



UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

TRABALHO DE FORMATURA

IMPLEMENTAÇÃO DE SISTEMA DE
CONTROLE E SUPERVISÃO NA CÉLULA DE
MISTURA DO CIS-POLI.

DOCUMENTAÇÃO FINAL

VOLUME I

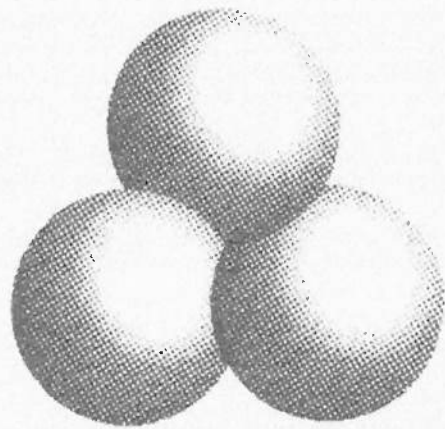
Daniel de Almeida Okino
Gustavo Soares de Arruda Borges

372472
1027905

Orientador: Prof. Dr. Marcos R. Pereira Barretto

São Paulo, 15 de Dezembro de 1998.

9.7 (more excel)
Wamy



CIS
Poli

Centro de Integração de Sistemas

Dedicamos esse trabalho

ao colega Paulo Cesar Debenest, pelo seu companheirismo e amizade
e ao Prof. Dr. Marcos R. P. Barretto, por sua orientação.

Agradecemos a:

Anfang Automação Industrial

Unisoft Sistemas Ltda.

Rockwell Automation do Brasil Ltda.

pelo apoio ao projeto.

RESUMO

Utilizando-se o sistema físico (hardware) instalado no CIS-Poli, implantou-se um sistema flexível de controle para a célula de mistura e instalou-se um supervísório (SCADA) para monitorar os processos. Para tanto, realizou-se um estudo sobre a arquitetura interna de um PLC e sua programação (Ladder), e conceitos estruturais de um sistema supervísório. Após esta etapa de estudos, utilizou-se a metodologia PFS/MFG para estruturar a solução e implantar o sistema de controle e supervisão da célula de mistura. Ao final da implementação houve uma etapa de testes para validação e aperfeiçoamento do sistema.

ÍNDICE

Introdução	1
Parte I - Conceitos	3
1. Controladores Lógicos Programáveis (CLP)	3
1.1 A Evolução dos CLPs	3
1.2 Arquitetura Interna dos CLPs	5
1.3 Arquiteturas de Entrada e Saída Remota	8
1.4 Especificação do CLP	9
1.5 Linguagens de Programação	11
1.5.1 Álgebra Booleana	12
1.5.2 IL (Instruction List)	12
1.5.3 ST (Structured Text)	13
1.5.4 LD (Ladder Diagram)	13
1.5.5 FBD (Function Block Diagram)	14
1.5.6 Fluxograma	15
1.5.7 SFC (Sequential Flow Chart)	16
1.5.8 Decision Table	17
1.6 O CLP-5/15-B	18
2. Sistemas SCADA	21
2.1 Arquitetura dos sistemas SCADA	21
2.2 O SCADA UniSoft	24
2.2.1 Introdução	24
2.2.2 Banco de dados	28
2.2.3 Telas	31
2.2.4 Alarmes e tendências	33
2.2.5 Ambiente	41
2.2.6 Características diferenciadoras	44
2.2.7 Comercialização	48
3. A Metodologia de Implantação	50
3.1 Modelo GRAI para CIM	50
3.1.1 Sistema físico	51
3.1.2 Sistema de controle	51
3.1.3 Proposta de Estruturação Hierárquica dos centros de decisão	53
3.1.4 Estrutura de um Centro de Decisão	53
3.2 Metodologia PFS/MFG	55
3.2.1 Production Flow Schema (PFS)	56
3.2.2 Mark Flow Graph (MFG)	57
3.3 Metodologia de Conversão MFG-Ladder	60

Parte II - Estruturação do Projeto	61
4. Objetivos do Projeto	61
5. Instalações e Equipamentos	62
6. Dispositivos de Controle	64
7. Definição dos Níveis de Controle	71
8. Estruturação do Sistema de Controle	72
Parte III – Implantação do Projeto	75
9. Projeto de Software	75
9.1 Sistema de Controle do Nível Estação de Trabalho (CLP)	76
9.1.1 Mapas de memória do CLP	77
9.1.2 Funções e Rotinas implementadas	93
9.1.1 Rotina Principal	94
9.1.2 Função Inicializa	101
9.1.3 Função Le_Volume	104
9.1.4 Função TB1_TM1	106
9.1.5 Função TB1_TM2	110
9.1.6 Função TB2_TM1	114
9.1.7 Função TB2_TM2	118
9.1.8 Função TM1_TE1	122
9.1.9 Função TM1_TE2	126
9.1.10 Função TM1_TE3	130
9.1.11 Função TM2_TE1	134
9.1.12 Função TM2_TE2	138
9.1.13 Função TM2_TE3	142
9.1.14 Função MIX_TM1	146
9.1.15 Função MIX_TM2	149
9.1.16 Função TE1_OUT	152
9.1.17 Função TE2_OUT	156
9.1.18 Função TE3_OUT	160
9.1.19 Função COR1_TM1	164
9.1.20 Função COR1_TM2	168
9.1.21 Função COR2_TM1	172
9.1.22 Função COR2_TM2	176
9.1.23 Função COR3_TM1	181
9.1.24 Função COR3_TM2	186
9.1.25 Função TM1_OUT	190
9.1.26 Função TM2_OUT	194
9.1.27 Função WAT_TM1	198
9.1.28 Função WAT_TM2	202
9.1.29 Função WAT_TE1	206
9.1.30 Função WAT_TE2	210
9.1.31 Função WAT_TE3	214
9.1.32 Função LE_REC1	218
9.1.33 Função LE_REC2	245

9.2	Sistema de Controle do Nível Célula (SCADA)	263
9.2.1	Criação do Banco de Dados	263
9.2.2	Comunicação SCADA - CLP	265
9.2.3	Armazenamento de Receitas (Salvar/Recuperar)	271
9.2.4	Sinótico	272
9.2.5	Receita	273
10.	Testes	275
Parte IV – Considerações Finais		276
Bibliografia		278

INTRODUÇÃO

O Centro de Integração de Sistemas (CIS-POLI/AC) é o resultado de uma iniciativa conjunta da Escola Politécnica da Universidade de São Paulo (EPUSP) e da Andersen Consulting, com a participação de diversos fornecedores nacionais, tendo como objetivo implementar um ambiente de pesquisa e desenvolvimento na área de Manufatura Integrada por Computador (C.I.M.), treinamento e especialização de profissionais nas tecnologias envolvidas e integração de sistemas computacionais de diversos fabricantes de hardware e software.

A estrutura organizacional do CIS-POLI/AC foi elaborada tomando como base uma empresa hipotética, fabricante de detergentes e amaciantes de forma a viabilizar um dos objetivos do CIS-POLI/AC, para mostrar a flexibilidade de produção, ou seja, a possibilidade de alteração dos produtos, tanto em formulação quanto em quantidades e tamanhos.

Fisicamente o CIS-POLI/AC é uma planta piloto de 300 m² para produção de detergentes e amaciantes, localizada no Prédio da Engenharia Mecânica da EPUSP. O processo de fabricação pode ser dividido em três etapas principais:

- Reação para obtenção de matérias primas;
- Mistura da base de detergente / amaciante com corantes e aromatizantes;
- Envasamento dos produtos finais.



A área de Reação é responsável pelo preparo de matérias primas. Compõe-se basicamente de dois reatores, nos quais as matérias ativas reagem, segundo quantidades e

perfis de temperatura controlados e adequados ao produto que se deseja obter. Os produtos são as matérias primas para fabricação de detergentes/amaciantes aqui denominados base de detergente/amaciante, respectivamente. As variáveis de processo controladas são temperatura, pressão, balanceamento em massa das matérias primas e vazão de saída do tanque de reação.

Na área de Mistura as bases de detergente/amaciante são diluídas em água e misturadas com corantes e aromatizantes através de um processo em lotes (“batch”). Compõe-se basicamente de dois tanques para estocagem de base, dois tanques misturadores, três tanques para armazenamento de corantes e três tanques de estocagem de produtos acabados. As diversas combinações de base e corante geram seis produtos acabados distintos nesta etapa de produção. Controla-se nesta etapa as quantidades de base, água e corante misturadas entre si.

Na área de Envasamento os produtos acabados são envasados em frascos de 250, 500 e 1000 ml, resultando então em dezoito produtos diferentes que as instalações podem produzir. As variáveis controladas estão relacionadas com a coordenação das estações robotizadas de envasamento.

PARTE I - CONCEITOS

1. CONTROLADORES LÓGICOS PROGRAMÁVEIS (CLP)

Por Controlador Lógico Programável (CLP ou PLC - Programmable Logical Controller) se entende os equipamentos baseados em microprocessadores que se destinam ao controle de sistemas discretos. Atualmente os CLPs apresentam características extras como: facilidade de comunicação de dados (com computadores e periféricos), grande variedade de módulos para aplicações específicas (controle de motores, visão, controle PID) e grande capacidade de memória/processamento. Os itens abaixo apresentam uma descrição resumida dos CLPs visando identificar suas funcionalidades. O objetivo é fornecer subsídios para a compreensão do projeto como um todo. No final do capítulo descreve-se o CLP-5-15/B, o controlador da Allen-Bradley utilizado na célula de mistura do CIS-Poli.

1.1 A Evolução dos CLPs

Os Controladores Lógicos Programáveis (CLPs, ou Programmable Logic Controllers - PLC) surgiram da necessidade de controle de Sistemas a Eventos Discretos (SEDs) através de um dispositivo mais compacto, flexível e de fácil manutenção em relação aos painéis de relês eletromagnéticos anteriormente usados nesse controle. Pode-se dizer que a história dos CLPs teve início em 1968, quando a GM divulgou a especificação técnica necessária para um equipamento que necessitava, que posteriormente se seriam chamados de CLPs.

A especificação técnica da GM para os primeiros CLPs foi a seguinte:

- fácil de ser programado;
- fácil manutenção (plug-in);
- alta confiabilidade;
- menor que os painéis a relés;
- comunicação de dados com computadores;
- preço competitivo em relação aos relés e/ou eletrônicos;
- sinal de entrada de 115Vca;
- sinal de saída de 115Vca@2A mínimo;
- modular para pequenas e grandes aplicações.

No final de 1969 apareceram os primeiros produtos comerciais atendendo a especificação da GM, os primeiros CLPs.

No final da década de 70 surgem os CLPs baseados em microprocessadores de 8 bits.

Em meados da década de 80 aparecem os CLPs baseados em microprocessadores de 16 bits. Os CLPs transformam-se em CPs (Controladores Programáveis), passando a fazer também controle analógico. Nessa época, aparecem os sistemas SCADA (Supervisory Control and Data Acquisition).

No final da década de 80 aparecem os CLPs baseados em microprocessadores de 32 bit. Surgem os CLPs de baixo custo.

Em meados da década de 90 surgem os CLPs de baixíssimo custo. Surgem também os CLPs com funcionalidades estendidas, como facilidade de comunicação de dados (através de redes locais, Internet, Field-Bus, ...), módulos para aplicações específicas (acionamento de motores, código de barras, visão, ...), capacidade de memória e processamento maior, etc.

1.2 Arquitetura Interna dos CLPs

A especificação de modularização dada pela GM em 1968 sobrevive até hoje, podendo ser observado que os módulos de I/O dos CLPs são separados (fisicamente) da unidade de processamento. Na figura a seguir se observa uma esquematização da arquitetura de hardware do CLP.

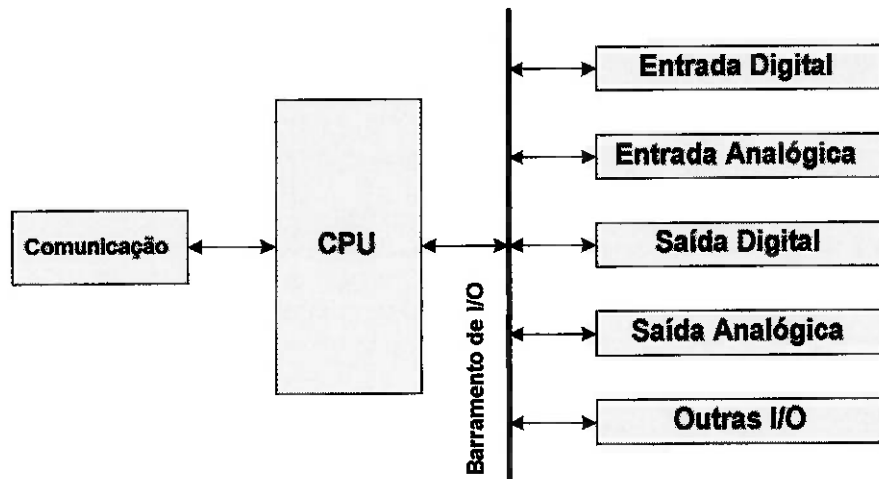


Figura 1: Arquitetura de Hardware de um CLP.

Além desses módulos de I/O, encontra-se disponíveis no mercado uma infinidade de módulos para aplicações específicas. Dentre eles, podemos destacar os seguintes módulos:

- contador de pulsos (para encoder/resolver);
- PID;
- dispositivos de entrada: teclado, botoeira, etc;
- dispositivos de saída: monitores, painéis de luzes, etc;
- controladores de velocidade (em especial para controle de motores);
- controladores fuzzy;
- periféricos de computador: impressoras, floppy, etc).

A memória presente na CPU do CLP pode ser dividida em duas partes: a Memória de Programa e a Memória de Dados. A memória de programa armazena das instruções de funcionamento do CLP, o programa que norteia seu funcionamento. A memória de dados armazena tanto as variáveis utilizadas no programa quanto o mapeamento virtual (imagem) dos dispositivos de entrada e saída. A Figura 2 apresenta a divisão da memória do CLP.

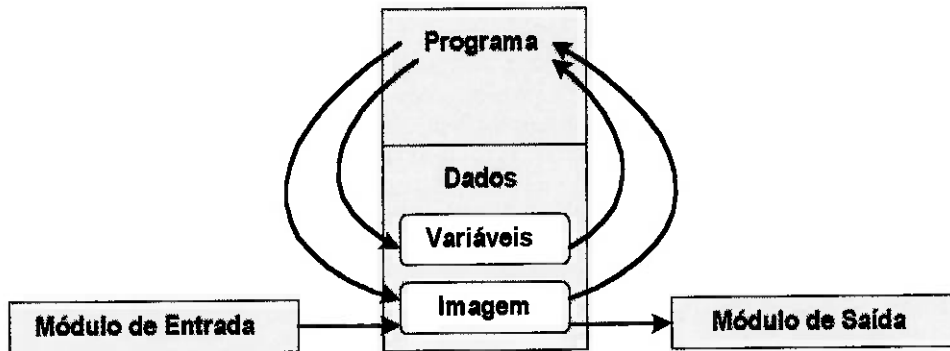


Figura 2: Estruturação da memória em um CLP.

O ciclo de processamento do CLP pode ser dividido em três etapas: a aquisição das entradas, o processamento e a atualização das saídas. Verifica-se, portanto, que o tempo de ciclo de processamento não dependerá somente da velocidade da CPU.

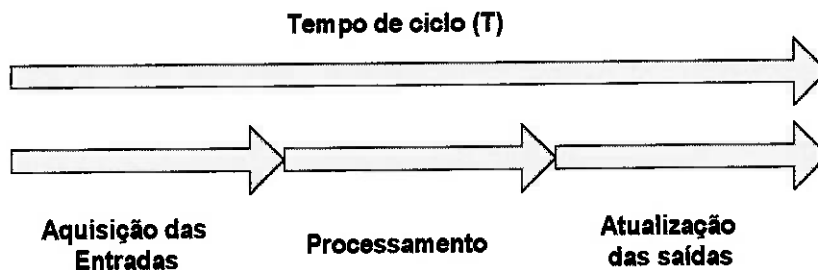


Figura 3: Tempo de ciclo de um CLP.

O tempo de ciclo dependerá: do CLP, em relação a velocidade da CPU e a velocidade de I/O (aquisição de entradas e atualização de saídas); e do programa implementado, em relação ao tamanho (número de linhas) e a complexidade das funções utilizadas.

Como o tempo de ciclo depende do programa e da velocidade dos módulos de I/O utilizados, sua quantificação é difícil. Para efeitos de dimensionamento (da velocidade do ciclo do CLP), assume-se que o tempo maior é gasto na varredura de entradas e saídas, podendo-se ter uma estimativa do tempo de ciclo pelo número de I/O existentes.

Uma alternativa para se ter alterações de entradas/saídas mais rápidas é a utilização de entrada/saída imediata. Isso é possível já que a maioria dos fabricantes de CLP disponibiliza comandos de programação que “param” a varredura do programa para atualizar entradas e saídas.

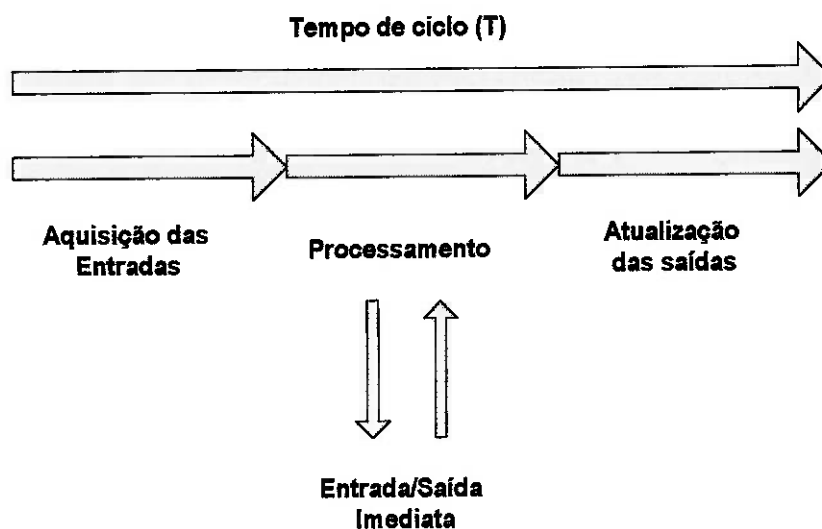


Figura 4: Entrada e Saída Imediata.

A vantagem da entrada imediata é a maior velocidade de resposta às mudanças de entrada. Entretanto, deve-se ter em mente que na entrada imediata cada linha do programa pode receber um valor diferente da entrada, dificultando a criação de lógica dividida em várias linhas. A entrada imediata prejudica a velocidade de processamento do programa, já que a

varredura do programa deve parar até que a entrada seja lida. Como aplicações usuais de entrada imediata têm-se: procedimentos de emergência e procedimentos de interrupção.

Maior velocidade de alteração da saída é a grande vantagem da saída imediata. Porém, tem-se várias desvantagem, como: cada linha de programa pode gerar um valor diferente à saída, dificultando a criação de lógica dividida em várias linhas; menor velocidade de processamento, já que a varredura do programa é interrompida para a atualização da saída; possibilidade de uma saída mudar mais de uma vez em um único ciclo; possibilidade de ruído na saída. Como aplicações usuais de saída imediata têm-se: procedimentos de emergência e procedimentos de interrupção.

1.3 Arquiteturas de Entrada e Saída Remota

Quando se deseja acomodar muitos módulos de I/O em um rack, depara-se com o problema do tamanho do barramento. Muitas vezes, o barramento original do CLP não suporta o número de I/O que se deseja. Nesse caso, pode-se partir para as seguintes soluções de arquiteturas de entrada e saída:

- extensão de barramento;
- remotas;
- redes de CLP.

