIDAS NA "ESTAÇÃO A" .....	52
FIGURA 5.30 - DISPARO DA 'TRANSIÇÃO' TEMPORIZADA NA "ESTAÇÃO B" E ALTERAÇÃO DO TEMPO PARA 3.....	52
FIGURA 5.31 - MENSAGENS EXIBIDAS NA "ESTAÇÃO B" .....	53
FIGURA 5.32 - NÃO HOUVE DISPARO DE 'TRANSIÇÕES' NA "ESTAÇÃO A", MAS A VARIÁVEL TEMPO ALTEROU-SE DE 3 PARA 6.....	53
FIGURA 5.33 - MENSAGENS EXIBIDAS NA "ESTAÇÃO A" .....	54
FIGURA 5.34 - DISPARO DA 'TRANSIÇÃO' TEMPORIZADA NA "ESTAÇÃO B" E ALTERAÇÃO DO TEMPO DE 3 PARA 6 .....	54
FIGURA 5.35 - MENSAGENS EXIBIDAS NA "ESTAÇÃO B" .....	55
FIGURA 5.36 - MENSAGENS EXIBIDAS NA "ESTAÇÃO A" APÓS TENTATIVA DE EXECUÇÃO DE MAIS UM PASSO.....	55
FIGURA 5.37 - MENSAGENS EXIBIDAS NA "ESTAÇÃO B" APÓS TENTATIVA DE EXECUÇÃO DE MAIS UM PASSO.....	56
FIGURA 5.38 - MODELO A SER SIMULADO ENTRE TRÊS ESTAÇÕES .....	56
FIGURA 5.39 - ANEL LÓGICO ENTRE TRÊS ESTAÇÕES .....	57
FIGURA 5.40 - CONFIGURAÇÃO DE CONEXÃO DA "ESTAÇÃO A".....	57
FIGURA 5.41 - CONFIGURAÇÃO DE CONEXÃO DA "ESTAÇÃO B" .....	58
FIGURA 5.42 - CONFIGURAÇÃO DE CONEXÃO DA "ESTAÇÃO C".....	58
FIGURA 5.43 - MODELO A SER SIMULADO NA "ESTAÇÃO A" .....	59
FIGURA 5.44 - MENSAGENS EXIBIDAS APÓS O PREPARO DA RDP NA "ESTAÇÃO A" .....	59
FIGURA 5.45 - MODELO A SER SIMULADO NA "ESTAÇÃO B" .....	60
FIGURA 5.46 - MENSAGENS EXIBIDAS APÓS O PREPARO DA RDP NA "ESTAÇÃO B".....	60
FIGURA 5.47 - MODELO A SER SIMULADO NA "ESTAÇÃO B" .....	61
FIGURA 5.48 - MENSAGENS EXIBIDAS APÓS O PREPARO DA RDP NA "ESTAÇÃO C".....	61
FIGURA 5.49 – NÃO HOUVE DISPARO DE 'TRANSIÇÕES' NA "ESTAÇÃO A" .....	62
FIGURA 5.50 - MENSAGENS EXIBIDAS NA "ESTAÇÃO A" APÓS O PRIMEIRO PASSO .....	62
FIGURA 5.51 - DISPARO DA 'TRANSIÇÃO' FUNDIDA DA "ESTAÇÃO B" .....	62
FIGURA 5.52 – MENSAGENS EXIBIDAS NA "ESTAÇÃO B" APÓS O DISPARO DA 'TRANSIÇÃO' FUNDIDA ...	63
FIGURA 5.53 - DISPARO DA 'TRANSIÇÃO' FUNDIDA DA "ESTAÇÃO C" .....	63
FIGURA 5.54 - MENSAGENS EXIBIDAS NA "ESTAÇÃO C" APÓS O DISPARO DA 'TRANSIÇÃO' FUNDIDA ...	64
FIGURA 5.55 - NÃO HOUVE DISPARO DE 'TRANSIÇÕES' NA "ESTAÇÃO A" .....	64
FIGURA 5.56 - MENSAGENS EXIBIDAS NA "ESTAÇÃO A" APÓS O SEGUNDO PASSO .....	65
FIGURA 5.57 – SEGUNDO DISPARO DA 'TRANSIÇÃO' FUNDIDA DA "ESTAÇÃO B" .....	65
FIGURA 5.58 - MENSAGENS EXIBIDAS NA "ESTAÇÃO B" APÓS O SEGUNDO DISPARO DA 'TRANSIÇÃO' FUNDIDA.....	65
FIGURA 5.59 – SEGUNDO DISPARO DA 'TRANSIÇÃO' FUNDIDA DA "ESTAÇÃO C" .....	66
FIGURA 5.60 - MENSAGENS EXIBIDAS NA "ESTAÇÃO C" APÓS O SEGUNDO DISPARO DA 'TRANSIÇÃO' FUNDIDA.....	66
FIGURA 5.61 - NÃO HOUVE DISPARO DE 'TRANSIÇÕES' NA "ESTAÇÃO A", NO ENTANTO, O VALOR DE TEMPO FOI ATUALIZADO PARA 1 .....	67
FIGURA 5.62 - MENSAGENS EXIBIDAS NA "ESTAÇÃO A" APÓS O TERCEIRO PASSO.....	67
FIGURA 5.63 - NÃO HOUVE DISPARO DE 'TRANSIÇÕES' NA "ESTAÇÃO B", NO ENTANTO, O VALOR DE TEMPO FOI ATUALIZADO PARA 1 .....	68
FIGURA 5.64 - MENSAGENS EXIBIDAS NA "ESTAÇÃO B" APÓS O TERCEIRO PASSO.....	68
FIGURA 5.65 - DISPARO DA 'TRANSIÇÃO' TEMPORIZADA NA "ESTAÇÃO C" E ATUALIZAÇÃO DO TEMPO PARA VALOR 1 .....	68
FIGURA 5.66 - MENSAGENS EXIBIDAS NA "ESTAÇÃO C" APÓS O DISPARO DA 'TRANSIÇÃO' TEMPORIZADA .....	69
FIGURA 5.67 - DISPARO DA 'TRANSIÇÃO' TEMPORIZADA DA "ESTAÇÃO A" E ATUALIZAÇÃO DE VALOR TEMPO DE 1 PARA 2.....	69
FIGURA 5.68 - MENSAGENS EXIBIDAS NA "ESTAÇÃO C" APÓS O DISPARO DA 'TRANSIÇÃO' TEMPORIZADA .....	69
FIGURA 5.69 - NÃO HOUVE DISPARO DE 'TRANSIÇÕES' NA "ESTAÇÃO B", NO ENTANTO, O VALOR DE TEMPO FOI ATUALIZADO DE 1 PARA 2 .....	70
FIGURA 5.70 - MENSAGENS EXIBIDAS NA "ESTAÇÃO B" APÓS O QUARTO PASSO .....	70

FIGURA 5.71 – SEGUNDO DISPARO DA 'TRANSIÇÃO' TEMPORIZADA DA "ESTAÇÃO C" E ATUALIZAÇÃO DO VALOR TEMPO DE 1 PARA 2 .....	70
FIGURA 5.72 - MENSAGENS EXIBIDAS NA "ESTAÇÃO C" APÓS O SEGUNDO DISPARO DA 'TRANSIÇÃO' TEMPORIZADA.....	71
FIGURA 5.73 - DISPARO DA 'TRANSIÇÃO' TEMPORIZADA DA "ESTAÇÃO A" E ATUALIZAÇÃO DO VALOR TEMPO DE 2 PARA 3.....	71
FIGURA 5.74 - MENSAGENS EXIBIDAS NA "ESTAÇÃO A" APÓS O DISPARO DA 'TRANSIÇÃO' TEMPORIZADA .....	71
FIGURA 5.75 - NÃO HOUVE DISPARO DE 'TRANSIÇÕES' NA "ESTAÇÃO B", NO ENTANTO, O VALOR TEMPO FOI ATUALIZADO DE 2 PARA 3.....	72
FIGURA 5.76 - MENSAGENS EXIBIDAS NA "ESTAÇÃO B" APÓS O QUINTO PASSO .....	72
FIGURA 5.77 - NÃO HOUVE DISPARO DE 'TRANSIÇÕES' NA "ESTAÇÃO C", NO ENTANTO, O VALOR TEMPO FOI ATUALIZADO DE 2 PARA 3.....	72
FIGURA 5.78 - MENSAGENS EXIBIDAS NA "ESTAÇÃO C" APÓS O QUINTO PASSO .....	73
FIGURA 5.79 - MENSAGENS EXIBIDAS NA "ESTAÇÃO A" APÓS TENTATIVA DE EXECUÇÃO DE MAIS UM PASSO.....	73
FIGURA 5.80 - MENSAGENS EXIBIDAS NA "ESTAÇÃO B" APÓS TENTATIVA DE EXECUÇÃO DE MAIS UM PASSO.....	73
FIGURA 5.81 - MENSAGENS EXIBIDAS NA "ESTAÇÃO C" APÓS TENTATIVA DE EXECUÇÃO DE MAIS UM PASSO.....	73

## **Lista de Tabelas**

TABELA 2-1 - ELEMENTOS PNML (FREITAS; SILVA, 2006) .....	12
TABELA 4-1 - POSSÍVEIS VALORES PARA 'STATUS' (JUNQUEIRA, 2006) .....	36
TABELA 4-2 - VALORES POSSÍVEIS PARA O PARÂMETRO DEADLOCK .....	37

## Resumo

Com os crescentes avanços tecnológicos e com o desenvolvimento da tecnologia de comunicação, de informática e de mecatrônica, várias indústrias de manufatura estão se estabelecendo de maneira distribuída e dispersa. De forma a lidar com a descentralização das tarefas, maior especialização de cada tipo de processo e automatização das operações que envolvem interações complexas entre as partes distribuídas numa mesma planta ou dispersas geograficamente, faz-se necessário o uso de métodos eficientes como a modelagem e a simulação. A modelagem permite a construção de modelos para representar sistemas reais existentes ou hipotéticos, cujas características e comportamento/desempenho podem ser analisados por meio de simulações. Entre elas destaca-se a utilização da simulação distribuída, cuja execução de um único modelo de simulação é distribuída entre múltiplos processadores, de modo a reduzir significativamente o tempo necessário para se obter os resultados da simulação. O objetivo desse projeto é o estudo e o desenvolvimento de um sistema computacional destinado à modelagem e simulação distribuídas de sistemas produtivos baseados em Rede de Petri. Assim, utiliza-se a Rede de Petri para a modelagem dos elementos básicos do sistema, que por meio de uma interface gráfica amigável, o usuário poderá criar ou editar modelos, bem como executar simulações de forma isolada ou distribuída.

## Abstract

Due to the technological advances and the development of communication, information and mechatronics, many manufacturing industries have been established on a distributed and disperse way. Efficient methods like modeling and simulation may be used to deal with complex systems. These systems are characterized by decentralized tasks, the specialization of process and automation of operations that involves complex interactions between the distributed parts, which may be geographically dispersed. Modeling allows the development of models to represent existent or hypothetical real systems which features can be analyzed by simulations. Between the different types of simulation, the distributed simulation is related to the execution of a unique model that is distributed between multiple processors, in order to reduce significantly the time to get the simulation results. The project goal is the study and development of a computational system for distributed modeling and simulation of productive systems based on Petri Net. Therefore, Petri Net is used to model basic elements of the system, allowing the user to make and edit models by accessible graphical interface as well as executing simulations in a distributed or locally way.

## 1 Introdução

Devido à globalização do mercado e a necessidade de atender demandas locais, algumas empresas vêm se estabelecendo de maneira distribuída e dispersa. Assim, é estratégico que se definam formas eficientes para integrar os diversos sistemas que compõem a empresa e/ou integrar a empresa com outras entidades.

No entanto, nesse contexto em que os mercados estão se tornando mais globais e independentes de barreiras geográficas, uma integração entre vários sistemas heterogêneos implica em um aumento da complexidade do sistema resultante. De forma a lidar com esta complexidade e facilitar o estudo e projeto de novos sistemas, faz-se necessário o uso de modelos, que são representações abstratas de sistemas, onde são considerados apenas os elementos mais importantes para o estudo em questão (ARATA, 1994). Para modelos em que há uma grande variedade e quantidade de informações, tal que seja impossível desenvolver uma técnica analítico-dedutiva que atenda a todas as situações possíveis, uma saída é a utilização de técnicas de simulação (ARATA, 2005). Dentre essas técnicas, destaca-se a simulação distribuída, a qual trata da evolução de situações/cenários do sistema em computadores geograficamente dispersos, conectados por meio de uma rede de comunicação LAN (*Local Area Network*) ou WAN (*Wide Area Network*), visando a redução do tempo de simulação, o uso eficiente de recursos computacionais, a tolerância a falhas, entre outras razões (JUNQUEIRA, 2006).

Os modelos baseados em sistemas a eventos discretos (SEDs), cujas tarefas ou processos podem ser tratados como uma sucessão de estados discretos que são alterados devido à ocorrência de eventos vistos como instantâneos, isto é, a dinâmica do processo é governada pela ocorrência desses eventos discretos, são intensamente utilizados para a descrição e análise de sistemas produtivos. E, para o apoio à análise desses modelos, a Rede de Petri (RdP) é considerada uma técnica de comprovada eficiência que permite além da modelagem, a simulação do sistema, e o seu acompanhamento por meio de representação gráfica, além de ser capaz de

especificar funcionalmente sistemas que possuem características de concorrência, sincronismo e paralelismo (JUNQUEIRA, 2001).

### ***1.1 Objetivo***

Este trabalho tem assim o objetivo de implementar um sistema computacional destinado à modelagem e simulação distribuídas de sistemas produtivos baseados em Rede de Petri Lugar/Transição, onde os modelos em RdP do tipo Lugar/Transição distribuídos em diferentes computadores poderão ser executados.

## 2 Conceitos de ferramentas

A seguir são apresentados os principais conceitos, teorias, tecnologias e ferramentas consideradas para o desenvolvimento do presente trabalho.

Do ponto de vista da teoria, foram considerados inicialmente conceitos como Rede de Petri (RdP), UML (*Unified Modeling Language*), simulação distribuída.

No que se refere à implementação, foi utilizado como ferramenta o *software* Visual Studio 2010.

### 2.1 UML (*Unified Modeling Language*)

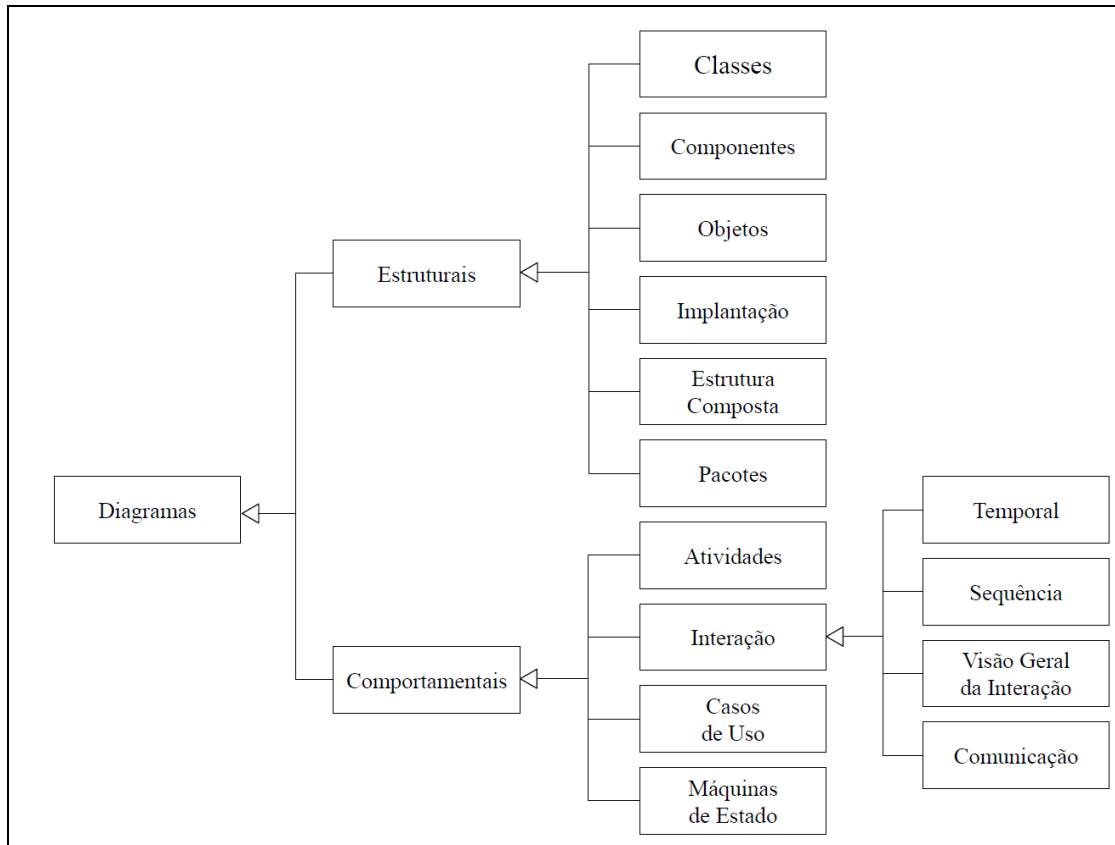
De acordo com (BOOCH *et al.*, 2005), a *Unified Modeling Language* (UML) pode ser definida como uma linguagem padrão para modelagem de projeto de *software*. A UML pode ser utilizada para visualizar, especificar, construir e documentar artefatos de um sistema de *software*.

A UML é uma linguagem visual, ou seja, constituída de elementos gráficos utilizados na modelagem, permitindo, assim, a construção de diagramas que representam as diversas perspectivas do sistema (BEZERRA, 2007).

A UML inclui 13 tipos de diagramas. Na Figura 2.3 são listados os diagramas UML.

Os diagramas UML mais utilizados são: de casos de uso, de classes, de sequência, de estados, de atividades, de componentes e de colaboração.

De acordo com os pesquisadores, não necessariamente todos os diagramas UML devem ser obrigatoriamente desenvolvidos e, no presente projeto em particular, o foco está nos diagramas de casos de uso, de classes e de sequência os quais serão apresentados a seguir.



**Figura 2.1 - Diagramas definidos pela UML adaptado de OMG (2007).**

### 2.1.1 Diagrama de classes

As classes são os mais importantes elementos de um sistema orientado a objetos (BOOCH *et al.*, 2005). Uma classe descreve um conjunto de objetos que compartilham os mesmos atributos, operações, relações e semânticas.

Graficamente, como ilustrado na Figura 2.4, uma classe possui um nome capaz de distingui-la de outras classes; atributo(s) que é uma propriedade identificada de uma classe e pode assumir um conjunto específico de valores; e operação(ões) que correspondem à descrição de ações que os objetos de uma classe podem executar. Uma classe implementa uma ou mais interfaces.

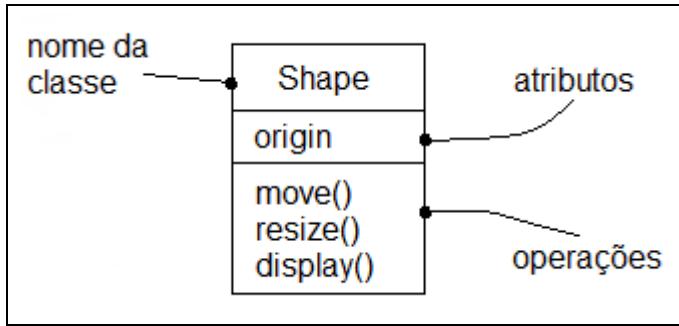
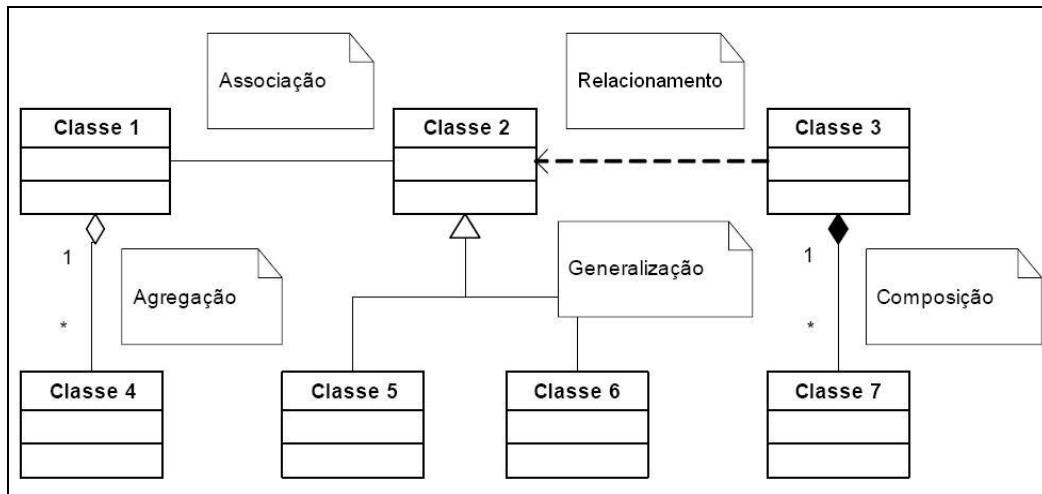


Figura 2.2 - Exemplo de representação de uma classe adaptado de (BOOCH, et al., 2005).

Para que haja comunicação entre os objetos, é necessário que exista algum tipo de associação (relacionamento) entre eles, que pode ser de cinco tipos e que estão ilustrados na Figura 2.5 (BEZERRA, 2007):

- **Associação:** Representa uma relação estrutural entre duas classes. No diagrama, uma associação é representada como uma linha sólida que unem duas classes.
- **Agregação:** Estabelece uma relação entre todo-parte, sendo que a parte pode existir sem o todo. É ilustrado como uma associação com um losango não preenchido no extremo relativo ao todo.
- **Composição:** É uma forma de agregação na qual as partes são inseparáveis do todo. É ilustrado como uma associação com um losango preenchido no extremo relativo ao todo.
- **Generalização:** É uma relação de herança em que uma classe (subclasse) herda os atributos e métodos de outra classe (superclasse). Uma generalização é ilustrada no diagrama como uma seta fechada não preenchida apontando para a superclasse.
- **Dependência:** Essa relação indica que uma classe utiliza as informações e serviços fornecidos por outra classe. No diagrama é representado por meio de uma linha tracejada com uma seta aberta ligando da classe dependente para a classe da qual ela depende.



**Figura 2.3 - Ilustração das formas de associações (JUNQUEIRA, 2006)**

Assim, a construção de um diagrama de classes envolve um conjunto de classes conectadas por meio de relacionamentos citados acima.

### 2.1.2 Diagrama de casos de uso

Esse diagrama ilustra o aspecto dinâmico do sistema, de forma a facilitar a visualização em um alto nível de abstração das funcionalidades do sistema. Assim, a finalidade é a representação dos elementos externos e os requisitos (ou parte destes) de um sistema e as maneiras segundo as quais eles as utilizam (BEZERRA, 2007).

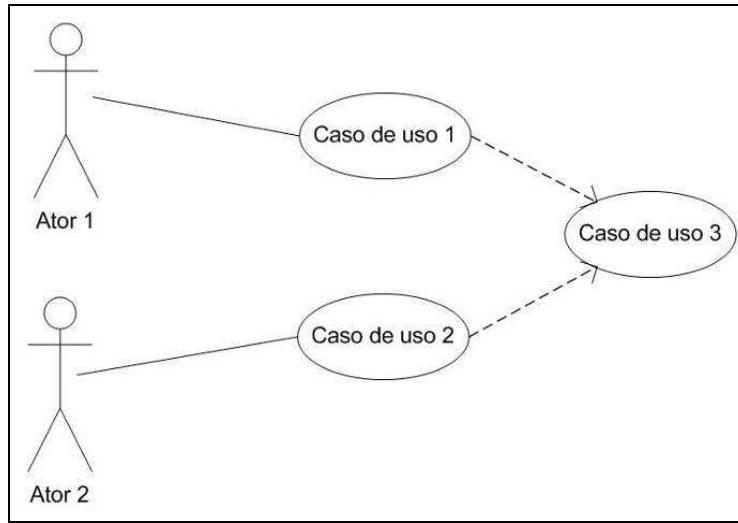
O diagrama de casos de uso é representado, graficamente, por atores, casos de uso, e relacionamento entre esses elementos.

A Figura 2.6 ilustra um exemplo de caso de uso.

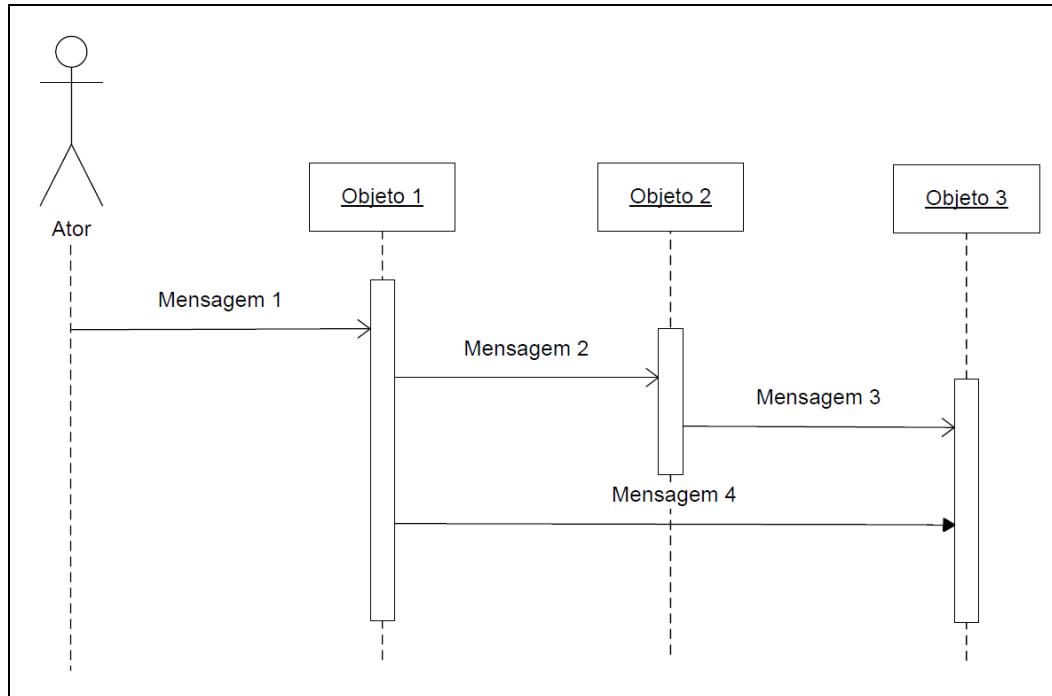
### 2.1.3 Diagrama de sequência

O diagrama de sequência mostra a ordem temporal das mensagens trocadas entre os objetos. A Figura 2.7 ilustra um exemplo de diagrama de sequência com os elementos básicos.

Nota-se que os atores presentes no diagrama de casos de uso podem ser apresentados no diagrama de sequência. Os atores são representados com a mesma notação gráfica utilizada no diagrama de casos de uso.



