

Capítulo 5

O PETRILab

Apresentamos todos os conceitos necessários para o entendimento do funcionamento do PETRILab. Vamos neste capítulo apresentar o programa em si, sua arquitetura básica, sua interface, seus recursos e sua forma de utilização. Este capítulo está organizado da seguinte forma: na seção 5.1 será apresentada uma breve história do programa; na seção 5.2 será mostrada uma visão geral de seus recursos; a seção 5.3 contém o guia de utilização do usuário; por fim, na seção 5.4, é exibida uma lista de teclas de atalho do programa.

5.1 Breve História

A elaboração do PETRILab começou no segundo semestre de 2013, como um projeto de iniciação científica do Laboratório de Controle e Automação (LCA) da Universidade Federal do Rio de Janeiro (UFRJ). A proposta era que fosse desenvolvido um meio automático de realizar a conversão proposta no Capítulo 4 deste trabalho: a conversão de uma RPIC em um diagrama Ladder.

A linguagem de programação escolhida para a criação do programa foi o Python, por ser uma linguagem orientada a objetos, simples e robusta. Em suas primeiras versões, o PETRILab não contava com interface gráfica; a modelagem de Redes de Petri era feita por linha de comando, assim como a execução de eventos e mudança de condições. Uma pequena parte do código da classe de RPIC criada no Python pode ser vista na Figura 5-1. Não é o objetivo deste trabalho explicar o código completo do programa – já que o

programa completo tem mais de 3.000 linhas. No entanto, o Apêndice A explica a arquitetura básica do programa, para permitir futuro desenvolvimento por outra pessoa.

```
class CIPN():
    def __init__(self, Pc, Tc, Prec, Postc, x0c, Inc, C, Ec, lc, D, ld, A, la, parent=None):
        self.parent = parent
        self.x0c = x0c
        self.Pc, self.Tc = Pc, Tc
        self.Prec, self.Postc, self.Inc = Prec, Postc, Inc
        self.C, self.Ec, self.lc, self.D = C, Ec, lc, D
        self.ld, self.A, self.la = ld, A, la
        self.x, self.toFire = x0c, [False]*Tc
        self.timers = []
        for t in range(Tc):
            self.timers.append([])
            self.eimpulse = [1]*len(Ec)
        def setImpulse(self, e, x):
            self.eimpulse[e]=x
        def isFireable(self, t):
            for i in range(self.Pc):
                if self.x[i]<self.Prec[i][t]:
                    return False
                if self.Inc[i][t]:
                    if self.x[i]>=self.Inc[i][t]:
                        return False
            if not self.C[self.lc[t][0]] or not self.Ec[self.lc[t][1]]:
                return False
            return True
        def setFireable(self):
```

Figura 5-1: Parte do código do PETRILab

Na versão inicial, para criar uma RPIC era necessário criar um objeto da classe CIPN, que tem métodos para a exibição, execução de eventos e mudança de condições da rede. A Figura 5-2 mostra como era feita a criação de uma rede simples com: dois lugares p_1 e p_2 ; uma transição t_1 ; um arco de p_1 a t_1 e outro de t_1 a p_2 , ambos com peso 1; um evento e_1 sincronizado com a transição t_1 ; e marcação inicial [2 0]. Ao disparar o evento de índice 0 (e_1), vemos que a marcação da rede muda para [1 1], como esperado.

Posteriormente, mesmo ainda não tendo sido feita uma interface gráfica, foi implementada a função de conversão em diagramas Ladder de acordo com o método proposto no Capítulo 4 deste trabalho. Um trecho do código de conversão pode ser visto na Figura 5-3; ele converte a rede em objetos do diagrama – criados em um outro módulo –, passando como argumento os rótulos correspondentes.

O diagrama Ladder podia então ser exibido “desenhado em texto”, conforme mostra a Figura 5-4. A formatação era precária, e o texto gerado não muito intuitivo, mas o diagrama correspondia ao método de conversão proposto. Na figura, [] representa um contato; () representa uma bobina, que terá um S dentro se for do tipo Set; < > representa um bloco de função; e || representa itens paralelos.

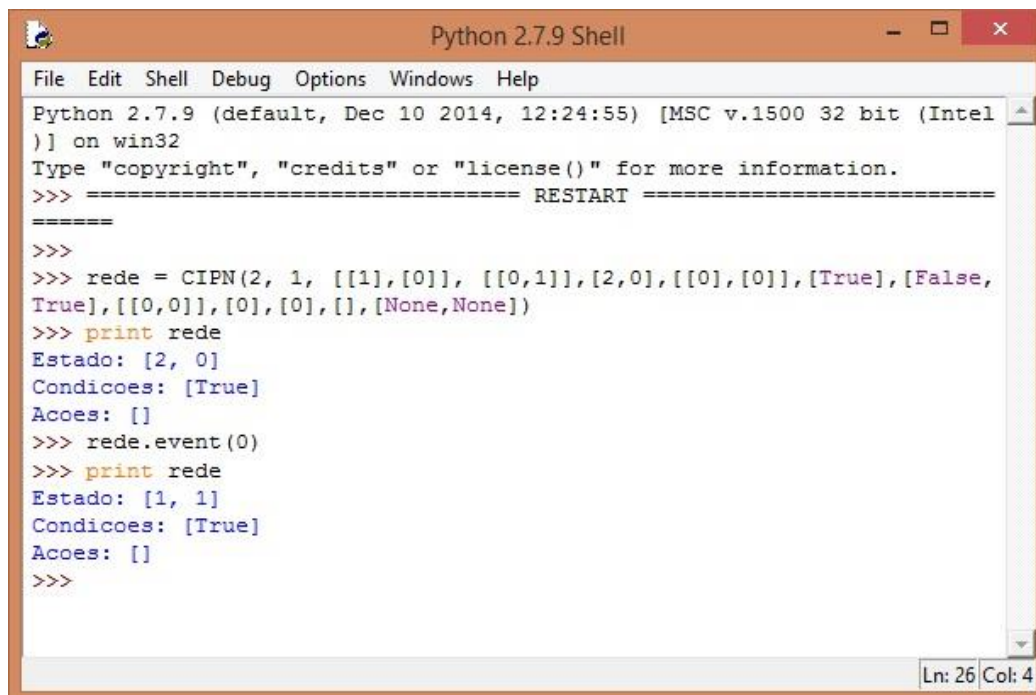


Figura 5-2: Criação de RPIC em uma versão antiga do PETRILab

```
def convert(pn, labels=None):
    # Modulo de eventos externos
    m1 = Ladder()
    l = Linha()
    for i in range(len(pn.Ec)-1):
        if True in map(lambda x: x[1]==i, pn.lc):
            if labels:
                l += Contato(labels[2][i], 1)
            else:
                l += Contato('S'+str(i), 1)
        if pn.eimpulse[i] == 1:
            l += Borda(1)
            if labels:
                l += Bobina(labels[2][i]+'r', 1)
            else:
                l += Bobina('S'+str(i)+'r', 1)
        else:
            l += Borda(0)
```

Figura 5-3: Trecho de código de conversão de RPIC em diagramas Ladder

```

>>> ladder = convert(rede)
>>> for i in ladder:
    print i

|--[S0]----[P]----(S0r)--|
|--[S0r]----<COMP X0>=1>----(B0)--|
|--[B0]----(<SUB X0-=1>||<ADD X1+=1>)--|
|--[*B*]----(<MOVE X0=2>||(S-B))--|

>>>

```

Figura 5-4: Ladder gerado em forma textual

Em meados de 2014 o programa começou a ganhar uma interface gráfica, que foi finalizada no final do ano. Ele será apresentado em detalhes nas seções seguintes.

5.2 Visão Geral

O PETRILab é um programa de modelagem e simulação de Redes de Petri Interpretadas para Controle, que conta com o recurso de conversão automática em diagramas Ladder. Para utilizá-lo, o usuário deve ter a versão 2 do Python instalada em seu computador, que pode ser baixada gratuitamente em <http://www.python.org/>. Atenção: o programa não é compatível com a versão 3 do Python, pois diversas rotinas são alteradas em relação à versão 2. O site ainda desenvolve as duas versões em separado, basta baixar o lançamento mais recente da segunda versão.