Utilizando-se um extensor de barramento, pode-se conectar mais módulos de I/O numa mesma CPU de CLP. Como características dessa arquitetura de entrada e saída têm-se:

- módulos adicionais entram na varredura;
- pequena distância possível para a extensão;
- dificuldade na instalação em outros gabinetes (muitos fios).

Uma arquitetura alternativa é a de remotas. Como características da arquitetura de remotas, têm-se:

- módulos adicionais entram na varredura;
- longa distância (alguns Km);
- conexão em rede (poucos fios);
- configuração da varredura por segmentos;
- limite no número de remotas.

O último tipo de arquitetura é a arquitetura de entrada e saída por redes de CLP. Suas características são:

- módulos adicionais não entram na varredura;
- módulos adicionais podem ser consultados via I/O direta;
- longas distâncias;
- conexão em redes (poucos fios).

A aplicação dirá qual a arquitetura de entrada e saída mais adequada.

1.4 Especificação do CLP

A seguir apresenta-se os pontos principais que se deve discutir quando da especificação de um CLP e seus periféricos para um sistema de controle.

Quanto ao Sistema:

- Contexto;
- Processamento;
 - ◆ Arquiteturas de Entrada e Saída Remota;

- I/O;
- Tolerância a Falhas (Disponibilidade).

Quanto a CPU:

- I/O;
- Memória;
- Comunicação;
- Velocidade de processamento;
- Outros:
 - ♦ Memória EEPROM;
 - ♦ Disquete.

Quanto a I/O:

- I/O digitais;
- I/O analógicas;
- I/O especiais;
- Chassis/gabinetes;
- Fonte de alimentação.

Quanto ao Software de Programação:

- Plataforma;
- Qualidade.

1.5 Linguagens de Programação

O IEC (International Electrotechnical Committee) é um dos órgãos internacionais que desenvolve padrões para as linguagens para o controle de SEDs. Os padrões são definidos de modo flexível para estabelecer as especificações mínimas a serem respeitadas e as regras para expansões futuras.

Pode-se classificar as linguagens de programação quanto ao tipo em: textuais, gráficas e tabulares. A seguir tem-se as linguagens de programação classificadas de acordo com esse critério (ref. Miyagi).

- Textuais:
 - ♦ Álgebra Booleana;
 - ♦ IL (Instruction List);
 - ♦ ST (Structured Text).

- Gráficas:
 - ♦ LD (Ladder Diagram);
 - ♦ FBD (Function Block Diagram);
 - ♦ Fluxograma;
 - ♦ SFC Sequential Flow Chart).

- Tabulares:
 - ♦ Decision Table.

1.5.1 Álgebra Booleana

A programação é feita utilizando expressões booleanas. Como não contém elementos de temporização e sequencialização, não é utilizada na sua forma pura para descrever programas para SEDs. É utilizada na concepção e análise lógica.

$$O10 = I1 \cdot \bar{I2} \cdot (I3 \cdot \bar{I4} \cdot I5 \cdot I6) \cdot I7 + I8 \cdot I9$$

Figura 5: Exemplo de programação por Álgebra Booleana.

1.5.2 IL (Instruction List)

Como o próprio nome indica, é uma lista contínua com comandos correspondentes a funções elementares (internos) do CLP. Há funções como Load (LD), And, Or, Store (ST) e códigos (endereços) das entradas e saídas. Os comandos são dispostos numa sequência correspondente a sua ordem de execução.

LD	I1
ANDN	I2
AND (I3
ANDN	I4
OR (I5
AND	I6
)	
)	
AND	I7
OR (I8
AND	I9
)	
ST	O10

Figura 6: Exemplo de programação por IL.

1.5.3 ST (Structured Text)

É uma linguagem de programação de alto nível, sendo que a forma (sequência) do texto não tem relação com sua ordem de execução. A principal vantagem é a possibilidade de estruturação de programas com processamento numérico, operações de comparação, comandos If, Case, For, etc.

O10:= I1& not(I2)&(I3& not(I4)orI5& I6)& I7orI8& I9

Figura 7: Exemplo 1 de programação por ST.

```
IF AAA>=LIMITE
  THEN
    PESO:=AAA*BBB*BCD_TO_INT(CCC);
    OK:=1;
  ELSE
    FOR I:=1 TO 100 BY 2 DO
      PESO:=DDD(I)+PESO;
    END_FOR;
    OK:=0;
  END_IF
```

Figura 8: Exemplo 2 de programação por ST.

1.5.4 LD (Ladder Diagram)

Tem o formato da representação de diagramas de relês eletromagnéticos, acrescidos de funções específicas para o controle de CLPs. É uma das linguagens mais utilizadas, devido a própria história dos CLPs (ligada aos antigos painéis de controle por relês eletromagnéticos).

Essa será a linguagem utilizada na programação do CLP-5/15-B que controla a célula de mistura do CIS-Poli.

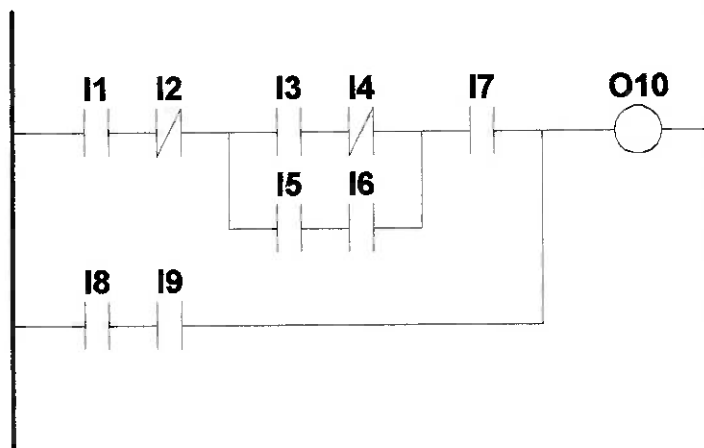


Figura 9: Exemplo de programação por LD.

1.5.5 FBD (Function Block Diagram)

Essa linguagem utiliza combinações de blocos And, Or, Soma, Multiplicação, Comparação, etc para representar o programa de controle. Quando somente blocos de funções lógicas são utilizados, também é chamado de Diagrama de Circuito Lógico.

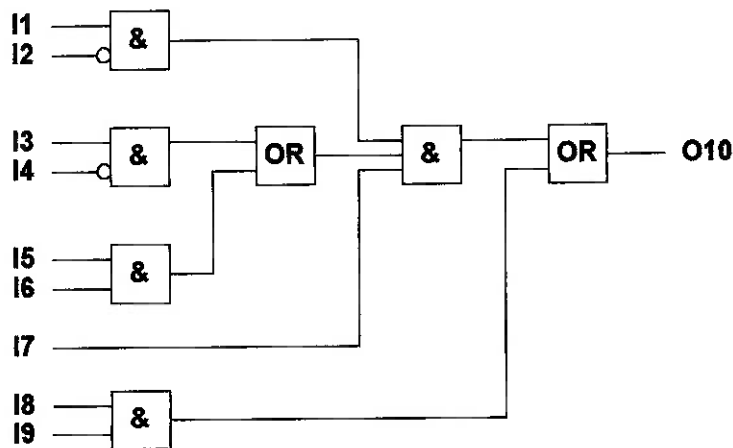


Figura 10: Exemplo de programação por FBD.

1.5.6 Fluxograma

Consiste na mesma técnica de fluxogramas utilizada na descrição de programas de computador, adaptada para o controle de SEDs. É adequada para controle puramente sequencial.

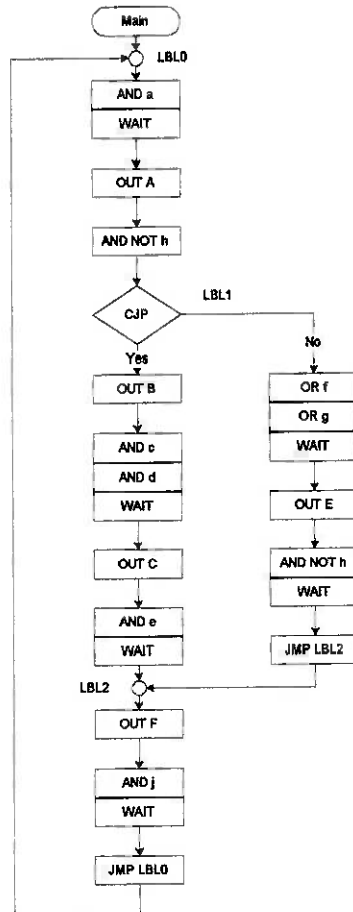


Figura 11: Exemplo de programação por Fluxograma.

1.5.7 SFC (Sequential Flow Chart)

Essa forma de descrição originou-se das Redes de Petri, sendo muito adequada para controle de SEDs. É caracterizada por steps (passos, condições) e transitions (transições, eventos).

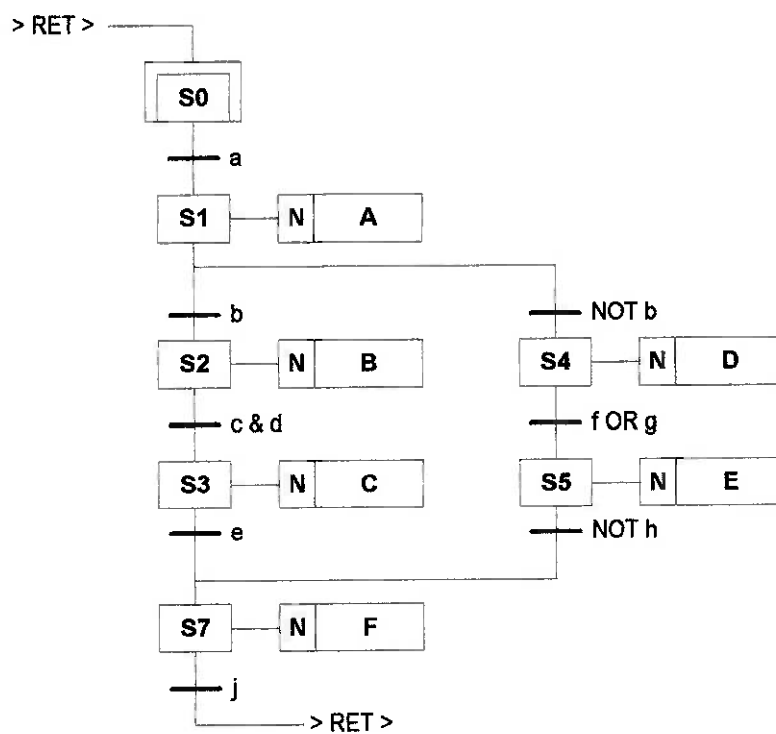


Figura 12: Exemplo de programação por SFC.

1.5.8 Decision Table

É uma forma de descrição de controle de SEDs através de tabelas. As tabelas contém para cada passo (step), a ação correspondente ao passo, a identificação dos próximos passos possíveis e as condições necessárias para a transição para o próximo passo.

	step	S0	S1	S2	S3	S4	S5	S6	
condição	a	Y							
	b		Y	N					
	c			Y					
	d			Y					
	e				Y				
	f					Y	Y		
	h							N	
	j								Y
	condição	A		Y	Y				
		B			Y				
C					Y				
D						Y	Y		
E								Y	
F									Y
próximo	step	S1	S2	S4	S3	S6	S5	S6	
								S0	

Figura 13: Exemplo de programação por Decision Table.

1.6 O CLP-5/15-B

O controle do CIS-Poli é realizado por um controlador da Allen-Bradley (Rockwell Automation) da família CLP-5. Trata-se de um sistema composto por um controlador CLP-5/15-B (1785-LT), a estrutura de I/O (chassi de I/O), um módulo remoto e diversos racks de I/O digital e analógica.

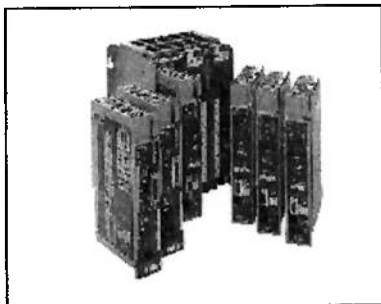


Figura 14: CLP-5/15.

O controlador CLP-5 possui uma memória central CMOS/RAM de leitura e escrita para armazenar as instruções dos programas de aplicação, valores numéricos (constantes do programa) e o estado dos dispositivos de I/O. Visto que a memória do CLP é programável, o programa de aplicação pode ser facilmente alterado, possibilitando mudanças rápidas no sistema de controle da planta. A memória do CLP-5/15 pode ser expandida através dos Módulos de Expansão de Memória 1771-MR (de 4K Words) ou 1771-MS (de 8K Words), ou então ter a memória copiada através do Módulo de Memória EEPROM 1785-MJ (de 8K Words de memória).

O CLP-5/15 possui as seguintes características técnicas:

O CLP-5/15 possui as seguintes características técnicas:

- Programação através de diagramas de relês (ladder diagrams), incluindo instruções avançadas como: matemáticas, conversão de dados, controle PID, transferência de dados, controle do programa, etc;
- Realizar backup da memória com módulo EEPROM;
- Execução repetitiva de uma sequência de instruções, através da rotina de interrupção com temporização selecionável;
- Programar rotinas de falhas para responder a falhas do sistema;
- Inserir, adicionar, alterar ou eliminar instruções ou linhas sem parar o sistema, através da programação on-line.

A especificação técnica do CLP-5/15 é a seguinte:

- Capacidade de memória: 6K Words;
- Capacidade máxima de memória: 14K Words;
- Backup de memória com módulo EEPROM: 8K Words;
- Pontos de I/O: 512;
- Números de chassis de I/O: 4;
- Número de gavetas de I/O em uma configuração típica: 1 local e 3 remotas;
- Compatibilidade de I/O: toda estrutura 1771 de I/O;
- Endereçamento de I/O:
 - ♦ 2 ranhuras: módulos de 8 pontos qualquer combinação, de 16 pontos devem ser utilizados aos pares, isto é, uma entrada e uma saída;
 - ♦ 1 ranhura: módulos de 8 ou 16 pontos em qualquer combinação.
- Comunicação:
 - ♦ modo controlador (varre I/O local e remotas);
 - ♦ modo adaptador (adaptador de um controlador principal);
 - ♦ rede de comunicação Data High Plus;
 - ♦ rede Data High Plus através de módulo adaptador de comunicação.
- Programação:
 - ♦ programa na forma de diagramas de contato (Ladder Diagram);
 - ♦ configuração de I/O.
- Conjunto de instruções:
 - ♦ de tipo relê até avançadas;
 - ♦ controle PID;
 - ♦ interrupção com temporização selecionável;
 - ♦ diagnósticos de máquina.
- Expansão de memória (opcional):
 - ♦ módulo de expansão de memória CMOS RAM 1785-MR de 4K Words;
 - ♦ módulo de expansão de memória CMOS RAM 1785-MS de 8K Words;
 - ♦ módulo de memória EEPROM 1785-MJ de 8K words para backup.

- Configuração da memória:
 - ♦ até 1000 arquivos de programas;
 - ♦ até 1000 arquivos de dados.
- Varredura do programa:
 - ♦ 2 ms/K Words (lógica de bit);
 - ♦ 8 ms/K Words (típico).
- Varredura de I/O (típico):
 - ♦ 1 ms/gaveta de I/O local;
 - ♦ 7 ms/gaveta de I/O remota.
- Relógio de calendário:
 - ♦ variação máxima a 60°C: ± 3 minutos por mês;
 - ♦ variação típica a 20°C: ± 20 segundos por mês;
 - ♦ precisão: uma varredura de programa.
- Terminal de programação: Terminal Industrial 1770-T5;
- Corrente através da placa de fundo do chassi de I/O: 2,5A
- Bateria de Back-up: bateria de lítium 1770-XY (ou similar de 3,6V).

2. SISTEMAS SCADA

Os sistemas do tipo SCADA (Supervisory Control and Data Acquisition) caracterizam-se por gerenciar a produção de uma empresa, atingindo todos os níveis supervisórios, controlando e coletando dados.

2.1 Arquitetura dos sistemas SCADA

Funções SCADA

Os sistemas SCADA são de extrema importância para o controle do processo de produção de uma empresa, não somente a nível de chão de fábrica, mas como também a nível gerencial. A Figura 15 seguinte apresenta a relação do sistema SCADA com as atividades de processo e gerência:

GERÊNCIA

- ordens
- estado
- qualidade
- previsões
- produtividade
- perdas
- sugestões de melhoria



PROCESSO

- ponto de operação
- etapas
- parâmetros
- falhas
- acompanhamento
- tendências

Figura 15: Funções SCADA

Arquitetura Interna dos Sistemas SCADA

Para supervisionar e adquirir dados do processo um Sistema SCADA necessita de uma estrutura que permita uma eficiente troca de informações entre usuário e processo.

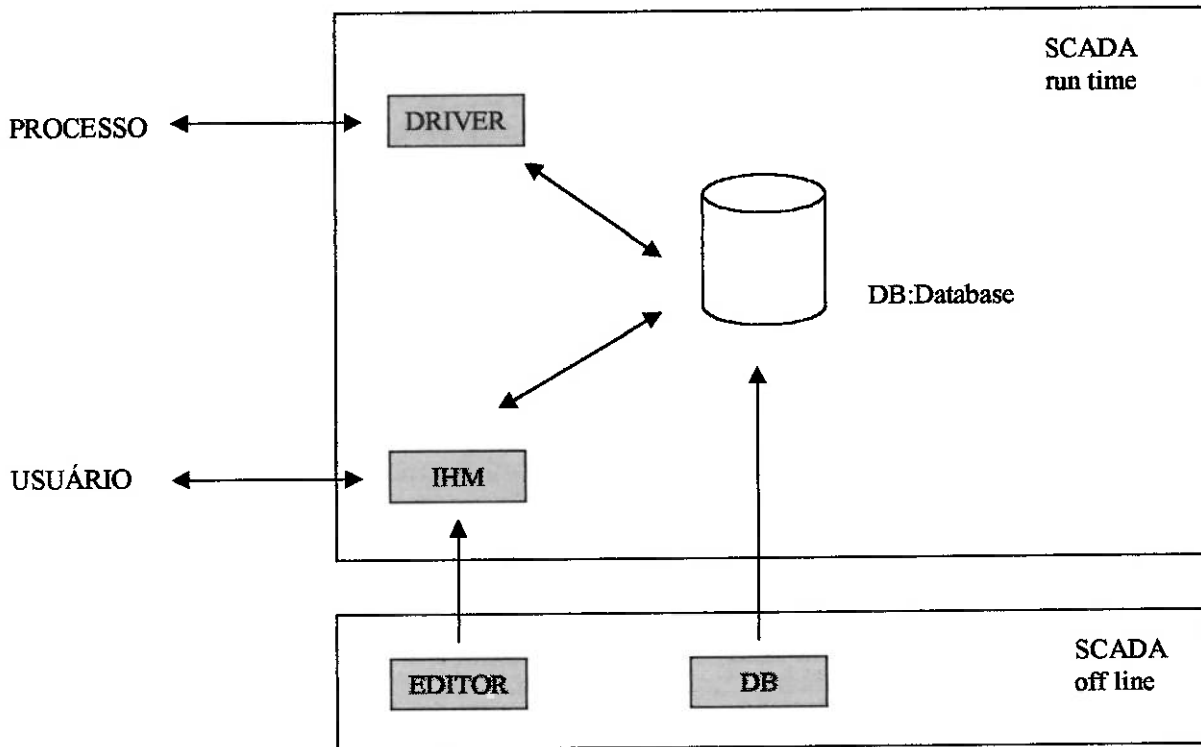


Figura 16: Arquitetura Interna dos Sistemas SCADA

O banco de dados (Database) deve ser configurado adequadamente para receber informações relativas ao processo. Para tal deve-se conhecer os tipos de dados (E/S analógica, E/S digital, Derivados) e seus atributos (nome, endereço de varredura, conversões).

