



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS
DAS5203 - MODELAGEM E CONTROLE DE SISTEMAS A EVENTOS DISCRETOS

RELATÓRIO - TRABALHO 1

Artur Klann Schmitt
Eduardo Rosa de Lima
Maria Clara Castro Valle
Pedro Henrique Melo

Florianópolis, 2023

1. INTRODUÇÃO

O presente relatório tem como objetivo apresentar a estação de manufatura *Testing*, que faz parte do Sistema Modular de Produção (MPS) da FESTO, disponível no laboratório LAI do Departamento de Automação e Sistemas da UFSC. Além de descrevê-la detalhadamente, também será descrita a modelagem implementada por meio de Redes de Petri Rotuladas Estendidas (RPRE) e, posteriormente, uma proposta de controle para o sistema com Redes de Petri Interpretadas para Controle (RPIC).

Para a implementação no Controlador Lógico Programável (CLP) da bancada, um método de conversão para a linguagem *Ladder* foi utilizado e a programação foi realizada no *TIA Portal*, software de automação da Siemens.

2. APRESENTAÇÃO DO SISTEMA

A bancada *Testing* tem como objetivo central a conferência da altura da peça recebida da bancada *Distributing*. Apenas peças da altura correta devem continuar na planta, de forma que aquelas que são mais altas ou mais baixas que a especificação são descartadas para um *buffer*. Como no laboratório estão à disposição dois tipos de peças, é dito que as mais baixas são as desejadas e as mais altas descartadas.

Com o recebimento de uma peça da primeira bancada - *Distributing* -, o elevador inicia sua operação e sobe. No final de seu curso, então, a altura da peça é medida através do deslocamento de um módulo comparador. A partir disso, o sistema decide entre mandar a peça para a próxima bancada ou descartá-la. Caso a peça seja da altura adequada, a bancada espera o sinal de confirmação da bancada seguinte (*Separating*) e liga o ar da rampa, seguido da ativação do atuador que empurra a peça ao próximo módulo e da volta do elevador a sua posição inicial. Caso a peça seja alta demais, o atuador faz o elevador descender e em seguida a peça é empurrada para o *buffer* de descarte. Nesse ponto, a bancada está apta para o recebimento de novas peças.

O funcionamento descrito está atrelado à presença de 8 sensores e 5 atuadores na bancada. Quanto aos atuadores, dois estão relacionados ao movimento vertical do elevador (Desce Linear e Sobe Linear), um é referente ao ativamento do ar da rampa (Ar Rampa), um para acionar o cilindro que empurra a peça para o descarte ou para a rampa (Empurra peça) e o último para liberar a bancada anterior (IP N FO). Quanto aos sensores, um sensor óptico é utilizado para detectar a presença de peça no início (PART AV), dois são sensores de fim de curso para saber posição do elevador (1B1 e 1B2), um indica a ação do cilindro ejeter (2B1), outro indica o tamanho da peça (B5), um recebe o sinal de liberação da próxima bancada (IP FI) e um que impede que o elevador suba caso detectado algum objeto (B4). O oitavo sensor presente na bancada é um sensor óptico para detecção de cor (B2). Esse, porém, não será utilizado já que a última estação *Sorting* realiza a separação colorimétrica.

Abaixo, na Figura 1, encontra-se a tabela com os sensores e atuadores da bancada, seguidos de seus respectivos nomes e descrições. Observação: o sensor “B2” não é utilizado e alguns endereços não estão necessariamente corretos.