Algumas de suas principais características são:

- Interface gráfica simples e intuitiva
- Simulação completa da rede, com o controle da execução de eventos e mudança de condições
- Suporte a transições temporizadas
- Indicadores de execução de ações impulsioneis
- Proteção contra travamentos em loopings infinitos
- Conversão instantânea de RPICs em diagramas Ladder
- Possibilidade de salvar imagens da RPIC e do diagrama
- Teclas de atalho para facilitar a inserção e edição de RPICs

Com essas características, o programa se mostra uma plataforma completa para a modelagem de RPICs. A interface gráfica, combinada com teclas de atalho, permite que

as redes sejam desenhadas muito rapidamente. A rede criada através da linha de comando na Figura 5-2 pode ser desenhada facilmente com alguns cliques, resultando na rede mostrada na Figura 5-5.

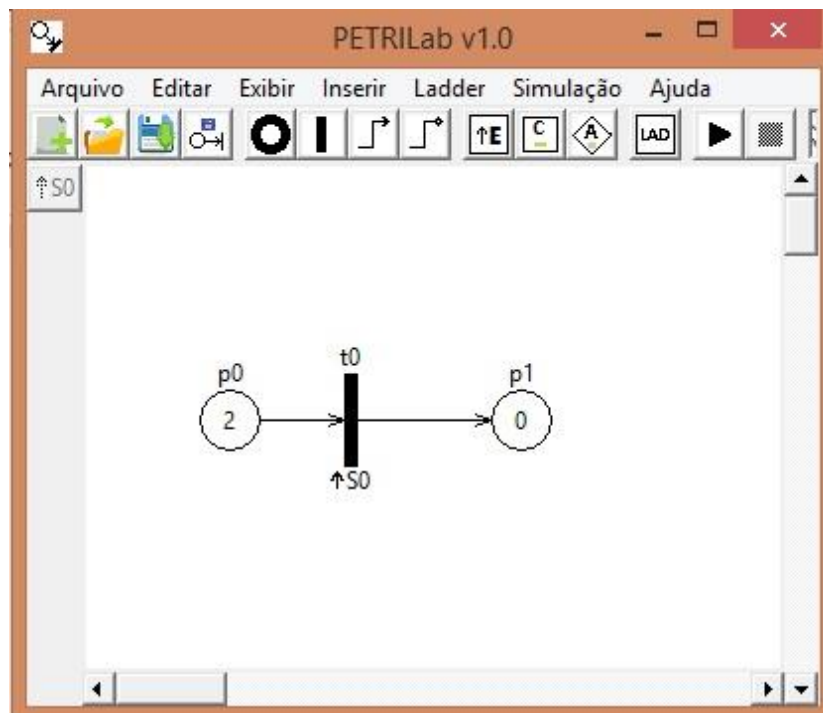


Figura 5-5: Rede de Petri da Figura 5-2 desenhada

Clicando no botão *LAD*, mostrado na barra de ferramentas da Figura 5-5, ou apertando a tecla de atalho <G>, uma nova janela com o diagrama é imediatamente exibida, como mostra a Figura 5-6. Podemos ver que o diagrama corresponde ao gerado textualmente na Figura 5-4, ambos de acordo com o método de conversão proposto.

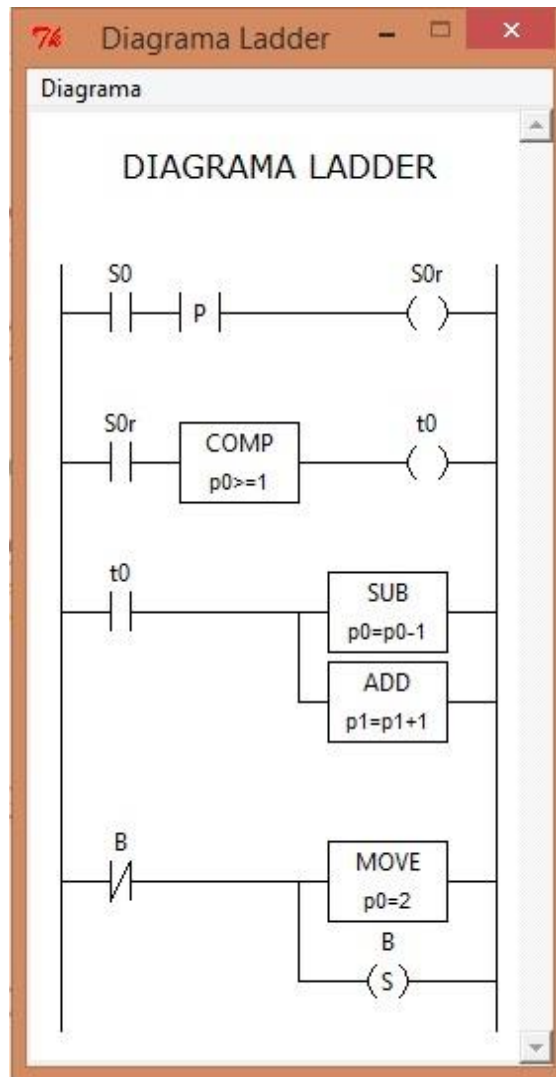


Figura 5-6: Diagrama Ladder gerado a partir da RPIC da Figura 5-5

5.3 Guia do Usuário

5.3.1 Download e instalação

O projeto do PETRILab está hospedado no site do Sourceforge [5], sob o endereço <https://sourceforge.net/projects/petrilab/>. Na página inicial do projeto existe um botão de Download, que aponta para a versão mais recente do programa, como mostra a Figura 5-7. Clique no botão para baixá-la.



Figura 5-7: Página do PETRILab no Sourceforge

O arquivo está compactado no formato *.zip*, e é necessário descompactá-lo. Se o sistema não tiver um descompactador nativo, é necessário baixar um, como o WinRAR [6]. A extração do arquivo é ilustrada na Figura 5-8.

Após a extração, será criada uma pasta contendo os arquivos do programa. Para executá-lo, basta abrir o arquivo *petrilab.pyw*, como mostrado na Figura 5-9. Pode ser conveniente criar um atalho para esse arquivo em sua pasta de preferência.



Figura 5-8: Extração do PETRILab

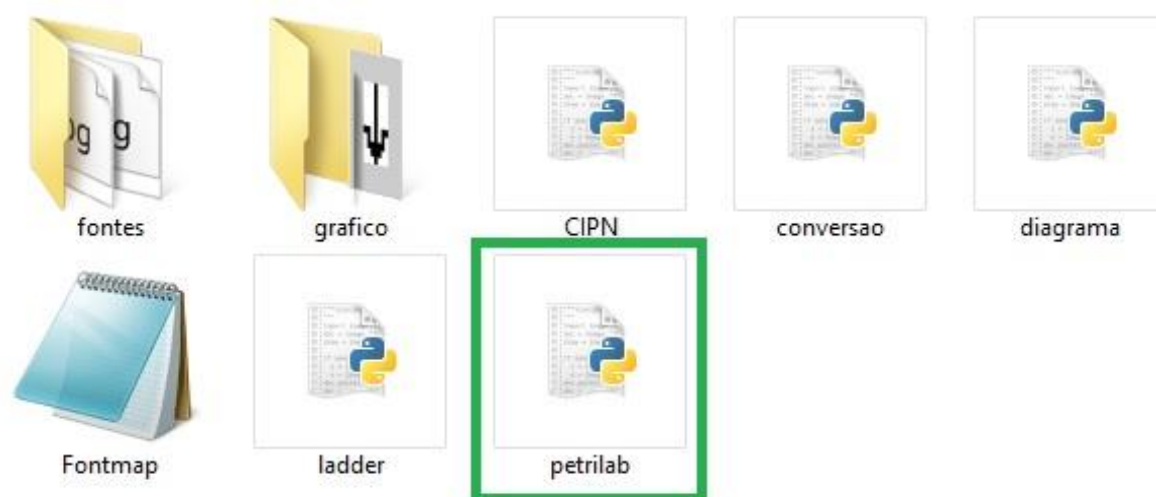


Figura 5-9: Execução do PETRILab

5.3.2 Interface

O PETRILab conta com uma interface gráfica simples e intuitiva, como exibido na Figura 5-10.

