

DANIELA HOFER SANTOS
LEONARDO MONTEIRO MAZZARIOL

IMPLEMENTAÇÃO DE UM AMBIENTE PARA A SIMULAÇÃO
DISTRIBUÍDA DE SISTEMAS PRODUTIVOS

São Paulo

2007

DANIELA HOFER SANTOS
LEONARDO MONTEIRO MAZZARIOL

IMPLEMENTAÇÃO DE UM AMBIENTE PARA A SIMULAÇÃO
DISTRIBUÍDA DE SISTEMAS PRODUTIVOS

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

Curso de Graduação:
Engenharia Mecatrônica

Orientador:
Prof. Dr. Paulo Eigi Miyagi

São Paulo
2007

FICHA CATALOGRÁFICA

Santos, Daniela Hofer

Implementação de um ambiente para simulação distribuída de sistemas / D.H. Santos, L.M. Mazzariol. -- São Paulo, 2007. 94 p.

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

1.Redes de Petri 2.Redes de computadores 3.Simulação de sistemas I.Mazzariol, Leonardo Monteiro II.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos III.t.

DEDICATÓRIA

Dedicamos este trabalho àqueles que nos ampararam em todas as etapas da vida, sob todas as circunstâncias, incansavelmente.

AGRADECIMENTOS

Ao nosso orientador, Prof. Dr. Paulo Eigi Miyagi, pelas conversas que propiciaram uma visão mais ampla da engenharia profissional, bem como pelas inúmeras sugestões que colaboraram para a execução desse trabalho.

Ao Paulo Flávio Bertechini Gomes, pela grande consideração na fase final e apresentação.

Aos professores da Escola Politécnica da USP que nos motivaram e inspiraram ao longo desses 5 anos de convivência e aprendizado.

Aos meus pais, Cícero e Isabel, pelo constante apoio e carinho desde sempre e, especialmente, nesses últimos 5 anos. Obrigada pelos “colinhos” oferecidos e todos os outros pequenos momentos.

Ao Leo, pelo constante carinho, compreensão e cuidado demonstrados durante a execução desse trabalho. Esta experiência só fortaleceu nossa relação.

Aos amigos que compreenderam as ausências e a todos que contribuíram para que este trabalho chegasse ao fim.

(Daniela)

Aos meus pais, Aline e Alfredino, pelo constante aprendizado e vivências. Poucas histórias seriam tão completas quanto àquelas de anbos. Principalmente à minha mãe por ter atendido as ordens/pedidos de café fresco.

À Dani, por tudo o que é possível lembrar e por enxerga sempre alguém além de mim.

Ao Érico que continuou a me convidar para distrações mesmo após repetidas negativas.

Por fim, à vida pela ironia, exemplos bons e ruins e conhecimento infindável.

(Leonardo)

"Eu não sei quase nada, mas desconfio de
muita coisa".

(João Guimarães Rosa)

RESUMO

Os sistemas produtivos têm evoluído de modo a atender os novos padrões de custo e qualidade demandados pelos mercados. A distribuição de processos produtivos por diferentes plantas industriais, assim como automação de processos, surge como parte da solução para este desafio. Além disso, através dos avanços em telecomunicações e supervisão remota, instalações produtivas em localidades distintas podem ser interligadas, tal que atuem de modo sincronizado e cooperativo assegurando produtividade e qualidade do produto final. Nesse contexto, a modelagem de sistemas e a simulação distribuída são fundamentais para o projeto e verificação da viabilidade de futuras instalações ou aprimoramento das já existentes. Assim, o objetivo deste trabalho é a implementação de um ambiente para a simulação de modelos de sistemas produtivos em rede de Petri distribuídas em diferentes computadores. Considera-se que a comunicação entre os modelos é realizada através de LAN/WAN e processos específicos que asseguram a sincronização dos eventos. A verificação e validação do ambiente são realizadas a partir de testes envolvendo sistemas virtuais.

ABSTRACT

Production systems have evolved in order to reach new costs and quality levels demanded by markets. The distribution of productive processes by different industrial plants, as well as specialization focused in process automation, appears as part of the solution to attend these requirements. With telecommunication and remote supervised development, productive plants in distinct location can be linked, such that they can act in synchronized and cooperative way, assuring productivity and quality to the final product. In this subject, modeling distributed systems and their distributed simulation are essential for design and viability verification of future plants or improvement of existing ones. Thus, the purpose of this work is the implementation of an environment (software) for simulation of production system modeled in Petri net, distributed in different workstations. LAN/WAN structure is considered and so are specific processes that guarantee events synchronization. The environment verification and validation will be carried through from tests involving virtual systems.

LISTA DE FIGURAS

Fig. 2.1 – Gráfico de evolução de estados característica de um sistema a eventos discretos (CURY, 2001).	4
Fig. 2.2 – Representação dos nós de uma rede de Petri.....	6
Fig. 2.3 - Exemplo de rede de Petri	6
Fig. 2.4 - Exemplo de rede de Petri Marcada	7
Fig. 2.5 - Matrizes Pré e Pos e M_0 da rede da Fig. 2.4	8
Fig. 2.6 - Representação gráfica de uma RdP.....	10
Fig. 2.7 - Notação em PNML de lugar (JÜNGEL et al., 2000).....	11
Fig. 2.8 - Notação em PNML de transição (JÜNGEL et al., 2000).....	11
Fig. 2.9 - Notação em PNML de arco (JÜNGEL et al., 2000).....	12
Fig. 2.10 - Exemplo de funcionamento do mecanismo de token ring (Estação A começa enviando uma mensagem – token – e B só poderá enviar a sua mensagem assim que o token enviado por A completar a sua circulação pelo anel).....	15
Fig. 3.1 – Algoritmo da EnviaRecebeChamada de Método.....	18
Fig. 3.2 – Algoritmo da VerificaChamadaMétodo.....	19
Fig. 3.3 – Circulação de chamada de método.....	20
Fig. 3.4 - Gerenciamento das Mensagens (Controle Token ring).....	22
Fig. 3.5 – Fluxograma da sequência de simulação (Freitas 2006 modificado).	24
Fig. 3.6 – Janela Inicial do AS.	28
Fig. 3.7 – Status de estação Iniciada e a qual endereço está conectada.....	29
Fig. 3.8 – Modelo para simulação distribuída (JUNQUEIRA 2006).	29
Fig. 3.9 – Janela de Edição	30
Fig. 3.10 - Inclusão de Texto no Modelo	30
Fig. 3.11 - Inserção de Lugares e Transições	31
Fig. 3.12 - Inserção dos Arcos Orientados.....	31
Fig. 3.13 - Modelo Gráfico Obtido.....	32
Fig. 3.14 - Gravação do Arquivo XML	32
Fig. 3.15 - Arquivo em XML	33
Fig. 3.16 – Partes em destaque do arquivo devem ser modificadas	33
Fig. 3.17 - Partes inseridas e modificadas em destaque	34
Fig. 3.18 – Modelo final da estação “B”	35
Fig. 3.19 – Campos para dados de comunicação.....	35

Fig. 3.20 – Anel lógico com endereços de 3 estações em 2 computadores	36
Fig. 4.1 – Estado inicial da estação “A”	38
Fig. 4.2 – Disparo de T0.....	38
Fig. 4.3 – Disparo de T2.....	39
Fig. 4.4 – Disparo de T3.....	39
Fig. 4.5 – Deadlock e evolução de tempo para $t=5$	40
Fig. 4.6 – Fim da Simulação na Estação A	40
Fig. 4.7 - Estado inicial da estação “B”	41
Fig. 4.8 – Estação negociando evolução do tempo	41
Fig. 4.9 – Evolução do tempo para $t=3$	42
Fig. 4.10 – Disparo de T0.....	42
Fig. 4.11 – Disparo de T1_B/T1_C.....	43
Fig. 4.12 – Fim da Simulação na Estação B	43
Fig. 4.13 – Estado inicial da estação “C”	44
Fig. 4.14 – Disparo de T0.....	44
Fig. 4.15 – Evolução do tempo para $t=3$	45
Fig. 4.16 – Evolução do tempo para $t=5$	45
Fig. 4.17 - Disparo de T1_C/T1_B.....	46
Fig. 4.18 – Fim da Simulação na Estação C	47
Fig. 4.19 – Rede modelada no HPSim.....	48
Fig. 4.20 – Estação A modelada no AS	49
Fig. 4.21 – Estação B modelada no AS.....	49
Fig. 4.22 – Estação C modelada no AS.....	49

LISTA DE TABELAS

Tabela 2.1 – Elementos PNML (Adaptado de FREITAS 2006)	10
Tabela 3.1 - Uma sequência de chamadas de método.	19
Tabela 3.2 – Composição de um <code>token</code>	20
Tabela 3.3 – Possíveis valores para o campo de <i>status</i> (JUNQUEIRA 2006).	21
Tabela 3.4 – Atributos modificados da <i>tag</i> “transition”.	23
Tabela 3.5 – Atributos Específicos criados para a <i>tag</i> “transition”.	23
Tabela 3.6 - Valores possíveis para o parâmetro <i>deadlock</i>	25
Tabela 3.7 - Correspondência entre nomes e variáveis.	27
Tabela 4.1 – Comparação da ordem de disparos de transições com avanço do tempo	50

LISTA DE PALAVRAS RESERVADAS

Redes de Petri (Fonte: Arial 12 pontos)

- Arco(s) (Orientado(s), Habilitador (es) ou inibidor(es);
- Capacidade(s);
- Disparo(s);
- Lugar(es);
- Marca(s);
- Marcação;
- Marcada(s);
- Pré (pós) condição (ões)
- Transição (ões) (habilitada(s), instantânea(s), disparável(is)).

Token ring (Fonte: Courier New 12 pontos)

- Anel;
- Estação (ões) ;
- Mestre;
- Token (ring) ;

Comunicação (Fonte: Times New Roman 12 pontos sublinhada)

- Servidor(es);
- Cliente(s);
- Socket(s);

SUMÁRIO

1 INTRODUÇÃO	1
2 CONCEITOS FUNDAMENTAIS	3
2.1 Sistema a Eventos Discretos	3
2.2 Ambiente de Simulação (AS).....	5
2.3 Rede de Petri	5
2.4 Java <i>Server/Client</i>	12
2.5 Simulação Distribuída.....	13
2.6 Ferramenta para Modelagem e Simulação de Sistemas.....	15
3 IMPLEMENTAÇÃO DO AS.....	16
3.1 Especificações básicas de um AS	16
3.2 Suportes considerados	16
3.3 Estrutura do AS	17
4 EXEMPLO DE APLICAÇÃO.....	38
4.1 Estudo de Caso 1	38
4.2 Estudo de Caso 2	48
5 CONCLUSÕES	51
ANEXO A – ALGORITMO DE VERIFICAÇÃO/ AVALIAÇÃO DOS PARÂMETROS LOCAIS	52
ANEXO B – CÓDIGOS FONTE.....	53
B.1. TrataTokenRecebe().....	53
B.2. TrataTokenEnvia()	55
B.3. ControleToken()	57
B.4. verificaChamadaDeMetodo	58
B.5. EnviaRecebeChamada.....	59
B.6. ExecutaRede.passo()	60
ANEXO C – CÓDIGOS EM XML (MODIFICADOS).....	61
C.1. Estação A do Estudo de Caso 1 - EstA.xml.....	61

C.2.Estação B do Estudo de Caso 1 – EstB.xml	62
C.3.Estação C do Estudo de Caso 1 – EstC.xml	63
C.4.Estação A – Estudo de Caso 2	64
C.5.Estação B – Estudo de Caso 2	66
C.6.Estação C – Estudo de Caso 2	67
ANEXO D – ARQUIVOS DE RESULTADOS	68
D.1.Estação A – Estudo de Caso 2	68
D.2.Estação B – Estudo de Caso 2	79
D.3.Estação C – Estudo de Caso 2	86
REFERÊNCIAS BIBLIOGRÁFICAS.....	92

1 Introdução

A distribuição de operações produtivas por diferentes plantas industriais, assim como especialização de tarefas voltada à automação de processos, surge como forma de atender as exigências cada vez mais rigorosas de maior qualidade e menor custo demandados por diferentes mercados. Atualmente, são muitos os exemplos de empresas com plantas produtivas dispersas geograficamente (ABB, 2007; TOYOTA, 2007). Com esta estrutura, as empresas procuram explorar estrategicamente as vantagens locais inerentes de cada planta e seus recursos e facilidades. Assim, instalações produtivas em localidades distintas são interligadas tal que atuem de modo sincronizado e cooperativo assegurando produtividade e qualidade do produto final.

Estes fatos têm assim demandado propostas de novas abordagens e técnicas de modelagem e análise para concepção, integração e avaliação destes sistemas, pois empresas que possuem soluções efetivas na área de integração desses sistemas terão vantagem sobre seus concorrentes, podendo se voltar ao atendimento das necessidades de seus clientes e à introdução de produtos inovadores (CHENG, 2006).

Durante as últimas décadas a evolução na tecnologia de informação, mecatrônica e comunicação trouxe grandes mudanças nos mercados e organizações em todo o mundo. Essas mudanças induziram o desenvolvimento de conceitos de organizações virtuais, integração de processos, cadeias de abastecimento centradas no consumidor e comércio eletrônico (e-commerce) (MONOSTORI, 2006).

Com a modelagem de plantas existentes e em operação, podem ser analisados e testados processos alternativos de produção e com isso, detectar problemas, como gargalos (pontos críticos onde o fluxo de produção encontra alguma dificuldade em seguir adiante), e as suas possíveis soluções (auxiliando inclusive na escolha da melhor dentre várias possibilidades). A modelagem também é utilizada para plantas ainda em fase de projeto, para prever possíveis situações não planejadas, e assim otimizar a planta antes mesmo de sua operação (MIYAGI, 2006). A grande variedade de classes de redes de Petri (RdP) permite inúmeras formas de se modelar as características desejadas de determinado processo, sendo necessário apenas definir o escopo e o detalhamento almejado para escolher o tipo adequado de RdP (GOMES, 2005). Destarte, a simulação distribuída é uma ferramenta fundamental para o projeto e verificação da viabilidade de futuras instalações ou aprimoramento das já existentes (JUNQUEIRA, 2006).

Assim, este projeto visa a implementação de um ambiente para o apoio à análise de modelos de sistemas produtivos em rede de Petri (RdP) distribuídas em diferentes computadores. O ambiente aqui envolve hardware e software adequados, isto é, considera-se que num sistema produtivo distribuído tem-se vários computadores, cada um com softwares e banco de dados sobre o módulos e recursos adequados para análise e simulação de seus processos produtivos.

Em relação aos modelos dos processos produtivos, a RdP é considerada por esta ser uma técnica com recursos adequados para descrever tanto os aspectos estruturais como funcionais. A idéia aqui é que, qualquer que seja o modelo existente de um processo, este sempre pode ser devidamente convertido numa rede de Petri.

Em Junqueira (2006) foi proposto um sistema computacional para modelagem e simulação distribuídas de modelos em rede de Petri e, com base nas funções e regras do ambiente de modelagem e simulação (AMS) foi definido o seguinte escopo para este estudo:

- Uma ferramenta para análise de sistemas produtivos através de simulação de modelos em RdP;
- A simulação de modelos em RdP de forma distribuída com outros AMSs, através do uso da estrutura Servidor/Cliente Java.

Assim sendo, o objetivo do presente trabalho é o desenvolvimento de um Ambiente de Simulação (AS), onde os modelos em RdP distribuídos em diferentes computadores poderão ser executados e analisados, explorando os seguintes conceitos e tecnologias:

- XML como notação dos dados obtidos a partir dos modelos em RdP;
- Java, para a implementação do ambiente (software) em si, com seus objetos e serviços;
- TCP/IP como protocolo de comunicação.

2 Conceitos Fundamentais

2.1 Sistema a Eventos Discretos

Uma definição de “sistemas” se faz necessária, visto a ampla utilização do termo durante todo o trabalho. Algumas das definições encontradas na literatura são apresentadas a seguir (MIYAGI, 1996):

- Combinação de partes coordenadas entre si para formarem um conjunto unificado;
- Combinação de componentes que agem em conjunto para desempenhar uma função que se torna impossível na ausência de qualquer das partes;
- Uma parte limitada do universo que interage com o mundo externo através das fronteiras que o delimitam.

Como os sistemas produtivos estão se tornando cada vez mais complexos, novas técnicas de modelagem têm sido desenvolvidas para permitir a análise do seu comportamento dinâmico, e apontar eventuais melhoras ou soluções de problemas. Estes sistemas são assim classificados em três categorias (VILLANI, 2004):

- Sistemas a eventos discretos (SEDs) – sistemas cujas variáveis de estado são modificadas apenas com a ocorrência de eventos instantâneos, de duração nula;
- Sistemas de variáveis contínuas – sistemas cujas variáveis de estado variam continuamente no tempo;
- Sistemas híbridos – sistemas que contém variáveis discretas e contínuas.

Como o enfoque aqui está na parte de integração de modelos distribuídos, a visão disto como SED pode ser adotada e lista-se a seguir outras características relevantes desta classe de sistema:

- Não são descritíveis por equações diferenciais;
- Suas variáveis de estado são “binárias” – ou “acontecem” ou “não acontecem”;
- Caracterizam-se melhor pela evolução baseada na ocorrência de eventos instantâneos do que em função do tempo.

Conforme ilustrado na Fig. 2.1, a ocorrência de um evento (representados por α , β e λ) causa uma **transição** ou mudança de estado no sistema, de forma que sua evolução no tempo pode ser representada pela trajetória percorrida no seu espaço de estados (representados por x_1 , x_2 , x_3 e x_4).

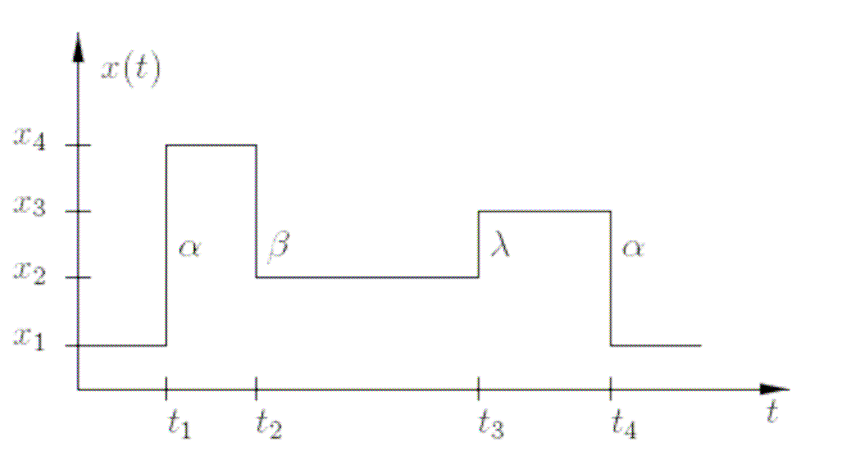


Fig. 2.1 – Gráfico de evolução de estados característica de um sistema a eventos discretos (CURY, 2001).

Os SEDs também podem ser ditos instantâneos, quando sua saída depende apenas da entrada presente (sistema sem memória), ou dinâmicos, quando sua saída depende da entrada e dos estados passados (sistema com memória) (MIYAGI, 1996).

A forma de análise dos sistemas depende de qual tipo de estudo ou informações se deseja avaliar. Para os SEDs, a simulação tem sido considerada a solução mais adequada, diferentemente do que ocorre em sistemas contínuos, onde se tem utilizado técnicas analíticas relativamente precisas para tratar os problemas. Como os modelos de simulação de SEDs do mundo real são relativamente complexos e a quantidade de informações manipuladas é muito grande, utilizam-se computadores para executar a sua simulação. Uma das muitas opções de modelagem de SEDs são as chamadas redes de Petri (RdP) (JUNQUEIRA, 2006).

A simulação é, pois, uma forma prática de análise de SEDs, e seus principais componentes (definidos pela dinâmica do sistema e pelo o objeto de interesse) são:

- Entidade – o objeto de interesse;
- Atributo – propriedade desse objeto;
- Atividade – representa uma ação dentro do sistema;
- Evento – ocorrência que altera o estado do sistema;
- Estado – representa uma situação do sistema.

2.2 Ambiente de Simulação (AS)

Um AS é um software onde o usuário realiza simulações. Nesse trabalho, essas simulações envolvem modelos em Rede de Petri (uma das formas de modelagem de sistemas produtivos existentes, que será discutida em mais detalhes na seção 2.3).

Além disso, essa simulação pode tratar modelos em diferentes computadores, comportando-se como se fossem uma única simulação, constituindo a chamada simulação distribuída.

2.3 Rede de Petri

O tipo mais simples de rede de Petri (RdP), chamado de RdP ordinária, foi criado por Carl Adam Petri em 1962 em sua tese de doutorado. É uma modelagem que permite a representação de características intrínsecas de funcionamento de um sistema, tais como:

- Comunicação;
- Sincronismo;
- Paralelismo;
- Concorrência.

A representação de SEDs por RdP se inicia sempre pela definição dos componentes básicos do sistema, que podem ser divididos em:

- Ativos: são capazes de produzir, transportar ou alterar itens;
- Passivos: podem armazenar e/ou tornar visíveis os itens.

O relacionamento destes componentes no modelo é indicado através de **arcos-orientados**, que podem representar relações, tais como: conexões lógicas, direito de acesso, causa e efeito.

Na modelagem das características dinâmicas do sistema, **marcas** são distribuídas nos elementos passivos para definir o estado inicial, as condições para que o processo ocorra e a evolução deste.

Uma RdP pode ser representada por um grafo com dois tipos de nós: **lugares** e **transições** (Fig. 2.2). Lugares e transições são ligados por **arcos orientados**, de modo que um tipo de nó só pode estar diretamente ligado ao outro tipo de nó. Assim, é chamado de

grafo bipartido, ou seja, um grafo constituído por dois conjuntos de nós, onde nós de mesmo conjunto **não** podem estar diretamente interligados. Assim, arcos orientados ligam lugares a transições e vice-versa.

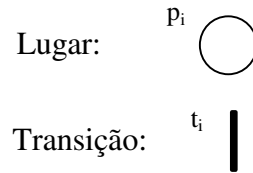


Fig. 2.2 – Representação dos nós de uma rede de Petri

Um exemplo de RdP é apresentado na Fig. 2.3:

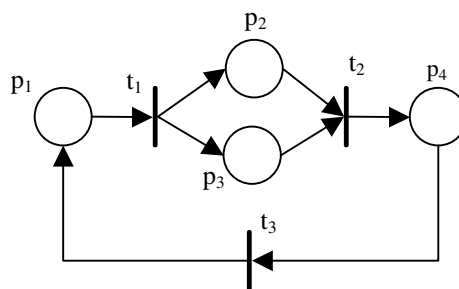


Fig. 2.3 - Exemplo de rede de Petri

O estado de um SED pode ser representado através de uma RdP marcada. Uma marca é representada por um ponto negro colocado num determinado lugar.

Segundo Junqueira (2006), a evolução dos estados desse sistema é baseada em regras que definem as condições para a ocorrência de eventos e o resultado dessas ocorrências. A seguir é apresentada a definição formal das RdP.

Definição: (Cardoso; Valette, 1997 apud VILLANI 2004) – Uma *rede de Petri marcada* é um par $N = \langle R, M_0 \rangle$, onde:

- R é uma rede de Petri definida pela 4-tupla $\langle P, T, Pre, Pos \rangle$, onde:
 - $P = \{p_1, p_2, p_3, \dots, p_m\}$ é um conjunto finito de lugares;
 - $T = \{t_1, t_2, t_3, \dots, t_n\}$ é um conjunto finito de transições;
 - $P \cap T = \emptyset, P \cup T \neq \emptyset$;
 - $Pre: P \times T \rightarrow \mathbb{N}$ define os arcos de entrada das transições (\mathbb{N} é o conjunto de números naturais);
 - $Pos: T \times P \rightarrow \mathbb{N}$ define os arcos de saída das transições.
 - $M_0: P \rightarrow \mathbb{N}$ é a marcação inicial da rede.