Um sistema SCADA deve possuir uma interface homem-máquina amigável, possibilitando que o usuário intervenha no processo ou receba informações armazenadas no banco de dados. Dessa forma, a interface deve permitir entrada de dados e exibir variáveis de processo através de textos, barras, cores e cinóticos.

Para que seja realizada a comunicação do Sistema SCADA com o processo é necessário um driver que pode ler ou escrever tags no banco de dados. Este driver recebe informações do processo através de equipamentos como coletores de dados ou PLCs.

Alarmes e Tendências

Um Sistema de SCADA deve permitir a configuração de alarmes e tendências para garantir a segurança e eficiência do processo. O objetivo dos alarmes é informar aos operadores sobre qualquer anomalia ou mudança de estado durante o processo, para que possam tomar medidas para superar essas anomalias. Já os gráficos de tendência têm como objetivo informar aos operadores sobre o desenvolvimento do processo.

Controle de Batelada

Quando trata-se de um processo do tipo batelada, o Sistema SCADA deve permitir salvar e recuperar arquivos de dados históricos do banco de dados, utilizando arquivos com base de tempo, ou com nomes definidos pelo usuário.

2.2 O SCADA UniSoft

2.2.1 Introdução

Unisoft é um poderoso e flexível software para automação de processos, permitindo criar aplicações em diversos segmentos:

- aquisição de dados;
- interfaces homem-máquina;
- estações de supervisão local de processos industriais;
- estações concentradoras de dados em processos distribuídos;
- estações remotas de supervisão e operação;
- integração de dados e comunicação com sistemas corporativos.

Utiliza microcomputadores e estações de trabalho interligados em tempo real a processos ou máquinas, através de controladores programáveis, I/O remotos, ou outros equipamentos de aquisição de dados.

Possuindo versões para Windows 3.11, Windows 95 e Windows NT, Unisoft permite a criação desde interfaces de visualização de dados até a integração completa de sistemas, com dados distribuídos em rede local.

Unisoft fornece as funções necessárias para controle e supervisão de processos (MMI - Man-Machine Interface), assim como para o armazenamento e aquisição de dados (SCADA - Supervisory Control And Data Acquisition) oferecendo assim uma poderosa ferramenta para automação de processos.

Utiliza-se ao máximo dos recursos fornecidos em cada versão do Windows, para oferecer uma ferramenta poderosa e flexível para automação de processos, aliada a múltiplos recursos para manipulação de dados.

2.2.1.1 Plataformas

Windows 3.11

Mínimo: 486 DX2, 8MB de RAM

Recomendado: Pentium, 16 MB de RAM

Windows 95

Mínimo: 486 DX2, 16 MB de RAM

Recomendado: Pentium, 16 MB de RAM

Windows NT

Mínimo: Pentium, 16 MB de RAM

Recomendado: Pentium, 32 MB de RAM

2.2.1.2 Módulos

Módulo Principal para a Seleção e Criação de Aplicações



Application Manager : módulo para selecionar e criar diretórios de aplicação, selecionar e criar diretórios de aplicação, selecionar os módulos que serão ativados automaticamente na inicialização do sistema e o(s) driver(s) de comunicação.

Módulos de Configuração de Aplicações



Appbuild.exe

Application Builder: editor gráfico orientado a objeto para a criação de telas dinâmicas e saída e entrada de dados, que permitem uma interface amigável entre o supervisor e o usuário. Com opções para suporte a bitmaps (incluindo animações).



Dbm.exe

Database Manager: ambiente MDI (Multiple Document Interface) com módulos para a definição do banco de dados da aplicação, classes, alarmes, gráficos de tendência, relatórios, receitas, eventos escalonados e planilhas matemáticas.



Security.exe

Security System: módulo de configuração para definir os grupos de usuários e níveis de segurança, que inibem o acesso de operadores a determinadas telas ou variáveis da aplicação.



Drvconf.exe

Driver Configuration: módulo para configuração dos drivers de comunicação.



Uddeconf.exe

Unisoft DDE Client Configurator: módulo para a configuração do Unisoft como cliente DDE.



Uniodbc.exe

ODBC Configurator: módulo para a configuração do driver ODBC.

Módulos Utilitários:



Dbspy.exe

Database Spy: ferramenta para auxílio no desenvolvimento de aplicativos, utilizada para leitura ou forçamento dos valores de variáveis.



Logwin.exe

LogWin: ferramenta para auxílio no desenvolvimento de aplicativos, que apresenta mensagens, códigos de erro e eventos na execução da aplicação.



Dbfinder.exe

Database Find/Create Tags: ferramenta para verificação rápida de um tag contido na aplicação e criação de novos tags.



Trans.exe

Translation Tool: ferramenta para a geração de arquivos de tradução para aplicações em vários idiomas.



Dbpack.exe

Database Pack Utility: otimiza o tamanho do arquivo do banco de dados, eliminando as linhas em branco e otimiza a leitura das telas de sinótico.



Logon.exe

Log On/off: conexão e saída do usuários no sistema.

Módulos para Execução da Aplicação ou de Runtime:



Startup.exe

Startup: módulo para inicializar automaticamente os outros módulos de execução.



Driver.exe

Driver RunTime: executa o driver selecionado e configurado.



Viewer.exe

Viewer: módulo de execução que implementa a interface homem-máquina.



Bgtask.exe

Background Tasks: módulo de execução responsável pelas tarefas que rodam em background como cálculos matemáticos, históricos, alarmes e escalonadores.

2.2.2 Banco de dados

Tags são variáveis utilizadas para configurar todos os módulos de supervisão. Tags podem ser pontos de comunicação com o equipamento de campo, resultados de cálculos realizados pelo módulo Mathsheet, pontos de alarme, valores de entrada do operador, variáveis internas do sistema etc.

No UniSoft, existem os tags criados pelo usuário, chamados de tags da aplicação e os tags internos do sistema. Tanto os tags internos como os tags da aplicação podem ser utilizados em qualquer módulo. A única diferença entre os dois tipos é que internos possuem

funções pré-definidas. Ambos os tipos de tags têm seus valores armazenados no Application Database durante a execução do módulo Background Task.

2.2.2.1 Tipos de Tags

Quanto aos valores que os tags podem assumir, existem quatro tipos-padrão:

- **Boolean** (digitais) : variável booleana ou digital (0 ou 1).
- **Integer** (analógicos) : número inteiro (pode ser positivo, negativo ou zero).
- **Real** (ponto flutuante) : número real; representado com uma parte fracionária. O tamanho máximo de dígitos é de 8 bytes.
- **String** (ASCII) : cadeia ou seqüência de caracteres. Pode conter qualquer valor como letras, números ou caracteres especiais, até 256 (de 0 a 255) caracteres.

Além dos tipos-padrão, o usuário pode definir novas classes de tags:

Class (classes) : classes são estruturas que permitem um alto grau de encapsulamento no banco de dados da aplicação.

Vetores

Os tags no UniSoft podem ter um único valor escalar ou conter um vetor de valores. Tag vetor é um conjunto de tags de mesmo nome, com índices distintos. O tamanho máximo do tag vetor é de 512 posições (de 0 a 512).

Utilizar tags vetores é interessante pois em muitas situações pode-se poupar configuração com seu uso. Os tags vetores usados nos módulos Database Manager e Application Builder, reduzem muito o tempo de desenvolvimento de aplicativos.

2.2.2.2 Tags especiais

Tags Internos

São tags pré-definidos no UniSoft, que possuem funções específicas nas tarefas de supervisão. Alguns exemplos de tags internos:

TagName	Descrição
Hour	hora corrente (0 a 23) (integer)
Minute	minuto (0 a 59) (integer)
Second	segundo (0 a 59) (integer)
Day	dia corrente (1 a 31) (integer)
Month	mês (1 a 12) (integer)
Year	ano (1994 -) (integer)
DayOfYear	dia do ano (1 a 366) (integer)
Yesterday	ontem (1 a 31) (integer)
Tomorrow	amanhã (1 a 31) (integer)
Time	hora HH:MM:SS (string)
Date	dia DD-MM-AA (string)
Goto	função matemática (string)
Next	função matemática (integer)
Label	função matemática (string)
Hint	exibe o hint corrente (string)

Obs: Os tags internos não podem ser alterados pelo usuário !

Tags da Aplicação

São todos os tags criados pelo usuário nas configurações como telas de sinóticos, leitura e escrita no equipamento de campo, para controle, auxiliares para cálculos matemáticos etc.

2.2.2.3 Acesso externo ao banco de dados

Interfaces DDE, NetDDE e ODBC. DDE - Dynamic Data Exchange, permite trocar valores com outros programas Windows, como Excel, Access, etc. NetDDE, permite trocar valores com outros programas Windows rodando em outros computadores da rede. ODBC, permite o acesso a base de dados relacionais da rede local.

2.2.3 Telas

O módulo Application Builder: permite a criação de desenhos de telas, objetos dinâmicos, campos de entrada e saída de dados e configuração de teclas de comando.

O objetivo deste módulo é permitir a criação das telas de interface do sistema, a configuração de funções necessárias para uma aplicação e servir de ponto central para acessar todos os outros módulos do UniSoft.

O Application Builder possui ferramentas que permitem o desenho das telas de sinóticos e a criação das dinâmicas de interface com o operador. É dividido em dois ambientes: Object Editor e Bitmap Editor.

Cada tela é uma coleção de objetos estáticos e dinâmicos. A criação de objetos é feita pelo editor de objetos Object Editor. Uma tela pode conter também um bitmap opcional que atua como um pano de fundo da janela de objetos. Para ajudar no desenvolvimento deste tela, o AppBuilder inclui também o editor de bitmap Bitmap Editor.

Para iniciar a operação no Application Builder é necessário abrir ou criar um arquivo através do menu principal, conforme apresentado na tela abaixo:

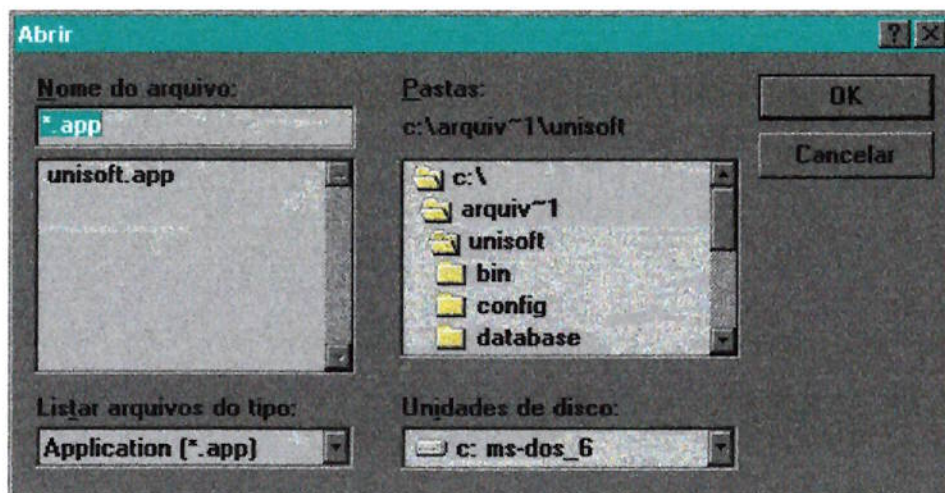


Figura 17: Tela do menu principal

A figura 18 apresenta a tela própria para a construção da aplicação em si, através de objetos que serão criados com a ajuda da barra de ferramentas chamada Toolbar (Figura 19).

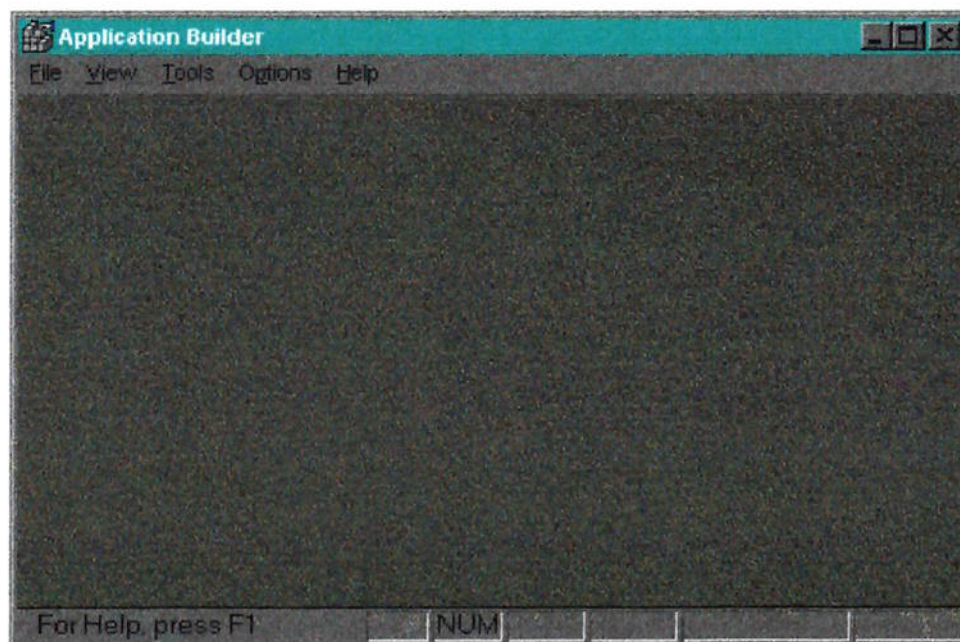


Figura 18: Tela do Application Builder



Figura 19: Barra de Ferramentas
do Application Builder

2.2.4 Alarmes e tendências

2.2.4.1 Configuração de alarmes

No módulo Database Manager, o usuário define as mensagens de alarme que o UniSoft irá emitir sob causas pré-estabelecidas.

O objetivo dos alarmes é fazer com que os operadores sejam informados sobre qualquer anomalia ou mudança de estado durante todo o processo e, então, poderem tomar medidas necessárias para superar essas anomalias e manter um bom desenvolvimento do processo.

Para exibir mensagens de alarme numa tela de sinótico também é necessário criar a dinâmica de alarme no Editor de Objetos.

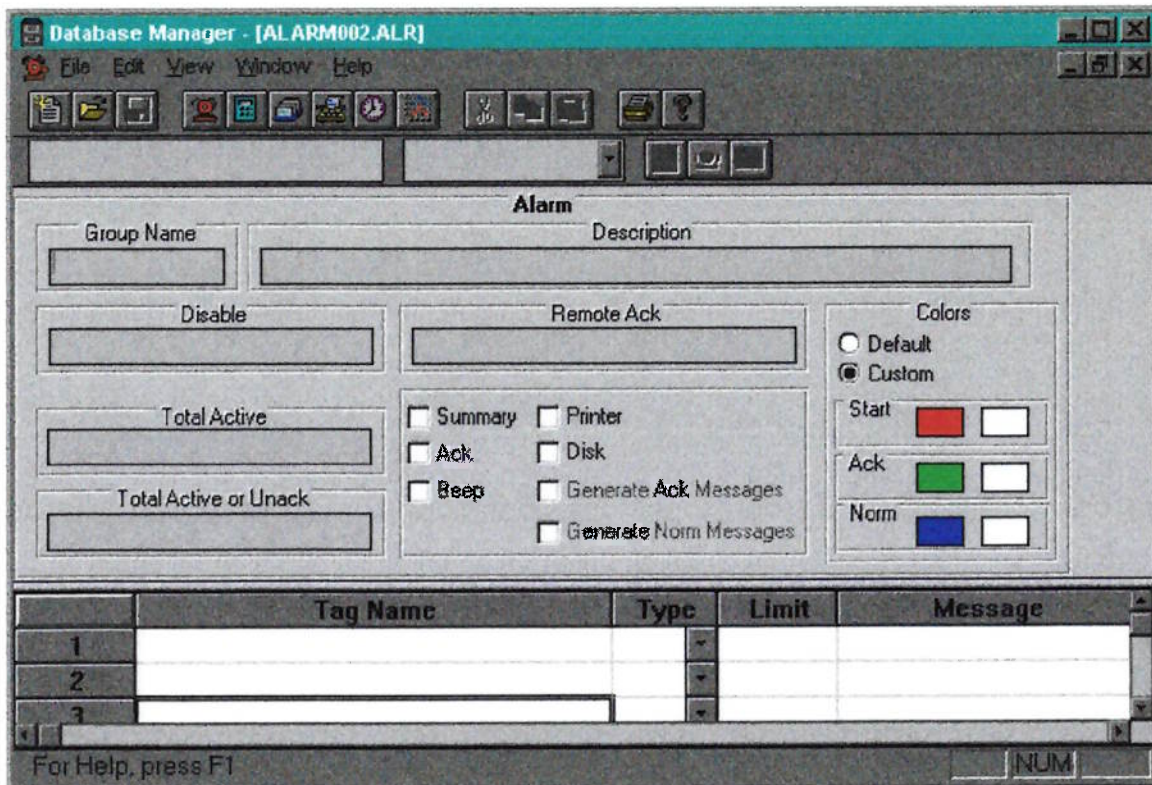


Figura 20: Planilha de configuração de alarmes

A planilha de configuração de alarmes define um conjunto de características comuns a todos os alarmes do grupo:

Description - Este campo é preenchido por um comentário e tem por objetivo auxiliar na documentação da aplicação.

Ack - Quando selecionado, exige reconhecimento de mensagens de alarme.

Group Name - Nome usado para diferenciar os grupos de alarmes.

Beep - Quando selecionado, ele é acionado até o reconhecimento do alarme.

Disable - Desabilita todos os alarmes do grupo. Esse campo deve ser preenchido com um tag. Se o valor do tag for maior que zero o grupo estará desabilitado, não gerando mensagens. Se o campo não for preenchido, o grupo estará continuamente habilitado. Se o tag digitado não tiver sido criado, aparecerá uma janela para a confirmação da criação do tag .

Printer - Quando selecionado, indica que as mensagens de alarme desse grupo serão impressas.

Remote Ack - Tag para reconhecimento de alarme. O reconhecimento ocorre quando este tag muda de valor.

Disk - Quando selecionado, indica que as mensagens de alarme desse grupo serão armazenadas em disco rígido.

Total Active - Campo que armazenará o total de alarmes ativos no grupo. Seu valor será atualizado pelo sistema cada vez que ocorrer um alarme. Deve ser preenchido com um tag. Se o tag digitado não tiver sido criado, aparecerá uma janela para a confirmação da criação do tag.

Generate Ack Message - Quando selecionado, irá gerar mensagens quando os alarmes desse grupo forem reconhecidos.

Total Active or Unack - Campo que armazenará o total de alarmes ativos ou não-reconhecidos no grupo. Seu valor será atualizado pelo sistema cada vez que ocorrer um alarme. Deve ser preenchido com um tag. Se o tag digitado não tiver sido criado, aparecerá uma janela para a confirmação da criação do tag .

Generate Norm Message - Quando selecionado, irá gerar mensagens quando os alarme desse grupo voltarem ao normal.

Summary - Quando selecionado, gera um sumário com as mensagens de alarme desse grupo.

Nota: Se a opção Summary não for selecionada, os alarmes deste grupo não aparecerão nos objetos de alarmes no Viewer.

Color - Esta opção é utilizada para definir as cores do sumário de alarmes para os objetos de alarme definidos no AppBuilder. Se o radio button escolhido for para Custom, cada entrada no sumário de alarmes usará a cor customizada de acordo com o grupo de alarme ao qual a mensagem pertence, caso contrário (Default radio button), serão utilizadas as cores pré-definidas.

