



Universidade Federal  
do Rio de Janeiro  
Escola Politécnica

## PROJETO DE UM SISTEMA DE AUTOMAÇÃO PARA UM ELEVADOR DE PASSAGEIROS UTILIZANDO REDES DE PETRI

João Feliz de Carvalho Napoli

Projeto de Graduação apresentado ao Curso  
de Engenharia Elétrica da Escola Politécnica,  
Universidade Federal do Rio de Janeiro, como  
parte dos requisitos necessários à obtenção do  
título de Engenheiro.

Orientador: Marcos Vicente de Brito Moreira

Rio de Janeiro  
Março de 2013

PROJETO DE UM SISTEMA DE AUTOMAÇÃO PARA UM ELEVADOR DE  
PASSAGEIROS UTILIZANDO REDES DE PETRI

João Feliz de Carvalho Napoli

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO  
CURSO DE ENGENHARIA ELÉTRICA DA ESCOLA POLITÉCNICA  
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE  
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE  
ENGENHEIRO ELETRICISTA.

Examinado por:

---

Prof. Marcos Vicente de Brito Moreira, D.Sc.

---

Prof. Richard Magdalena Stephan, Dr.-Ing.

---

Prof. Oumar Diene, D.Sc.

RIO DE JANEIRO, RJ – BRASIL  
MARÇO DE 2013

Napoli, João Feliz de Carvalho

Projeto de um Sistema de Automação para um Elevador de Passageiros Utilizando Redes de Petri/João Feliz de Carvalho Napoli. – Rio de Janeiro: UFRJ/ Escola Politécnica, 2013.

XIV, 58 p.: il.; 29, 7cm.

Orientador: Marcos Vicente de Brito Moreira

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia Elétrica, 2013.

Referências Bibliográficas: p. 57 – 58.

1. Elevador. 2. Rede de Petri. 3. Automação. 4. CLP. 5. Ladder. I. Moreira, Marcos Vicente de Brito. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia Elétrica. III. Título.

*Dedico, não apenas este trabalho  
mas todo meu sucesso, nesta  
empreitada que é a graduação  
em Engenharia Elétrica, a toda  
minha família. Em especial, a  
minha avó, Maria Julia de  
Carvalho Pereira Alves Mendes,  
que apesar de não estar conosco,  
sem dúvida está comemorando de  
um lugar melhor. (In Memoriam)*

”A inspiração existe, mas tem  
que te encontrar trabalhando.”

*Picasso, P.*

# Agradecimentos

Agradeço a Deus. O que Deus faz por nós, vai além do que enxerga nossa vã compreensão.

Aos meus pais Umberto Giovanni Napoli e Maria Teresa Pereira Alves Mendes, por tudo; tudo que sou e sei é devido a base que recebi de ambos.

A minha família e a meus amigos, que estiveram ao meu lado em todos os momentos, e que tornaram meus dias mais alegres, fazendo toda a jornada valer a pena.

A meus colegas de trabalho, por todo apoio, compreensão e incentivo.

A todos que não foram citados, mas de alguma forma fizeram parte da minha trajetória, e direta ou indiretamente contribuíram para que eu chegasse onde cheguei.

A todas as dificuldades que enfrentei; pois as dificuldades são o tempero do sucesso.

Por fim, mas não menos importante, ao meu orientador e professor Marcos Vicente de Brito Moreira, pelo sucesso em consolidar e multiplicar o conhecimento.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Eletricista.

Projeto de um Sistema de Automação para um Elevador de Passageiros Utilizando  
Redes de Petri

João Feliz de Carvalho Napoli

Março/2013

Orientador: Marcos Vicente de Brito Moreira

Curso: Engenharia Elétrica

Este trabalho tem como objetivo utilizar uma rede de Petri interpretada para controle na modelagem de sistemas a eventos discretos, aplicada ao controle supervisório de um elevador de passageiros. Além disso, objetiva caracterizar e exemplificar o funcionamento de Controladores Lógicos Programáveis (CLPs), e abordar a programação *Ladder*, expondo um método simples e efetivo de conversão RPIC para *Ladder*.

*Palavras-chave:* Sistemas a eventos discretos, Redes de Petri interpretadas para controle, Ladder, Controladores Lógicos Programáveis.

Abstract of Undergraduate Project presented to POLI/UFRJ as partial fulfillment of requirements for the degree of Engineer.

AUTOMATION SYSTEM DESIGN FOR A PASSENGER ELEVATOR USING  
PETRI NETS

João Feliz de Carvalho Napoli

March/2013

Advisor: Marcos Vicente de Brito Moreira

Course: Electrical Engineering

In this project the architecture of a elevator controller modeled by a Petri Net is presented. The implemented PLC based approach provides particularly simple constructions methods for supervisory controllers based on discrete event systems. In addition, methods for the conversion of the Petri net lift controller into a ladder diagram for implementation on a programmable logic controller (PLC) are presented. The conversion methods lead to PLC programming codes that preserve the structure and represent the evolution of the tokens of the Petri Net.

*Keywords:* Elevator, Petri Net, Automation, PLC, Ladder.

# Sumário

<b>Lista de Figuras</b>	xii
<b>Lista de Tabelas</b>	xiv
<b>1 Introdução</b>	1
<b>2 Fundamentos Básicos de Redes de Petri</b>	3
2.1 Rede de Petri . . . . .	3
2.2 Marcação da Rede de Petri . . . . .	4
2.3 Transição Temporizada . . . . .	5
2.4 Dinâmica da Rede de Petri . . . . .	5
2.5 Redes de Petri com Arco Inibidor . . . . .	7
2.6 Redes de Petri Interpretadas para Controle . . . . .	8
2.7 Formas de Obtenção de Redes de Petri . . . . .	10
2.7.1 Composição Modular . . . . .	10
2.7.2 Refinamentos Sucessivos . . . . .	11
2.7.3 Híbrida . . . . .	11
<b>3 Controlador Lógico Programável</b>	12
3.1 História e Definição . . . . .	12
3.2 Linguagens de Programação . . . . .	14
<b>4 Diagrama <i>Ladder</i></b>	15
4.1 Componentes de um Diagrama <i>Ladder</i> . . . . .	15
4.1.1 Contatos . . . . .	16

4.1.2	Bobinas . . . . .	19
4.1.3	Temporizadores . . . . .	21
<b>5</b>	<b>Conversão de RPIC para Ladder</b>	<b>22</b>
5.1	Módulo de inicialização . . . . .	23
5.2	Módulo de Eventos Externos . . . . .	23
5.3	Módulo das Condições . . . . .	24
5.4	Módulo da Dinâmica . . . . .	25
5.5	Módulo das Ações . . . . .	26
5.6	Informações Complementares . . . . .	27
<b>6</b>	<b>Elevadores</b>	<b>28</b>
6.1	Aspectos Construtivos Gerais . . . . .	28
6.1.1	Máquina de Tração . . . . .	29
6.1.2	Sistema de Freio . . . . .	30
6.1.3	Portas do vão e da Cabina . . . . .	30
6.1.4	Sistemas de Segurança . . . . .	31
6.2	Sistema de Automação . . . . .	31
6.2.1	Sensores - Dispositivos de Monitoração . . . . .	32
6.2.2	Lógica de Atendimento de Chamadas . . . . .	33
6.2.3	Comando Automático Coletivo . . . . .	34
6.2.4	Comando Automático Coletivo Seletivo na Descida . . . . .	34
6.2.5	Comando Automático Coletivo Seletivo na Subida e na Descida	34
<b>7</b>	<b>Obtenção do Código <i>Ladder</i> a Partir da RPIC do Sistema de Automação de um Elevador de Passageiros</b>	<b>36</b>
7.1	Definições do Sistema . . . . .	36
7.1.1	Informações Complementares dos Componentes . . . . .	37
7.2	Obtenção do Controlador em RPIC . . . . .	38
7.2.1	Análise das Etapas de Atendimento . . . . .	39
7.2.2	Definições e Símbologia Adotada . . . . .	39

7.2.3	Módulos da RPIC . . . . .	40
7.2.4	RPIC Completa do Sistema . . . . .	45
7.3	Conversão de RPIC para Ladder . . . . .	49
<b>8</b>	<b>Conclusões</b>	<b>55</b>
	<b>Referências Bibliográficas</b>	<b>57</b>

# Listas de Figuras

2.1	Grafo da rede de Petri e seus elementos básicos.	4
2.2	Rede de Petri do exemplo 1, marcada por uma ficha no lugar $p_0$ .	5
2.3	Representação do arco habilitador	6
2.4	Rede de Petri do exemplo 1, antes e depois do disparo de $t_0$ , respectivamente.	7
2.5	Grafo de uma rede de Petri utilizando arco inibidor.	8
2.6	Fluxo de dados em uma rede de Petri interpretada para controle.	9
2.7	Representação gráfica de uma rede de Petri interpretada para controle.	10
3.1	Esquema ilustrativo de automação com CLP do exemplo 3.	13
3.2	Esquema ilustrativo de um CLP.	14
4.1	Ilustração do diagrama Ladder básico.	16
4.2	Ilustração do contato NA e sua tabela verdade.	17
4.3	Ilustração do contato NF e sua tabela verdade.	17
4.4	Ilustração do contato detetor de borda de subida e sua tabela verdade.	18
4.5	Ilustração do contato detetor de borda de descida e sua tabela verdade.	19
4.6	Ilustração da bobina simples e sua tabela verdade.	20
4.7	Ilustração da bobina <i>SET</i> com variável $y$ associada.	20
4.8	Ilustração da bobina <i>RESET</i> com variável $y$ associada.	21
4.9	Ilustração do temporizador TON.	21
5.1	RPIC que modela o sistema do exemplo 3.	22
5.2	Módulo de inicialização do modelo RPIC para Ladder.	23

5.3	Módulo de eventos externos do modelo RPIC para Ladder. . . . .	24
5.4	Módulo das condições do modelo RPIC para Ladder. . . . .	25
5.5	Módulo da dinâmica do modelo RPIC para Ladder. . . . .	26
5.6	Módulo das ações do modelo RPIC para Ladder. . . . .	27
6.1	Ilustração do sensor de posição da cabina. . . . .	32
7.1	Da esquerda pra direita: Lugar $P_0$ ; arco ordinário; transição instantânea $T_0$ ; arco habilitador; lugar $P_1$ ; arco inibidor e transição temporizada $T_1$ . . . . .	40
7.2	Módulo da posição da cabina. . . . .	40
7.3	Módulo da lógica de atendimento. . . . .	42
7.4	Módulo da máquina de tração. . . . .	42
7.5	Módulo das botoeiras. . . . .	43
7.6	Módulo da porta automática. . . . .	44
7.7	RPIC do controlador automático do elevador. . . . .	45
7.8	Módulos de inicialização (i) e de eventos externos (ii) do controlador do elevador. . . . .	49
7.9	Módulo das condições do controlador do elevador. . . . .	50
7.10	Segunda parte do módulo das condições do controlador do elevador. .	51
7.11	Módulo da dinâmica do controlador do elevador. . . . .	52
7.12	Módulo da dinâmica do controlador do elevador. . . . .	53
7.13	Módulos da dinâmica (i) e das ações (ii) do controlador do elevador. .	54

# **Lista de Tabelas**

7.1	Estados iniciais, ações e significado dos lugares. . . . .	47
7.2	Receptividades, eventos e condições associadas às transições. . . . .	48

# Capítulo 1

## Introdução

Pode-se dizer que até meados do século XIX a vida era horizontal, já que quase não havia construções com mais de três pavimentos. Com o acelerado crescimento das grandes metrópoles, e o espaço físico cada vez mais escasso, a demanda por uma solução vertical chamou a atenção de Elisha Graves Otis. Mais de 160 anos depois, o que pode ser visto é que o elevador está presente no cotidiano de todos, seja em casa, trabalho, ou nas lojas e supermercados. A vida moderna e vertical como conhecemos hoje não existiria sem este grande mas, por vezes, imperceptível advento da engenharia.

Hoje em dia, com os avanços da tecnologia e computação, várias ferramentas foram desenvolvidas com o intuito de modelar de forma eficiente e precisa os sistemas automatizados de controle de elevadores. O controle automático de um elevador tem como finalidade garantir o deslocamento seguro, rápido e eficiente, seja para passageiros ou para cargas.

Com o passar do tempo, grandes evoluções na área de eletrônica e instrumentação possibilitaram sensores cada vez melhores e confiáveis, e com isso, grandes projetos de automação puderam ser desenvolvidos. Com isso, as técnicas de projeto também precisaram sofrer modificações para que pudessem alcançar o grau de sofisticação e complexidade demandado. Dessa necessidade de padronização e consolidação das técnicas de projeto que surgiu a motivação para este trabalho. Alguns trabalhos anteriores foram feitos a respeito do assunto abordado, o que demonstra o interesse da

comunidade acadêmica a cerca desta temática. Dentre eles podemos citar CABRAL [1], BOTELHO [2] e MONTEZANO [3].

Este trabalho tem como objetivo aplicar os conceitos de redes de Petri [4, 5] na modelagem de um sistema de automação de elevador residencial, caracterizar e exemplificar o funcionamento de Controladores Lógicos Programáveis (CLPs), e abordar a programação Ladder, expondo o método MOREIRA *et al.* [6] de conversão de redes de Petri interpretadas para controle (RPIC) para Ladder como meio de obtenção de sistemas de automação baseados em CLPs.

No capítulo 2 serão abordados os fundamentos básicos das redes de Petri bem como métodos de composição e estruturação. A partir do capítulo 3 será apresentado um resumo a respeito de CLPs, contendo seus aspectos construtivos e de funcionamento, e então, no capítulo 4 será apresentada a linguagem *Ladder* de programação, que é a principal linguagem utilizada por CLPs. O método de conversão de RPIC para *Ladder* será exposto no capítulo 5, onde um exemplo ilustrará a aplicação. No capítulo 6 são introduzidos conceitos e terminologias sobre elevadores necessários para a compreensão da aplicação deste trabalho, que será desenvolvida ao longo do capítulo 7. Por fim, no capítulo 8, será feito o fechamento do trabalho e as devidas conclusões serão expostas.

# Capítulo 2

## Fundamentos Básicos de Redes de Petri

Neste capítulo são abordados os principais fundamentos que definem as Redes de Petri, de forma objetiva e aplicada a este projeto. Uma rede de Petri é uma forma de representação matemática para sistemas a eventos discretos. [1–5].

### 2.1 Rede de Petri

Uma rede de Petri pode ser definida em um processo de duas etapas. A definição do grafo da Rede de Petri, também chamado de estrutura da rede de Petri; e a segunda etapa, que consiste em adicionar ao grafo um estado inicial, resultando assim no modelo completo da rede de Petri.

**Definição 1** *Uma rede de Petri  $\mathcal{N}$  é uma quíntupla:*

$$\mathcal{N} = (P, T, Pre, Pos, x_0) \quad (2.1)$$

□

- $P = \{p_0, p_1, \dots, p_n\}$  é o conjunto de lugares;
- $T = \{t_0, t_1, \dots, t_m\}$  é o conjunto de transições;

- $Pre : P \times T \rightarrow \mathbb{N}$  é a função dos arcos de **entrada** das transições;
- $Pos : T \times P \rightarrow \mathbb{N}$  é a função dos arcos de **saída** das transições;
- $x_0 : P \rightarrow \mathbb{N}$  é a função de marcação dos lugares.

Sendo assim, existe um arco ligando um lugar  $p_i$  a uma transição  $t_j$  se somente se  $Pre(p_i, t_j) > 0$ . Analogamente, existe um arco ligando uma transição  $t_j$  a um lugar  $p_i$  se somente se  $Pos(t_j, p_i) > 0$ . O conjunto de lugares de entrada (transições de entrada) de uma transição  $t_j \in T$  (lugar  $p_i \in P$ ) é denotado por  $I(t_j)$  ( $I(p_i)$ ) e é formado por lugares  $p_i \in P$  (transições  $t_j \in T$ ).

Como forma de representação gráfica para as redes de Petri, os lugares são representados por círculos, as transições por barras, e as relações por arcos orientados com peso, como pode ser visto na figura 2.1.