A estrutura da rede de Petri pode ser representada por um conjunto de matrizes, onde as colunas mapeiam as transições e as linhas, os lugares. A matriz Pre indica os arcos de lugares a transições e a matriz Pos indica os arcos de transições a lugares. A matriz M indica os lugares com marcas e, M_0 é a matriz da marcação inicial da rede. Cada elemento da matriz indica o peso do arco que liga um lugar a uma transição (no grafo, o peso de um arco é assinalado como um tipo de rótulo próximo ao mesmo e é omitido se for igual a um). Caso o valor do elemento da matriz seja 'zero', não há arco entre os nós correspondentes. As representações (gráfica e matricial) são apresentadas na Fig. 2.4 e na Fig. 2.5, respectivamente.

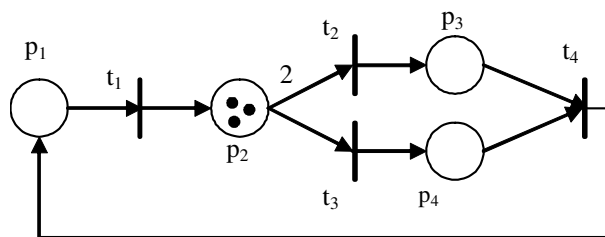


Fig. 2.4 - Exemplo de rede de Petri Marcada

Na representação matricial, o termo $Pre(t_i, p_j)$ pode ser entendido como a quantidade de marcas retiradas do lugar p_j quando do disparo da transição t_i . Assim, quando o termo (t_1, p_1) é igual a '1', o disparo da transição em questão consome uma marca de p_1 . No caso

da matriz Pos, ao invés de marcas a retirar de um lugar, o valor indica o número de marcas a serem colocadas no lugar em função do disparo da transição correspondente.

$$\text{Pre} = \begin{matrix} & \begin{matrix} t_1 & t_2 & t_3 & t_4 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix} \quad \text{Pos} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad M_0 = \begin{bmatrix} 0 \\ 3 \\ 0 \\ 0 \end{bmatrix}$$

Fig. 2.5 - Matrizes Pré e Pos e M_0 da rede da Fig. 2.4

A evolução de uma RdP marcada resulta do disparo das transições habilitadas. Uma transição t_j está habilitada quando suas pré-condições¹ são satisfeitas, ou seja, quando os lugares à sua entrada possuem um número maior ou igual ao peso dos arcos que fazem a ligação àquela transição. Os lugares p_j , devem assim possuir um número de marcas $M(p_j) \geq \text{Pre}(p_j, t_i)$.

Quando ocorre o disparo de uma transição t_i , são retiradas $\text{Pre}(p_j, t_i)$ marcas de cada lugar de entrada, e são colocadas, em cada lugar p_k de saída da transição, $\text{Pos}(p_k, t_i)$ marcas.

Com a necessidade de representação de modelos mais complexos, e com diferentes características, foram desenvolvidas variações da RdP, para suprir necessidades de novos atributos, relações de tempo e eventos estocásticos, mas mantendo o conceito inicial. Apresenta-se a seguir alguns tipos de RdP com suas características principais (MIYAGI, 1996; JUNQUEIRA, 2006):

- RdP ordinárias - os lugares possuem capacidade ilimitada para armazenar as marcas.
- RdP Condição-Evento (RdP-CE) – os lugares possuem marcação binária, ou seja, só podem possuir uma ou nenhuma marca.

¹ Isto se refere a uma RdP ordinária. Para RdP Condição-Evento e RdP Lugar-Transição, é necessário também avaliar as pós-condições.

- RdP Lugar-Transição (RdP-LT) – os lugares podem possuir mais de uma marca (porém, com capacidade limitada), e os arcos são ponderados, i.e., possuem 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 Colorida – possui marcas individualizadas, o que permite que as transições ocorram de forma diferenciada, dependendo do tipo de marca considerada.
- RdP Orientada a Objetos – inclui recursos para descrever propriedades e entidades que ocorrem na abordagem orientada a objetos.

2.3.1 Linguagem PNML

A *Petri Net Markup Language* (PNML) é uma linguagem baseada em XML² que define uma RdP enumerando todos os seus elementos básicos, ou seja, os lugares, transições e arcos.

PNML é uma linguagem comum e independente das ferramentas, e foi concebida como um formato universal para o intercâmbio das informações contidas em uma RdP (POST & WERF, 2004).

Entre as principais vantagens dessa notação para as RdP estão a flexibilidade (é possível representar vários tipos de RdP), compatibilidade (permite a definição de diferentes *tags* para caracterizar diferentes características da rede) e a ausência de ambigüidades entre modelos e tipos de RdP (pois cada modelo e cada tipo traz informações diferentes entre si) (FREITAS, 2006).

² XML (*eXtensible Markup Language*) é uma notação baseada em *tags* para identificar, categorizar e organizar as informações em arquivos de dados (DYKES et al., 2005 apud FREITAS, 2006).

Os elementos constituintes da notação PNML, *tags*, estão relacionados na Tabela 2.1:

Tabela 2.1 – Elementos PNML (Adaptado de FREITAS 2006)

Tipo	Tag	Atributo XML	Descrição
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
<i>Offset</i>	<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

Nas Figuras Fig. 2.7, Fig. 2.8 e Fig. 2.9 têm-se um exemplo da notação PNML para a rede da Fig. 2.6.

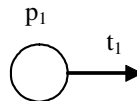


Fig. 2.6 - Representação gráfica de uma RdP


```

<place id="p1">
  <graphics>
    <position x="20" y="40"/>
  </graphics>
  <name>
    <value>ready to produce</value>
    <graphics>
      <offset x="-10" y="10"/>
    </graphics>
  </name>
  <initialMarking>
    <value>P</value>
    <graphics>
      <offset x="-1" y="-1"/>
    </graphics>
  </initialMarking>
</place>

```

Fig. 2.7 - Notação em PNML de lugar (JÜNGEL et al., 2000).

```

<transition id="t1">
  <graphics>
    <position x="20" y="20"/>
  </graphics>
  <toolspecific tool="PN4all" version="0.1">
  </toolspecific>
</transition>

```

Fig. 2.8 - Notação em PNML de transição (JÜNGEL et al., 2000).

Novas *tags* com significados específicos podem ser adicionadas aumentando a quantidade de informação que a linguagem consegue traduzir. Barros & Gomes (2004) propõem uma variação da PNML que implementa o conceito de “fusão de transições”, a ser apresentado no item 2.5.1, criando uma espécie de lista de transições que devem ser interpretadas como uma só durante a simulação.

```

<arc id="a1" source="p1" target="t1">
  <graphics>
    <position x="10" y="30"/>
    <position x="10" y="10"/>
  </graphics>
  <inscription>
    <value>x</value>
    <graphics>
      <offset x="-6" y="-16"/>
    </graphics>
  </inscription>
  <type value="normal"/>
</arc>

```

Fig. 2.9 - Notação em PNML de arco (JÜNGEL et al., 2000).

2.4 Java Server/Client

No Ambiente proposto, as diversas estações devem comunicar-se para garantir que a simulação seja conduzida corretamente, assim é necessário estabelecer uma comunicação entre os processos de software de cada estação.

A comunicação entre esses processos mais utilizada atualmente tem sido a feita por *sockets*. Existem duas alternativas em Java para criação de comunicação via *socket*: o modo orientado a conexão, que funciona sobre o protocolo TCP³, e o modo orientado a mensagem (datagrama), chamado UDP⁴. Ambas utilizam o endereçamento IP⁵ como forma de acessar as estações desejadas (NUNES, 2004).

- A linguagem Java possui amplo suporte e classes desenvolvidas para se construir conexões (*servidores* e *clientes*) TCP/IP. A biblioteca *java.net* possui a classe “*ServerSocket*” e “*Socket*” que servem para efetuar tais conexões (JAVA PLATFORM, 2007).

³ TCP: *Transmission Control Protocol*. Neste caso, o *servidor* escolhe uma “porta” (do inglês, *port*) e aguarda conexões nesta porta. Para conectar-se, o *cliente* deve saber o endereço do *servidor* e a porta disponível e então efetuar a solicitação para conexão.

⁴ UDP: *User Datagram Protocol*. Neste caso, através do endereço e porta, é enviada uma mensagem (datagrama) sem que haja confirmação de comunicação com o destinatário. No caso de não haver disponibilidade do destinatário para recebimento, a mensagem é perdida sem que o remetente seja informado.

⁵ IP: *Internet Protocol* compatível com a versão 4 (4 números de 16bits, por ex., 255.255.255.255).

Assim, as criações de um servidor e um cliente são simplificadas, bastando utilizar os respectivos construtores em Java:

```
ServerSocket servidorSocket = new ServerSocket (Numero_Porta);
Socket clienteSocket = new Socket (endereço_ip, Numero_Porta);
```

Nessas classes são disponibilizadas funções para conexão, envio e recebimento de mensagens etc.

2.5 Simulação Distribuída

A análise de sistemas tem se tornado cada vez mais complicada devido à crescente complexidade, tanto em tamanho quanto em detalhamento desses sistemas, portanto, é fundamental levantar formas alternativas para realizar tais análises.

Junqueira (2006) citou diversas formas de realizar a distribuição do processamento a fim de diminuir o custo dessa tarefa. Também propôs uma forma de implementar a simulação distribuída de modo a assegurar um desempenho computacional adequado. Uma vantagem adicional é a possibilidade de se utilizar, numa simulação, modelos “fechados”⁶ sobre os quais se tem pouca ou nenhuma informação.

2.5.1 Interface entre os modelos

A interface entre os modelos em RdP distribuídos deve ser construída de forma a permitir que estes se comuniquem tal que a simulação possa se desenvolver, bem como descrever corretamente as relações entre os modelos.

Existem diversas formas pelas quais se podem relacionar dois modelos, sendo que alguns autores (BASTIDE, 1995; LAKOS, 1995; WANG, 1998; apud JUNQUEIRA, 2006) utilizam-se da relação entre mais de um tipo de elemento da RdP (lugares e transições) simultaneamente. 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:

⁶ Na modelagem pode se considerar que plantas produtivas inter-relacionadas e mutuamente dependentes podem ser vistas como interligadas, mas de modo a não permitir acessos aos processos internos realizados em cada uma delas. As informações se restringem àquelas que habilitam as etapas devidas do processo em cada planta. Em outras palavras, vê-se as plantas como “caixas-pretas”, garantindo o sigilo dos processos das partes.

- Fusão de lugares – dois lugares, de modelos diferentes, comportam-se como se fossem um só, ou seja, recebem e perdem as **marcas** simultaneamente;
- Fusão de transições – duas transições, de modelos diferentes, comportam-se como uma, ou seja, só disparam se as **pré** e as **pós-condições** de ambas conjuntamente são atendidas;
- Comunicação por arcos habilitadores – a conexão de arcos habilitadores⁷ entre os modelos permite que se controlem os disparos de transições.

Dentre estas, a fusão de transições está sendo considerada no presente trabalho por ser a forma mais simples de “administrar” as **marcas** (uma vez que na notação na fusão de lugares pode-se ter situações de dubiedade quando de requisições simultâneas) e por incluir menos elementos na RdP em relação à interface por arcos habilitadores.

2.5.2 Algoritmos de Sincronização

A sincronização do tempo de simulação é importante para fazer com que as relações do sistema real sejam corretamente simuladas, sem que “causas e efeitos” percam suas relações devido a erros na sequência de simulação. Junqueira (2006) cita duas abordagens de sincronização: a conservadora e a otimista. Na conservadora (FUJIMOTO, 2003), cada evento só pode ser processado quando todos os eventos anteriores já ocorreram, enquanto na otimista, pode haver a ocorrência de um evento sem que todos os anteriores tenham ocorrido, porém, necessita considerar métodos de detecção e correção.

Além de ser mais simples, já que não envolve detecção e correção, a abordagem conservadora é aqui considerada por garantir que a sequência de execução será respeitada em todas as etapas das simulações.

2.5.3 Rede de Comunicação

Para gerenciamento da rede de comunicação, o protocolo para implementar o conceito de sincronização conservadora é inspirado no mecanismo de `token ring`. Essa técnica de controle de mensagens foi proposta pela primeira vez em 1969, e gerencia uma

⁷ Arcos que habilitam o **disparo** apenas quando satisfeita determinada condição.

rede de comunicação numa topologia em forma de anel, com uma única série de bits (token) que circula continuamente de estação em estação (PANDOLFI, 2005).

Todas as estações podem receber o token a qualquer momento, mas num determinado instante apenas uma pode enviá-lo, garantindo que as mensagens somente serão descartadas após terem circulado por todas as estações. Isto acontece porque há apenas **um** token circulando na rede (JUNQUEIRA, 2006). A Fig. 2.10 ilustra esse mecanismo de gerenciamento de mensagens numa rede de comunicação.

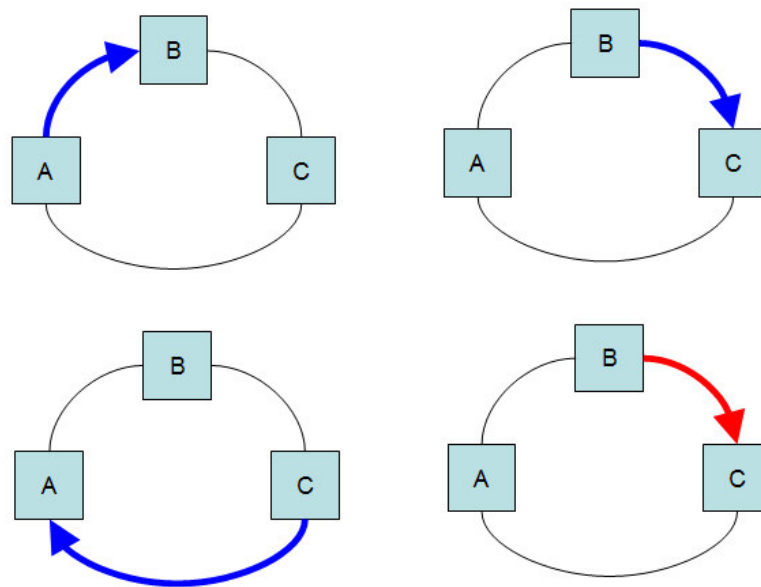


Fig. 2.10 - Exemplo de funcionamento do mecanismo de token ring (Estação A começa enviando uma mensagem – token – e B só poderá enviar a sua mensagem assim que o token enviado por A completar a sua circulação pelo anel).

2.6 Ferramenta para Modelagem e Simulação de Sistemas

Considera-se que o ambiente de simulação deve envolver tanto a etapa de modelagem como a de simulação, baseados em RdP. Por meio de uma interface gráfica, o ambiente implementado deve permitir ao usuário a criação ou edição de modelos em RdP, bem como a simulação deste, entre outras funcionalidades.

Do ponto de vista desse trabalho, rédeas regras de evolução da RdP serão utilizadas como base para o desenvolvimento da simulação distribuída, sendo necessária a inclusão de classes e protocolos de comunicação entre as simulações, bem como outros mecanismos, se necessários. O desenvolvimento de funcionalidades modulares deverá ser prioridade,

porquanto torna possível o futuro desenvolvimento e atualização de funcionalidades desse ambiente.

3 Implementação do AS

Nesse item, os principais pontos do projeto serão apresentados. Os conceitos apresentados anteriormente foram utilizados e desenvolvidos para implementar cada etapa do projeto.

A implementação do AS teve como ponto de partida o levantamento teórico apresentado anteriormente, que levou às definições de especificações e ferramentas para o projeto. As classes do programa e alguns testes executados também serão discutidos.

3.1 Especificações básicas de um AS

Partindo da proposta e dos estudos realizados, obteve-se a definição das tarefas a serem realizadas pelo ambiente de simulação (AS). Assim, este deverá:

- Constituir um sistema modular, para permitir futura integração a outros sistemas;
- Comunicar-se com um outro AS;
- Gerar respostas ao usuário, informando sobre o status da simulação, e sobre possíveis problemas.

3.2 Suportes considerados

Essa etapa incluiu a definição das ferramentas e linguagens a serem utilizadas, dentre elas, destaca-se:

- Linguagem: o XML para descrição da RdP;
- Método: o conceito de fusão de transições (da RdP) como forma de modelar a interface dos modelos;
- Técnica: o protocolo de comunicação em TCP/IP, implementado em Java para as interfaces entre os processos computacionais.

3.3 Estrutura do AS

Com a utilização dos métodos, técnicas e ferramentas citados determinou-se as classes que o ambiente necessita para que a simulação possa ser realizada, bem como a comunicação necessária para tal.

O estudo de trabalhos no mesmo contexto levantou algumas opções para a linguagem utilizada na implementação das funções de comunicação e os aspectos, tanto positivos quanto negativos (PANDOLFI, 2005; FREITAS, 2006) de cada uma. Tanto por algumas vantagens observadas, como pela familiaridade e fácil acesso⁸, o Java foi escolhido para implementar a comunicação.

Esta classe efetiva a comunicação entre os ambientes, i.e., programar as funções de controle do envio e o recebimento de informações, adotando a abordagem conservadora na sincronização, via protocolo *token ring*. São considerados dois tipos de mensagem para efetuar a comunicação entre duas estações:

- i) chamada de método;
- ii) circulação do *token*.

A chamada de método é uma consulta a uma outra estação acerca da situação de determinada transição (que está associada a alguma outra pela estrutura de fusão entre elas). A comunicação para este fim usa o mecanismo do *token ring*, onde apenas a estação de interesse decodifica a citada mensagem. Esta é circulada no *anel* de modo que é repassada até a estação destinatária, sem que seu conteúdo seja alterado durante o percurso.

Uma função dentro dessa classe envia⁹, de uma estação, por exemplo, a “pergunta” para uma estação C e aguarda a resposta. Na estação C¹⁰, caso a transição esteja habilitada, ocorre o envio da resposta afirmativa para estação A. Somente então A

⁸ A linguagem Java está em amplo desenvolvimento pela SUN MICROSYSTEMS (2007) e possui inúmeros tutoriais (JAVA\ WEB DEVELOPER, 2007), bem como compiladores e editores disponíveis gratuitamente na internet (NETBEANS, 2007; TEXTPAD, 2007).

⁹ Essa função chama-se *EnviaRecebeChamada de Método*, cujo algoritmo é representado pela Fig. 3.1 e o código encontra-se no anexo 1.B.5.

¹⁰ A função que gera a resposta e/ou dispara, chama-se *VerificaChamadaMétodo*. Algoritmo na Fig. 3.2, código no anexo 1.B.4

decodifica a mensagem e informa sobre o disparo da transição questionada e reenvia a nova mensagem solicitando o disparo. Caso C não esteja habilitada, é enviado o mesmo tipo de mensagem, porém negativa.

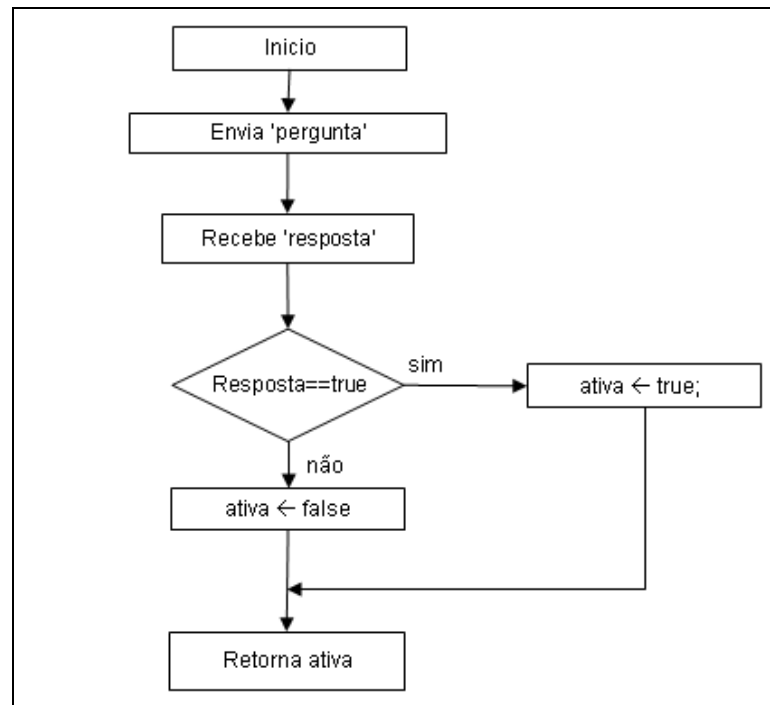


Fig. 3.1 – Algoritmo da EnviaRecebeChamada de Método

O padrão criado se diferencia da mensagem “token” e carrega os seguintes dados:

- String que pode conter “pergunta” (caso seja uma chamada) ou “resposta”;
- Char contendo o nome da estação remetente;
- Char contendo o nome da estação destinatária;
- String com o nome da transição (na estação destinatária);
- Boolean contendo:
 - No caso de chamadas, se o remetente faz uma verificação da condição (false) ou uma ordem de disparo (true);
 - No caso de respostas, se a transição está (true) ou não (false) habilitada na estação. Numa resposta a uma ordem de disparo, será “false” se houver algum erro;

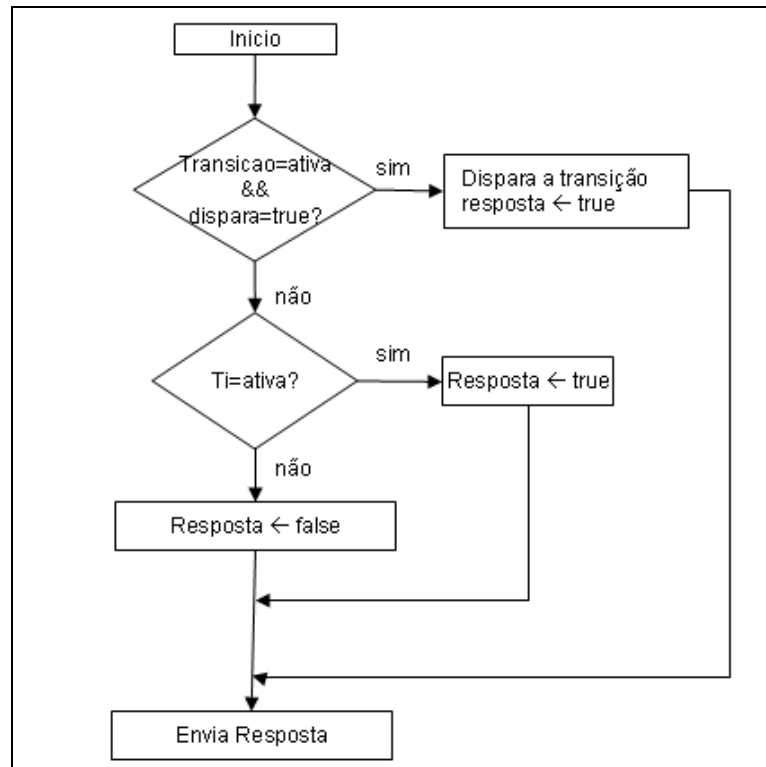


Fig. 3.2 – Algoritmo da VerificaChamadaMétodo

Na Tabela 3.1 é apresentada uma possível sequência de troca de mensagem entre duas estações “A” e “C”.

Tabela 3.1 - Uma sequência de chamadas de método.

Remetente	Destinatária	Mensagem	Evento
A	C	“chamada A C T1 false”	“A” pergunta a “C” se T1 está habilitada.
C	A	“resposta C A T1 false”	“C” avisa que T1 não está habilitada.
C	A	“chamada C A T2 false”	“C” pergunta a “A” se T2 está habilitada.
A	C	“resposta A C T2 true”	“A” avista que T2 está habilitada. Mas T2 não é disparada ainda.
C	A	“chamada C A T2 true”	“C” solicita a “A” o disparo da T2 e “C” dispara a sua T1 local.
A	C	“resposta A C T2 true”	“A” confirma o disparo.