**Figura 2.4 - Exemplo de um diagrama de casos de uso**



**Figura 2.5 - Exemplo de um diagrama de seqüência**

## 2.2 **Rede de Petri**

A Rede de Petri (RdP) surgiu com o trabalho de doutorado de Carl Adam Petri que, posteriormente, com o surgimento das aplicações da teoria de RdP, uma série inesgotável de trabalhos ligados à rede de Petri tem sido produzida. Uma razão para isso é o vasto espectro de aplicação dessa rede (COSTA, 1995).

Segundo (MURATA, 1989), RdP é uma técnica de modelagem matemática e gráfica aplicável a muitos sistemas, com grande potencial para a descrição de sistemas de processamento de informações caracterizados como concorrentes, assíncronos, paralelos, distribuídos, não determinísticos e/ou estocásticos. Como uma técnica gráfica, a Rede de Petri pode ser usada como uma alternativa aos diagramas de fluxo de controle e de dados.

### 2.2.1 Estrutura de uma Rede de Petri

De acordo com (MURATA, 1989), uma RdP é formada por 4 partes: um conjunto de ‘lugares’  $P$ , um conjunto de ‘transições’  $T$ , uma função de entrada  $I$  e uma função de saída  $O$ . As funções de entrada e saída relacionam ‘transições’ e ‘lugares’. A função de entrada  $I$  é um mapeamento de uma ‘transição’  $t_j$  a uma coleção de ‘lugares’  $I(t_j)$  conhecidos como ‘lugares de entrada’, enquanto a função de saída  $O$  é um mapeamento de uma ‘transição’  $t_j$  a uma coleção de ‘lugares’  $O(t_j)$  conhecidos como ‘lugares de saída’.

Por definição, uma estrutura de Rede de Petri  $C$  é uma 4tupla,  $C = (P, T, I, O)$ , onde:

$P = \{p_1, p_2, \dots, p_n\}$  é um conjunto finito de ‘lugares’, onde  $n \geq 0$ .

$T = \{t_1, t_2, \dots, t_m\}$  é um conjunto finito de ‘transições’, onde  $m > 0$ .

O conjunto de ‘lugares’ e o conjunto de ‘transições’ são disjuntos, ou seja,  $P \cap T = \emptyset$ . Um ‘lugar’  $p_i$  é um ‘lugar de entrada’ de uma ‘transição’  $t_j$  se  $p_i \in I(t_j)$  e  $p_i$  é um ‘lugar de saída’ se  $p_i \in O(t_j)$ .

### 2.2.2 Grafo de Rede de Petri

O desenvolvimento de trabalhos em RdP é realizado geralmente com base na definição apresentada anteriormente. Contudo, a representação gráfica desta estrutura é a principal forma utilizada para ilustrar os conceitos de modo mais prático.

A Rede de Petri é representada por um multigrafo orientado e bipartido formado por ‘nós’ e ‘arcos’. Os ‘nós’ são representados por círculos, que correspondem aos ‘lugares’, ou por ‘barras’, que representam as ‘transições’. ‘Arcos orientados’ conectam ‘lugares’ e ‘transições’. Um ‘arco’ nunca leva diretamente de um ‘lugar’ para outro ‘lugar’, ou de uma ‘transição’ para outra

‘transição’. Ao lado de um ‘arco’ pode haver um número, denominado ‘peso’. Quando não há indicação desse número o seu ‘peso’ é 1.

‘Arcos’ que se direcionam de um ‘lugar’ para uma ‘transição’ definem o ‘lugar’ como sendo um ‘lugar de entrada’ da ‘transição’. Múltiplas entradas de uma ‘transição’ são indicados por múltiplos ‘arcos’ que partem de ‘lugares’ e chegam à ‘transição’. Um ‘lugar de saída’ é indicado por ‘arcos’ partindo de uma ‘transição’ e chegando no ‘lugar’. Da mesma forma, múltiplas ‘saídas’ de uma ‘transição’ são indicadas por múltiplos ‘arcos’ partindo da ‘transição’ e chegando aos ‘lugares de saída’ da mesma.

Uma RdP é um multi grafo por este permitir múltiplos ‘arcos’ de um ‘nó’ para outro. Em adição, desde que os ‘arcos’ são direcionados, o grafo é dito direcionado. Considerando-se ainda que seus ‘nós’ são divididos em dois conjuntos distintos, e que os ‘arcos’ unem sempre um elemento de um conjunto a outro de outro conjunto, dizemos que o grafo é bipartido. Assim, a denominação multigrafo direcionado bipartido é justa. (PETERSON, 1977).

### 2.2.3 ‘Marcação’ de uma Rede de Petri

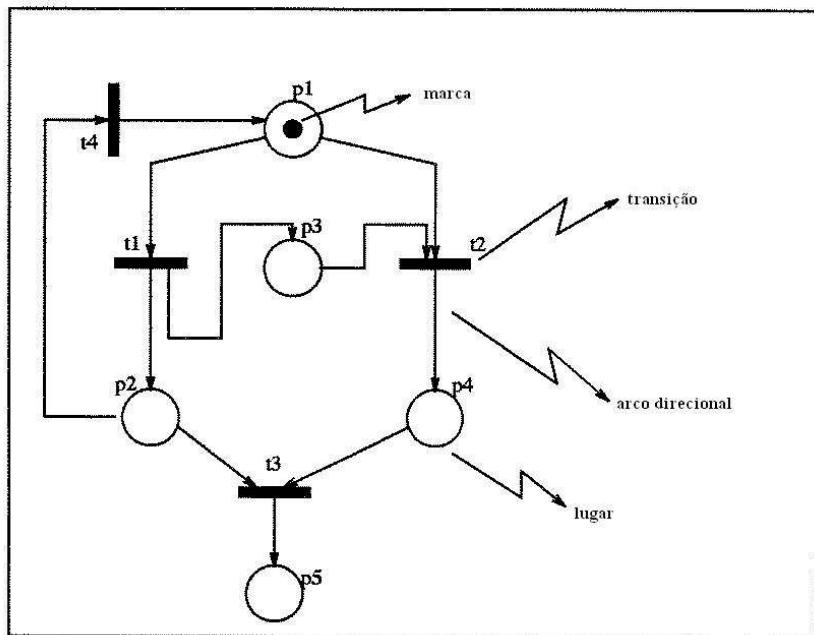
De acordo com (PETERSON, 1977), uma ‘marcação’  $\mu$  é uma associação de ‘marcas’ (ou ‘fichas’) à ‘lugares’ da RdP. Pode-se dizer que estas ‘marcas’ residem dentro dos ‘lugares’ da rede e são representadas graficamente como pequenos círculos pretos.

As ‘marcas’ são usadas para definir a execução de uma RdP, ou seja, demonstrar a dinâmica dos sistemas modelados por estruturas descritas pela RdP.

A ‘marcação’  $\mu$  pode ser definida como um vetor  $\mu = (\mu_1, \mu_2, \dots, \mu_n)$ , onde  $n = |P|$  e cada  $\mu_i \in \mu$ ,  $i = 1, \dots, n$ . Neste vetor, o valor do  $i$ -ésimo elemento indica o número de ‘marcas’ no respectivo ‘lugar’  $p_i$  na rede.

A Figura 2.1 ilustra uma RdP com seus principais elementos.

Pode-se dizer que um ‘estado’ e uma ‘marcação’ são formas diferentes de dizer a mesma coisa, pelo menos em RdP. O ‘disparo’ de uma ‘transição’ representa a mudança do ‘estado’ da rede por uma mudança na ‘marcação’ da rede (PETERSON, 1997).



**Figura 2.6 - Exemplo de grafo de Rede de Petri adaptado de COSTA (1995)**

A mudança de ‘estado’ causada pelo ‘disparo’ de uma ‘transição’  $t$  é definida por uma função  $\delta$ .

#### 2.2.4 Regras para execução da Rede de Petri

Segundo (PETERSON, 1977), a execução da RdP é controlada pelo número de ‘marcas’ na Rede de Petri que é modificado pelo ‘disparo’ das ‘transições’. O ‘disparo’ de uma ‘transição’ implica na remoção de ‘marcas’ de seus ‘lugares de entrada’ e criação de ‘marcas’ que são distribuídas em seus ‘lugares de saída’. Uma ‘transição’ está ‘habilitada’ para ‘disparar’ se seus ‘lugares de entrada’ possuem, pelo menos, a mesma quantidade de ‘marcas’ que o ‘peso’ de seus ‘arcos’. O ‘disparo’ de uma ‘transição’ não pode remover ‘marcas’ inexistentes em um ‘lugar’. Caso não haja ‘marcas’ suficientes em qualquer ‘lugar de entrada’ de uma ‘transição’, não poderá ocorrer o ‘disparo’. Assim, apenas ‘transições’ ‘habilitadas’ é que podem ser objeto de ‘disparo’.

Enquanto houver pelo menos uma ‘transição’ ‘habilitada’, as ‘transições’ de uma RdP podem continuar ‘disparando’ indefinidamente. Entretanto, na prática consideram-se ‘marcações’ específicas para verificar se determinados ‘estados’ foram atingidos.

Informações mais detalhadas sobre RdP podem ser encontradas em (MURATA, 1989) e (PETERSON, 1977).

### 2.2.5 Tipos de Redes de Petri

Com a necessidade de representação de modelos mais complexos, e com diferentes características, foram desenvolvidas extensões da RdP de modo a suprir suas necessidades, como a inclusão de novos atributos, relações de tempo e eventos estocásticos, porém mantendo-se a simplicidade conceitual. Apresenta-se a seguir alguns tipos de RdP com suas características principais (MIYAGI, 1996; JUNQUEIRA, 2006):

- RdP ordinária – os ‘lugares’ possuem ‘capacidade’ ilimitada para armazenar as ‘marcas’;
- RdP Condição-Evento – os ‘lugares’ possuem ‘marcação’ binária, ou seja, cada ‘lugar’ só pode possuir no máxima uma ‘marca’;
- RdP Lugar-Transição – nesse tipo de rede os ‘lugares’ podem possuir mais de uma ‘marca’, porém sua ‘capacidade’ é finita e cada ‘arco’ possui seu próprio ‘peso’, um valor inteiro que indica quantas ‘marcas’ são retiradas ou inseridas com o ‘disparo’ da ‘transição’;
- RdP temporizada – as ‘transições’ possuem um parâmetro que representa o tempo que a ‘marca’ leva para sair da pré-condição até chegar na pós-condição;
- RdP numérica – nesse tipo de rede associa-se variáveis numéricas às ‘transições’, ‘lugares’ e ‘marcas’ e condições para o ‘disparo’ das ‘transições’;
- RdP colorida (CPN) – possui ‘marcas’ individualizadas, o que permite que as ‘transições’ ocorram de forma diferenciada, dependendo do tipo de ‘marca’ considerada;
- RdP orientada a objetos – é uma RdP que inclui recursos para descrever propriedades e entidades que ocorrem na abordagem orientada a objetos.

## 2.2.6 Linguagem PNML

A *Petri Net Markup Language* (PNML) (WEBER, 2006) é uma linguagem de intercâmbio para RdP cujo formato é baseado em *XML* (W3C, 2009).

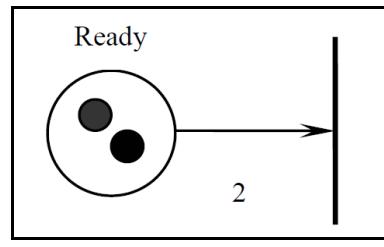
A PNML é capaz de distinguir características gerais de todas as RdP ('lugares', 'arcos', 'transições') e as características específicas de cada tipo de RdP ('marcações', tempo, tipo dos 'lugares').

Os principais elementos constituintes da notação PNML estão relacionados na Tabela 2.1.

**Tabela 2-1 - Elementos PNML (FREITAS; SILVA, 2006)**

<b>Tipo</b>	<b>Tag</b>	<b>Attributo XML</b>	<b>Descrição</b>
Documento PNML	<pnml>	--	--
Rede de Petri	<net>	id: ID	Identificador
Lugar	<place>	id: ID	Identificador
Transição	<transition>	id: ID	Identificador
Arco	<arc>	id: ID	Identificador
		Source: IDRef	Referência para elemento da RdP
		Target: IDRef	Referência para elemento da RdP
Gráfico	<graphics>	--	Posição do elemento na tela
Nome	<name>	--	Nome do elemento na RdP
Ferramenta	<toolspecific>	tool: string	Ferramenta utilizada para leitura do modelo em PNML
		version: string	
Posição	<position>	x: X	Coordenada absoluta no eixo X
		y: Y	Coordenada absoluta no eixo Y
Offset	<offset>	x: X	Coordenada relativa no eixo X
		y: Y	Coordenada relativa no eixo Y
Texto	<text>	--	Comentário, número, etc
Inscrição	<inscription>	--	Inscrição (peso) de um arco
Marcação inicial	<initialMarking>	--	Marcação inicial de um lugar

A Figura 2.7 apresenta um modelo em RdP e sua descrição em PNML.



(a)

```

<pnml xmlns="http://www.example.org/pnml">
  <net id="n1" type="" href="http://www.example.org/pnml/PTNet">
    <name>
      <text>A Petri Net example</text>
    </name>
    <place id="p1">
      <graphics>
        <position x="20" y="20"/>
      </graphics>
      <name>
        <text>Ready</text>
      </name>
      <graphics>
        <offset x="-10" y="-8"/>
      </graphics>
    </place>
    <initialMarking>
      <text>2</text>
    </initialMarking>
    <transition id="t1">
      <graphics>
        <position x="20" y="20"/>
      </graphics>
      <toolspecific tool="PN4all" version="1.0">
        <arc id="a1" source="p1" target="t1">
          <graphics>
            <position x="30" y="5"/>
            <position x="60" y="5"/>
          </graphics>
          <inscription>
            <text>2</text>
          </inscription>
        </arc>
      </toolspecific>
    </transition>
  </net>
</pnml>

```

(b)

**Figura 2.7 - (a) Exemplo de um modelo em RdP (b) Descrição em PNML (FREITAS; SILVA, 2006)**

### 2.3 Simulação Distribuída

Com a dispersão geográfica das instalações físicas de indústrias de manufatura e a dificuldade em analisar sistemas devido à crescente complexidade, torna-se fundamental o desenvolvimento de soluções para abstrair as principais características do sistema e analisar o seu comportamento.

Entre elas, a simulação distribuída surgiu como uma forma de reduzir o tempo de seu processamento, permitindo a representação e o tratamento de problemas mais complexos. Fujimoto (1999) *apud* (JUNQUEIRA, 2006) apresenta várias vantagens de se utilizar simulação entre múltiplos computadores:

- Redução do tempo de execução – por meio da distribuição de um modelo de simulação entre várias máquinas;

- Distribuição geográfica – com as máquinas distribuídas geograficamente e com programas de simulação em execução, há a possibilidade de criar mundos virtuais com muitos participantes que podem estar em diferentes localidades geográficas;
- Integração de simuladores combinando simulações que são executadas em máquinas de diferentes fabricantes, explorando melhor assim os recursos computacionais existentes;
- Tolerância a falhas – a simulação poderá ser prosseguida entre os demais computadores, caso um processador falhe.

### 2.3.1 Algoritmos de sincronização

O algoritmo de sincronização garante que as relações de causalidade do sistema real sejam corretamente reproduzidas na simulação de modo que as relações do sistema real sejam corretamente simuladas, sem que causas e efeitos percam suas relações devido a erros na sequência de simulação. Com o intuito de resolver os problemas de causalidade, em (JUNQUEIRA, 2006) são citadas duas abordagens de sincronização: a conservadora e a otimista.

Essas duas abordagens assumem que a simulação consiste de um conjunto de processos lógicos (PLs) que se comunicam por meio de troca de mensagens ou eventos. O objetivo do mecanismo de sincronização é garantir que os PLs processem os eventos na ordem cronológica, de modo que a simulação em paralelo reproduza os mesmos resultados de uma simulação sequencial.

Na abordagem conservadora (FUJIMOTO, 1999; JUNQUEIRA, 2006), um PL não pode processar um evento até que se tenha a garantia que este é seguro. Nesse caso, as mensagens que chegam a um PL são armazenadas em uma fila ordenada pela sua ordem cronológica conforme o seu *timestamp* (atributo da mensagem que indica o instante em que esta foi gerada no PL que a enviou). Mensagens nulas são enviadas de forma a evitar travamentos, e eventos considerados inseguros são bloqueados de modo a evitar erros de causalidade.

Na abordagem otimista, é permitida a violação da relação de causalidade localmente, mas estão sujeitas a detecção e processo de

recuperação neste caso. Além disso, os modelos otimistas apresentam maior consumo do computador, acarretando certa degradação do desempenho destes.

Assim, neste trabalho a abordagem conservadora será considerada por garantir que a sequência de execução seja respeitada em todas as etapas das simulações.

### 2.3.2 Protocolo Token Ring

Proposto em 1969 e desenvolvido pela IBM, o protocolo *Token Ring* é um tipo de protocolo cuja técnica é baseada na utilização de um *token*, operando em uma topologia de anel (JUNQUEIRA, 2006). Esta técnica consiste em transportar de uma estação a outra, no anel, uma única série de *bits*.

Todas as estações podem receber o *token*, porém somente uma estação pode transmitir, garantindo, portanto, que os dados transmitidos não colidam, fato que ocorre quando duas ou mais estações iniciam uma transmissão simultaneamente sem nenhuma delas terem detectado que o canal, que são transmitidos os dados, esteja ocupado.

Quando uma estação detecta a passagem de um *token*, se a prioridade do mesmo for menor ou igual à prioridade do *frame* (conjunto de informações de tamanho pré-definido) que deseja enviar, ela captura o *token*, transforma-o no início do *frame* a ser transmitido.

A Figura 2.8 ilustra o funcionamento do mecanismo *token ring* onde a estação A deseja transmitir os dados para a estação C:

A estação A captura o *token* e o transforma no início de um *frame* e o transmite para o anel com os dados, os endereços de origem e destino e outras informações. Os dados são então recebidos pela estação B, que lê o endereço de destino e, como os dados não são para esta estação, devolve-os para o anel sem modificação. A estação C recebe os dados, lê o endereço de destino e, como os dados são para ela, esta copia os dados para o seu *buffer* interno, devolve os dados para o anel e marca o *frame* como “copiado”. Os dados passam pela estação D da mesma forma que pela B. Quando os dados são retornados à estação A, ela lê os endereços de origem e destino, espera o

*frame* voltar, de forma a verificar se os dados foram recebidos com sucesso pela estação de destino, retirando-os do anel em seguida.

Durante a passagem do *token*, as demais estações podem fazer a reserva do *token* atribuindo ao seu campo de reserva um nível de prioridade. A maior prioridade tem preferência por capturar o *token*.

Após terminar a transmissão dos dados ou esgotado o tempo limite de transmissão concedido a cada estação, esta libera o *token* alterando a prioridade deste para o nível de prioridade presente no seu campo de reserva.

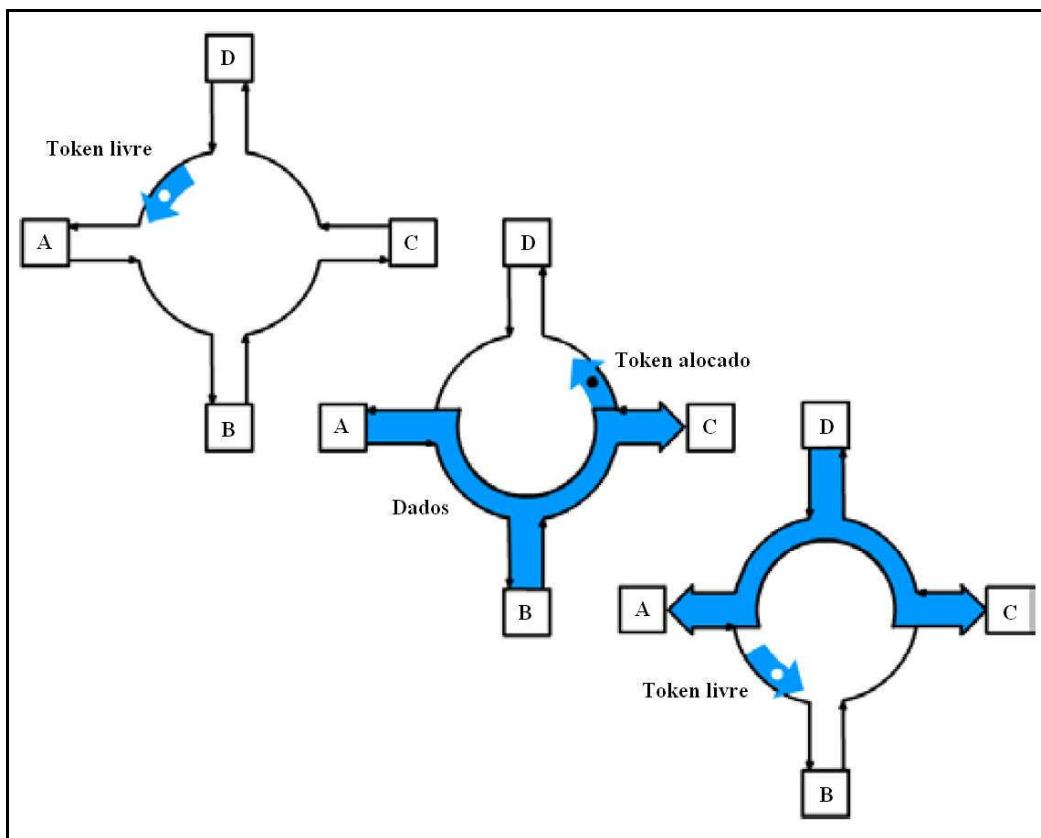


Figura 2.8 - Exemplo de funcionamento de uma rede Token Ring (JUNQUEIRA, 2006)

### 2.3.3 Interface entre modelos

Para que os modelos em RdP se comuniquem de tal forma que a simulação possa se desenvolver, deve-se implementar interfaces entre modelos. Assim, de acordo com (JUNQUEIRA, 2006), pode-se definir:

- Modelo requisitante: modelo em RdP que requisita um segundo modelo por meio de uma interface ('lugar', 'transição' ou 'arco habilitador');

- Modelo requisitado: modelo em RdP que, por meio de uma interface (lugar, transição ou arco habilitador), disponibiliza uma funcionalidade para que possa ser utilizada por outros modelos;
- Interface requisitante: interface utilizada para chamar do modelo requisitante uma funcionalidade disponível no modelo requisitado ou interface utilizada para retornar uma resposta do modelo requisitado para o modelo requisitante;
- Interface requisitada: interface utilizada para receber chamadas requisitando uma funcionalidade no modelo requisitado ou a interface utilizada, no modelo requisitante, para receber a resposta enviada pelo modelo requisitado.

Para o relacionamento entre modelos, existem diversas formas de modelar a interface através de elementos da RdP. SIBERTIN-BLANC (1993) *apud* (JUNQUEIRA, 2006) utilizou-se da relação entre um só tipo de elemento, o que simplifica a modelagem da interface. Assim, pode-se citar três conceitos para a modelagem da interface:

- Fusão de ‘lugares’: na interface entre modelos através de fusão de ‘lugares’, dois ‘lugares’ pertencentes a modelos distintos se comportam como se fossem um mesmo ‘lugar’. Uma desvantagem desse tipo de interface é a dúbia interpretação em caso de dois modelos distintos requisitarem simultaneamente a mesma funcionalidade de um terceiro modelo.
- Fusão de ‘transições’: na interface entre modelos através de fusão ‘transições’, duas ‘transições’ pertencentes a modelos distintos em RdP comportam-se como se fossem a mesma ‘transição’. Assim, para que uma ‘transição’ dispare, é necessário que as pré-condições e as pós-condições estejam satisfeitas para as ‘transições’ pertencentes a ambos modelos. Esta abordagem evita-se dúvidas quanto ao modelo que efetivamente requisitou primeiro o modelo requisitado, uma vez que a ‘marcação’ somente poderá evoluir quando as ‘transições’ de interface dos modelos requisitante e requisitado estão disponíveis.

- Comunicação por ‘arcos habilitadores’: nesse tipo de comunicação, a conexão de ‘arcos habilitadores’ entre os modelos permite que se controlem os disparos de ‘transições’. A utilização desta abordagem não se mostra eficaz, uma vez que pode ocorrer de tanto a ‘transição’ requisitante quanto a ‘transição’ requisitada ser disparada primeiro. Como alternativa, pode-se utilizar um outro ‘arco habilitador’ ou um novo elemento responsável pelo gerenciamento da comunicação entre modelos. No entanto, tais soluções se mostram complexas.

A partir das características das três abordagens citadas acima, a fusão de ‘transições’ está sendo considerada no presente trabalho, uma vez que nesta abordagem não há dubialidade como apresentada na fusão de ‘lugares’ e por incluir menos elementos na RdP em relação à comunicação por ‘arcos habilitadores’.

#### **2.4 Visual Studio**

Para a implementação do sistema computacional foi utilizada a linguagem C#, uma linguagem de programação orientada a objetos produzida pela Microsoft como parte da plataforma .NET e que permite integrar com outros sistemas físicos.

O desenvolvimento do *software* se deu por meio do uso de um ambiente de desenvolvimento integrado (IDE) totalmente gráfico, o Visual Studio 2010, que permite desenvolver diversas aplicações, facilitando principalmente a construção de interfaces.

### 3 Projeto do ambiente de modelagem e simulação

O projeto do ambiente de modelagem e simulação inclui em especificar os requisitos do *software* (funcionais e não-funcionais), bem como o uso dos principais diagramas UML (casos de uso, classes e sequência) facilitando, portanto, a implementação do sistema computacional.

#### 3.1 Especificação dos requisitos

De acordo com (SOMMERVILLE, 2006), os requisitos de um sistema são descrições de serviços fornecidos pelo sistema e suas restrições operacionais. Esses requisitos podem refletir nas necessidades dos usuários por um sistema que ajude a resolver alguns problemas como, por exemplo, controlar dispositivo, solicitar um pedido ou encontrar informações.

Os requisitos de um sistema são classificados em requisitos funcionais e não-funcionais.

Os requisitos funcionais são instruções de serviços que o sistema deve fornecer, ou seja, como o sistema deve responder a uma dada informação inserida e como o sistema deve se comportar em situações particulares. Em alguns casos, os requisitos funcionais podem também expressar explicitamente o que o sistema não deve fazer.

Os requisitos não funcionais são restrições nos serviços ou funções oferecidas pelo sistema. Incluem as restrições de tempo, restrições no desenvolvimento do processo e padrões. Requisitos não funcionais geralmente são aplicados a sistemas como um todo.

Os requisitos não funcionais podem ser classificados em tipos conforme a Figura 3.1.

Assim, com base na teoria, são especificados os requisitos do *software* de simulação:

##### 1. Requisitos Funcionais:

- Criar, editar e simular uma dada Rede de Petri escrita em linguagem PNML (script PNML).
- Comunicar-se com outros computadores através de rede Local ou Internet.

- Arquivar scripts de Rede de Petri, bem como recuperar este arquivo.

## 2. Requisitos de Usabilidade:

- Conhecimento prévio por parte do usuário da linguagem PNML, suas classes e parâmetros obrigatórios.
- Editor de texto para a criação e edição do *script* de PNML.
- Esquema gráfico representando o *script* de PNML para auxiliar o usuário na criação e edição.