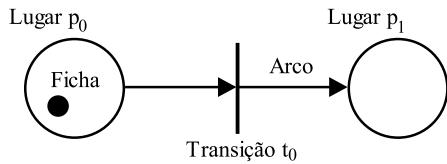


Figura 2.1: Grafo da rede de Petri e seus elementos básicos.

## 2.2 Marcação da Rede de Petri

Para que as transições disparem, é necessário que estejam habilitadas. Uma transição é dita estar habilitada quando o número de fichas em cada um de seus lugares de entrada é maior ou igual ao peso do arco que liga esse lugar à transição. A marcação para as redes de Petri é definida matematicamente utilizando-se o vetor coluna  $\underline{x}_0 = [x_0(p_1), x_0(p_2), \dots, x_0(p_n)]^T$ , formado pelo número inicial de fichas em cada lugar  $p_i$ , para  $i = 1, \dots, n$ , e como resultado da função de marcação obtém-se uma rede de Petri.

Em uma rede de Petri, o vetor de marcação de lugares  $\underline{x}_0$  é o estado inicial do sistema que a rede de Petri representa. A cada evolução do sistema, um novo estado

é alcançado e há uma nova marcação de lugares na rede de Petri. O exemplo 1 ilustra uma rede de Petri marcada.

**Exemplo 1** Seja um grafo de uma rede de Petri definido por:  $P = \{p_0, p_1, p_2\}$ ,  $T = \{t_0, t_1\}$ ,  $Pre(p_0, t_0) = 1$ ,  $Pre(p_0, t_1) = 1$ ,  $Pre(p_2, t_1) = 1$ ,  $Pos(t_0, p_1) = 1$  e  $Pos(t_1, p_1) = 1$ . Suponha que, através da função de marcação inicial, o vetor de marcação de estados inicial seja  $\underline{x}_0 = [1 \ 0 \ 0]^T$ . A rede de Petri com a marcação correspondente pode ser vista na figura 2.2.

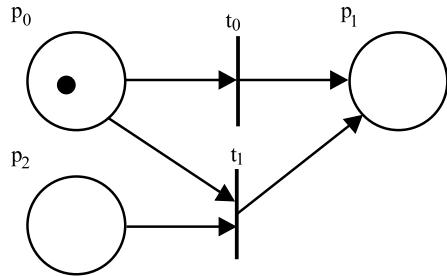


Figura 2.2: Rede de Petri do exemplo 1, marcada por uma ficha no lugar  $p_0$ .

## 2.3 Transição Temporizada

Uma transição  $t_j$  é dita ser temporizada quando possui um atraso de disparo  $d_j$  associado a ela, ou seja, quando a transição se torna habilitada, esta não dispara instantaneamente, disparando apenas após decorrido o tempo programado  $d_j$ , com contagem iniciada imediatamente após a habilitação. Neste caso,  $T$  pode ser partitionado como  $T_0 \cup T_d$ , tal que,  $T_0$  é o conjunto de transições não temporizadas e  $T_d$  é o conjunto de transições temporizadas.

## 2.4 Dinâmica da Rede de Petri

O mecanismo de transição de estado da rede de Petri é providenciado pela marcação das fichas ao longo da rede. Uma transição quando habilitada, pode disparar fazendo o sistema evoluir ao próximo estado, logo, a evolução do sistema ocorre de

acordo com os disparos das transições. Ao disparar, a rede de Petri alcança uma nova marcação  $\underline{x}'$ . A evolução é melhor compreendida observando a definição 2 e o exemplo 2.

**Definição 2** *Evolução da marcação de uma rede de Petri é dada por:*

$$x'(p_i) = x(p_i) - Pre(p_i, t_j) + Post(t_j, p_i), \text{ para } i = 1, \dots, n. \quad (2.2)$$

□

É fácil notar que a definição 2 permite que  $p_i$  seja, simultaneamente, um lugar de entrada e de saída da transição  $t_j$ . A dinâmica deste caso em particular, onde são retiradas  $Pre(p_i, t_j)$  fichas de  $p_i$  e então, imediatamente são colocadas  $Post(t_j, p_i)$  fichas de volta, pode ser representada graficamente através de um arco ordinário com a seta não preenchida, ilustrado na figura 2.3. Este arco recebe o nome de arco habilitador, pois apesar de habilitar a transição associada  $t_j$ , ao disparar, o número de fichas do lugar  $p_i$  que a habilitou não é alterado.

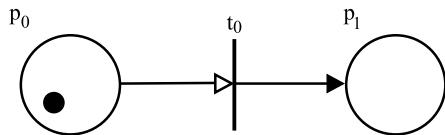


Figura 2.3: Representação do arco habilitador

Para ilustrar o processo de disparo de uma transição, mostrando a evolução da rede de Petri antes e depois do disparo, segue o exemplo 2.

**Exemplo 2** *A partir da rede de Petri do exemplo 1, que pode ser vista graficamente na figura 2.2, é possível notar que a transição  $t_0$  está habilitada e, portanto, pode disparar. Imediatamente após  $t_0$  disparar, como  $Pre(p_0, t_0) = 1$ , o lugar  $p_0$  perde uma ficha e, como  $Post(t_0, p_1) = 1$ , uma ficha é colocada no lugar  $p_1$ , resultando na rede de Petri que é mostrada na figura 2.4.*

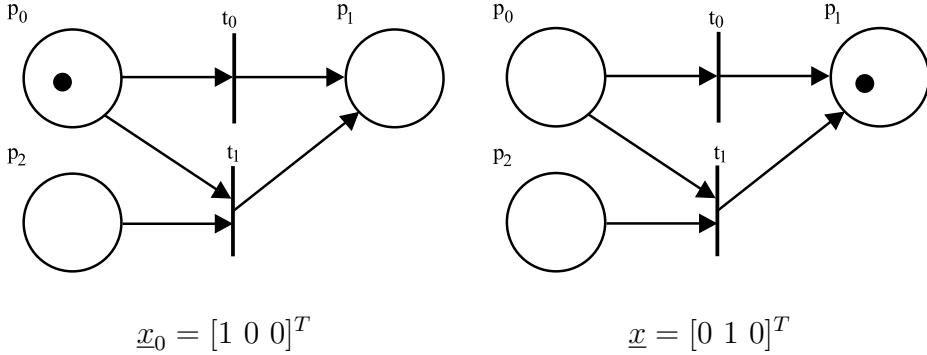


Figura 2.4: Rede de Petri do exemplo 1, antes e depois do disparo de  $t_0$ , respectivamente.

## 2.5 Redes de Petri com Arco Inibidor

Uma rede de Petri com arco inibidor é uma rede de Petri  $\mathcal{N}$  com a adição da função  $In$ , que é uma função de inibição que define os arcos inibidores de um lugar para uma transição. Dessa forma, é possível utilizar arcos inibidores, cuja função é apenas de conectar um lugar  $p_i$  a uma transição  $t_j$ , fazendo parte da pré-condição de  $t_j$ . A transição  $t_j$  é habilitada apenas quando o lugar  $p_i$  não possuir fichas. Sendo,  $In : P \times T \rightarrow \mathbb{N}$  é a função que define os arcos inibidores.

**Definição 3** Uma rede de Petri com arco inibidor  $\mathcal{W}$  é uma sextupla:

$$\mathcal{W} = (P, T, Pre, Pos, x_0, In) \quad (2.3)$$

□

A dinâmica do arco inibidor é diferente da vista na subseção 2.4, já que, após o disparo da transição associada, mantém a marcação do lugar  $p_i$  inalterada. A forma gráfica de se representar o arco inibidor, é através de uma linha com círculo na ponta, representada na figura 2.5.

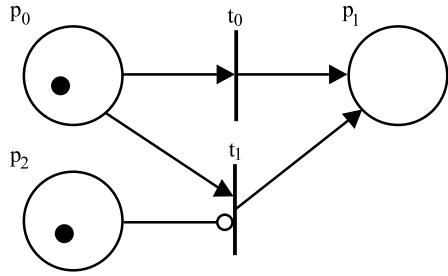


Figura 2.5: Grafo de uma rede de Petri utilizando arco inibidor.

Na figura 2.5, a transição  $t_1$  não pode disparar pois está desabilitada pela presença de uma ficha em  $p_2$ . Entretanto, caso  $p_2$  não tivesse uma ficha,  $t_1$  estaria habilitada.

## 2.6 Redes de Petri Interpretadas para Controle

As redes de Petri são muito eficientes na modelagem de sistemas a eventos discretos, porém sua aplicabilidade na modelagem de controladores é limitada. Para este fim, faz-se necessário o uso das redes de Petri interpretadas para controle, ou apenas RPIC.

A estrutura da RPIC é tal que permite a modelagem do controlador, admitindo condições impostas por eventos externos, condições de habilitação de arco imposta pelo grafo, além de poder associar ações aos lugares. Isso nos leva à definição 4.

**Definição 4** *Uma rede de Petri interpretada para controle é uma undécupla:*

$$RPIC = (P, T, Pre, Pos, x_0, In, E, C, \xi, O, Q) \quad (2.4)$$

em que  $(P, T, Pre, Pos, x_0, In)$  é o grafo de uma rede de Petri segura, ou seja, o número de fichas de cada lugar só pode ser zero ou um;  $E = \{e_j : t_j \in T_0\}$  e  $C = \{c_j : t_j \in T_0\}$ , que são respectivamente, os conjuntos de eventos e de condições associados as transições não temporizadas  $T_0$ ; Analogamente,  $\xi = \{d_j \in \mathbb{R} : t_j \in T_d\}$  é o conjunto de atrasos associados as transições temporizadas  $T_d$ ;  $O$  e  $Q$  são, respec-

tivamente, os conjuntos das operações e ações contínuas ou impulsionais associadas aos lugares. Para condições  $c_j$  não especificadas, é atribuído  $c_j = 1$ . Por fim, se o evento  $e_j$  não é especificado, é considerado como o evento sempre ocorrente.

□

As redes de Petri Interpretadas para Controle são divididas em dois módulos distintos chamados, respectivamente, de módulo de processamento de dados e módulo de controle. A figura 2.6 ilustra o fluxo de dados, bem como os módulos discriminados de uma rede de Petri Interpretada para Controle.

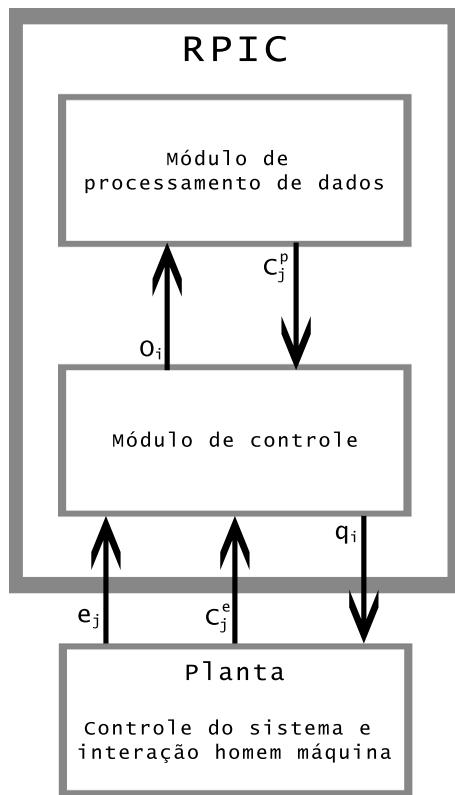


Figura 2.6: Fluxo de dados em uma rede de Petri interpretada para controle.

No módulo de processamento de dados são executadas as operações através das ordens de operação  $O_i$  recebidas do controlador a eventos discretos, em seguida este retorna as informações ao controlador, relacionadas às condições internas associadas aos dados processados  $C_j^p$ ; A RPIC recebe informações do ambiente, que consistem em variáveis booleanas  $C_j^e$  e eventos  $e_j$ , e envia ações  $q_i$  de saída para o ambiente. As ações de saída podem ser impulsionais, contínuas ou variáveis numéricas. Note

que o modelo pressupõe que o cálculo é feito de forma instantânea, logo, o resultado está disponível imediatamente.

Na figura 2.7 estão representados um lugar e uma transição de uma RPIC. A transição  $t_j$  irá disparar quando habilitada, se somente se, a condição  $c_j$  for verdadeira e, quando o evento  $e_j$  ocorrer. Quando o lugar  $p_i$  recebe uma ficha, uma ação contínua  $q_i$  é iniciada e uma operação  $O_i$  é realizada. A ação contínua é mantida enquanto houver ficha em  $p_i$ .

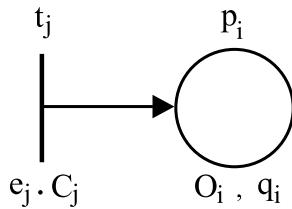


Figura 2.7: Representação gráfica de uma rede de Petri interpretada para controle.

## 2.7 Formas de Obtenção de Redes de Petri

A estruturação de um modelo em redes de Petri tem como objetivo facilitar a composição da rede, oferecendo ferramental e metodologia para iniciar a criação da mesma. As três principais abordagens são: (i) *Bottom-up*, ou composição modular; (ii) *Top-down*, ou refinamentos sucessivos; (iii) híbrida. Pode-se entender melhor cada uma delas no transcorrer das subseções 2.7.1, 2.7.2 e 2.7.3.

### 2.7.1 Composição Modular

Pode-se resumir a composição modular - *Bottom-up* - como decomposição e composição. A decomposição envolve a divisão do sistema em vários subsistemas, de forma que um sistema complexo possa ser decomposto em sistemas menores mais simples. Em seguida, faz se a composição hierárquica, na qual, esses subsistemas são integrados formando o modelo completo do sistema.

### **2.7.2 Refinamentos Sucessivos**

Neste método, o processo é iniciado com uma versão simplificada do projeto, e então, aplicar refinamentos sucessivos de lugares, transições ou subestruturas, que é o equivalente a substituir estes por sub-redes mais complexas ou módulos pré-projetados. A cada evolução do sistema, o nível de detalhamento é refinado, até que se alcance o nível desejado. O método de refinamentos sucessivos é mais eficiente em estruturas de tarefas hierárquicas, onde cada tarefa consiste de sub-tarefas associadas.

### **2.7.3 Híbrida**

Como o próprio nome sugere, consiste na combinação dos métodos das subseções 2.7.1 e 2.7.2. Enquanto se obtém o refinamento da rede, passo a passo, incluindo assim os detalhes pertinentes a operação, através do método de refinamentos sucessivos; a composição modular visa a correta construção de interações entre os subsistemas ou o correto detalhamento da operação dos processos.

# Capítulo 3

## Controlador Lógico Programável

Controlador lógico programável, conhecido também como **CLP** ou da sua sigla em inglês **PLC** (*Programmable logic controller*), é um dispositivo eletrônico digital projetado para automatizar sistemas industriais. Basicamente, o CLP é um dispositivo capaz de armazenar instruções, que utiliza um conjunto de entradas digitais ou analógicas para, baseado em sua programação, comandar um conjunto de saídas digitais ou analógicas [1–3, 7].

### 3.1 História e Definição

Sua invenção data de 1968, quando o engenheiro Richard Morley da General Motors, percebendo a grande necessidade das indústrias manufatureiras de possuir uma solução modular, de fácil modificação e que permitisse alterações sem a necessidade de grandes modificações elétricas e mecânicas, desenvolveu junto com sua equipe o primeiro CLP. Desde então, o CLP sofreu inúmeras modificações, se tornando um dispositivo indispensável nas indústrias e processos automatizados.

O CLP se integra a planta industrial por intermédio de sensores e atuadores. Os sensores são dispositivos de entrada, capazes de traduzir condições físicas em sinais elétricos que possam ser interpretados pelo CLP. Já os atuadores, são os dispositivos de saída, que convertem uma instrução do CLP em sinal elétrico, capaz de executar alguma ação física na planta. Através desse modelo aparentemente

simples, o conjunto tem aplicação praticamente ilimitada, podendo ser utilizado virtualmente em qualquer processo industrial que demande automação.