Arquivo de histórico de alarmes

Os histórico de alarmes, quando habilitado no grupo, é salvo no arquivo:

`\app\ALARM\ALyyymm.ALH` , onde

yy: ano do histórico

mm: mês do histórico

app: diretório da aplicação

É um arquivo ASC com o seguiten formato:

Alarm Summary (primeira linha)

(segunda linha em branco)

<1>|<2>|<3>|<4>|<5>|<6>|<7>|<8>|<9>|<10>|<11>|<12>|<13>|<14>|<15>| (próximas linhas)

Onde:

<1> = reservado, (número interno de versão do documento, atual é 000)

<2> = data de início do alarme

<3> = hora de início do alarme HH:MM:SS

<4> = nome do tag

<5> = 1:alarme está esperando ack 0:outro caso

<6> = 1:alarme está ativo 0:outro caso

<7> = valor do tag

<8> = número do grupo de alarme

<8> = prioridade do alarme

<9> = texto do campo selection

<10> = tipo de alarme

<11> = 1:tipo de alarme requer ack 0:outro caso

<12> = data de término do alarme

<13> = hora de término do alarme HH:MM:SS

<14> = data de reconhecimento do alarme

<15> = hora de reconhecimento do alarme, HH:MM:SS

2.2.4.2 Configuração de tendências

O objetivo dos gráficos de tendência é informar os operadores sobre o desenvolvimento do processo. Neste módulo o usuário definirá a configuração de grupos de trend que o UniSoft irá gerar.

O módulo Trend tem como funções primordiais:

- Monitorar tags gerando curvas de tendência nas telas de sinóticos tanto on-line, como de dados armazenados em disco;
- Armazenar os dados de monitoração em arquivos de histórico.

No caso de gráfico de tendência de histórico, o primeiro passo é a escolha das variáveis que serão armazenadas em arquivos de histórico.

Esta escolha é feita de forma bem simples, apenas clicando sobre a variável em questão no Application Database.

Apresenta-se, a seguir, a planilha de configuração de tendências:

	Tag Name	Dead Band
1		
2		

Figura 21: Planilha de Configuração de tendências

Description - Este campo é preenchido por um comentário e tem por objetivo auxiliar na documentação da aplicação.

Disable - Desabilita todas as curvas de tendência do grupo. Esse campo deve ser preenchido com um tag.

Se o valor do tag for maior que zero o grupo estará desabilitado, não gerando curvas de tendências. Se o campo não for preenchido, o grupo estará continuamente habilitado.

Se o tag digitado não tiver sido criado, aparecerá uma janela para a confirmação da criação do tag .

File Life Time (days) - Determina a quantidade de dias que um arquivo histórico ficará armazenado em disco. Após o tempo determinado o arquivo será automaticamente eliminado. Essa opção só é utilizada para arquivos baseados na data.

Date(Default) - Quando selecionado, são gerados arquivos de histórico baseados na data do arquivamento. Neste caso os arquivos gerados são:

`\app\HST\ggyyyyymmdd.HST` onde,

app = diretório da aplicação

gg = número do grupo de histórico (em hexadecimal)

yyyy = ano

mm = mês

dd = dia

Batch - Quando selecionado, serão criados arquivos de históricos, utilizando como nome o valor no campo de edição. Este campo pode conter valores de tags.

Exemplo: c:\history\file{TagBatchNumber}.hst

Para visualização de curvas on-line no Viewer/Application Builder, devem estar habilitados os arquivos baseados em data. Para curvas históricas os dois tipos de arquivos podem ser usados.

Save on Trigger - Quando selecionada, esta opção grava o histórico de tendências toda vez que houver um trigger no tag especificado (histórico por evento, sendo que este evento poderá ser definido no módulo Scheduler).

Save on TagChange - Quando selecionada, esta opção grava o histórico de tendências toda vez que houver uma mudança no valor do tag especificado (histórico por período).

TagName - Esse campo é preenchido com o nome do tag para a geração de curvas de tendências.

Se o tag digitado não tiver sido criado, aparecerá uma janela para a confirmação da criação do tag .

Dead Band - Determina o valor do filtro para as variações do tag.

Para exibir um gráfico de tendência numa tela de sinótico também é necessário criar a dinâmica de trend no Editor de Objetos. Até 8 curvas podem ser expostas simultaneamente na tela de tendências. O usuário também conta com ferramentas auxiliares contidas no menu.

Para executar as funções acima, o módulo Trend faz constantes amostragens dos valores dos tags, ou seja, a cada intervalo de tempo o módulo lê os valores dos tags no DataBase e exibe na tela e/ou registra em disco. Tanto o intervalo de tempo como os tags que são monitorados são configurados pelo usuário. Nem todos os tags da aplicação são de interesse para este tipo de monitoração: o usuário configura os tags que desejar no módulo.

Processos distintos exigem tempos distintos para realizar a amostragem: 10 segundos, 30 segundos, 5 minutos (bases de tempo que serão criadas no módulo Scheduler).

Dependendo do processo, não há necessidade que este módulo seja executado o tempo todo. Pode existir um horário fixo ou é possível configurar o momento de início e fim da execução do módulo.

Existe um outro caso comum em aplicações nas quais é necessária a amostragem contínua, ou seja, o módulo trend colhe amostras o tempo todo. Neste caso não é configurado o início de execução, indicando que a amostragem será iniciada assim que partir o sistema. Porém, é necessário indicar o momento de finalizar a execução para que o arquivo de registro seja fechado, e um outro seja iniciado.

2.2.5 Ambiente

2.2.5.1 Segurança

O sistema interno de segurança permite a criação de proteções, com grupos de usuários e senhas, não apenas para a execução da aplicação, mas também para cada parte da ferramenta de desenvolvimento, definindo os grupos de usuários e níveis de segurança, que inibem o acesso de operadores a determinadas telas ou variáveis da aplicação. Inclui também segurança para o desenvolvimento (engenharia) da aplicação.

2.2.5.2 Drivers

Drivers de comunicação disponíveis para aplicações do Unisoft:

Manufacturer	Protocol/Network	Equipments	Interface
ABB	Proprietary	Micas -MS3	RS232
ACS	Proprietary		RS232
AEG	Modbus		RS232
Alfa	Proprietary		RS232
Allen-Bradley	DataHighway	Family 2	RS232+KE/KF
Allen-Bradley	D.H.Plus	Family 5	RS232+KE/KF
Allen-Bradley	Proprietary	SLC500	RS232 (DF1)
Altronic	Proprietary	DataLogger	RS232
Altronic	Proprietary	DataLogger with Radio	RS232
Altus	Alnet	AL1000	RS232
Altus	Alnet	AL2000/AL600	RS232
Altus	Alnet	QUARK	RS232
Ascon	Proprietary	XS, XP, XE	RS232+YALS
Ascon	Proprietary	BCR CPU 1	RS232+YALS
Atos		MPC506	RS485
BCM	Proprietary	108x/1086/2085	RS485
Bosch	BUEP19E	CL200/300/500	RS232
Davis	Proprietary	Weather Wizard	RS232
DCA Perroti	HLLAPI	(Link to IBM mainframe)	
Digicon	Master-Slave	D6810	RS232
Ecil/Axiom	(PC I/O board)	AX5411	
Ecil/Axiom	(PC I/O board)	AX5214	
Ecil/Axiom	(PC I/O board)	AX5232	
ELO	Elonet	MUG	RS232
Ero Electronics	Modbus	Lfs	RS485
Ero Electronics	Proprietary	MD331	RS485
Ero Electronics	Proprietary	FMC	RS485
ESB	ModBus	Vip D3	RS485
Fascitec	Proprietary	MFC	RS232
Foxboro	Proprietary	743V,743C	RS485
Foxboro	SNP	761C,762C	RS485
GE-Fanuc	Proprietary		RS232
Gefran/Adameo	Proprietary		RS485
Hitachi	Proprietary	H-Series	RS232
Hitachi	Proprietary	EB-Series	RS232
IBH	Proprietary	Virtual PLC	PC-Board
Klockner-Moeller	SUCOM 1	LPC 42	RS485
Klockner-Moeller	SUCOM-A	PS3	RS485
Klockner-Moeller	SUCOM-A	PS4-101/111	RS485
Klockner-Moeller	SUCOM-A	PS4-201(-MM1)	RS485
Mastushita	Proprietary	FP1-Cxx	RS232
MCS		CNC SX	RS232

Mitsubishi	Proprietary	FX-232AW	RS232
Novus	Proprietary	N2000	RS232
National Instr.	Proprietary	GPIB	IEE 488.1
Omron		E5CK	RS232
Philips	VDU		RS232+CI21
Schenck	Proprietary	Disomat C	RS232
Siemens	Profibus FDL	AG095	CP5412/A1
Siemens	3964R	S5	RS232+RK512
Siemens	AS511	S5	RS232
Siemens	AS511	S5-945	RS232
Siemens	Profibus DP	S5	CP5412
Siemens	Profibus FMS	S5	CP5412
Siemens	Profibus FMS	S5	CP5412/A2
Siemens	MPI Library	S7	MPI/Interface
Smar	Proprietary	CD600	RS232+ICS2.0
Smar	HART	LD301/TT301	Hart Modem
Smar	SP-50	LD/TT	PC Card
Spirax Sarco	Proprietary	SP30	RS232
Toledo	Proprietary	TBLAN	TBLAN/Interface
Usiminas	Compaq		RS232
Weg	Proprietary	A500	RS232
Weg	Proprietary	CPWA250	RS232
Weg	Pro+C31 proprietary	TP11	RS232
Yokogawa	Proprietary	HR2500E	RS232
Yokogawa	BINARY	DA100	RS232
Yokogawa	ASCII	DA100	RS232
Yokogawa	ASCII	UT37/UT38	RS422

2.2.5.3 Configurações distribuídas

O sistema é fornecido em duas versões:

- Versão de Engenharia (CNF) - Permite ao usuário configurar e executar as suas aplicações;
- Versão Run Time (RT) - Permite ao usuário executar uma aplicação já configurada.

2.2.6 Características diferenciadoras



UniSoft “Application Programming Interface” (UNIAPI)

Todos os módulos UniSoft são desenvolvidos usando uma biblioteca aberta (UNIAPI). Isto possibilita a expansão fácil e a capacidade de manter a evolução tanto do “kernel” quanto das tarefas de aplicação independentemente.



Editor de Telas Orientado a Objeto Simples e Poderoso

Agrupamento/desagrupamento hierárquico

Acesso à edição de componentes internos do grupo

Sem restrição nas operações de agrupamento e mudança de escala em “bitmaps”, dinâmicas, objetos, entradas de texto, botões, objetos de alarme e gráfico de tendência.

Cálculo de tamanho use o tamanho do grupo original, quando criado.

Acesso rápido às propriedades dinâmicas do objeto.

Editor de “bitmap” de fundo.

Diálogos modais.

Linha de status no modo Runtime.



Banco de Dados Orientado à Objeto

Tags Vetor: Qualquer tag no banco de dados pode ser um vetor de até 512 posições. Em qualquer parte do software (em todos os módulos!) onde se preenche um nome de tag, é possível usar TAG[1] or TAG[OutroTag], simplificando muitas configurações e possibilitando multiplexar telas, receitas e até mesmo a interface de comunicação.

Tags Indiretos (ponteiros): Usando o acesso @TAG, é feito uma leitura ou escrita indireta. Por exemplo, se o tag X tem o valor “Setpoint”, e usamos o acesso @X, estamos lendo ou escrevendo no tag SETPOINT. Em qualquer lugar onde se preenche um nome de tag (como o nome de uma pena no objeto de gráfico), é possível usar @TAG. x)

Classes: É possível definir uma estrutura como: ClassPID { PV, CV, SET, KP, KI, KD }, e declarar tags ou mesmo vetores do tipo ClassPID que vão conter grupos de valores, ao invés de valores simples.



Configuração “On-line”

As tarefas de runtime aceitam novas configurações imediatamente, sem precisar reiniciar os programas. É possível mudar também endereços no PLC e até mesmo o tipo de um tag. (Disponível em outros produtos, porém, limitado a partes específicas da configuração)



Fácil Inclusão de Novos Símbolos na Biblioteca

Permite inclusão de objetos com qualquer tipo de propriedade dinâmica em segundos. Seleciona e customiza os objetos na biblioteca rapidamente!



Função de Relatórios no Produto, Dispensando Excel

Imprime relatórios gráficos ou texto na impressora ou em disco. Possui editor de relatórios-texto próprio, dispensando o DDE nesta função.



Receitas em formato ASCII e DBF

Permite acesso direto a arquivos ASCII e DBF, dispensando o uso de ODBC nesta função.



DDE, NetDDE, ODBC

Usando NetDDE entre duas estações UniSoft, nós usamos uma mensagem DDE customizada automaticamente, com grupos de tags, ao invés de gerar uma mensagem para cada tag.



Sistema de Segurança

O sistema interno de segurança permite a criação de proteções, com grupos de usuários e senhas, não apenas para a execução da aplicação, mas também para cada parte da ferramenta de desenvolvimento.



Arquivos de histórico em batelada

Permite salvar e recuperar os arquivos de dados históricos, utilizando arquivos com base de tempo, ou com nomes definidos pelo usuário. Essencial quando o processo está vinculado com um processo tipo batelada.



Alarmes

Formata livremente as mensagens de alarmes.

Permite uma chave secundária de busca.

Na configuração, os alarmes podem ser acessados por grupos ou por tag.



Desenvolvimento de interfaces de comunicação simplificada

O UniSoft dispõe de um “Driver Toolkit” que torna o desenvolvimento de drivers de comunicação rápidos e simples. Basta criar um pequeno arquivo DLL, com o código específico para o seu equipamento.



Tradução das aplicações

Para criar uma aplicação que precisa ser executada em vários idiomas, basta criar uma única aplicação no seu idioma preferido, e criar os arquivos texto com os textos para a tradução. A troca de idiomas pode ser dinâmica.

2.2.7 Comercialização

2.2.7.1 Preços

Sistema Operacional	Pontos de Comunicação	Versão	Valor (R\$)
Windows 3.11	75	Engenharia	1.950
Windows 3.11	75	Run Time	1.463
Windows 3.11	150	Engenharia	2.700
Windows 3.11	150	Run Time	2.025
Windows 3.11	300	Engenharia	4.600
Windows 3.11	300	Run Time	3.450
Windows 3.11	full (32000)	Engenharia	8.400
Windows 3.11	full (32000)	Run Time	6.300
Windows NT	300	Engenharia	5.230
Windows NT	300	Run Time	3.925
Windows NT	full (32000)	Engenharia	9.500
Windows NT	full (32000)	Run Time	7.125

2.2.7.2 Integradores e certificação

A Unisoft é uma empresa credenciada pela Microsoft, inclusive com desenvolvimento.

Bases:

Brasil - 2.000

Europa - 800

Integradores regulares:

Brasil - 60

Europa - 120

3. A METODOLOGIA DE IMPLANTAÇÃO

A dimensão de um projeto de controle de um Sistema a Eventos Discretos (SED) requer uma metodologia de implantação para ser bem sucedida. Aliado a metodologia aparece a figura do modelo estrutural adotado, que influenciará toda a condução do projeto. Nos itens a seguir se detalhará o modelo adotado para a conceituação desse projeto de CIM (Computer Integrated Manufacturing) e a metodologia PFS/MFG de descrição do sistema de controle de um SED.

3.1 Modelo GRAI para CIM

Na Universidade de Bordeaux (França), um grupo de estudiosos ligados a integração e automatização da manufatura desenvolveu um modelo conceitual de esquematização de um sistema produtivo, voltado para o objetivo de identificar os níveis de controle. Esse modelo recebeu o nome de GRAI, devido ao grupo de estudiosos que o criou (Groupe de Recherche d'Automatique Industriale).

O modelo GRAI divide o sistema produtivo em duas partes:

- Sistema físico.
- Sistema de controle da produção;

A entrada para o sistema de controle são as informações sobre os recursos (humanos ou materiais), ordens de produção, energia, etc. O sistema físico tem como entradas os componentes, matérias-primas, recursos, etc, e tem como saída os produtos. O sistema de controle da produção é comparável ao sistema de controle de uma máquina. Como um controlador numérico que envia comandos para uma máquina-ferramenta, o sistema de

controle da produção envia instruções para o sistema físico e este retorna informações sobre seu estado.

3.1.1 Sistema físico

O sistema físico é um conjunto de unidades de manufatura cuja função é transformar matéria-prima, componentes ou partes semi-acabadas em produtos completos de acordo com o plano mestre de manufatura ou as especificações da engenharia.

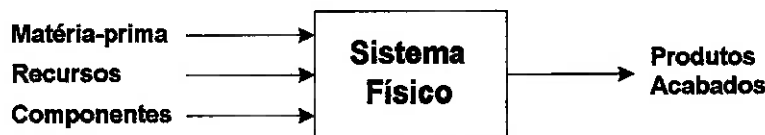


Figura 22: Modelo do Sistema Físico.

O sistema físico consiste de homens, máquinas, ferramentas e do ambiente de manufatura. As atividades de transformação podem envolver mistura, adição de corantes, estocagem, envazamento, ou qualquer combinação destes processos. O sistema físico, por si só, não é capaz de atingir seus objetivos. Por esta razão é necessário supervisioná-lo através de um sistema de controle.

3.1.2 Sistema de controle

Num sistema de controle pode-se distinguir duas partes:

- Sistema de decisão;
- Sistema de informação.

A eficiência do sistema de controle depende da consistência e qualidade das decisões e de uma boa transmissão de informações. A tarefa deste sistema é tomar decisões. Algumas destas decisões são óbvias e seu efeito é conhecido. A consequência do restante das decisões é de difícil previsão. Infelizmente, estas são geralmente as mais importantes. Portanto, é necessário projetar um sistema de controle que tome o mínimo deste tipo de decisões. Além disso, para tomar decisões o sistema de controle deve ter informações sobre o sistema em tempo real, sobre seu passado e extrapolações do futuro. Assim, uma decisão pode ter natureza estática ou dinâmica, sendo necessário que o sistema de informação esteja conectado com o sistema de decisão.

A Figura 23 esquematiza a estruturação de um sistema de produção de acordo com o modelo GRAI.

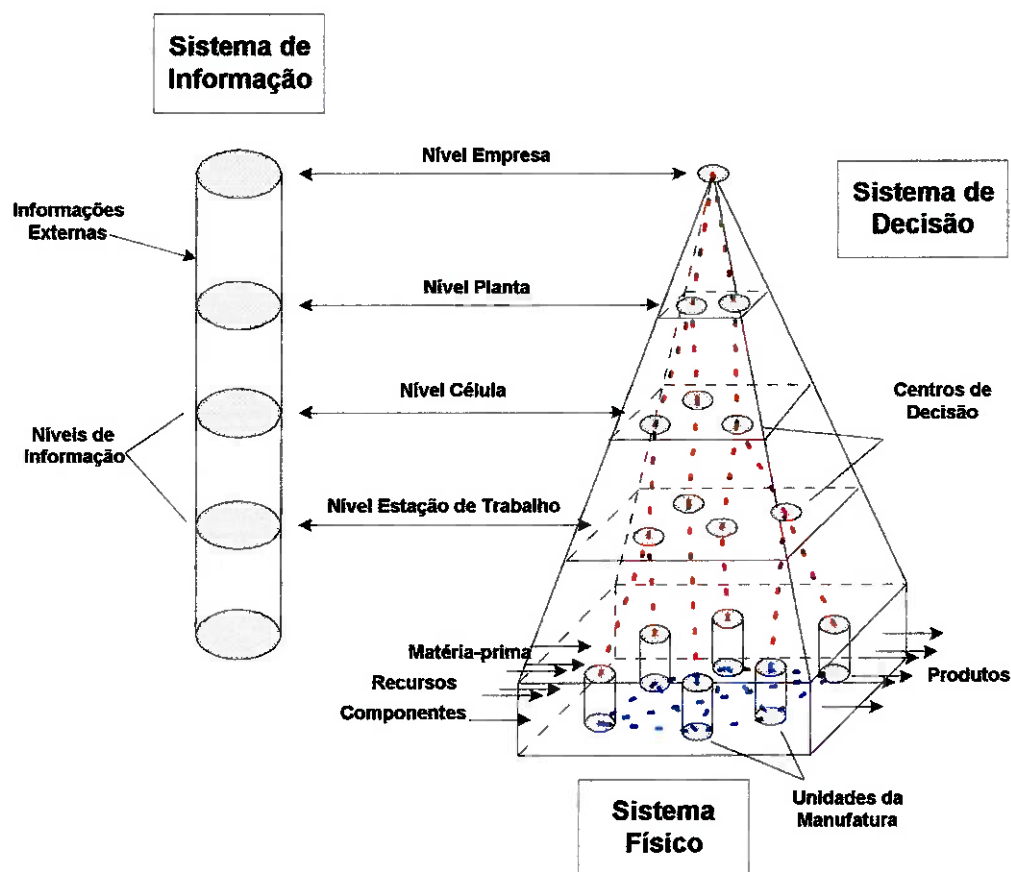


Figura 23: Modelo GRAI para um sistema de produção.