## 3. Requisitos de Confiabilidade:

- Tratamento de problemas com a conexão entre os computadores.
- Capacidade de recuperar o último estado antes do problema de conexão.
- Criação de log caso haja erros durante a execução, bem como a interrupção da execução em caso de erro fatal.

## 4. Requisitos de Desempenho:

- Capacidade de comunicar-se com outro computador em tempo inferior ao da execução do *script* localizado neste segundo computador, para que a simulação distribuída de fato se torne mais eficiente que a simulação local.

## 5. Requisitos de Suportabilidade:

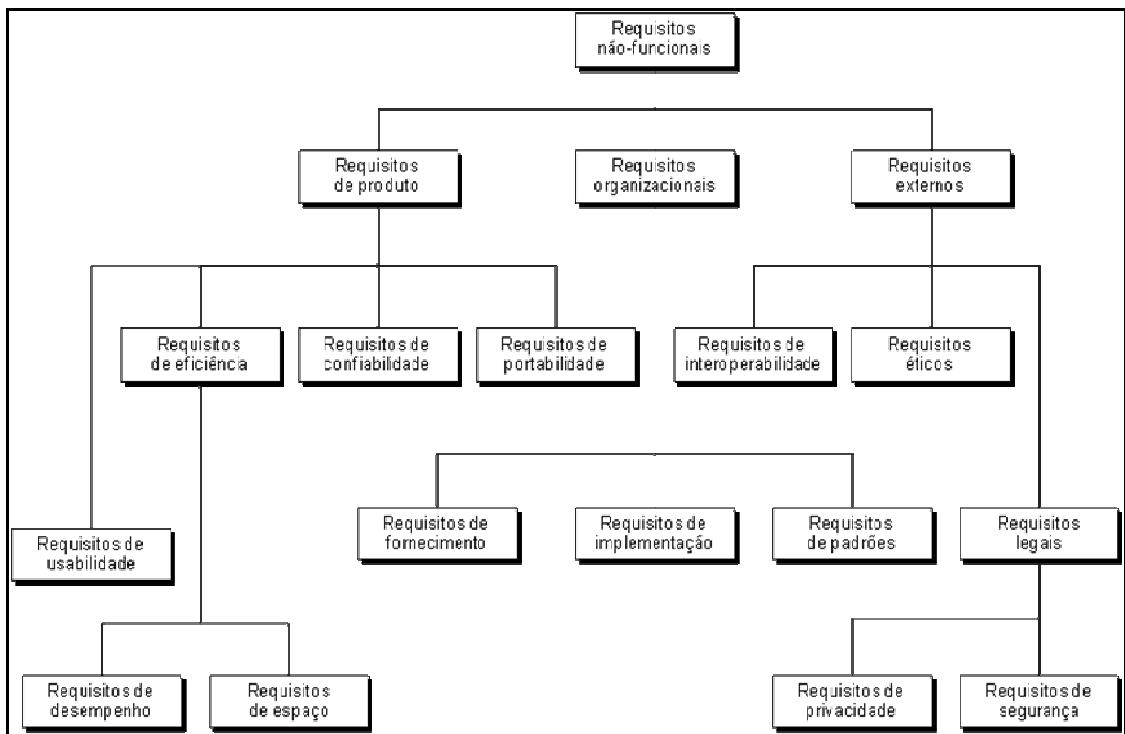
- Teste de conexão, incluindo *ping*<sup>1</sup>, e futuramente, otimização do anel de conexão para maior velocidade de comunicação.

## 6. Requisitos de Implementação:

- Utilização da linguagem C# desenvolvida pela Microsoft para a plataforma .NET.
- Divisão entre as responsabilidades de comunicação, simulação e arquivamento, para futuras adaptações, manutenções e/ou extensões.

---

<sup>1</sup> Aplicativo que utiliza o protocolo ICMP (*Internet Control Message Protocol*) e permite o usuário verificar a conectividade entre dois *hosts*.

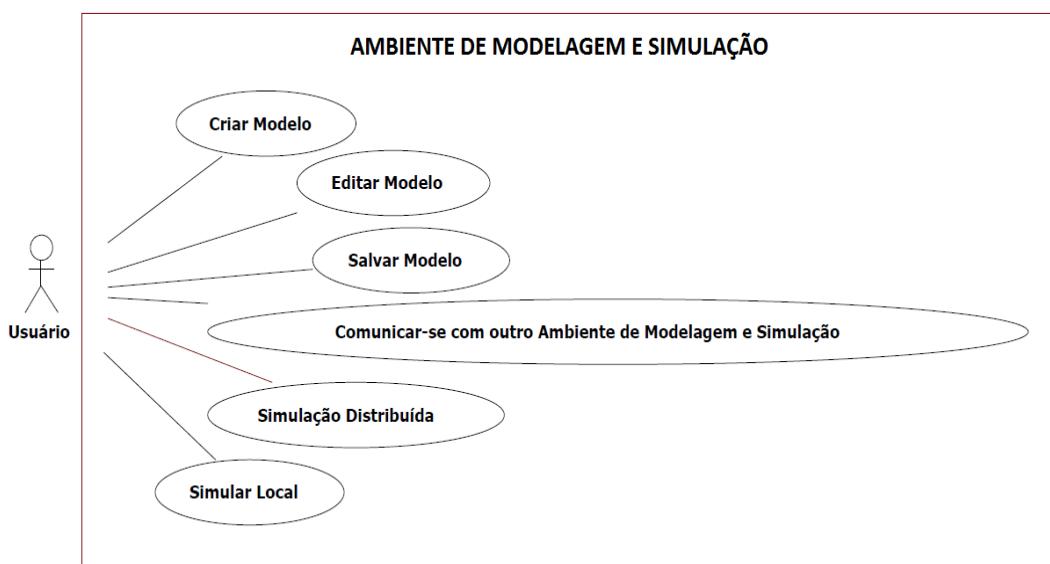


**Figura 3.1 - Classificação dos requisitos não-funcionais adaptado de (SOMMERVILLE, 2006)**

### 3.2 Modelagem UML

A seguir é apresentada a modelagem realizada por meio de diagramas UML: Casos de uso, Classes e Sequência.

#### 3.2.1 Diagrama de Casos de uso



**Figura 3.2 - Casos de uso do Ambiente de Modelagem e Simulação**

### 3.2.2 Diagrama de Classes

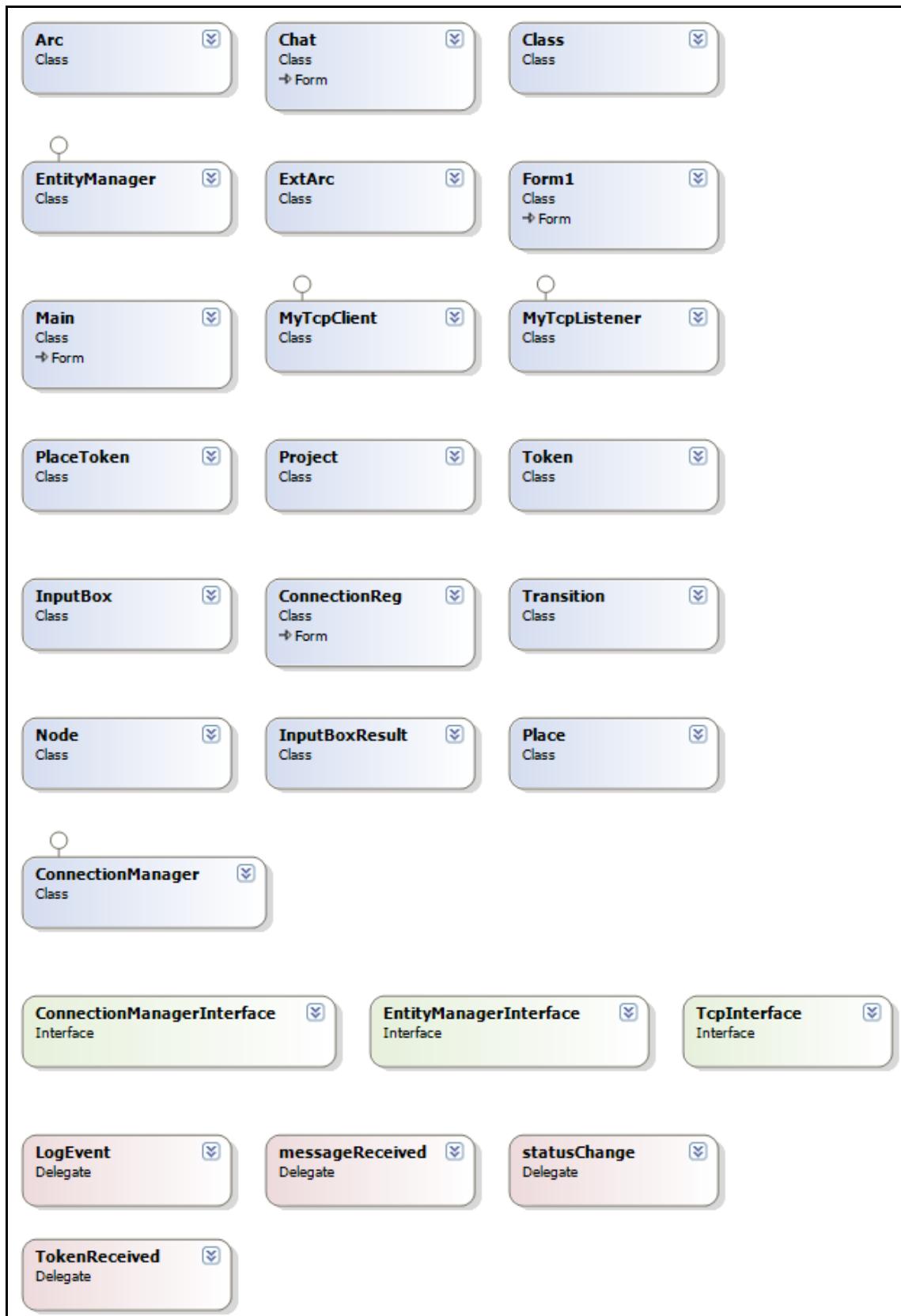


Figura 3.3 - Classes, Interfaces e Delegates

As classes “ConectionManager”, “EntityManager”, “MyTcpClient” e “MyTcpListener”, implementam interfaces. Tal uso torna flexível mudanças no código fonte.

As classes “Arc”, “Place”, “Token” e “Transition” são classes responsáveis pelas entidades de um RdP, ou seja, representam os objetos ‘arco’, ‘lugar’, ‘marca’ e ‘transição’. Outras classes como “ExtArc”, “PlaceToken” e “Node” são classe que servem de auxílio para o relacionamento entre ‘arcos’, ‘lugares’, ‘marcas’ e ‘transições’ de uma RdP, bem como tratar a dinâmica existente entre esses elementos.

A classe “Chat” é uma classe por onde o usuário terá acesso à conexão e testes de envio de dados à estação remota.

Já “Form1” é a interface gráfica, por meio do qual o usuário terá acesso à criação, edição e simulação de modelos.

Já o *delegate* é um tipo de referência que pode ser usado para encapsular um nome ou um método anônimo, ou seja, são variáveis que ao invés de armazenarem algum valor, armazenam o endereço de memória em que o valor se encontra. Isso torna possível a execução de métodos sem que seja necessário saber quais métodos são esses, já que o vínculo dos *delegates* aos métodos pode ser realizado dinamicamente ou em tempo de execução. *Delegates* são a base para eventos.

### 3.2.3 Diagrama de Sequência

A seguir serão apresentados os diagramas de sequência das principais funcionalidades do sistema computacional:

A Figura 3.4 indica o envio de dados pela rede, envolvendo um método de compactação de dados “ZipStr” antes de enviar o dado. Caso tenha sucesso no envio, a variável “status” é alterada para “Sent”, caso contrário, será alterada para “Error”.

A Figura 3.5 indica o recebimento de dados pela rede. Neste caso, há uma espera por alguma mensagem. No momento do recebimento dessa mensagem, ela é descompactada pelo método “UnZip” e é confirmada o recebimento dela.

A Figura 3.6 indica a conexão a um servidor, criando um objeto cliente, por meio da instância da classe “MyTcpClient”. Após a criação do cliente, é adicionada a esse objeto uma funcionalidade, correspondente a criação de *log* (onde são armazenadas as mensagens ou informações).

A Figura 3.7 representa a abertura de projeto, ou seja, a sequência dessa a abertura envolve a localização ou o caminho, e o nome do projeto. Após essa definição, o projeto é montado elemento por elemento na interface gráfica até a sua finalização.

A sequência de salvar projetos está ilustrada na Figura 3.8, onde é obtido o nome e o caminho antes de realmente salvar o projeto.