O exemplo 3 ilustra de forma simplificada todo o conceito e funcionamento.

**Exemplo 3** Deseja-se automatizar um processo simples, que consiste em ligar uma eletrobomba quando o nível de água estiver abaixo da metade do reservatório e desligar quando o nível chegar ao topo. Os sensores de nível fornecem sinal lógico 1 na presença de líquido e 0 na ausência. O atuador por sua vez, liga a eletrobomba quando recebe sinal 1 e desliga com sinal 0. A programação no CLP é feita de forma que ao receber em sua entrada nível 0 do sensor 2 (nível de água abaixo da metade), sua saída vai a nível lógico 1 (liga a eletrobomba), permanecendo em 1, até que receba em sua entrada sinal de nível lógico 1 do sensor 1 (nível de água no topo) e então mude a saída para nível 0 (desliga a eletrobomba). Note que a figura 3.1 está fora de escala e foi omitida a parte de potência.

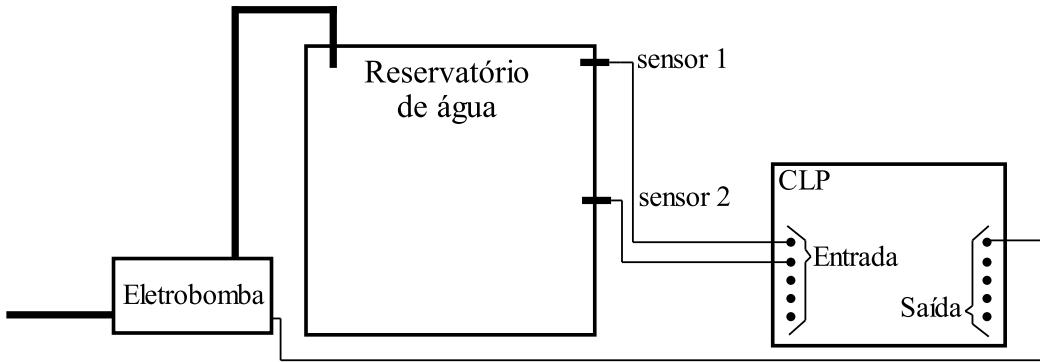


Figura 3.1: Esquema ilustrativo de automação com CLP do exemplo 3.

O CLP tem um funcionamento robusto, que consiste em realizar ciclos de varredura. O ciclo de varredura pode ser dividido basicamente em três etapas: leitura do registro de entradas; leitura e execução do programa; atualização do registro de saídas. É na terceira etapa que também é realizada a atualização de outras variáveis que representam resultados aritméticos, de contagem e temporizadores utilizados no programa. Concluídas as etapas, o ciclo de varredura retorna à primeira etapa onde então um novo ciclo de varredura se inicia. O tempo gasto para o CLP executar

cada ciclo de varredura é chamado de tempo de varredura; o tempo de varredura depende de vários fatores, mas em geral é da ordem de dezenas de milissegundos [7]. A figura 3.2 representa o CLP em módulos.

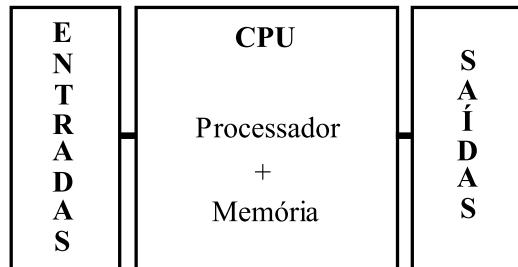


Figura 3.2: Esquema ilustrativo de um CLP.

## 3.2 Linguagens de Programação

As linguagens de programação em que o CLP pode ser programado, definidas pela norma internacional IEC61131-3 [8], são: (i) diagrama bloco de funções; (ii) diagrama *Ladder*; (iii) sequenciamento gráfico de funções (em inglês, sequential function chart - SFC); (iv) lista de instruções e (v) texto estruturado [1].

Entre as cinco linguagens, o diagrama *Ladder* é o mais utilizado pela indústria e está disponível em quase todos os CLPs.

# Capítulo 4

## Diagrama *Ladder*

No capítulo 3 foram apresentados os conceitos sobre os controladores lógico programáveis e suas principais características. Este capítulo apresenta a linguagem de programação de CLPs mais usada, o diagrama *Ladder*, mostrando seus componentes principais e, em seguida, um método de conversão de redes de Petri Interpretadas para Controle para diagramas *Ladder* é apresentado. O método de conversão é uma forma útil de obtenção da linguagem *Ladder*, a ser usado no CLP, com a vantagem de ter sido previamente modelada em rede de Petri. A programação direta em *Ladder*, apesar de possível, não é interessante já que, devido a complexidade dos sistemas, é difícil garantir um modelo robusto e otimizado [1–3, 7–9].

### 4.1 Componentes de um Diagrama *Ladder*

A lógica *Ladder* consiste basicamente em controlar o acionamento de saídas, dependendo da combinação lógica dos contatos de entrada.

A técnica adotada para descrever uma função lógica utilizando contatos e relés é o diagrama de contatos *Ladder*. De notação simples, o diagrama de contatos é composto de duas barras verticais, que simulam os polos positivo e negativo de uma bateria. Os componentes que interligam as duas barras verticais, comandam o fluxo de “corrente virtual” entre elas. Essa “corrente virtual” flui sempre do polo positivo em direção ao negativo como na figura 4.1.

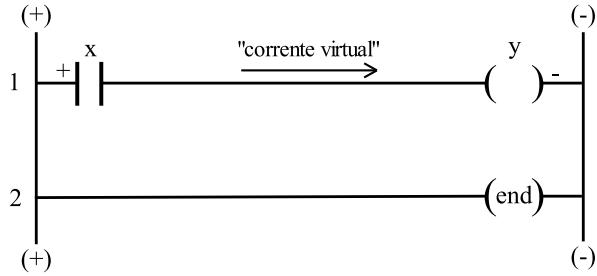


Figura 4.1: Ilustração do diagrama Ladder básico.

O *Ladder* também possui além de contatos e bobinas, temporizadores, contadores, instruções comparativas, instruções de cálculos matemáticos elementares, e complexos. Contatos, bobinas e temporizadores serão os elementos utilizados neste trabalho.

#### 4.1.1 Contatos

O contato é um dos principais componentes de um diagrama *Ladder*. Representando variáveis digitais que podem ser de entrada ou variáveis auxiliares do programa *Ladder*, são divididos em quatro tipos principais: (i) Contatos normalmente abertos; (ii) contatos normalmente fechados; (iii) contatos detectores de borda de subida; (iv) contatos detectores de borda de descida.

##### Contato NA

O contato normalmente aberto (NA) é fechado quando, a variável booleana a que está associado possuir valor lógico verdadeiro, nível lógico 1. Analogamente, o contato é aberto quando a variável booleana associada estiver em nível lógico 0, ou seja, valor lógico falso. A variável a qual o contato está associado pode ser tanto uma das entradas do CLP, quanto uma variável interna do programa *Ladder*. Na figura 4.2 é possível visualizar um contato normalmente aberto e suas relações.

Nível lógico da variável x	Elemento da lógica de controle	Atuação do contato lógico	Fluxo de corrente lógica
0		NÃO	NÃO
1		SIM	SIM

Figura 4.2: Ilustração do contato NA e sua tabela verdade.

### Contato NF

O contato normalmente fechado (NF) é o dual do NA, ou seja, aberto quando a variável booleana a que está associado possuir valor lógico verdadeiro, nível lógico 1 e, fechado quando a variável booleana associada estiver em nível lógico 0, ou seja, valor lógico falso.

Nível lógico da variável x	Elemento da lógica de controle	Atuação do contato lógico	Fluxo de corrente lógica
0		NÃO	SIM
1		SIM	NÃO

Figura 4.3: Ilustração do contato NF e sua tabela verdade.

Tal qual o contato NA, a variável a qual o contato NF está associado pode ser tanto uma das entradas do CLP, quanto uma variável interna do programa *Ladder*. Na figura 4.3 é possível visualizar um contato normalmente fechado e suas relações.

### Contato Detector de Borda de Subida

O contato detetor de borda de subida é um tipo especial de contato, utilizado para detectar a borda de subida da variável booleana associada a ele. Seu funcionamento é similar ao contato NA, entretanto sua atuação é tal que ao detectar a mudança do nível lógico de 0 para 1 (borda de subida), fecha o contato durante apenas

um ciclo de varredura, abrindo-o assim que o ciclo terminar. Note que, se o nível lógico permanecer 1 no próximo ciclo, o CLP mantém o contato aberto até que seja detectada novamente uma mudança de 0 para 1.

Um exemplo de aplicação para este tipo de contato, consiste em um determinado processo onde se deseja realizar a contagem de peças que passam por um sensor, onde o tempo no qual a peça permanece em frente ao sensor poderia ser superior a duração do ciclo de varredura, ocasionando em uma falsa leitura e possível registro incorreto de mais de uma peça, se fosse utilizado um contato NA comum. Na figura 4.4 é possível visualizar um contato detetor de borda de subida e suas relações.

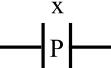
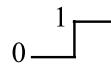
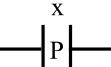
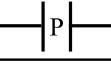
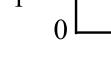
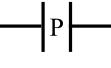
Nível lógico da variável x	Elemento da lógica de controle	Atuação do contato lógico	Fluxo de corrente lógica
0		NÃO	NÃO
0 		SIM, durante apenas um ciclo de varredura	SIM, durante apenas um ciclo de varredura
1		NÃO	NÃO
1 		NÃO	NÃO

Figura 4.4: Ilustração do contato detetor de borda de subida e sua tabela verdade.

A representação gráfica da figura 4.4, utilizando um contato NA com a letra “P” em seu interior, é a notação utilizada em CLPs *Siemens*, podendo diferir nos CLPs de outros fabricantes.

### Contato Detector de Borda de Descida

O contato detetor de borda de descida é o dual do detetor de borda de subida, ou seja, detecta a borda de descida da variável booleana associada a ele. Seu funcionamento é tal que ao detectar a mudança do nível lógico de 1 para 0 (borda de descida), fecha o contato durante apenas um ciclo de varredura, abrindo-o assim que o ciclo terminar.

Na figura 4.5 é possível visualizar um contato detetor de borda de descida e suas relações.

Nível lógico da variável x	Elemento da lógica de controle	Atuação do contato lógico	Fluxo de corrente lógica
0		NÃO	NÃO
0		NÃO	NÃO
1		NÃO	NÃO
1		SIM, durante apenas um ciclo de varredura	SIM, durante apenas um ciclo de varredura

Figura 4.5: Ilustração do contato detetor de borda de descida e sua tabela verdade.

A representação gráfica da figura 4.5, utilizando um contato NF com a letra “N” em seu interior, é a notação utilizada em CLPs *Siemens*, podendo diferir nos CLPs de outros fabricantes.

#### 4.1.2 Bobinas

Assim como os contatos, as bobinas também são parte fundamental de um diagrama *Ladder*. As bobinas são os componentes responsáveis por atualizar as informações de saída e armazenar informações de estado em variáveis internas. De acordo com a lógica associada às bobinas no diagrama *Ladder*, estas modificam o estado lógico de variáveis booleanas. As bobinas podem ser de três tipos diferentes: (i) bobina simples; (ii) bobina *SET*; (iii) bobina *RESET*.

##### Bobinas Simples

As bobinas simples, são as bobinas que alteram o estado de variáveis booleanas, internas ou de saída, para o mesmo nível lógico que se encontram seus terminais em um dado instante. Equivale a dizer que, o nível lógico das variáveis é igual ao nível lógico das bobinas associadas. Desta forma, quando uma bobina é energizada,

o nível lógico de sua variável associada passa a valer 1, e quando desenergizada, o nível lógico da variável associada passa a ser 0. A figura 4.6 mostra a simbologia de uma bobina simples.

Fluxo de corrente lógica	Elemento da lógica de controle	Nível lógico da variável $y$
NÃO	$\overline{-(y)}$	0
SIM	$-(y)$	1

Figura 4.6: Ilustração da bobina simples e sua tabela verdade.

### Bobinas *SET*

As bobinas *SET*, são bobinas que ao serem acionadas, alteram o estado de variáveis booleanas, internas ou de saída, para o nível lógico 1, por tempo indefinido, ou até que seja resetada por uma bobina *RESET*. Ou seja, assim que ativada, retém o nível lógico 1 independente do ciclo de varredura. Isso ocorre pois as bobinas *SET* são retentoras de informação lógica, e dependem que outra linha de comando do *Ladder* mude seu valor lógico para 0. A figura 4.7 mostra a simbologia de uma bobina *SET*.

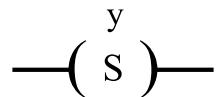


Figura 4.7: Ilustração da bobina *SET* com variável  $y$  associada.

### Bobinas *RESET*

As bobinas *RESET*, são as duais das bobinas *SET*, ou seja, ao serem acionadas, alteram o estado de variáveis booleanas, internas ou de saída, para o nível lógico 0, por tempo indefinido, ou até que seja alterada por uma bobina *SET*. Assim que ativada, descarta o nível lógico 1 armazenado anteriormente. Isso ocorre pois as bobinas *RESET* são retentoras de nível lógico 0. A figura 4.7 mostra a simbologia adotada para uma bobina *RESET*.

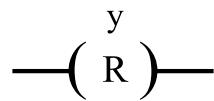


Figura 4.8: Ilustração da bobina *RESET* com variável *y* associada.

### 4.1.3 Temporizadores

Os temporizadores são blocos de funções muito utilizados em automação de processos. Existem diversos tipos de temporizadores, que podem atuar de variadas formas na lógica de processo. Neste trabalho, utilizaremos somente o temporizador do tipo TON (*Timer On Delay*).

#### Temporizador TON (*Timer On Delay*)

As variáveis associadas aos parâmetros de um bloco temporizador de um controlador *SIEMENS*, são PT, ET, IN, e Q, que representam: (i) PT é o tempo de atraso; (ii) ET é o tempo decorrido; (iii) IN sinal lógico na entrada do TON que inicia a contagem de tempo; (iv) Q é a saída com o atraso inserido pelo TON. A figura 4.8 mostra o símbolo adotado em CLPs *Siemens* para o temporizador.

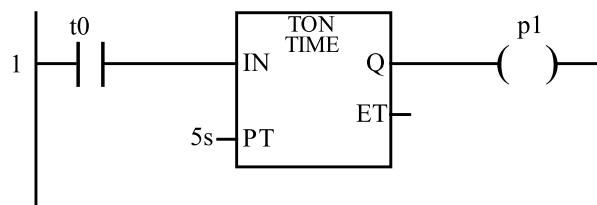


Figura 4.9: Ilustração do temporizador TON.

# Capítulo 5

## Conversão de RPIC para Ladder

Nos capítulos 2, 3 e 4 foram apresentadas as redes e Petri, os CLPs e a linguagem de programação *Ladder*. Entretanto, um importante elo de ligação entre os três, ainda não foi mostrado. Neste capítulo, será apresentado um método de conversão de RPIC para *Ladder*, de forma a fornecer um modelo de conversão, que permita utilizar as teorias de rede de Petri, para obtenção do código *Ladder* para o CLP. O método que será apresentado foi proposto por MOREIRA *et al.* [6].

A conversão de RPIC para *Ladder* é um processo dividido em 5 módulos fundamentais: (i) módulo de inicialização; (ii) módulo de eventos; (iii) módulo de condições; (iv) módulo da dinâmica; (v) módulo das ações.

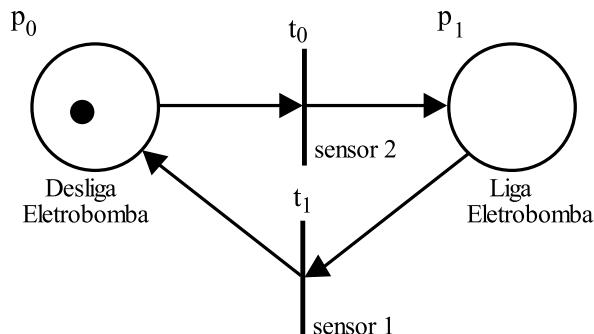


Figura 5.1: RPIC que modela o sistema do exemplo 3.