Cada estação possui um Cliente e um Servidor, implementados para criar o mecanismo `token ring`. A estrutura resultante permite que se organize a comunicação entre as estações tal que se assegure um sentido único de movimentação das mensagens. Em outras palavras, a estação A envia somente para B e recebe somente de C, como as setas representadas na Fig. 3.3

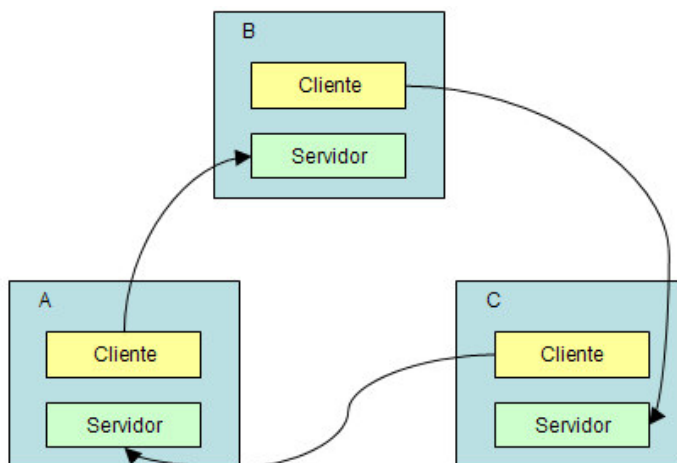


Fig. 3.3 – Circulação de chamada de método

O conjunto de tipos de dados desta mensagem (compostos em uma `string`) a serem trocados durante um ciclo lógico é explicitado na Tabela 3.2

Tabela 3.2 – Composição de um `token`.

Posição na <code>string</code>	Nome	Descrição
1	Identificação da Estação	Identifica a última estação a alterar os dados do <code>token</code> .
2	Tempo Futuro	Tempo de simulação requisitado pela estação indicada na posição 1 da <code>string</code> .
3	Status	Contém o estado atual da estação indicada na posição 1. (possíveis valores na Tabela 3.3)
4	Instrução	Informa a todas estações qual a instrução a ser executada. (por exemplo, o início ou término da simulação)
5	Erro	Notifica erros de simulação (falha na comunicação etc.).

Tabela 3.3 – Possíveis valores para o campo de *status* (JUNQUEIRA 2006).

Valores do campo de <i>status</i>	Significado
0	Nenhuma estação está usando o token.
1	A estação está verificando o status corrente de outras estações.
2	A estação está enviando uma ordem para que a demais estações atualizem seus tempos de simulação com base no campo de <i>tempo futuro</i> (posição 3 na string).
3	A estação entrou em <i>deadlock</i> ¹¹ .

A estação mestre (neste programa a estação mestre será sempre a estação denominada “A”¹²) inicia a comunicação enviando um token com o comando de “início” (“1”, na posição 4 da string, conforme tabela 3.1). Assim que recebem esse comando, a(s) estação (ões) iniciam o disparo das transições instantâneas que não dependem de outras (ou seja, as transições possíveis), e enviam as requisições de disparo (ou chamadas de métodos, definidos na classe) se tiverem alguma transição dependente.

Em seguida, uma estação pode requisitar a evolução do tempo de simulação (posição 2 na string) necessária para progredir a sua própria simulação. Essa requisição será avaliada pelas outras estações e o menor tempo requisitado será atendido via comando da estação requisitante, uma vez que as outras tenham requisições de tempo maiores (valor da posição de status é 2, segundo Tabela 3.3). A cada iteração, as estações verificam as chamadas de método para as transições envolvidas.

Esse ciclo se repete até que a simulação seja interrompida pelo usuário ou todas as estações entrem em *deadlock*.

O protocolo TCP/IP foi implementado de forma que duas estações (virtuais ou não) se comuniquem. A troca de informações inicia-se do cliente para o servidor e só evolui quando o servidor envia uma resposta (confirmação de recebimento). Isso garante a

¹¹ *Deadlock*: Quando o modelo não possui mais transições disparáveis, diz-se que ele está em *deadlock*.

¹² O procedimento de inicialização das estações e conseqüente início da simulação será apresentado no decorrer deste capítulo. .

abordagem conservadora na comunicação, i.e., não há continuação da simulação sem que o token circule entre as estações envolvidas.

A Fig. 3.4 traz um fluxograma de funcionamento do controle da comunicação entre as estações. No Anexo 1.B.3, é apresentado o código fonte dessa função responsável pelo controle da comunicação.

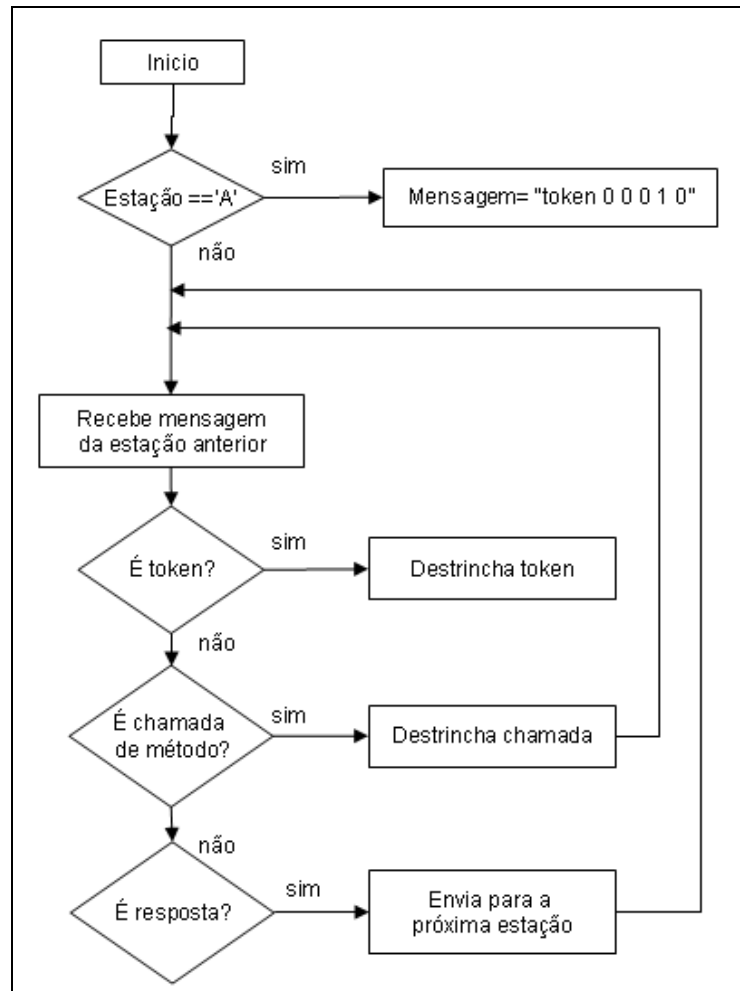


Fig. 3.4 - Gerenciamento das Mensagens (Controle Token ring)

Assume-se que os modelos a serem simulados são redes de Petri. Se por um lado a representação gráfica da RdP facilita em muito a visualização das características do sistema, por outro é fortemente desejável um padrão computacional para definir os dados do modelo e facilitar a manipulação de informações por programas. Como apresentado em Freitas (2006), a notação XML (eXtensible Markup Language), mais precisamente a PNML (Petri Net Markup Language, baseada em XML) foi aqui considerada pois padroniza essa representação e facilita o intercâmbio de informações entre softwares.

Além disso, se necessário, pode-se definir novos *tags* em PNML para incluir mais informações sobre a rede, pois diferentes tipos de RdP ou diferentes algoritmos estão em constante desenvolvimento [13].

O levantamento realizado sobre padrões para fusões de transições e/ou temporizações indicou a inexistência de soluções comprovadas, assim, Tabela 3.4 e Tabela 3.5 apresentam-se os atributos adicionais introduzidos.

Tabela 3.4 – Atributos modificados da tag “transition”.

Tipo	Tag	Atributo XML	Descrição
Transição	<transition>	id: ID	Identificador
		x: X	Coordenada Absoluta no eixo X
		y: Y	Coordenada Absoluta no eixo Y
		type: TYPE	Tipo da Transição (Normal- “normal”, Temporizada- “TT” ou (envolvida em) Fusão – “TF”)

Tabela 3.5 – Atributos Específicos criados para a tag “transition”.

Tipo da Transição	Atributo XML	Descrição
Envolvida em Fusão Type:”TF”	s1:S1	Nome Estação 1
	s1:S1	Nome Estação 2
	t1:t1	Nome Transição da Estação 1
	t2:t2	Nome Transição da Estação 2
Temporizada Type:TT	Time:TEMPO	Tempo necessário para disparo.

Assim como no caso citado em Freitas (2006), esta classe implementa a leitura dos arquivos XML/PNML e os transforma em dados próprios para serem tratados pelo AS.

Baseado no algoritmo “jogador de marcas”, descrito e implementado em FREITAS (2006), e também no conceito de “*imply*” da lógica-linear para RdP implementado em MAZZARIOL (2005), foi desenvolvida uma classe que gerencia os disparos das transições e as alterações da marcação de uma RdP durante a simulação.

Na Fig. 3.5 tem-se o fluxograma apresentado em Freitas (2006) que indica as etapas necessárias para implementação do algoritmo. Nesta figura, as partes incluídas estão em destaque. O código-fonte dessa função está no anexo 1.B.6.

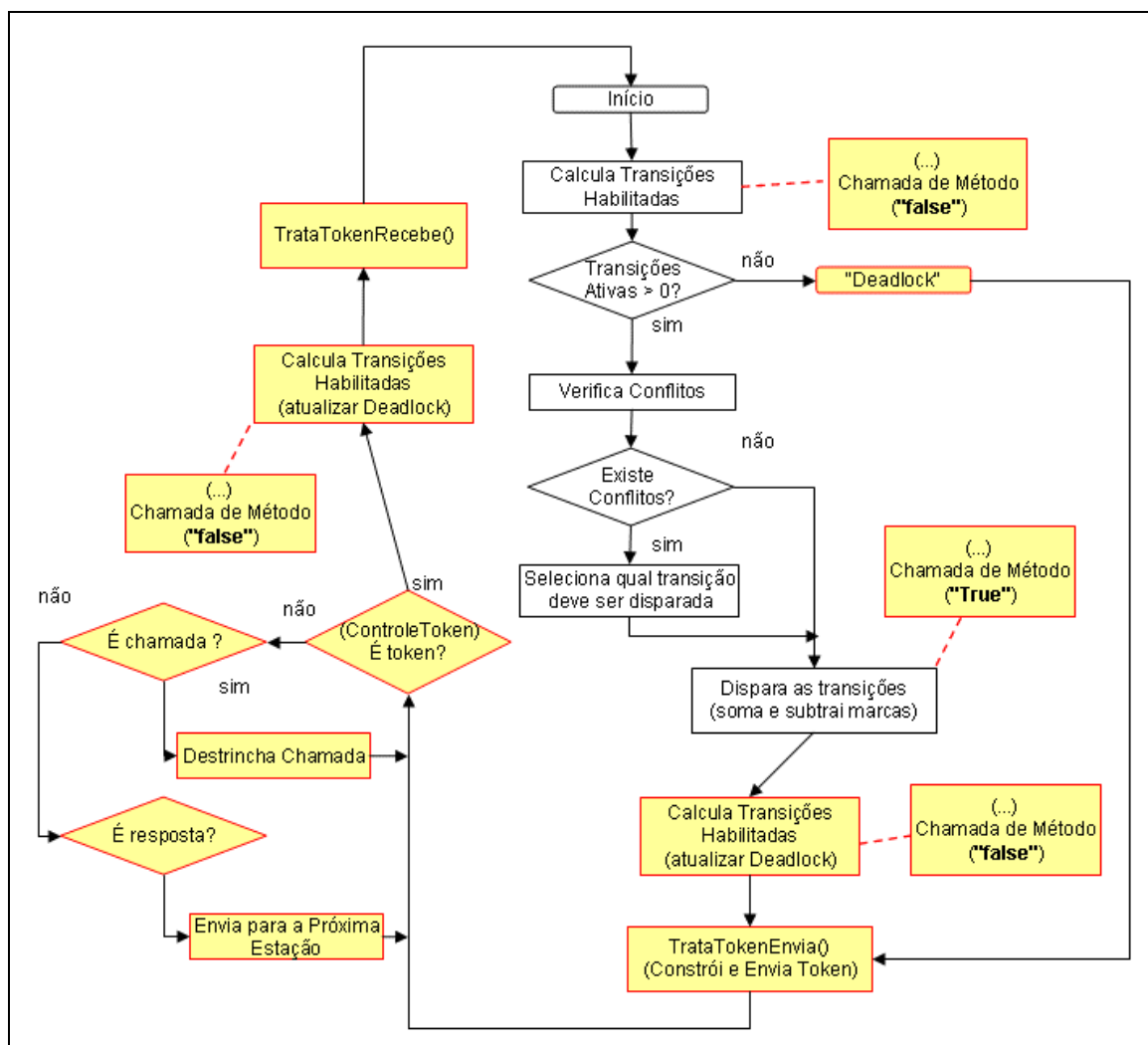


Fig. 3.5 – Fluxograma da seqüência de simulação (Freitas 2006 modificado).

O algoritmo “jogador de marcas” analisa as transições e separa as ativas, inclusive fazendo as chamadas a outras estações (via `VerificaChamadaMetodo`), disparando as transições ativas em seguida.

O losango que contém os dizeres "ControleToken" é o responsável pelo correto fluxo de mensagens. Só será chamado o "algoritmo jogador de marcas" (que de fato controla os disparos da simulação) se a mensagem recebida de estação anterior for um token. Se for uma "chamada" para a estação em questão, será invocada a função "VerificaChamadaMetodo". Se contiver "resposta", certamente não será para esta estação,

uma vez que a *EnviaRecebeChamada* (a qual solicita esse tipo de mensagem) fica aguardando sem deixar o programa siga antes de receber e interpretar essa informação.

A verificação e avaliação dos parâmetros do modelo são a chave para a sincronia da simulação. É através desses que se assegura a correta evolução do tempo de simulação, tal que todas as estações sigam a mesma velocidade de mudança, respeitando-se assim todas as relações de causa-consequência do modelo simulado e do tipo de abordagem (conservadora) adotado.

Um desses parâmetros é o *deadlock*, que pode assumir valores de acordo com o estado da estação com relação às suas transições. Os valores que ele pode assumir são apresentados na Tabela 3.6.

Tabela 3.6 - Valores possíveis para o parâmetro *deadlock*

Valor	Significado
0	Existem transições instantâneas ou não disparáveis.
1	Existem transições temporizadas com tempo superior ao tempo corrente de simulação.
2	Não mais transições disparáveis em qualquer tempo.
3	Existem transições fundidas parcialmente ¹³ habilitadas.

Para correta sequência do programa, o *deadlock* de valor “3” foi inserido de modo que as negociações de tempo ocorressem paralelas às chamadas de método (verificação da habilitação das transições fundidas).

Definição (JUNQUEIRA 2006), o *token* deverá ser atualizado ou conservado, conforme as seguintes regras:

¹³ Diz-se parcialmente habilitada a transição que está habilitada no modelo local, porém espera a transição do outro modelo ser habilitada.

REGRA 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 da sua condição.

REGRA 2

O modelo local está em *deadlock*, pois não possui mais transições instantâneas disparáveis em sua lista de disparos. Neste caso, ele deve consultar os demais modelos a fim de evoluir o tempo de simulação do sistema.

REGRA 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 dentre os modelos.

REGRA 4

O modelo local está em *deadlock* e recebe o `token` onde um outro modelo, com tempo de simulação superior ao local, está fazendo uma consulta. Neste caso, ele deve sobrescrever os campos de identificação e tempo futuro com os seus próprios valores, passando a fazer a consulta aos demais.

REGRA 5

O modelo local está em *deadlock* e recebe o `token` onde um outro modelo informa que está sem transições disparáveis. Neste caso, ele deve sobrescrever os campos de identificação e tempo futuro e trocar o status do `token` para consulta, informando que este modelo tem transições na sua lista de disparos.

REGRA 6

O modelo local recebe uma instrução para atualizar seu tempo local de simulação. Neste caso, ele deve atualizar o seu relógio local com o tempo informado no `token` e alterar sua variável de *deadlock* para zero, a fim de disparar suas transições, caso exista alguma transição disparável neste novo tempo local.

REGRA 7

O modelo local recebe de volta o `token` que enviou, sem alterações, onde informava aos demais modelos que se encontrava sem mais transições disparáveis. Neste

caso, ele deve atualizar o campo de erro do `token`, informando que o sistema como um todo está travado e a simulação deve ser encerrada.

REGRA 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, conseqüentemente, o do sistema. Neste caso, ele altera o status do `token` para que seja informado às demais estações que estas devem atualizar seus relógios locais de simulação com o valor utilizado na consulta.

REGRA 9

O modelo local recebe de volta o `token` que enviou, os demais modelos já atualizaram seus relógios locais restando apenas ele mesmo. Assim, ele libera o `token` e troca o seu status de *deadlock* para “0”, ou seja, o modelo não está mais em *deadlock*, podendo disparar suas transições.

Um fluxograma resumindo essas regras e as atualizações, seja das variáveis locais ou das a serem enviadas por `token`, encontra-se no anexo A. A correspondência entre os parâmetros modificados por essas regras, as variáveis relacionadas no fluxograma e os nomes destas no código em Java é apresentado na Tabela 3.7.

Tabela 3.7 - Correspondência entre nomes e variáveis.

Posição na String	Nome	Variável Fluxograma	Variável Código
1	Identificação da Estação	VarTokenId	Token.id
2	Tempo Futuro	VarTokenTempoFuturo	Token.tempoFuturo
3	Status	VarTokenStatus	Token.status
4	Instrução	VarTokenInstrucao	Token.instrucao
5	Erro	VarTokenErro	Token.erro

Na implementação optou-se por criar duas funções para tratar o `token`, uma no envio e outra no recebimento. No envio são aplicadas as regras 1, 2, 3, 4, 5 e 8. E no recebimento as regras 6, 7 e 9. Os códigos dessas funções estão nos anexos B.1 (TrataTokenEnvia) e B.2 (TrataTokenRecebe), respectivamente.

O ambiente possui uma interface com o usuário que permite informá-lo de como a estação “vê”¹⁴ o modelo a ser simulado, do tempo de simulação, sobre possíveis erros, interrupção da comunicação etc. Grande parte da visualização das redes foi obtido com o uso das classes implementadas por FREITAS (2006), seguida de modificações. Dentre elas a inserção de indicador do relógio local, informações sobre a conexão e as cores dos tipos de transição inseridas ("Rosa" para transições fundidas e "Azul" para as temporizadas).

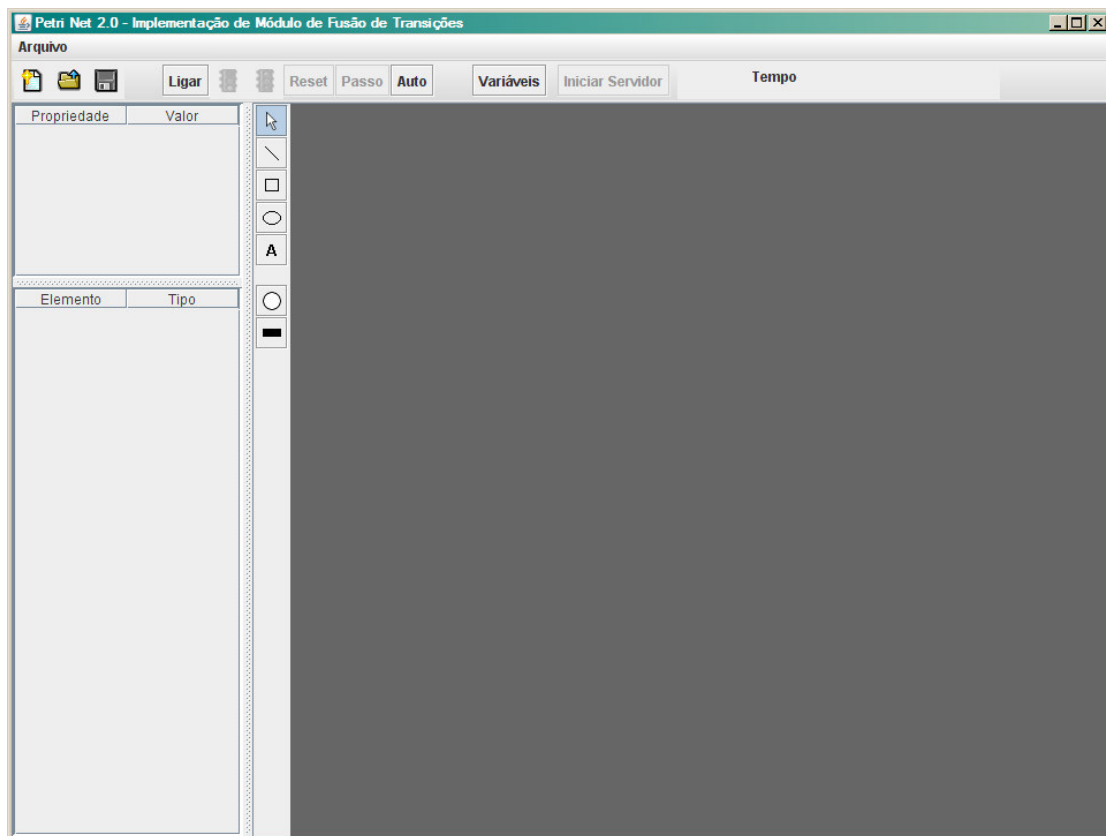


Fig. 3.6 – Janela Inicial do AS.

¹⁴ Transição correspondente no modelo remoto, i.e., a outra parte da Transição fundida na outra estação.

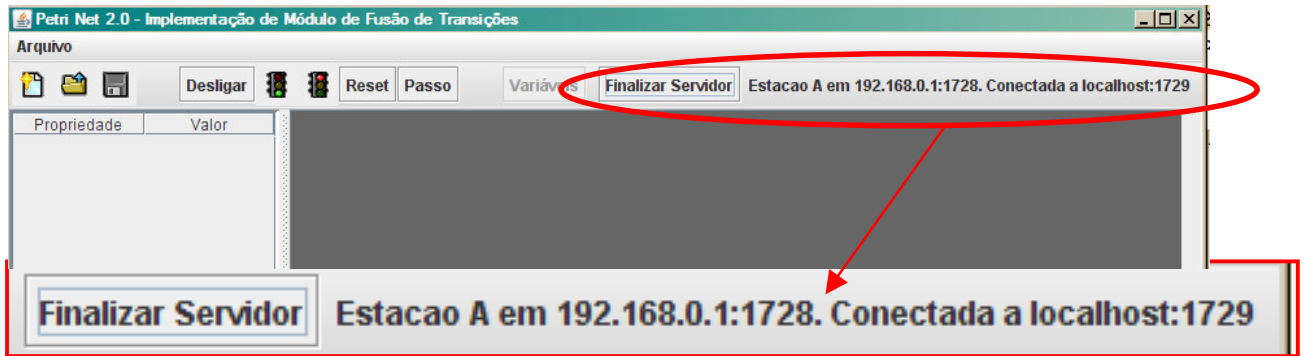


Fig. 3.7 – Status de estação Iniciada e a qual endereço está conectada.

Para encerrar a conexão, basta clicar no botão “Finalizar Servidor”. O procedimento para criação dos modelos e execução da simulação deve ser feito conforme orientações a seguir.

O primeiro passo para simulação distribuída é a criação dos modelos que serão as estações representativas da realidade. Na Fig. 3.8 estão modeladas 3 estações (Modelos A, B e C) sendo que a estação B, que possui tanto transição temporizada quanto fundida, será utilizada como exemplo para este procedimento.

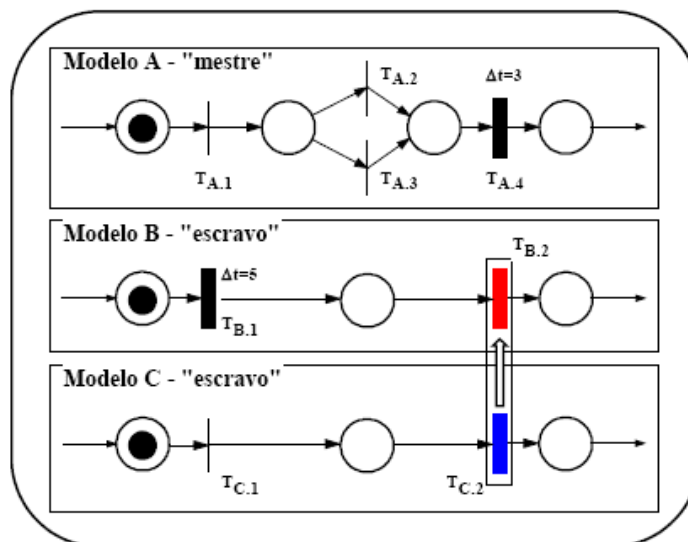


Fig. 3.8 – Modelo para simulação distribuída (JUNQUEIRA 2006).