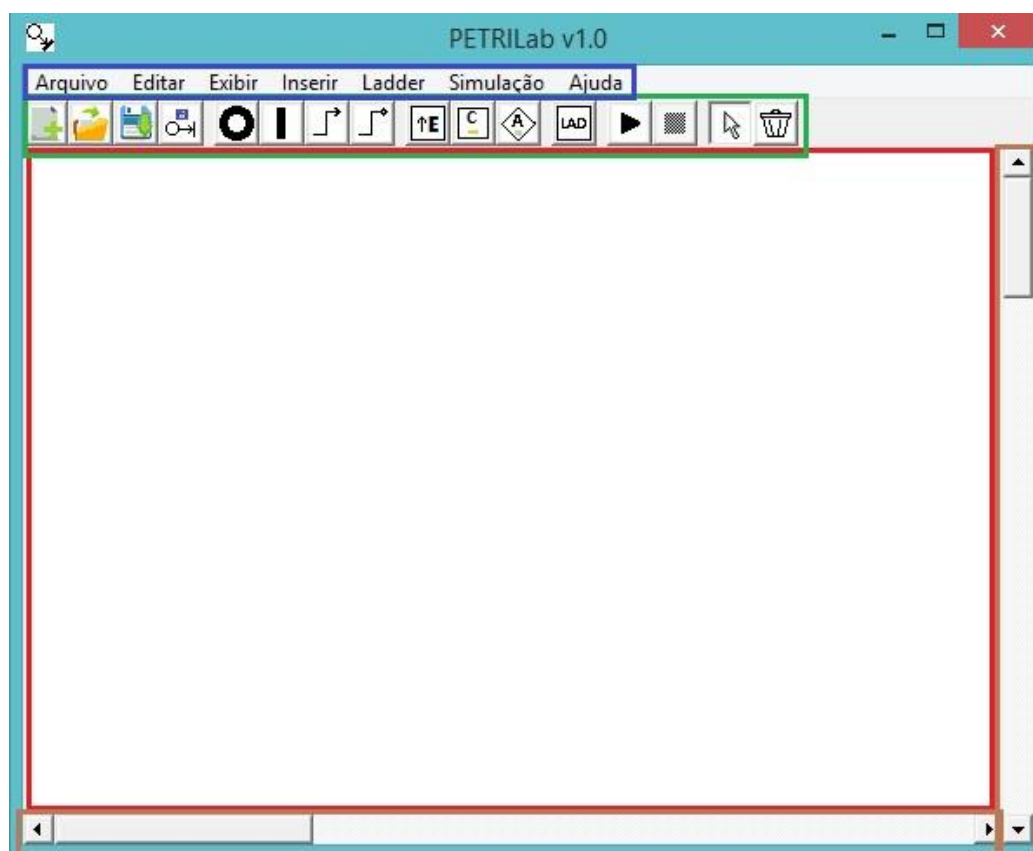


Figura 5-10: Interface do programa

Na Figura 5-10, a área marcada em vermelho é a área de desenho, onde serão inseridos os lugares, transições e arcos da Rede de Petri. Essa área pode ser estendida através das barras de rolagem marcadas em marrom. A barra de ferramentas, marcada em verde, contém botões para acesso rápido de funções, como gerenciamento do arquivo, inserção de elementos, geração de Ladder, simulação e ferramentas de edição. Por fim, a área em azul mostra os menus do programa, onde é possível acessar todas as funções presentes nele.

5.3.3 Inserir elementos

a) Lugar

Para inserir um lugar na rede, clique no botão marcado na Figura 5-11. Alternativamente, vá em *Inserir*→*Lugar (L)*, ou use a tecla de atalho <L>. Em seguida, clique em um ponto da área de desenho. Em qualquer momento o usuário pode apertar a tecla <Esc> para cancelar a inserção.

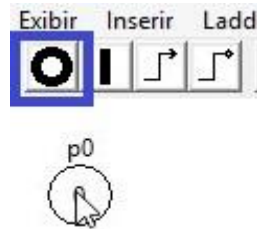


Figura 5-11: Inserção de lugar

b) Transição

Para inserir uma transição na rede, clique no botão marcado na Figura 5-12. Alternativamente, vá em *Inserir*→*Transição (T)*, ou use a tecla de atalho <T>. Em seguida, clique em um ponto da área de desenho. Em qualquer momento o usuário pode apertar a tecla <Esc> para cancelar a inserção.

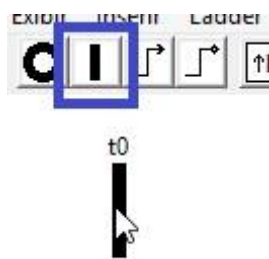


Figura 5-12: Inserção de transição

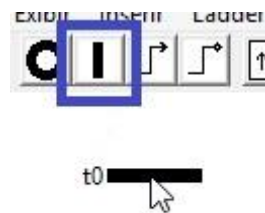


Figura 5-13: Inserção de transição rotacionada

Uma transição pode ser rotacionada no ato de inserção, bastando para isso que o usuário aperte <Clique-Direito> em seu mouse. A Figura 5-13 ilustra uma transição rotacionada sendo inserida.

c) Arco

Um arco pode ser inserido de um lugar a uma transição, ou de uma transição a um lugar. Para inserir um arco, clique no botão marcado na Figura 5-14. Alternativamente, vá em *Inserir*→*Arco*, ou use a tecla de atalho <A>. O cursor se transformará na cruz destacada na Figura 5-14. Ao clicar em um lugar ou transição, o cursor se transformará em uma cruz menor, mostrado na Figura 5-15, indicando que a origem do arco já foi definida, restando informar seu destino. Em qualquer momento o usuário pode apertar a tecla <Esc> para cancelar a inserção.

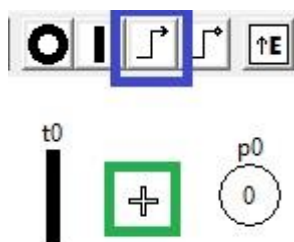


Figura 5-14: Inserção de arco



Figura 5-15: Origem do arco definida

Para definir o destino do arco, basta clicar no lugar ou transição de destino desejado, como mostrado na Figura 5-16. O programa reajusta automaticamente a posição exata do arco para melhor exibição.

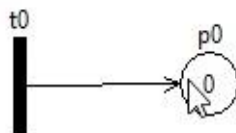


Figura 5-16: Arco inserido

O usuário pode também adicionar diversos segmentos ao arco antes de definir seu destino. Para isso, basta clicar em qualquer lugar da área de desenho, como ilustrado na Figura 5-17. O cursor se manterá como uma cruz pequena. Para finalizar o arco, basta clicar em um lugar ou transição, como mostrado na Figura 5-18.

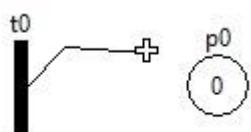


Figura 5-17: Arco segmentado sendo inserido

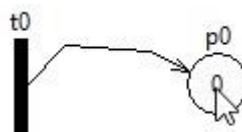


Figura 5-18: Arco segmentado finalizado

d) Arco inibidor

Um arco inibidor pode ser inserido apenas de um lugar a uma transição. Para inseri-lo, clique no botão marcado na Figura 5-19. Alternativamente, vá em *Inserir*→*Arco Inibidor*, ou use a tecla de atalho <I>. O cursor se transformará na cruz destacada na Figura 5-19. Ao clicar em um lugar, o cursor se transformará em uma cruz menor, mostrado na Figura 5-20, indicando que a origem do arco já foi definida, restando informar seu destino. Em qualquer momento o usuário pode apertar a tecla <Esc> para cancelar a inserção.

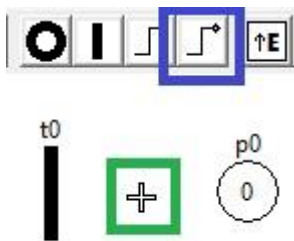


Figura 5-19: Inserção de arco inibidor



Figura 5-20: Origem do arco inibidor definida

Para definir o destino do arco, basta clicar na transição de destino desejada, como mostrado na Figura 5-21. O programa reajusta automaticamente a posição exata do arco para melhor exibição.