Para demonstração do método de conversão, será utilizada a RPIC apresentada no exemplo 3. O lugar  $p_0$  está associado a “liga eletrobomba”; o lugar  $p_1$  associado

a “desliga eletrobomba”; a transição  $t_0$  associada ao sensor 2; a transição  $t_1$  está associada ao sensor 1.

## 5.1 Módulo de inicialização

Este é o módulo responsável por inicializar o código, representando a marcação inicial da rede de Petri  $x_0$  (capítulo 2.2). O módulo contém apenas uma linha no diagrama, composta por um contato NF, associado a uma variável binária interna `aux_ini` que, no primeiro ciclo de varredura, energiza bobinas de set associadas aos lugares que contém uma ficha na marcação inicial. Após o ciclo de varredura inicial, o contato NF é aberto. Logo, assim que é executado, o módulo atribui valor lógico 1 às bobinas cujas variáveis booleanas associadas representam os lugares com marcação inicial verdadeira, e abre o contato NF permanentemente. Não é necessário fazer a marcação dos lugares com marcação inicial lógica 0, já que, por padrão, todas as variáveis do CLP iniciam com valor lógico 0. A figura 5.2, mostra o módulo de inicialização da RPIC ilustrada na figura 5.1, cuja marcação inicial é  $p_0 = 1$  e  $p_1 = 0$ .

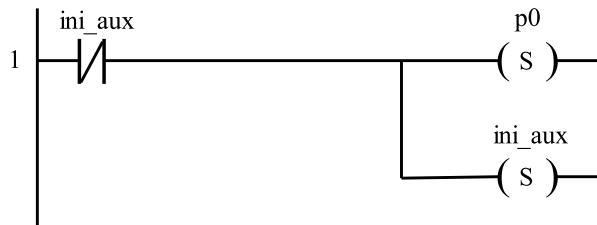


Figura 5.2: Módulo de inicialização do modelo RPIC para Ladder.

## 5.2 Módulo de Eventos Externos

O módulo de eventos externos, como o próprio nome diz, é o módulo associado às entradas do CLP, ou seja, a eventos externos como sinais de sensores. Os módulos de eventos externos, em geral, são associados a contatos do tipo detetor de borda de subida ou descida. O módulo de eventos externos do exemplo 3, que está re-

presentado na figura 5.3, possui os sensores 1 e 2. É importante observar que, em nosso exemplo, o sensor 2 deve detectar a borda de descida, e o sensor 1 a borda de subida.

Note que a notação pode variar dependendo do modelo do CLP utilizado.

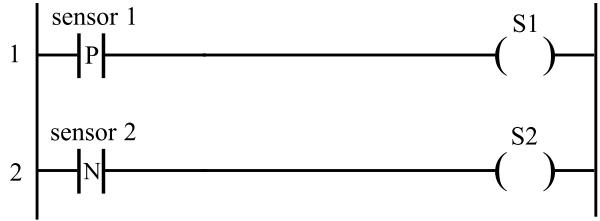


Figura 5.3: Módulo de eventos externos do modelo RPIC para Ladder.

### 5.3 Módulo das Condições

No módulo das condições ficam representadas todas as condições de habilitação das transições da RPIC, e as condições impostas ao sistema tais como o estado dos sensores. A conversão é feita associando-se contatos, que representam os lugares  $p_i$ , em conjunto com as condições impostas, formando assim a lógica de habilitação das respectivas transições  $t_j$ . Uma vez satisfeitas as condições, a bobina que representa a transição é energizada. O módulo das condições do exemplo 3, que está representado na figura 5.4, possui duas linhas referentes às transições  $t_0$  e  $t_1$ .

Os arcos inibidores mostrados na seção ?? devem ser representados neste módulo. Para tal, um contato NF deve ser inserido em série com as condições de habilitação da transição associada, de forma que quando habilitado, desabilite a transição associada, funcionando assim como um arco inibidor. Ou seja, o contato NF quando em nível lógico 1 abre, impedindo que a bobina que representa a variável booleana associada a transição seja ativada.

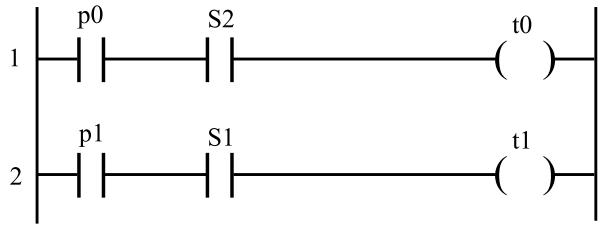


Figura 5.4: Módulo das condições do modelo RPIC para Ladder.

## 5.4 Módulo da Dinâmica

Após o disparo de uma transição  $t_j$ , o número de fichas da rede de Petri deve ser atualizado. Esse processo é descrito pela equação de estado da rede de Petri. Portanto, para a construção do módulo que representa a dinâmica da rede de Petri é utilizada a sua matriz incidência. O módulo da dinâmica possui  $m$  linhas, de forma que cada linha é associada a uma transição  $t_j$ . Uma bobina de *SET* ou *RESET* é usada para alocar o valor um ou zero à variável binária que representa o número de fichas de um lugar da rede de Petri. Então, após a habilitação das transições, ocorre o disparo simultâneo, levando a uma nova marcação da rede de Petri.

O disparo simultâneo de transições rotuladas com o mesmo evento  $e_o$  da RPIC pode levar à situação em que a transição de saída de um lugar que tem uma ficha,  $p_i$ , dispara ao mesmo tempo em que uma transição de entrada de  $p_i$  também dispara. Nesse caso,  $p_i$  precisa continuar com uma ficha após a ocorrência do evento  $e_o$ . Dependendo de como foi implementado o *Ladder* da dinâmica da rede de Petri, a marcação do lugar  $p_i$  pode, de forma incorreta, ser igual a zero após a ocorrência de  $e_o$ . Uma maneira simples de contornar esse problema é considerar duas linhas ao invés de uma para representar a mudança de marcação dos lugares após o disparo de uma transição  $t_j$ . Na primeira linha, uma associação em série de contatos NF é adicionada para verificar se uma transição de entrada do único lugar de entrada de  $t_j$  satisfaz as condições de disparo. Se satisfizer, o lugar de entrada de  $t_j$  deve permanecer com uma ficha, o que implica que a bobina de *RESET* associada com

o lugar de entrada de  $t_j$  não pode ser energizada. A segunda linha garante que as bobinas SET dos lugares de saída de  $t_j$  são energizadas.

Note que, arcos inibidores não afetam o módulo da dinâmica da rede de Petri, apenas afetam o módulo das condições como descrito na seção 5.3.

O módulo da dinâmica do exemplo 3, que está representado na figura 5.5, possui duas linhas referentes as variáveis booleanas associadas as transições  $t_0$  e  $t_1$ .

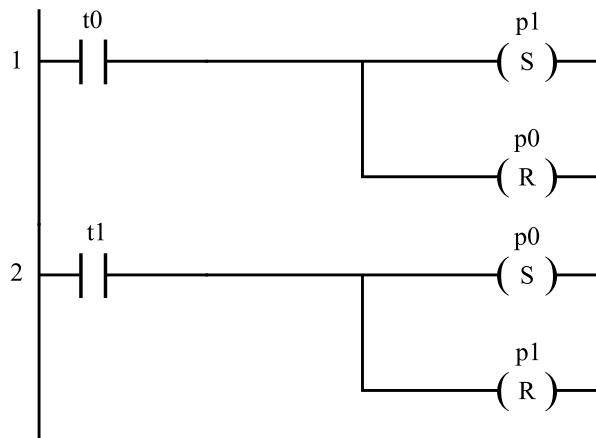


Figura 5.5: Módulo da dinâmica do modelo RPIC para Ladder.

## 5.5 Módulo das Ações

O módulo das ações é responsável por associar bobinas de saída do CLP, aos lugares da RPIC que possuem ações associadas. Desta forma, quando o lugar  $p_i$  que possui uma ação associada receber uma ficha, o CLP vai comandar a ação externa, a ele associada. As ações podem ser de dois tipos: (i) Contínuas; são ações que ocorrem apenas enquanto o lugar  $p_1$  assume valor lógico 1. (ii) Impulsionais; são ações que ocorrem quando há uma mudança no estado lógico do lugar.

A conversão é feita associando-se contatos, que representam as variáveis booleanas associadas aos lugares, às respectivas bobinas de saída do CLP. O módulo das ações do exemplo 3, que está representado na figura 5.6, possui apenas uma linha, pois não é necessário a ação “desliga eletrobomba”, apenas a ação “liga eletrobomba”. Isso acontece pelo fato da bobina de saída estar associada ao lugar

$p_1$  de forma que quando  $p_1$  assume valor lógico 1, a saída encontra-se “ligada” e quando quando  $p_1$  assume valor lógico 0, a saída encontra-se “desligada”. Esse é um exemplo de ação contínua.

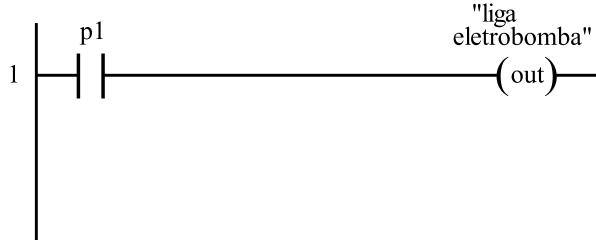


Figura 5.6: Módulo das ações do modelo RPIC para Ladder.

## 5.6 Informações Complementares

O método MOREIRA *et al.* [6], da forma como foi organizado, permite evitar o efeito avalanche porque, nesta ordem, as condições de disparo das transições são verificadas no módulo das condições, e somente depois, é feita a evolução das fichas [1]. Além disso, o método permite a obtenção da rede de Petri original, a partir de um diagrama *Ladder* pronto, apenas por observação. O código obtido a partir do método é claro e bem estruturado, fato que viabiliza futuras alterações no código original, tornando-o adaptável.

O efeito avalanche é um problema de implementação decorrente da transposição de duas ou mais transições, fazendo com que estados que deveriam ser alcançados na rede de Petri sejam ignorados no diagrama *Ladder*.

# Capítulo 6

## Elevadores

O elevador é um dispositivo de transporte, utilizado para mover bens ou pessoas, verticalmente ou diagonalmente. Existem dois tipos principais de elevadores; Elevadores Hidráulicos e Elevadores de Tração. O primeiro, é constituído basicamente de um pistão hidráulico, conectado a um sistema de bombas e válvulas, capazes de direcionar e pressurizar o fluido hidráulico acionando o pistão que se estende movendo o elevador. Já o segundo, movido a tração, é o sistema mais utilizado em todo mundo, e consiste em utilizar cabos de aço, polias e contrapesos, acionados por motor elétrico, para fornecer a força motora para levar a cabina ao ponto desejado. [3, 10–12].

Neste capítulo serão abordados os aspectos construtivos dos elevadores e seu funcionamento, bem como uma apresentação de seus principais componentes.

### 6.1 Aspectos Construtivos Gerais

Nos elevadores de tração, a cabina é sustentada por cabos de aço, presos a um sistema de roldana com cavidades especiais para impedir o deslizamento do cabo. A roldana é acoplada à máquina de tração, que em geral é um motor a indução acionado por um conversor eletrônico, por meio de uma transmissão mecânica ou um eixo direto. Quando a máquina de tração entra em movimento, aciona a roldana que por sua vez move a cabina para cima ou para baixo. Normalmente, o conjunto

de roldanas, máquina de tração e o sistema de controle ficam na casa de máquinas do elevador, localizada acima do último pavimento atendido pelo mesmo.

Os cabos de aço são conectados também a um sistema de contrapesos, que em geral possuem o peso da cabina acrescido de 40% do seu valor de carga nominal. Em outras palavras, quando a cabina está carregada com 40% de seu valor nominal de capacidade de carga, o conjunto está perfeitamente balanceado.

O propósito do contrapeso é equilibrar ao máximo o conjunto e assim poupar energia, uma vez que o motor precisará fornecer energia apenas para tirar a cabina da inércia. Um sistema sem contrapesos necessitaria de máquinas muito maiores e acarretaria em um desperdício enorme de energia.

Ambos, cabina e contrapeso, são fixados a um sistema de trilhos guia, localizados ao longo de toda extensão do poço do elevador. As guias evitam que o carro e o contrapeso balancem em excesso, e servem também de fixação para o sistema de freios de emergência [12].

### 6.1.1 Máquina de Tração

Responsável pelo movimento do elevador, a máquina de tração é quem converte a energia elétrica proveniente da rede em energia cinética para realizar o trabalho de mover a cabine. Usualmente, o motor elétrico escolhido para este fim pode ser de quatro tipos: (i) Motor CA de uma velocidade; (ii) Motor CA de duas velocidades; (iii) Motor a indução acionado por conversor eletrônico; (iv) Motor de corrente contínua [11].

A principal diferença de cada máquina é o perfil de velocidade. Pode-se destacar que as máquinas CA de apenas uma velocidade não são mais usadas devido ao fato de não atenderem aos rígidos padrões de conforto e consumo elétrico, e também, que os motores CC estão caindo em desuso, devido a complexidade e tamanho dos acionamentos que, em geral, são conversores do tipo *Ward-Leonard* Clássicos. O uso de motores CC somente é justificado em casos de modernização de antigos elevadores, nos quais não há necessidade de trocar a máquina CC, troca-se apenas o acionamento

por um sistema eletrônico, chamado também de *Ward-Leonard* Estático [9].

A máquina CA de duas velocidades é semelhante a CA de uma velocidade, mas com a vantagem de poder diminuir sua velocidade em dois estágios, suavizando a parada da cabina.

O motor de indução com conversor eletrônico, ou da sigla em inglês "VVVF" (*Variable Voltage, Variable Frequency*), por ser uma solução com controle de velocidade preciso e atender a todos os parâmetros de conforto e consumo de energia elétrica, é atualmente a melhor opção para aplicação em elevadores.

### 6.1.2 Sistema de Freio

O sistema de freio de um elevador de passageiro é o responsável por desacelerar a cabina e mantê-la estacionada nos pavimentos. Em sistemas que utilizam o motor a indução com conversor eletrônico, o próprio motor é responsável por desacelerar a unidade, e o freio serve apenas para mantê-la parada. Além disso, o sistema tem de estar energizado para liberação do freio, logo, na ausência de energia elétrica o sistema freia automaticamente evitando a queda da cabine [10, 11].

### 6.1.3 Portas do vão e da Cabina

As portas automáticas utilizadas em elevadores de passageiros são mais do que apenas item de conforto. Um refinado sistema de acoplamentos e sensores garante que a porta tenha sua abertura condicionada à presença da cabina no pavimento. Além disso, sensores impedem o movimento da cabina caso não haja a confirmação de fechamento de todas as portas do conjunto, inferindo assim um nível de segurança extra, contra quedas no vão aberto. Portas modernas possuem também um dispositivo fotoelétrico capaz de detectar obstáculos que impeçam seu fechamento acidental, evitando assim, cortes e esmagamentos [10, 11].

#### **6.1.4 Sistemas de Segurança**

Apesar de inspirar medo em muitas pessoas, os elevadores são máquinas extremamente seguras. Existem vários sistemas redundantes que garantem que a cabina nunca caia. O primeiro deles são os próprios cabos de aço. Cada cabo de aço individual é projetado para suportar o peso de todo o conjunto, cabina, carga nominal e contrapeso. Apesar disso, os elevadores possuem múltiplos cabos de aço, sendo, em geral, usados oito cabos. Mesmo com a ruptura de um dos cabos, o elevador permanecerá no lugar.

Caso haja uma falha maior e todos os cabos, ou alguma peça de sustentação da roldana venha a falhar, sistemas integrados de frenagem de emergência atuarão evitando a queda. O principal sistema integrado de frenagem é ativado por força centrífuga. Cabos presos a cabina são acoplados ao regulador de velocidade, que gira livremente, dentro de uma faixa pré determinada de velocidade. Quando a velocidade da cabina atinge a velocidade máxima nominal do regulador, a força centrífuga do giro ativa travas que bloqueiam o regulador, tensionando os cabos, que por sua vez ativam os freios de emergência. Estes cabos não exercem função de sustentação da cabine, apenas ativam o freio [12].