Após abrir o programa, deve-se clicar no botão “novo” (📄) da interface e aguardar a janela de edição.

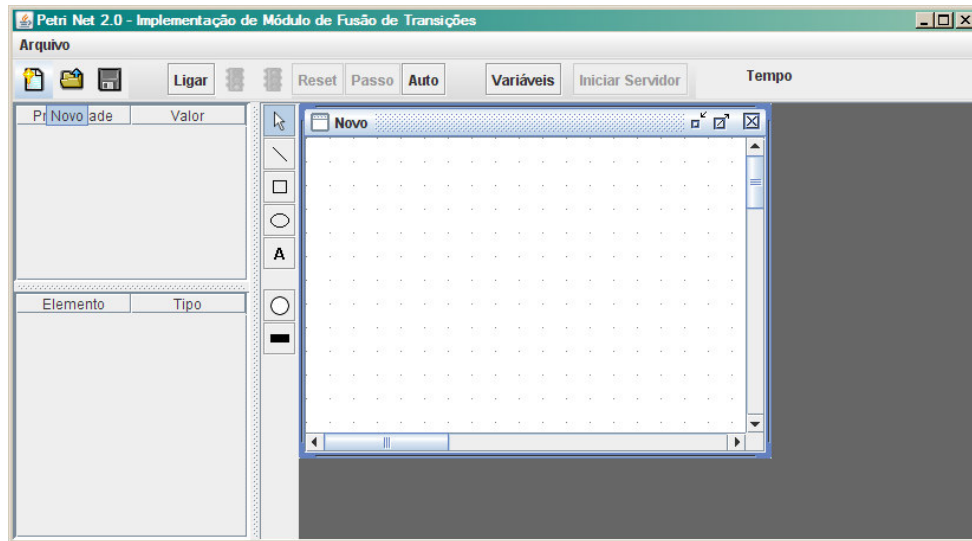


Fig. 3.9 – Janela de Edição

Para inserir um texto (“Estação B” no caso), clique no botão “A” (**A**) e depois no lugar desejado para posicioná-lo. Digite o texto no quadro que aparecer e depois aperte “ok”.

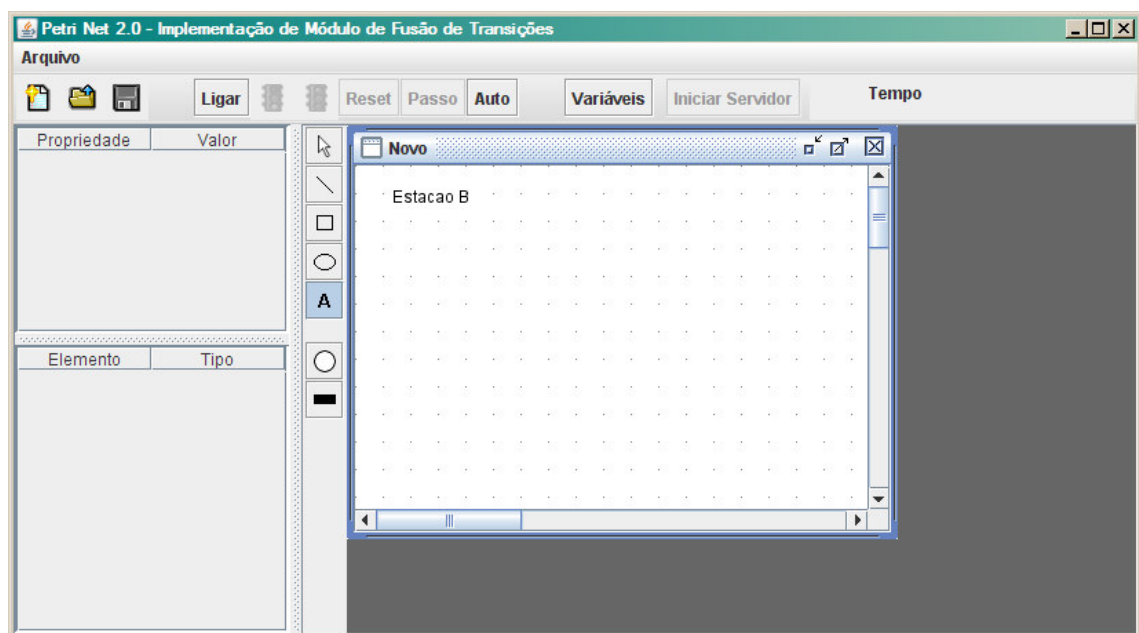




Fig. 3.10 - Inclusão de Texto no Modelo

A inserção dos lugares é feita clicando no botão “” em destaque (Fig. 3.11) e posicionando-o na área de edição. Da mesma forma ocorre com as transições, “”

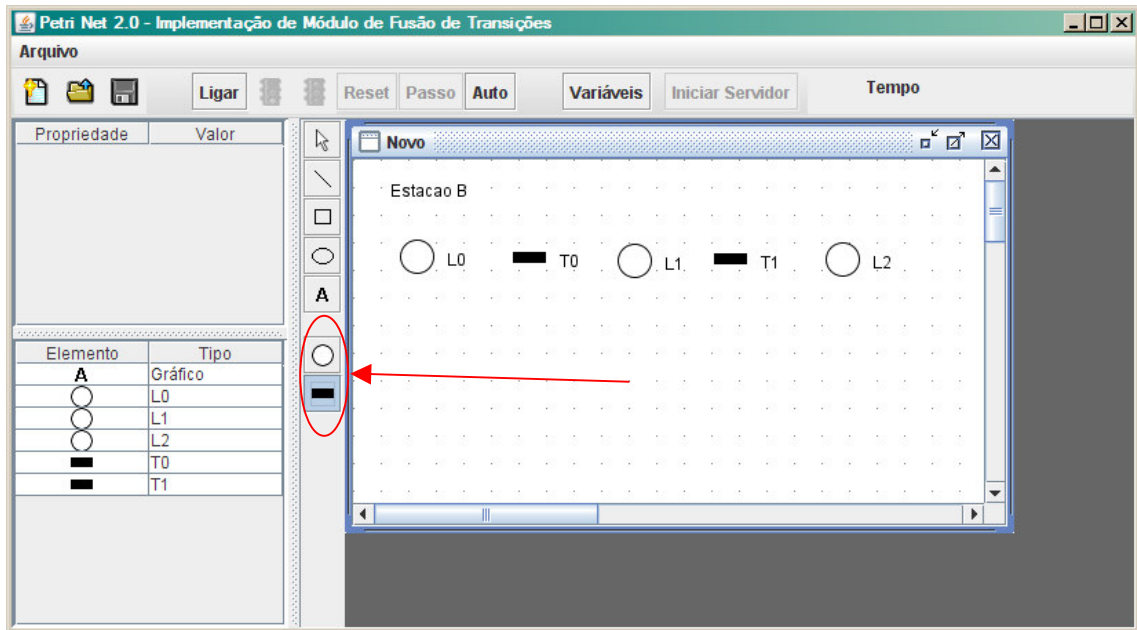



Fig. 3.11 - Inserção de Lugares e Transições

Para criar os arcos é necessário apertar o botão “” e depois no lugar ou na transição. Aparecerá uma circunferência azul claro, deve-se clicar nesta e arrastar até o ponto de destino para obter o arco orientado.

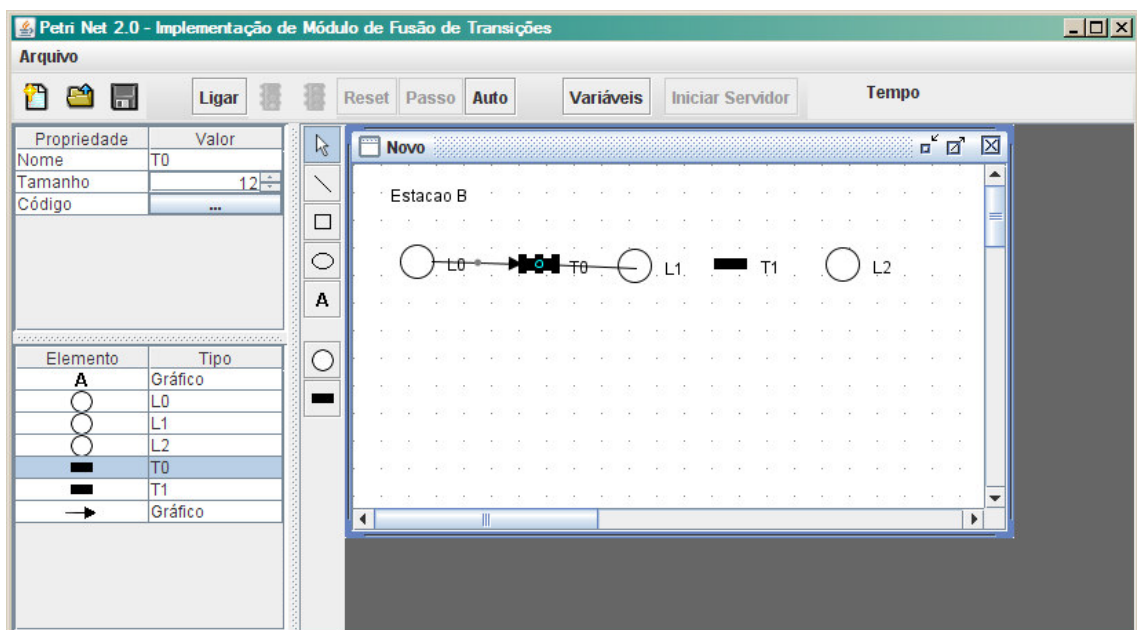


Fig. 3.12 - Inserção dos Arcos Orientados

A rede obtida ao final dessas etapas é apresentada na Fig. 3.13

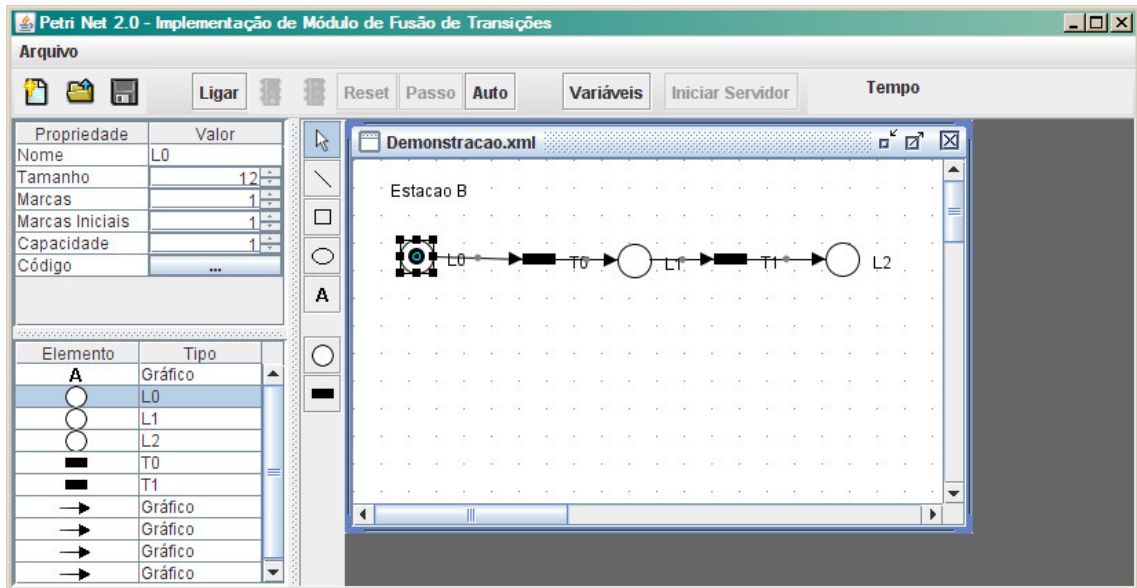



Fig. 3.13 - Modelo Gráfico Obtido

Para guardar o arquivo, basta clicar no botão salvar “” e escolher o diretório e nome apropriado.

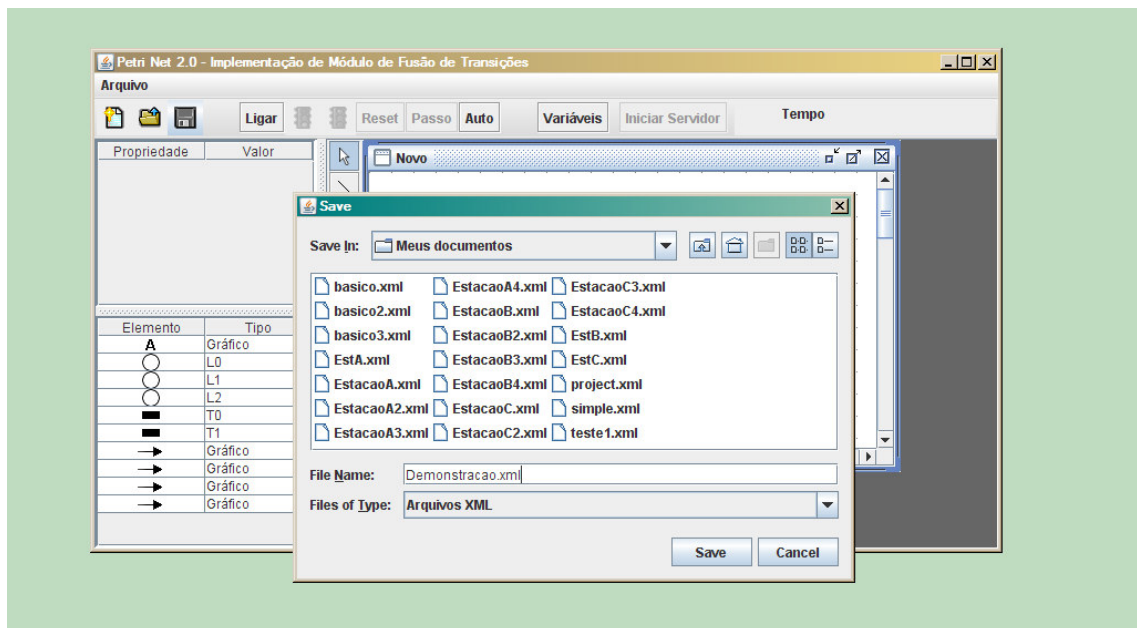


Fig. 3.14 - Gravação do Arquivo XML

A criação de modelos com fusão de transições e transições temporizadas exige um procedimento adicional, i.e., inserção de algumas propriedades manualmente.

Abrindo em um editor de texto (Notepad, por ex.), o arquivo .xml.

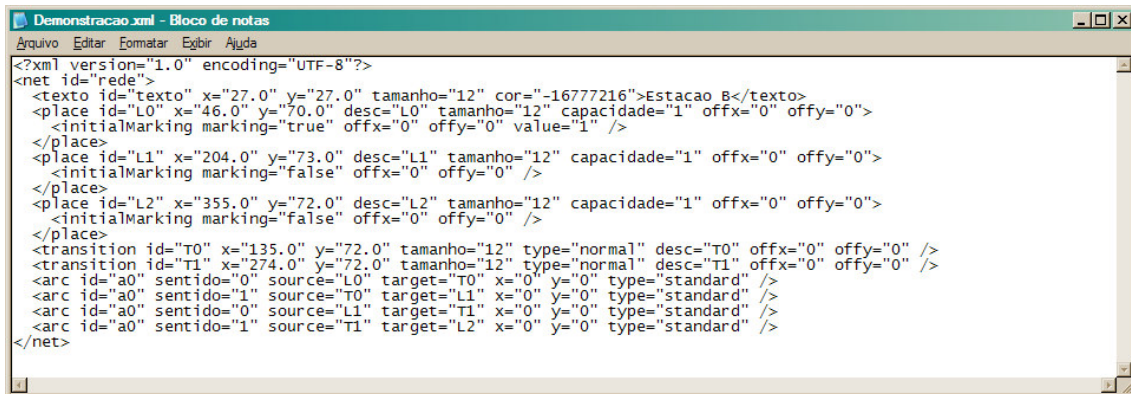


Fig. 3.15 - Arquivo em XML

Agora, é imprescindível modificar as transições T0 e T1, tal que fiquem temporizada e fundida, respectivamente.



Fig. 3.16 – Partes em destaque do arquivo devem ser modificadas

As modificações são necessárias para criar as transições temporizadas e fundidas, já que todas as transições criadas no editor são instantâneas.

Para a transição T0:

- Alterar tipo ("type") de "normal" para "TT" (Transição Temporizada);
- Inserir o tempo de disparo (time="5");

Para a transição T1:

- Alterar tipo ("type") de "normal" para "TF" (Transição Fundida);
- Inserir o nome da estação local (s1="B");
- Inserir o nome da estação remota (s2="C");
- Inserir o nome da transição na estação local (t1="T1");
- Inserir o nome da transição na estação remota (t2="T1");

```
<?xml version="1.0" encoding="UTF-8"?>
<net id="rede">
  <texto id="texto" x="27.0" y="27.0" tamanho="12" cor="-16777216">Estacao B</texto>
  <place id="L0" x="46.0" y="70.0" desc="L0" tamanho="12" capacidade="1" offx="0" offy="0">
    <initialMarking marking="true" offx="0" offy="0" value="1"/>
  </place>
  <place id="L1" x="204.0" y="73.0" desc="L1" tamanho="12" capacidade="1" offx="0" offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <place id="L2" x="355.0" y="72.0" desc="L2" tamanho="12" capacidade="1" offx="0" offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>

  <transition id="T0" x="135.0" y="72.0" tamanho="12" type="TT" desc="T0" offx="0" offy="0"
time="5" />
  <transition id="T1" x="274.0" y="72.0" tamanho="12" type="TF" desc="T1" offx="0" offy="0"
s1="B" s2="C" t1="T1" t2="T1" />

  <arc id="a0" sentido="0" source="L0" target="T0" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T0" target="L1" x="0" y="0" type="standard" />
  <arc id="a0" sentido="0" source="L1" target="T1" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T1" target="L2" x="0" y="0" type="standard" />
</net>
```

Fig. 3.17 - Partes inseridas e modificadas em destaque

Ao abrir o arquivo novamente, as transições agora possuem cores, Azul para temporizada ("TT") e rosa para fundidas ("TF"). As transições que são instantâneas ("normal") possuem cor preta.

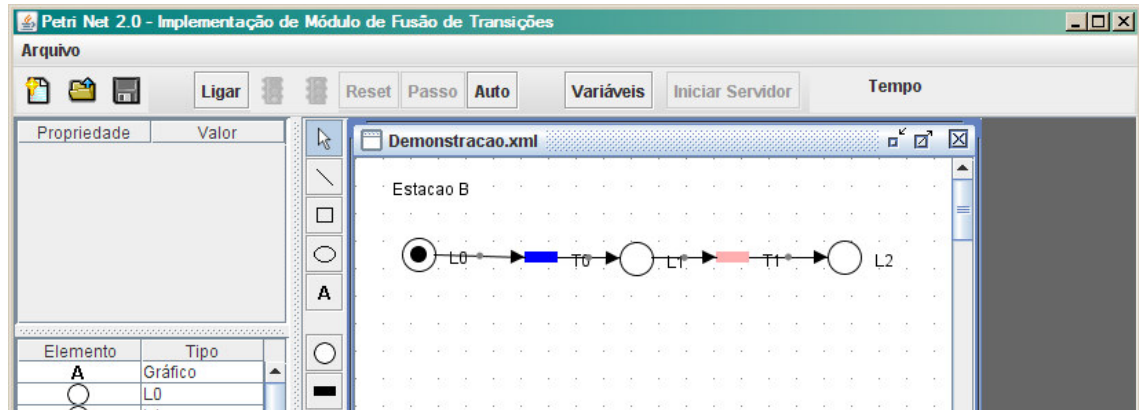


Fig. 3.18 – Modelo final da estação “B”

O passo seguinte é abrir os modelos e preparar as estações. Clicando em “abrir” e escolhendo os modelos. No caso “EstA.xml”, “EstB.xml” e “EstC.xml”, que estão nos anexos C.1, C.2 e C.3, respectivamente.

Após apertar o botão “**Ligar**” surgem campos necessários para a conexão das estações.

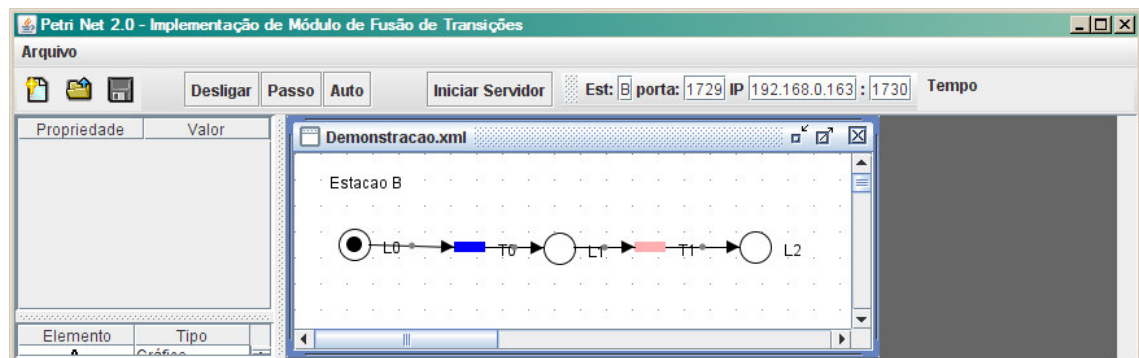


Fig. 3.19 – Campos para dados de comunicação

Estação: B.

Porta: 1729.

Próxima estação → IP 192.168.0.163 na porta 1730.

As estações A, B e C estão ligadas da seguinte forma:

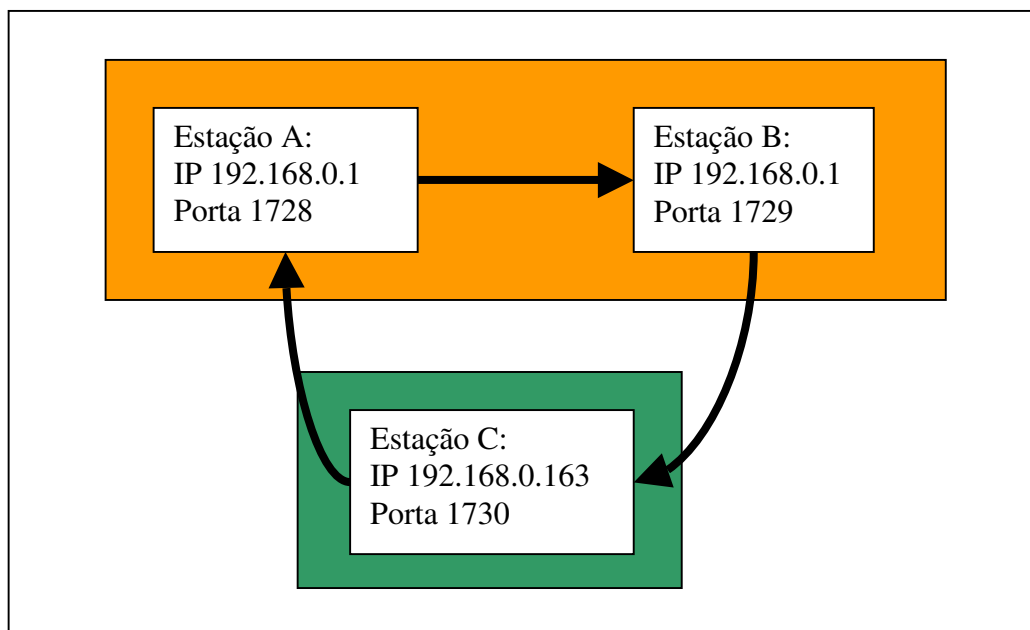


Fig. 3.20 – Anel lógico com endereços de 3 estações em 2 computadores

As estações devem ser inicializadas a partir da última, no caso a estação C, até a estação A. Os dados e a ordem são apresentados a seguir:

ArquivoModelo: EstC.xml

Estação: C.