3.1.3 Proposta de Estruturação Hierárquica dos centros de decisão

No modelo da figura anterior, pode-se observar três sub-sistemas:

- Sistema de decisão, que toma decisões e realiza o controle;
- Sistema físico, que executa as atividades de manufatura;
- Sistema de informação, que interliga os dois anteriores.

A concepção proposta tem uma estrutura hierárquica. A função de cada nível é determinada pela tarefa que ele executa no sistema e pelo período de tempo para o qual as decisões são tomadas.

As informações e decisões são detalhadas em cada nível. Assim, a estrutura da informação deve ser coerente com a decisão em seu nível. Quanto mais próximo do plano horizontal, mais próxima esta a decisão do sistema físico e mais precisamente deve ser especificada a informação. Os centros de decisão podem ser estruturados de acordo com o seguinte critério:

- Tempo de resposta para a tomada de decisão;
- Ciclo de produto;
- Frequência de decisões.

3.1.4 Estrutura de um Centro de Decisão

Após o modelo conceitual do centro de decisão ter sido construído, é necessário escrever cada atividade do centro. O modelo contém os elementos básicos, funções e variáveis de ação ou relacionamento do centro de decisão, e os dados sobre os quais o “tomador de decisão” tem influência. As decisões são tomadas levando-se em conta a eficiência e performance dos elementos envolvidos. O “tomador de decisão” define a alocação de recursos

para um objetivo desejado e as responsabilidades para os níveis inferiores. Fornece informações sobre o estado, performance medida e dados técnicos para um dado nível.

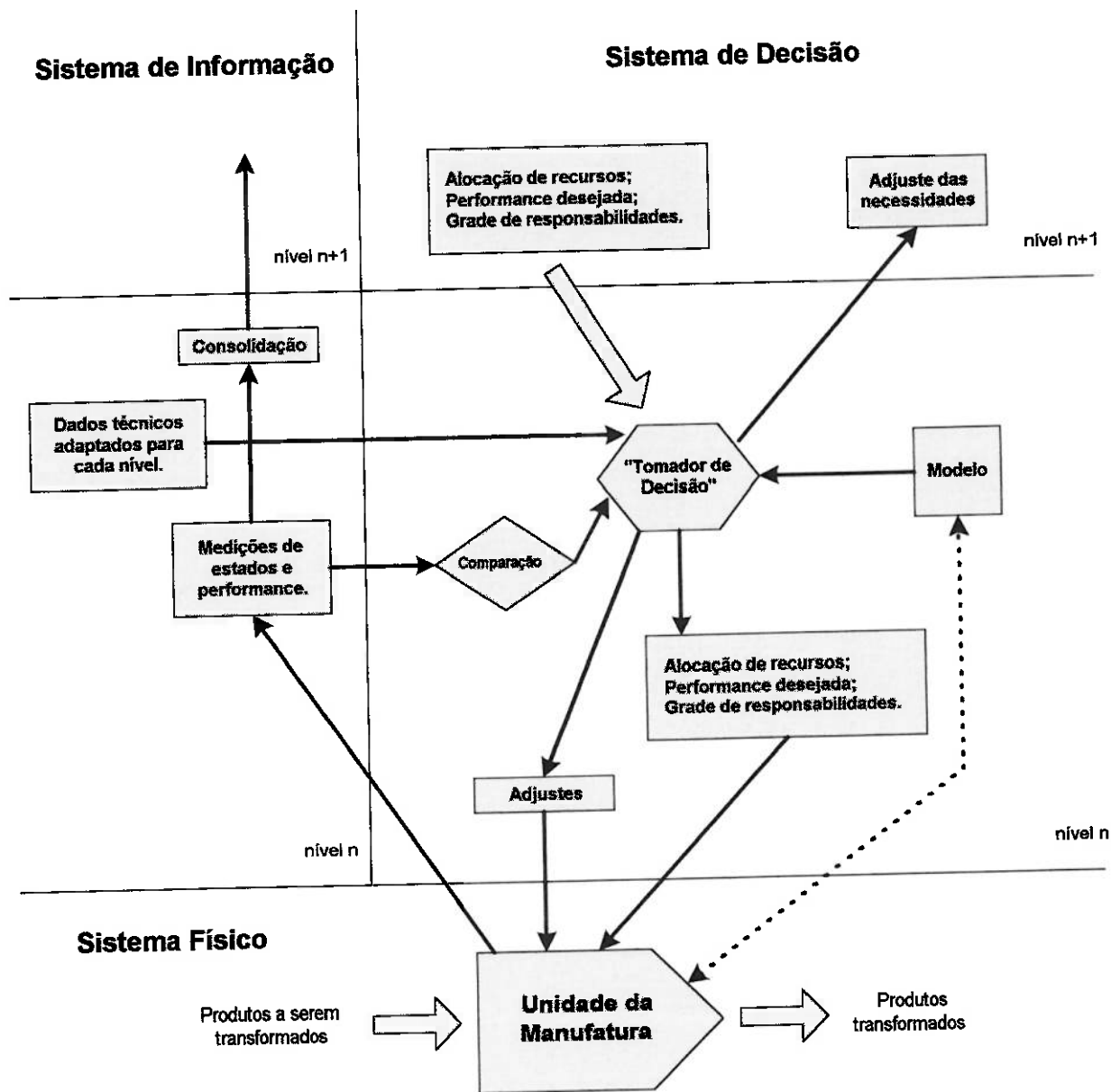


Figura 24: Modelo GRAI para o centro de decisão.

Também é importante coordenar as atividades entre os níveis de centro de decisão. Um centro de decisão de um dado nível define os elementos requeridos para a coordenação do centro de decisão do nível abaixo.

Os modelos do centro de decisão e da estrutura hierárquica dos centros é uma parte muito importante do método GRAI. Com a ajuda desse modelo, pode-se ter uma visão geral da empresa, incluindo a estrutura hierárquica, visão detalhada do fluxo de informações e as funções de decisão de cada centro. No projeto em questão, atuar-se-á somente nos dois primeiros níveis de controle (nível estação de trabalho e nível célula). Devido à complexidade das decisões a serem tomadas nos níveis superiores, esta tarefa caberá aos usuários do sistema, que com base nas informações recebidas dos dois primeiros níveis, tomarão as decisões para o gerenciamento do sistema. Deve-se lembrar que uma automação dos dois níveis superiores de decisão podem ser feitas por softwares de ERP (no nível empresa) e software de CAPP (no nível planta).

3.2 Metodologia PFS/MFG

Devido a dimensão e complexidade de um sistema de controle, deve-se procurar utilizar uma metodologia de implantação que evolua de um modelo simples e gradativamente se detalhe até chegar a um nível em que possa ser efetivamente implantado no sistema. Nesse contexto, as representações de Sistemas a Eventos Discretos (SED) através de grafos é de grande valia, pois permitem uma melhor visualização dos fluxos de controle. As Redes de Petri se destacam nesse propósito. Como variações das Redes de Petri para aplicações específicas em SED tem-se a metodologia PFS/MFG, que permite uma modelagem inicial do sistema utilizando uma linguagem natural (de fácil compreensão) e ao decorrer do projeto se detalha até atingir um nível de detalhamento suficiente para a implantação do sistema. Nos próximos itens se apresentarão de forma sucinta os conceitos de PFS e MFG. Uma abordagem mais detalhada desses tópicos pode ser encontrada na referência Miyagi.

3.2.1 Production Flow Schema (PFS)

O projeto de sistemas de controle de SED (Sistemas a Eventos Discretos) através da descrição de estados e eventos advinda das Redes de Petri podem ser hierarquicamente organizados. Os eventos mais gerais podem ser tratados como macro-eventos.

Tratando-se esse projeto de SED de forma hierarquizada, pode-se adotar uma abordagem top-down na análise do sistema, partindo-se de um nível macro para um nível detalhado.

O PFS (Production Flow Schema) é uma técnica para projeto de SED que permite atingir esses objetivos. Inicialmente se considera a divisão do sistema em um pequeno número de partes discretas, para que a identificação dessas partes seja de fácil compreensão. Gradativamente se detalha o modelo, passando a um nível de descrição maior.

Os elementos estruturais fundamentais do PFS são:

- Atividades: corresponde aos elementos ativos, responsáveis pela produção, transporte e modificação dos itens;
- Distribuidores: correspondem aos elementos passivos, que são capazes de armazenar, permanecer em certo estado e tornar visíveis os itens;
- Arcos: representam a relação entre os elementos anteriores, indicando uma conexão lógica, proximidade física, direito de acesso e conexões diretas. Representa uma relação lógica, abstrata entre os componentes, não sendo necessariamente um componente real do sistema.

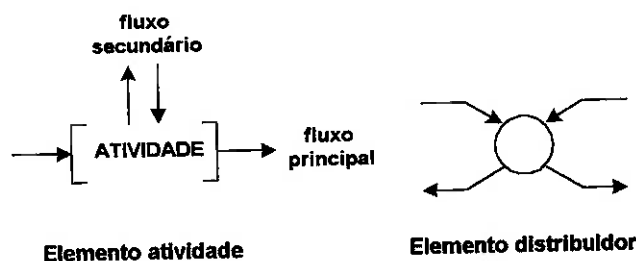


Figura 25: Elementos estruturais do PFS.

O exemplo a seguir mostra uma aplicação da modelagem PFS no início da descrição de um sistema de uma célula de manufatura. Esse exemplo foi extraído da referência Miyagi.

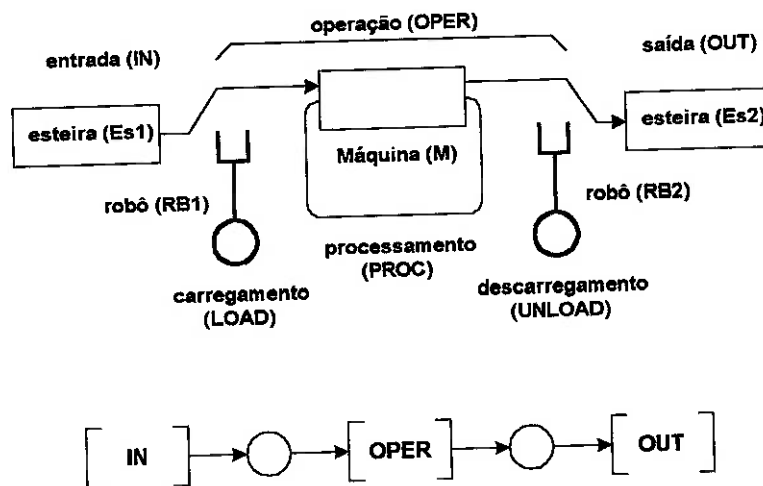


Figura 26: Exemplo de modelagem utilizando PFS.

Como observado no exemplo anterior, a grande vantagem do PFS em relação as Redes de Petri convencionais está na possibilidade de se inserir inscrições em linguagem natural em seus elementos, permitindo uma descrição estruturada mais clara do sistema.

A partir dessa técnica, vários recursos para o controle de fluxo são inseridos no modelo PFS, de forma a permitir a sua conversão num MFG (Mark Flow Graph).

3.2.2 Mark Flow Graph (MFG)

Considerando-se os SEDs (Sistemas a Eventos Discretos) como redes evento-condição, o sistema terá como característica o assíncronismo, a sequencialização o paralelismo e a concorrência. O seu comportamento dinâmico dependerá dessa relação entre as condições e os eventos a ela associados. Uma das formas de se representar esses sistemas é através do

MFG (Mark Flow Graph), que é um grafo derivado das Redes de Petri especificamente estruturado para a representação de sistemas discreto.

Os elementos estruturais do MFG são:

- **Box:** indica uma condição;
- **Transição:** indica um evento;
- **Arco orientado:** indica a relação entre as condições e os eventos. É representado por uma seta;
- **Marca:** indica a manutenção de uma condição, sendo representada por um ponto negro no interior do Box;
- **Porta (Gate):** habilita ou inibe a ocorrência de um evento (transição), podendo ser uma porta habilitadora ou inibidora. A porta habilitadora é representada por uma linha com um círculo negro na extremidade conectada a transição. A porta inibidora é representada por uma linha com um círculo branco na extremidade conectada a transição. As portas podem ser de origem interna ou externa.
- **Arco de sinal de saída:** envia um sinal binário do Box para os dispositivos externos do grafo. É representado por uma linha que conecta esses dois elementos. Quando o Box estiver marcado, o sinal é “1”, quando não estiver o sinal será “0”.

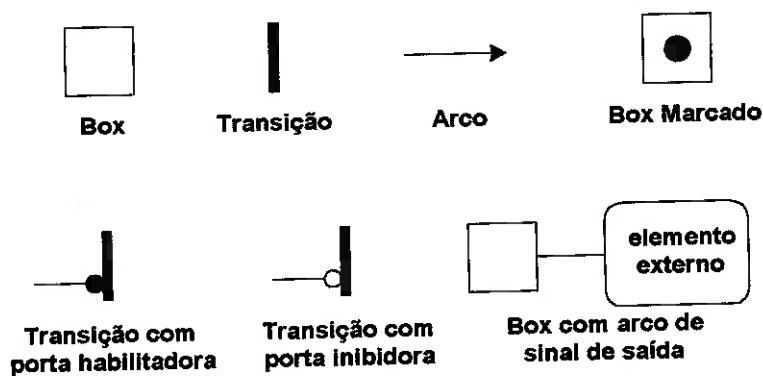


Figura 27: Elementos estruturais do MFG.

O exemplo a seguir representa o MFG do exemplo da Figura 26.

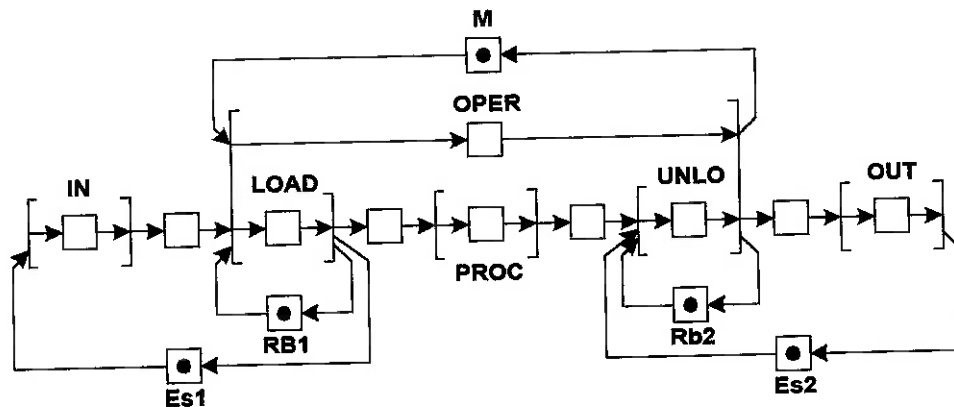


Figura 28: Modelo MFG do exemplo de Figura 26.

Considerando-se os elementos anteriormente descritos, as tarefas de um SED são descritas pelos Boxes que enviam sinais de estado. O início e o término das tarefas representadas pelas transições. As portas de transição (inibidoras ou habilitadoras) podem descrever os elementos externos, possibilitando a representação das relação entre o controlador e o objeto de controle.

A utilização da metodologia PFS/MFG no projeto de sistemas de controle consiste no refinamento gradativo do modelo PFS. O processo de projeto é baseado na substituição das atividades e distribuidores da rede PFS por uma rede (PFS ou MFG), visando aumentar o grau de descrição do modelo. Pode-se assim obter representações gerais e detalhadas do sistema, de forma que cada uma dessas representações satisfaça aos propósitos das pessoas ligadas ao projeto.

3.3 Metodologia de Conversão MFG-Ladder

O MFG permite uma modelagem do sistema e do controle através de grafos, o que facilita a sua compreensão e desenvolvimento. Mas, a programação do CLP não pode ser feita em MFG. Assim, houve a necessidade de se criar uma metodologia que convertesse os modelos MFG em Ladder, a fim de permitir a programação do CLP Allen-Bradley.

O diagrama abaixo explica, através de um exemplo genérico, como é feita essa conversão.

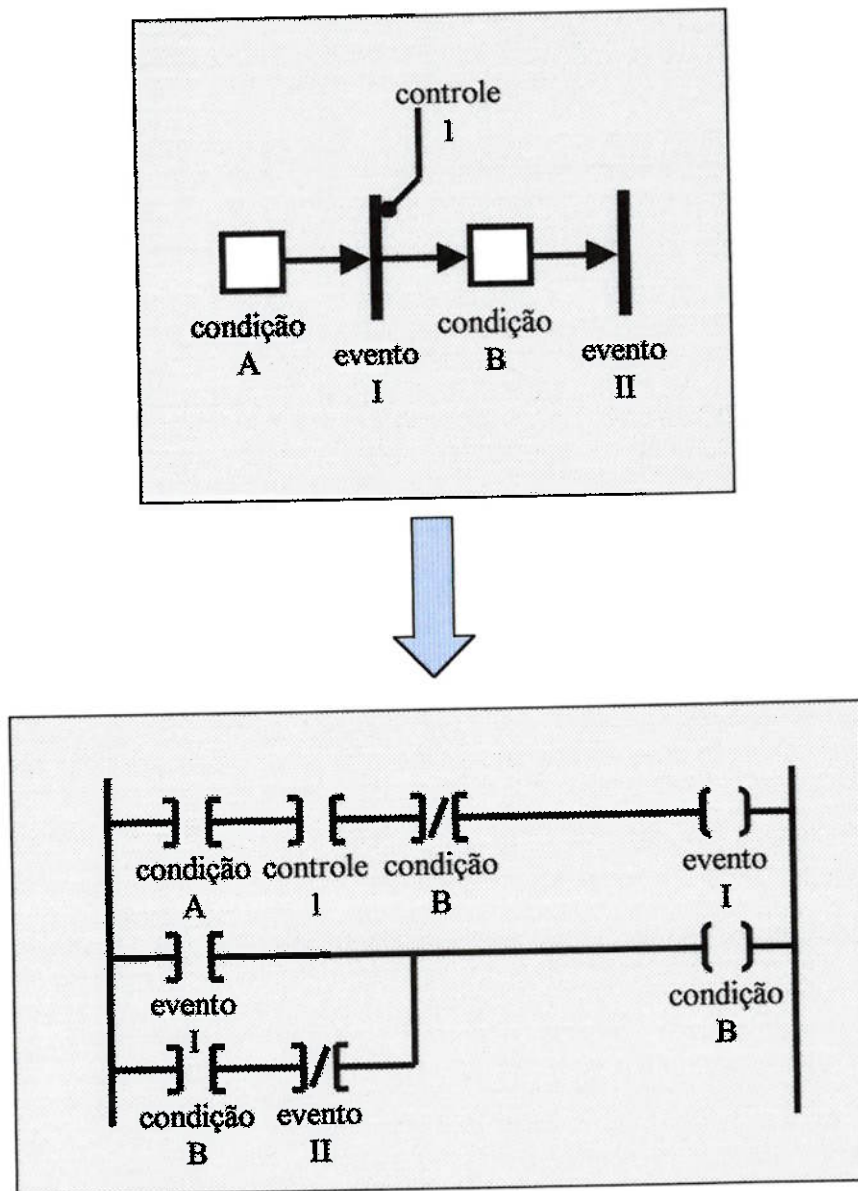


Figura 29: Diagrama de conversão MFG-Ladder

PARTE II - ESTRUTURAÇÃO DO PROJETO

4. OBJETIVOS DO PROJETO

Implantação de sistema de controle para a célula de mistura do CIS-Poli, visando basicamente os níveis Estação de Trabalho e Célula (vide item Definição dos Níveis de Controle). Esse sistema deve ser flexível em seu controle. Por flexibilidade entende-se a possibilidade de alteração fácil das receitas produzidas na célula, bem como a possibilidade de uma maior intervenção humana no processo (para fins de manutenção e ajustes).