Figura 1 - Sensores e Atuadores

Estação: Testing 150.168.14.24		
Endereço	Descrição	Observação
I0.0	PART_AV	Modulo de Verificação (Peça Disponível)
I0.1	B2	Sensor Óptico (Classificação por Cor)
I0.2	B4	Sensor Retroreflexivo (Segurança para o módulo elevador)
I0.3	B5	Módulo Comparador (Seleciona por tamanho)
I0.4	1B1	Atuador Linear de Dupla Ação // Deslocamento do Elevador (Cilindro levantado)
I0.5	1B2	Atuador Linear de Dupla Ação // Deslocamento do Elevador (Cilindro abaixado)
I0.6	2B1	Atuador Linear de Dupla Ação // Expulsão das Peças (Cilindro Ejeta)
I0.7	IP FI	Sensor Óptico Transmissor
I2.0	Start	Botão
I2.1	Stop	Botão
I2.2	Manual Mode	
I2.3	Reset	Botão
Q0.0	Desce Linear	Atuadores Lineares
Q0.1	Sobe Linear	Atuadores Lineares
Q0.2	Empurra Peça	Ejeta
Q0.3	Ar Rampa	Válvula Reguladora de Fluxo para Controle de Vazão do Ar na Rampa
Q0.7	IP N FO	Sensor Óptico Receptor
Q2.0	Start	LED
Q2.1	Reset	LED
Q2.2	Reseting	Luz Q1
Q2.3	Running	Luz Q2

3. MODELAGEM DO SISTEMA EM MALHA FECHADA USANDO RPRE

3.1. Modelagem modular

Para iniciar a modelagem do sistema em malha fechada, a abordagem utilizada pelo grupo consistiu em dividir a bancada *Testing* em três módulos, que foram modelados individualmente, a fim de simplificar o processo. Dessa forma, a modelagem do elevador, da rampa e do buffer pode ser vista nas Figuras 2, 3 e 4.

É interessante destacar que o grupo optou por modelar o *buffer* da bancada como se possuísse capacidade infinita, em virtude da falta de um sensor que indica quando está cheio. Dessa forma, o lugar “Peças Descartadas” visto na Figura Z, é o único lugar não seguro da RPRE.

Figura 2 - Modelagem do elevador

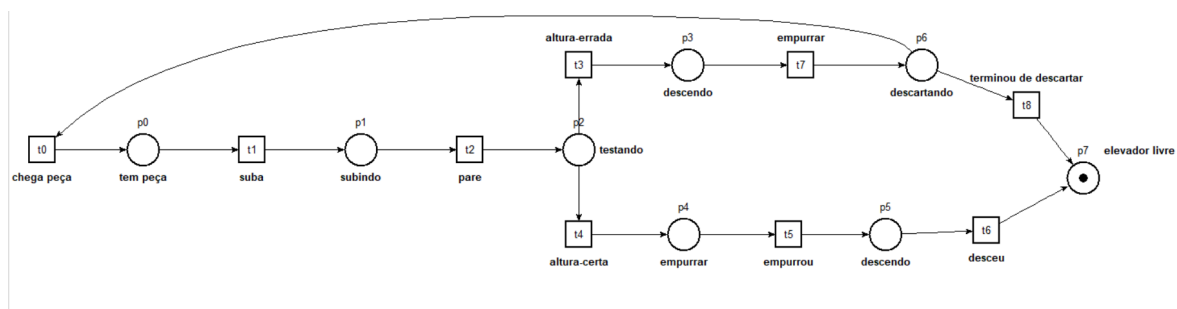


Figura 3 - Modelagem da rampa

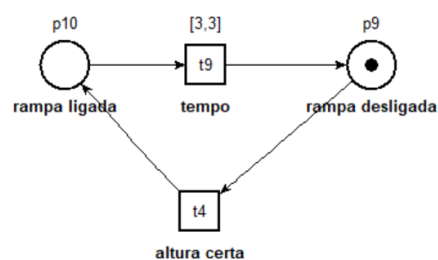
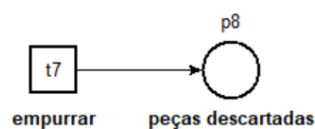