Porta: 1730.

Próxima estação: IP 192.168.0.1 na porta 1728.

(Clicar em “**Iniciar Servidor**”)

ArquivoModelo: EstB.xml

Estação: B.

Porta: 1729.

Próxima estação: IP 192.168.0.163 na porta 1730.

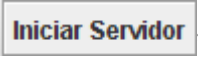
(Clicar em “**Iniciar Servidor**”)



ArquivoModelo: EstA.xml

Estação: A.

Porta: 1728.

Próxima estação: IP 192.168.0.1 na porta 1729.

(Clicar em “”)

A estação A estará pronta para se escolher o modo automático (“”) ou passo-a-passo (“”). No exemplo a ser mostrado, todas as estações foram rodadas passo-a-passo e a cada mudança na interface, foi tirada uma “impressão” da tela. Para se acompanhar as mudanças, o arquivo de texto gerado na simulação será intercalado com as respectivas mudanças gráficas.

4 Exemplo de Aplicação

4.1 Estudo de Caso 1

Nos anexos C.1, C.2 e C.3, estão inclusos os códigos em XML (modificados conforme o procedimento de criação e simulação apresentado anteriormente) das estações simuladas nos itens seguintes. As redes de Petri das 3 estações que serão simuladas foram apresentadas anteriormente na Fig. 4.1 (como exemplo para o procedimento).

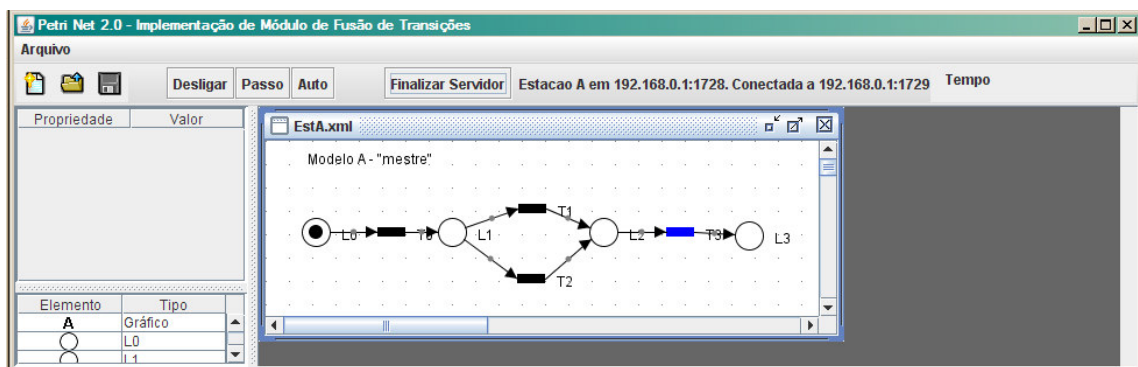


Fig. 4.1 – Estado inicial da estação “A”

Estacao A a conectar-se no 192.168.0.1 na porta 1729

Socket Conectado.

Estacao A a ouvir na porta 1728

ServerSocket Conectado.

Estacao: A recebe 'token 0 0 0 1 0'

T0 foi disparada. Tempo= 0

Estacao: A envia 'token 0 0 0 1 0'

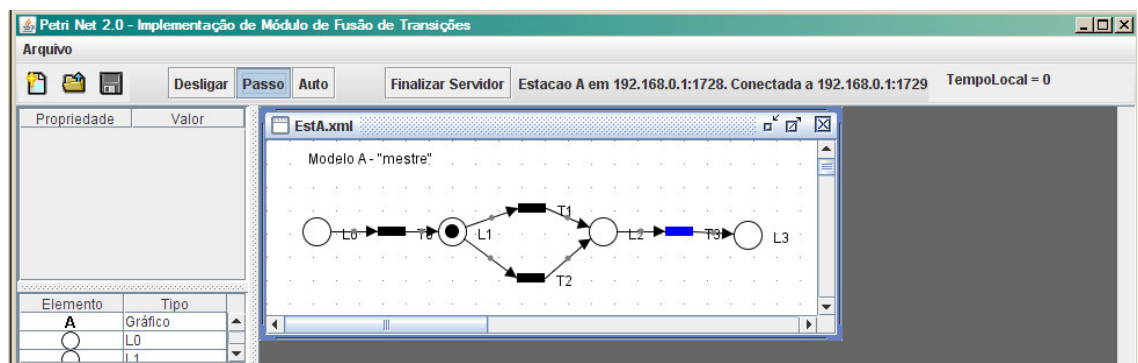


Fig. 4.2 – Disparo de T0

Estacao: A recebe 'token B 5 1 1 0'

T2 foi disparada. Tempo= 0

Estacao: A envia 'token A 3 1 1 0'

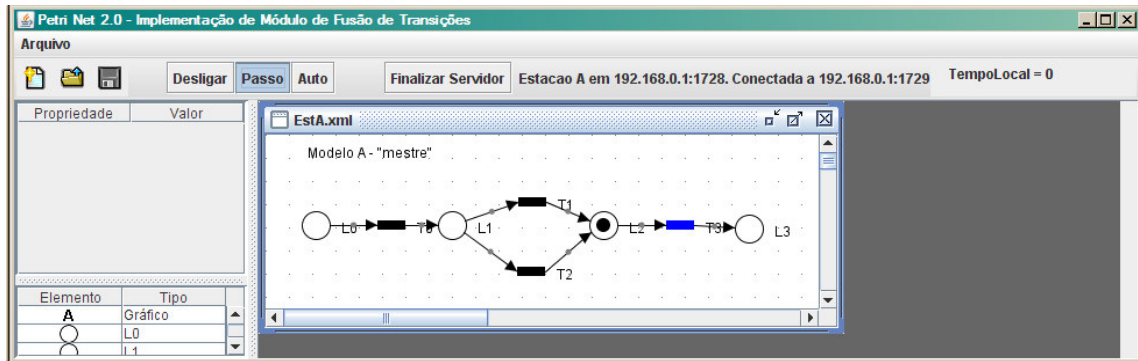


Fig. 4.3 – Disparo de T2

Estacao: A recebe 'token A 3 1 1 0'

Estacao: A envia 'token A 3 2 1 0'

Estacao: A recebe 'token A 3 2 1 0'

T3'temporizada' foi disparada. Tempo= 3

Estacao: A envia 'token A 3 3 1 0'

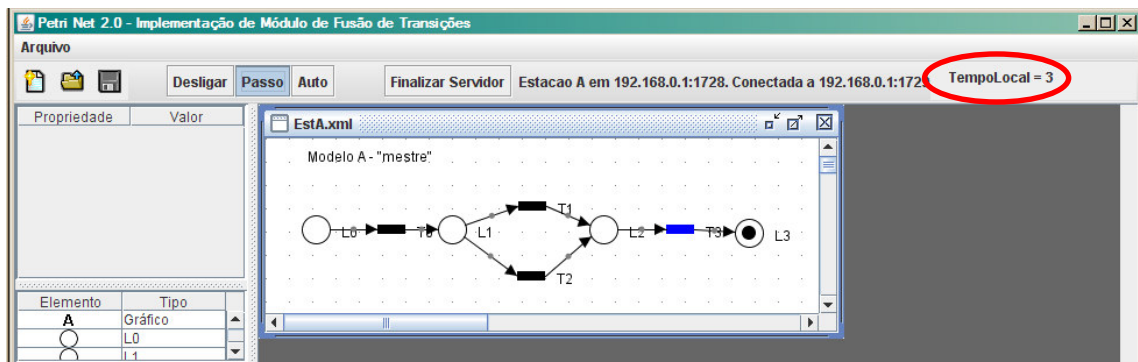


Fig. 4.4 – Disparo de T3

Estacao: A recebe 'token B 5 1 1 0'

Estacao: A envia 'token B 5 1 1 0'

Estacao: A recebe 'token B 5 2 1 0'

Estacao: A envia 'token B 5 2 1 0'

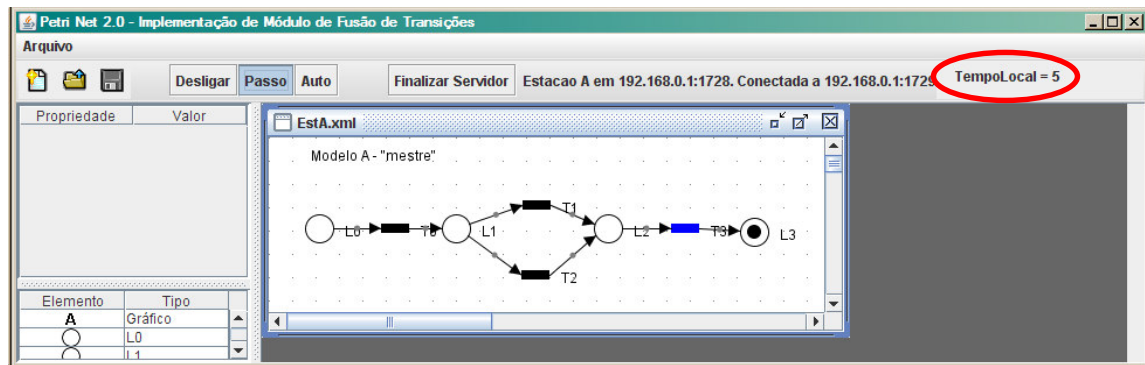


Fig. 4.5 – Deadlock e evolução de tempo para t=5

Estacao: A recebe 'token B 5 3 1 0'

Estacao: A envia 'token B 5 3 1 0'

Estacao: A recebe 'token B 5 3 0 1'

Estacao: A envia 'token B 5 3 0 1'

Socket Closed.

ServerSocket Closed.

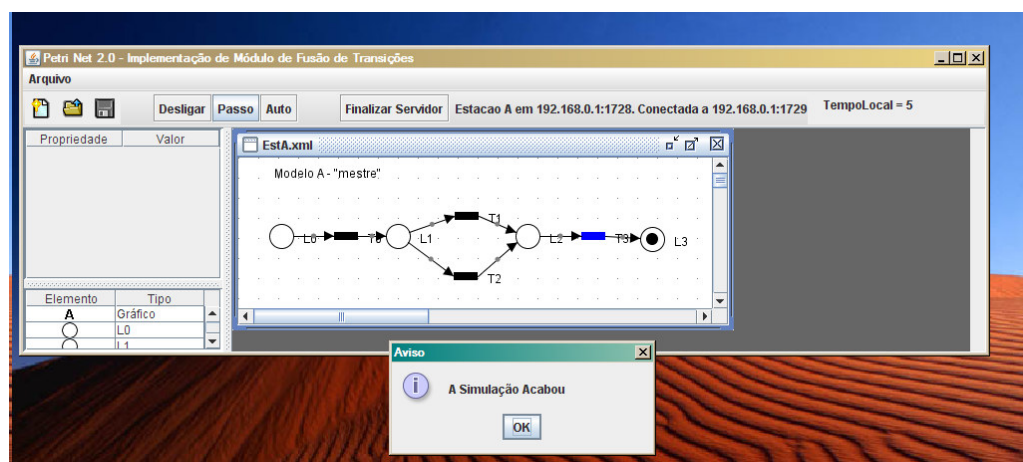


Fig. 4.6 – Fim da Simulação na Estação A

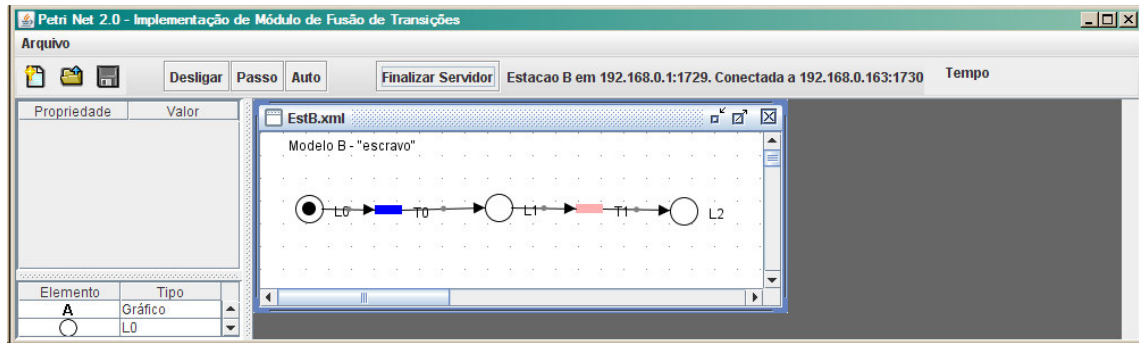


Fig. 4.7 - Estado inicial da estação “B”

Estacao B a ouvir na porta 1729

ServerSocket Conectado.

Estacao B a conectar-se no 192.168.0.163 na porta 1730

Socket Conectado.

Estacao: B recebe 'token 0 0 1 0'

Estacao: B envia 'token B 5 1 1 0'

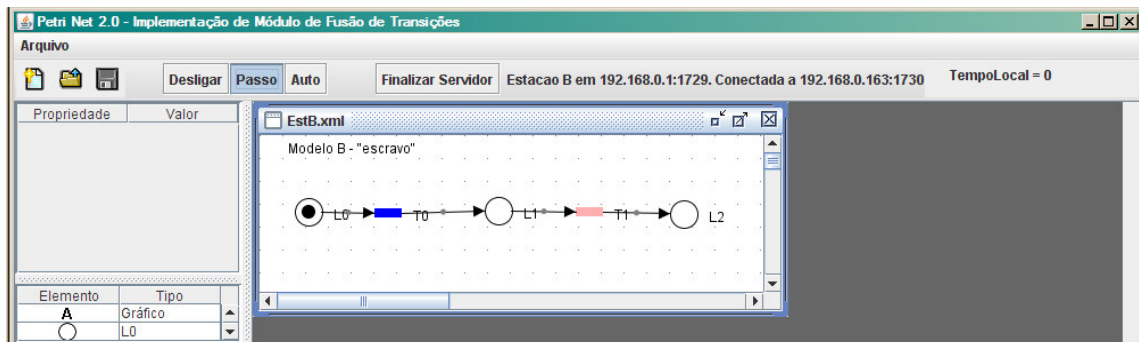


Fig. 4.8 – Estação negociando evolução do tempo

Estacao: B recebe 'chamada C B T1 false'

Estacao: B envia 'resposta B C false'

Estacao: B recebe 'token A 3 1 1 0'

Estacao: B envia 'token A 3 1 1 0'

Estacao: B recebe 'chamada C B T1 false'

Estacao: B envia 'resposta B C false'

Estacao: B recebe 'token A 3 2 1 0'

Estacao: B envia 'token A 3 2 1 0'

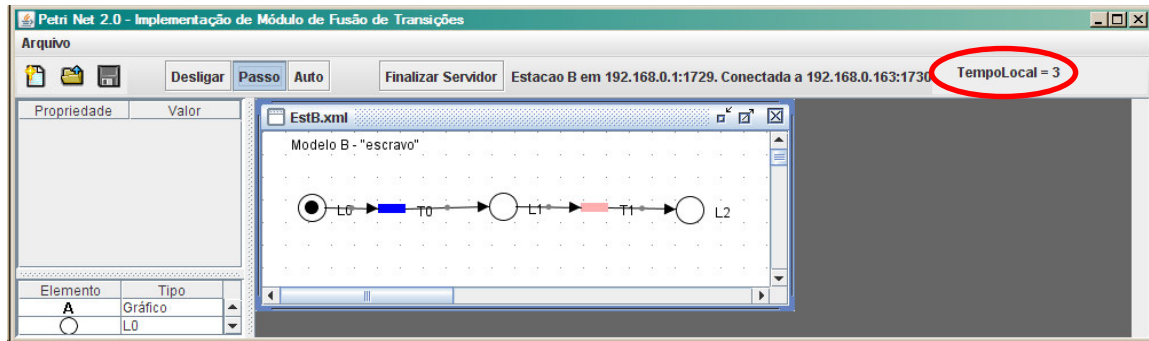


Fig. 4.9 – Evolução do tempo para t=3

Estacao: B recebe 'chamada C B T1 false'

Estacao: B envia 'resposta B C false'

Estacao: B recebe 'token A 3 3 1 0'

Estacao: B envia 'token B 5 1 1 0'

Estacao: B recebe 'chamada C B T1 false'

Estacao: B envia 'resposta B C false'

Estacao: B recebe 'token B 5 1 1 0'

Estacao: B envia 'token B 5 2 1 0'

Estacao: B recebe 'chamada C B T1 false'

Estacao: B envia 'resposta B C false'

Estacao: B recebe 'token B 5 2 1 0'

T0'temporizada' foi disparada. Tempo= 5

Estacao: B envia 'chamada B C T1 false'

Estacao: B recebe 'resposta C B true'

Estacao: B envia 'token B 5 3 1 0'

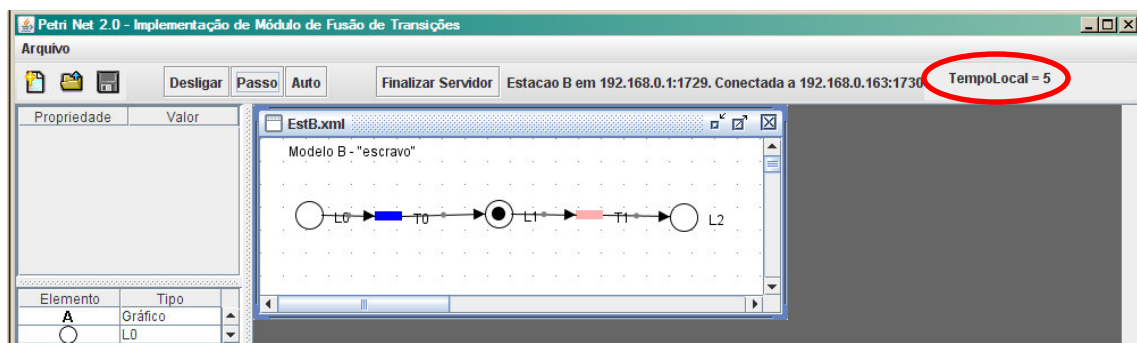


Fig. 4.10 – Disparo de T0

Estacao: B recebe 'chamada C B T1 false'

Estacao: B envia 'resposta B C true'

Estacao: B recebe 'chamada C B T1 true'

T1_B/T1_C 'fusion' foi disparada. Tempo= 5

Estacao: B envia 'resposta B C true'

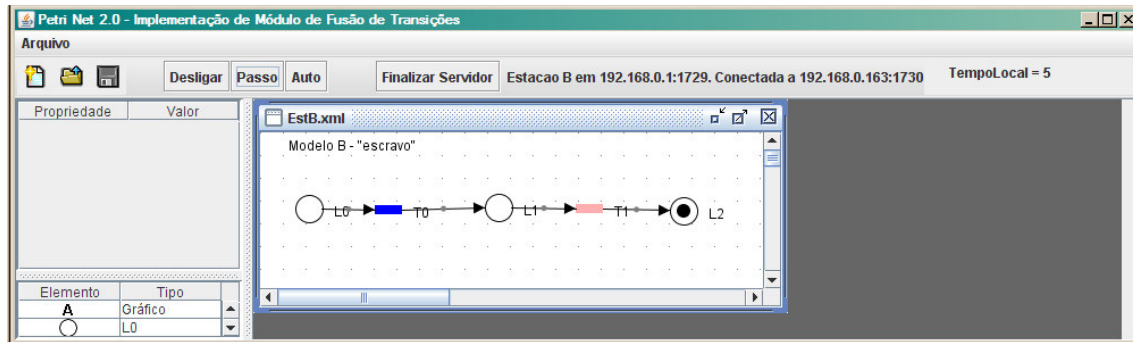


Fig. 4.11 – Disparo de T1_B/T1_C

Estacao: B recebe 'token B 5 3 1 0'

Estacao: B envia 'token B 5 3 0 1'

Socket Closed.

ServerSocket Closed.

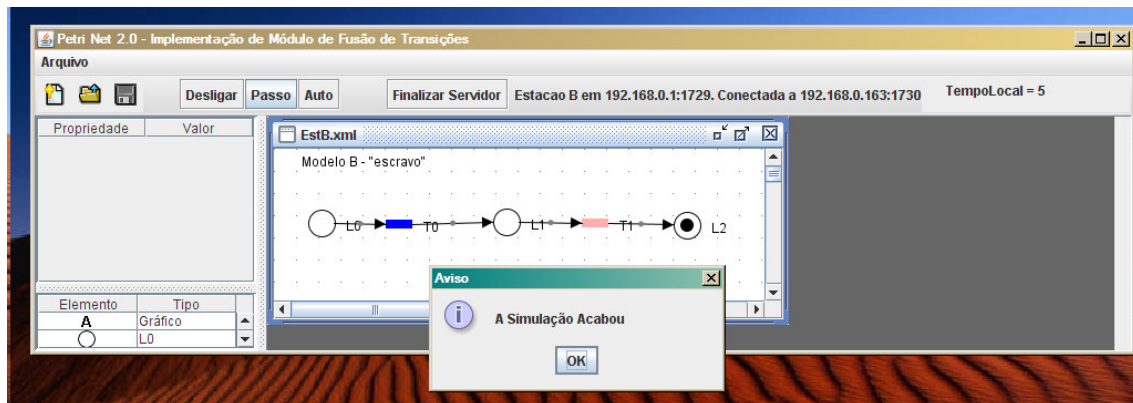


Fig. 4.12 – Fim da Simulação na Estação B

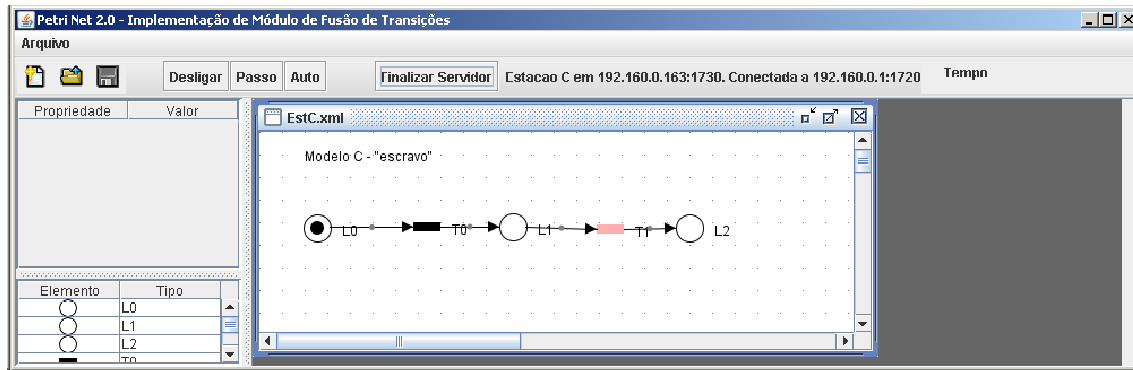


Fig. 4.13 – Estado inicial da estação “C”

Estacao C a ouvir na porta 1730

ServerSocket Conectado.

Estacao C a conectar-se no 192.168.0.1 na porta 1728

Socket Conectado.