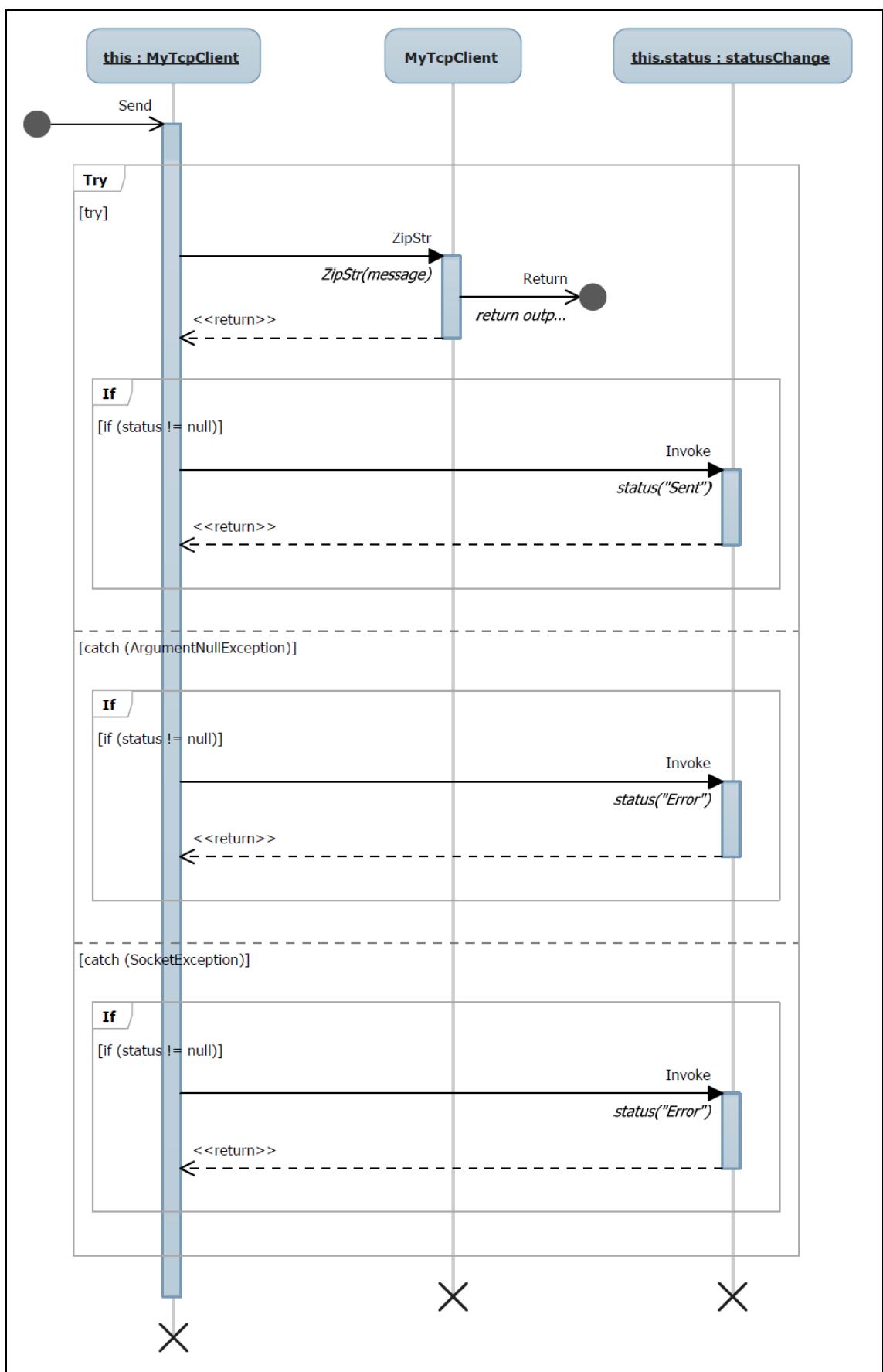


Figura 3.4 – Enviar dados pela rede

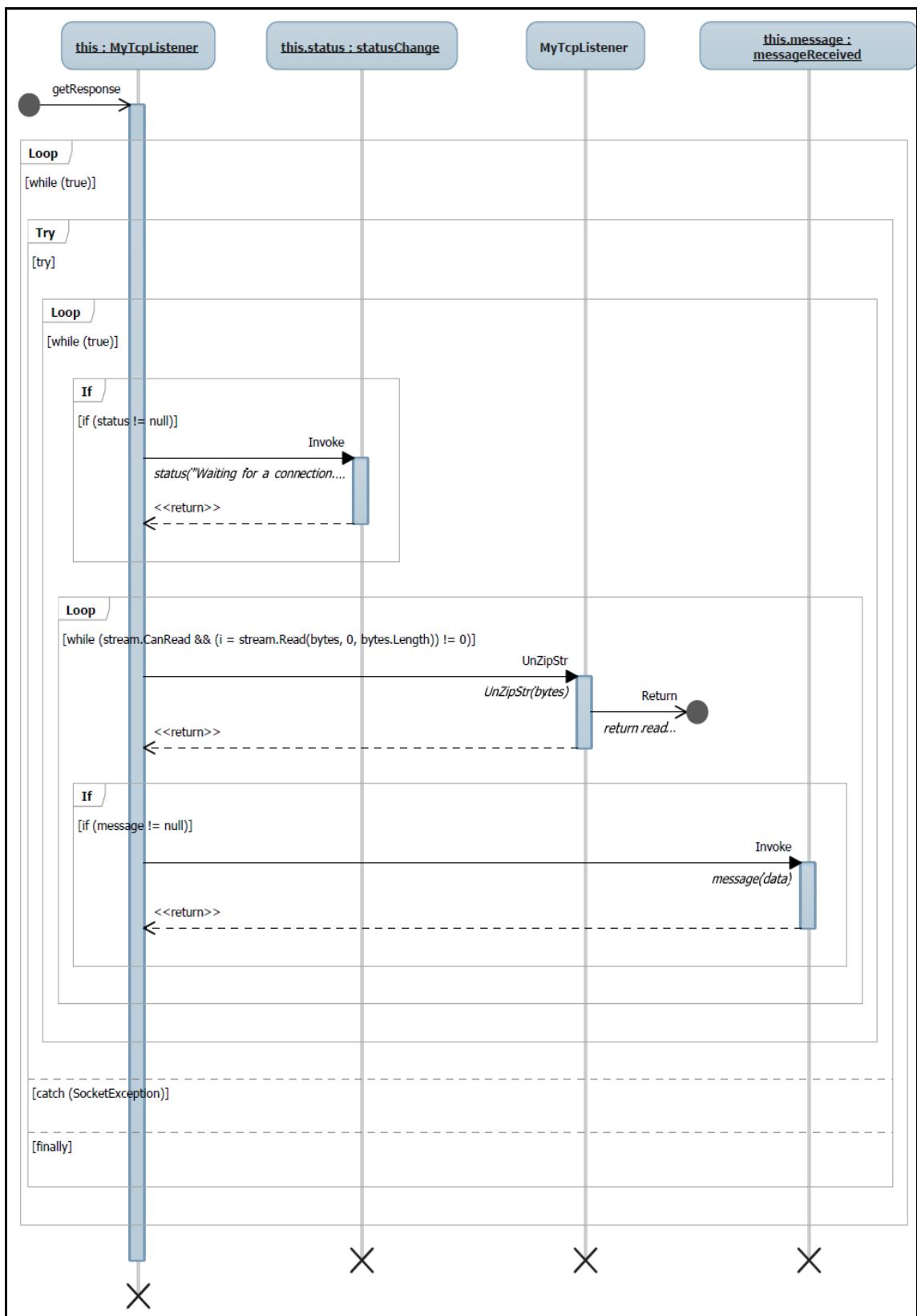


Figura 3.5 - Receber dados em uma rede

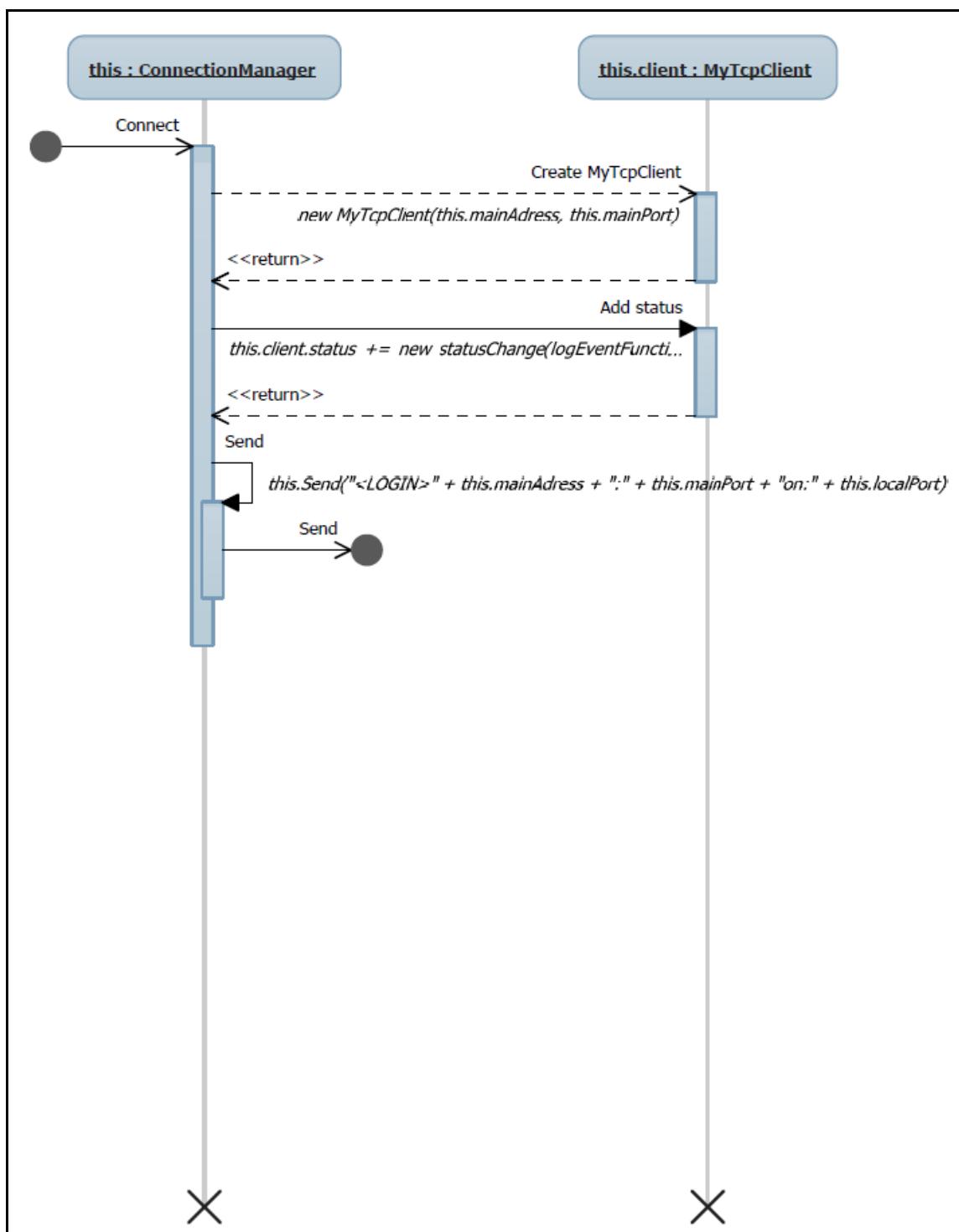


Figura 3.6 - Realizar a conexão a um servidor

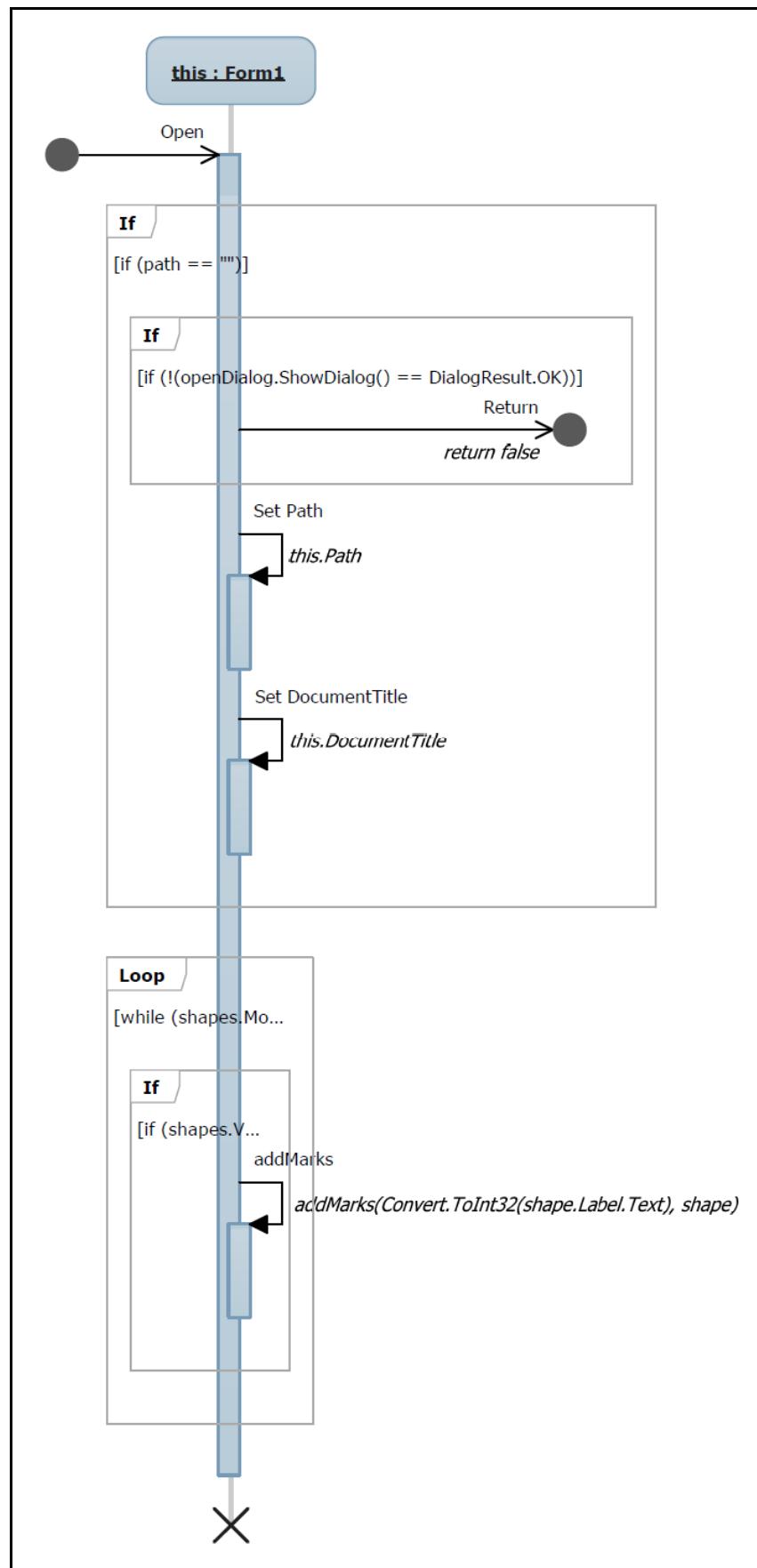


Figura 3.7 - Abrir projeto

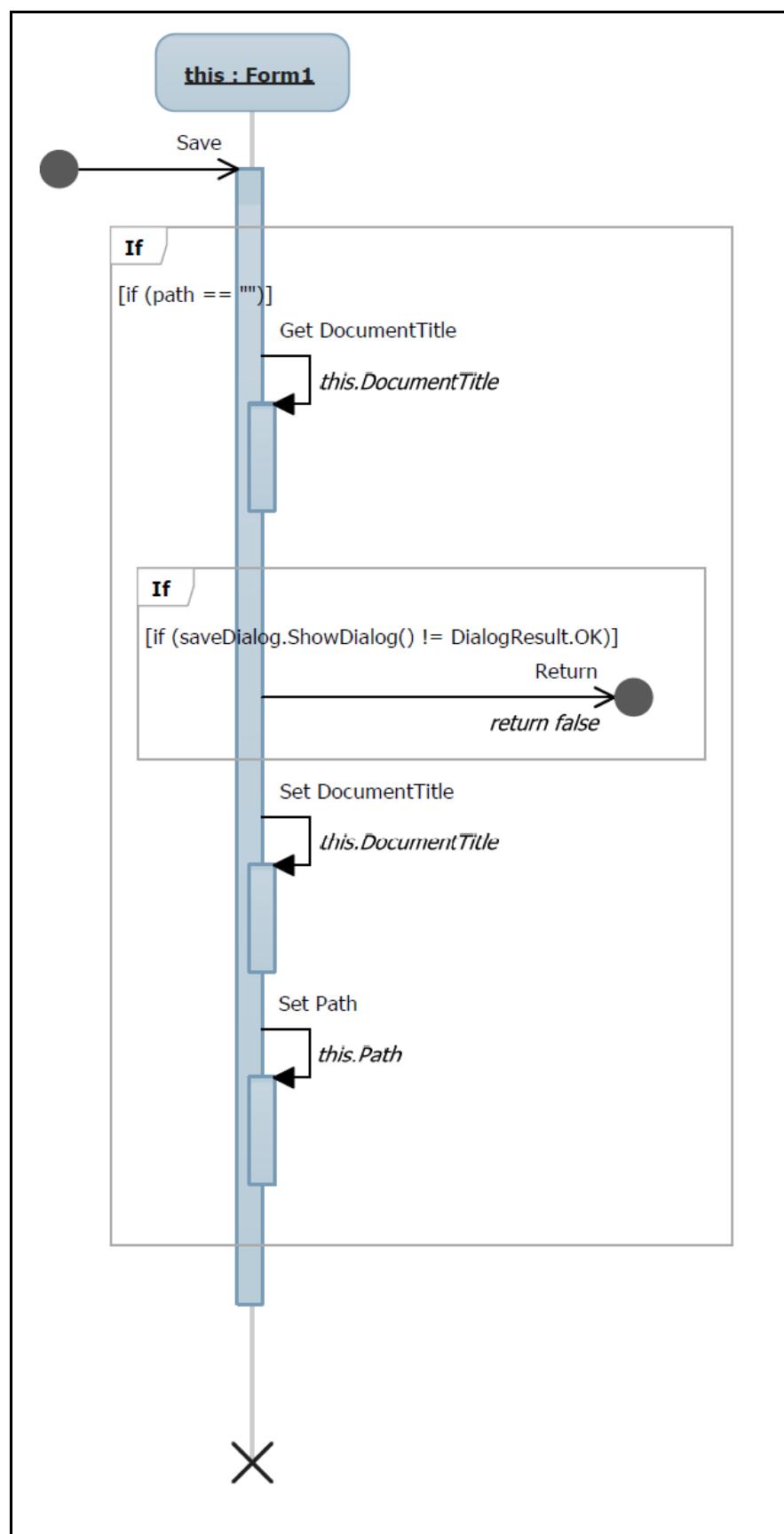


Figura 3.8 - Salvar projeto

## 4 Estrutura do ambiente de modelagem e simulação

Neste item serão abordados os pontos principais do ambiente de modelagem e simulação implementado neste projeto, levando-se em conta os conceitos teóricos apresentados anteriormente.

São também apresentados os componentes utilizados na implementação do software de modelagem e simulação, bem como sua estrutura e os suportes considerados.

### 4.1 *Linguagem para descrição da RdP*

Para descrever, armazenar e compartilhar os modelos em RdP, está sendo adotada neste projeto a linguagem *XML* (W3C, 2009). Todos os elementos que compõem um modelo estão presentes em um mesmo documento, delimitados por *tags*, facilitando o compartilhamento de modelos, uma vez que todos os seus modelos estão contidos no mesmo documento.

Além disso, baseado na linguagem PNML e com a finalidade de facilitar a simulação dos modelos, foi implementado um conversor de arquivo *XML*, capaz de converter em um novo documento *XML*, porém simplificado e com as principais entidades da RdP. Tal conversor foi implementado especificamente para a simulação dos modelos, visto que o arquivo *XML* gerado a partir da modelagem possui marcadores e atributos que podem ser dispensados, facilitando, portanto, o processo da simulação.

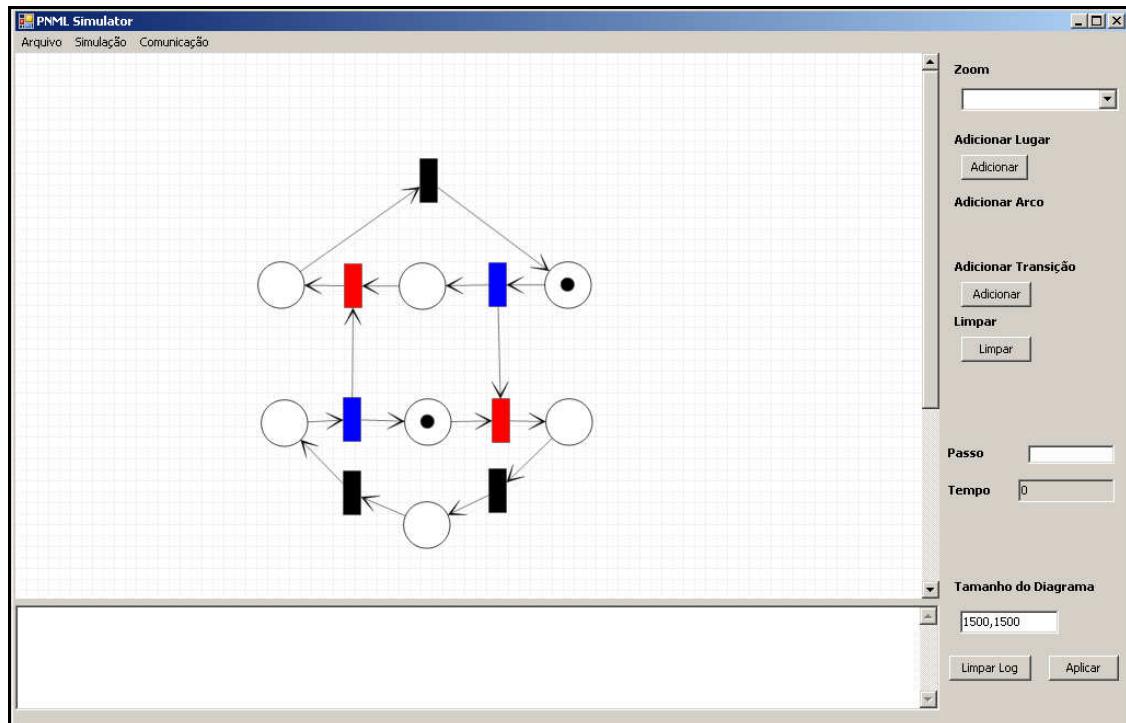
### 4.2 *Interface de usuário*

A interface de usuário possui recursos amigáveis, de modo que o usuário possa criar e editar modelos em RdP, além de realizar a simulação local ou isolada, e distribuída.

Para o desenvolvimento dessa interface foi utilizado o componente Crainiate ERM Diagram 4.1 (CRAINIADE SOFTWARE, 2010), capaz também de construir outros tipos de interfaces gráficas como, por exemplo, diagramas de processos, de banco de dados, organizacionais, fluxogramas, entre outros. Uma vantagem de utilizar este componente para o desenvolvimento do ambiente de modelagem e simulação de modelos em RdP é a possibilidade de

salvar os modelos em formato *XML* (W3C, 2009), que poderá ser convertido para o padrão PNML de forma a facilitar o processo de simulação.

A Figura 4.1 ilustra a interface gráfica desenvolvida com o Crainiate contendo uma RdP de exemplo.



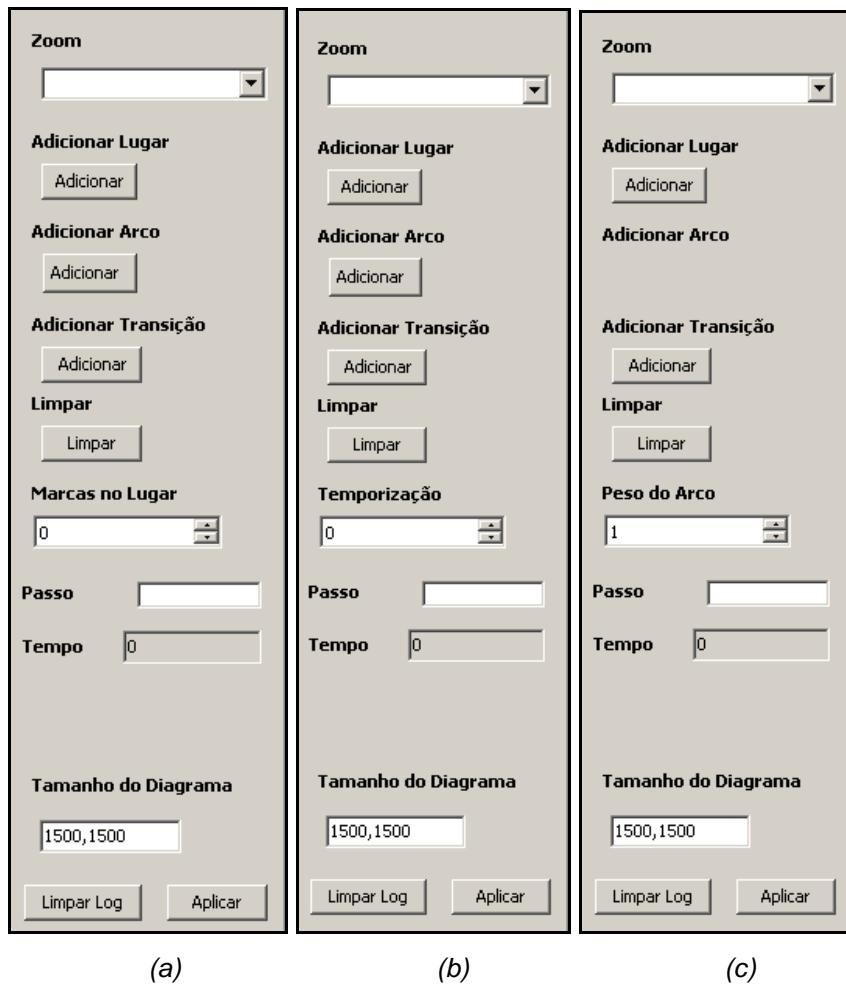
**Figura 4.1 - Interface de usuário com exemplo de RdP**

As funções presentes à direita da interface são (Figura 4.2):

- “Zoom”: Por meio desse comando é possível selecionar o Zoom com os seguintes valores: 25%, 50%, 75%, 100% e 200%;
- “Adicionar Lugar”: Clicando no botão “Adicionar” logo abaixo de “Adicionar Lugar”, um novo ‘lugar’ (sem ‘marcas’) é adicionado no modelo. Para adicionar ‘marcas’ nesse novo ‘lugar’, é necessário selecionar o ‘lugar’, de modo a aparecer uma caixa numérica (“Marcas no Lugar”), por onde é possível inserir a quantidade de ‘marcas’. Para quantidade de ‘marcas’ acima de quatro, a representação é feita por meio de números inscritos no ‘lugar’;
- “Adicionar Transição”: clicando no botão “Adicionar” logo abaixo de “Adicionar Transição”, uma ‘transição’ é adicionada no modelo. Para que a ‘transição’ se torne fundida, é necessário inserir um ‘arco’ entre duas ‘transições’ sendo que a direção desse ‘arco’ deve apontar da ‘transição’

do modelo requisitante para a ‘transição’ do modelo requisitado. Assim, ao inserir um ‘arco’ entre duas ‘transições’, a ‘transição’ do modelo requisitante muda de cor preta para azul, enquanto a ‘transição’ do modelo requisitado muda de cor preta para vermelha, facilitando a identificação do tipo de ‘transição’. É possível também selecionar o tipo de ‘transição’, ou seja, se é temporizada ou instantânea. Para isso, é necessário selecionar a ‘transição’, de modo a aparecer uma caixa numérica correspondente a “Temporização”. O valor zero corresponde à ‘transição’ instantânea, enquanto para valores diferentes de zero corresponde ao tempo necessário para o disparo da ‘transição’;

- “Adicionar Arco”: para adicionar um ‘arco’ é necessário selecionar uma ‘transição’ ou um ‘lugar’ de origem, de modo a habilitar o botão “Adicionar”. Uma vez habilitado esse botão, é necessário clicá-lo uma vez e selecionar a ‘transição’ ou o ‘lugar’ de destino para que seja completada a criação do ‘arco’. Para adicionar o ‘peso’ do ‘arco’, é necessário selecionar o ‘arco’ em questão, de modo a habilitar a caixa numérica “Peso do Arco”, correspondente à adição do ‘peso’ no formato numérico. Quando é criado o ‘arco’ sem a modificação do seu ‘peso’, o ‘arco’ possuirá ‘peso’ 1;
- “Passo”: corresponde ao passo da simulação, ou seja, o representa o estado em que se encontram os modelos;
- “Tempo”: indica o tempo local de simulação;
- “Tamanho do diagrama”: é possível definir o tamanho do diagrama. Como padrão o tamanho é “1500x1500”;
- “Limpar Log”: Limpa todas as mensagens exibidas durante a execução do programa.



**Figura 4.2 - Funções à direita da interface de usuário. Habilitação de novas funções ao selecionar: (a) um ‘lugar’, (b) uma ‘transição’ e (c) um ‘arco’.**

Os menus possuem as seguintes funções:

- Menu “Arquivo” (Figura 4.3):
  - “Novo”: cria um novo projeto;
  - “Abrir”: abre um projeto no formato XML e exibe-o no ambiente;
  - “Salvar”: salva o projeto;
  - “Salvar como...”: salva o projeto, com a possibilidade de alterar o nome ou a localização;
  - “Imprimir”: realiza a impressão do projeto que está sendo exibido no ambiente;
  - “Visualizar Impressão...”: exibe a aparência do projeto impresso;
- Menu “Simulação” (Figura 4.4):

- “Simulação Local”: para realizar a simulação local ou isolada, deve-se primeiro carregar um projeto, ou seja, escolher a função “Preparar” dentro do menu “Simulação Local”, de modo a habilitar a simulação “Passo a Passo” e a simulação em “Diversos Passos”;
- “Simulação Distribuída”: do mesmo modo que a “Simulação Local”, deve-se carregar um projeto, ou seja, escolher a função “Preparar” dentro do menu “Simulação Distribuída”. Nesse menu há a possibilidade de realizar a simulação “Passo a Passo”.
- Menu “Comunicação” (Figura 4.5):
  - “Registrar-se”: uma nova janela é exibida para a conexão entre duas estações (Figura 4.6). Nessa janela deve-se escolher a porta local, a máquina, que corresponde ao IP da máquina remota e a porta correspondente à máquina remota.

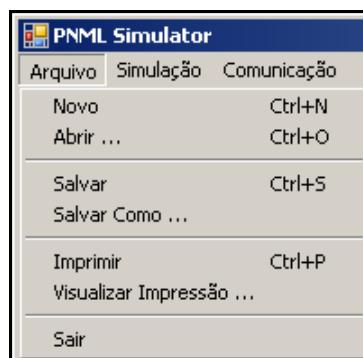


Figura 4.3 - Comandos presentes no menu "Arquivo"

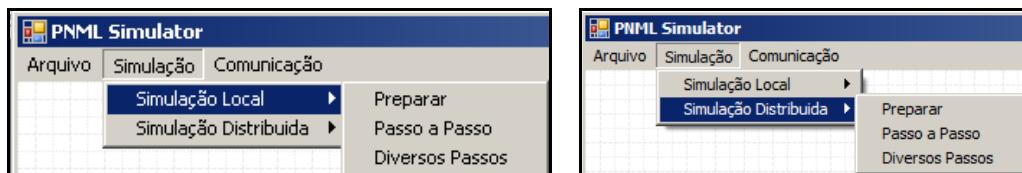


Figura 4.4 - Comandos presentes no menu "Simulação"

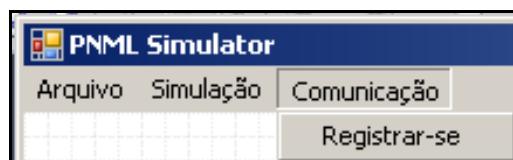
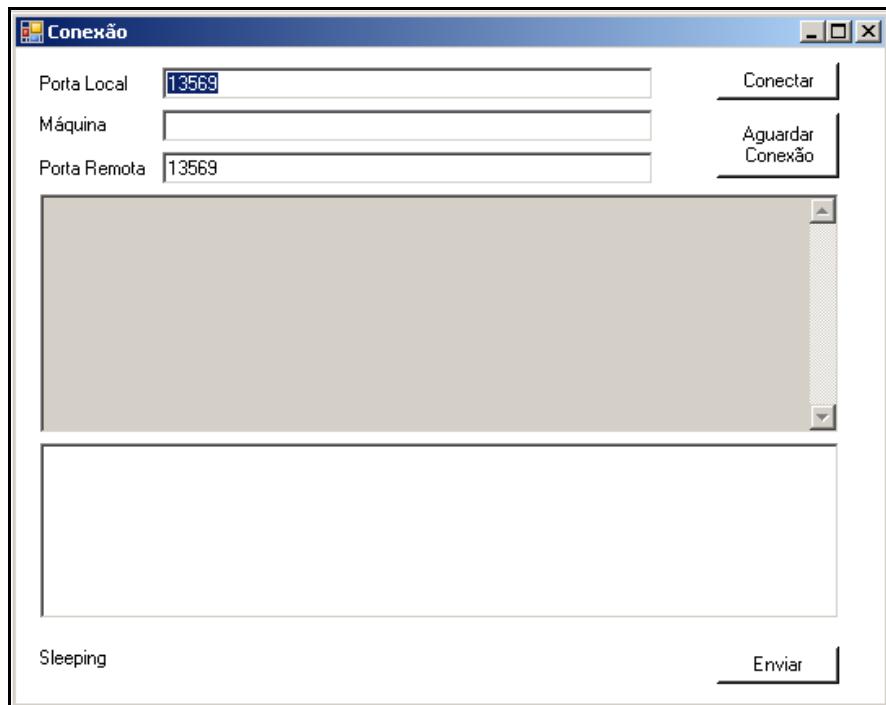


Figura 4.5 - Comandos presentes no menu "Comunicação"



**Figura 4.6 - Janela para conectar-se com outra estação**

### 4.3 Servidor/Cliente

A plataforma .NET fornece amplo suporte e classes desenvolvidas para se construir conexões (servidor/cliente) do tipo TCP/IP.

A classe `TCPClient`, presente nessa plataforma, fornece métodos simples para conectar, enviar e receber dados em uma rede.

Já a classe `TCPListener` fornece métodos simples que escutam e aceitam solicitações de conexão.

Tais classes têm como finalidade criar um cliente e um servidor, isto é, uma instância que utiliza os serviços, e outra instância que fornece os serviços, respectivamente. Assim, é possível implementar o protocolo *Token Ring*, criando-se um anel de modo que cada computador oferece o serviço de escutar e transmitir o *token*.

### 4.4 Mecanismo para o gerenciamento da simulação distribuída

Conforme citado no item 3.1.3, cada estação possui um Cliente e um Servidor para a implementação do protocolo *Token Ring*. Tal estrutura permite que cada estação conheça a identidade da próxima estação, assegurando um sentido único de movimentação das mensagens. Concomitantemente,

mensagens do tipo ‘chamadas de métodos’ podem ser enviadas pela rede de comunicação.

‘Chamada de método’ (JUNQUEIRA, 2006) é um tipo de mensagem relacionada à troca de informações com modelos em outras estações através de fusão de ‘transições’, ou seja, é uma consulta a cerca da situação da ‘transição’ (associada a alguma outra ‘transição’ pela estrutura de fusão entre elas) de outra estação.

O gerenciamento da simulação é distribuído entre as estações. E, de acordo com a abordagem conservadora, o *token* sincroniza o tempo de simulação dos modelos em todas as estações. Os dados que estão contidos em um *token*, e que podem ser modificados por qualquer uma das estações pertencentes ao anel, são (JUNQUEIRA, 2006):

- ‘identificação da estação’ – este campo indica a última estação a alterar os dados do *token*;
- ‘tempo futuro’ – tempo de simulação da RdP requisitado pela estação indicada na ‘Identificação da estação’;
- ‘status’ – contém o status atual da estação indicada na ‘Identificação da estação’. A Tabela 4-1 indica os possíveis valores de status;
- ‘instrução’ – contém a instrução que será enviada a outras estações como, por exemplo, iniciar ou parar a simulação;
- ‘erro’ – gerencia erros de simulação como, por exemplo, desconexão por problemas de comunicação.

**Tabela 4-1 - Possíveis valores para ‘status’ (JUNQUEIRA, 2006)**

<b>Valores de ‘status’</b>	<b>Significado</b>
0	Nenhuma estação está usando o <i>token</i>
1	A estação está verificando o ‘status’ corrente de outras estações
2	A estação está enviando uma ordem para que as demais estações atualizem seus tempos de simulação com base no ‘tempo futuro’.
3	A estação entrou em <i>deadlock</i> <sup>2</sup>

---

<sup>2</sup> Fenômeno que ocorre quando o modelo não possui mais ‘transições’ disparáveis.

Para que ocorra a sincronia da simulação é necessário verificar e avaliar os parâmetros do ambiente de simulação. Um desses parâmetros é o *deadlock*, que pode assumir valores de acordo com o estado do modelo com relação a suas ‘transições’. A Tabela 4-2 indica os valores que o parâmetro *deadlock* pode assumir.

**Tabela 4-2 - Valores possíveis para o parâmetro deadlock**

Valor	Significado
0	O modelo local não está em <i>deadlock</i>
1	O modelo local não possui ‘transições’ instantâneas disparáveis
2	O modelo local não possui ‘transições’ disparáveis (instantâneas ou temporizadas)

Para o tratamento do *token*, JUNQUEIRA (2006) definiu nove regras para a atualização ou conservação do mesmo. As regras são:

1. o modelo local está em *deadlock* pois não possui mais ‘transições’ ‘disparáveis’ em sua lista de ‘disparos’. Neste caso, ele deve avisar os demais modelos (nas demais estações) que ele está em *deadlock*;
2. o modelo local está em *deadlock* pois não possui mais ‘transições’ instantâneas ‘disparáveis’. Neste caso, ele deve consultar o ‘status’ dos demais modelos a fim de evoluir o tempo de simulação do sistema como um todo;
3. o modelo local não está em *deadlock* e recebe o *token* onde um outro modelo está tentando fazer uma consulta ou avisar que está em *deadlock*. Neste caso, ele deve reiniciar os campos do *token* e informar qual o seu tempo local de simulação, pois este é o menor tempo entre os modelos;
4. o modelo local está em *deadlock* (não possui ‘transições’ instantâneas ‘disparáveis’) e recebe o *token* onde um outro modelo, com tempo de simulação superior ao local, está tentando fazer uma consulta. Neste caso, ele deve sobreescriver a ‘identificação da estação’ e o ‘tempo futuro’ com seus próprios valores, passando este a fazer a consulta;

5. o modelo local está em *deadlock* (não possui ‘transições’ instantâneas ‘disparáveis’) e recebe o *token* onde um outro modelo informa que está sem ‘transições’ (instantâneas e temporizadas) ‘disparáveis’. Neste caso, ele deve sobrescrever a ‘identificação da estação’ e o ‘tempo futuro’ com seus próprios valores, e trocar o ‘status’ do *token* para consulta, informando que ao menos este modelo não está sem ‘transições’ em sua lista de ‘disparos’;
6. o modelo local recebe uma instrução para atualizar o seu tempo local de simulação. Neste caso, ele deve atualizar o seu relógio local com o tempo informado no *token* e alterar seu parâmetro *deadlock* para zero, a fim de ‘disparar’ suas ‘transições’, caso exista alguma ‘disparável’ neste novo tempo local;
7. o modelo local recebe de volta o *token* que enviou, sem alterações, onde informava aos demais modelos que este se encontrava sem mais ‘transições’ (instantâneas e temporizadas) ‘disparáveis’. Neste caso, ele deve atualizar o ‘erro’ do *token* informando que o sistema como um todo está travado e a simulação deve ser encerrada;
8. o modelo local recebe de volta o *token* que enviou, sem alterações, onde fazia uma consulta aos demais modelos de forma a tentar atualizar o seu tempo local de simulação e, consequentemente, o do sistema. Neste caso, ele altera o ‘status’ do *token* de forma a informar às demais estações para atualizarem seus relógios locais de simulação com o valor utilizado na consulta;
9. o modelo local recebe de volta o *token* que enviou. Os demais modelos já atualizaram seus relógios locais, restando apenas ele. Assim, ele libera o *token* e troca seu ‘status’ de *deadlock* para zero, ou seja, o modelo não está mais em *deadlock*, podendo ‘disparar’ suas ‘transições’.

## 5 Testes de simulação

Neste item são apresentados os testes de simulação local e distribuída.

### 5.1 Simulação Local ou Isolada

Este teste inclui simular um modelo localmente, sem a necessidade de comunicar-se com outras estações. Para isso foi criado um modelo de exemplo (Figura 5.1), que inclui todos os elementos de uma RdP, inclusive ‘transições’ fundidas e temporizadas.