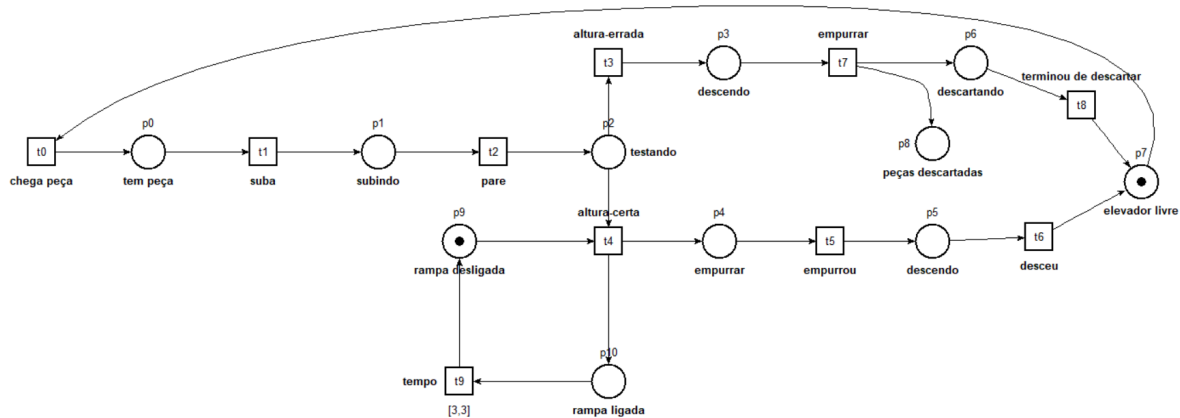
Figura 4 - Modelagem do buffer



3.2. Resultado da composição

Em seguida, por fusão de transições, os módulos foram agrupados, formando a RPRE vista na Figura 5.

Figura 5 - RPRE da bancada *Testing*

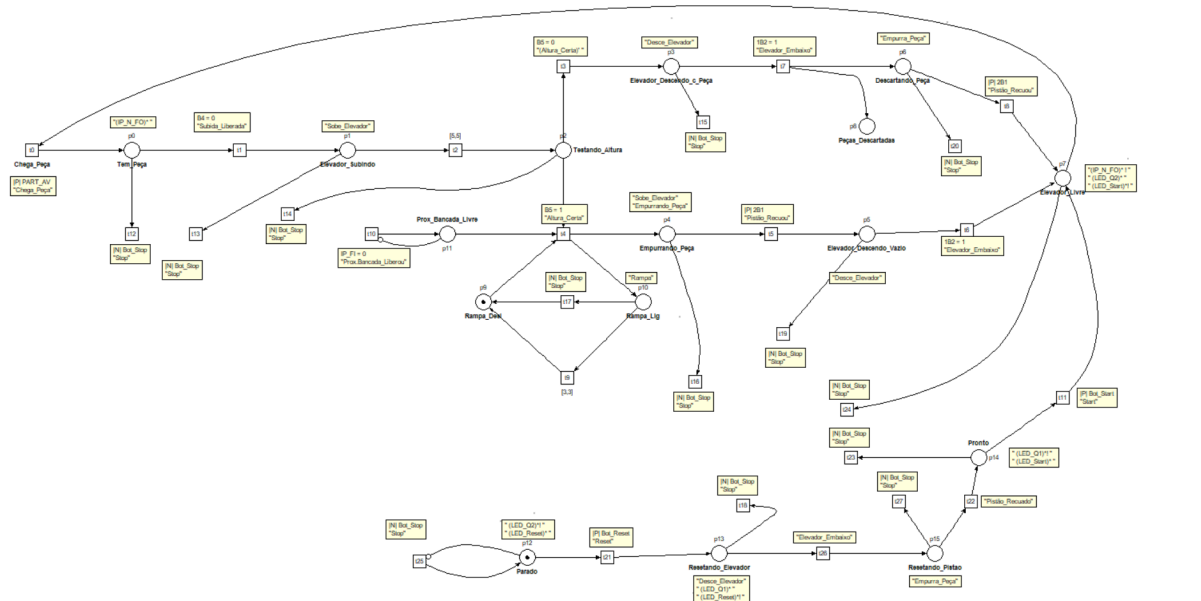


4. PROPOSTA DE CONTROLE DO SISTEMA

4.1. RPIC

A RPIC construída no *software* Tina pode ser observada na Figura 6. Comentários em Tags foram colocados em transições e lugares, para associar os sensores e ações dos atuadores, respectivamente. Uma transição temporizada está indicada pelo intervalo de acionamento, em segundos. A notação usada para a borda de subida e descida dos sensores foi “|P|” e “|N|”, respectivamente. Algumas ações, por sua vez, foram modeladas para serem do tipo contínua, indicadas apenas pelo nome dado ao atuador, enquanto que outras do tipo impulsional são identificadas por um * no lugar onde são *setadas* e *! onde são *resetadas*.