Estacao: C recebe 'token B 5 1 1 0'

T0 foi disparada. Tempo= 0

Estacao: C envia 'chamada C B T1 false'

Estacao: C recebe 'resposta B C false'

Estacao: C envia 'token B 5 1 1 0'

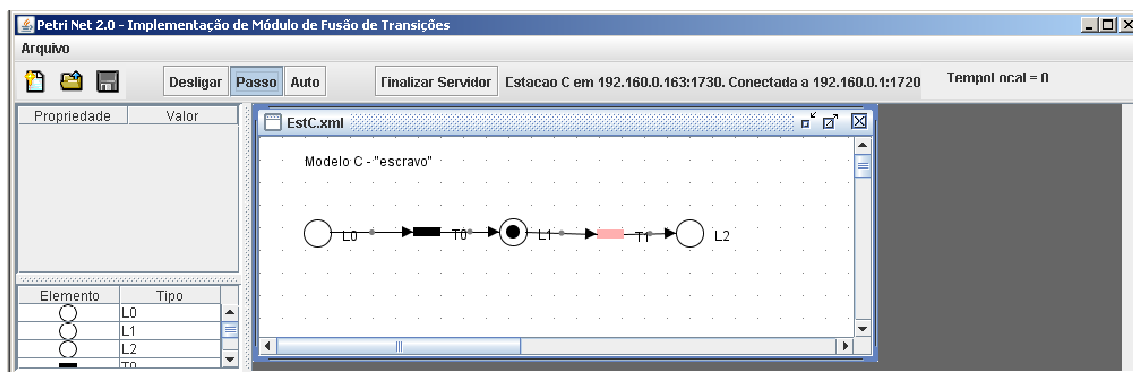


Fig. 4.14 – Disparo de T0

Estacao: C recebe 'token A 3 1 1 0'

Estacao: C envia 'chamada C B T1 false'

Estacao: C recebe 'resposta B C false'

Estacao: C envia 'token A 3 1 1 0'

Estacao: C recebe 'token A 3 2 1 0'

Estacao: C envia 'chamada C B T1 false'

Estacao: C recebe 'resposta B C false'

Estacao: C envia 'token A 3 2 1 0'

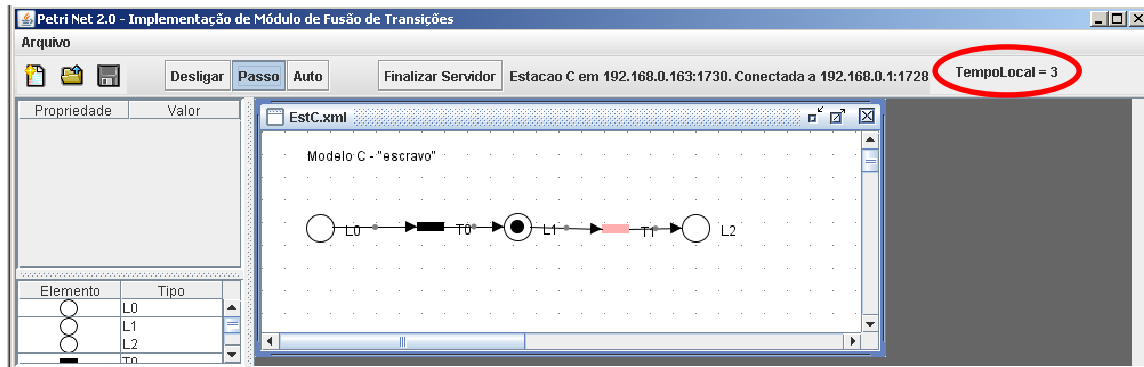


Fig. 4.15 – Evolução do tempo para t=3

Estacao: C recebe 'token B 5 1 1 0'

Estacao: C envia 'chamada C B T1 false'

Estacao: C recebe 'resposta B C false'

Estacao: C envia 'token B 5 1 1 0'

Estacao: C recebe 'token B 5 2 1 0'

Estacao: C envia 'chamada C B T1 false'

Estacao: C recebe 'resposta B C false'

Estacao: C envia 'token B 5 2 1 0'

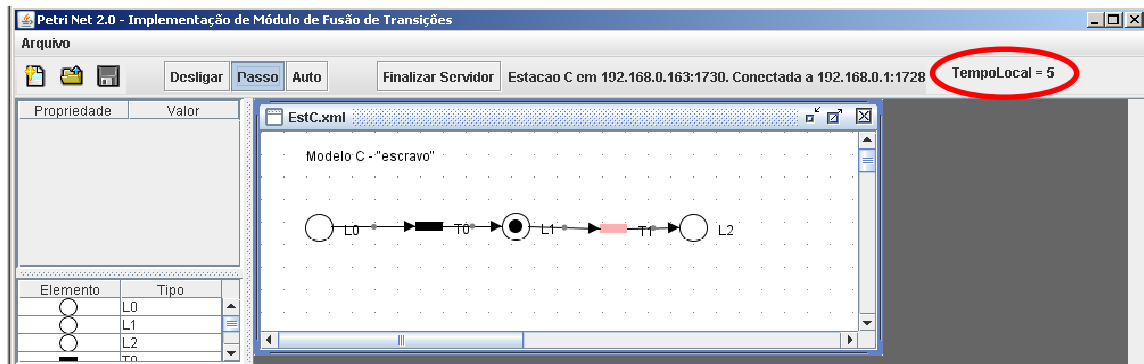


Fig. 4.16 – Evolução do tempo para t=5

Estacao: C recebe 'chamada B C T1 false'

Estacao: C envia 'resposta C B true'

Estacao: C recebe 'token B 5 3 1 0'

Estacao: C envia 'chamada C B T1 false'

Estacao: C recebe 'resposta B C true'

Estacao: C envia 'chamada C B T1 true'

Estacao: C recebe 'resposta B C true'

T1_C/T1_B 'fusion' foi disparada.

Tempo= 5

Estacao: C envia 'token B 5 3 1 0'

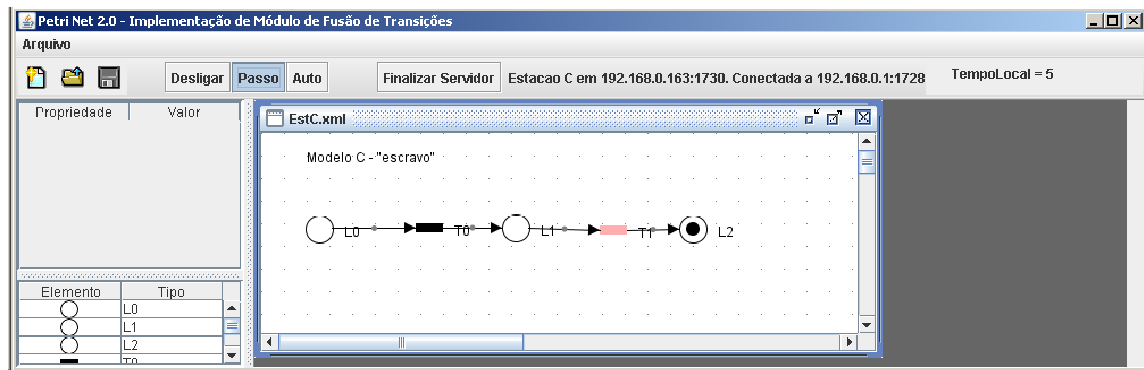


Fig. 4.17 - Disparo de T1_C/T1_B

Estacao: C recebe 'token B 5 3 0 1'

Estacao: C envia 'token B 5 3 0 1'

Socket Closed.

ServerSocket Closed.

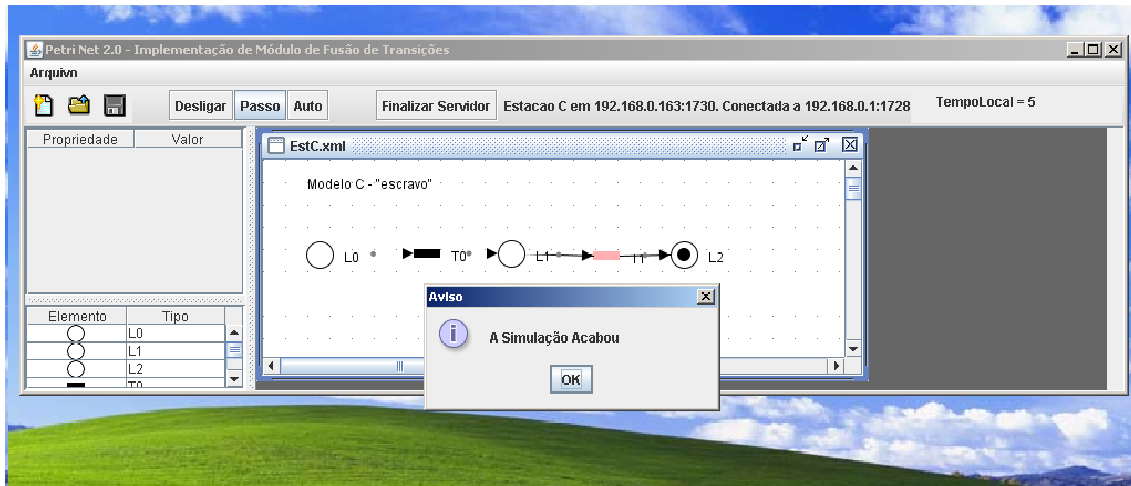


Fig. 4.18 – Fim da Simulação na Estação C

Comparando-se a sequência de simulação dos três modelos com o resultado apresentado em Junqueira (2006) pode-se afirmar que o AS simulou o sistema de forma correta, disparando as transições temporizadas no tempo correto e as transições de modelos distintos instantaneamente (ou seja, no mesmo tempo de simulação).

Assim, pode-se afirmar que o gerenciamento da comunicação e a negociação de evolução do tempo de simulação entre os modelos está correto.

4.2 Estudo de Caso 2

A rede criada para fins de simulação representa hipoteticamente um processo produtivo (ou serviço) onde se produz algo que será embalado e estocado. Uma possível situação onde isso ocorre é uma Gráfica. A linha principal é representada pela estação A, a linha auxiliar (ou terceirizada) pela estação B e a estação C representa a impressão e entrega de capas dos livros.

As transições temporizadas T0_A e T0_B seriam fábricas de papel (celulose), T2_A a máquina de impressão e a T0_C a fabricação das capas (outra planta). A fundida T3_A/T1_C representaria a entrega/disponibilidade para montagem no final da linha.

Nas Fig. 4.19, Fig. 4.20, Fig. 4.21 e Fig. 4.22 são apresentados os modelos do sistema proposto no HPSim e no AS (os códigos em XML das 3 estações do AS estão no anexos C.4, C.5 e C.6).

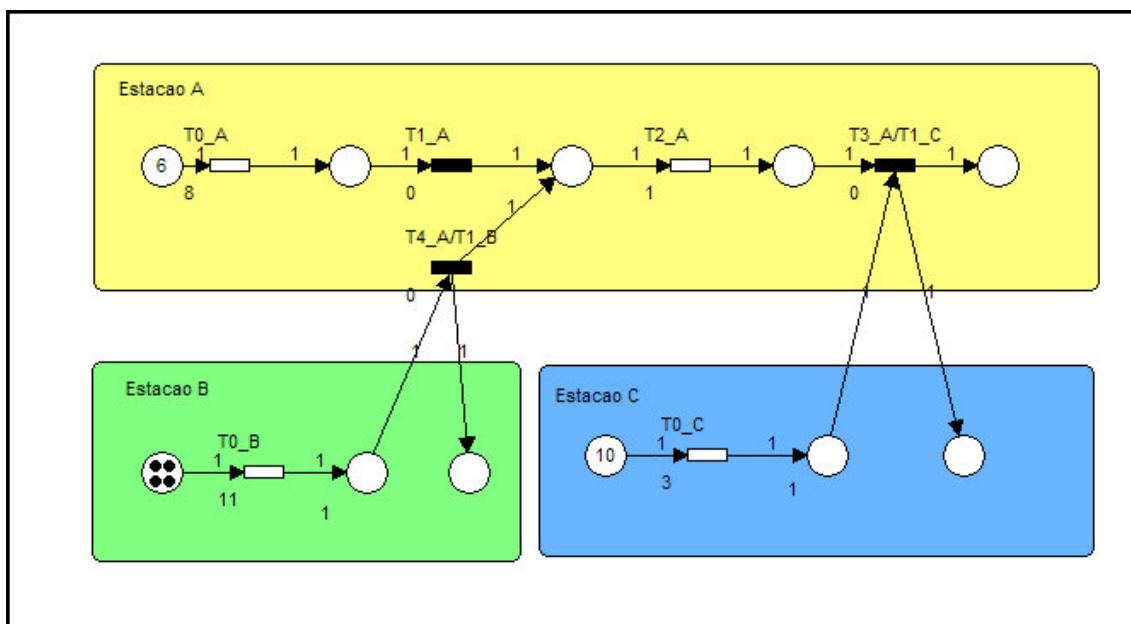


Fig. 4.19 – Rede modelada no HPSim

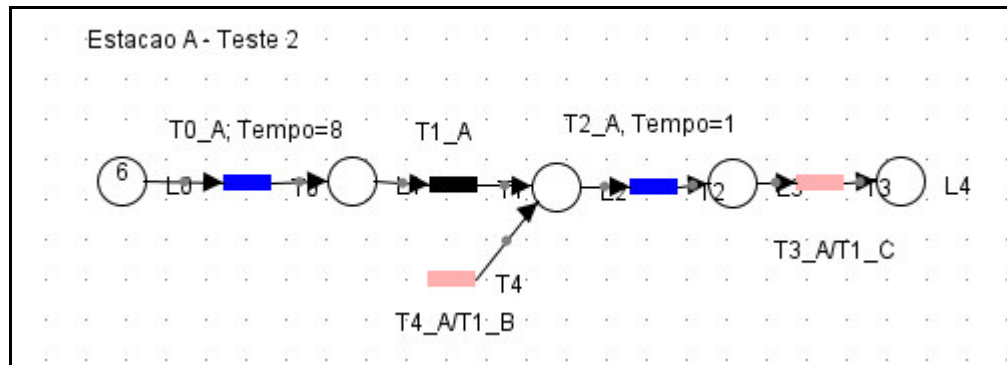


Fig. 4.20 – Estação A modelada no AS

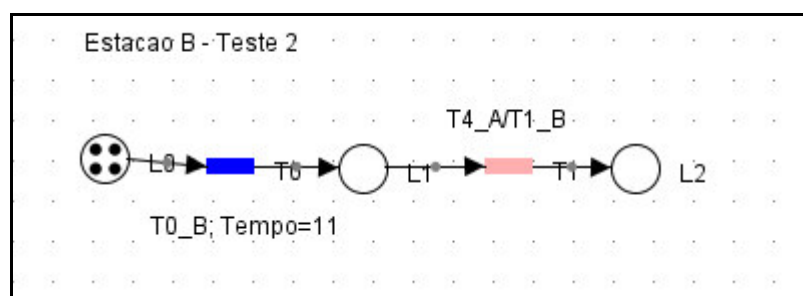


Fig. 4.21 – Estação B modelada no AS

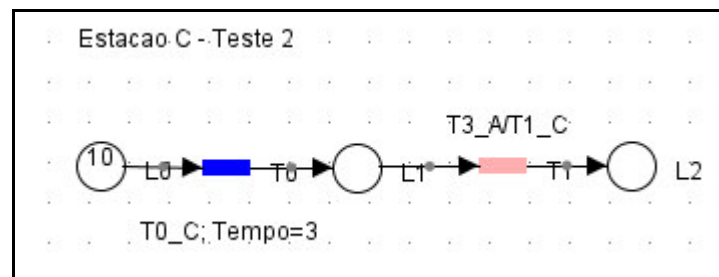


Fig. 4.22 – Estação C modelada no AS

Na Tabela 4.1 serão apresentadas a ordem de disparos de todas as transições, bem como o tempo de simulação onde eles ocorrem. Dessa forma, é possível verificar que as respostas obtidas nas duas simulações foram idênticas. No ANEXO D – Arquivos de Resultados, serão apresentados o arquivo de resultados que o AS imprime para que o usuário tenha outra forma de acompanhar a evolução da simulação, além da interface gráfica.

Tabela 4.1 – Comparação da ordem de disparos de transições com avanço do tempo

Tempo	HPSim			AS		
	EstacaoA	EstacaoB	EstacaoC	EstacaoA	EstacaoB	EstacaoC
3			T0_C			T0
8	T0_A T1_A			T0 T1		
9	T2_A T3_A/T1_C		T3_A/T1_C	T2 T3_A/T1_C		T1
11	T4_A/T1_B	T0_B T4_A/T1_B		T4_A/T1_B	T0 T1_B/T4_A	
12	T2_A T3_A/T1_C		T0_C T3_A/T1_C	T2 T3_A/T1_C		T0 T1
15			T0_C			T0
16	T0_A T1_A			T0 T1		
17	T2_A T3_A/T1_C		T3_A/T1_C	T2 T3_A/T1_C		T1
20			T0_C			T0
22	T4_A/T1_B	T0_B T4_A/T1_B		T4_A/T1_B	T0 T1_B/T4_A	
23	T2_A T3_A/T1_C		T3_A/T1_C	T2 T3_A/T1_C		T1
24	T0_A T1_A			T0 T1		
25	T2_A			T2		
26	T3_A/T1_C		T0_C T3_A/T1_C	T3_A/T1_C		T0 T1
29			T0_C			T0
32	T0_A T1_A			T0 T1		
33	T4_A/T1_B T2_A T3_A/T1_C	T0_B T4_A/T1_B	T3_A/T1_C	T2 T3_A/T1_C T4_A/T1_B	T0 T1_B/T4_A	T1
34	T2_A			T2		
36	T3_A/T1_C		T0_C T3_A/T1_C	T3_A/T1_C		T0 T1
39			T0_C			T0
40	T0_A T1_A			T0 T1		
41	T2_A T3_A/T1_C		T3_A/T1_C	T2 T3_A/T1_C		T1
44	T4_A/T1_B	T0_B T4_A/T1_B	T0_C	T4_A/T1_B	T0 T1_B/T4_A	T0
45	T2_A T3_A/T1_C		T3_A/T1_C	T2 T3_A/T1_C		T1
48	T0_A T1_A		T0_C	T0 T1		T0
49	T2_A T3_A/T1_C		T3_A/T1_C	T2 T3_A/T1_C		T1

5 Conclusões

A importância da simulação distribuída tem sido destacada em vários trabalhos dada as condições de produção atuais e previstas pela maioria das empresas. Assim, esse trabalho é considerado uma contribuição principalmente para o caso de modelos em RdP.

Nesse contexto, o AS se mostrou uma ferramenta robusta e útil para avaliação de sistemas modelados em diferentes estações de trabalho. A interface e os arquivos de resultado permitem ao usuário acompanhar ou rever a evolução da simulação.

E, além disso, o AS mantém funcionalidades interessantes para simulação em RdP, demonstrando que o conceito de incluir novos módulos, adicionando funções, se mostra uma forma valiosa de se modificar um software, pois permite a implementação de novos conceitos e tecnologias, sem que outras vantagens sejam perdidas. Assim, pode-se dizer que raramente um software chega ao fim de sua evolução, sempre há formas de aprimorá-lo, seja incluindo essas novas funções, seja otimizando as já existentes.

A modelagem de redes Orientadas a Objeto frequentemente se utiliza de diversos objetos compartilhando as mesmas transições, i.e., mais de duas transições fundidas em uma só representação, como no Sistema de Aterrissagem (VILLANI 2004). É patente a necessidade de modificação do código tal que seja possível fundir mais de duas transições por vez.

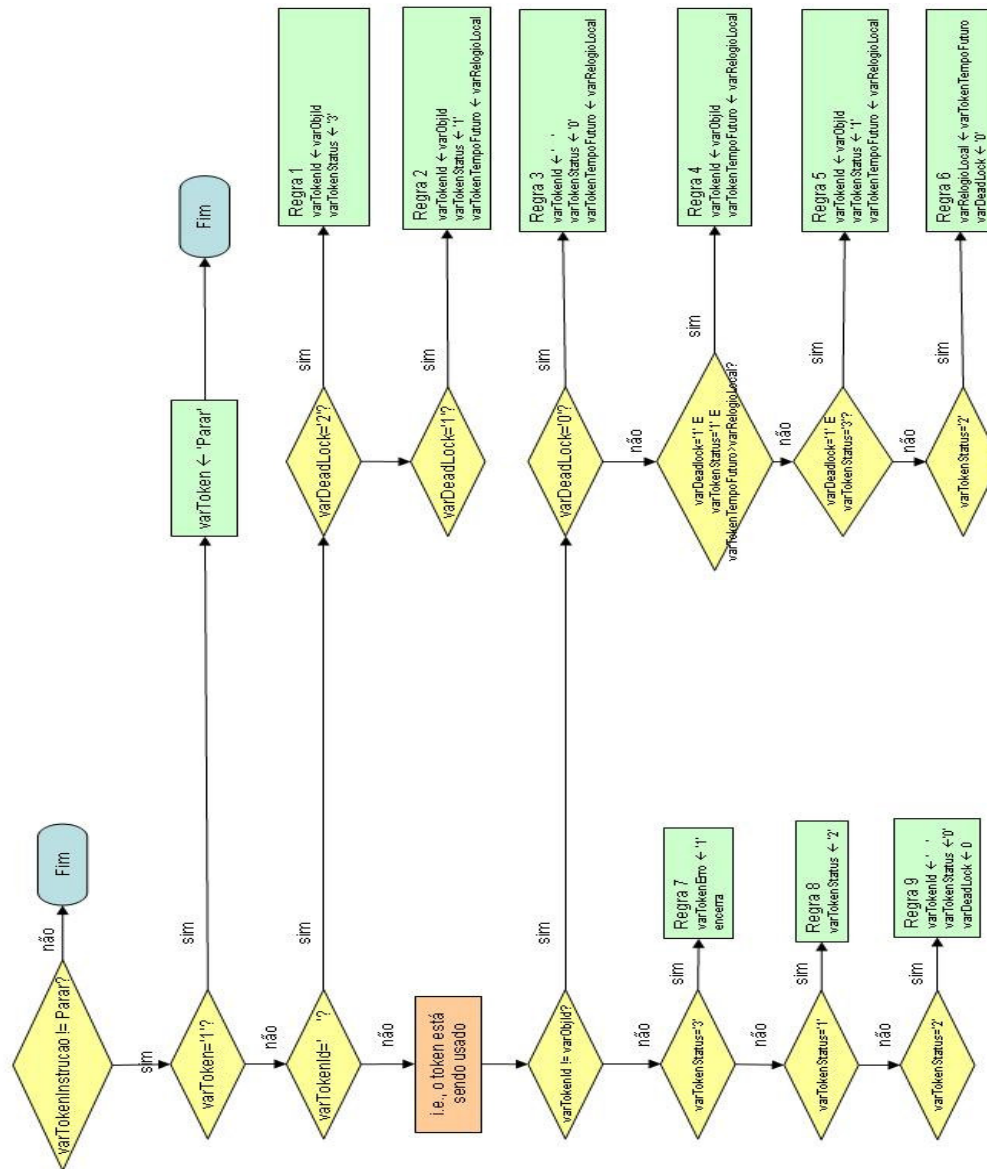
A modelagem de sistemas supervisórios híbridos pode ser objeto de estudo para implementações futuras, desde que o conceito "híbrido" seja incrementado ao código já existente.

Quanto à interface gráfica, é possível a evolução de modo a permitir a edição dos tipos de transição inseridos, modificação do tempo (nas temporizadas) e alteração das transições correspondentes no modelo de fusões. Pode-se pensar, ainda, numa possível interface entre plantas reais e plantas simuladas, ou efetuar o controle de processos produtivos via interface gráfica.

Por fim, para facilitar ainda mais a análise, um módulo que colete as informações sobre o gerenciamento de recursos, disponibilidade e sugestões para melhorias, poderia tornar a simulação mais proveitosa no sentido de tomada de decisões.

ANEXO A – Algoritmo de Verificação/ Avaliação dos Parâmetros Locais

Esse é o algoritmo completo da sequência de avaliação e atualização de parâmetros (deadlock, campos do token e tempo local de simulação), essenciais para a correta evolução da simulação distribuída.