## **6.2 Sistema de Automação**

Os elevadores modernos possuem seu controle baseado em CLPs. O trabalho do CLP é processar todas as informações relevantes acerca do estado do elevador, e então atuar acionando o motor de forma precisa para que a cabina possa ser posicionada corretamente e respeitando as solicitações externas. Para que este processo funcione, o CLP necessita de três informações básicas, são elas:

- O destino da solicitação, ou seja, para onde a cabina deve ir;
- A posição exata de cada pavimento;
- A posição atual da cabina;

Descobrir para onde a cabina deve ir é simples. As botoeiras localizadas no interior da cabina e em cada pavimento se encarregam de levar o sinal elétrico para o CLP com a informação das solicitações. Existem várias maneiras de monitorar a posição dos pavimentos e da cabina. Nos sistemas elementares, é comum o uso de sensores ópticos ou magnéticos.

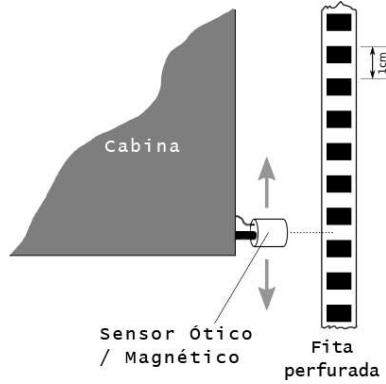


Figura 6.1: Ilustração do sensor de posição da cabina.

Sensores ópticos funcionam lendo uma espécie de fita perfurada fixada em toda a extensão do poço do elevador, ilustrado na figura 6.1. Pela contagem dos furos, a posição da cabina em relação aos pavimentos é facilmente obtida. Esse sistema permite também que o CLP varie a velocidade do motor, de forma a diminuir o gradiente de aceleração e desaceleração da cabina, na etapa de nivelamento com os pavimentos [12].

### 6.2.1 Sensores - Dispositivos de Monitoração

Sensores são dispositivos responsáveis por mensurar as grandezas e estados físicos do sistema, convertendo as em sinais elétricos que os CLPs são capazes de interpretar. Existem diversos sensores utilizados pelo sistema de controle de elevadores que merecem ser destacados.

## **Encoder**

Sensor óptico, responsável por fazer a leitura de velocidade e posição da cabina. Podem ser do tipo linear ou rotativo; enquanto os lineares estão localizados no vão do elevador, os rotativos são acoplados a algum elemento rotativo, como o motor. São robustos, precisos e de longa vida útil.

## **Sensor Anti-esmagamento da Porta da Cabina**

Sensor anti-esmagamento, também conhecido como sensor cortina de luz, é um dispositivo composto por uma série de emissores e receptores infravermelho, instalados em ambos os lados das portas da cabina. Quando algo se encontra entre as portas, o facho infravermelho é interrompido e o sensor envia um sinal lógico para o CLP indicando obstrução da porta.

## **Sensores Mecânicos**

Sensores mecânicos, ou chaves de fim de curso, são sensores que necessitam que haja um contato físico entre o objeto e o sensor. De construção simples, podem ser utilizados em diversas aplicações como detectar abertura e fechamento de portas, deslocamento máximo e até a posição da cabina.

### **6.2.2 Lógica de Atendimento de Chamadas**

Existem dois tipos principais e já consolidados princípios de atendimento de chamados, são eles: (i) Princípio coletivo, no qual o elevador atende os chamados de forma sequencial, de acordo com a proximidade dos pavimentos solicitantes, respeitando o sentido do movimento; (ii) Princípio seletivo, no qual o elevador seleciona o pavimento mais alto ou mais baixo entre as solicitações, realizando os atendimentos a partir deste, de forma sequencial [3, 11].

### **6.2.3 Comando Automático Coletivo**

No manual da empresa SCHINDLER [11] é dito que o comando automático coletivo é caracterizado pela existência de botões de chamada, um para cada pavimento, instalados na cabina, e possuir um único botão de chamada instalado em cada pavimento, todos ligados ao painel central, de tal maneira que todas as chamadas fiquem nele registradas. O carro vai efetuando as paradas em ordem sequencial independentemente da ordem em que as chamadas tenham sido registradas e prossegue no sentido do movimento inicial atendendo a todas as chamadas feitas. Aplica-se a edifícios de poucos andares (de 2 até 3 pavimentos) e pouco movimento, em que o tráfego predominante seja entre andares, como estabelecimentos comerciais e industriais pequenos.

### **6.2.4 Comando Automático Coletivo Seletivo na Descida**

Segundo o manual da empresa SCHINDLER [11], comando automático coletivo seletivo na descida é o comando automático coletivo no qual as chamadas de pavimento somente são atendidas quando o elevador se movimenta em sentido descendente, a partir da chamada superior. Aplica-se a edifícios em que o movimento principal é constituído pelo tráfego entre o térreo e os demais pavimentos, sem que haja tráfego apreciável entre os próprios pavimentos. É, portanto, o sistema ideal para edifícios de apartamentos.

### **6.2.5 Comando Automático Coletivo Seletivo na Subida e na Descida**

É definido segundo o manual da empresa SCHINDLER [11] como sendo o comando automático coletivo no qual existem nos pavimentos intermediários, dois botões, um de “subida” e um de “descida”, e um botão nos pavimentos extremos. Neste sistema de comando as chamadas de pavimento para subir são selecionadas separadamente das chamadas de pavimento para descer, sendo atendidas primeiramente todas as

chamadas em um dos sentidos para depois serem atendidas as de sentido oposto. Aplica-se a edifícios onde o fluxo predominante seja entre os andares, tais como escritórios em geral ou de uma única entidade, repartições públicas, etc. Em edifícios residenciais se aplica ao pavimento térreo sempre que existirem pavimentos inferiores de garagem.

# Capítulo 7

## Obtenção do Código *Ladder* a Partir da RPIC do Sistema de Automação de um Elevador de Passageiros

A aplicação prática deste trabalho consiste em projetar o controlador de um elevador residencial, utilizando-se redes de Petri interpretadas para controle. A partir da RPIC, o método de conversão proposto por MOREIRA *et al.* [6] é aplicado para obtenção do sistema de automação do elevador.

A rede de Petri foi modelada e simulada com o auxílio do software HPSim v1.1. O programa em *Ladder* seguiu a notação do CLP Siemens S7-1200 CPU 1214C disponível no Laboratório de Controle e Automação (LCA) do Departamento de Engenharia Elétrica da UFRJ.

### 7.1 Definições do Sistema

O objetivo deste trabalho é projetar um controlador automático de elevador residencial de passageiros, modelado em RPIC, obtendo posteriormente seu código *Ladder*.

As premissas de projeto são:

- Elevador para três pavimentos;
- Uma boteira de chamada por andar, totalizando três;
- Painel interno da cabina com 3 botões, um para cada andar;
- Porta automática com sensores de fim de curso indicando sua total abertura e total fechamento;
- Três contatos de fim de curso para indicação da posição do pavimento;
- Máquina de tração composta por motor de indução e conversor eletrônico;

Para o controlador automático foi adotada a lógica de atendimento coletivo seletivo na descida, como descrito no capítulo 6.2.4. Sempre que parar em um pavimento, o elevador deve abrir a porta e aguardar 5 segundos antes de continuar com o atendimento das outras solicitações. A cabina só trafega após verificação do fechamento das portas.

### 7.1.1 Informações Complementares dos Componentes

Nesta seção é apresentada uma breve descrição da lógica dos componentes usados, e sua forma de ligação no CLP.

#### Máquina de Tração

A máquina de tração utilizada é um motor a indução acionado por conversor eletrônico *Variable Voltage Variable Frequency* (VVVF). O conversor possui três entradas de controle, são elas: (i) IN-1 aciona o freio interno do motor; (ii) IN-2 motor move a cabina para cima; (iii) IN-3 motor move a cabina para baixo. As entradas do controlador são acionadas por nível lógico 1 de forma contínua, ou seja, executa a ação enquanto for mantido o nível lógico 1 em sua entrada.

Para questões de simplificação, foi assumido que o conjunto controlador e máquina de tração é capaz de realizar a parada da cabina sem trancos, praticamente de forma instantânea.

### **Sensores de Fim de Curso para Indicação da Posição do Pavimento**

Os sensores de fim de curso são sensores simples, de acionamento mecânico, que fornecem nível lógico 1 quando a cabina se encontra nivelada ao pavimento.

### **Botoeiras dos Andares e da Cabina**

As botoeiras são do tipo contato normalmente aberto.

### **Sistema de Freio**

O elevador possui sistema de frenagem incorporado à máquina de tração.

### **Porta Automática**

A porta automática possui duas entradas, uma para acionar a abertura e outra para o fechamento, e ainda é dotada de dois sensores de fim de curso, um para indicar que a porta está aberta e outro para indicar que a porta está fechada.

## **7.2 Obtenção do Controlador em RPIC**

A RPIC segue o funcionamento e definições para redes de Petri interpretadas para controle descritos no capítulo 2. Para a obtenção da rede de Petri interpretada pra controle do elevador proposto, alguns passos foram seguidos. Neste projeto foi utilizado o método de composição *Bottom-up*, apresentado na seção 2.7.1. Partindo do sistema mais geral, foi feita a divisão em vários subsistemas, e em seguida fez-se a composição hierárquica para se obter a RPIC desejada. Foi optado por uma rede de Petri segura, ou seja, arcos com peso unitário e lugares que podem ter no máximo uma ficha em cada estado alcançável do sistema.

### **7.2.1 Análise das Etapas de Atendimento**

Um ciclo normal de atendimento no modelo proposto engloba de forma geral as seguintes etapas: (i) cabina parada e sem registros de chamadas; (ii) solicitação da cabina por botoeira de pavimento; (iii) o controlador registra a chamada, fecha as portas e executa a rotina de atendimento; (iv) a máquina de tração é acionada fazendo a cabina mover-se até o pavimento solicitante, parando e abrindo as portas; (v) o passageiro seleciona o andar desejado através da botoeira da cabina. (vi) o controlador registra a chamada, fecha as portas e executa a rotina de atendimento; (vii) a máquina de tração é acionada fazendo a cabina mover-se até o pavimento solicitado, finalizando o atendimento.

A lógica de atendimento implementada foi a coletiva, seletiva na descida. De forma que as chamadas dos pavimentos dois e três são atendidas no modo seletivo na descida, ou seja, somente são atendidas quando o elevador está em sentido descendente. De forma análoga, a cabina opera em modo coletivo, ou seja, deve atender as chamadas de forma sequencial obedecendo apenas ao sentido de deslocamento da cabina.

### **7.2.2 Definições e Símbologia Adotada**

O modelo básico foi feito em rede de Petri autônoma, definindo somente as transições e lugares principais do diagrama macro do sistema. Em um primeiro momento, foram definidos os subsistemas necessários para compor a rede de Petri, e as relações entre eles. Posteriormente, todos os módulos foram adicionados e a RPIC foi obtida a partir deste.

A símbologia utilizada segue o modelo do HPSim v1.1. Foram utilizados transições instantâneas e temporizadas, arcos ordinário, habilitador e inibidor, representados na figura 7.1



Figura 7.1: Da esquerda pra direita: Lugar  $P_0$ ; arco ordinário; transição instantânea  $T_0$ ; arco habilitador; lugar  $P_1$ ; arco inibidor e transição temporizada  $T_1$ .

### 7.2.3 Módulos da RPIC

Os módulos da rede de Petri interpretada para controle do sistema de controle do elevador, são:

- Posição da cabina;
- Lógica de atendimento;
- Máquina de tração;
- Botoeiras;
- Porta automática;

#### Posição da Cabina

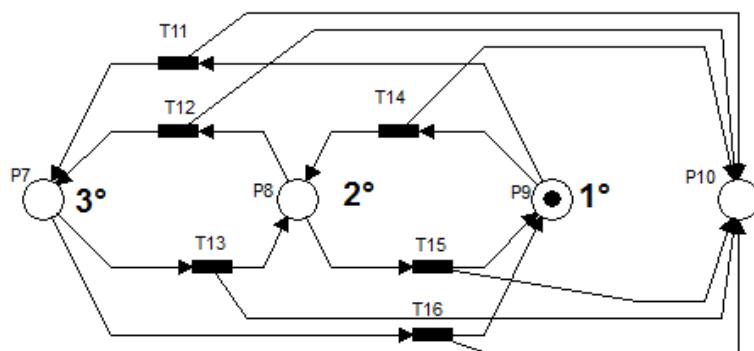


Figura 7.2: Módulo da posição da cabina.

O módulo da posição da cabina, representado na figura 7.2, é o módulo composto por quatro lugares  $P = \{p_7, p_8, p_9, p_{10}\}$ , dos quais  $p_7$ ,  $p_8$  e  $p_9$  representam os pavimentos

atendidos pelo elevador e  $p_{10}$  representa a chegada da cabina ao pavimento, e seis transições  $T_0 = \{t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}\}$ , que representam as mudanças de pavimento e são associadas aos sensores de fim de curso.

## Lógica de Atendimento

O módulo da lógica de atendimento, representado na figura 7.3, é composto por nove lugares  $P = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_{11}, p_{19}\}$ , quatorze transições ordinárias  $T_0 = \{t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{21}, t_{28}, t_{33}\}$  e uma transição temporizada  $T_d = \{t_{17}\}$ .

Os lugares  $p_0$ ,  $p_1$  e  $p_2$  são responsáveis por registrar a chamada, perdendo ficha apenas quando a chamada é atendida. Registrada a chamada, os lugares  $p_4$  e  $p_6$  registram o sentido do movimento da cabina, de acordo com a posição da cabina em relação à chamada. Enquanto o lugar  $p_4$  recebe ficha das transições  $t_0$ ,  $t_1$  e  $t_6$  e perde através da transição  $t_4$ , o lugar  $p_6$  recebe ficha das transições  $t_7$ ,  $t_9$  e  $t_{10}$  e perde através da transição  $t_8$ . Ambos perdem ficha apenas quando todas as chamadas em um mesmo sentido foram atendidas.

Já os lugares  $p_3$ ,  $p_5$  e  $p_{19}$  e as transições  $t_2$ ,  $t_3$ ,  $t_5$ ,  $t_{21}$ ,  $t_{33}$  e  $t_{38}$  são responsáveis por fazer a diferenciação entre chamadas da cabina e chamadas do pavimento, ativando assim a lógica coletiva ou seletiva na descida respectivamente. Todas as chamadas provenientes dos pavimentos só devem ser atendidas quando a cabina estiver em sentido descendente, a partir do mais alto pavimento solicitado, enquanto as chamadas provenientes da cabina devem ser atendidas de forma sequencial, respeitando apenas o sentido do movimento.

O lugar  $p_{11}$  indica que a cabina está livre para realizar atendimentos, deste modo, exerce a função de habilitar as transições  $t_0$ ,  $t_1$ ,  $t_6$ ,  $t_7$ ,  $t_9$  e  $t_{10}$ . A transição temporizada  $t_{17}$  é ativada quando a porta da cabine está aberta e  $p_{11}$  não possui fichas, o que garante a permanência mínima de cinco segundos da cabina nos pavimentos antes de atender a próxima chamada.