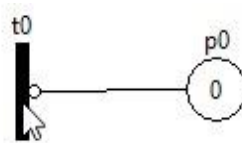


Figura 5-21: Arco inibidor inserido

O usuário pode também adicionar diversos segmentos ao arco antes de definir seu destino. Para isso, basta clicar em qualquer lugar da área de desenho, como ilustrado na Figura 5-22. O cursor se manterá como uma cruz pequena. Para finalizar o arco, basta clicar em uma transição, como mostrado na Figura 5-23.

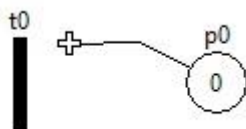


Figura 5-22: Arco inibidor segmentado

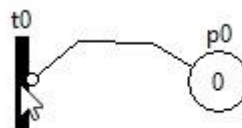


Figura 5-23: Arco inibidor segmentado finalizado

e) Evento

Para adicionar um evento, basta clicar no botão marcado na Figura 5-24, ou ir em *Inserir*→*Evento*, ou ainda apertar a tecla de atalho <E>. Um botão de evento será inserido automaticamente no canto esquerdo do programa, como mostra a Figura 5-24. Para associar eventos às transições, veja a seção 5.3.4b.

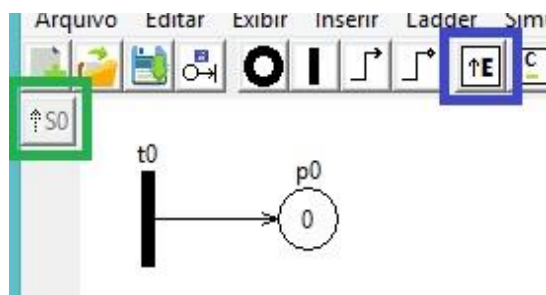


Figura 5-24: Inserção de evento

f) Condição

Para adicionar uma condição, basta clicar no botão marcado na Figura 5-25, ou ir em *Inserir*→*Condição*, ou ainda apertar a tecla de atalho <C>. Um botão de condição será inserido automaticamente na parte superior da área de desenho, como mostra a Figura 5-25. Para associar condições às transições, veja a seção 5.3.4b.

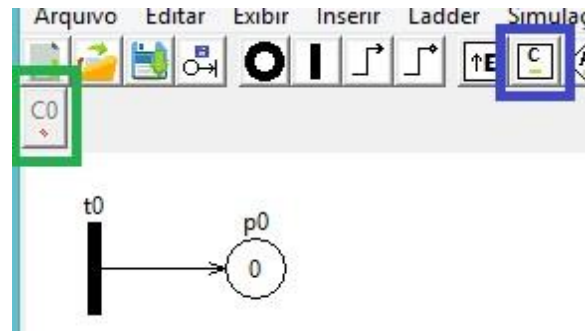


Figura 5-25: Inserção de condição

g) Ação impulsional

Para adicionar uma ação, basta clicar no botão marcado na Figura 5-26, ou ir em *Inserir*→*Ação*, ou ainda apertar a tecla de atalho <K>. Um indicador de ação impulsional será inserido automaticamente no canto direito do programa, como mostra a Figura 5-26. Para associar ações impulsionais aos lugares, veja a seção 5.3.4a.

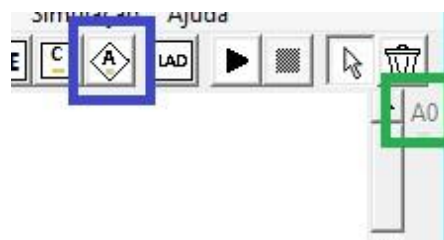


Figura 5-26: Inserção de ação impulsional

5.3.4 Editar elementos

a) Lugares

Para editar um lugar, basta apertar <Duplo-Clique-Esquerdo> com o mouse em cima do lugar desejado. A interface mostrada na Figura 5-27 irá aparecer; nela é possível alterar o rótulo do lugar, alterar sua marcação inicial, ou associar a ele ações impulsionais já

criadas. O lugar será então alterado conforme as modificações realizadas, como mostra a Figura 5-28. Caso seja de interesse a geração do diagrama Ladder da rede, recomenda-se que o rótulo dos lugares tenha, no máximo, três dígitos, para que ele não extravase o desenho dos blocos de funções.

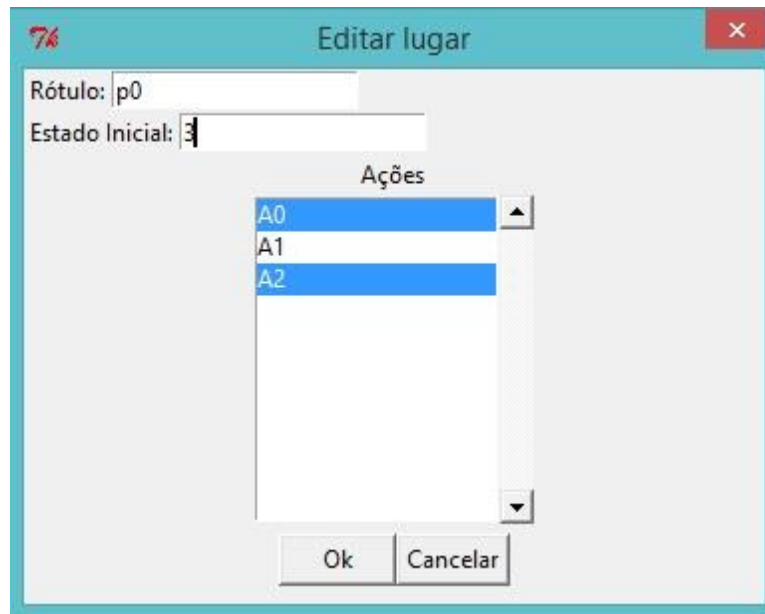


Figura 5-27: Edição de lugar



Figura 5-28: Lugar editado

b) Transições

Para editar uma transição, basta apertar <Duplo-Clique-Esquerdo> com o mouse em cima da transição desejada. A interface mostrada na Figura 5-29 irá aparecer; nela é possível alterar o rótulo da transição, o evento associado, a condição associada ou o atraso associado. A transição será, então, alterada conforme as modificações realizadas, como

mostra a Figura 5-30. Caso seja de interesse a geração do diagrama Ladder da rede, recomenda-se que o rótulo das transições tenha, no máximo, três dígitos.

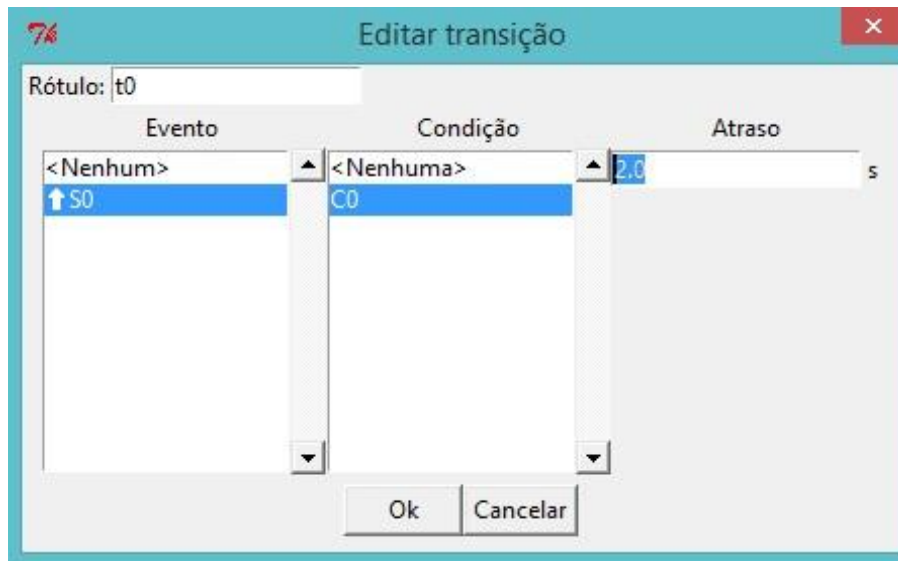


Figura 5-29: Edição de transição



Figura 5-30: Transição editada

Uma transição já inserida também pode ser rotacionada, basta que o usuário aperte <Clique-Direito> com o ponteiro do mouse em cima da transição, como ilustrado na Figura 5-31.