ANEXO B – Códigos Fonte

B.1. TrataTokenRecebe()

Esse código implementa as Regras 6, 7 e 9 do fluxograma do Anexo A, e é acionada ao se receber o token da estação anterior.

```
public void TrataToken_Recebe() {
    char stationName = Conecta.getServidor().stationName;

    if(this.instrucao!=0){ // 0=parar;
        if(this.erro==1){
            //Verificacao de Erros
            //Se foi detectado algum erro, como deadlock de todos modelos, a
            //simulação deve
            //ser interrompida!!!
            this.instrucao=0;//faco parar
            //JOptionPane.showMessageDialog(null, "A Simulação Acabou", "Aviso",
            JOptionPane.INFORMATION_MESSAGE);
        } else if(this.id=='0'){
            //Mantido para que a estrutura do fluxograma seja mantida.
        }
        else if(this.id!=stationName){//testo se eh a MINHA info (meu TOKEN)
            if(this.deadlock==0){
                //Mantido pela mesma razao.
            } else if(this.status==2){
                //REGRA 6
                //O modelo local recebe uma instrução para atualizar seu tempo
                local de simulação. Neste caso,
                //ele deve atualizar o seu relógio local com o tempo informado no
                token e alterar sua variável de
                //deadlock para zero, a fim de disparar suas transições, caso
                exista alguma disparável neste
                //novo tempo local.
                this.setRelogioLocal(this.tempoFuturo);
                this.setDeadlock(0);
            }
        }
        else if(this.status==3){
            //REGRA7
            //O modelo local recebe de volta o token que enviou, sem alterações,
            onde informava aos demais
            //modelos que se encontrava sem mais transições disparáveis. Neste
            caso, ele deve atualizar o
            //campo de erro do token, informando que o sistema como um todo está
            travado e a simulação deve
            //ser encerrada.
            this.erro=1; //comeca a passar o token para dizer que acabou.
        }
        else if(this.status==2){
            //REGRA9
            //O modelo local recebe de volta o token que enviou, os demais modelos
            já atualizaram seus
```

```
        //relógios locais restando apenas ele mesmo. Assim, ele libera o token
e troca o seu status
        //de "deadlock" para "0", ou seja, o modelo não está mais em deadlock,
podendo disparar suas
        //transições.
        this.id='0';
        this.status=0;
        this.setDeadlock(0); //o pessoal concordou, preciso atualizar o MEU
relógio LOCAL
        this.setRelogioLocal(this.tempoFuturo);
    }
    } else { //Chegou ao FIM
    }
}
```

B.2. TrataTokenEnvia()

Esse código implementa as Regras 1, 2, 3, 4, 5 e 8 do fluxograma do Anexo A, e é chamado antes de se enviar o token para a próxima estação.

```
public void TrataTokenEnvia(){
    char stationName = Conecta.getServidor().stationName;

    if(this.instrucao!=0){ // 0=parar;
        if(this.erro==1){
            //Verificacao de Erros
            //Se foi detectado algum erro, como deadlock de todos modelos, a
            //simulação deve
            //ser interrompida!!!
            this.instrucao=0;//faco parar
        } else if(this.id=='0'){
            if(this.deadlock==2){
                //REGRA 1
                //O modelo local está em deadlock pois naum possui mais transições
                //disparáveis
                //(instantaneas ou temporizadas) na sua lista de disparos, neste
                //caso, ele deve avisar
                // os demais modelos que ele está em deadlock.
                this.id=stationName;
                this.status=3; //i.e., estou no deadlock 2.
            }
            else if(this.deadlock==1){
                //REGRA 2
                //O modelo local está em deadlock, pois não possui mais transições
                //instantaneas disparaveis
                //Neste caso, ele deve consultar o status dos demais modelos, afim
                //de evoluir o tempo de simulacao
                //do sistema.
                this.id=stationName;
                this.status=1;
                this.tempoFuturo=this.tempoProximo;
            }
        }
        else if(this.id!=stationName){//testo se eh a MINHA info (meu TOKEN)
            if(this.deadlock==0){
                //REGRA 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 dentre os modelos.
                this.id='0';
                this.status=0;
                this.tempoFuturo=this.relogioLocal;
            } else if(this.deadlock==1 && this.status==1 &&
                (this.tempoFuturo>this.tempoProximo) ){
                //REGRA 4
```

```

        //O modelo local está em deadlock e recebe o token onde um outro
        modelo, com tempo de simulação
        //superior ao local, está fazendo uma consulta. Neste caso, ele
        deve sobrescrever os campos de
        //identificação e tempo futuro com os seus próprios valores,
        passando a fazer a consulta
        //aos demais.
        this.id=stationName;
        this.tempoFuturo=this.tempoProximo;
        if(this.tempoLocalAnterior==this.relogioLocal){
            this.tempoLocalAnterior=this.relogioLocal;
        }
    } else if(this.deadlock==1 && this.status==3){
        //REGRA 5
        //O modelo local está em deadlock e recebe o token onde um outro
        modelo informa que está sem
        //transições disparáveis. Neste caso, ele deve sobrescrever os
        campos de identificação e tempo
        //futuro e trocar o status do token para consulta, informando que
        este modelo não está sem
        //transições na sua lista de disparos.
        this.id=stationName;
        this.status=1;
        this.tempoFuturo= this.tempoProximo;
        if(this.tempoLocalAnterior==this.relogioLocal){
            this.tempoLocalAnterior=this.relogioLocal;
        }
    }
}
else if(this.status==1){
    //REGRA8
    //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,
    //conseqüentemente, o do sistema. Neste caso, ele altera o status do
    token tal que informe as
    //demais estações para atualizarem seus relógios locais de simulação
    com o valor utilizado na
    //consulta.
    this.status=2;
}
}
}
}

```

B.3. ControleToken()

Esse código fonte implementa a função responsável pelo controle da comunicação, separando token de chamadas de método e chamando funções específicas para cada um. Todas as mensagens trocadas entre as estações são avaliadas por essa função.

```
public void ControleToken () {

    String mensagem_recebida = "nichts";

    if      (Conecta.getServidor().stationName=='A'      &&      this.status==-1/*&&
this.primeiraVez==true*/) {
        this.primeiraVez = false;
        mensagem_recebida ="token 0 0 0 1 0";

    } else {
        mensagem_recebida = Conecta.getServidor().recebe();
    }

    if (mensagem_recebida.indexOf("token")>=0){ //se naum tiver dá -1
        destrinchaToken(mensagem_recebida);
    }
    else if(mensagem_recebida.indexOf("chamada")>=0){
        //isChamada vê o destinatário e toma decisao (reenvia ou destrincha)
        isChamada(mensagem_recebida);
        ControleToken();
    }
    else if(mensagem_recebida.indexOf("resposta")>=0){
        Conecta.getServidor().envia(mensagem_recebida);
        //- Reenvia Resposta
        //o recebimento de resposta é dado na PetriNet.java qdo solicitado.
        ControleToken();
    }

}
```

B.4. verificaChamadaDeMetodo

Essa é a função responsável pela verificação das chamadas de método, ou seja, ela retorna o estado (habilitada ou não) das transições, além de disparar a transição caso receba uma ordem de disparo.

```
public boolean verificaChamadaDeMetodo(String transicao, boolean dispara){

    boolean ativa = false;
    System.out.println("Trans: "+transicao);
    System.out.println("dispara= "+dispara);

    for(int i=0; i<transicoes.size();i++){
        if(((TransicaoAbstrata)transicoes.get(i)).getNome().equals(transicao)){
            ativa = ((TransicaoAbstrata)transicoes.get(i)).isAtivaLocal();

            if(ativa==true && dispara==true) { //dispara Trans..." +transicao);
                TransicaoAbstrata t = (TransicaoAbstrata)transicoes.get(i);

                Transicao tf = (Transicao)t;
                System.out.println(tf.getTransName1()+"_"+tf.getStation1()+"/"+
                    tf.getTransName2()+"_"+tf.getStation2()+" " +
                    "'fusion' foi disparada. Tempo= "+Token.getToken().getRelogioLocal());
                Conecta.getServidor().gravacaoResultado(tf.getTransName1()+"_"+tf.getStation1()+"/"+
                    tf.getTransName2()+"_"+tf.getStation2()+" " +
                    "'fusion' foi disparada. Tempo= "+Token.getToken().getRelogioLocal());

                //aqui "dispara"
                t.setAtiva(false);
                t.setAtivaLocal(false);

                if (t instanceof Transicao) {
                    ArrayList arcosativos = t.getArcos();
                    for (int j=0; j<arcosativos.size(); j++) {
                        ArcoAbstrato a = (ArcoAbstrato)arcosativos.get(j);
                        if (a instanceof Arco) {
                            Arco arco = (Arco)a;
                            LugarAbstrato l = arco.getLugar();
                            if (arco.getSentido()==Arco.LUGAR_TRANSICAO) {
                                l.addTokens(-arco.getPeso());
                            } else if (arco.getSentido()==Arco.TRANSICAO_LUGAR) {
                                l.addTokens(arco.getPeso());
                            }
                            atualizasinais(l);
                        }
                    }
                }
                Controle.getControle().repintarJanelaInterna();
            }
        }
    }
    return ativa;
}
```


B.5. EnviaRecebeChamada

A função trata do envio e recebimento de chamadas de método, ou seja, ela envia uma “pergunta”, espera pela “resposta” e trata essa resposta, repassando apenas a informação chave que é a habilitação (ou não) da transição na estação remota.

```
public boolean enviaRecebeChamada(String transicao, char station, boolean dispara){
    boolean ativa;
    String pergunta, resposta;
    //monta a mensagem padrão
    pergunta = "chamada "+stationName+" "+station+" "+transicao+" "+dispara;
    //estacao, codigo, ti etc.
    envia(pergunta);
    Conecta.getServidor().gravacaoResultado("Estacao:
"+Conecta.getServidor().stationName+" envia '"+pergunta+"'");
    resposta = recebe(); //a resposta= "resposta remetente destinatario
ativa/inativa"
    Conecta.getServidor().gravacaoResultado("Estacao:
"+Conecta.getServidor().stationName+" recebe '"+resposta+"'");
    //destrinchando a mensagem
    resposta = resposta.trim(); //cleanin'up
    //resposta="resposta A B sim";
    //resposta
    String resp_chamada = resposta.substring(0, resposta.indexOf(" ")).trim();
    // remetente destinatario ativa/inativa
    // A B sim
    String info = resposta.substring(resposta.indexOf(" ")); //pegando a posicao do
    " "
    info = info.trim();
    int i = info.indexOf(" ");
    //A
    String remet = info.substring(0, i);
    //destinatario ativa/inativa
    info = info.substring(i);
    info = info.trim();
    i = info.indexOf(" ");
    //destinatario
    String dest = info.substring(0,i);
    info = info.trim();
    i = info.indexOf(" ");
    String ativa_inativa = info.substring(i).trim();
    if (ativa_inativa.equals("true")) ativa=true;
    else ativa=false;
    return ativa;
}
```

B.6. ExecutaRede.passo()

Essa função é a que controla o fluxo básico do programa, desde a avaliação das transições ativas localmente até as comunicações via token ou chamada de método.

```
public void passo() {
    Controle.getControle().refresh_tempo();//imprimir tempo
    painel.repaint(); //para mostrar mudanças ocorridas via chamada de método...
    painel.paintImmediately(0,0,1280,800);
    rede.separarTransicoesAtivas(); //para rodar analise de deadlock...
    Token.getToken().TrataToken_Recebe();
    if (rede.passo()) {
        ArrayList lugares = rede.getLugares();
        ArrayList elementos = painel.getElementosGraficos();
        for (int i=0; i<lugares.size(); i++) {
            Lugar l = (Lugar)lugares.get(i);
            String nome = l.getNome();
            JLugar jlugar = null;
            for (int j=0; j<elementos.size(); j++) {
                ElementoGrafico eg = (ElementoGrafico)elementos.get(j);
                if (eg instanceof JLugar) {
                    JLugar jl = (JLugar)eg;
                    if (jl.getNome().equals(nome)) jlugar = jl;
                }
            }
            jlugar.setTokens(l.getTokens());
        }
        Controle.getControle().atualizarVariaveis();
    }
    Controle.getControle().refresh_tempo();
    painel.repaint();
    painel.paintImmediately(0,0,1280,800);
    Token.getToken().TrataToken_Envia();
    Token.getToken().enviaToken();
    //fica aqui até receber o token...tb checa as chamadas de metodo e dispara...
    //Espera mensagem se a instrucao nao for "parar"
    if (Token.getToken().getInstrucao()!=0) Token.getToken().ControleToken();
}
```

ANEXO C – Códigos em XML (modificados)

Todas as estações modeladas no AS têm seu código em XML, que depois é interpretado pelo programa para visualização dos modelos. Assim, nessa seção são apresentados os códigos de todas as estações apresentadas nos estudos de caso.

C.1. Estação A do Estudo de Caso 1 - EstA.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<net id="rede">
  <texto id="texto" x="36.0" y="20.0" tamanho="12" cor="-16777216">Modelo A -
  "mestre"</texto>
  <place id="L0" x="43.0" y="78.0" desc="L0" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="true" offx="0" offy="0" value="2" />
  </place>
  <place id="L1" x="160.0" y="78.0" desc="L1" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <place id="L2" x="290.0" y="78.0" desc="L2" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <place id="L3" x="415.0" y="81.0" desc="L3" tamanho="12" capacidade="2" offx="0"
  offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <transition id="T0" x="108.0" y="78.0" tamanho="12" type="normal" desc="T0"
  offx="0" offy="0" />
  <transition id="T1" x="229.0" y="58.0" tamanho="12" type="normal" desc="T1"
  offx="0" offy="0" />
  <transition id="T2" x="228.0" y="118.0" tamanho="12" type="normal" desc="T2"
  offx="0" offy="0" />
  <transition id="T3" x="356.0" y="78.0" tamanho="12" type="TT" desc="T3" offx="0"
  offy="0" time="3"/>
  <arc id="a0" sentido="0" source="L0" target="T0" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T0" target="L1" x="0" y="0" type="standard" />
  <arc id="a0" sentido="0" source="L1" target="T1" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T1" target="L2" x="0" y="0" type="standard" />
  <arc id="a0" sentido="0" source="L1" target="T2" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T2" target="L2" x="0" y="0" type="standard" />
  <arc id="a0" sentido="0" source="L2" target="T3" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T3" target="L3" x="0" y="0" type="standard" />
</net>
```

C.2. Estação B do Estudo de Caso 1 – EstB.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<net id="rede">
  <texto id="texto" x="25.0" y="16.0" tamanho="12" cor="-16777216">Modelo B -
"escravo"</texto>
  <place id="L0" x="43.0" y="67.0" desc="L0" tamanho="12" capacidade="1" offx="0"
offy="0">
    <initialMarking marking="true" offx="0" offy="0" value="2" />
  </place>
  <place id="L1" x="210.0" y="66.0" desc="L1" tamanho="12" capacidade="1" offx="0"
offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <place id="L2" x="373.0" y="69.0" desc="L2" tamanho="12" capacidade="2" offx="0"
offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <transition id="T0" x="113.0" y="68.0" tamanho="12" type="TT" desc="T0" offx="0"
offy="0" time="5"/>
  <arc id="a0" sentido="0" source="L0" target="T0" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T0" target="L1" x="0" y="0" type="standard" />
  <transition id="T1" x="290.0" y="67.0" tamanho="12" type="TF" desc="T1" offx="0"
offy="0" s1="B" s2="C" t1="T1" t2="T1"/>
  <arc id="a0" sentido="0" source="L1" target="T1" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T1" target="L2" x="0" y="0" type="standard" />
</net>

```

C.3. Estação C do Estudo de Caso 1 – EstC.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<net id="rede">
  <place id="L0" x="51.0" y="85.0" desc="L0" tamanho="12" capacidade="1" offx="0"
offy="0">
    <initialMarking marking="true" offx="0" offy="0" value="2" />
  </place>
  <place id="L1" x="224.0" y="84.0" desc="L1" tamanho="12" capacidade="1" offx="0"
offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <place id="L2" x="380.0" y="85.0" desc="L2" tamanho="12" capacidade="2" offx="0"
offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <transition id="T0" x="148.0" y="84.0" tamanho="12" type="normal" desc="T0"
offx="0" offy="0" />
  <transition id="T1" x="310.0" y="86.0" tamanho="12" type="TF" desc="T1" offx="0"
offy="0" s1="C" s2="B" t1="T1" t2="T1"/>
  <arc id="a0" sentido="0" source="L0" target="T0" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T0" target="L1" x="0" y="0" type="standard" />
  <arc id="a0" sentido="0" source="L1" target="T1" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T1" target="L2" x="0" y="0" type="standard" />
  <texto id="texto" x="40.0" y="26.0" tamanho="12" cor="-16777216">Modelo C -
"escravo"</texto>
</net>

```

C.4. Estação A – Estudo de Caso 2

```

<?xml version="1.0" encoding="UTF-8"?>
<net id="rede">
  <place id="L0" x="58.0" y="91.0" desc="L0" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="true" offx="0" offy="0" value="1" />
  </place>
  <place id="L1" x="173.0" y="91.0" desc="L1" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <place id="L2" x="275.0" y="94.0" desc="L2" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <place id="L3" x="363.0" y="92.0" desc="L3" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <place id="L4" x="447.0" y="91.0" desc="L4" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <transition id="T0" x="121.0" y="92.0" tamanho="12" type="TT" desc="T0" offx="0"
  offy="0" time="2"/>
  <transition id="T1" x="224.0" y="93.0" tamanho="12" type="normal" desc="T1"
  offx="0" offy="0" />
  <transition id="T2" x="324.0" y="94.0" tamanho="12" type="normal" desc="T2"
  offx="0" offy="0" />
  <transition id="T3" x="407.0" y="92.0" tamanho="12" type="TF" desc="T3" offx="0"
  offy="0" s1="A" s2="C" t1="T3" t2="T1"/>
  <transition id="T4" x="223.0" y="140.0" tamanho="12" type="TF" desc="T4" offx="0"
  offy="0" s1="A" s2="B" t1="T4" t2="T1"/>
  <arc id="a0" sentido="0" source="L0" target="T0" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T0" target="L1" x="0" y="0" type="standard" />
  <arc id="a0" sentido="0" source="L1" target="T1" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T1" target="L2" x="0" y="0" type="standard" />
  <arc id="a0" sentido="0" source="L2" target="T2" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T2" target="L3" x="0" y="0" type="standard" />
  <arc id="a0" sentido="0" source="L3" target="T3" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T3" target="L4" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T4" target="L2" x="0" y="0" type="standard" />
  <place id="L5" x="98.0" y="139.0" desc="L5" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="true" offx="0" offy="0" value="1" />
  </place>
  <transition id="T5" x="139.0" y="178.0" tamanho="12" type="normal" desc="T5"
  offx="0" offy="0"/>
  <place id="L6" x="167.0" y="138.0" desc="L6" tamanho="12" capacidade="1" offx="0"
  offy="0">

```

```

    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <arc id="a0" sentido="0" source="L5" target="T5" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T5" target="L6" x="0" y="0" type="standard" />
  <arc id="a0" sentido="0" source="L6" target="T4" x="0" y="0" type="standard" />
  <texto id="texto" x="41.0" y="25.0" tamanho="12" cor="-16777216">Estacao A -
Teste 2</texto>
</net>

```

C.5. Estação B – Estudo de Caso 2

```

<?xml version="1.0" encoding="UTF-8"?>
<net id="rede">
  <texto id="texto" x="37.0" y="17.0" tamanho="12" cor="-16777216">Estacao B -
  Teste 2</texto>
  <place id="L0" x="46.0" y="79.0" desc="L0" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="true" offx="0" offy="0" value="1" />
  </place>
  <place id="L1" x="174.0" y="80.0" desc="L1" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <place id="L2" x="311.0" y="80.0" desc="L2" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <transition id="T0" x="109.0" y="83.0" tamanho="12" type="TT" desc="T0" offx="0"
  offy="0" time ="3"/>
  <transition id="T1" x="247.0" y="82.0" tamanho="12" type="TF" desc="T1" offx="0"
  offy="0" s1="B" s2="A" t1="T1" t2="T4"/>
  <arc id="a0" sentido="0" source="L0" target="T0" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T0" target="L1" x="0" y="0" type="standard" />
  <arc id="a0" sentido="0" source="L1" target="T1" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T1" target="L2" x="0" y="0" type="standard" />
</net>

```


C.6. Estação C – Estudo de Caso 2

```
<?xml version="1.0" encoding="UTF-8"?>
<net id="rede">
  <texto id="texto" x="37.0" y="17.0" tamanho="12" cor="-16777216">Estacao C -
  Teste 2</texto>
  <place id="L0" x="46.0" y="79.0" desc="L0" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="true" offx="0" offy="0" value="1" />
  </place>
  <place id="L1" x="174.0" y="80.0" desc="L1" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <place id="L2" x="311.0" y="80.0" desc="L2" tamanho="12" capacidade="1" offx="0"
  offy="0">
    <initialMarking marking="false" offx="0" offy="0" />
  </place>
  <transition id="T0" x="109.0" y="83.0" tamanho="12" type="TT" desc="T0" offx="0"
  offy="0" time="3"/>
  <transition id="T1" x="247.0" y="82.0" tamanho="12" type="TF" desc="T1" offx="0"
  offy="0" s1="C" s2="A" t1="T1" t2="T3"/>
  <arc id="a0" sentido="0" source="L0" target="T0" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T0" target="L1" x="0" y="0" type="standard" />
  <arc id="a0" sentido="0" source="L1" target="T1" x="0" y="0" type="standard" />
  <arc id="a0" sentido="1" source="T1" target="L2" x="0" y="0" type="standard" />
</net>
```

ANEXO D – Arquivos de Resultados

Aqui estão os arquivos que o AS gera para controle e/ou revisão das simulações realizadas. Como no Estudo de Caso 1 os arquivos foram analisados um a um como exemplo, a seguir estão apenas os arquivos do Estudo de caso 2

D.1. Estação A – Estudo de Caso 2

```
Estacao A a conectar-se no localhost
na porta 1729
Socket Conectado.
Estacao A a ouvir na porta 1728
ServerSocket Conectado.
Estacao: A recebe 'token 0 0 0 1 0'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'token A 8 1 1 0'
Estacao: A recebe 'token C 3 1 1 0'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'token C 3 1 1 0'
Estacao: A recebe 'token C 3 2 1 0'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'token C 3 2 1 0'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'token 0 3 0 1 0'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
```