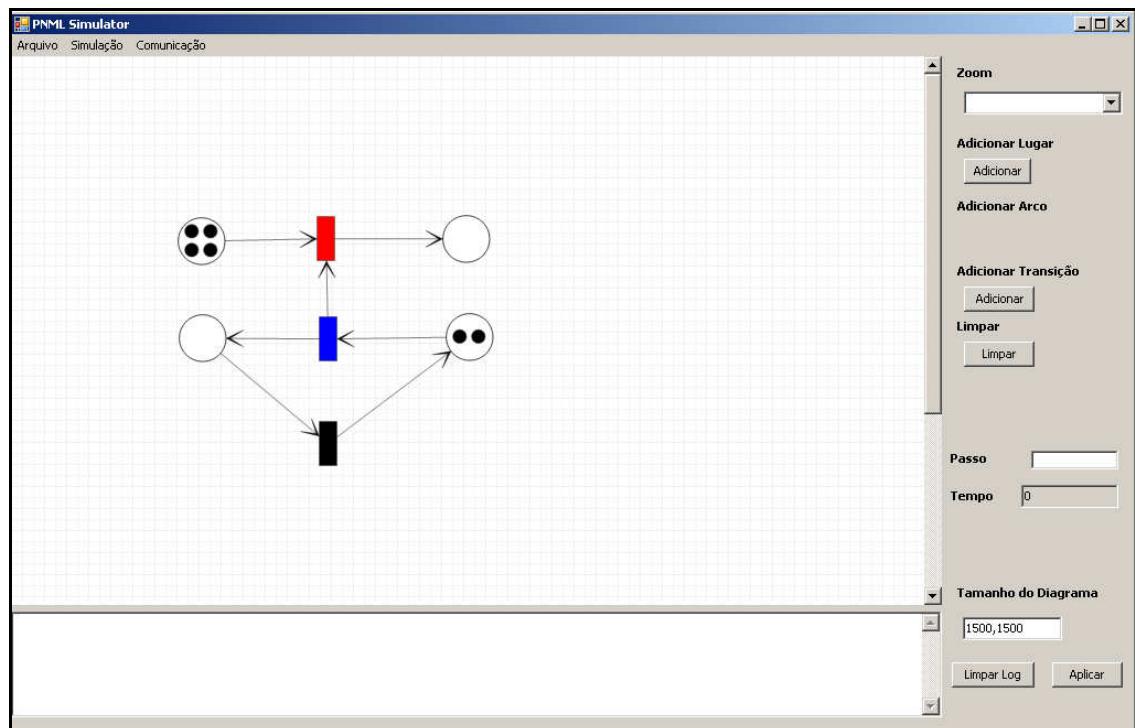


Figura 5.1 - Modelo a ser simulado

Neste exemplo, serão também mostradas as mensagens exibidas durante a simulação.

Para o início da simulação local, é necessário preparar o modelo (menu “Simulação”, “Simulação Local”, item “Preparar”). Com isso, o modelo a ser simulado é dividido em duas Redes (“Rede 1” é o modelo que possui a ‘transição’ requisitada e “Rede 2” é o modelo que possui ‘transição’ requisitante e a temporizada), novas mensagens são exibidas (Figura 5.2) e o modelo está

pronto para realizar a simulação, sendo que cada elemento do modelo é identificado por “transition”, “Line” ou “place” seguido de um número.

Mensagens com valores de *deadlock* podem ser classificadas de acordo com a Tabela 4-2, onde “Deadlock 1” ocorre quando o modelo local não possui ‘transições’ instantâneas ‘disparáveis’ e “Deadlock 2” é quando o modelo local não possui mais ‘transições’ a serem disparadas.

```

Place place1 carregado com 0 tokens.
Place place0 carregado com 2 tokens.
Place place2 carregado com 0 tokens.
Transição instantânea transition6 carregada.
Transição instantânea transition5 carregada.
Place place3 carregado com 4 tokens.
Transição transition4 temporizada em 3 carregada.
Arco Line1 de place3 para transition6 com peso de 2 carregado.
Arco Line2 de transition6 para place1 com peso de 1 carregado.
Arco Line3 de place0 para transition5 com peso de 1 carregado.
Arco Line4 de transition5 para place2 com peso de 1 carregado.
Arco Line5 de place2 para transition4 com peso de 1 carregado.
Arco Line6 de transition4 para place0 com peso de 1 carregado.
Arco Externo Line7 de transition5 para transition6 carregado.

```

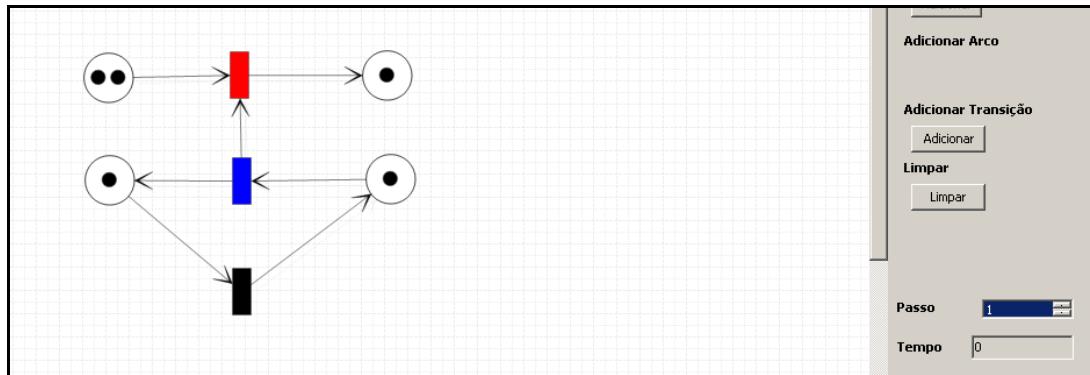
**Figura 5.2 - Mensagens exibidas após a preparação do modelo**

Alguns detalhes importantes sobre as mensagens exibidas:

- O ‘arco’ que une o ‘lugar’ que possui quatro marcas a uma ‘transição’ requisitada possui peso igual a 2.
- A ‘transição’ (em preto) é uma ‘transição’ temporizada cujo valor é 3.
- Outras ‘transições’ são instantâneas.
- Outros ‘arcos’ possuem ‘peso’ unitário.

Uma vez preparado o modelo, inicia-se a simulação local. Como exemplo, será iniciado o modo “Passo a Passo” (menu “Simulação”, “Simulação Local”, “Passo a Passo”). Com isso, conforme a Figura 5.3, e a mensagem exibida (Figura 5.4) ocorreu um disparo da ‘transição’ fundida.

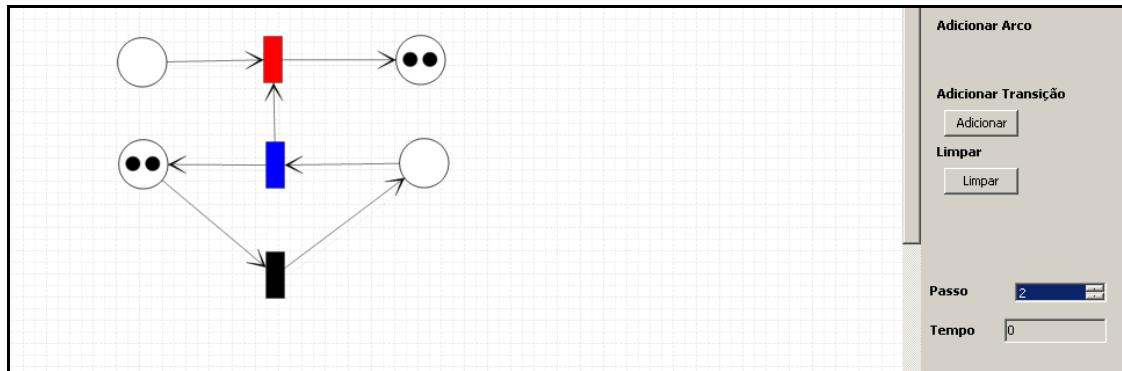
Executando mais um passo, haverá novamente o disparo da ‘transição’ fundida (Figura 5.5) e sua respectiva mensagem é exibida (Figura 5.6).



**Figura 5.3 - Evolução da RdP após o disparo da 'transição' fundida**

Avaliando rede 1  
 Avaliando rede 2  
 Existe ao menos uma transição instantânea sem fusão para disparar.  
 Atualizando rede 1 em 0  
 Atualizando rede 2 em 0  
 Transição transition5 disparada.

**Figura 5.4 - Mensagem após o disparo da 'transição'**



**Figura 5.5 - Evolução da RdP após o segundo disparo da 'transição' fundida**

Avaliando rede 1  
 Avaliando rede 2  
 Existe ao menos uma transição instantânea sem fusão para disparar.  
 Atualizando rede 1 em 0  
 Atualizando rede 2 em 0  
 Transição transition5 disparada.

**Figura 5.6 - Mensagem após o segundo disparo da 'transição' fundida**

Ao executar novamente mais um passo, uma vez que a 'transição' fundida não está mais habilitada, é disparada a 'transição' temporizada, alterando-se o valor "Tempo", que corresponde ao tempo local de simulação, para 3. As Figuras 5.7 e 5.8 representam a mudança devida ao disparo da 'transição' e as mensagens exibidas após esse disparo.

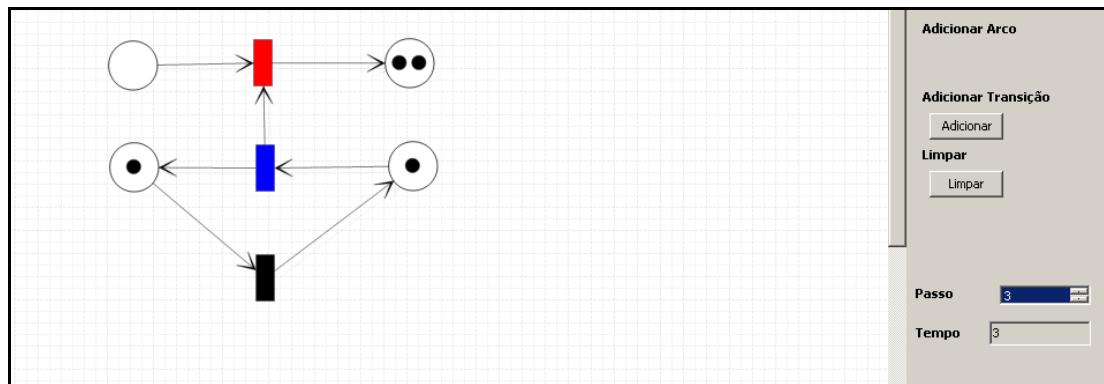


Figura 5.7 - Disparo da 'transição' temporizada

Avaliando rede 1  
 Rede 1 sem transições prontas para iniciar.(Deadlock 2)  
 Avaliando rede 2  
 Rede 2 sem transições instantâneas prontas para iniciar.(Deadlock 1)  
 Transição transition4 pronta para disparar.  
 Atualizando rede 1 em 3  
 Atualizando rede 2 em 3  
 Transição transition4 disparada.

Figura 5.8 - Mensagens exibidas após o disparo da 'transição' temporizada

Novamente, como somente a ‘transição’ temporizada está habilitada, ao executar mais um passo, o valor “Tempo” é alterado para 6 (Figura 5.9), uma vez que a ‘transição’ disparada possui valor de “Temporização” igual a 3. As mensagens exibidas após o disparo estão representadas na Figura 5.10.

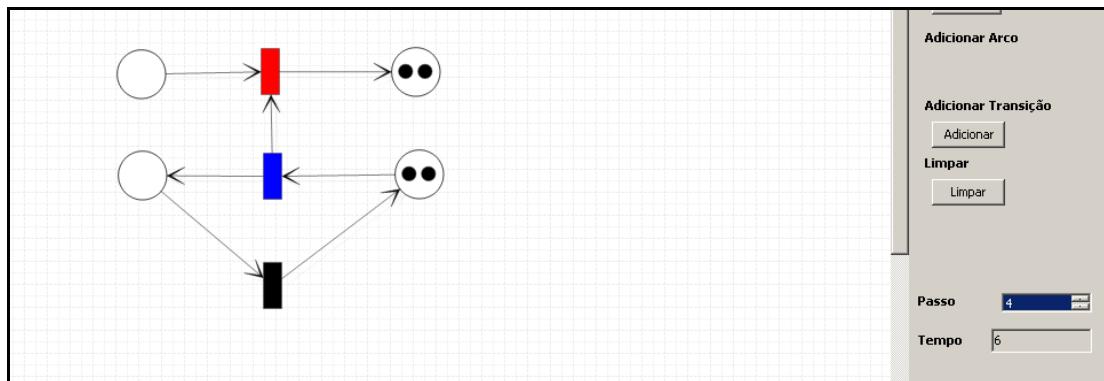


Figura 5.9 – Segundo disparo da 'transição' temporizada

Após esse disparo, ao tentar evoluir a RdP com mais um passo, mensagens exibindo que ambas as redes estão em *deadlock* com valor 2 indicam o fim da simulação (Figura 5.11).

Avaliando rede 1  
 Rede 1 sem transições prontas para iniciar.(Deadlock 2)  
 Avaliando rede 2  
 Rede 2 sem transições instantâneas prontas para iniciar.(Deadlock 1)  
 Transição transition4 pronta para disparar.  
 Atualizando rede 1 em 3  
 Atualizando rede 2 em 3  
 Transição transition4 disparada.

**Figura 5.10 - Mensagens exibidas após o segundo disparo da ‘transição’ temporizada**

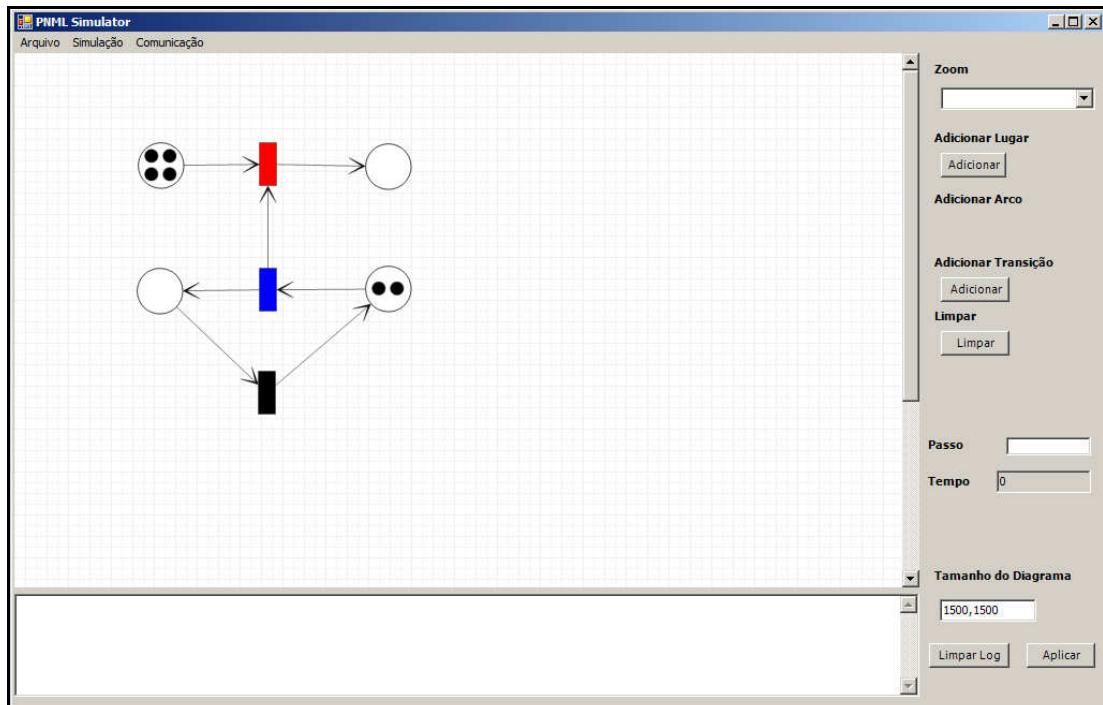
Avaliando rede 1  
 Rede 1 sem transições prontas para iniciar.(Deadlock 2)  
 Avaliando rede 2  
 Rede 2 sem transições prontas para iniciar.(Deadlock 2)

**Figura 5.11 - Ambos os modelos em deadlock indicam o fim da simulação**

## 5.2 Simulação distribuída entre duas estações

Este teste inclui modelar e simular modelos em RdP de forma distribuída entre duas estações (“Estação A” e “Estação B”).

Para realizar essa simulação (Figura 5.12) foi utilizado o mesmo exemplo da Figura 5.1, o qual há ‘transições’ fundidas e temporizadas, bem como outros elementos principais de uma RdP.



**Figura 5.12 - Modelo a ser simulado**

De modo a criar um anel lógico entre os dois computadores (Figura 5.13), foram configurados as conexões entre eles. A Figura 5.14 indica a configuração realizada na “Estação A” e a Figura 5.15 indica a configuração realizada na “Estação B”. Assim, uma vez configuradas as estações, ambas devem aguardar a conexão, para que uma estação possa conectar-se a outra.

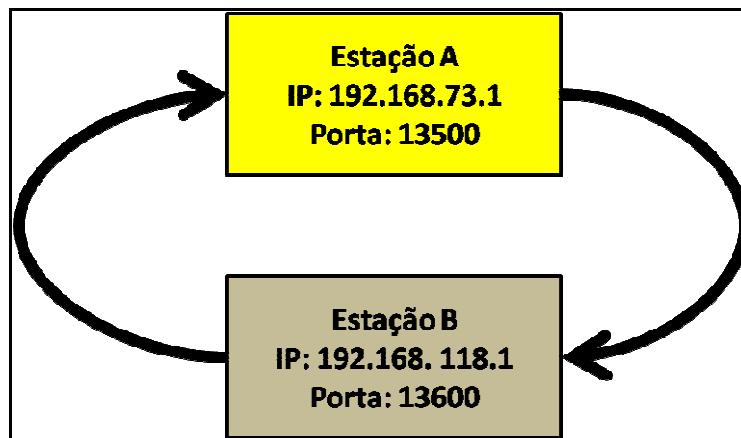


Figura 5.13 – Anel lógico entre duas estações

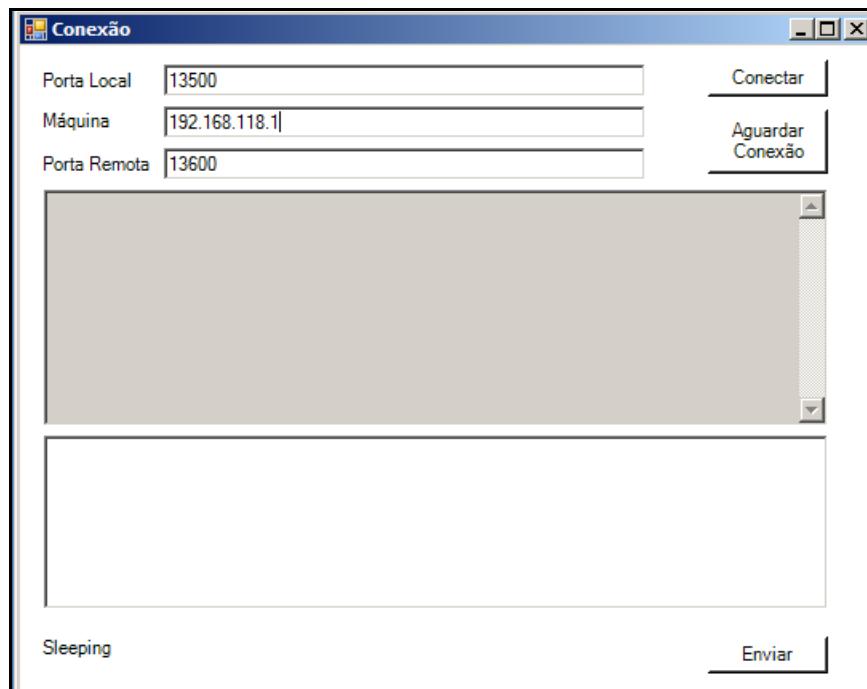


Figura 5.14 - Configuração de conexão da "Estação A"

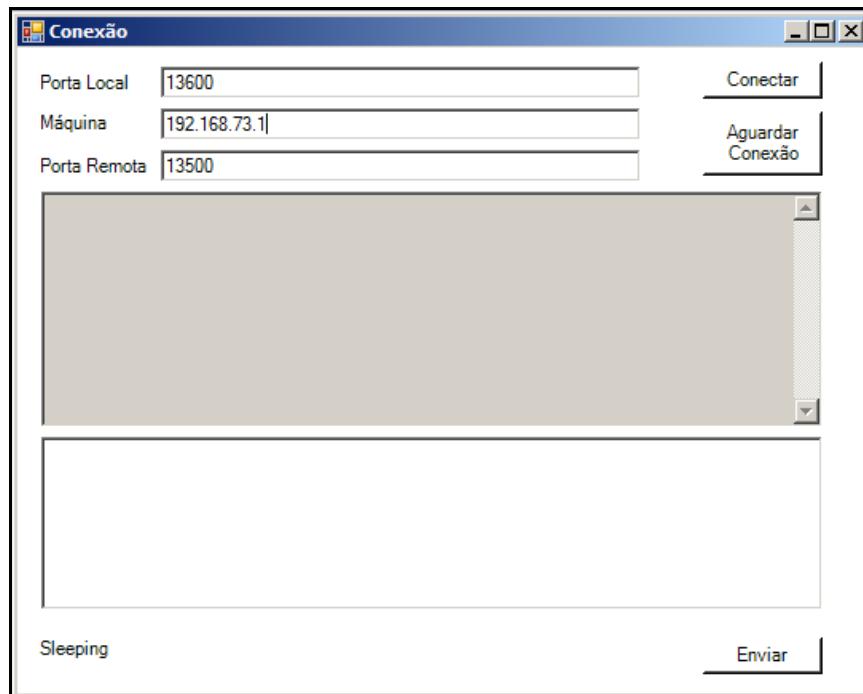


Figura 5.15 - Configuração de conexão da "Estação B"

Após ambas as estações estarem conectadas, as RdP são preparadas, ou seja, são carregadas nas estações para que cada uma possa realizar a simulação de uma parte da RdP. Nesta etapa, basta apenas que uma das estações solicite a preparação, para que o modelo seja carregado nas outras estações. As Figuras 5.16 e 5.18 indicam os modelos carregados na “Estação A” e “Estação B”, respectivamente.

Ao distribuir os modelos entre as estações, cada estação irá simular somente os modelos visíveis (opacos), ou seja, os modelos transparentes, que não possuem ‘marcas’, não serão simulados.

Neste exemplo, todas as estações foram simuladas passo a passo e a cada mudança na interface foi tirada uma “impressão” da tela.

## PASSO 0:

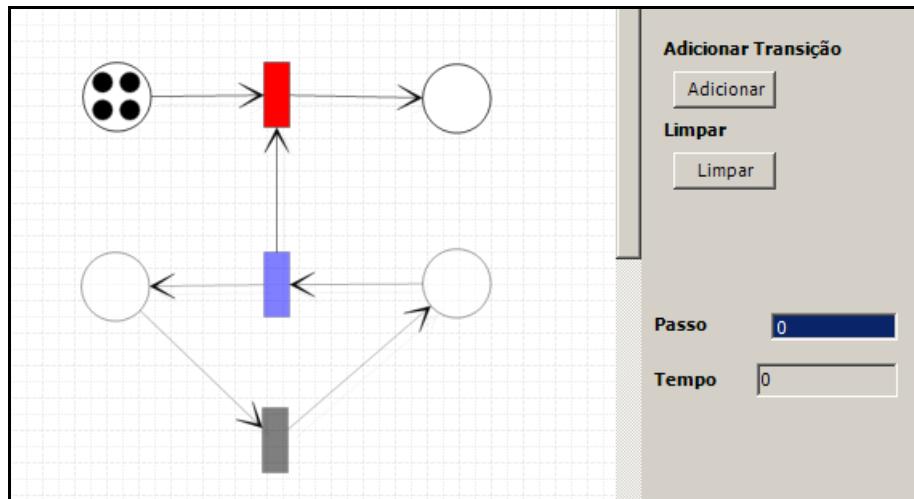


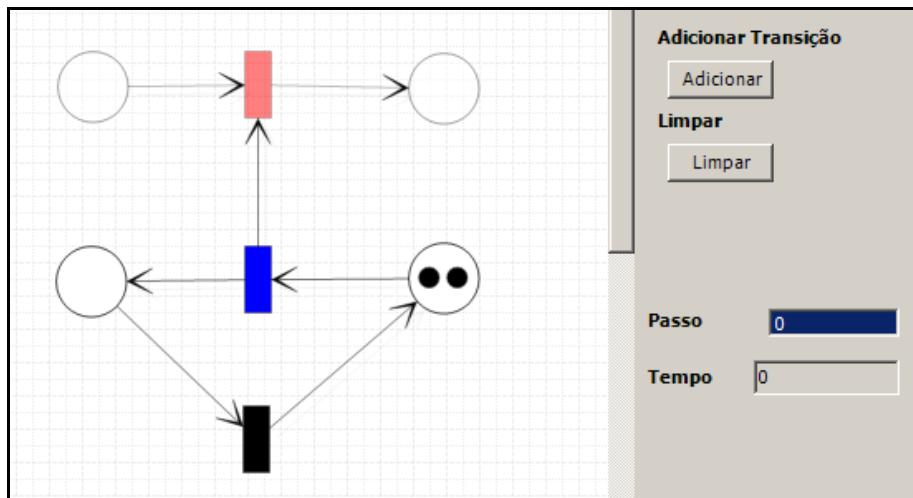
Figura 5.16 - Modelo a ser simulado na "Estação A"

```

Sent
Estações carregadas.
Place place1 carregado com 0 tokens.
Place place0 carregado com 2 tokens.
Place place2 carregado com 0 tokens.
Transição instantânea transition6 carregada.
Transição instantânea transition5 carregada.
Place place3 carregado com 4 tokens.
Transição transition4 temporizada em 3 carregada.
Arco Line1 de place3 para transition6 com peso de 2 carregado.
Arco Line2 de transition6 para place1 com peso de 1 carregado.
Arco Line3 de place0 para transition5 com peso de 1 carregado.
Arco Line4 de transition5 para place2 com peso de 1 carregado.
Arco Line5 de place2 para transition4 com peso de 1 carregado.
Arco Line6 de transition4 para place0 com peso de 1 carregado.
Arco Externo Line7 de transition5 para transition6 carregado.
2 redes distintas carregadas.
Sent
Reservando classe 1.
Waiting for a connection...
Reserva concluída, liberando token.
Sent
Waiting for a connection...
Rede 2 reservada pela estação 192.168.118.1:13600.
Sent
Waiting for a connection...
Todas classes reservadas.
Waiting for a connection...