Figura 5-31: Rotação de transição

c) Arcos e arcos inibidores

Para editar qualquer tipo de arco, basta apertar <Clique-Duplo> com o ponteiro do mouse em cima do arco correspondente. É possível, então, editar seu peso, como mostra a Figura 5-32.



Figura 5-32: Edição de arco

O peso do arco será então mostrado no meio do arco correspondente. Um algoritmo garante que o peso esteja sempre no meio do arco, mesmo que ele seja segmentado, como ilustrado na Figura 5-33.

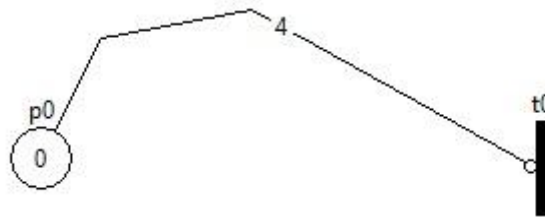


Figura 5-33: Arco editado

d) Eventos, condições e ações

Para editar eventos, ações e condições, basta apertar <Clique-Direito> com o mouse em cima do botão correspondente. É possível então editar seus rótulos, e, para o caso de eventos, o seu tipo de borda, como ilustrado da Figura 5-34 a Figura 5-36.

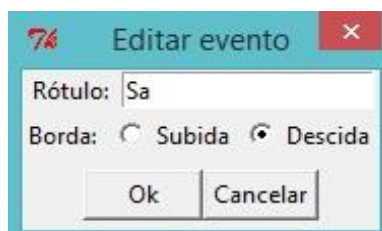


Figura 5-34: Edição de evento

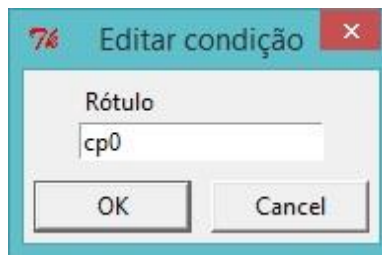


Figura 5-35: Edição de condição

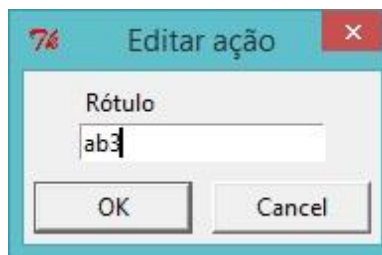


Figura 5-36: Edição de ação

5.3.5 Mover elementos

Para mover um lugar ou uma transição, basta segurar <Clique-Esquerdo> no mouse com o ponteiro em cima deles, e então mover para onde desejar. Quando o elemento estiver no local desejado, basta soltar o botão. Quando movemos um lugar ou transição, os arcos se ajustam automaticamente; no entanto, arcos segmentados só movimentam seu primeiro e último segmentos. Esse comportamento é ilustrado nas Figura 5-37 (a) e (b).

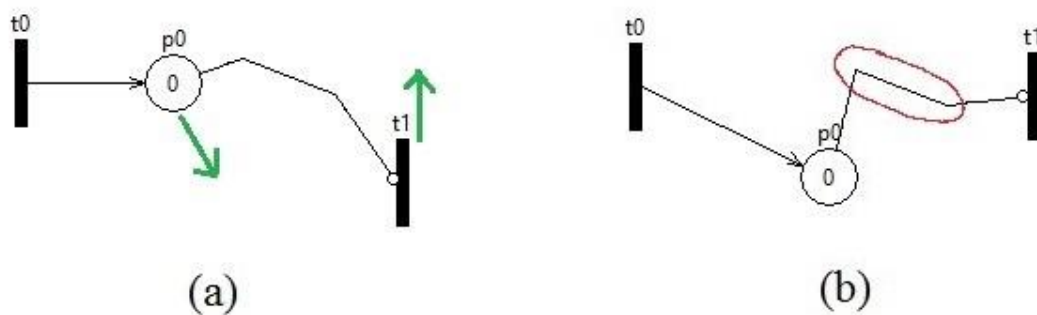


Figura 5-37: Movendo lugares e transições

5.3.6 Remover elementos

a) Lugares, transições, arcos e arcos inibidores

Para remover elementos presentes na área de desenho, é necessário entrar no modo de remoção, clicando no botão mostrado na Figura 5-38 (a) ou apertando a tecla de atalho <X>. O cursor se tornará uma mão, indicando o modo de remoção. Para deletar um elemento, basta então apertar <Click-Esquerdo> com o ponteiro do mouse em cima dele; todos os arcos de saída ou chegada do elemento removido também serão removidos, com ilustram a Figura 5-38 (b).

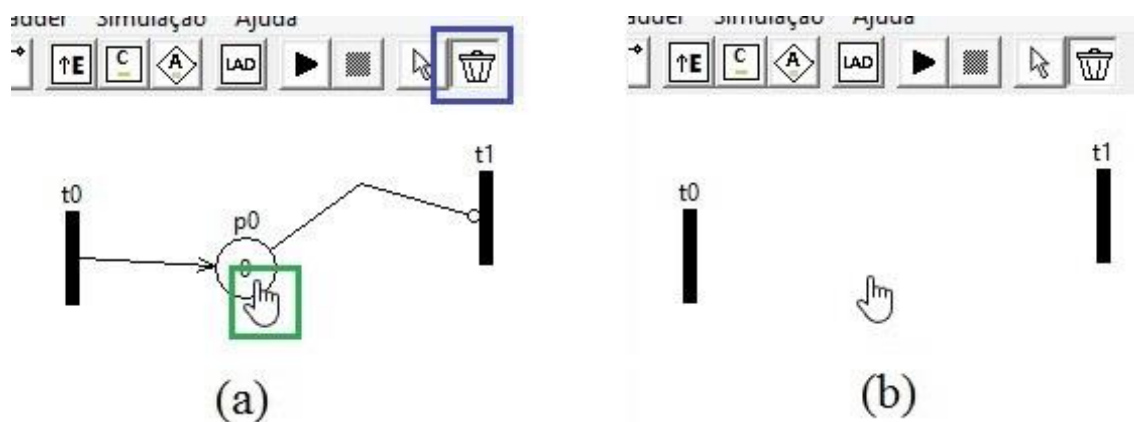


Figura 5-38: Remoção de elementos da área de desenho

Para voltar ao modo de seleção, basta clicar no ícone de seleção marcado na Figura 5-39; o cursor então voltará a ser uma seta. Alternativamente, é possível apertar a tecla de atalho <Q>, ou a tecla <Esc>.

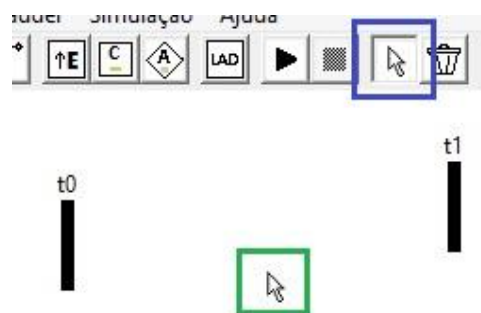


Figura 5-39: Retornando ao modo de seleção

b) Eventos, condições e ações

Para remover eventos, condições ou ações, basta apertar <Clique-Direito> sobre o botão correspondente; ele desaparecerá, como ilustrado nas Figura 5-40 (a) e (b). Qualquer lugar ou transição associado com o item deletado será automaticamente desassociado.

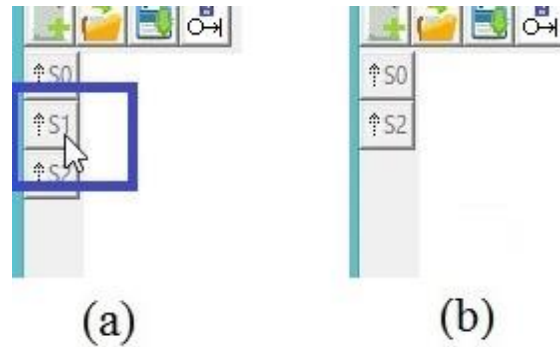


Figura 5-40 Remoção de eventos, condições e ações

5.3.7 Exibir ou ocultar rótulos

Pode de interesse do usuário ocultar os rótulos de lugares e transições, deixando apenas a mostra os eventos, condições e ações associados. Para fazer isso, basta ir em *Exibir*→*Ocultar Rótulos*, como mostra a Figura 5-41. Os rótulos serão, então, ocultados, como pode ser visto na Figura 5-42; para exibi-los novamente, basta desmarcar a opção.