```
Estacao: A envia 'token A 8 1 1 0'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'token A 8 1 1 0'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'token A 8 2 1 0'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'token A 8 2 1 0'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
T0'temporizada' foi disparada. Tempo=
8
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'token 0 8 0 1 0'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
```

Estacao: A recebe 'token B 11 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 T1 foi disparada. Tempo= 8
 Estacao: A envia 'token A 9 1 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token A 9 1 1 0'
 Estacao: A envia 'token A 9 2 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token A 9 2 1 0'
 T2'temporizada' foi disparada. Tempo= 9
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token 0 9 0 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'chamada C A T3 true'
 T3_A/T1_C 'fusion' foi disparada. Tempo= 9
 Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'token C 3 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 3 1 1 0'
 Estacao: A recebe 'token C 3 2 1 0'
 Estacao: A envia 'chamada A B T1 false'

Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 3 2 1 0'
 Estacao: A recebe 'token C 12 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 12 1 1 0'
 Estacao: A recebe 'token B 11 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token B 11 1 1 0'
 Estacao: A recebe 'token B 11 2 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token B 11 2 1 0'
 Estacao: A recebe 'chamada B A T4 false'
 Estacao: A envia 'resposta A B true'
 Estacao: A recebe 'token C 12 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A true'
 Estacao: A envia 'chamada A B T1 true'
 Estacao: A recebe 'resposta B A true'
 T4_A/T1_B 'fusion' foi disparada. Tempo= 11
 Estacao: A envia 'token C 12 1 1 0'
 Estacao: A recebe 'token C 12 2 1 0'
 T2'temporizada' foi disparada. Tempo= 12
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A false'
 Estacao: A envia 'chamada A B T1 false'

Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token B 22 2 1 0'
 Estacao: A recebe 'chamada B A T4 false'
 Estacao: A envia 'resposta A B true'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token 0 22 0 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A true'
 Estacao: A envia 'chamada A B T1 true'
 Estacao: A recebe 'resposta B A true'
 T4_A/T1_B 'fusion' foi disparada.
 Tempo= 22
 Estacao: A envia 'token A 23 1 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token A 23 1 1 0'
 Estacao: A envia 'token A 23 2 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token A 23 2 1 0'
 T2'temporizada' foi disparada. Tempo= 23
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token 0 23 0 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'chamada C A T3 false'

Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'chamada C A T3 true'
 T3_A/T1_C 'fusion' foi disparada.
 Tempo= 23
 Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'token C 20 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 20 1 1 0'
 Estacao: A recebe 'token C 20 2 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 20 2 1 0'
 Estacao: A recebe 'token C 26 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token A 24 1 1 0'
 Estacao: A recebe 'token A 24 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token A 24 2 1 0'
 Estacao: A recebe 'token A 24 2 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 T0'temporizada' foi disparada. Tempo= 24
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token 0 24 0 1 0'
 Estacao: A recebe 'token C 26 1 1 0'

Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 T1 foi disparada. Tempo= 24
 Estacao: A envia 'token A 25 1 1 0'
 Estacao: A recebe 'token A 25 1 1 0'
 Estacao: A envia 'token A 25 2 1 0'
 Estacao: A recebe 'token A 25 2 1 0'
 T2'temporizada' foi disparada. Tempo= 25
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token A 32 1 1 0'
 Estacao: A recebe 'token C 26 1 1 0'
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 26 1 1 0'
 Estacao: A recebe 'token C 26 2 1 0'
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 26 2 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'token 0 26 0 1 0'

Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A C T1 true'
 Estacao: A recebe 'resposta C A true'
 T3_A/T1_C 'fusion' foi disparada. Tempo= 26
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token A 32 1 1 0'
 Estacao: A recebe 'token C 29 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 29 1 1 0'
 Estacao: A recebe 'token C 29 2 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 29 2 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token 0 29 0 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token A 32 1 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'

Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token A 32 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token A 32 2 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token A 32 2 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 T0'temporizada' foi disparada. Tempo= 32
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token 0 32 0 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token B 33 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 T1 foi disparada. Tempo= 32
 Estacao: A envia 'token B 33 1 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token B 33 2 1 0'
 T2'temporizada' foi disparada. Tempo= 33

Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token 0 33 0 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'chamada C A T3 true'
 T3_A/T1_C 'fusion' foi disparada. Tempo= 33
 Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'token C 29 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 29 1 1 0'
 Estacao: A recebe 'token C 29 2 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 29 2 1 0'
 Estacao: A recebe 'token C 36 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 36 1 1 0'
 Estacao: A recebe 'token B 33 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token B 33 1 1 0'
 Estacao: A recebe 'token B 33 2 1 0'

Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token B 33 2 1 0'
 Estacao: A recebe 'chamada B A T4 false'
 Estacao: A envia 'resposta A B true'
 Estacao: A recebe 'token C 36 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A true'
 Estacao: A envia 'chamada A B T1 true'
 Estacao: A recebe 'resposta B A true'
 T4_A/T1_B 'fusion' foi disparada.
 Tempo= 33
 Estacao: A envia 'token A 34 1 1 0'
 Estacao: A recebe 'token A 34 1 1 0'
 Estacao: A envia 'token A 34 2 1 0'
 Estacao: A recebe 'token A 34 2 1 0'
 T2'temporizada' foi disparada. Tempo= 34
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token A 40 1 1 0'
 Estacao: A recebe 'token C 36 1 1 0'
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 36 1 1 0'
 Estacao: A recebe 'token C 36 2 1 0'
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'

Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 36 2 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'token 0 36 0 1 0'
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A C T1 true'
 Estacao: A recebe 'resposta C A true'
 T3_A/T1_C 'fusion' foi disparada.
 Tempo= 36
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token A 40 1 1 0'
 Estacao: A recebe 'token C 39 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 39 1 1 0'
 Estacao: A recebe 'token C 39 2 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 39 2 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token 0 39 0 1 0'

Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 39 2 1 0'
 Estacao: A recebe 'token C 44 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 44 1 1 0'
 Estacao: A recebe 'token C 44 2 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A true'
 Estacao: A envia 'chamada A B T1 true'
 Estacao: A recebe 'resposta B A true'
 T4_A/T1_B 'fusion' foi disparada.
 Tempo= 44
 Estacao: A envia 'token C 44 1 1 0'
 Estacao: A recebe 'token C 44 2 1 0'
 Estacao: A envia 'token C 44 2 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token 0 44 0 1 0'
 Estacao: A envia 'token A 45 1 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token A 45 1 1 0'
 Estacao: A envia 'token A 45 2 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'

Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C false'
 Estacao: A recebe 'token A 45 2 1 0'
 T2'temporizada' foi disparada. Tempo= 45
 Estacao: A envia 'chamada A C T1 false'
 Estacao: A recebe 'resposta C A true'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token 0 45 0 1 0'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'chamada C A T3 false'
 Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'chamada C A T3 true'
 T3_A/T1_C 'fusion' foi disparada.
 Tempo= 45
 Estacao: A envia 'resposta A C true'
 Estacao: A recebe 'token C 44 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 44 1 1 0'
 Estacao: A recebe 'token C 44 2 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 44 2 1 0'
 Estacao: A recebe 'token C 48 1 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'
 Estacao: A envia 'token C 48 1 1 0'
 Estacao: A recebe 'token C 48 2 1 0'
 Estacao: A envia 'chamada A B T1 false'
 Estacao: A recebe 'resposta B A false'

```

Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
T0'temporizada' foi disparada. Tempo=
48
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'token 0 48 0 1 0'
Estacao: A recebe 'token C 48 1 1 0'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
T1 foi disparada. Tempo= 48
Estacao: A envia 'token C 48 1 1 0'
Estacao: A recebe 'token C 48 2 1 0'
Estacao: A envia 'token C 48 2 1 0'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'token 0 48 0 1 0'
Estacao: A envia 'token A 49 1 1 0'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'token A 49 1 1 0'
Estacao: A envia 'token A 49 2 1 0'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C false'
Estacao: A recebe 'token A 49 2 1 0'
T2'temporizada' foi disparada. Tempo=
49
Estacao: A envia 'chamada A C T1
false'
Estacao: A recebe 'resposta C A true'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'token 0 49 0 1 0'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C true'
Estacao: A recebe 'chamada C A T3
false'
Estacao: A envia 'resposta A C true'
Estacao: A recebe 'chamada C A T3
true'

```

```

T3_A/T1_C 'fusion' foi disparada.
Tempo= 49
Estacao: A envia 'resposta A C true'
Estacao: A recebe 'token B 49 3 1 0'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'chamada A B T1
false'
Estacao: A recebe 'resposta B A false'
Estacao: A envia 'token B 49 3 1 0'
Estacao: A recebe 'token B 49 3 0 1'
Estacao: A envia 'token B 49 3 0 1'
Socket Closed.
ServerSocket Closed.
Estacao A a conectar-se no localhost
na porta 1729
Socket Conectado.
Estacao A a ouvir na porta 1728
ServerSocket Conectado.

```



```
Estacao: B recebe 'token 0 32 0 1 0'
Estacao: B envia 'token B 33 1 1 0'
Estacao: B recebe 'chamada A B T1 false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1 false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token B 33 1 1 0'
Estacao: B envia 'token B 33 2 1 0'
Estacao: B recebe 'chamada A B T1 false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token 0 33 0 1 0'
Estacao: B envia 'token B 33 1 1 0'
Estacao: B recebe 'chamada A B T1 false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1 false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token C 29 1 1 0'
Estacao: B envia 'token C 29 1 1 0'
Estacao: B recebe 'chamada A B T1 false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1 false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token C 29 2 1 0'
Estacao: B envia 'token C 29 2 1 0'
Estacao: B recebe 'chamada A B T1 false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1 false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token C 36 1 1 0'
Estacao: B envia 'token B 33 1 1 0'
Estacao: B recebe 'chamada A B T1 false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1 false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1 false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token B 33 1 1 0'
Estacao: B envia 'token B 33 2 1 0'
Estacao: B recebe 'chamada A B T1 false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1 false'
```

```

Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token B 33 2 1 0'
T0'temporizada' foi disparada. Tempo=
33
Estacao: B envia 'chamada B A T4
false'
Estacao: B recebe 'resposta A B true'
Estacao: B envia 'token 0 33 0 1 0'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A true'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A true'
Estacao: B recebe 'chamada A B T1
true'
T1_B/T4_A 'fusion' foi disparada.
Tempo= 33
Estacao: B envia 'resposta B A true'
Estacao: B recebe 'token A 34 1 1 0'
Estacao: B envia 'token A 34 1 1 0'
Estacao: B recebe 'token A 34 2 1 0'
Estacao: B envia 'token A 34 2 1 0'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token A 40 1 1 0'
Estacao: B envia 'token A 40 1 1 0'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token C 36 1 1 0'
Estacao: B envia 'token C 36 1 1 0'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token A 40 1 1 0'
Estacao: B envia 'token A 40 1 1 0'

```



```

T1_B/T4_A 'fusion' foi disparada.
Tempo= 44
Estacao: B envia 'resposta B A true'
Estacao: B recebe 'token C 44 1 1 0'
Estacao: B envia 'token C 44 1 1 0'
Estacao: B recebe 'token C 44 2 1 0'
Estacao: B envia 'token C 44 2 1 0'
Estacao: B recebe 'token A 45 1 1 0'
Estacao: B envia 'token A 45 1 1 0'
Estacao: B recebe 'token A 45 2 1 0'
Estacao: B envia 'token A 45 2 1 0'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token 0 45 0 1 0'
Estacao: B envia 'token B 45 3 1 0'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token C 44 1 1 0'
Estacao: B envia 'token C 44 1 1 0'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token C 44 2 1 0'
Estacao: B envia 'token C 44 2 1 0'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token C 48 1 1 0'
Estacao: B envia 'token C 48 1 1 0'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token 0 48 0 1 0'
Estacao: B envia 'token B 48 3 1 0'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'

```

```

Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token C 48 1 1 0'
Estacao: B envia 'token C 48 1 1 0'
Estacao: B recebe 'token C 48 2 1 0'
Estacao: B envia 'token C 48 2 1 0'
Estacao: B recebe 'token A 49 1 1 0'
Estacao: B envia 'token A 49 1 1 0'
Estacao: B recebe 'token A 49 2 1 0'
Estacao: B envia 'token A 49 2 1 0'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token 0 49 0 1 0'
Estacao: B envia 'token B 49 3 1 0'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'chamada A B T1
false'
Estacao: B envia 'resposta B A false'
Estacao: B recebe 'token B 49 3 1 0'
Estacao: B envia 'token B 49 3 0 1'
Socket Closed.
ServerSocket Closed.
Estacao B a ouvir na porta 1729
ServerSocket Conectado.
Estacao B a conectar-se no localhost
na porta 1730
Socket Conectado.

```

D.3. Estação C – Estudo de Caso 2

```

Estacao C a ouvir na porta 1730
ServerSocket Conectado.
Estacao C a conectar-se no localhost
na porta 1728
Socket Conectado.
Estacao: C recebe 'token A 8 1 1 0'
Estacao: C envia 'token C 3 1 1 0'
Estacao: C recebe 'token C 3 1 1 0'
Estacao: C envia 'token C 3 2 1 0'
Estacao: C recebe 'token C 3 2 1 0'
T0'temporizada' foi disparada. Tempo=
3
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token 0 3 0 1 0'
Estacao: C recebe 'token A 8 1 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 8 1 1 0'
Estacao: C recebe 'token A 8 2 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 8 2 1 0'
Estacao: C recebe 'token B 11 1 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token B 11 1 1 0'
Estacao: C recebe 'token A 9 1 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 9 1 1 0'

```

```

Estacao: C recebe 'token A 9 2 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 9 2 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'token B 11 1 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C true'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C true'
Estacao: C envia 'chamada C A T3 true'
Estacao: C recebe 'resposta A C true'
T1_C/T3_A 'fusion' foi disparada.
Tempo= 9
Estacao: C envia 'token C 3 1 1 0'
Estacao: C recebe 'token C 3 1 1 0'
Estacao: C envia 'token C 3 2 1 0'
Estacao: C recebe 'token C 3 2 1 0'
Estacao: C envia 'token C 12 1 1 0'
Estacao: C recebe 'token B 11 1 1 0'
Estacao: C envia 'token B 11 1 1 0'
Estacao: C recebe 'token B 11 2 1 0'
Estacao: C envia 'token B 11 2 1 0'
Estacao: C recebe 'token 0 11 0 1 0'
Estacao: C envia 'token C 12 1 1 0'
Estacao: C recebe 'token C 12 1 1 0'
Estacao: C envia 'token C 12 2 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'
Estacao: C recebe 'token C 12 2 1 0'
T0'temporizada' foi disparada. Tempo=
12
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C true'
Estacao: C envia 'token 0 12 0 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'chamada A C T1
true'
T1_C/T3_A 'fusion' foi disparada.
Tempo= 12
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'token A 16 1 1 0'

```

```

Estacao: C envia 'token C 15 1 1 0'
Estacao: C recebe 'token C 15 11 1 0'
Estacao: C envia 'token C 15 2 1 0'
Estacao: C recebe 'token C 15 2 1 0'
T0'temporizada' foi disparada. Tempo=
15
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token 0 15 0 1 0'
Estacao: C recebe 'token A 16 1 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 16 1 1 0'
Estacao: C recebe 'token A 16 2 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 16 2 1 0'
Estacao: C recebe 'token B 22 1 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token B 22 1 1 0'
Estacao: C recebe 'token A 17 1 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 17 1 1 0'
Estacao: C recebe 'token A 17 2 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'

```

```

Estacao: C envia 'token A 17 2 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'token B 22 1 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C true'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C true'
T1_C/T3_A 'fusion' foi disparada.
Tempo= 17
Estacao: C envia 'token C 15 1 1 0'
Estacao: C recebe 'token C 15 1 1 0'
Estacao: C envia 'token C 15 2 1 0'
Estacao: C recebe 'token C 15 2 1 0'
Estacao: C envia 'token C 20 1 1 0'
Estacao: C recebe 'token C 20 1 1 0'
Estacao: C envia 'token C 20 2 1 0'
Estacao: C recebe 'token C 20 2 1 0'
T0'temporizada' foi disparada. Tempo=
20
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token 0 20 0 1 0'
Estacao: C recebe 'token B 22 1 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token B 22 1 1 0'
Estacao: C recebe 'token B 22 2 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token B 22 2 1 0'
Estacao: C recebe 'token 0 22 0 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token 0 22 0 1 0'
Estacao: C recebe 'token A 23 1 1 0'

```

```

Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 23 1 1 0'
Estacao: C recebe 'token A 23 2 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 23 2 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'token B 33 1 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C true'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C true'
Estacao: C envia 'chamada C A T3 true'
Estacao: C recebe 'resposta A C true'
T1_C/T3_A 'fusion' foi disparada.
Tempo= 23
Estacao: C envia 'token C 20 1 1 0'
Estacao: C recebe 'token C 20 1 1 0'
Estacao: C envia 'token C 20 2 1 0'
Estacao: C recebe 'token C 20 2 1 0'
Estacao: C envia 'token C 26 1 1 0'
Estacao: C recebe 'token A 24 1 1 0'
Estacao: C envia 'token A 24 1 1 0'
Estacao: C recebe 'token A 24 2 1 0'
Estacao: C envia 'token A 24 2 1 0'
Estacao: C recebe 'token B 33 1 1 0'
Estacao: C envia 'token C 26 1 1 0'
Estacao: C recebe 'token A 25 1 1 0'
Estacao: C envia 'token A 25 1 1 0'
Estacao: C recebe 'token A 25 2 1 0'
Estacao: C envia 'token A 25 2 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'
Estacao: C recebe 'token A 32 1 1 0'
Estacao: C envia 'token C 26 1 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'

Estacao: C recebe 'token C 26 1 1 0'
Estacao: C envia 'token C 26 2 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'
Estacao: C recebe 'token C 26 2 1 0'
T0'temporizada' foi disparada. Tempo=
26
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C true'
Estacao: C envia 'token 0 26 0 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'chamada A C T1
true'
T1_C/T3_A 'fusion' foi disparada.
Tempo= 26
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'token A 32 1 1 0'
Estacao: C envia 'token C 29 1 1 0'
Estacao: C recebe 'token C 29 1 1 0'
Estacao: C envia 'token C 29 2 1 0'
Estacao: C recebe 'token C 29 2 1 0'
T0'temporizada' foi disparada. Tempo=
29
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token 0 29 0 1 0'
Estacao: C recebe 'token A 32 1 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 32 1 1 0'
Estacao: C recebe 'token A 32 2 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 32 2 1 0'
Estacao: C recebe 'token B 33 1 1 0'

```

```

Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token B 33 1 1 0'
Estacao: C recebe 'token B 33 2 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token B 33 2 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'token B 33 1 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C true'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C true'
Estacao: C envia 'chamada C A T3 true'
Estacao: C recebe 'resposta A C true'
T1_C/T3_A 'fusion' foi disparada.
Tempo= 33
Estacao: C envia 'token C 29 1 1 0'
Estacao: C recebe 'token C 29 1 1 0'
Estacao: C envia 'token C 29 2 1 0'
Estacao: C recebe 'token C 29 2 1 0'
Estacao: C envia 'token C 36 1 1 0'
Estacao: C recebe 'token B 33 1 1 0'
Estacao: C envia 'token B 33 1 1 0'
Estacao: C recebe 'token B 33 2 1 0'
Estacao: C envia 'token B 33 2 1 0'
Estacao: C recebe 'token 0 33 0 1 0'
Estacao: C envia 'token C 36 1 1 0'
Estacao: C recebe 'token A 34 1 1 0'
Estacao: C envia 'token A 34 1 1 0'
Estacao: C recebe 'token A 34 2 1 0'
Estacao: C envia 'token A 34 2 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'
Estacao: C recebe 'token A 40 1 1 0'
Estacao: C envia 'token C 36 1 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'

Estacao: C recebe 'token C 36 1 1 0'
Estacao: C envia 'token C 36 2 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A false'
Estacao: C recebe 'token C 36 2 1 0'
T0'temporizada' foi disparada. Tempo=
36
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C true'
Estacao: C envia 'token 0 36 0 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'chamada A C T1
true'
T1_C/T3_A 'fusion' foi disparada.
Tempo= 36
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'token A 40 1 1 0'
Estacao: C envia 'token C 39 1 1 0'
Estacao: C recebe 'token C 39 1 1 0'
Estacao: C envia 'token C 39 2 1 0'
Estacao: C recebe 'token C 39 2 1 0'
T0'temporizada' foi disparada. Tempo=
39
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token 0 39 0 1 0'
Estacao: C recebe 'token A 40 1 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 40 1 1 0'
Estacao: C recebe 'token A 40 2 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 40 2 1 0'
Estacao: C recebe 'token B 44 1 1 0'

```



```

Estacao: C envia 'token A 49 1 1 0'
Estacao: C recebe 'token A 49 2 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C false'
Estacao: C envia 'token A 49 2 1 0'
Estacao: C recebe 'chamada A C T1
false'
Estacao: C envia 'resposta C A true'
Estacao: C recebe 'token B 49 3 1 0'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C true'
Estacao: C envia 'chamada C A T3
false'
Estacao: C recebe 'resposta A C true'
Estacao: C envia 'chamada C A T3 true'
Estacao: C recebe 'resposta A C true'
T1_C/T3_A 'fusion' foi disparada.
Tempo= 49
Estacao: C envia 'token B 49 3 1 0'
Estacao: C recebe 'token B 49 3 0 1'
Estacao: C envia 'token B 49 3 0 1'
Socket Closed.
ServerSocket Closed.
Estacao C a ouvir na porta 1730
ServerSocket Conectado.
Estacao C a conectar-se no localhost
na porta 1728
Socket Conectado.

```

Referências Bibliográficas

ABB. About ABB. Disponível em <

<http://www.abb.com/cawp/abbzh252/93ACCD6F33415725C1256AE7004E7188.aspx?v=71836&e=us>>. Acesso em 16 de junho de 2007.

BARROS, J.P.; GOMES, L.. Operational PNML: Towards a PNML Support for Model Construction and Modification. **Workshop on the Definition, Implementation and Application of a Standard Interchange Format for Petri Nets (Satellite event at the International Conference on Application and Theory of Petri Nets)**. Bolonha, Itália, 2004.

BARROS, J.P.; GOMES, L.. Structuring and Composability Issues in Petri Nets Modeling. **IEEE Transactions On Industrial Informatics**. Vol.1, No2. 2, May 2005.

CARDOSO, J. ; VALETTE, R. **Redes de Petri**. Florianópolis: Editora da UFSC, 1997. 212p.

CHENG, J.; CHENG, L.. Dispersed Networked Manufacturing Mode and Its Application in China. **IEEE International Conference on Industrial Informatics**. 2006

CURY, J. E. R.. Teoria de Controle Supervisório de Sistemas a Eventos Discretos. **V Simpósio Brasileiro de Automação Inteligente**. Canela – RS, 2001. (acessado em 16 de junho de 2007 em <<http://www.das.ufsc.br/~cury/cursos/apostila.pdf>>).

FREITAS, R. G.; SILVA, V. A. **Ferramenta Baseada em Rede de Petri para Modelagem, Simulação, Programação e Supervisão de Sistemas de Automação**. 2006. 84p. Trabalho de Formatura – EPUSP. São Paulo, 2006.

FUJIMOTO, Richard M.. Distributed Simulation Systems. **Proceedings of the 2003 Winter Simulation Conference**. - College of Computing, Georgia Institute of Technology. EUA, 2003.

JAVA PLATFORM, Standard Edition 6 API Specification. Class ServerSocket. <<http://java.sun.com/javase/6/docs/api/java/net/ServerSocket.html>> e Class Socket

<<http://java.sun.com/javase/6/docs/api/java/net/Socket.html>>; Acesso em 20 de setembro de 2007.

Java/ Web Developer® <<http://www.webdeveloper.com/java/>>. Acesso em 15 de junho de 2007.

JÜNGEL, M; KINDLER, E.; WEBER, M. The Petri Net Markup Language. Berlim, 2000. (acesso em 15 de junho de 2007: <http://www2.informatik.hu-berlin.de/top/pnml/about.html>).

JUNQUEIRA, F. **Modelagem e Simulação Distribuída de Sistemas Produtivos**. 2006. 222p. Tese de Doutorado – Escola Politécnica da Universidade de São Paulo. São Paulo, 2006.

MAZZARIOL, L. M. Ferramenta computacional, baseada em lógica linear, para a análise de sistemas a eventos discretos. **13º Simpósio Internacional de Iniciação Científica da USP (SIICUSP) / CICTE 2005** [Resumo]. São Carlos, 2005.

MIYAGI, P. E. **Controle Programável – Fundamentos do Controle de Sistemas a Eventos Discretos**. São Paulo: Editora Edgard Blücher, 1996.

MIYAGI, P. E. , **Introdução a Simulação Discreta**, EPUSP [Apostila] São Paulo – 2006.

MONOSTORI, L.; VÁNCZA, J.; KUMARA, S. R. T. Agent-Based Systems for Manufacturing. **Annals of the CIRP**, 2006.

NetBeans Integrated Development Environment (IDE). <<http://www.netbeans.org/>>. Acesso em 15 de junho de 2007.

NUNES, L.R.. Sockets em Java. Artigo sobre *networking* em Java. 2004. Disponível em <<http://www.sumersoft.com/publicacoes/SocketsEmJAVA.pdf>>. Acesso em 20 de setembro de 2007.

PANDOLFI, C. K. **Comunicação entre Programas para a Simulação Distribuída de Sistemas Produtivos**. 2005. 126p. Trabalho de Formatura – EPUSP. São Paulo, 2005.

Petri Net Markup Language. <<http://www2.informatik.hu-berlin.de/top/pnml/>>. Acesso em 15 de junho de 2007.

POST, R.D.J; WERF, J.M.E.M van der. EPNML 1.1 - an XML format for Petri nets. TU Eindhoven, 2004. (acessado em 15 de junho de 2007: <http://www.petriweb.org/specs/>).

SUN MICROSYSTEMS. The Source for Java Developers. <<http://java.sun.com>>. Acesso em 15 de junho de 2007.

TextPad – The Editor for Windows. < <http://www.textpad.com/>>. Acesso em 15 de junho de 2007.

TOYOTA. Globalizing and Localizing Manufacturing. Disponível em < <http://www.toyota.co.jp/en/vision/globalization/subsidiaries.html>>. Acesso em 16 de junho de 2007.

VILLANI, E. **Modelagem e Análise de Sistemas Supervisórios Híbridos**. 2004. 339p. Tese de Doutorado – Escola Politécnica da Universidade de São Paulo. São Paulo - 2004.

ZHANG, Y.; GREGORY, M.; SHI, Y.. Foundations of Global Engineering Networks: Essential Characteristics of Effective Engineering Networks. **IEEE International Conference on Management of Innovation and Technology**, 2006