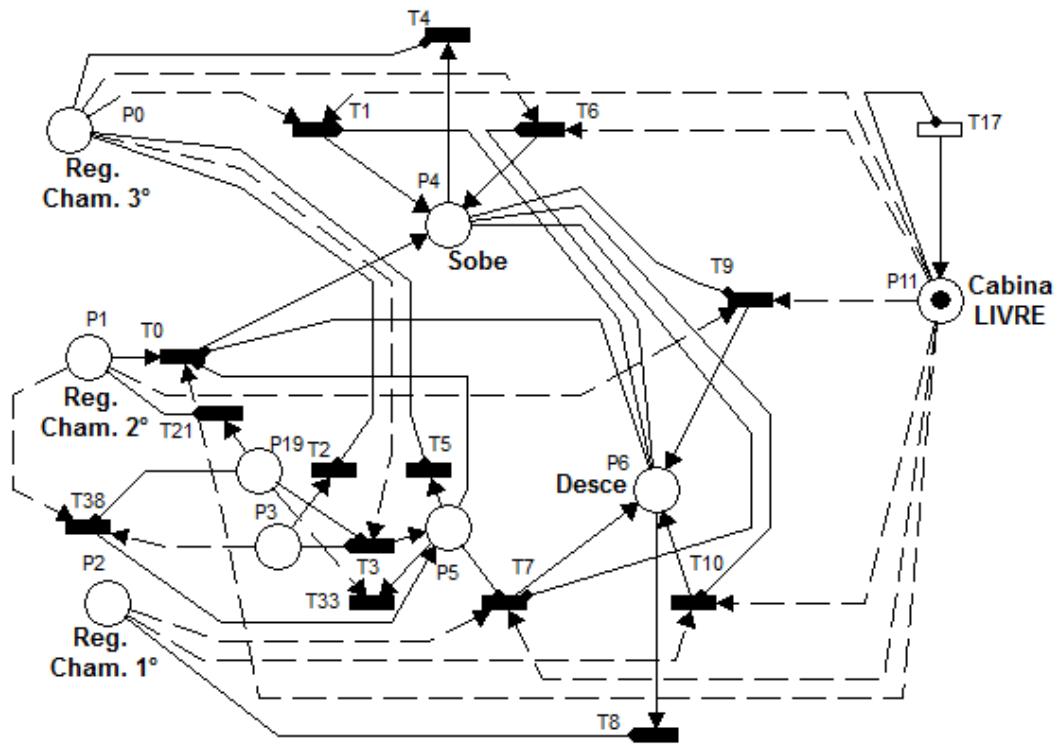


Figura 7.3: Módulo da lógica de atendimento.

### Máquina de Tração

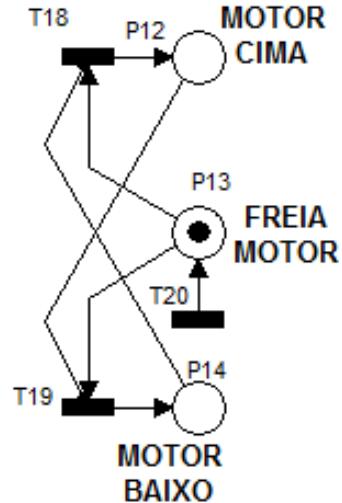


Figura 7.4: Módulo da máquina de tração.

O módulo da máquina de tração representado na figura 7.4, é pequeno, devido ao fato que foi admitido que o conversor de acionamento da máquina tem lógica interna

própria, restando ao controlador do elevador apenas ordenar sua ação. É composto por três lugares  $P = \{p_{12}, p_{13}, p_{14}\}$  e três transições  $T_0 = \{t_{18}, t_{19}, t_{20}\}$ .

Os lugares  $p_{12}$  e  $p_{14}$  são os responsáveis por comandar a máquina em sentido ascendente ou descendente respectivamente. O lugar  $p_{13}$  é o responsável por ativar o freio da máquina. As transições  $t_{18}$  e  $t_{19}$  são responsáveis por enviar a ficha para os lugares  $p_{12}$  e  $p_{14}$ , e também garantem que o motor nunca seja ativado simultaneamente em sentido ascendente e descendente.

A transição  $t_{20}$  é responsável por ativar o freio motor.

### Botoeiras

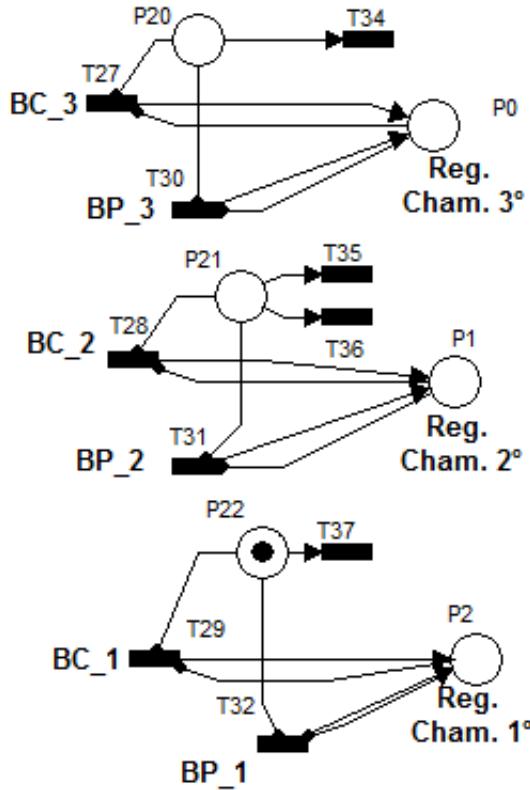


Figura 7.5: Módulo das botoeiras.

O módulo das botoeiras está representado na figura 7.5, é o módulo composto por seis lugares  $P = \{p_0, p_1, p_2, p_{20}, p_{21}, p_{22}\}$  e dez transições  $T = \{t_{27}, t_{28}, t_{29}, t_{30}, t_{31}, t_{32}, t_{34}, t_{35}, t_{36}, t_{37}\}$ .

Os lugares  $p_{20}$ ,  $p_{21}$  e  $p_{22}$  são responsáveis por registrar a posição da cabina pa-

rada, perdendo ficha quando a cabina entra em movimento. Desta forma, inibem as transições  $t_{27}, t_{28}, t_{29}, t_{30}, t_{31}$  e  $t_{32}$  associadas as variáveis de entrada das seis botoeiras de disparar, quando a cabina já se encontra no andar, e liberam o disparo assim que a porta se feche e a cabina entre em movimento, através do disparo das transições  $t_{34}, t_{35}, t_{36}$  e  $t_{37}$ .

Os três lugares  $p_0, p_1$  e  $p_2$  são compartilhados com a lógica de atendimento, e representam o registro da chamada.

## Porta Automática

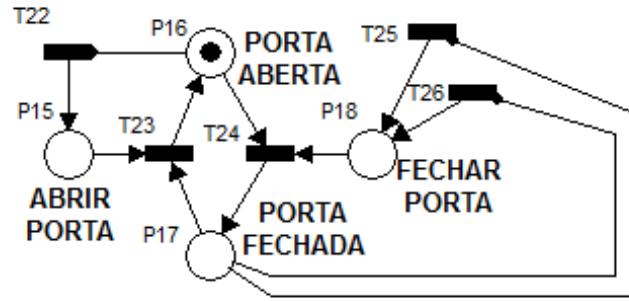


Figura 7.6: Módulo da porta automática.

Por fim, o módulo da porta automática representado na figura 7.6, é o módulo composto de quatro lugares  $P = \{p_{15}, p_{16}, p_{17}, p_{18}\}$  e cinco transições  $T = \{t_{22}, t_{23}, t_{24}, t_{25}, t_{26}\}$ .

Os lugares  $p_{16}$  e  $p_{17}$  representam os estados da porta, aberta e fechada respectivamente. A mudança de ficha entre os dois estados ocorre quando as transições  $t_{23}$  e  $t_{24}$  são ativadas através dos sensores de fim de curso da porta associados a elas.

As transições  $t_{22}, t_{25}$  e  $t_{26}$  são responsáveis por comandar a abertura e fechamento das portas, integrando o módulo aos demais módulos.

#### 7.2.4 RPIC Completa do Sistema

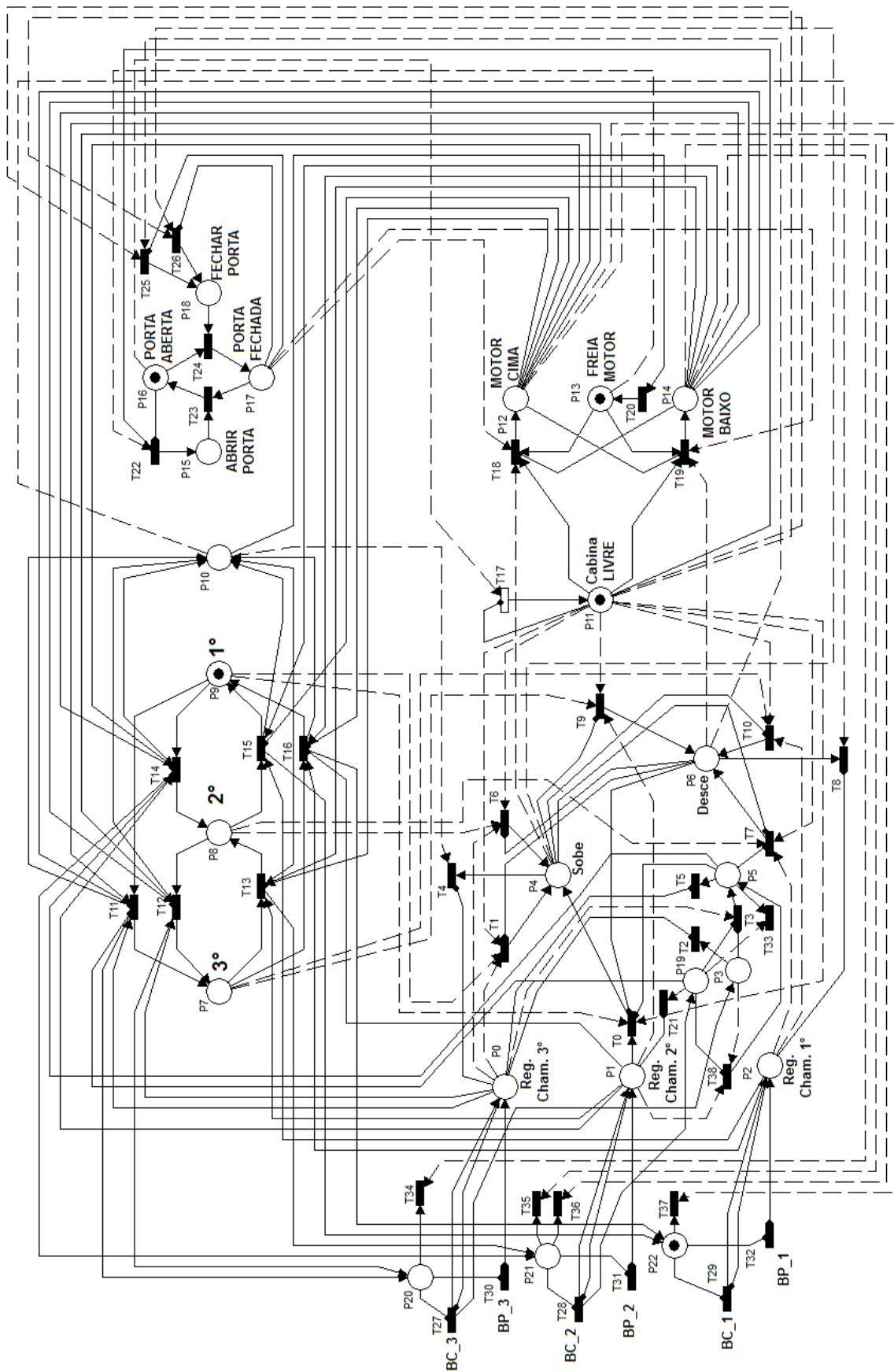


Figura 7.7: RPIC do controlador automático do elevador.

Após criados todos os módulos que compõe a rede de Petri interpretada para controle do elevador de passageiros, foram necessários alguns ajustes e fusões de estados para que juntos, os módulos pudessem formar a RPIC que representa o sistema completo. Completa, a rede de Petri possui vinte e três lugares e trinta e nove transições, que juntas compõem a rede de Petri interpretada para controle que representa o controlador proposto. No total, sessenta e quatro arcos foram usados para interligar os módulos entre si e fazer o conjunto funcionar como uma única rede. Toda a rede foi testada com êxito através do programa HPSim v1.1.

Foram procuradas condições lógicas comuns a cada módulo, de forma que um módulo pudesse fornecer a informação necessária ao funcionamento do outro. Um exemplo é o módulo da lógica de atendimento, que com base na informação da posição da cabina recebida do módulo da posição, no estado do motor recebido do módulo da máquina de tração, no estado das portas recebido do módulo das portas, recebe as chamadas do módulo da botoeira e controla todos os outros módulos de forma a atender a chamada.

Uma forma de entender a rede completa é observar a evolução das fichas em uma dada situação, que consiste em atender a um chamado da botoeira do segundo pavimento, quando o elevador se encontra parado no primeiro pavimento. O processo tem início quando a transição  $t_{31}$  dispara com o evento botão do segundo pavimento pressionado, colocando uma ficha no lugar  $p_1$  que registra a chamada para o segundo pavimento. Em seguida, como a cabina se encontra no primeiro pavimento, a transição  $t_0$  dispara enviando uma ficha para o lugar  $p_4$  que registra o sentido de subida. Como a cabina está livre ( $p_{11}$  possui ficha), a transição  $t_{26}$  dispara, enviando uma ficha para o lugar  $p_{18}$  que comanda o fechamento das portas. Quando o sensor de porta associado a transição  $t_{24}$  registra o fechamento da porta, dispara, enviando uma ficha para o lugar  $p_{17}$  que registra que a porta está fechada, e retira uma ficha do lugar  $p_{18}$ . Logo em seguida, a transição  $t_{18}$  dispara, retirando uma ficha dos lugares  $p_{11}$  e  $p_{13}$ , e enviando uma ficha para o lugar  $p_{12}$  que comanda o acionamento do motor para cima. O motor permanece acionado até que a transição

Tabela 7.1: Estados iniciais, ações e significado dos lugares.

Lugar	Estado inicial	Ação	Estado
$P_0$	0	—	Registro de chamada para o terceiro pavimento
$P_1$	0	—	Registro de chamada para o segundo pavimento
$P_2$	0	—	Registro de chamada para o primeiro pavimento
$P_3$	0	—	Registro de chamada proveniente da cabina
$P_4$	0	—	Registro para subida
$P_5$	0	—	Inibe a cabina de parar no segundo pavimento
$P_6$	0	—	Registro para descida
$P_7$	0	—	Cabina se encontra no terceiro pavimento
$P_8$	0	—	Cabina se encontra no segundo pavimento
$P_9$	1	—	Cabina se encontra no primeiro pavimento
$P_{10}$	0	—	Cabina chegou ao pavimento
$P_{11}$	1	—	Cabina se encontra livre
$P_{12}$	0	motor_cima	
$P_{13}$	1	motor_freia	
$P_{14}$	0	motor_baixo	
$P_{15}$	0	abrir_porta	
$P_{16}$	1	—	Porta da cabina aberta
$P_{17}$	0	—	Porta da cabina fechada
$P_{18}$	0	fechar_porta	
$P_{19}$	0	—	libera a cabina para parar no segundo pavimento
$P_{20}$	0	—	Cabina PARADA no terceiro pavimento
$P_{21}$	0	—	Cabina PARADA no segundo pavimento
$P_{22}$	1	—	Cabina PARADA no primeiro pavimento

$t_{18}$  é disparada pelo evento sensor do pavimento dois, enviando uma ficha para os lugares  $p_8$  e  $p_{10}$  e retirando uma ficha do lugares  $p_1$ ,  $p_9$  e  $p_{12}$ , desligando o motor, indicando que a cabina mudou de pavimento e que a chamada foi atendido. Nesse ponto, as transições  $t_4$  e  $t_{20}$  disparam, retirando uma ficha do lugar  $p_4$  e enviando uma ficha ao lugar  $p_{13}$  respectivamente, ativando o freio motor e apagando o registro de subida. Em seguida, a transição  $t_{22}$  dispara, enviando uma ficha para o lugar  $p_{15}$ , que comanda a abertura das portas. Quando o sensor de porta associado a transição  $t_{23}$  registra a abertura da porta, dispara, enviando uma ficha para o lugar  $p_{16}$  que registra que a porta está aberta, e retira uma ficha do lugar  $p_{15}$ . Por fim, a transição temporizada  $t_{17}$  é habilitada e após cinco segundos dispara, enviando uma ficha para o lugar  $p_{11}$  indicando que a cabina está novamente livre.