Para atingir tais objetivos, desenvolveram-se programas para o PLC que controla a célula de fabricação e implantou-se um sistema SCADA (Supervisory Control and Data Acquisition) para monitorar a célula e permitir intervenções dos usuários no sistema através de uma interface amigável.

5. INSTALAÇÕES E EQUIPAMENTOS

A célula de fabricação de detergente do complexo CIS-Poli esta instalada nas dependências do Departamento de Engenharia Mecânica da Escola Politécnica da USP. Compõe-se basicamente da estrutura produtiva e da sala de controle, onde também se localiza uma sala de reuniões e sala de apresentações.

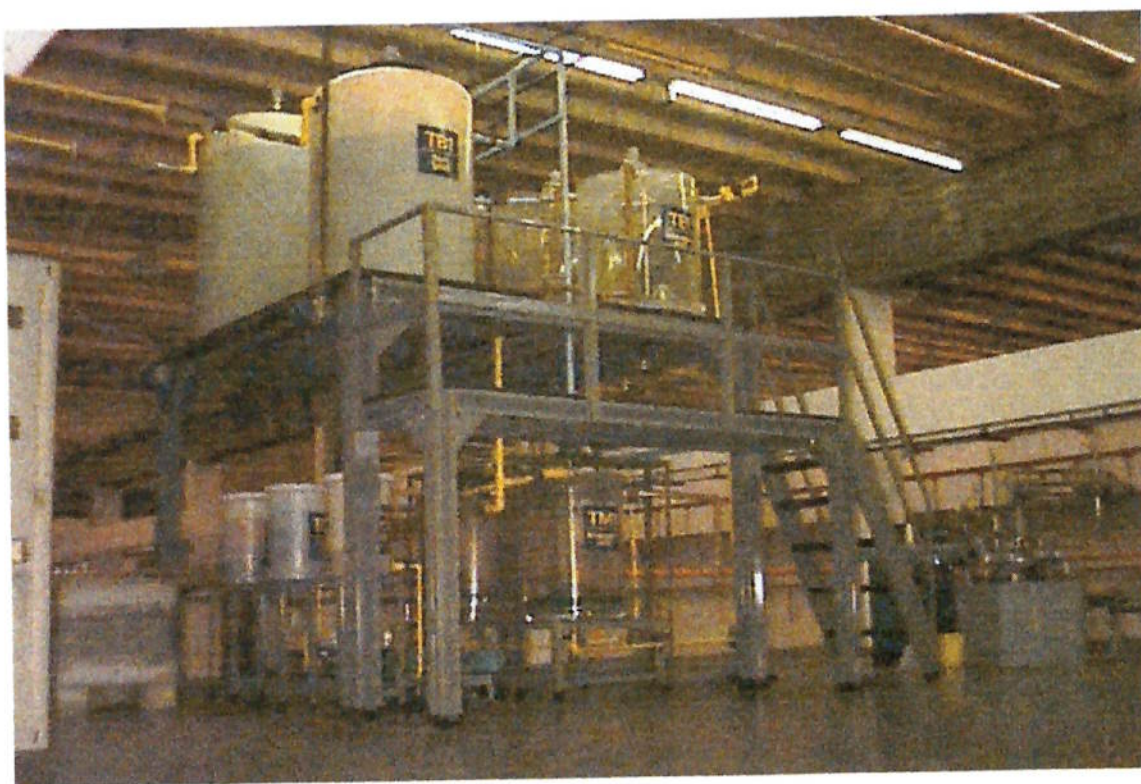


Figura 30: Instalações do CIS-POLI

Os equipamentos que compõe a estrutura produtiva da célula são os seguintes:

- tanque de estoque de base (TB1) com capacidade de 1200 litros;
- tanque de estoque de base (TB2) com capacidade de 1500 litros;
- 3 tanques de corante (TC1, TC2, TC3);

- tanque de mistura (TM1) com capacidade de 200 litros, provido de agitador mecânico acionados por motores elétricos;
- tanque de mistura (TM2) com capacidade de 400 litros; provido de agitador mecânico acionados por motores elétricos;
- tanque de estoque final (TE1) com capacidade de 400 litros;
- tanque de estoque final (TE2) com capacidade de 600 litros;
- tanque de estoque final (TE3) com capacidade de 800 litros;
- tubulações conectando os tanques entre si e os tanques e o meio externo;
- estação de tratamento de efluentes (tanque com oxigenação);
- válvulas solenóides utilizadas no controle do fluxo de líquidos entre os tanques e o meio externo;
- válvulas proporcionais, utilizadas no controle do nível dos tanques de base;
- 3 bombas dosadoras de embolo com os respectivos motores de acionamento, associadas aos tanques de corante;
- 2 bombas centrífugas para o recalque do líquido dos tanques de mistura para os de estoque;
- sensores analógicos de nível (volume) em todos os tanques da tubulação;
- sensores de pulso associados as bombas dosadoras;
- CLP da Allen-Bradley, modelo PLC-5-15/B;
- microcomputador PC 386 para a programação do CLP;
- microcomputador PC-Pentium 166MHz para o controle e supervisão da célula (SCADA);
- rede de comunicação serial entre o PC e o CLP.

6. DISPOSITIVOS DE CONTROLE

Os dispositivos de controle da célula podem ser divididos em sensores, atuadores e controladores.

Os sensores presentes na célula são:

- sensores analógicos de nível capacitivo e pressão, presentes em cada um dos tanques;

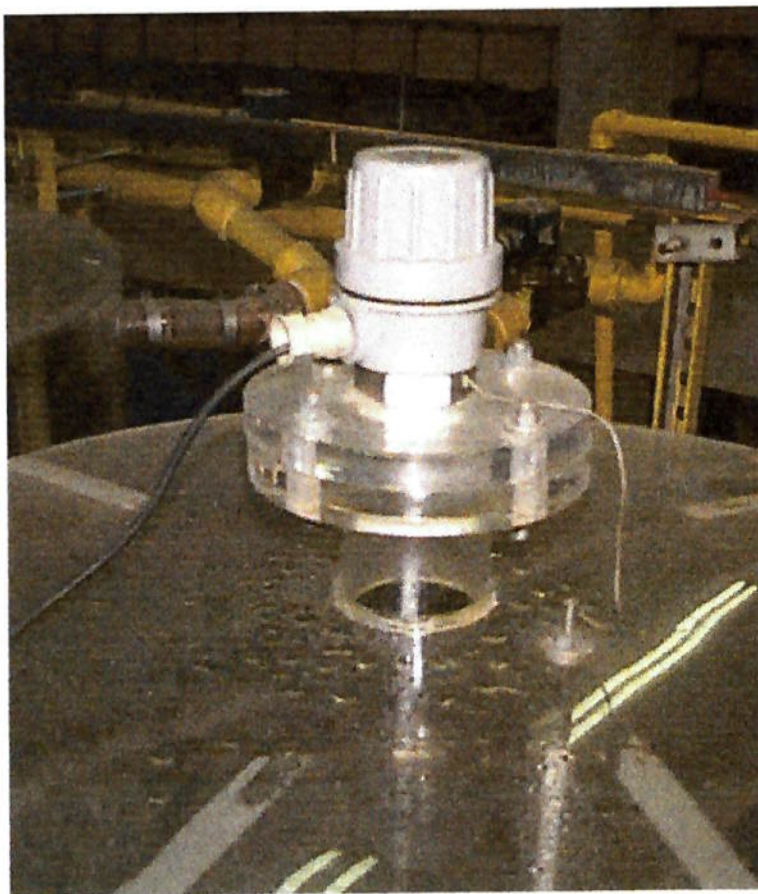


Figura 31: Sensor de nível capacitivo

Fabricante: SITRON

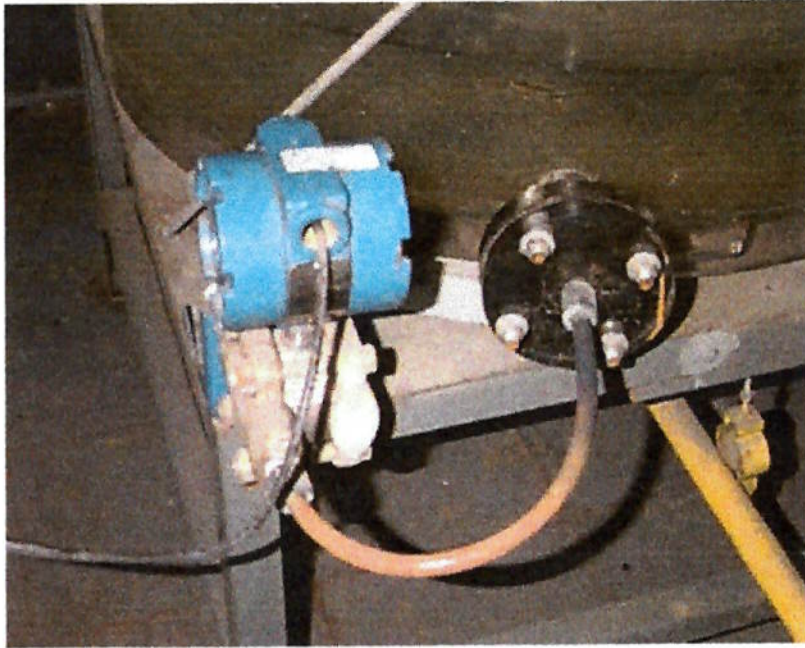


Figura 32: Sensor de nível de pressão

Fabricante: TRANSMITEL

- sensores de pulso associados as bombas dosadoras.

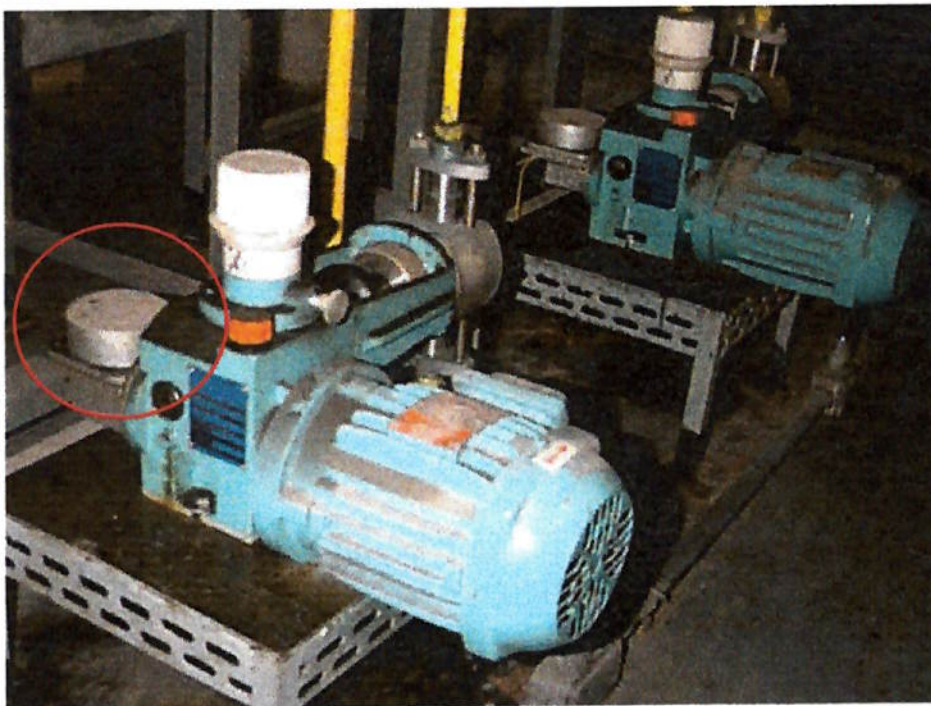


Figura 33: Sensor de pulso

Os atuadores são:

- válvulas solenóides utilizadas no controle do fluxo de líquidos entre os tanques e o meio externo;



Figura 34: Válvula solenóide

Fabricante: ASCO

Tensão: 110V

Potência: 15.4W

Pressão: 9Kg/cm²

- bombas dosadoras de embolo com os respectivos motores de acionamento, associadas aos tanques de corante;



Figura 35: Bomba dosadora

Fabricante: ALFA LAVAL

Tensão: 220V

Rotação: 3400 rpm

Potência: 0,75CV

- bombas centrífugas com os respectivos motores para o recalque do líquido dos tanques de mistura para os de estoque;



Figura 36: Bomba centrífuga

Fabricante: SCHNEIDER

Tensão: 220V

Rotação: 3470 rpm

Potência: 0,5CV

- agitadores mecânicos dos tanques de mistura, associados aos respectivos motores de acionamento.

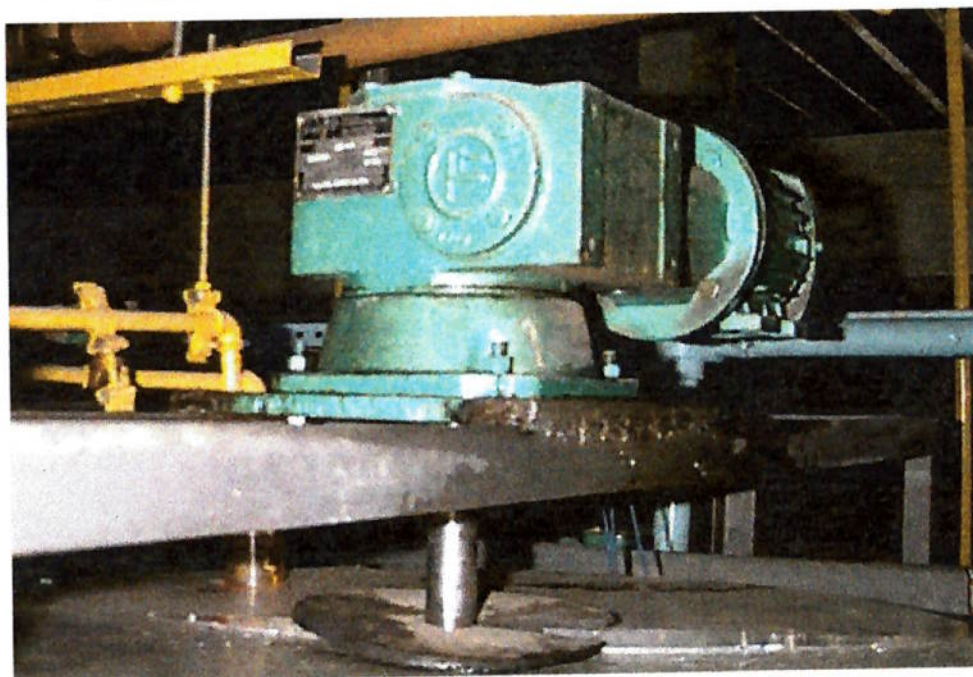


Figura 37: Agitador

Motor

Fabricante: WEG

Tensão: 220V

Rotação: 1710 rpm

Potência: 0,75CV

Redutor

Fabricante: TRANSMOTÉCNICA

Redução: 1:30

Os controladores são:

- CLP da Allen-Bradley, modelo PLC-5-15/B;

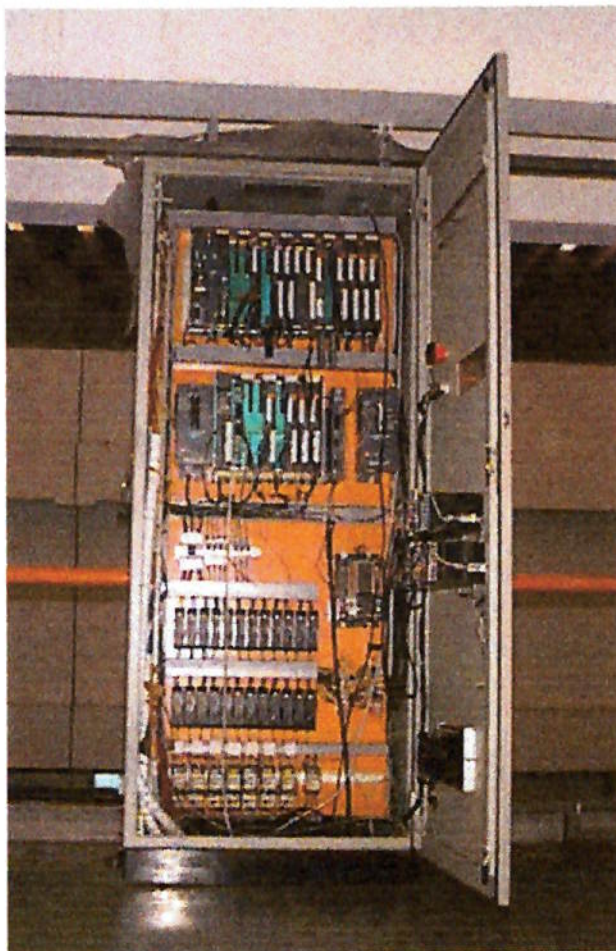


Figura 38: CLP Allen Bradley

- microcomputador PC-Pentium 166MHz para o controle e supervisão da célula (SCADA).

Com os sensores, atuadores e controladores anteriormente citados pode-se implementar um sistema de controle em malha fechada.

7. DEFINIÇÃO DOS NÍVEIS DE CONTROLE

De acordo com o modelo GRAI apresentado no Capítulo 3.1 da Parte I, no sistema de controle do CIS-Poli têm-se os seguintes níveis de controle:

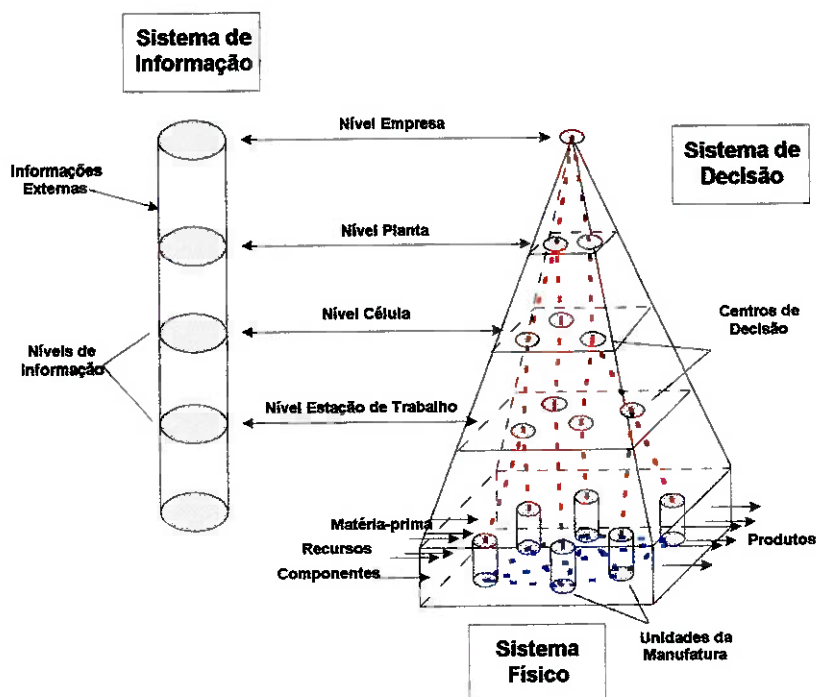


Figura 39: Modelo GRAI adotado utilizado no projeto.