```

Figura 5.17 - Mensagens exibidas após o preparo da RdP na "Estação A"



**Figura 5.18 - Modelo a ser simulado na "Estação B"**

```
temp192.168.118.1_13600.xml carregado.
Sent
Waiting for a connection...
Place place1 carregado com 0 tokens.
Place place0 carregado com 2 tokens.
Place place2 carregado com 0 tokens.
Transição instantânea transition6 carregada.
Transição instantânea transition5 carregada.
Place place3 carregado com 4 tokens.
Transição transition4 temporizada em 3 carregada.
Arco Line1 de place3 para transition6 com peso de 2 carregado.
Arco Line2 de transition6 para place1 com peso de 1 carregado.
Arco Line3 de place0 para transition5 com peso de 1 carregado.
Arco Line4 de transition5 para place2 com peso de 1 carregado.
Arco Line5 de place2 para transition4 com peso de 1 carregado.
Arco Line6 de transition4 para place0 com peso de 1 carregado.
Arco Externo Line7 de transition5 para transition6 carregado.
2 redes distintas carregadas.
Rede 1 reservada pela estação 192.168.118.1:13500.
Sent
Waiting for a connection...
Sent
Reservando classe 2.
Waiting for a connection...
Reserva concluída, liberando token.
Sent
Waiting for a connection...
```

**Figura 5.19 - Mensagens exibidas após o preparo da RdP na "Estação B"**

## PASSO 1:

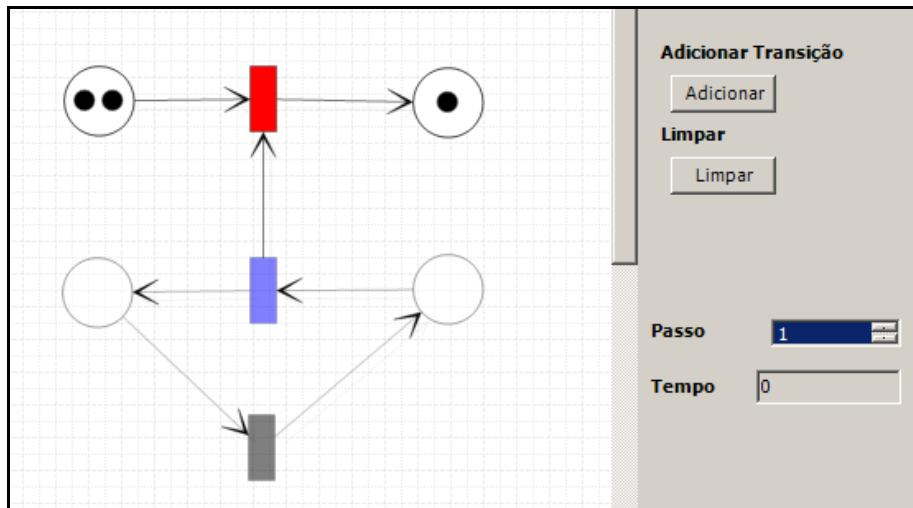


Figura 5.20 - Disparo da 'transição' fundida na "Estação A"

```

Sent
Avaliando rede 1
Rede 1 sem transições prontas para iniciar.(Deadlock 2)
Sent
Transição transition6 pronta, avaliada por 192.168.118.1:13600
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; 0 ; 1 ; umPasso ; 0
Sent
Atualizando rede 1 em 0
Waiting for a connection...
Token: 192.168.118.1:13500 ; 0 ; 2 ; umPasso ; 0
Todas estações em avançaram um passo.
Sent
Waiting for a connection...
Transição transition6 disparada remotamente por 192.168.118.1:13600
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; ; ; recarregar ; 0
Simulação Concluída.
Waiting for a connection...

```

Figura 5.21 - Mensagens exibidas na “Estação A” após o disparo da 'transição' fundida

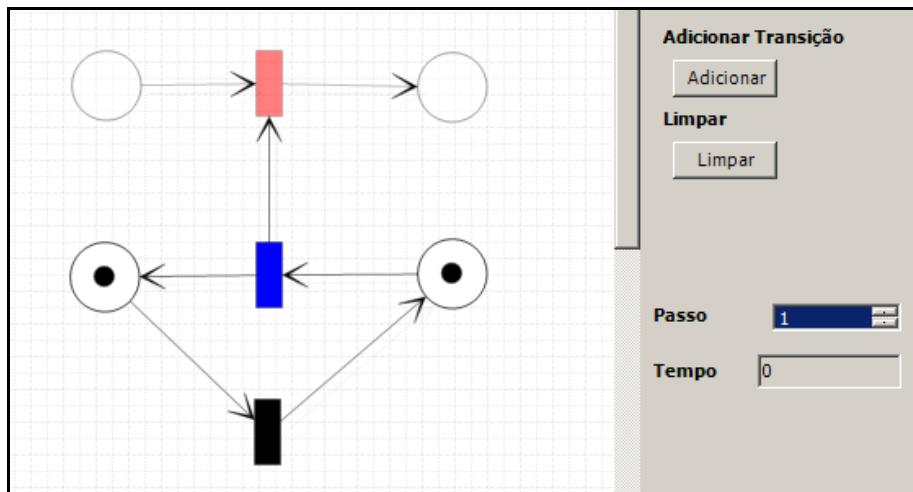


Figura 5.22 - Disparo da 'transição' fundida na "Estação B"

```

Token: ; 0 ; 0 ; 1 ; 0
Waiting for a connection...
Token: 192.168.118.1:13500 ; 0 ; 3 ; umPasso ; 0
Avaliando rede 2
Transição transition5 precisa avaliar transition6 localizada em 192.168.118.1:13500
Requisitando avaliação de transition6 em 192.168.118.1:13500
Sent
Waiting for a connection...
Transição transition6 pronta.
Todas transições necessárias foram avaliadas por esta estação.
Avaliando rede 2
Transição transition5 pronta para disparar.
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; 0 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 2 em 0
Transição transition5 solicita transition6 localizada em 192.168.118.1:13500
Requisitando disparo de transition6 em 192.168.118.1:13500
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...
Transição transition6 disparada com sucesso.
Todas estações requisitadas pela classe 2 responderam.
Atualizando rede 2 em 0
Transição transition5 disparada.
Waiting for a connection...
  
```

Figura 5.23 - Mensagens exibidas na “Estação B” após o disparo da 'transição' fundida

## PASSO 2:

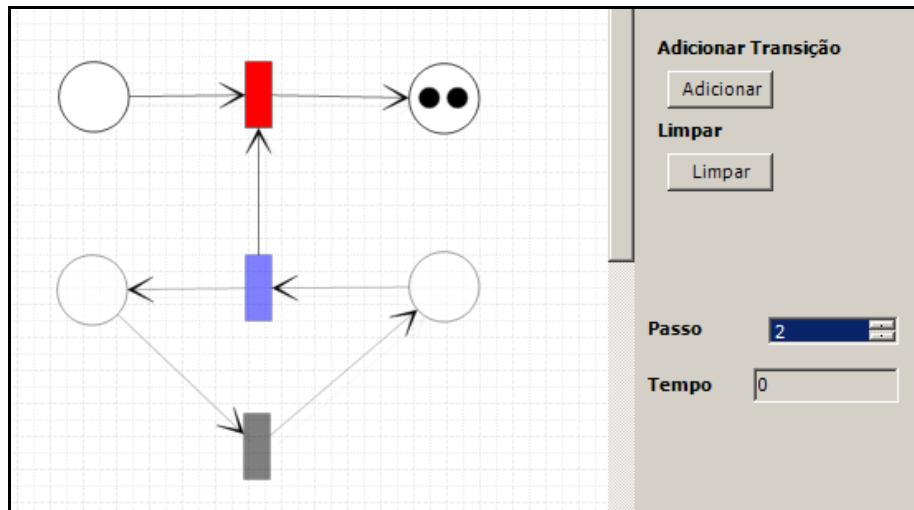


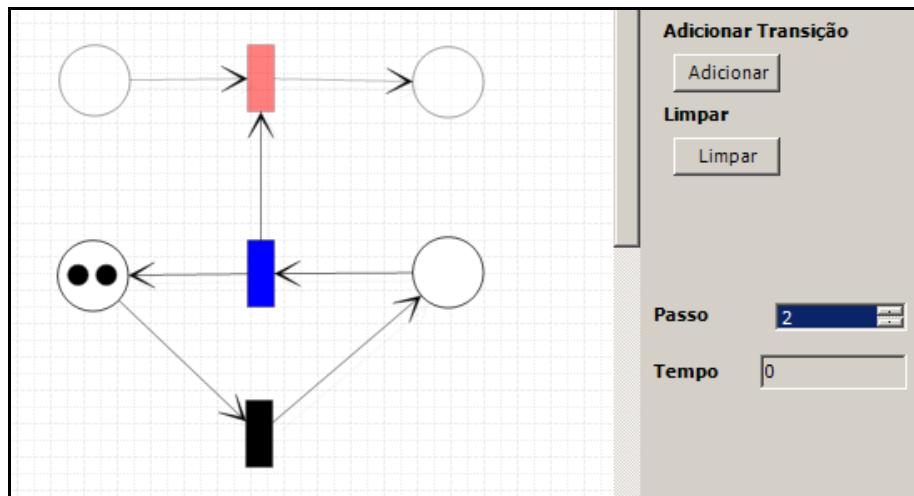
Figura 5.24 - Segundo disparo da ‘transição’ fundida na “Estação A”

```

Sent
Avaliando rede 1
Rede 1 sem transições prontas para iniciar.(Deadlock 2)
Sent
Transição transition6 pronta, avaliada por 192.168.118.1:13600
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; 0 ; 1 ; umPasso ; 0
Sent
Atualizando rede 1 em 0
Waiting for a connection...
Token: 192.168.118.1:13500 ; 0 ; 2 ; umPasso ; 0
Todas estações em avançaram um passo.
Sent
Waiting for a connection...
Transição transition6 disparada remotamente por 192.168.118.1:13600
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; ; ; recarregar ; 0
Simulação Concluida.
Waiting for a connection...

```

Figura 5.25 - Mensagens exibidas na “Estação A” após o segundo disparo da ‘transição’ fundida



**Figura 5.26 - Segundo disparo da ‘transição’ fundida na “Estação B”**

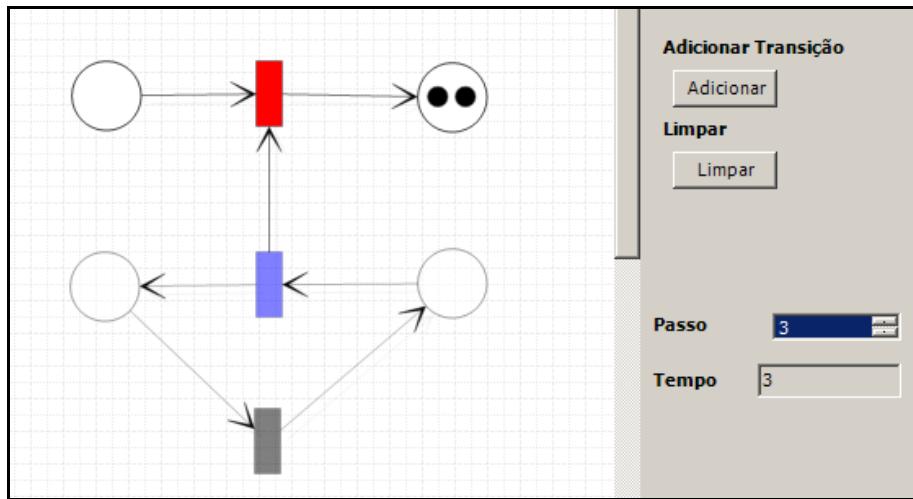
```

Token: ; 0 ; 0 ; 1 ; 0
Waiting for a connection...
Token: 192.168.118.1:13500 ; 0 ; 3 ; umPasso ; 0
Avaliando rede 2
Transição transition5 precisa avaliar transition6 localizada em 192.168.118.1:13500
Requisitando avaliação de transition6 em 192.168.118.1:13500
Sent
Waiting for a connection...
Transição transition6 pronta.
Todas transições necessárias foram avaliadas por esta estação.
Avaliando rede 2
Transição transition5 pronta para disparar.
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; 0 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 2 em 0
Transição transition5 solicita transition6 localizada em 192.168.118.1:13500
Requisitando disparo de transition6 em 192.168.118.1:13500
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...
Transição transition6 disparada com sucesso.
Todas estações requisitadas pela classe 2 responderam.
Atualizando rede 2 em 0
Transição transition5 disparada.
Waiting for a connection...

```

**Figura 5.27 - Mensagens exibidas na “Estação B” após o segundo disparo da ‘transição’ fundida**

### PASSO 3:



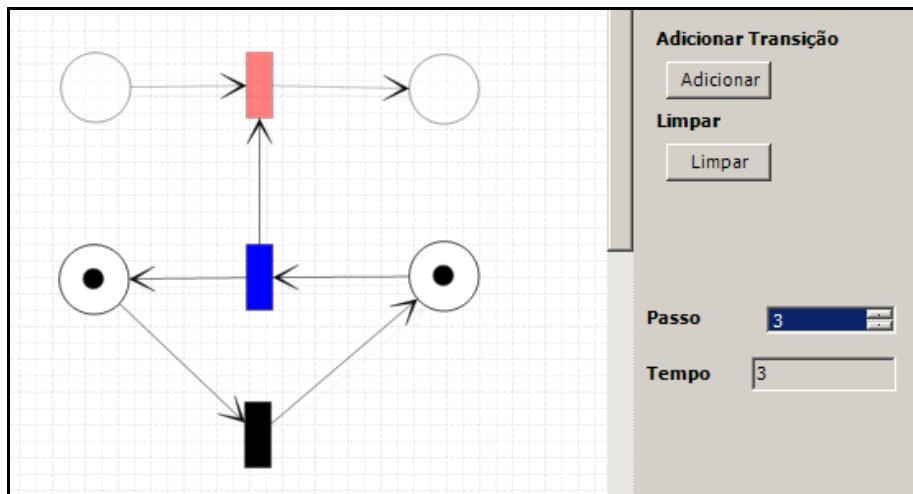
**Figura 5.28 - Não houve disparo de 'transições' na "Estação A", mas a variável Tempo alterou-se para 3**

```

Sent
Avaliando rede 1
Rede 1 sem transições prontas para iniciar.(Deadlock 2)
Sent
Token: 192.168.118.1:13500 ; 3 ; 1 ; umPasso ; 0
Sent
Atualizando rede 1 em 3
Waiting for a connection...
Token: 192.168.118.1:13500 ; 3 ; 2 ; umPasso ; 0
Todas estações em avançaram um passo.
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; ; ; recarregar ; 0
Simulação Concluída.
Waiting for a connection...

```

**Figura 5.29 - Mensagens exibidas na "Estação A"**



**Figura 5.30 - Disparo da 'transição' temporizada na "Estação B" e alteração do Tempo para 3**

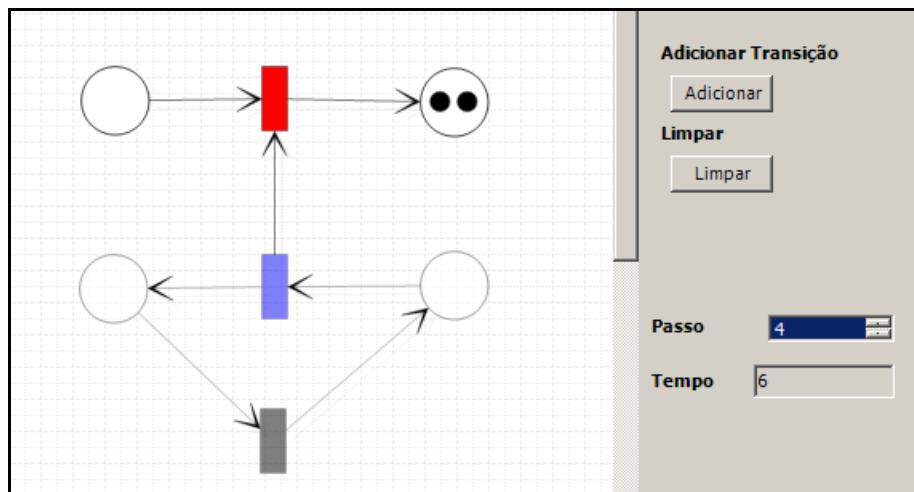
```

Token: ; 0 ; 0 ; 1 ; 0
Waiting for a connection...
Token: 192.168.118.1:13500 ; 0 ; 3 ; umPasso ; 0
Avaliando rede 2
Rede 2 sem transições instantâneas prontas para iniciar.(Deadlock 1)
Transição transition4 pronta para disparar.
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; 3 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 2 em 3
Transição transition4 disparada.
Waiting for a connection...
Token: 192.168.118.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...

```

**Figura 5.31 - Mensagens exibidas na "Estação B"**

#### PASSO 4:



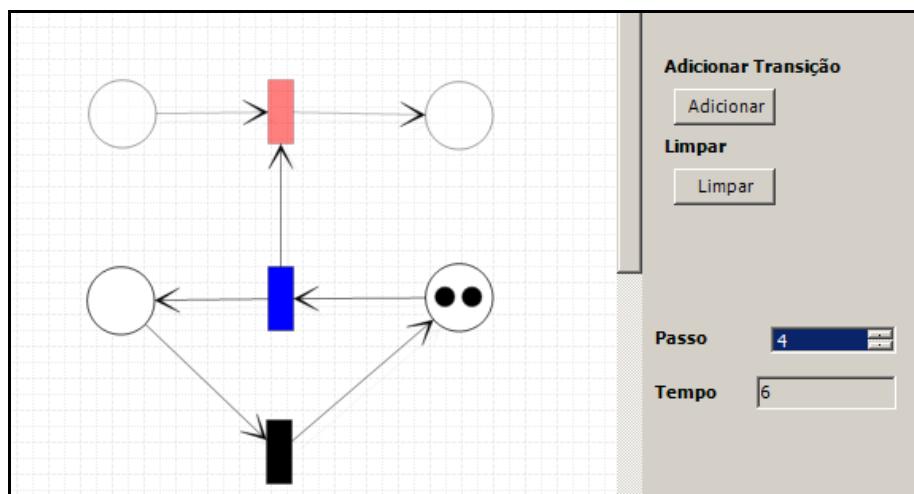
**Figura 5.32 - Não houve disparo de 'transições' na "Estação A", mas a variável Tempo alterou-se de 3 para 6**

```

Sent
Avaliando rede 1
Rede 1 sem transições prontas para iniciar.(Deadlock 2)
Sent
Transição transition6 não está pronta, avaliada para 192.168.118.1:13600
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; 3 ; 1 ; umPasso ; 0
Sent
Atualizando rede 1 em 3
Waiting for a connection...
Token: 192.168.118.1:13500 ; 3 ; 2 ; umPasso ; 0
Todas estações avançaram um passo.
Sent
Waiting for a connection...
Transição transition6 não pode ser disparada remotamente para 192.168.118.1:13600
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; ; ; recarregar ; 0
Simulação Concluída.
Waiting for a connection...

```

**Figura 5.33 - Mensagens exibidas na "Estação A"**



**Figura 5.34 - Disparo da 'transição' temporizada na "Estação B" e alteração do Tempo de 3 para 6**

```

Token: ; 0 ; 0 ; 1 ; 0
Waiting for a connection...
Token: 192.168.118.1:13500 ; 0 ; 3 ; umPasso ; 0
Avaliando rede 2
Transição transition5 precisa avaliar transition6 localizada em 192.168.118.1:13500
Requisitando avaliação de transition6 em 192.168.118.1:13500
Sent
Waiting for a connection...
Transição transition6 não está pronta.
Todas transições necessárias foram avaliadas por esta estação.
Avaliando rede 2
Rede 2 sem transições instantâneas prontas para iniciar.(Deadlock 1)
Transição transition4 pronta para disparar.
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; 3 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 2 em 3
Transição transition5 solicita transition6 localizada em 192.168.118.1:13500
Requisitando disparo de transition6 em 192.168.118.1:13500
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...
Transição transition6 não pode ser disparada.
Todas estações requisitadas pela classe 2 responderam.
Atualizando rede 2 em 3
Transição transition4 disparada.
Waiting for a connection...

```

**Figura 5.35 - Mensagens exibidas na "Estação B"**

### FIM DA SIMULAÇÃO:

Na tentativa de execução de mais um passo, o sistema percebe que não há mais ‘transições’ (instantâneas e temporizadas) disparáveis indicando, portanto, o fim da simulação.

```

Sent
Avaliando rede 1
Rede 1 sem transições prontas para iniciar.(Deadlock 2)
Sent
Transição transition6 não está pronta, avaliada para 192.168.118.1:13600
Sent
Waiting for a connection...
Token: 192.168.118.1:13500 ; 0 ; 3 ; umPasso ; 0
Todas estações em deadlock 2
Waiting for a connection...

```

**Figura 5.36 - Mensagens exibidas na "Estação A" após tentativa de execução de mais um passo**

```

Token: ; 0 ; 0 ; 1 ; 0
Waiting for a connection...
Token: 192.168.118.1:13500 ; 0 ; 3 ; umPasso ; 0
Avaliando rede 2
Transição transition5 precisa avaliar transition6 localizada em 192.168.118.1:13500
Requisitando avaliação de transition6 em 192.168.118.1:13500
Sent
Waiting for a connection...
Transição transition6 não está pronta.
Todas transições necessárias foram avaliadas por esta estação.
Avaliando rede 2
Rede 2 sem transições prontas para iniciar.(Deadlock 2)
Sent
Waiting for a connection...

```

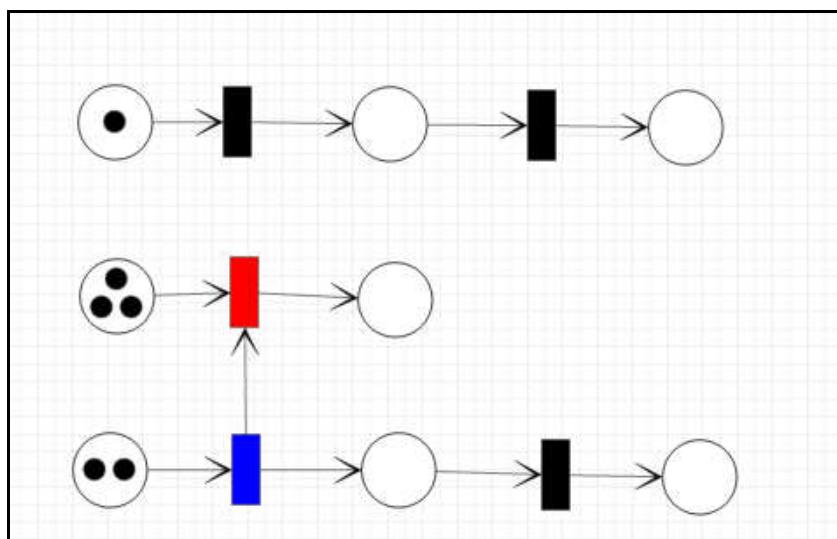
**Figura 5.37 - Mensagens exibidas na "Estação B" após tentativa de execução de mais um passo**

### 5.3 Simulação distribuída entre três estações

Este teste inclui modelar e simular modelos em RdP de forma distribuída entre três estações (“Estação A”, “Estação B” e “Estação C”).

Para realizar essa simulação foi utilizado um novo exemplo, ilustrado na Figura 5.38, o qual há ‘transições’ fundidas e temporizadas, bem como outros elementos principais de uma RdP.

A primeira ‘transição’ do primeiro modelo, que não possui ‘transições’ fundidas, possui valor de Tempo igual a 2. As outras ‘transições’ não fundidas de outros modelos possuem valores de Tempo iguais a 1.



**Figura 5.38 - Modelo a ser simulado entre três estações**

De modo a criar um anel lógico entre os três computadores (Figura 5.39), foram configurados as conexões entre eles. A Figura 5.40 indica a configuração realizada na “Estação A”, a Figura 5.41 indica a configuração realizada na “Estação B” e a Figura 5.42 indica a configuração realizada na “Estação C”. Assim, uma vez configuradas as estações, todas devem aguardar a conexão, para que uma estação possa conectar-se a outra.

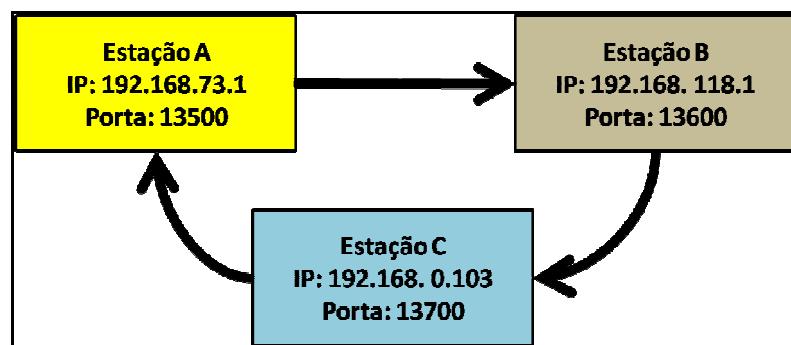


Figura 5.39 - Anel lógico entre três estações

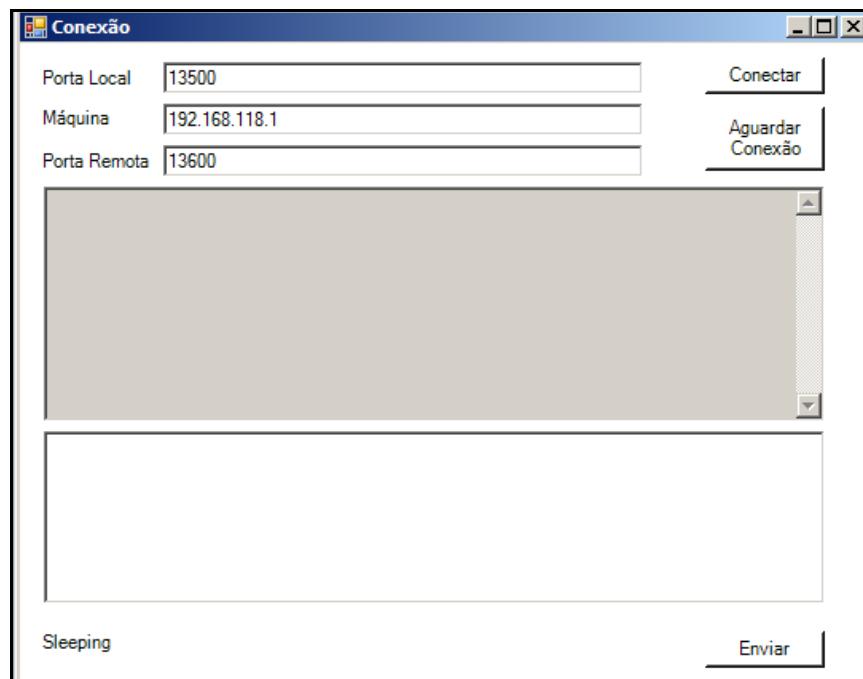


Figura 5.40 - Configuração de Conexão da "Estação A"

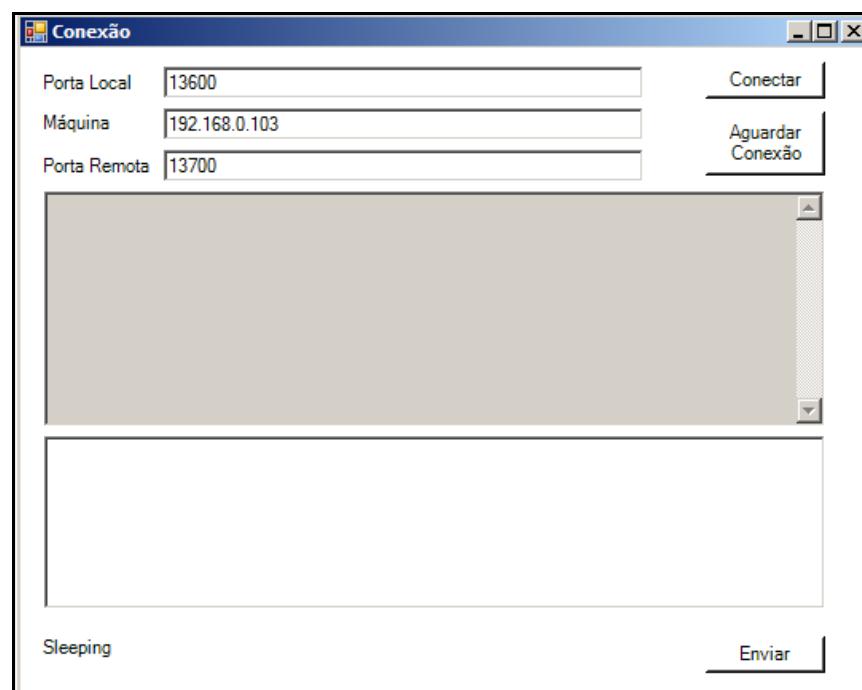


Figura 5.41 - Configuração de Conexão da "Estação B"

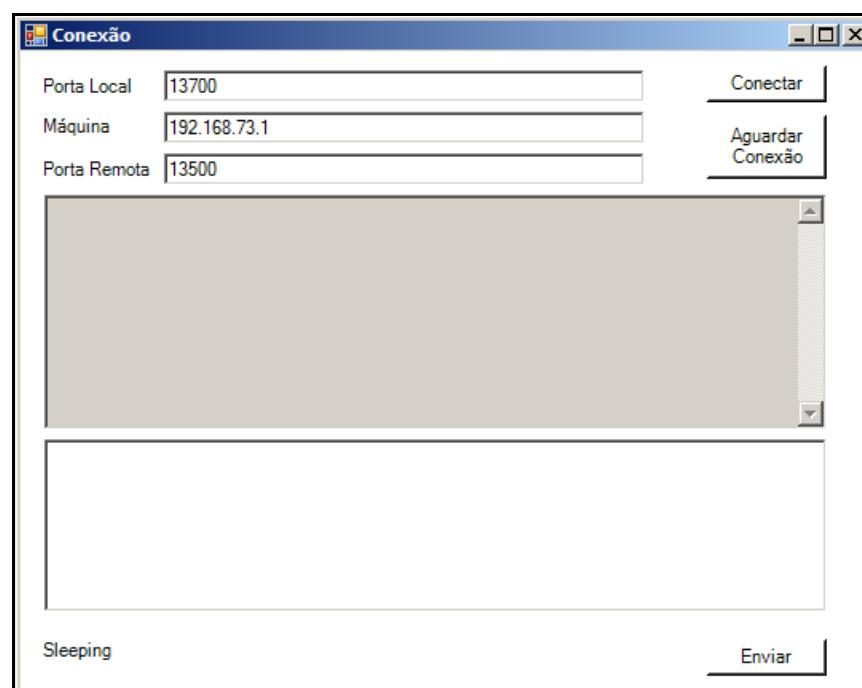


Figura 5.42 - Configuração de Conexão da "Estação C"

As Figuras 5.43, 5.45 e 5.47 indicam os modelos carregados na "Estação A", "Estação B" e "Estação C", respectivamente.