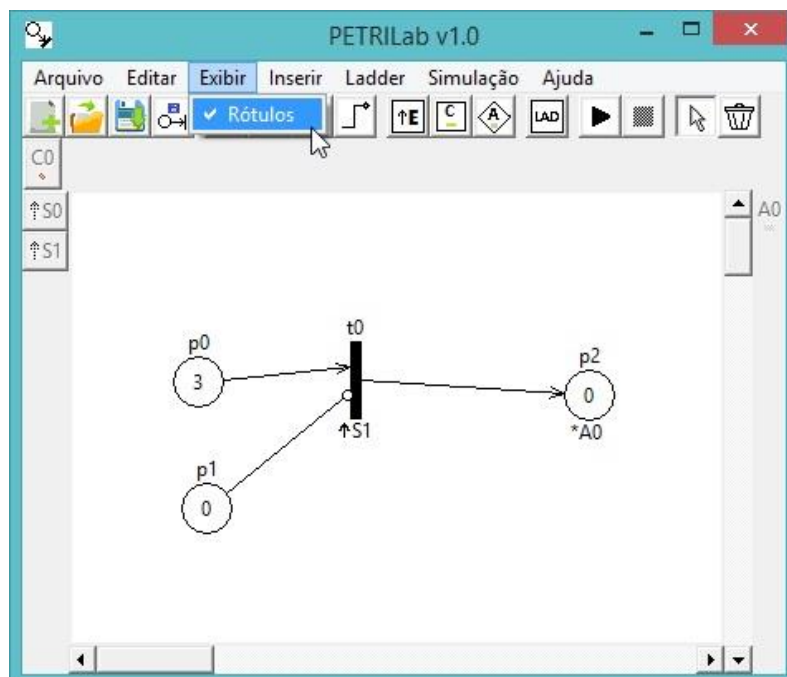


Figura 5-41: Ocultando rótulos

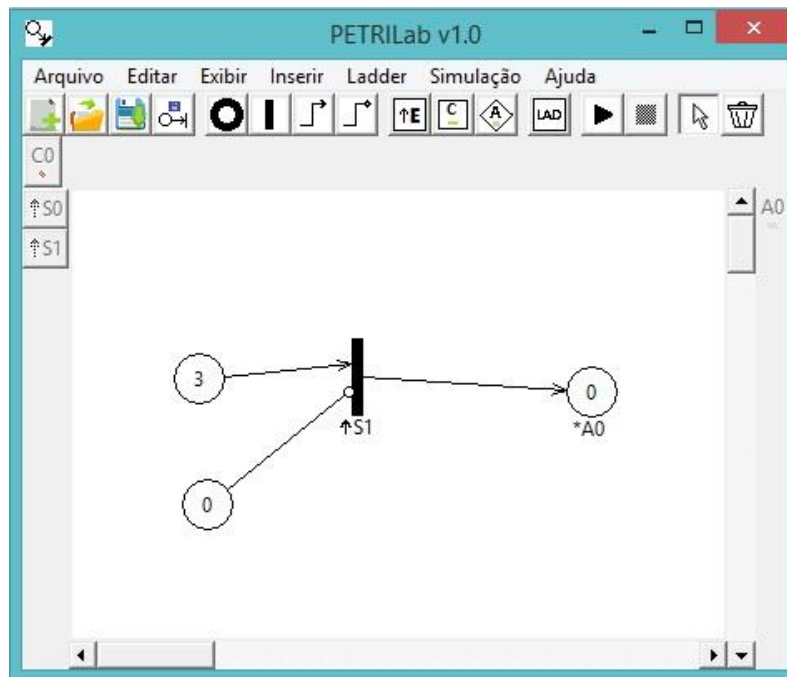
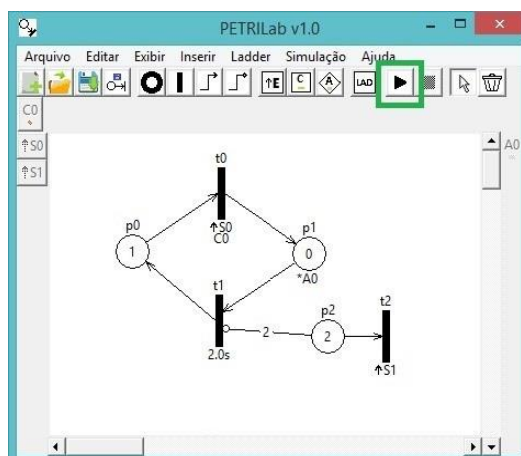


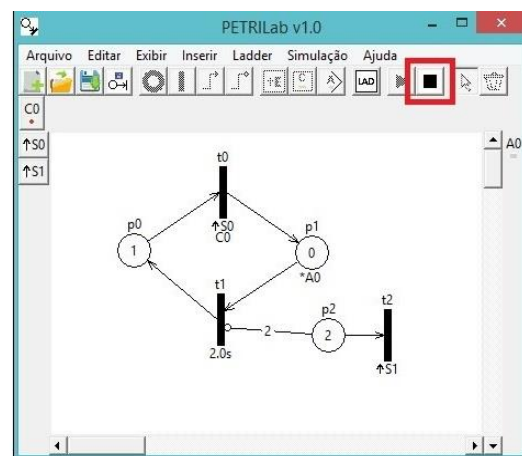
Figura 5-42: Rótulos ocultos

5.3.8 Simular a Rede de Petri

Uma vez modelada, a RPIC pode ser simulada de acordo com a dinâmica explicada na seção 2.4. Para entrar no modo de simulação, basta clicar no botão mostrado na Figura 5-43 (a), ou apertar a tecla de atalho <S>. Como pode ser visto na Figura 5-43 (b), todos os botões de edição se tornam desabilitados, enquanto os botões de eventos, condições, e os indicadores de ações se tornam habilitados. Para voltar ao modo de edição, basta clicar no botão destacado na Figura 5-43 (b) ou apertar a tecla de atalho <Z>.



(a)



(b)

Figura 5-43: Modo de simulação

O modo de simulação só poderá ser iniciado caso haja ao menos um lugar, uma transição e um arco na rede. Além disso, todas as transições da rede precisam de algum limitador para a sua ocorrência, seja ele um arco de entrada, um evento ou uma condição; isso é feito para evitar disparos infinitos de uma transição. O programa conta também com um sistema de prevenção de *loopings infinitos*: caso sejam executadas mais de 2000 iterações na rede sem a intervenção do usuário, a janela mostrada na Figura 5-44 será mostrada, e o usuário pode optar por interromper a simulação.



Figura 5-44: Proteção contra loopings infinitos

Para acionar um evento, basta clicar em seu botão correspondente na barra à esquerda da área de desenho. Para alternar o estado de uma condição, basta clicar em seu botão correspondente na barra superior à área de desenho; uma luz no botão indica se o estado lógico atual é 1 ou 0 – verde ou vermelho – como ilustrado nas Figura 5-45 (a) e (b). A rede responderá de acordo, evoluindo conforme explicado na seção 2.4.

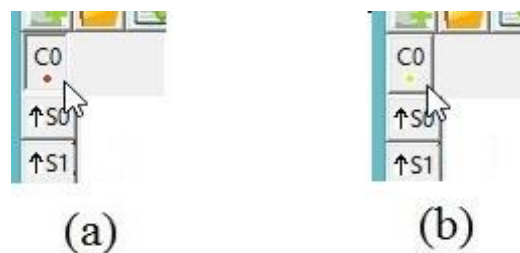


Figura 5-45: Alternando entre estado lógico de condições

Quando um lugar associado a uma ação impulsional muda sua marcação de 0 para 1, o indicador da ação associada pisca com uma luz verde durante 1 segundo, indicando que a ação foi executada, como mostram as Figura 5-46 (a) e (b).