Através das tabelas 7.1 e 7.2 pode se visualizado, respectivamente, as ações e estados associados aos lugares, e as receptividades associadas às transições.

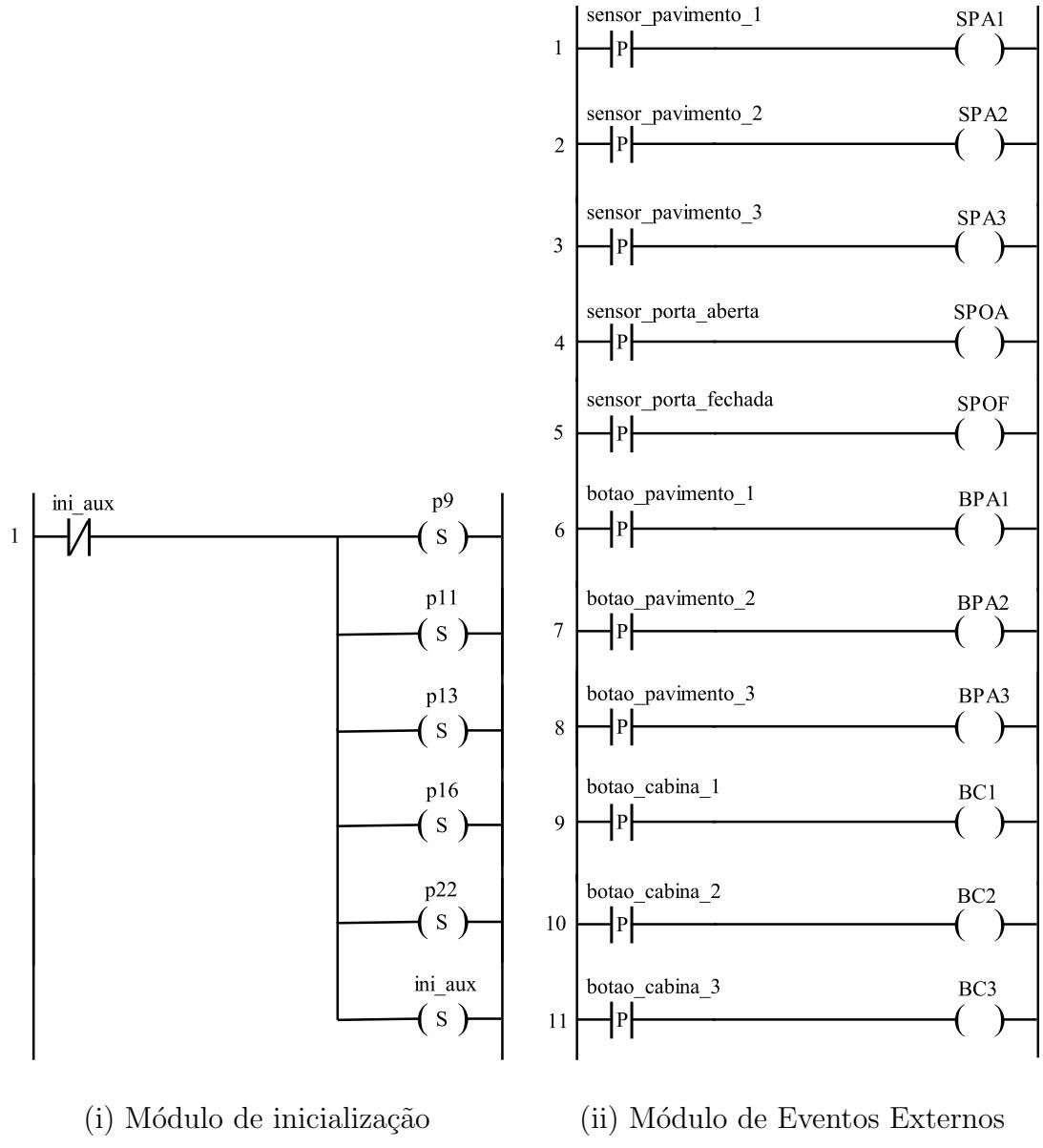
A RPIC completa, pode ser vista na figura 7.7.

Tabela 7.2: Receptividades, eventos e condições associadas às transições.

Transição	Receptividade	Condição	Evento
$T_0$	R0	1	1
$T_1$	R1	1	1
$T_2$	R2	1	1
$T_3$	R3	1	1
$T_4$	R4	1	1
$T_5$	R5	1	1
$T_6$	R6	1	1
$T_7$	R7	1	1
$T_8$	R8	1	1
$T_9$	R9	1	1
$T_{10}$	R10	1	1
$T_{11}$	R11	1	sensor_pavimento_3 ↑
$T_{12}$	R12	1	sensor_pavimento_3 ↑
$T_{13}$	R13	1	sensor_pavimento_2 ↑
$T_{14}$	R14	1	sensor_pavimento_2 ↑
$T_{15}$	R15	1	sensor_pavimento_1 ↑
$T_{16}$	R16	1	sensor_pavimento_1 ↑
$T_{17}$	R17	1	1
$T_{18}$	R18	1	1
$T_{19}$	R19	1	1
$T_{20}$	R20	1	1
$T_{21}$	R21	1	1
$T_{22}$	R22	1	1
$T_{23}$	R23	1	sensor_porta_aberta ↑
$T_{24}$	R24	1	sensor_porta_fechada ↑
$T_{25}$	R25	1	1
$T_{26}$	R26	1	1
$T_{27}$	R27	1	botao_cabina_3 ↑
$T_{28}$	R28	1	botao_cabina_2 ↑
$T_{29}$	R29	1	botao_cabina_1 ↑
$T_{30}$	R30	1	botao_pavimento_3 ↑
$T_{31}$	R31	1	botao_pavimento_2 ↑
$T_{32}$	R32	1	botao_pavimento_1 ↑
$T_{33}$	R33	1	1
$T_{34}$	R34	1	1
$T_{35}$	R35	1	1
$T_{36}$	R36	1	1
$T_{37}$	R37	1	1
$T_{38}$	R38	1	1

### 7.3 Conversão de RPIC para Ladder

A conversão de RPIC para *Ladder* foi feita por inspeção, a partir da figura 7.7, de acordo com o modelo abordado no capítulo 5. A representação do programa em *Ladder* que modela a RPIC do problema proposto pode ser vista nas figuras 7.8, 7.9, 7.10, 7.11, 7.12 e 7.13.



(i) Módulo de inicialização

(ii) Módulo de Eventos Externos

Figura 7.8: Módulos de inicialização (i) e de eventos externos (ii) do controlador do elevador.

$$\underline{x}_0 = [00000000010101001000001]^T \quad (7.1)$$

O módulo de inicialização possui a marcação inicial representada na equação 7.1, ou seja, o elevador se encontra parado, com a porta aberta, no primeiro andar, de cabina livre para realizar atendimentos e o com o freio motor ativado.

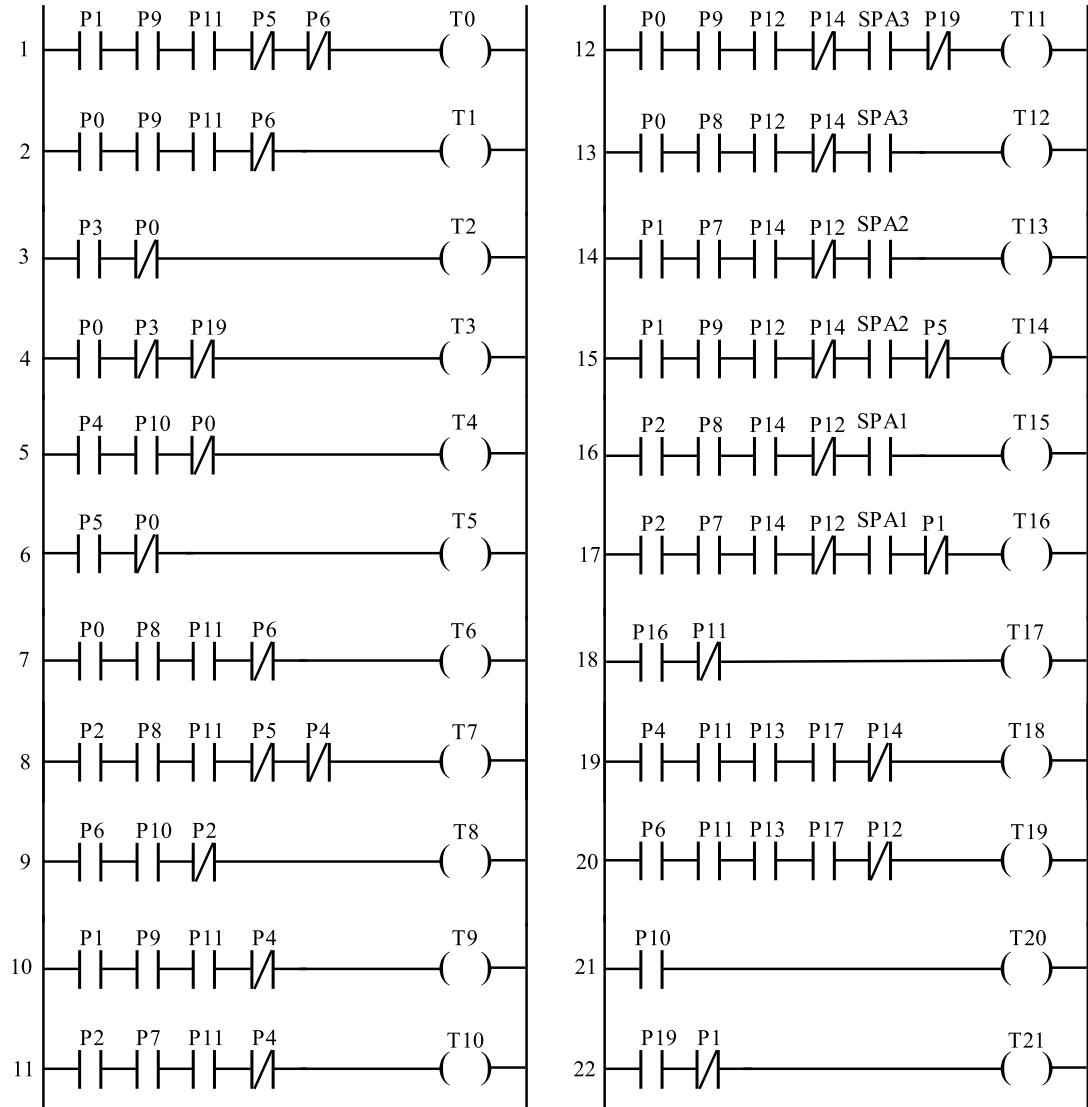


Figura 7.9: Módulo das condições do controlador do elevador.

O módulo de eventos externos é o responsável por detectar a borda de subida dos sensores e botoeiras, e memorizar por um ciclo de varredura esta transição em um variável interna do programa. Memoriza por exemplo que a botoeira do pavimento um foi pressionada e que o sensor do pavimento dois está ativo em um dado instante.

Já o módulo das condições, é responsável por estabelecer as condições de habi-

litação associadas a cada uma das transições. Além das condições relativas a outros lugares, este módulo recebe as condições relativas ao estado das entradas do CLP, que representam condições externas relacionadas ao disparo das transiões.

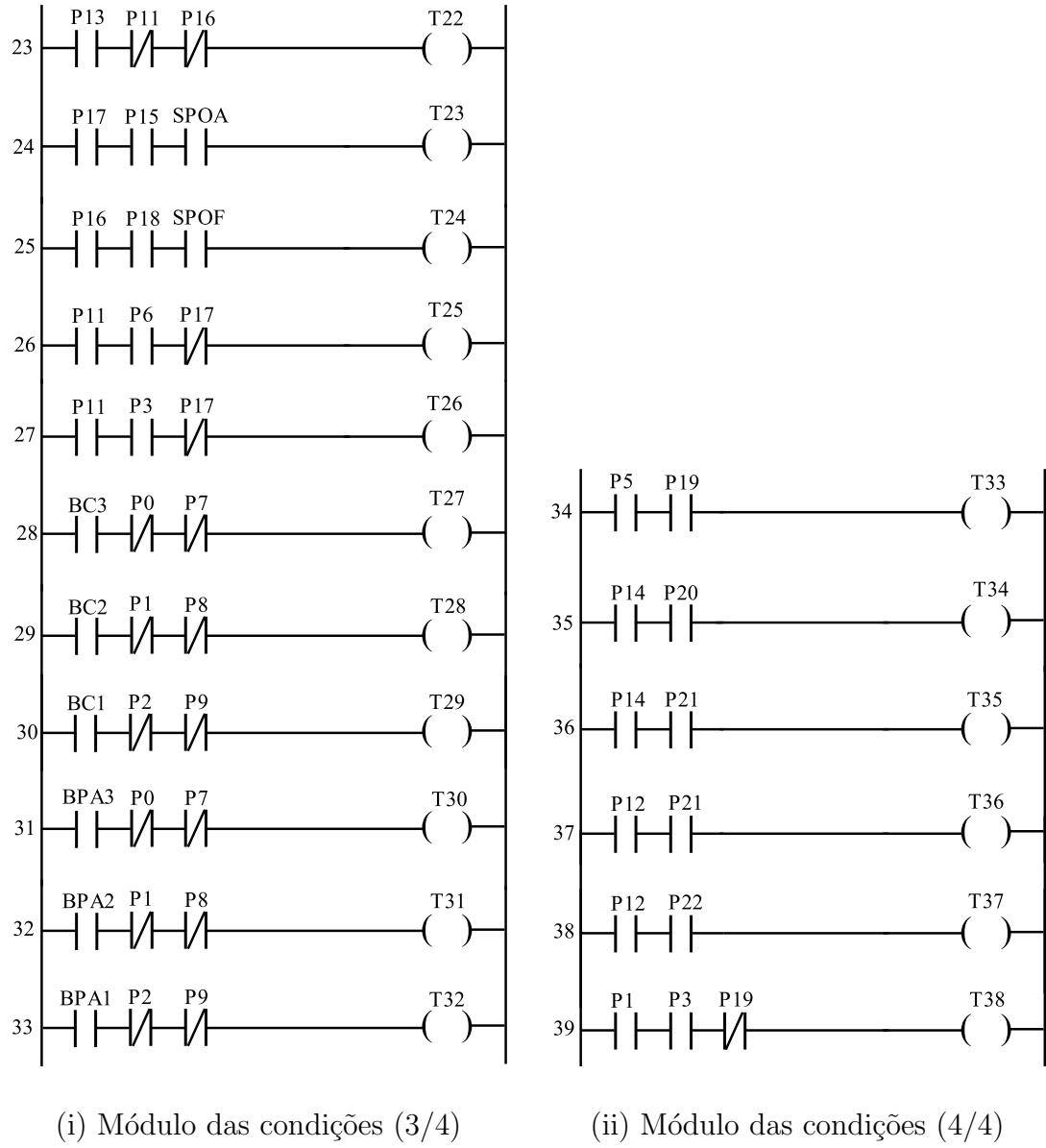


Figura 7.10: Segunda parte do módulo das condições do controlador do elevador.

O módulo da dinâmica é responsável por atualizar a marcação da rede após o disparo das transições. Note que neste módulo é utilizado o temporizador TON visto no capítulo 4. Este temporizador é o responsável por representar o tempo que a cabina do elevador deve permanecer no pavimento, de portas abertas, antes de atender ao próximo chamado.

Apesar de possuir o mesmo tamanho em número de linhas que o módulo das condições, o módulo da dinâmica parece maior. Isto se deve ao fato de que cada disparo de uma transição poder estar relacionado com a mudança de estado de diversos lugares em paralelo.