## PASSO 0:

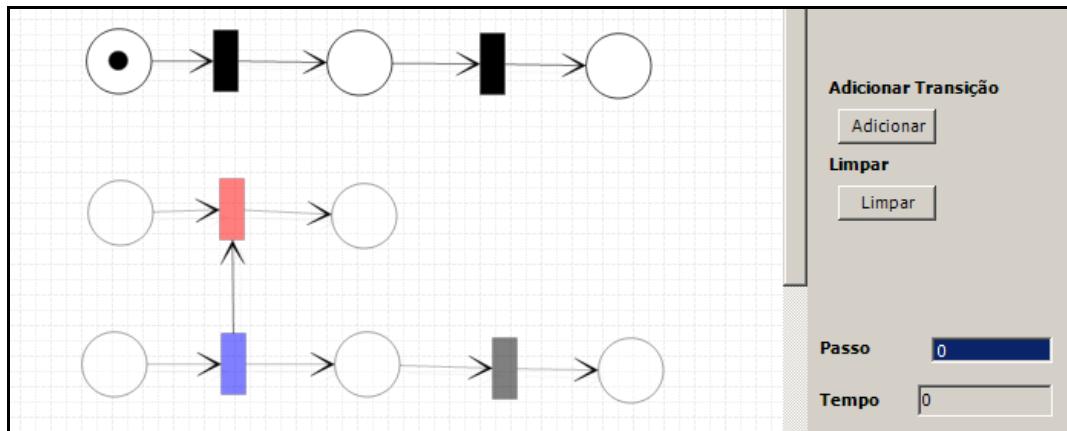


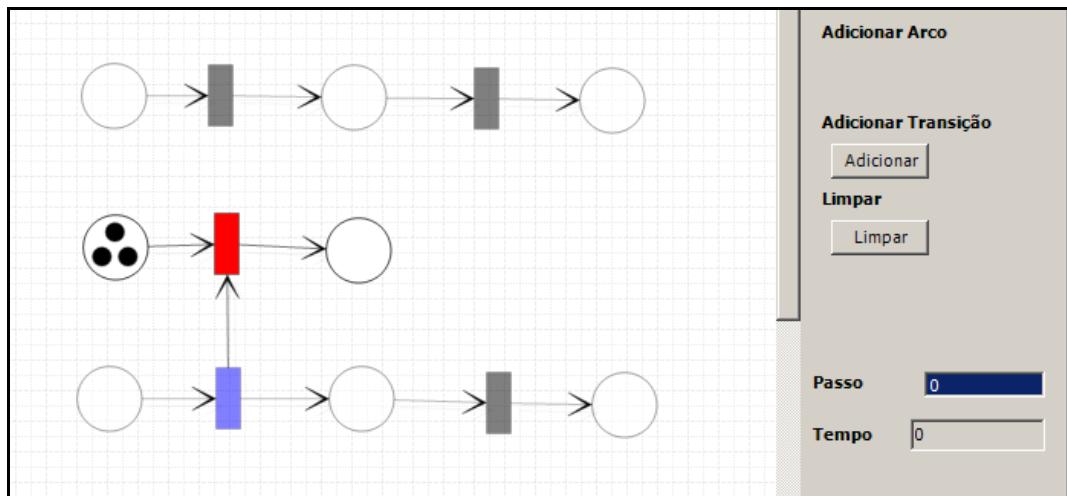
Figura 5.43 - Modelo a ser simulado na "Estação A"

```

Sent
Estações carregadas.
Transição transition12 temporizada em 2 carregada.
Transição transition8 temporizada em 1 carregada.
Place place4 carregado com 0 tokens.
Transição instantânea transition9 carregada.
Place place0 carregado com 0 tokens.
Transição instantânea transition10 carregada.
Place place2 carregado com 3 tokens.
Place place6 carregado com 0 tokens.
Place place5 carregado com 0 tokens.
Place place3 carregado com 1 tokens.
Transição transition11 temporizada em 1 carregada.
Place place7 carregado com 0 tokens.
Place place1 carregado com 2 tokens.
Arco Line1 de place3 para transition12 com peso de 1 carregado.
Arco Line2 de transition12 para place0 com peso de 1 carregado.
Arco Line3 de place0 para transition11 com peso de 1 carregado.
Arco Line4 de transition11 para place4 com peso de 1 carregado.
Arco Line5 de place2 para transition10 com peso de 1 carregado.
Arco Line6 de transition10 para place6 com peso de 1 carregado.
Arco Line7 de place1 para transition9 com peso de 1 carregado.
Arco Line10 de transition8 para place7 com peso de 1 carregado.
Arco Line8 de transition9 para place5 com peso de 1 carregado.
Arco Line9 de place5 para transition8 com peso de 1 carregado.
Arco Externo Line11 de transition9 para transition10 carregado.
3 redes distintas carregadas.
Sent
Reservando classe 1.
Waiting for a connection...
Reserva concluída, liberando token.
Sent
Waiting for a connection...
Rede 2 reservada pela estação 192.168.118.1:13600.
Sent
Rede 3 reservada pela estação 192.168.0.103:13700.
Sent
Waiting for a connection...
Todas classes reservadas.
Waiting for a connection...

```

Figura 5.44 - Mensagens exibidas após o preparo da RdP na “Estação A”

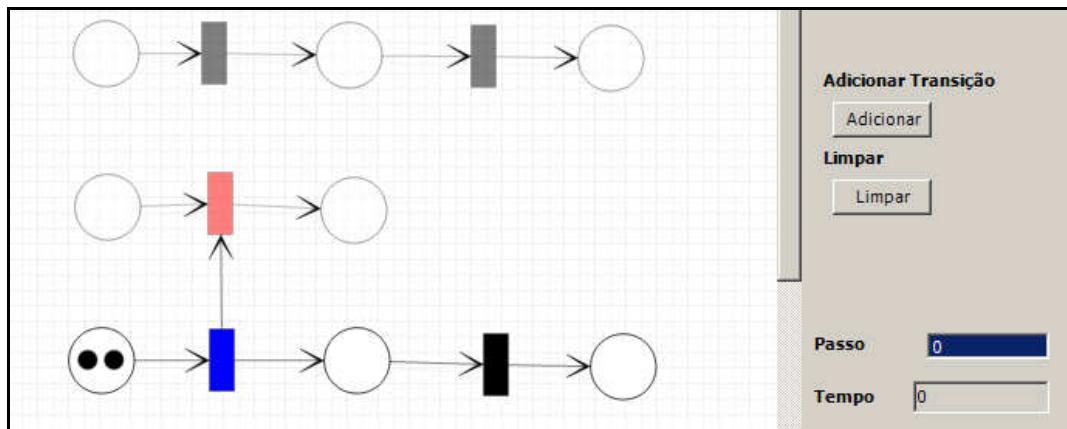


**Figura 5.45 - Modelo a ser simulado na "Estação B"**

```

temp192.168.118.1_13600.xml carregado.
Sent
Waiting for a connection...
Transição transition12 temporizada em 2 carregada.
Transição transition8 temporizada em 1 carregada.
Place place4 carregado com 0 tokens.
Transição instantânea transition9 carregada.
Place place0 carregado com 0 tokens.
Transição instantânea transition10 carregada.
Place place2 carregado com 3 tokens.
Place place6 carregado com 0 tokens.
Place place5 carregado com 0 tokens.
Place place3 carregado com 1 tokens.
Transição transition11 temporizada em 1 carregada.
Place place7 carregado com 0 tokens.
Place place1 carregado com 2 tokens.
Arco Line1 de place3 para transition12 com peso de 1 carregado.
Arco Line2 de transition12 para place0 com peso de 1 carregado.
Arco Line3 de place0 para transition11 com peso de 1 carregado.
Arco Line4 de transition11 para place4 com peso de 1 carregado.
Arco Line5 de place2 para transition10 com peso de 1 carregado.
Arco Line6 de transition10 para place6 com peso de 1 carregado.
Arco Line7 de place1 para transition9 com peso de 1 carregado.
Arco Line10 de transition8 para place7 com peso de 1 carregado.
Arco Line8 de transition9 para place5 com peso de 1 carregado.
Arco Line9 de place5 para transition8 com peso de 1 carregado.
Arco Externo Line11 de transition9 para transition10 carregado.
3 redes distintas carregadas.
Rede 1 reservada pela estação 192.168.73.1:13500.
Sent
Waiting for a connection...
Sent
Reservando classe 2.
Waiting for a connection...
Reserva concluída, liberando token.
Sent
Waiting for a connection...
Rede 3 reservada pela estação 192.168.0.103:13700.
Waiting for a connection...
  
```

**Figura 5.46 - Mensagens exibidas após o preparo da RdP na "Estação B"**



**Figura 5.47 - Modelo a ser simulado na "Estação B"**

```

temp192.168.0.103_13700.xml carregado.Sent
Transição transition12 temporizada em 2 carregada.
Transição transition8 temporizada em 1 carregada.
Place place4 carregado com 0 tokens.
Transição instantânea transition9 carregada.
Place place0 carregado com 0 tokens.
Transição instantânea transition10 carregada.
Place place2 carregado com 3 tokens.
Place place6 carregado com 0 tokens.
Place place5 carregado com 0 tokens.
Place place3 carregado com 1 tokens.
Transição transition11 temporizada em 1 carregada.
Place place7 carregado com 0 tokens.
Place place1 carregado com 2 tokens.
Arco Line1 de place3 para transition12 com peso de 1 carregado.
Arco Line2 de transition12 para place0 com peso de 1 carregado.
Arco Line3 de place0 para transition11 com peso de 1 carregado.
Arco Line4 de transition11 para place4 com peso de 1 carregado.
Arco Line5 de place2 para transition10 com peso de 1 carregado.
Arco Line6 de transition10 para place6 com peso de 1 carregado.
Arco Line7 de place1 para transition9 com peso de 1 carregado.
Arco Line10 de transition8 para place7 com peso de 1 carregado.
Arco Line8 de transition9 para place5 com peso de 1 carregado.
Arco Line9 de place5 para transition8 com peso de 1 carregado.
Arco Externo Line11 de transition9 para transition10 carregado.
3 redes distintas carregadas.
Waiting for a connection...
Rede 1 reservada pela estação 192.168.73.1:13500.
Sent
Waiting for a connection...
Rede 2 reservada pela estação 192.168.118.1:13600.
Sent
Waiting for a connection...
Sent
Reservando classe 3.
Waiting for a connection...
Reserva concluída, liberando token.
Sent
Waiting for a connection...

```

**Figura 5.48 - Mensagens exibidas após o preparo da RdP na “Estação C”**

### PASSO 1:

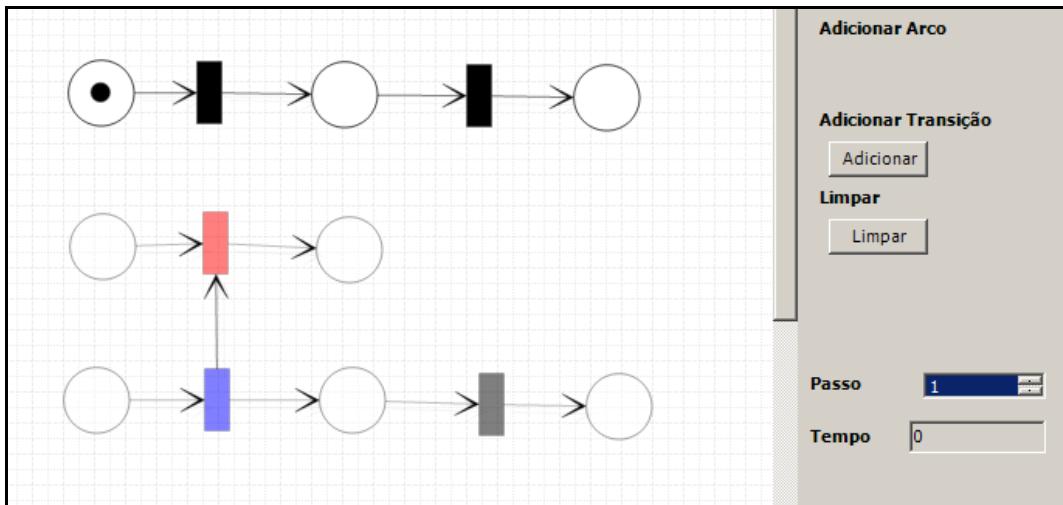


Figura 5.49 – Não houve disparo de ‘transições’ na “Estação A”

```

Sent
Avaliando rede 1
Rede 1 sem transições instantâneas prontas para iniciar.(Deadlock 1)
Transição transition12 pronta para disparar.
Sent
Token: 192.168.73.1:13500 ; 0 ; 1 ; umPasso ; 0
Sent
Atualizando rede 1 em 0
Waiting for a connection...
Token: 192.168.73.1:13500 ; 0 ; 2 ; umPasso ; 0
Todas estações em avançaram um passo.
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Simulação Concluida.
Waiting for a connection...

```

Figura 5.50 - Mensagens exibidas na "Estação A" após o primeiro passo

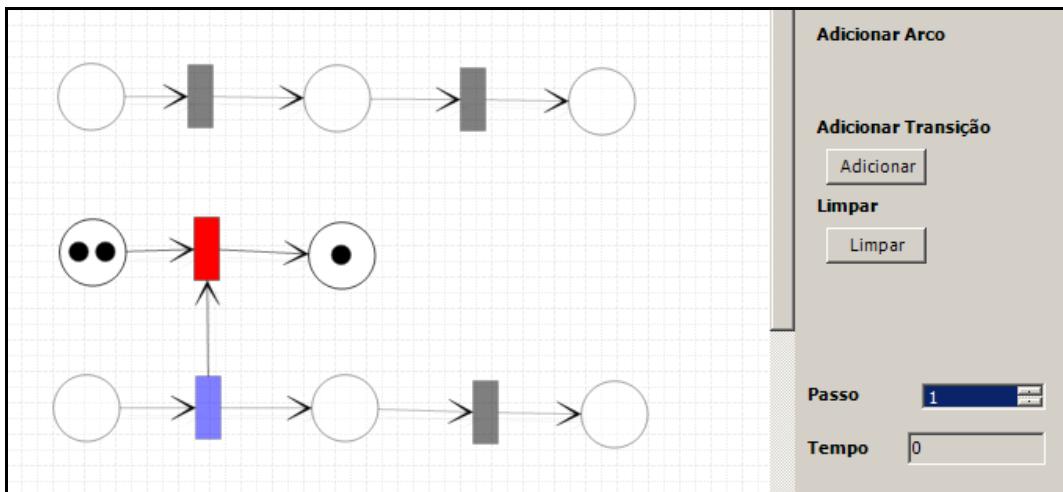


Figura 5.51 - Disparo da 'transição' fundida da "Estação B"

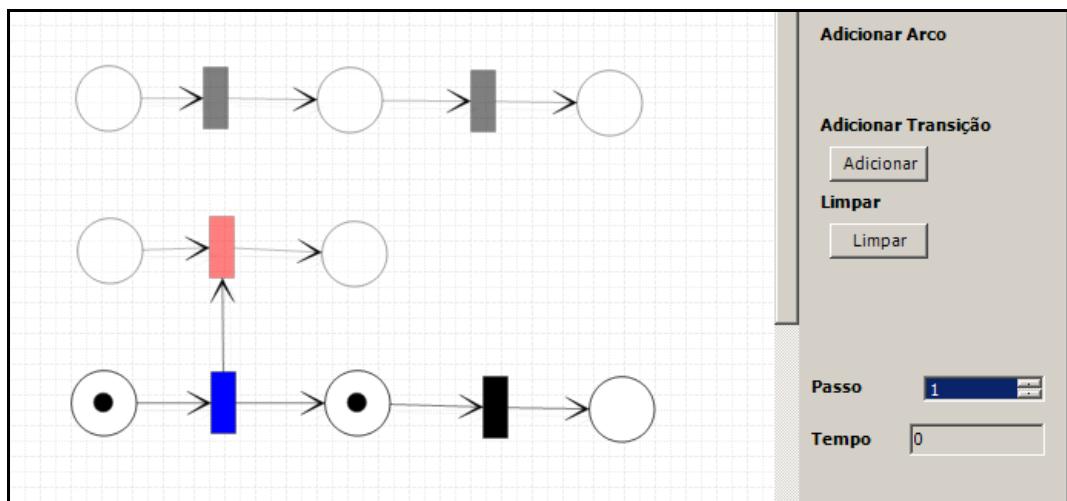
Token: ; 0 ; 0 ; 1 ; 0

```

Waiting for a connection...
Token: 192.168.73.1:13500 ; 2 ; 1 ; umPasso ; 0
Avaliando rede 2
Rede 2 sem transições prontas para iniciar.(Deadlock 2)
Sent
Waiting for a connection...
Transição transition10 pronta, avaliada por 192.168.0.103:13700
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; 0 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 2 em 0
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...
Transição transition10 disparada remotamente por 192.168.0.103:13700
Sent
Waiting for a connection...

```

**Figura 5.52 – Mensagens exibidas na “Estação B” após o disparo da ‘transição’ fundida**



**Figura 5.53 - Disparo da 'transição' fundida da “Estação C”**

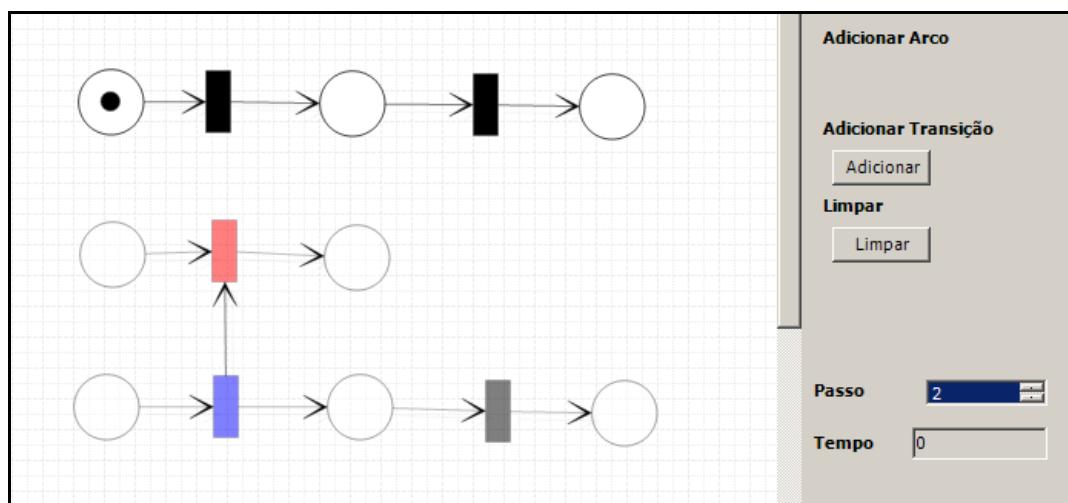
```

Token: 192.168.73.1:13500 ; 2 ; 1 ; umPasso ; 0
Avaliando rede 3
Transição transition9 precisa avaliar transition10 localizada em 192.168.118.1:13600
Requisitando avaliação de transition10 em 192.168.118.1:13600
Sent
Waiting for a connection...
Transição transition10 pronta.
Todas transições necessárias foram avaliadas por esta estação.
Avaliando rede 3
Transição transition9 pronta para disparar.
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; 0 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 3 em 0
Transição transition9 solicita transition10 localizada em 192.168.118.1:13600
Requisitando disparo de transition10 em 192.168.118.1:13600
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...
Transição transition10 disparada com sucesso.
Todas estações requisitadas pela classe 3 responderam.
Atualizando rede 3 em 0
Transição transition9 disparada.
Waiting for a connection...

```

**Figura 5.54 - Mensagens exibidas na "Estação C" após o disparo da 'transição' fundida**

## PASSO 2:



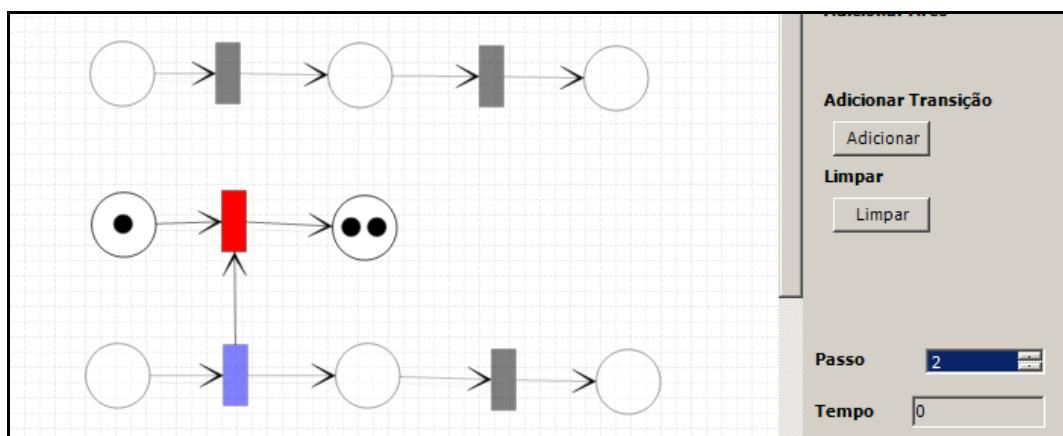
**Figura 5.55 - Não houve disparo de 'transições' na "Estação A"**

```

Sent
Avaliando rede 1
Rede 1 sem transições instantâneas prontas para iniciar.(Deadlock 1)
Transição transition12 pronta para disparar.
Sent
Token: 192.168.73.1:13500 ; 0 ; 1 ; umPasso ; 0
Sent
Atualizando rede 1 em 0
Waiting for a connection...
Token: 192.168.73.1:13500 ; 0 ; 2 ; umPasso ; 0
Todas estações em avançaram um passo.
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Simulação Concluida.
Waiting for a connection...

```

**Figura 5.56 - Mensagens exibidas na "Estação A" após o segundo passo**



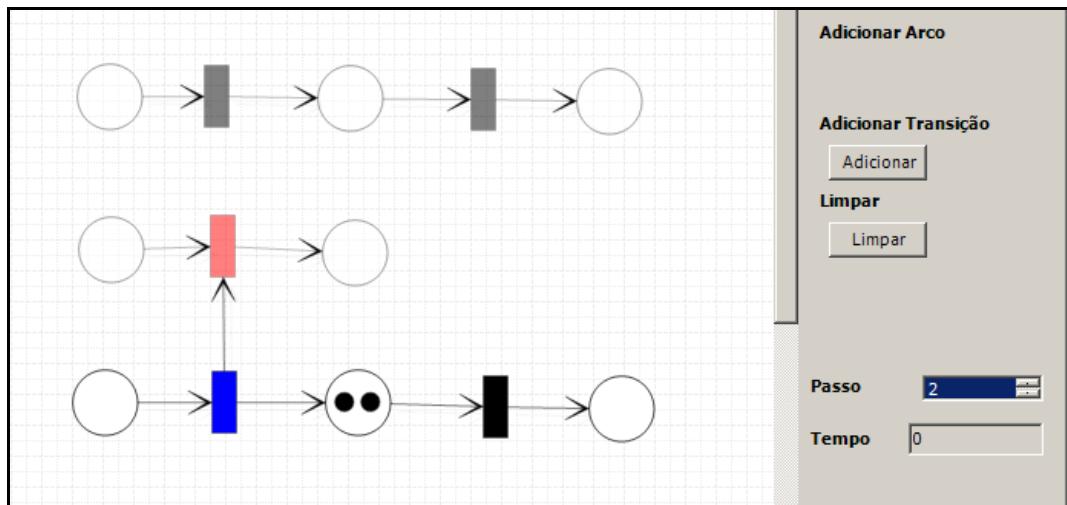
**Figura 5.57 – Segundo disparo da 'transição' fundida da "Estação B"**

```

Token: ; 0 ; 0 ; 1 ; 0
Waiting for a connection...
Token: 192.168.73.1:13500 ; 2 ; 1 ; umPasso ; 0
Avaliando rede 2
Rede 2 sem transições prontas para iniciar.(Deadlock 2)
Sent
Waiting for a connection...
Transição transition10 pronta, avaliada por 192.168.0.103:13700
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; 0 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 2 em 0
Waiting for a connection...
Transição transition10 disparada remotamente por 192.168.0.103:13700
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...