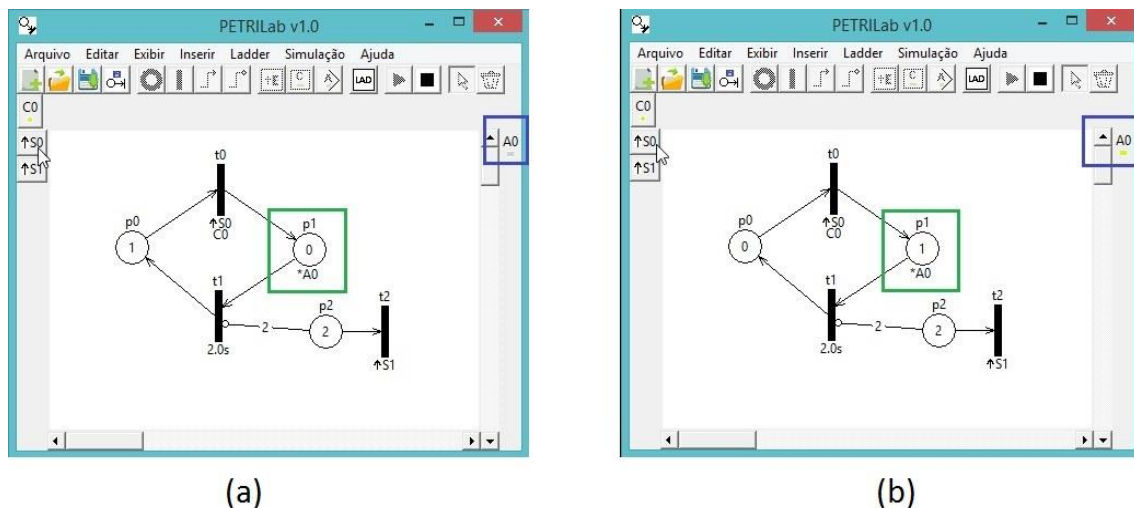


Figura 5-46: Execução de ações impulsivas

5.3.9 Conversão RPIC-LADDER

Uma das funções inovadoras do PETRILab é a conversão automática de uma Rede de Petri Interpretada para Controle em um diagrama Ladder. Essa conversão pode ser feita instantaneamente clicando-se no botão mostrado na Figura 5-47, ou apertando-se a tecla de atalho <G>.

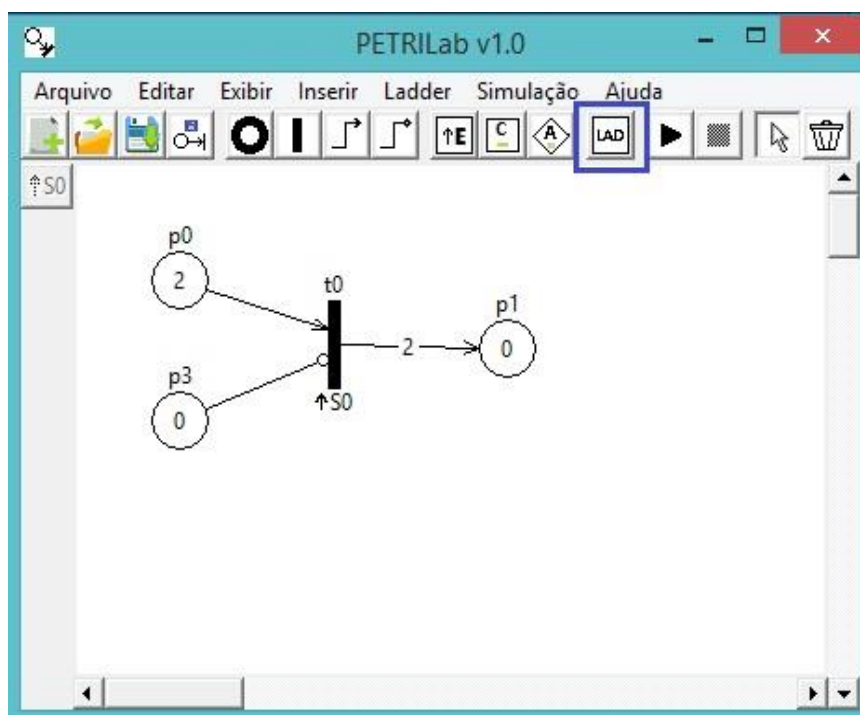


Figura 5-47: Geração de diagrama Ladder

O programa então abrirá uma janela com o diagrama gerado, de acordo com os passos propostos na Seção Capítulo 4 deste trabalho. A Figura 5-48 mostra o diagrama gerado para a rede da Figura 5-47.

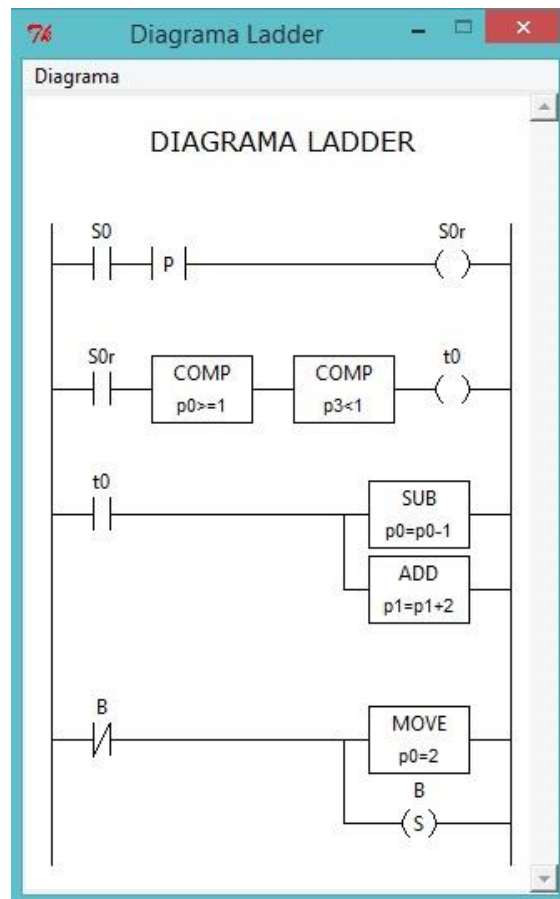


Figura 5-48: Diagrama Ladder gerado a partir da RPIC da Figura 5-47

5.3.10 Salvar imagem da RPIC e do LADDER

a) Grafo da RPIC

O grafo da RPIC modelada pode ser salvo em formato *encapsulated postscript (eps)* clicando-se no botão mostrado na Figura 5-49, ou no menu *Arquivo* → *Salvar Imagem do Grafo*. Caso o arquivo da rede já tenha sido salvo, um arquivo chamado *RPIC.eps* será gerado na mesma pasta dele; caso contrário, será gerado na pasta do PETRILab.

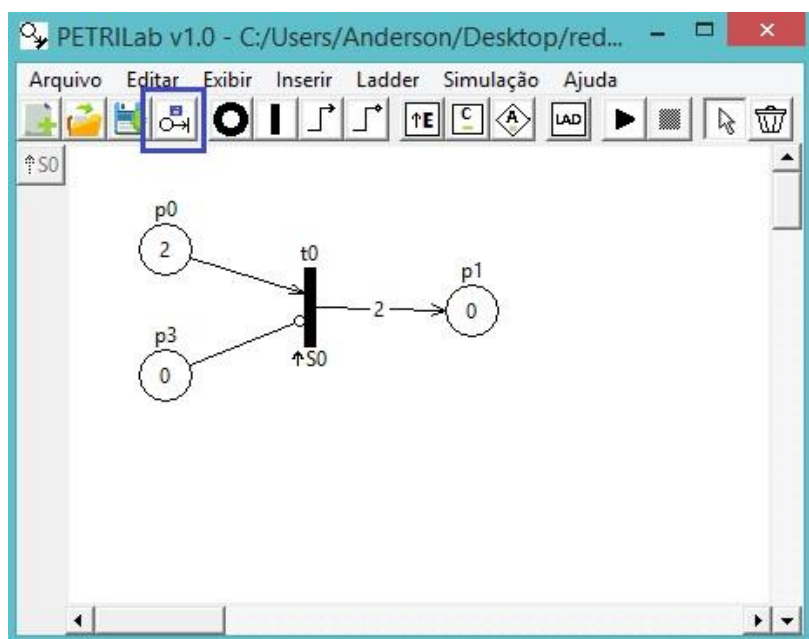


Figura 5-49: Salvando imagem do grafo

O arquivo *eps* pode então ser aberto em algum programa compatível, como o *Adobe Illustrator*, conforme mostra a Figura 5-50.