Figura 6 - RPIC para a bancada *Testing*

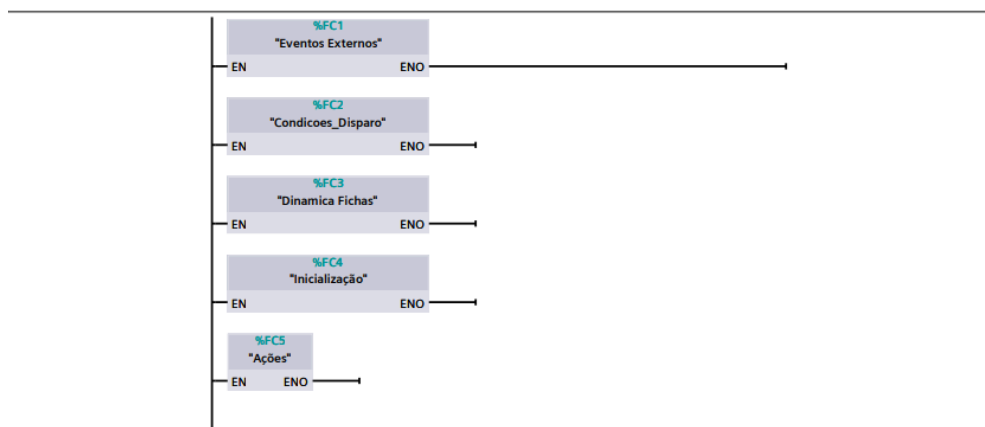


Uma imagem da RPIC em melhor definição está anexada ao fim do relatório, e o arquivo .nd (TINA) foi entregue em conjunto com o relatório.

4.2. Método de conversão de RPIC para Ladder

O método utilizado para converter a RPIC construída para Ladder, a fim de implementá-la no CLP, foi o proposto no artigo “Bridging the Gap Between Design and Implementation of Discrete-Event Controllers” (Moreira, Basilio, 2014). Como sugerido, a conversão é dividida em cinco módulos, e cada um deles foi construído em um bloco de função isolado, que posteriormente foram chamados no programa principal, como pode ser visto na Figura 7. Cada módulo será brevemente explicado e exemplificado a seguir, com alguns trechos do código desenvolvido, o qual pode ser visualizado integralmente no Anexo 1.

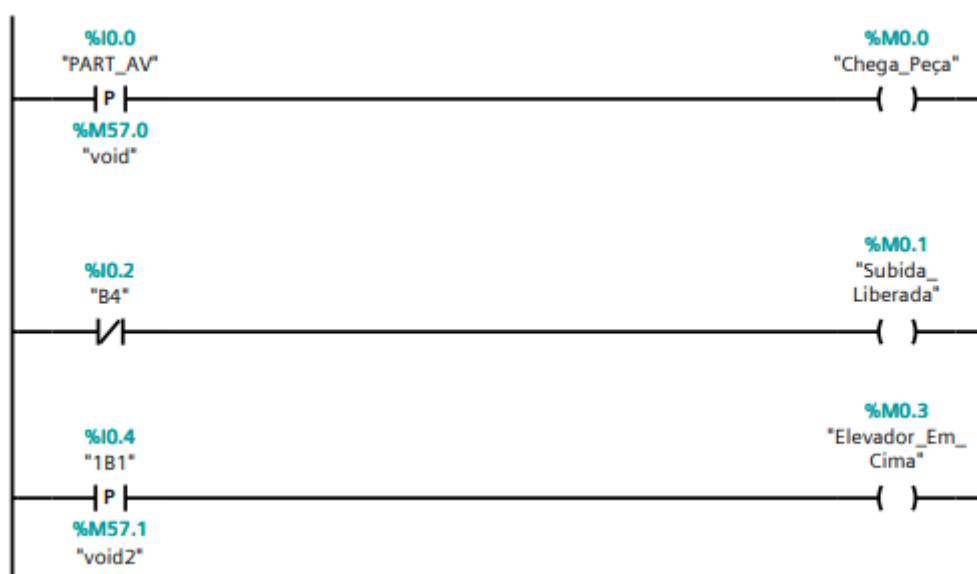
Figura 7 - Programa Principal do Ladder



4.2.1. Módulo 1: ocorrência de eventos externos

No primeiro módulo, todos os botões do painel de interface e os sensores da bancada são relacionados de alguma maneira - borda de subida (P), descida (N) ou diretamente - a um endereço de memória.