```

**Figura 5.58 - Mensagens exibidas na "Estação B" após o segundo disparo da 'transição' fundida**



**Figura 5.59 – Segundo disparo da 'transição' fundida da "Estação C"**

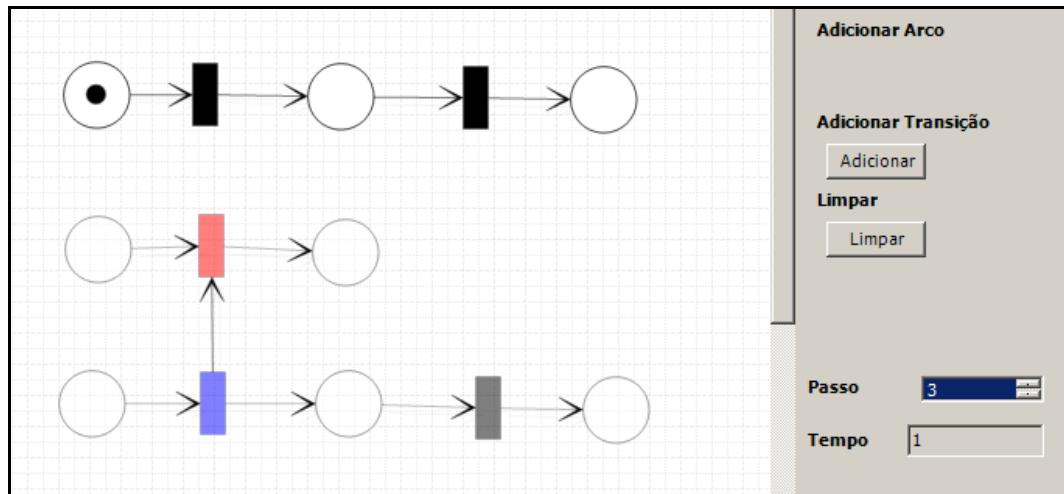
```

Token: 192.168.73.1:13500 ; 2 ; 1 ; umPasso ; 0
Avaliando rede 3
Transição transition9 precisa avaliar transition10 localizada em 192.168.118.1:13600
Requisitando avaliação de transition10 em 192.168.118.1:13600
Sent
Waiting for a connection...
Transição transition10 pronta.
Todas transições necessárias foram avaliadas por esta estação.
Avaliando rede 3
Transição transition9 pronta para disparar.
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; 0 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 3 em 0
Transição transition9 solicita transition10 localizada em 192.168.118.1:13600
Requisitando disparo de transition10 em 192.168.118.1:13600
Sent
Waiting for a connection...
Transição transition10 disparada com sucesso.
Todas estações requisitadas pela classe 3 responderam.
Atualizando rede 3 em 0
Transição transition9 disparada.
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...

```

**Figura 5.60 - Mensagens exibidas na "Estação C" após o segundo disparo da 'transição' fundida**

### PASSO 3:



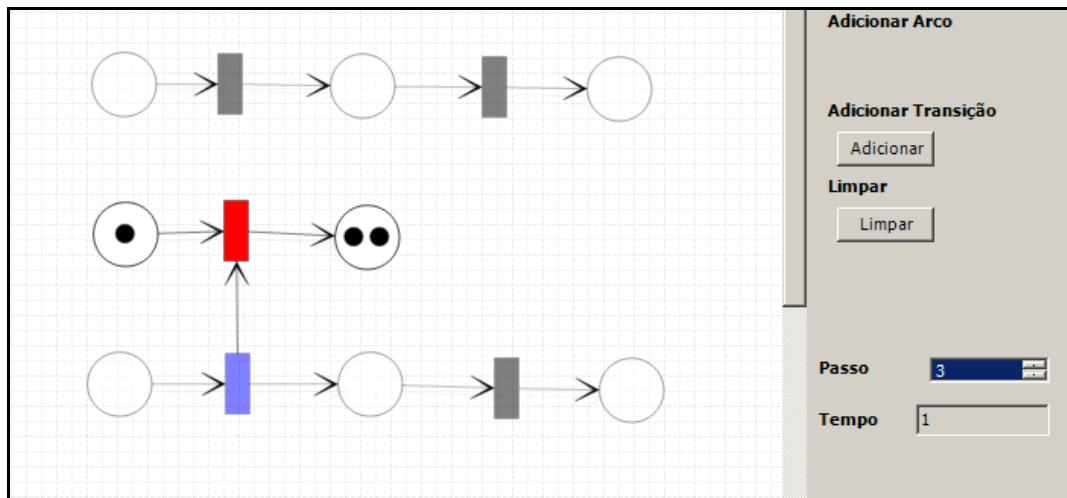
**Figura 5.61 - Não houve disparo de 'transições' na “Estação A”, no entanto, o valor de Tempo foi atualizado para 1**

```

Sent
Avaliando rede 1
Rede 1 sem transições instantâneas prontas para iniciar.(Deadlock 1)
Transição transition12 pronta para disparar.
Sent
Token: 192.168.73.1:13500 ; 1 ; 1 ; umPasso ; 0
Sent
Atualizando rede 1 em 1
Waiting for a connection...
Token: 192.168.73.1:13500 ; 1 ; 2 ; umPasso ; 0
Todas estações em avançaram um passo.
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Simulação Concluida.
Waiting for a connection...

```

**Figura 5.62 - Mensagens exibidas na "Estação A" após o terceiro passo**



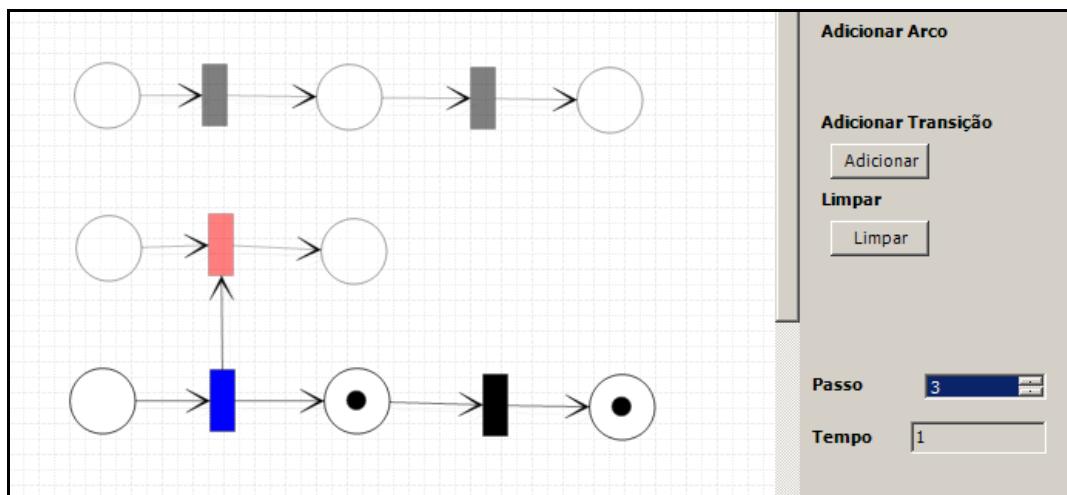
**Figura 5.63 - Não houve disparo de 'transições' na “Estação B”, no entanto, o valor de Tempo foi atualizado para 1**

```

Token: ; 0 ; 0 ; 1 ; 0
Waiting for a connection...
Token: 192.168.73.1:13500 ; 2 ; 1 ; umPasso ; 0
Avaliando rede 2
Rede 2 sem transições prontas para iniciar.(Deadlock 2)
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; 1 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 2 em 1
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...

```

**Figura 5.64 - Mensagens exibidas na "Estação B" após o terceiro passo**



**Figura 5.65 - Disparo da 'transição' temporizada na “Estação C” e atualização do Tempo para valor 1**

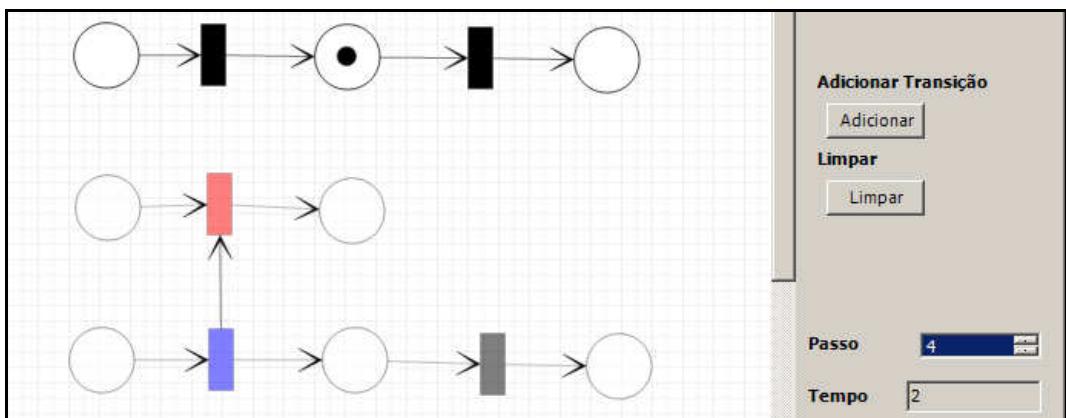
```

Token: 192.168.73.1:13500 ; 2 ; 1 ; umPasso ; 0
Avaliando rede 3
Rede 3 sem transições instantâneas prontas para iniciar.(Deadlock 1)
Transição transition8 pronta para disparar.
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; 1 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 3 em 1
Transição transition8 disparada.
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...

```

**Figura 5.66 - Mensagens exibidas na "Estação C" após o disparo da 'transição' temporizada**

#### PASSO 4:



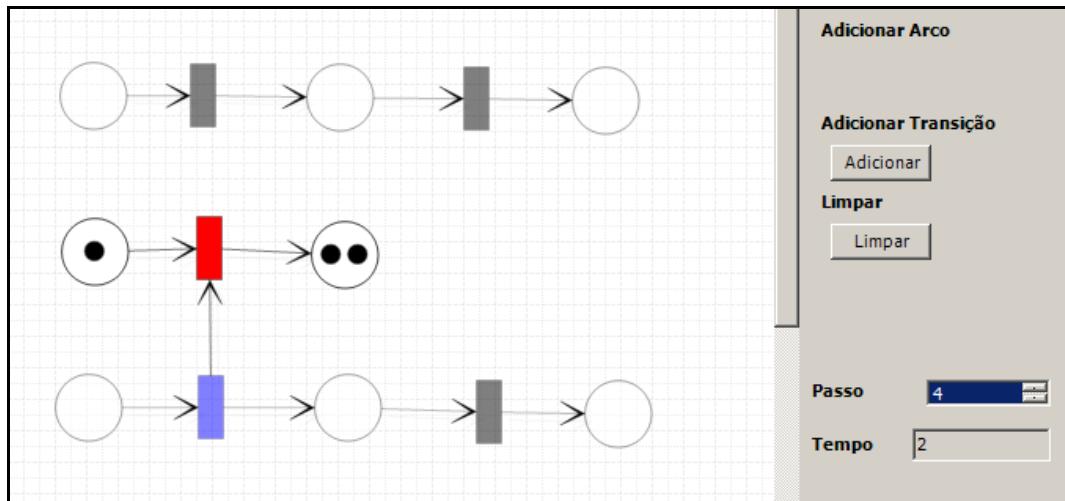
**Figura 5.67 - Disparo da 'transição' temporizada da “Estação A” e atualização de valor Tempo de 1 para 2**

```

Sent
Avaliando rede 1
Rede 1 sem transições instantâneas prontas para iniciar.(Deadlock 1)
Transição transition12 pronta para disparar.
Sent
Token: 192.168.73.1:13500 ; 1 ; 1 ; umPasso ; 0
Sent
Atualizando rede 1 em 1
Transição transition12 disparada.
Waiting for a connection...
Token: 192.168.73.1:13500 ; 1 ; 2 ; umPasso ; 0
Todas estações em avançaram um passo.
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Simulação Concluída.
Waiting for a connection...

```

**Figura 5.68 - Mensagens exibidas na "Estação C" após o disparo da 'transição' temporizada**

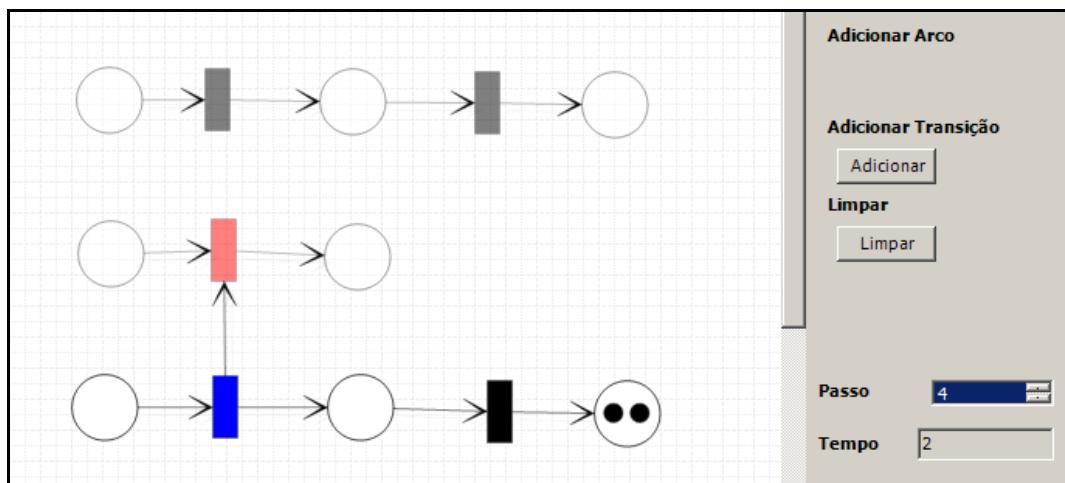


**Figura 5.69 - Não houve disparo de 'transições' na “Estação B”, no entanto, o valor de Tempo foi atualizado de 1 para 2**

```

Token: ; 0 ; 0 ; 1 ; 0
Waiting for a connection...
Token: 192.168.73.1:13500 ; 1 ; 1 ; umPasso ; 0
Avaliando rede 2
Rede 2 sem transições prontas para iniciar.(Deadlock 2)
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; 1 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 2 em 1
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...
  
```

**Figura 5.70 - Mensagens exibidas na "Estação B" após o quarto passo**



**Figura 5.71 – Segundo disparo da 'transição' temporizada da “Estação C” e atualização do valor Tempo de 1 para 2**

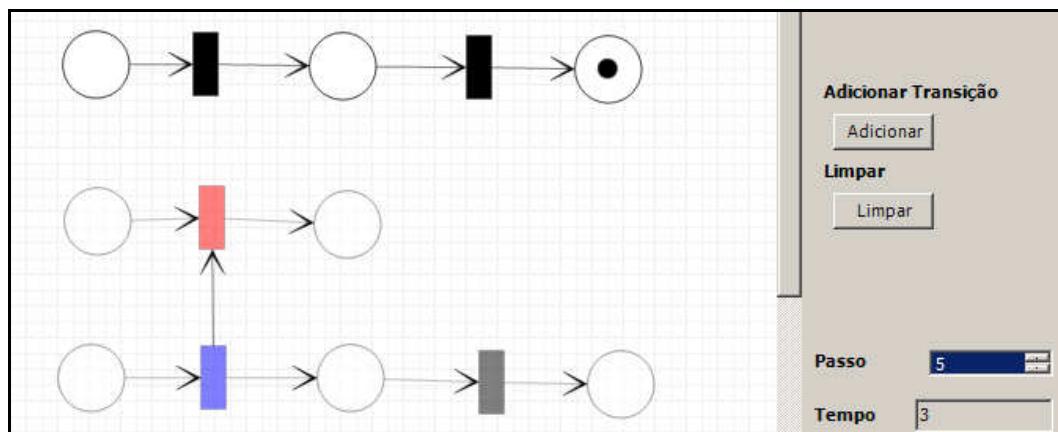
```

Token: 192.168.73.1:13500 ; 1 ; 1 ; umPasso ; 0
Avaliando rede 3
Rede 3 sem transições instantâneas prontas para iniciar.(Deadlock 1)
Transição transition8 pronta para disparar.
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; 1 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 3 em 1
Transição transition8 disparada.
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...

```

**Figura 5.72 - Mensagens exibidas na "Estação C" após o segundo disparo da 'transição' temporizada**

## PASSO 5:



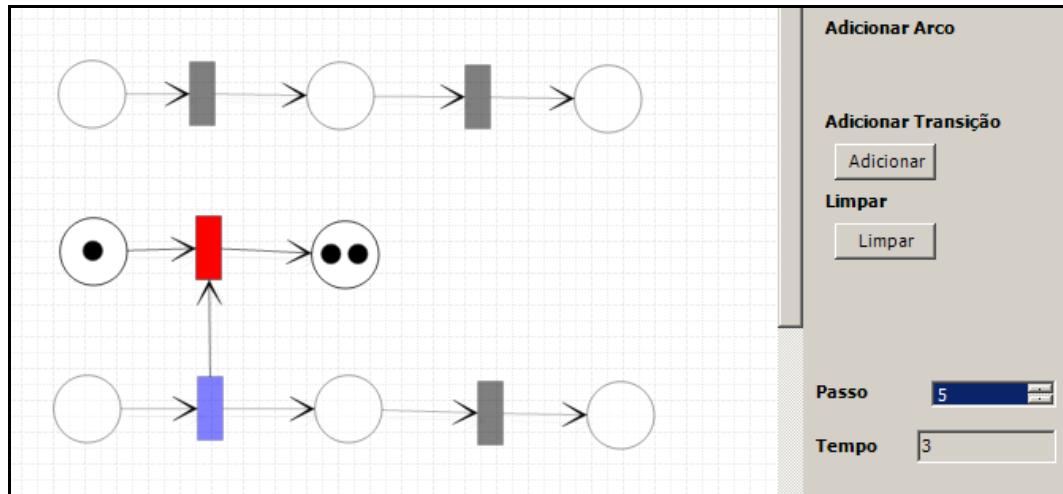
**Figura 5.73 - Disparo da 'transição' temporizada da “Estação A” e atualização do valor Tempo de 2 para 3**

```

Sent
Avaliando rede 1
Rede 1 sem transições instantâneas prontas para iniciar.(Deadlock 1)
Transição transition11 pronta para disparar.
Sent
Token: 192.168.73.1:13500 ; 1 ; 1 ; umPasso ; 0
Sent
Atualizando rede 1 em 1
Transição transition11 disparada.
Waiting for a connection...
Token: 192.168.73.1:13500 ; 1 ; 2 ; umPasso ; 0
Todas estações em avançaram um passo.
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Simulação Concluída.
Waiting for a connection...

```

**Figura 5.74 - Mensagens exibidas na "Estação A" após o disparo da 'transição' temporizada**

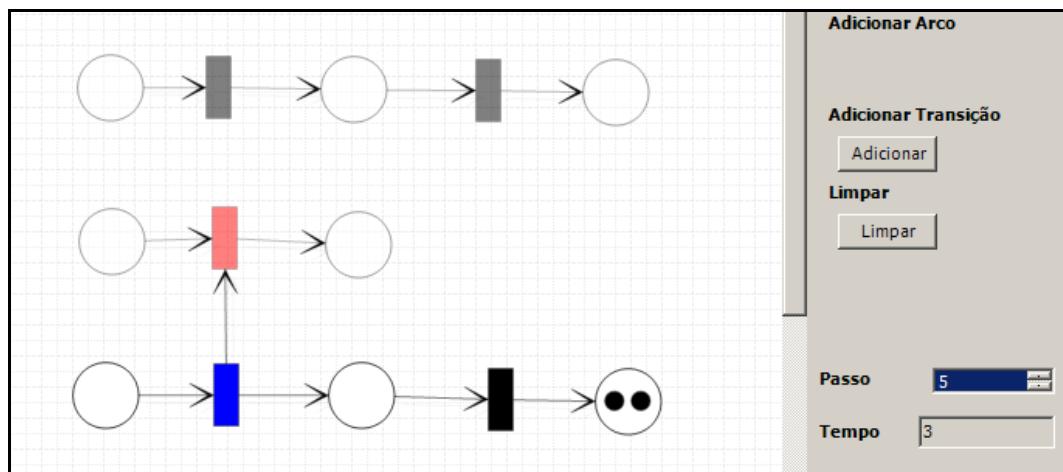


**Figura 5.75 - Não houve disparo de ‘transições’ na “Estação B”, no entanto, o valor Tempo foi atualizado de 2 para 3**

```

Token: ; 0 ; 0 ; 1 ; 0
Waiting for a connection...
Token: 192.168.73.1:13500 ; 1 ; 1 ; umPasso ; 0
Avaliando rede 2
Rede 2 sem transições prontas para iniciar.(Deadlock 2)
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; 1 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 2 em 1
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...
  
```

**Figura 5.76 - Mensagens exibidas na “Estação B” após o quinto passo**



**Figura 5.77 - Não houve disparo de ‘transições’ na “Estação C”, no entanto, o valor Tempo foi atualizado de 2 para 3**

```

Token: 192.168.73.1:13500 ; 1 ; 1 ; umPasso ; 0
Avaliando rede 3
Rede 3 sem transições prontas para iniciar.(Deadlock 2)
Sent
Waiting for a connection...
Token: 192.168.73.1:13500 ; 1 ; 2 ; umPasso ; 0
Estação remota atualizando rede.
Sent
Atualizando rede 3 em 1
Waiting for a connection...
Token: 192.168.73.1:13500 ; ; ; recarregar ; 0
Sent
Waiting for a connection...

```

**Figura 5.78 - Mensagens exibidas na "Estação C" após o quinto passo**

### FIM DA SIMULAÇÃO:

Na tentativa de execução de mais um passo, o sistema detecta que não há mais ‘transições’ (instantâneas e temporizadas) ‘disparáveis’ indicando, portanto, o fim da simulação.

```

Sent
Avaliando rede 1
Rede 1 sem transições prontas para iniciar.(Deadlock 2)
Sent
Token: 192.168.73.1:13500 ; 0 ; 3 ; umPasso ; 0
Todas estações em deadlock 2
Waiting for a connection...

```

**Figura 5.79 - Mensagens exibidas na "Estação A" após tentativa de execução de mais um passo**

```

Token: ; 0 ; 0 ; 1 ; 0
Waiting for a connection...
Token: 192.168.73.1:13500 ; 0 ; 3 ; umPasso ; 0
Avaliando rede 2
Rede 2 sem transições prontas para iniciar.(Deadlock 2)
Sent
Waiting for a connection...

```

**Figura 5.80 - Mensagens exibidas na "Estação B" após tentativa de execução de mais um passo**

```

Token: 192.168.73.1:13500 ; 0 ; 3 ; umPasso ; 0
Avaliando rede 3
Rede 3 sem transições prontas para iniciar.(Deadlock 2)
Sent
Waiting for a connection...

```

**Figura 5.81 - Mensagens exibidas na "Estação C" após tentativa de execução de mais um passo**

## 6 Conclusões

Neste trabalho foi implementado um sistema computacional que permite a criação, edição e simulação de modelos em RdP. Os tipos de RdP que essa ferramenta suporta são as RdP Condição-Evento e Lugar-Transição.

As simulações dos modelos em RdP podem ser realizadas tanto de forma isolada, quanto de forma distribuída. Com isso, torna-se possível descrever e analisar sistemas produtivos, uma vez que a dinâmica dos processos desses sistemas é governada pela ocorrência de eventos que são considerados instantâneos, ou seja, eventos discretos.

Quanto à interface de usuário, é possível evoluí-la tornando-a mais acessível, de modo a permitir que o usuário apenas arraste o componente da RdP, como ‘arco’, ‘transição’ e ‘lugar’, para o projeto que está sendo modelado, ao invés de clicar (para a criação do objeto) e depois arrastar o objeto para o lugar desejado. Para isso, um estudo mais aprofundado do componente de desenvolvimento de interface gráfica utilizado neste trabalho poderia ser uma solução para que possa ser implementada este tipo de funcionalidade.

Por fim, a modelagem e simulação de modelos em RdP de outros tipos, como as RdP orientada a objetos pode ser estudo para implementações futuras, uma vez que o uso do conceito de orientação a objetos possui vantagens tais como citados em JUNQUEIRA (2006): aumento da capacidade de se abstrair problemas, maior facilidade de reutilização de modelos, melhor suporte para confiança e segurança.

## Referências Bibliográficas

- ARATA, W. M. **Análise quantitativa de sistemas de manufatura: abordagem baseada em redes GSPN (“Generalized Stochastic Petri Nets”)**. 1994. 125p. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo. São Paulo, 1994.
- ARATA, W. M. **Representação computacional de Sistemas a Eventos Discretos considerando a heterogeneidade e a integração dos modelos**. 2005. 188p. Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo. São Paulo, 2005.
- BEZERRA, E., **Princípios de análise e projeto de sistemas com UML**. Rio de Janeiro. Editora Campus, 2007.
- BOOCH, G., RUMBAUGH, J., JACOBSON, I. **The Unified Modeling Language User Guide**. Addison Wesley Longman, Inc., 2005.
- COSTA, M. C. B. da. **Modelagem e simulação de protocolos de comunicação**. 1995. 136p. Dissertação (Mestrado) – Faculdade de Engenharia Elétrica – FEE, Universidade Estadual de Campinas – UNICAMP. Campinas, 1995.
- CRAINIMATE SOFTWARE, **ERM Diagram**. 2010. Disponível em: <<http://www.crainiate.net/products/erm4/>>. Acessado em: 19 de out. de 2010.
- FREITAS, R. G. de, SILVA, V. A. **Ferramenta baseada em rede de petri para modelagem, simulação, programação e supervisão de sistemas de automação**. 2006. 76p. Trabalho apresentado à Escola Politécnica da USP para obtenção do título de Engenheiro. São Paulo, 2006.
- FUJIMOTO, R. M. *Parallel and distributed simulation*. In. **Proceedings of the 1999 Winter Simulation Conference**, p. 122-131, 1999.
- JUNQUEIRA, F. **Modelagem de sistemas flexíveis de movimentação de materiais através de redes de Petri interpretadas**. 2001. 128p. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo. São Paulo, 2001.

- JUNQUEIRA, F. **Modelagem e simulação distribuída de sistemas produtivos.** 2006. 206p. Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo. São Paulo, 2006.
- MURATA, T. *Petri Nets – Properties, Analysis And Applications.* In: **Proceedings of the IEEE**, Vol. 77, No. 4, 1989.
- OMG. **OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2.** 2007. 722p. Disponível em: <<http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF/>>. Acesso em: 1 de jul. de 2010.
- PETERSON, J. L. *Petri Nets. Computing Surveys*, Vol. 9, No. 3, p. 223-252, 1977.
- SIBERTIN-BLANC, C. *A Client-Server Protocol for the Composition of Petri Nets.* In: **Proceedings 14th International Conference on Application and Theory of Petri Nets.** Chicago, Illinois, USA, p.377-396, 1993.
- W3C. **Extensible Markup Language.** 2009. Disponível em: <<http://www.w3.org/XML/>>. Acesso em: 6 de jul. de 2009.
- WEBER, M. **Petri Net Markup Language.** 2006. Disponível em: <<http://www2.informatik.hu-berlin.de/top/pnml/about.html>>. Acesso em: 3 de jun. de 2009.