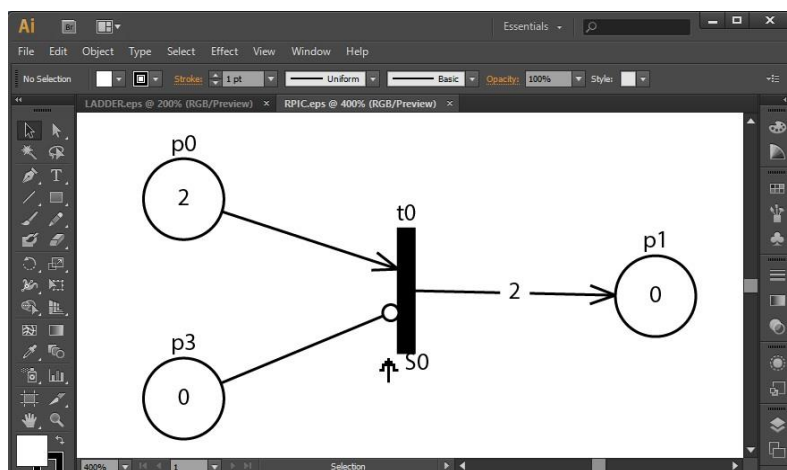


Figura 5-50: Grafo aberta no Adobe Illustrator

b) Diagrama Ladder

Para salvar uma imagem do diagrama Ladder em formato *eps*, basta ir ao menu *Diagrama*→*Salvar*, como mostra a Figura 5-51.

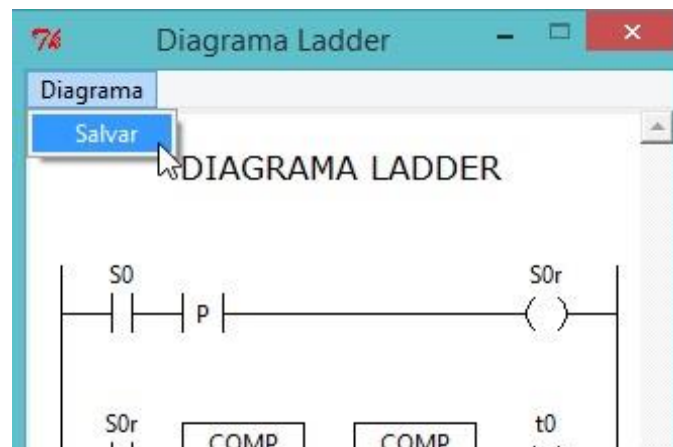


Figura 5-51: Salvando diagrama Ladder

A Figura 5-52 mostra o arquivo *eps* correspondente ao diagrama Ladder da figura 73.

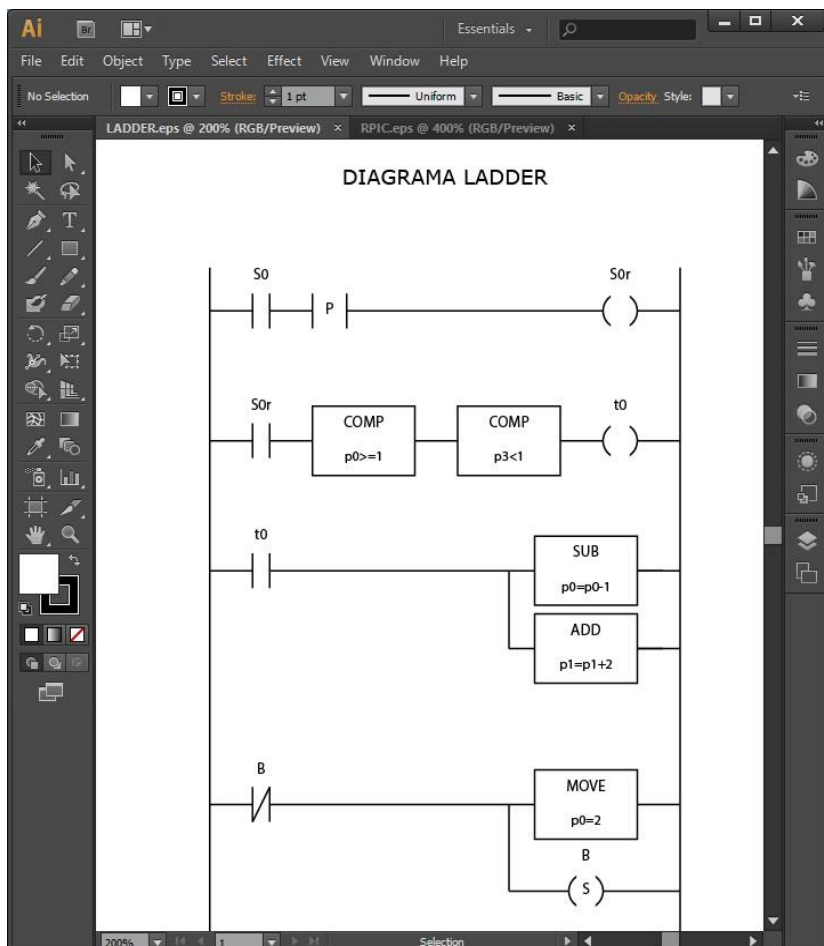


Figura 5-52: Diagrama Ladder aberto no Adobe Illustrator

5.3.11 Criar um Novo Arquivo, Abrir um Arquivo Salvo e Salvar Arquivo Editado

O PETRILab salva os arquivos de projeto com a extensão *.pl*; os arquivos são extremamente leves, dificilmente superando 20kB. As funções *Novo*, *Abrir* e *Salvar* podem ser executadas clicando-se nos botões mostrados na Figura 5-53 – que estão nessa mesma ordem – ou através do menu *Arquivo*.

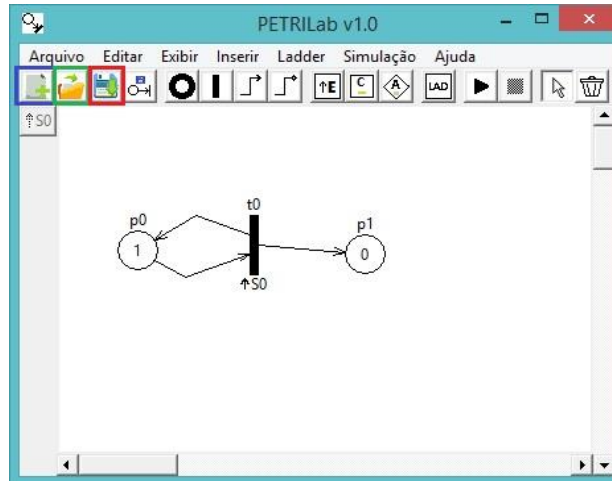


Figura 5-53: Novo, Abrir e Salvar

5.3.12 Ajuda e suporte

O usuário pode ter acesso a este guia através do menu *Ajuda*→*Guia do Usuário*, que abre no navegador um link para download do documento em *pdf* diretamente da página oficial do PETRILab no Sourceforge.

Para reportar bugs ou dar sugestões de melhorias, o usuário pode entrar no menu *Ajuda*→*Informar Bugs/Sugestões*, que abrirá no navegador a página de *Tickets* do PETRILab no Sourceforge. Alternativamente, os bugs e sugestões podem ser enviados por e-mail para *petrilab10@gmail.com*; no caso de bugs, é recomendado que o usuário anexe o arquivo *.pl* do projeto em que o bug ocorreu.

5.4 Lista de Teclas de Atalho

Tabela 5.1: Lista de teclas de atalho

Tecla de Atalho	Função
<CTRL-A>	Abrir arquivo
<CTRL-S>	Salvar arquivo
<CTRL-N>	Novo arquivo
<Esc>	Cancelar ação atual
<A>	Inserir arco
<C>	Inserir condição
<E>	Inserir evento
<I>	Inserir arco inibidor
<G>	Gerar diagrama Ladder
<K>	Inserir ação impulsional
<L>	Inserir lugar
<Q>	Modo de seleção
<S>	Iniciar simulação
<T>	Inserir transição
<X>	Modo de remoção
<Z>	Parar simulação