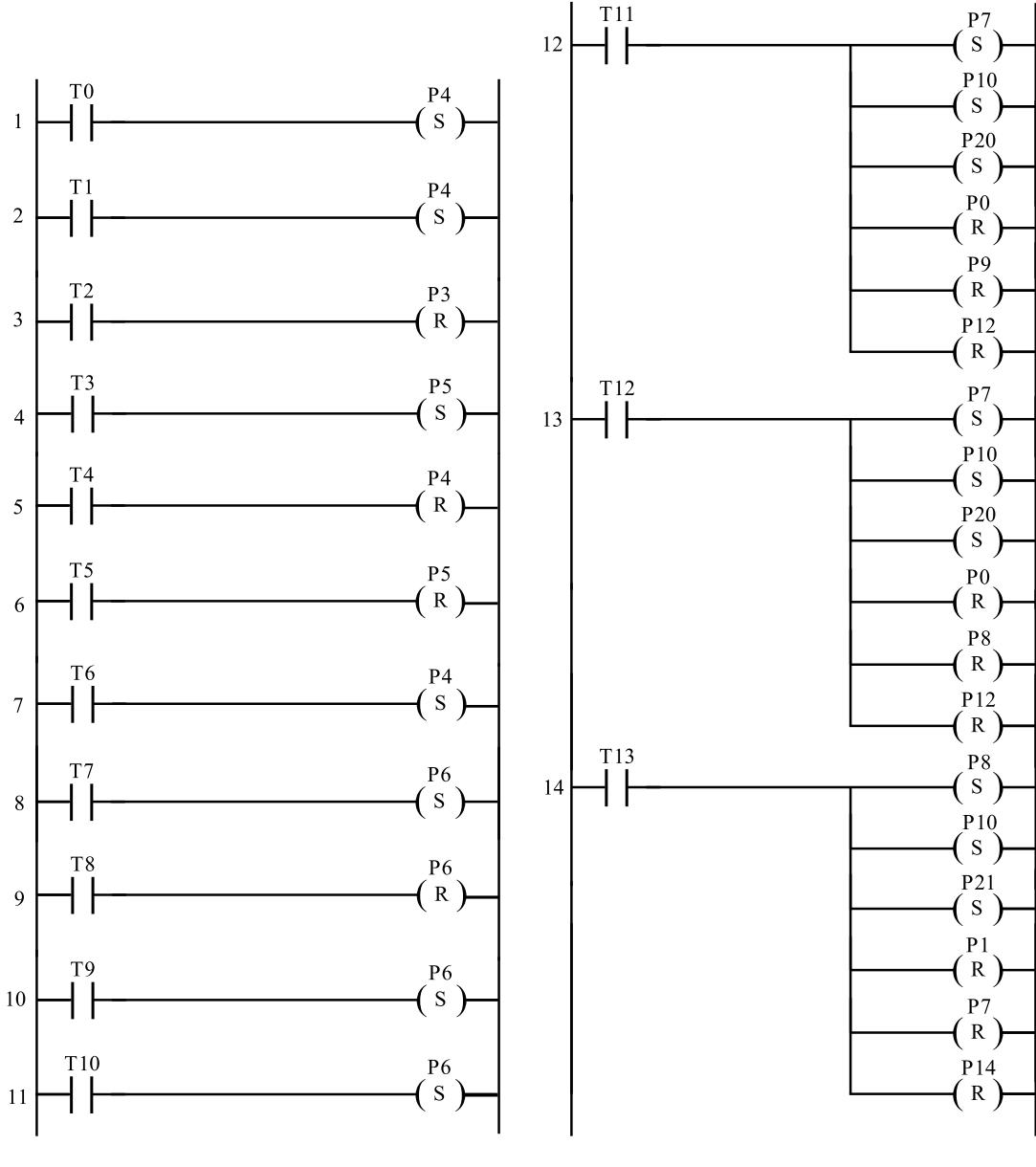
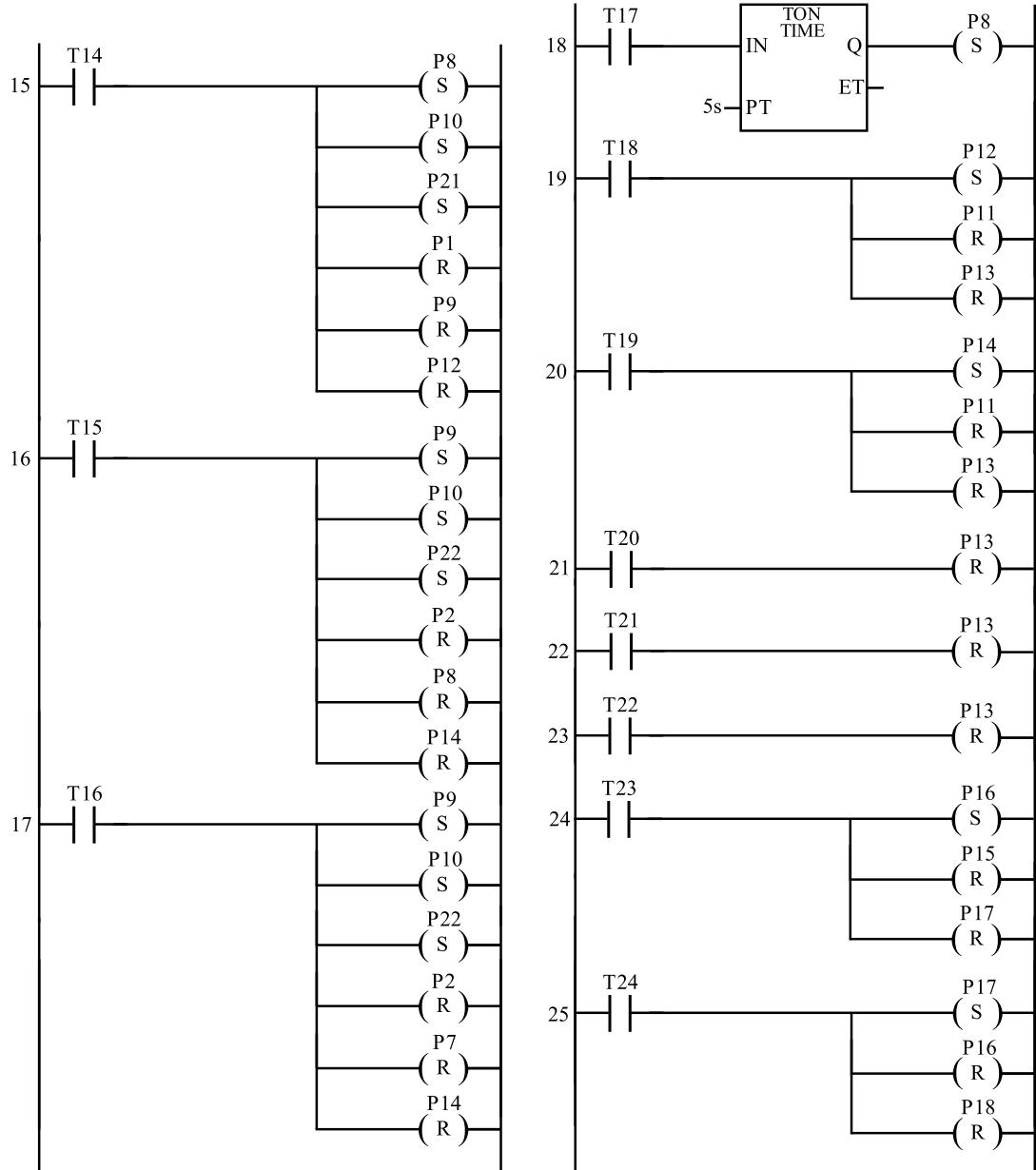


Figura 7.11: Módulo da dinâmica do controlador do elevador.

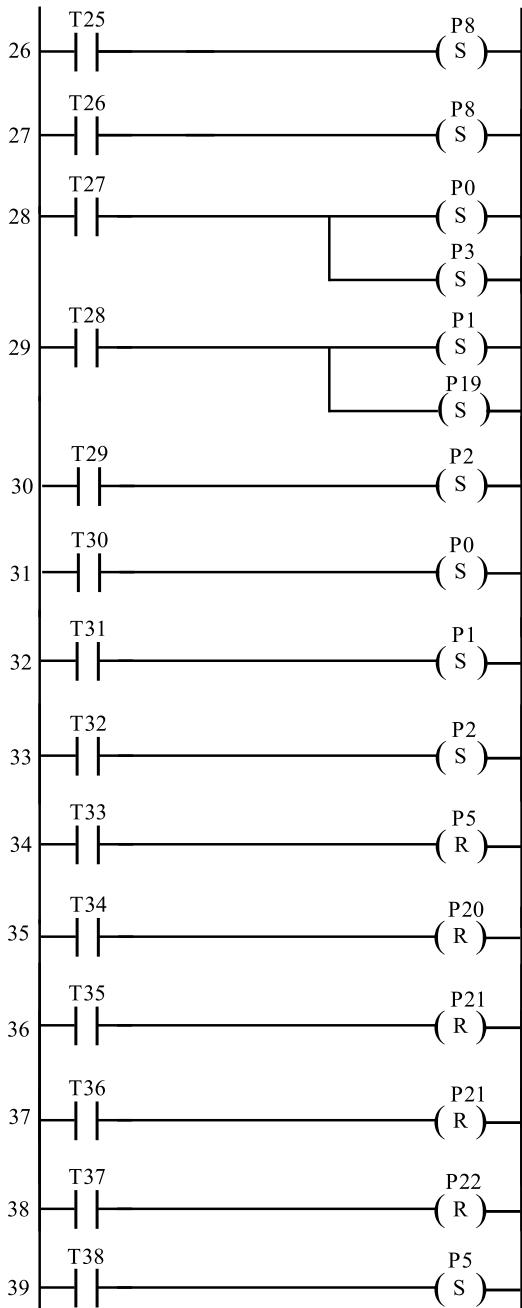


(i) Módulo da dinâmica (3/5)

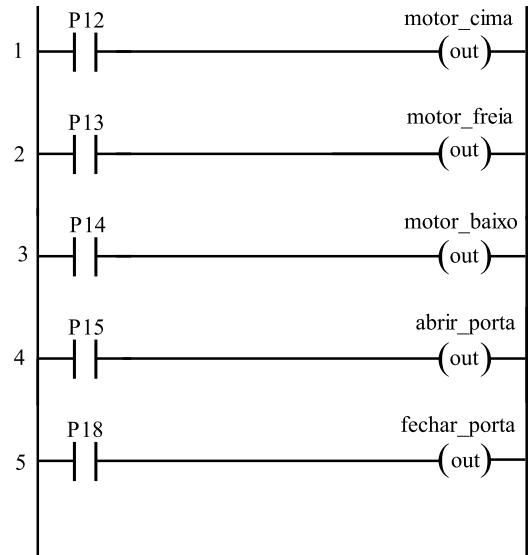
(ii) Módulo da dinâmica (4/5)

Figura 7.12: Módulo da dinâmica do controlador do elevador.

O módulo das ações associa uma bobina de saída do CLP ao lugar que possui uma ação contínua ou instantânea. Desta forma, mediante o disparo de uma determinada transição, as ações associadas aos lugares de saída são executadas. Pode ser visto que a RPIC deste trabalho possui cinco saídas associadas aos lugares  $p_{12}$ ,  $p_{13}$ ,  $p_{14}$ ,  $p_{15}$  e  $p_{18}$ , que comandam o acionamento da máquina de tração e das portas.



(i) Módulo da dinâmica



(ii) Módulo das ações

Figura 7.13: Módulos da dinâmica (i) e das ações (ii) do controlador do elevador.

Como pode ser visto através das figuras 7.8, 7.9, 7.10, 7.11, 7.12 e 7.13, o código *Ladder* resultante tem noventa e cinco linhas. Em linhas gerais, é possível dizer que o código é relativamente pequeno. Observando o diagrama *Ladder* obtido, conclui-se que o código resultante é claro, bem definido e estruturado. Desta forma, a RPIC pode ser obtida diretamente a partir do diagrama *Ladder*, e vice-versa.

# Capítulo 8

## Conclusões

Este trabalho teve como objetivo a elaboração de um sistema de automação para um elevador de passageiros, cujo modelo foi feito utilizando-se redes de Petri interpretadas para controle. Posteriormente, o diagrama *Ladder* do sistema foi obtido para implementação em um CLP.

Com a aplicação prática do capítulo 7, foi obtido o sistema de automação de um elevador de passageiros de três pavimentos, que posteriormente foi simulado no software HPSim v1.1 comprovando estar funcional. A partir da rede de Petri interpretada para controle do sistema de automação do elevador, foi possível aplicar o método MOREIRA *et al.* [6] de conversão de RPIC para *Ladder*, e com relativa facilidade o programa foi obtido, o que pôde demonstrar o grande potencial desta ferramenta.

Entre as vantagens da utilização das redes de Petri pode-se destacar a simplicidade e alto poder de modelagem, a facilidade de simulação, a modularidade e a fácil visualização do modelo, permitindo assim que erros de projeto possam ser detectados antes mesmo de ser posto em operação. Além disso, o programa da forma que é estruturado, permite que se obtenha a rede de Petri completa a partir do código *Ladder*. Um programa *Ladder* bem estruturado, aliado a robustez e flexibilidade dos CLPs, torna a automação de sistemas muito mais simples e confiável.

Como sugestões de trabalhos futuros, pode-se refinar a rede de Petri obtida, acrescentando mais subsistemas como alarmes, indicadores, detectores de carga na

cabina ou até criar uma rotina inteligente que avalie o tráfego de passageiros e selecione a lógica de atendimento mais apropriada. Outro possível trabalho futuro é a implementação de um modelo em escala do elevador de passageiros, para que seja possível avaliar as características do programa na prática. Os controladores possuem aplicações praticamente infinitas, e este trabalho apesar de direcionada a controle de elevadores, pode ser utilizado para outras aplicações na área de automação de sistemas.

# Referências Bibliográficas

- [1] CABRAL, F. G. *Uma Rede de Petri Diagnosticadora para Sistemas a Eventos Discretos Modelados por Autômatos Finitos.* Projeto final de graduação, UFRJ - Escola Politécnica, Rio de Janeiro, RJ - Brasil, 2013.
- [2] BOTELHO, D. S. *Projeto de um Sistema de Automação de uma Célula de Manufatura Utilizando CLP Siemens S7-1200.* Projeto final de graduação, UFRJ - Escola Politécnica, Rio de Janeiro, RJ - Brasil, 2011.
- [3] MONTEZANO, A. F. *Modelo em Rede de Petri de um Sistema de Automação de Elevador de Passageiros.* Projeto final de graduação, UFRJ - Escola Politécnica, Rio de Janeiro, RJ - Brasil, 2011.
- [4] CASSANDRAS, C., LAFORTUNE, S. *Introduction to Discrete Event System.* Secaucus, NJ, Springer-Verlag New York, Inc., 2008.
- [5] DAVID, R., ALLA, H. *Discrete, Continuous and Hybrid Petri Nets.* Springer, 2005.
- [6] MOREIRA, M. V., BOTELHO, D. S., BASILIO, J. C. “Ladder Diagram Implementation of Control Interpreted Petri Nets: a State Equation Approach”. In: *4th IFAC Workshop on Discrete-Event System Design*, pp. 85–90, Gandia Beach, Spain, 2009.
- [7] FRANCHI, C. M., CAMARGO, V. L. A. *Controladores Lógicos Programáveis - Sistemas Discretos.* 1 ed. São Paulo, Editora Érica, 2008.
- [8] ISO/IEC. *International standard IEC 61131-3.* ISO/IEC, 2001.
- [9] STEPHAN, R. M. *Acionamento, Comando e Controle de Máquinas Elétricas.* Editora Ciência Moderna, 2013.
- [10] ABNT/CB-04. *Norma NBR NM 207, Elevadores elétricos de passageiros - Requisitos de segurança para construção e instalação.* ABNT, 1999.

- [11] SCHINDLER. *Manual de Transporte Vertical em Edifícios; Elevadores de Passageiros, Escadas Rolantes, Obra Civil e Cálculo de Tráfego*. Elevadores Atlas Schindler, 2008.
- [12] HARRIS, T. Disponível em: <http://science.howstuffworks.com/transport/engines-equipment/elevator1.htm>, Acesso em: 26 Mar. 2013.