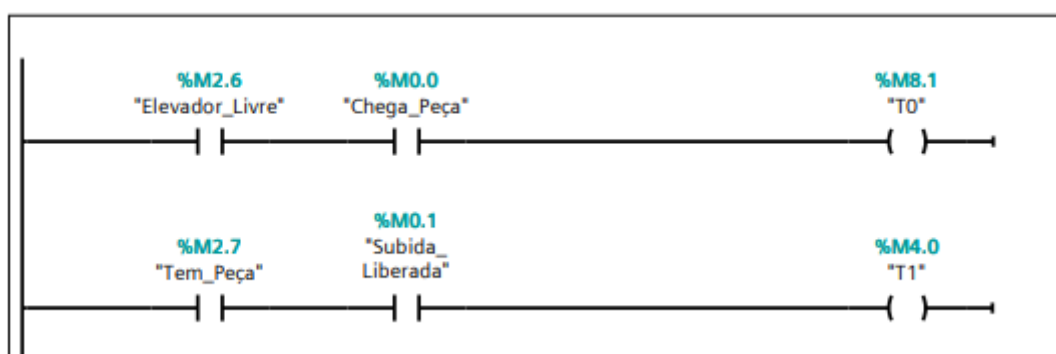
Figura 8 - Parte do código do Módulo 1



4.2.2. Módulo 2: condições para o disparo das transições

Neste módulo, cada transição é associada aos lugares e às variáveis de memória dos sensores que podem habilitá-las.

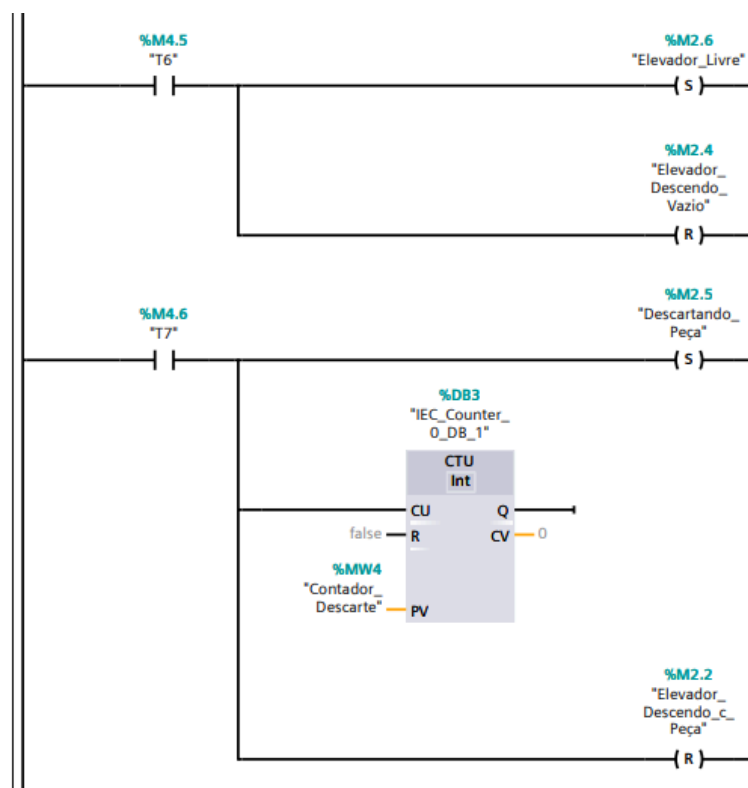
Figura 9 - Parte do código do Módulo 2



4.2.3. Módulo 3: evolução das fichas (dinâmica) na Rede de Petri

No terceiro módulo, é feita a dinâmica das fichas. Por ser uma RPIC segura - ou seja, cada lugar estar limitado ao recebimento de apenas uma ficha -, optou-se por utilizar bobinas do tipo *Set-Reset* para indicar a colocação ou retirada de uma ficha de um lugar. A única exceção é o lugar “Peças_Descartadas”, associado a um contador.