Nesse projeto, atuaremos somente nos níveis Célula e Estação de Trabalho. Nota-se na figura que para cada um dos níveis há um sistema de informação/controlado associado. Deve-se observar que o CIS-Poli possui estrutura para receber os demais níveis de controle necessários para uma total automação do sistema produtivo, ficando esse tarefa para trabalhos posteriores.

8. ESTRUTURAÇÃO DO SISTEMA DE CONTROLE

Utilizou-se a metodologia PFS/MFG para a estruturação do sistema de controle a ser implementado.

Devido a necessidade de flexibilidade requerida para o sistema de controle, estruturou-se a solução de modo que o controle no nível Estação de Trabalho (realizado pelo CLP) atue em operações elementares da célula, deixando a tarefa de controle global da célula a cargo do SCADA (sistema de controle do nível Célula). A Figura 40 esquematiza a estrutura proposta do sistema, com as funções elementares implantadas no CLP e a tarefa de controle e supervisão realizada pelo SCADA.

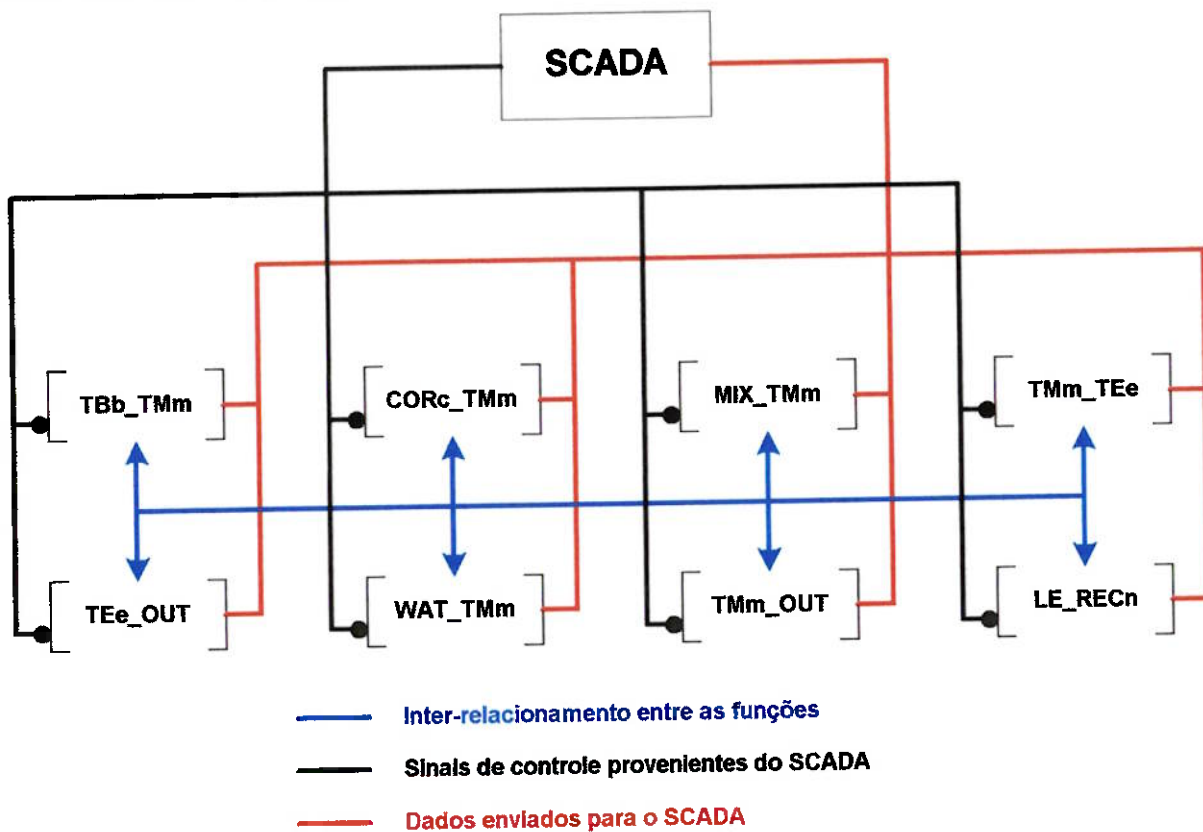


Figura 40: Estrutura proposta de controle da célula: Funções elementares implantadas no controladas e supervisionadas pelo SCADA.

As funções elementares são tratadas aqui de forma genérica, isto é, não se especifica a que tanque ou corante de referem. Assim, a função **TBb_TMm** na verdade representa 4 funções diferentes a serem implementadas, as funções: **TB1_TM1**, **TB1_TM2**, **TB2_TM1**, **TB2_TM2**. Mais informações são dadas nas descrição apresentadas a seguir:

- **TBb_TMm**: função que passa “X” litros de líquido o tanque de base “b” para o tanque de mistura “m” (tanques de base podem ser “b”=1,2; tanque de mistura podem ser “m”=1,2; “X” é um valor armazenado na memória do CLP que pode ser alterado pelo SCADA).
- **CORc_TMm**: função que coloca “Y” pulsos de corante do tanque de corante “c” no tanque de mistura “m” (o tanque de corante pode ser “c”=1,2,3; “Y” é um valor armazenado na memória do CLP que pode ser alterado pelo SCADA). Obs.: a quantidade de corante é medida em pulsos da bomba dosadora.
- **MIX_TMm**: função que mistura o conteúdo do tanque de mistura “m” por “Z” segundos. (“Z” é um valor armazenado na memória do CLP que pode ser alterado pelo SCADA).
- **TMm_TEe**: função que passa “W” litros de líquido do tanque de mistura “m” para o tanque de estoque “e”. (“W” é um valor armazenado na memória do CLP que pode ser alterado pelo SCADA).
- **TEe_OUT**: função que esvazia o tanque de estoque “e”, isto é, passa todo seu conteúdo para o esgoto.
- **TMm_OUT**: função que esvazia o tanque de mistura “m”, isto é, passa todo seu conteúdo para o esgoto.

- **WAT_TM_m**: função que passa “Q” litros de água da rede externa para o tanque de mistura “m”.
- **LE_REC_n**: lê e executa a receita (sequência de operações com seus devidos parâmetros). “n” pode ser 1 ou 2, já que pode-se executar duas receitas simultaneamente.

As funções implementadas no CLP se relacionam entre si, mas não estabelecem uma sequência de operação (primeiro encher, depois misturar, depois estocar, ...). O relacionamento ocorre porque as funções concorrem entre si pelos mesmos recursos existentes na planta. Por exemplo, não se pode passar líquido do tanque de base 1 (TB1) para o tanque de mistura 1 (TM1) e ao mesmo tempo misturar o tanque TM1, pois agitando a mistura não se tem precisão de leitura de nível. Nesse caso, considera-se que essas duas funções concorrem pelo mesmo recurso, o tanque TM1.

Nos próximos itens são apresentados os modelos das funções descritas anteriormente, concebidos utilizando a metodologia PFS/MFG.

PARTE III – IMPLANTAÇÃO DO PROJETO

9. PROJETO DE SOFTWARE

Seguindo o modelo GRAI que orientou o trabalho, o projeto de software foi dividido em duas partes: o software do CLP (nível Estação de Trabalho) e o do SCADA (nível Célula). Essa divisão foi feita pois os sistemas tem funções hierárquicas diferentes, portanto o desenvolvimento do controle para cada um desses níveis deve ser específico.

O diagrama abaixo ilustra a divisão de funções de cada um dos níveis citados:

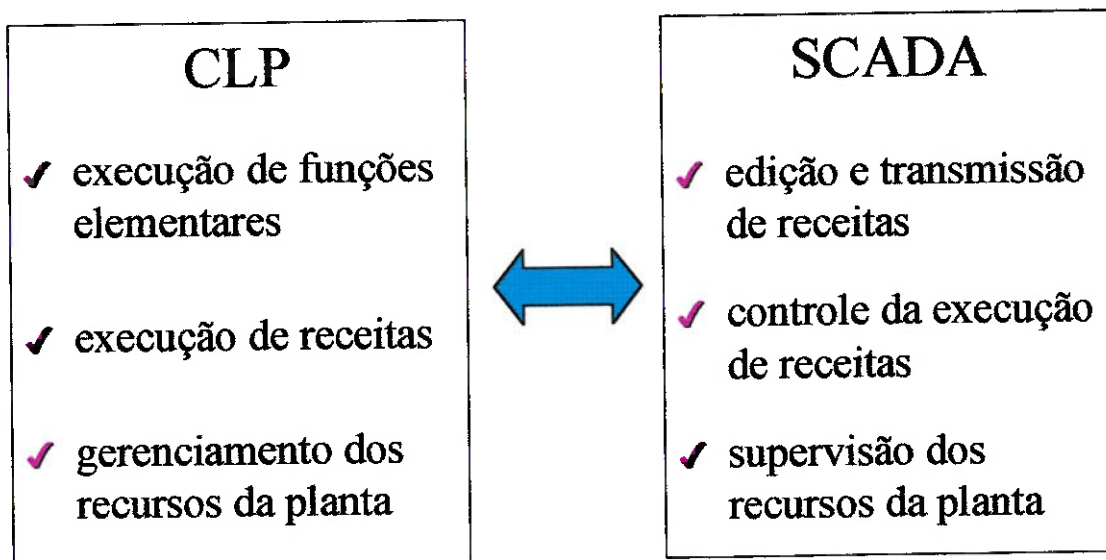


Figura 41: Divisão de funções entre os níveis de controle.

9.1 Sistema de Controle do Nível Estação de Trabalho (CLP)

Devido a complexidade do software de controle a ser implementado no CLP, utilizou-se a metodologia PFS/MFG na concepção das rotinas de controle. Por ter uma sintaxe gráfica e de fácil compreensão, o MFG facilita o desenvolvimento das soluções, quando comparado com a linguagem Ladder (de difícil compreensão).

Nos itens a seguir, serão apresentados o Mapa da memória do CLP e as Funções e Rotinas implementadas no CLP, documentando, assim, o desenvolvimento do software de controle do CLP.

9.1.1 Mapas de memória do CLP

Mapa de memória do PLC-5/15 do CIS-Poli

Conteúdo

Este arquivo XLS tem como função documentar a utilização de memória do PLC-5/15 do CIS-Poli. Cada pasta desse arquivo representa um arquivo da memória do PLC-5/15. Assim, temos as pastas O, I, S, B3, S4, C5, ...
Apresenta-se abaixo a descrição de cada uma das pastas.

Pasta	Descrição
O	Arquivo de bits que representa logicamente os dispositivos de saída digitais presentes na planta. Através dele pode-se controlar o estado das válvulas, motores e bombas.
I	Arquivo de bits que representa logicamente os dispositivos de entrada digitais presentes na planta. Através dele pode-se verificar o estado dos sensores digitais presentes na planta.
S	Arquivo de bits que apresenta os flags de estado do PLC.
B3	Arquivo de bits que pode ser visto como variáveis binárias a ser utilizadas na programação do PLC. Representam as variáveis de controle das funções implementadas no PLC, sempre em dupla: uma liga a função e a outra sinaliza o fim da execução da função.
T4	Arquivo dos parâmetros dos temporizadores utilizados na aplicação.
C5	Arquivo dos parâmetros dos contadores utilizados na aplicação.
R6	Arquivo das palavras de controle de funções da programação da linguagem Ladder da Allen-Bradley. Utilizados para o controle das funções de manipulação de arquivos de memória: FAL e FFU.
N7	Arquivos de inteiros que armazenam os volumes lidos e desejados para os tanques da planta.
F8	Arquivo de números reais que armazenam os fatores de conversão entre os valores de nível (de líquido) lidos nos sensores e o volume em litros nos tanques.
B9	Arquivo de bits utilizado para variáveis internas das funções implementadas no PLC. As variáveis de controle estão no arquivo B3.
B10	Arquivo de bits que armazena o estado inicial do arquivo B3. Esse arquivo é utilizado na função Inicializa para colocar o arquivo B3 em seu estado inicial.
B11	Arquivo de bits que armazena o estado inicial do arquivo B9. Esse arquivo é utilizado na função Inicializa para colocar o arquivo B9 em seu estado inicial.
B12	Arquivo de bits que armazena o estado das variáveis de inicializa (função que inicializa o estado do sistema).
N13	Arquivo de inteiros que armazenam as operações (em sua sequência de execução) da receita 1 a ser executada na planta de mistura. Esta intimamente ligada aos arquivos N14 e N15.
N14	Arquivo de inteiros que armazenam os parâmetros de volume das operações da receita 1 a ser executada na planta de mistura. Esta intimamente ligada aos arquivos N13.
N15	Arquivo de inteiros que armazenam os parâmetros de tempo das operações da receita 1 a ser executada na planta de mistura. Esta intimamente ligada aos arquivos N13.
N16	Arquivo de inteiros que armazenam as operações (em sua sequência de execução) da receita 2 a ser executada na planta de mistura. Esta intimamente ligada aos arquivos N17 e N18.
N17	Arquivo de inteiros que armazenam os parâmetros de volume das operações da receita 2 a ser executada na planta de mistura. Esta intimamente ligada aos arquivos N16.
N18	Arquivo de inteiros que armazenam os parâmetros de tempo das operações da receita 2 a ser executada na planta de mistura. Esta intimamente ligada aos arquivos N16.
T19	Arquivo do estado inicial do arquivo T4 (temporizadores). O arquivo T19 é utilizado no procedimento de inicialização do sistema.
C20	Arquivo do estado inicial do arquivo C5 (contadores). O arquivo C20 é utilizado no procedimento de inicialização do sistema.
R21	Arquivo do estado inicial do arquivo R6 (palavras de controle). O arquivo R21 é utilizado no procedimento de inicialização do sistema.

O - Arquivo de Output

Arquivo de bits que representa logicamente os dispositivos de saída digitais presentes na planta. Através dele pode-se controlar o estado das válvulas, motores e bombas.

Address	0	1	2	3	4	5	6	7
O:000/00								
O:000/10								
O:001/00								
O:001/10								
O:002/00								
O:002/10								
O:003/00								
O:003/10								
O:004/00								
O:004/10								
O:005/00								
O:005/10								
O:006/00								
O:006/10	Válvula TB1 TM1	Válvula TB2 TM1	Válvula COR1 TM1	Válvula COR2 TM1	Válvula COR3 TM1	Válvula WAT TM1	Válvula TM1 OUT	Válvula TM1 TE1
O:007/00								
O:007/10	Válvula TM1 TE2	Válvula TM1 TE3	Válvula TB1 TM2	Válvula TB2 TM2	Válvula COR1 TM2	Válvula COR2 TM2	Válvula COR3 TM2	Válvula WAT TM2
O:008/00								
O:008/10								
O:009/00								
O:009/10								
O:010/00								
O:010/10								
O:011/00								
O:011/10	Válvula TM2 OUT	Válvula TM2 TE1	Válvula TM2 TE2	Válvula TM2 TE3	Válvula WAT TB1	Válvula TB1 OUT	Válvula WAT TB2	Válvula TB2 OUT
O:012/00								
O:012/10	Válvula WAT TE1	Válvula TE1 OUT	Válvula WAT TE2	Válvula TE2 OUT	Válvula WAT TE3	Válvula TE3 OUT		
O:013/00								
O:013/10	Motor MIXER 1	Motor MIXER 2	Motor dosadora 1	Motor dosadora 2	Motor dosadora 3	Bomba recalque 1	Bomba recalque 2	

I - Arquivo de Input

Arquivo de bits que representa logicamente os dispositivos de entrada digitais presentes na planta. Através dele pode-se verificar o estado dos sensores digitais presentes na planta.

Address	0	1	2	3	4	5	6	7
I:010/00								
I:010/10								
I:011/00								
I:011/10								
I:012/00								
I:012/10								
I:013/00				Pulso da dosadora 1	Pulso da dosadora 2	Pulso da dosadora 3		
I:013/10								
I:014/00								
I:014/10								

S - Arquivo de Status

Arquivo de bits que apresenta os flags de estado do PLC.

Address	0	1	2	3	4
S:00					
S:05					
S:10		ver manual do CLP-5			
S:15					
S:20					
S:25					
S:30					

B3 - Arquivo de Bits (Variáveis principais)

Arquivo de bits que pode ser visto como variáveis binárias a ser utilizadas na programação do PLC. Representam as variáveis de controle das funções implementadas no PLC, sempre em dupla: uma liga a função e a outra sinaliza o fim da execução da função.

Address	0	1	2	3	4
B3/00	Controle: Botão ligar sistema.	Controle: Botão emergência.	Controle: executar TB1_TM1.	Flag: operação TB1_TM1 finalizada.	Controle: executar TB1_TM2.
B3/05	Flag: operação TB1_TM2 finalizada.	Controle: executar TB2_TM1.	Flag: operação TB2_TM1 finalizada.	Controle: executar TB2_TM2.	Flag: operação TB2_TM2 finalizada.
B3/10	Controle: executar TM1_TE1.	Flag: operação TM1_TE1 finalizada.	Controle: executar TM1_TE2.	Flag: operação TM1_TE2 finalizada.	Controle: executar TM1_TE3.
B3/15	Flag: operação TM1_TE3 finalizada.	Controle: executar TM2_TE1.	Flag: operação TM2_TE1 finalizada.	Controle: executar TM2_TE2.	Flag: operação TM2_TE2 finalizada.
B3/20	Controle: executar TM2_TE3.	Flag: operação TM2_TE3 finalizada.	Controle: executar COR1_TM1.	Flag: operação COR1_TM1 finalizada.	Controle: executar COR1_TM2.
B3/25	Flag: operação COR1_TM2 finalizada.	Controle: executar COR2_TM1.	Flag: operação COR2_TM1 finalizada.	Controle: executar COR2_TM2.	Flag: operação COR2_TM2 finalizada.
B3/30	Controle: executar COR3_TM1.	Flag: operação COR3_TM1 finalizada.	Controle: executar COR3_TM2.	Flag: operação COR3_TM2 finalizada.	Controle: executar MIX_TM1.
B3/35	Flag: operação MIX_TM1 finalizada.	Controle: executar MIX_TM2.	Flag: operação MIX_TM2 finalizada.	Controle: executar TM1_OUT.	Flag: operação TM1_OUT finalizada.
B3/40	Controle: executar TM2_OUT.	Flag: operação TM2_OUT finalizada.	Controle: executar TE1_OUT.	Flag: operação TE1_OUT finalizada.	Controle: executar TE2_OUT.
B3/45	Flag: operação TE2_OUT finalizada.	Controle: executar TE3_OUT.	Flag: operação TE3_OUT finalizada.	Controle: executar WAT_TM1.	Flag: operação WAT_TM1 finalizada.
B3/50	Controle: executar WAT_TM2.	Flag: operação WAT_TM2 finalizada.	Controle: executar WAT_TE1.	Flag: operação WAT_TE1 finalizada.	Controle: executar WAT_TE2.
B3/55	Flag: operação WAT_TE2 finalizada.	Controle: executar WAT_TE3.	Flag: operação WAT_TE3 finalizada.	Controle: executar LE_REC1.	Flag: operação LE_REC1 finalizada.
B3/60	Controle: executar LE_REC2.	Flag: operação LE_REC2 finalizada.	"sem uso"	"sem uso"	"sem uso"
B3/65	"sem uso"	"sem uso"	"sem uso"	Controle: parar execução de TB1_TM1.	Controle: parar execução de TB1_TM2.
B3/70	Controle: parar execução de TB2_TM1.	Controle: parar execução de TB2_TM2.	Controle: parar execução de TB2_TM2.	Controle: parar execução de TM1_TE1.	Controle: parar execução de TM1_TE2.
B3/75	Controle: parar execução de TM1_TE3.	Controle: parar execução de TM2_TE1.	Controle: parar execução de TM2_TE2.	Controle: parar execução de TM2_TE3.	Controle: parar execução de COR1_TM1.