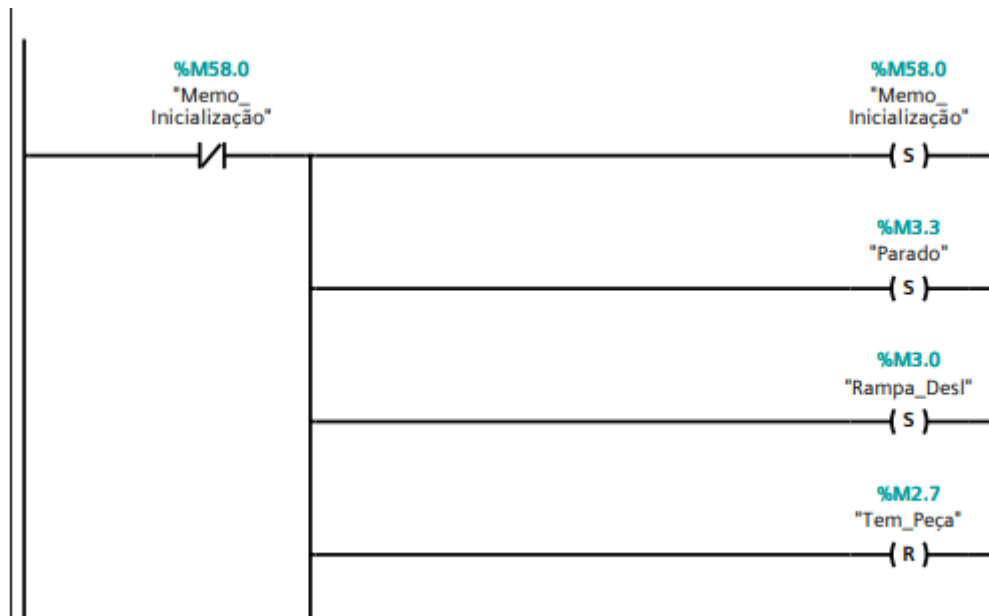
Figura 10 - Parte do código do Módulo 3



4.2.4. Módulo 4: inicialização (marcação inicial) da Rede de Petri

Neste módulo, a presença ou não de fichas em cada lugar quando o sistema for iniciado é definida. Para o lugar Peças_Descartadas, o contador é zerado. Uma memória de inicialização é utilizada para é utilizada para garantir que essa parte do código só seja executada uma única vez.

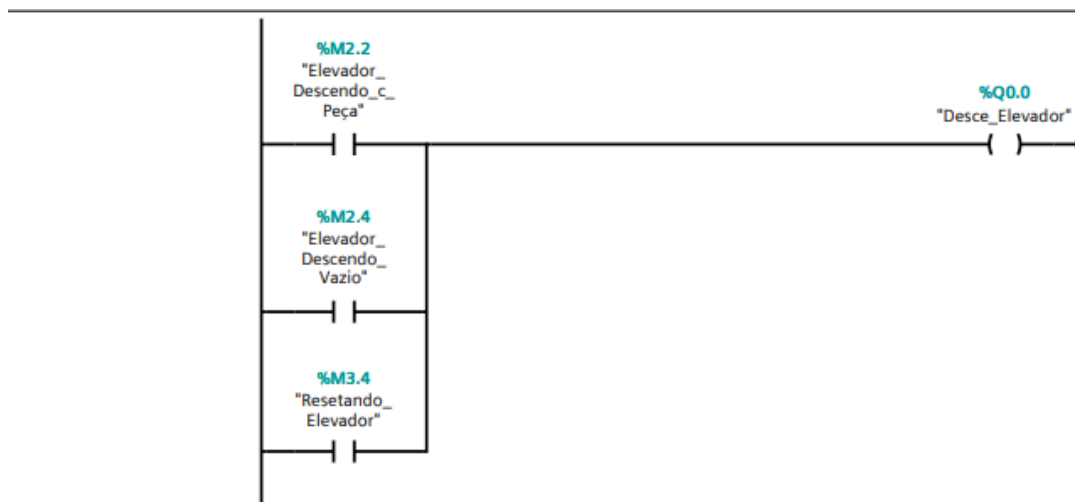
Figura 11 - Parte do código do Módulo 4



4.2.5. Módulo 5: ações em cada estado

Por fim, o último módulo associa os lugares às ações dos atuadores da bancada e também das luzes do painel de operação.

Figura 12 - Parte do código do Módulo 5



5. ANÁLISE DE PROPRIEDADES

Em teste utilizando as funções de simulação do *software* TINA, usado para a implementação de ambas as redes mencionadas neste artigo, averiguamos algumas das propriedades das redes modeladas. No entanto, o software não teve o resultado esperado, uma vez que ele não realiza sincronização dos eventos, isto é, duas transições habilitadas e vinculadas ao mesmo evento real podem disparar de maneira independente. Por conta disso, as transições vinculadas ao botão stop, que retiram ficha de todos os lugares da Rede de Petri, disparam independentemente, isto é, podendo causar parada da bancada sem saída das fichas, uma vez que no TINA o disparo da transição que leva para “Parado” e o disparo das transições que retiram ficha dos lugares correspondentes ao estado atual são independentes. Assim, a própria árvore de cobertura do software mostra possibilidade de cobertura (ω) em todos os lugares.

No entanto, analisando os resultados testados na bancada, bem como questionando criticamente o trabalho realizado pelo grupo, podemos concluir que nossas redes - tanto RPRE quanto RPIC - são livres de bloqueio do tipo *deadlock*, de conflitos e também que ambas são limitadas, condizendo com as propriedades do sistema linear da bancada modelada. A única exceção é o lugar da RPIC que conta a quantidade de peças descartadas - este, por ser um lugar com finalidade de armazenar um valor, não tem nenhuma transição de saída, apenas de entrada, e é não-seguro e ilimitado, para contar corretamente a quantidade de peças. Mesmo assim, vemos que esta característica não afeta de forma alguma o funcionamento do sistema, sendo inofensivo para o processamento da rede uma vez que todo o resto dela forma uma componente conservativa limitada (segura), reversível e livre de bloqueios. Ademais, pode-se afirmar que suas transições são vivas.

6. CONCLUSÃO

A partir de todo o trabalho desenvolvido, conclui-se que o grupo conseguiu compreender o funcionamento de toda a estação, incluindo o funcionamento dos sensores, atuadores e módulos e, a partir disso, modelá-los utilizando RPRE e RPIC. Ao fazer a conversão da RPIC para Ladder e as testagens na estação começarem a ser realizadas, algumas inconsistências no modelo de controle proposto inicialmente foram observadas, principalmente em função de alguns sensores operarem de maneira distinta à que o grupo esperava. Dessa forma, procurou-se modificar a RPIC inicial - sempre levando em conta a manutenção de propriedades importantes como vivacidade, limitação e conservatividade - e, conseqüentemente, o código em Ladder, até que toda operação da bancada se mostrasse satisfatória. O resultado foi a RPIC apresentada no presente relatório e a programação anexada como relatório extraído do *TIA Portal*, que foi comprovadamente eficiente para cumprir com os objetivos da estação *Testing* e garantir a continuidade efetiva do processo de produção.

É interessante ressaltar, também, que quesitos de segurança, eficiência e aplicabilidade dentro do contexto industrial foram tidos como prioridade ao tentar desenvolver toda a lógica da rotina de operação e reinicialização do sistema.

Finalmente, é de consenso do grupo que este trabalho foi extremamente importante para concluir o objetivo da primeira metade da disciplina: aprender a utilizar Redes de Petri como ferramenta para modelagem de Sistemas a Eventos Discretos e sua aplicabilidade dentro da programação de sistemas fabris reais.

7. ANEXO 1

Nas páginas a seguir, encontra-se o diagrama completo da programação em *Ladder* do controlador, exportado do *TIA Portal*, junto das outras informações técnicas do projeto e da Rede de Petri total exportada do TINA.