B3/80	Controle: parar execução de COR1_TM2.	Controle: parar execução de COR2_TM1.	Controle: parar execução de COR2_TM2.	Controle: parar execução de COR3_TM1.	Controle: parar execução de COR3_TM2.
B3/85	Controle: parar execução de MIX_TM1.	Controle: parar execução de MIX_TM2.	Controle: parar execução de TM1_OUT.	Controle: parar execução de TM2_OUT.	Controle: parar execução de TE1_OUT.
B3/90	Controle: parar execução de TE2_OUT.	Controle: parar execução de TE3_OUT.	Controle: parar execução de TE3_OUT.	Controle: parar execução de WAT_TM1.	Controle: parar execução de WAT_TM2.
B3/95	Controle: parar execução de WAT_TE1.	Controle: parar execução de WAT_TE2.	Controle: parar execução de WAT_TE3.		

T4 - Arquivo de Temporizador

Arquivo dos parâmetros dos temporizadores utilizados na aplicação.

Address	EN	TT	DN	BASE	PRE	ACC	Descrição
T4:00							Usado em MIX TM1.
T4:01							Usado em MIX TM2.
T4:02							Usado em TE1 OUT.
T4:03							Usado em TE2 OUT.
T4:04							Usado em TE3 OUT.
T4:05							Usado em COR1 TM1.
T4:06							Usado em COR1 TM2.
T4:07							Usado em COR2 TM1.
T4:08							Usado em COR2 TM2.
T4:09							Usado em COR3 TM1.
T4:10							Usado em COR3 TM2.
T4:11							Usado em LE REC1.
T4:12							Usado em LE REC2.

C5 - Arquivo de Contador

Arquivo dos parâmetros dos contadores utilizados na aplicação.

Address	CU	CD	DN	OV	UN	PRE	ACC	Descrição
C5:00								Usado em COR1 TM1.
C5:01								Usado em COR1 TM2.
C5:02								Usado em COR2 TM1.
C5:03								Usado em COR2 TM2.
C5:04								Usado em COR3 TM1.
C5:05								Usado em COR3 TM2.

R6 - Arquivo de Controle

Arquivo das palavras de controle de funções da programação da linguagem Ladder da Allen-Bradley.
Utilizados para o controle das funções de manipulação de arquivos de memória: FAL e FFU.

Address	EN	EU	DN	EM	ER	UL	IN	FD	LEN	Descrição
R6:00										Usado em LE_VOLUME.
R6:01										Usado em LE_VOLUME.
R6:02										Usado em LE_REC1.
R6:03										Usado em LE_REC1.
R6:04										Usado em LE_REC2.
R6:05										Usado em LE_REC2.
R6:06										Usado em LE_REC2.
R6:07										Usado em LE_REC2.

N7 - Arquivo de Inteiros

Arquivos de inteiros que armazenam os volumes lidos e desejados para os tanques da planta.

Address	0	1	2	3	4
N7:00	Volume de TM1 (em litros)	Volume de TM2 (em litros)	utilizado na leitura da placa analógica (volumes)	utilizado na leitura da placa analógica (volumes)	utilizado na leitura da placa analógica (volumes)
N7:05	utilizado na leitura da placa analógica (volumes)	Volume de TB1 (em litros)	Volume de TB2 (em litros)	Volume de TE1 (em litros)	Volume de TE2 (em litros)
N7:10	Volume de TE3 (em litros)	utilizado na leitura da placa analógica (volumes)	utilizado na leitura da placa analógica (volumes)	Palavra 0 de controle da leitura da placa analógica.	Palavra 1 de controle da leitura da placa analógica.
N7:15	Palavra 2 de controle da leitura da placa analógica.	Palavra 3 de controle da leitura da placa analógica.	Palavra 4 de controle da leitura da placa analógica.	Leitura da placa analógica. Palavra 0.	Leitura da placa analógica. Palavra 1.
N7:20	Leitura da placa analógica. Palavra 2.	Leitura da placa analógica. Palavra 3.	Leitura da placa analógica. Palavra 4.	Leitura da placa analógica. Palavra 5.	Leitura da placa analógica. Palavra 6.
N7:25	Leitura da placa analógica. Palavra 7.	Leitura da placa analógica. Palavra 8.	Leitura da placa analógica. Palavra 9.	Leitura da placa analógica. Palavra 10.	Leitura da placa analógica. Palavra 11.
N7:30	Leitura da placa analógica. Palavra 12.	Primeira conversão dos valores lidos na placa analógica. Palavra 0.	Primeira conversão dos valores lidos na placa analógica. Palavra 1.	Primeira conversão dos valores lidos na placa analógica. Palavra 2.	Primeira conversão dos valores lidos na placa analógica. Palavra 3.
N7:35	Primeira conversão dos valores lidos na placa analógica. Palavra 4.	Primeira conversão dos valores lidos na placa analógica. Palavra 5.	Primeira conversão dos valores lidos na placa analógica. Palavra 6.	Primeira conversão dos valores lidos na placa analógica. Palavra 7.	Primeira conversão dos valores lidos na placa analógica. Palavra 8.
N7:40	Primeira conversão dos valores lidos na placa analógica. Palavra 9.	Primeira conversão dos valores lidos na placa analógica. Palavra 10.	Primeira conversão dos valores lidos na placa analógica. Palavra 11.	Primeira conversão dos valores lidos na placa analógica. Palavra 12.	Delta_Volume da função TB1_TM1
N7:45	Volume_Final função TB1_TM1	Delta_Volume da função TB1_TM2	Volume_Final função TB1_TM2	Delta_Volume da função TB2_TM1	Volume_Final função TB2_TM1
N7:50	Delta_Volume da função TB2_TM2	Volume_Final função TB2_TM2	Volume máximo de TB1 (=1200)	Volume máximo de TB2 (=1500)	Volume máximo de TM1 (=200)
N7:55	Volume máximo de TM2 (=400)	Volume máximo de TE1 (=400)	Volume máximo de TE2 (=600)	Volume máximo de TE3 (=900)	Delta_Volume da função WAT_TM1
N7:60	Volume_Final função WAT_TM1	Delta_Volume da função WAT_TM2	Volume_Final função WAT_TM2	Delta_Volume da função WAT_TE1	Volume_Final função WAT_TE1
N7:65	Delta_Volume da função WAT_TE2	Volume_Final função WAT_TE2	Delta_Volume da função WAT_TE3	Volume_Final função WAT_TE3	"sem uso"
N7:70	"sem uso"	"sem uso"	"sem uso"	"sem uso"	Operação atual sendo executada em LE_REC1.
N7:75	Parâmetro volume da operação atual em LE_REC1.	Parâmetro tempo da operação atual em LE_REC1.	Operação atual sendo executada em LE_REC2.	Parâmetro volume da operação atual em LE_REC2.	Parâmetro tempo da operação atual em LE_REC2.

F8 - Arquivo de Flutuantes

Arquivo de números reais que armazena os fatores de conversão entre os valores de nível (de líquido) lidos nos sensores e o volume em litros nos tanques.

Address	0	1	2	3	4
F8:00	Fator de conversão entre a tensão do sensor e o volume no tanque TM1 (=0,0522160)	Fator de conversão entre a tensão do sensor e o volume no tanque TM2 (=0,1044320)	Vazio, devido conversões dos arquivos de leitura dos sensores de volume. (=1)	Vazio, devido conversões dos arquivos de leitura dos sensores de volume. (=1)	Vazio, utilizado nas conversões dos arquivos de leitura dos sensores de volume. (=1)
F8:05	Vazio, utilizado nas conversões dos arquivos de leitura dos sensores de volume. (=1)	Fator de conversão entre a tensão do sensor e o volume no tanque TB1 (=0,3865350)	Fator de conversão entre a tensão do sensor e o volume no tanque TB2 (=0,5493410)	Fator de conversão entre a tensão do sensor e o volume no tanque TE1 (=0,1121142)	Fator de conversão entre a tensão do sensor e o volume no tanque TE2 (=0,2400020)
F8:10	Fator de conversão entre a tensão do sensor e o volume no tanque TB2 (=0,3260510)	Vazio, devido conversões dos arquivos de leitura dos sensores de volume. (=1)	Vazio, devido conversões dos arquivos de leitura dos sensores de volume. (=1)		

B9 - Arquivo de Bits (Variáveis auxiliares)

Arquivo de bits utilizado para variáveis internas das funções implementadas no PLC. As variáveis de controle estão no arquivo B3.

Address	0	1	2	3	4
B9/00	Flag: executando TB1_TM1.	Flag: Vfinal de TB1_TM1 calculado.	Evento 1 de TB1_TM1.	Evento 3 de TB1_TM1.	Flag: TM1 disponível.
B9/05	Flag: válvula TB1_TM1 disponível.	Evento 2 de TB1_TM1.	Evento 1 de TB1_TM2.	Flag: executando TB1_TM2.	Flag: Vfinal de TB1_TM2 calculado.
B9/10	Evento 3 de TB1_TM2.	Flag: TM2 disponível.	Flag: válvula TB1_TM2 disponível.	Evento 2 de TB1_TM2.	Evento 1 de TB2_TM1.
B9/15	Flag: executando TB2_TM1.	Flag: Vfinal de TB2_TM1 calculado.	Evento 3 de TB2_TM1.	Flag: válvula TB2_TM1 disponível.	Evento 2 de TB2_TM1.
B9/20	Evento 1 de TB2_TM2.	Flag: executando TB2_TM2.	Flag: Vfinal de TB2_TM2 calculado.	Evento 3 de TB2_TM2.	Flag: válvula TB2_TM2 disponível.
B9/25	Evento 2 de TB2_TM2.	Evento 1 de TM1_TE1.	Flag: bomba recalque 1 disponível.	Flag: válvula TM1_TE1 disponível.	Flag: executando TM1_TE1.
B9/30	Evento 2 de TM1_TE1.	Evento 1 de TM1_TE2.	Flag: válvula TM1_TE2 disponível.	Flag: executando TM1_TE2.	Evento 2 de TM1_TE2.
B9/35	Evento 1 de TM1_TE3.	Flag: válvula TM1_TE3 disponível.	Flag: executando TM1_TE3.	Evento 2 de TM1_TE3.	Evento 1 de TM2_TE1.
B9/40	Flag: bomba recalque 2 disponível.	Flag: válvula TM2_TE1 disponível.	Flag: executando TM2_TE1.	Evento 2 de TM2_TE1.	Evento 1 de TM2_TE2.
B9/45	Flag: válvula TM2_TE2 disponível.	Flag: executando TM2_TE2.	Evento 2 de TM2_TE2.	Evento 1 de TM2_TE3.	Flag: válvula TM2_TE3 disponível.
B9/50	Flag: executando TM2_TE3.	Evento 2 de TM2_TE3.	Evento 1 de MIX_TM1.	Flag: executando MIX_TM1.	Evento 2 de MIX_TM1.
B9/55	Evento 1 de MIX_TM2.	Flag: executando MIX_TM2.	Evento 2 de MIX_TM2.	Evento 1 de TE1_OUT.	Flag: válvula TE1_OUT disponível.
B9/60	Flag: executando TE1_OUT.	Flag: checar volume de TE1_OUT.	Evento 3 de TE1_OUT.	Evento 2 de TE1_OUT.	Flag: TE1 disponível.
B9/65	Evento 1 de TE2_OUT.	Flag: TE2 disponível.	Flag: válvula TE2_OUT disponível.	Flag: executando TE2_OUT.	Flag: checar volume de TE2_OUT.
B9/70	Evento 3 de TE2_OUT.	Evento 2 de TE2_OUT.	Evento 1 de TE3_OUT.	Flag: TE3 disponível.	Flag: válvula TE3_OUT disponível.
B9/75	Flag: executando TE3_OUT.	Flag: checar volume de TE3_OUT.	Evento 3 de TE3_OUT.	Evento 2 de TE3_OUT.	Evento 1 de COR1_TM1.
B9/80	Flag: válvula COR1_TM1 disponível.	Flag: executando COR1_TM1.	Flag: inicia contador em COR1_TM1.	Evento 4 de COR1_TM1.	Evento 2 de COR1_TM1.
B9/85	Flag: dosadora/corante 1 disponível.	Evento 3 de COR1_TM1.	Evento 1 de COR1_TM2.	Flag: válvula COR1_TM2 disponível.	Flag: executando COR1_TM2.

B9/90	Flag: inicia contador em COR1_TM2.	Evento 4 de COR1_TM2.	Evento 2 de COR1_TM2.	Evento 3 de COR1_TM2.	Evento 1 de COR2_TM1.
B9/95	Flag: válvula COR2_TM1 disponível.	Flag: executando COR2_TM1.	Flag: inicia contador em COR2_TM1.	Evento 4 de COR2_TM1.	Evento 2 de COR2_TM1.
B9/100	Flag: dosadora/corante 2 disponível.	Evento 3 de COR2_TM1.	Evento 1 de COR2_TM2.	Flag: válvula COR2_TM2 disponível.	Flag: executando COR2_TM2.
B9/105	Flag: inicia contador em COR2_TM2.	Evento 4 de COR2_TM2.	Evento 2 de COR2_TM2.	Evento 3 de COR2_TM2.	Evento 1 de COR3_TM1.
B9/110	Flag: válvula COR3_TM1 disponível.	Flag: executando COR3_TM1.	Flag: inicia contador em COR3_TM1.	Evento 4 de COR3_TM1.	Evento 2 de COR3_TM1.
B9/115	Flag: dosadora/corante 3 disponível.	Evento 3 de COR3_TM1.	Evento 1 de COR3_TM2.	Flag: válvula COR3_TM2 disponível.	Flag: executando COR3_TM2.
B9/120	Flag: inicia contador em COR3_TM2.	Evento 4 de COR3_TM2.	Evento 2 de COR3_TM2.	Evento 3 de COR3_TM2.	Evento 1 de TM1_OUT.
B9/125	Flag: válvula TM1_OUT disponível.	Flag: executando TM1_OUT.	Evento 2 de TM1_OUT.	Evento 1 de TM2_OUT.	Flag: válvula TM2_OUT disponível.
B9/130	Flag: executando TM2_OUT.	Evento 2 de TM2_OUT.	Evento 1 de WAT_TM1.	Flag: executando WAT_TM1.	Flag: Vfinal de WAT_TM1 calculado.
B9/135	Evento 3 de WAT_TM1.	Evento 2 de WAT_TM1.	Flag: válvula WAT_TM1 disponível.	Evento 1 de WAT_TM2.	Flag: executando WAT_TM2.
B9/140	Flag: Vfinal de WAT_TM2 calculado.	Evento 3 de WAT_TM2.	Evento 2 de WAT_TM2.	Flag: válvula WAT_TM2 disponível.	Evento 1 de WAT_TE1.
B9/145	Flag: executando WAT_TE1.	Flag: Vfinal de WAT_TE1 calculado.	Evento 3 de WAT_TE1.	Evento 2 de WAT_TE1.	Flag: válvula WAT_TE1 disponível.
B9/150	Evento 1 de WAT_TE2.	Flag: executando WAT_TE2.	Flag: Vfinal de WAT_TE2 calculado.	Evento 3 de WAT_TE2.	Evento 2 de WAT_TE2.
B9/155	Flag: válvula WAT_TE2 disponível.	Evento 1 de WAT_TE3.	Flag: executando WAT_TE3.	Flag: Vfinal de WAT_TE3 calculado.	Evento 3 de WAT_TE3.
B9/160	Evento 2 de WAT_TE3.	Flag: válvula WAT_TE3 disponível.	Evento 1 de LE_REC1.	Flag: pega 1o da FIFO em LE_REC1.	Evento 6 de LE_REC1.
B9/165	Evento 2 de LE_REC1.	Flag: compara e inicia em LE_REC1.	Evento 3 de LE_REC1.	Evento 4 de LE_REC1.	"sem uso"
B9/170	Flag: oper. 0 (Fim da receita) em LE_REC1.	Flag: oper. 1 (TB1_TM1) em LE_REC1.	Flag: oper. 2 (TB1_TM2) em LE_REC1.	Flag: oper. 3 (TB2_TM1) em LE_REC1.	Flag: oper. 4 (TB2_TM2) em LE_REC1.
B9/175	Flag: oper. 5 (TM1_TE1) em LE_REC1.	Flag: oper. 6 (TM1_TE2) em LE_REC1.	Flag: oper. 7 (TM1_TE3) em LE_REC1.	Flag: oper. 8 (TM2_TE1) em LE_REC1.	Flag: oper. 9 (TM2_TE2) em LE_REC1.

B9/180	Flag: oper. 10 (TM2_TE3) em LE_REC1.	Flag: oper. 11 (COR1_TM1) em LE_REC1.	Flag: oper. 12 (COR1_TM2) em LE_REC1.	Flag: oper. 13 (COR2_TM1) em LE_REC1.	Flag: oper. 14 (COR2_TM2) em LE_REC1.
B9/185	Flag: oper. 15 (COR3_TM1) em LE_REC1.	Flag: oper. 16 (COR3_TM2) em LE_REC1.	Flag: oper. 17 (MIX_TM1) em LE_REC1.	Flag: oper. 18 (MIX_TM2) em LE_REC1.	Flag: oper. 19 (TM1_OUT) em LE_REC1.
B9/190	Flag: oper. 20 (TM2_OUT) em LE_REC1.	Flag: oper. 21 (TE1_OUT) em LE_REC1.	Flag: oper. 22 (TE2_OUT) em LE_REC1.	Flag: oper. 23 (TE3_OUT) em LE_REC1.	Flag: oper. 24 (WAT_TM1) em LE_REC1.
B9/195	Flag: oper. 25 (WAT_TM2) em LE_REC1.	Flag: oper. 26 (WAT_TE1) em LE_REC1.	Flag: oper. 27 (WAT_TE2) em LE_REC1.	Flag: oper. 28 (WAT_TE3) em LE_REC1.	"sem uso"
B9/200	"sem uso"	Flag: espera inicio em LE_REC1.	Evento 5 de LE_REC1.	Flag: esperando execução em LE_REC1.	Flag: operação finalizada em LE_REC1.
B9/205	Evento 1 de LE_REC2.	Flag: pega 1o da FIFO em LE_REC2.	Evento 6 de LE_REC2.	Evento 2 de LE_REC2.	Flag: compara e inicia em LE_REC2.
B9/210	Evento 3 de LE_REC1.	Evento 4 de LE_REC1.	"sem uso"	"sem uso"	"sem uso"
B9/215	"sem uso"	"sem uso"	"sem uso"	"sem uso"	"sem uso"
B9/220	Flag: oper. 0 (Fim da receita) em LE_REC2.	Flag: oper. 1 (TB1_TM1) em LE_REC2.	Flag: oper. 2 (TB1_TM2) em LE_REC2.	Flag: oper. 3 (TB2_TM1) em LE_REC2.	Flag: oper. 4 (TB2_TM2) em LE_REC2.
B9/225	Flag: oper. 5 (TM1_TE1) em LE_REC2.	Flag: oper. 6 (TM1_TE2) em LE_REC2.	Flag: oper. 7 (TM1_TE3) em LE_REC2.	Flag: oper. 8 (TM2_TE1) em LE_REC2.	Flag: oper. 9 (TM2_TE2) em LE_REC2.
B9/230	Flag: oper. 10 (TM2_TE3) em LE_REC2.	Flag: oper. 11 (COR1_TM1) em LE_REC2.	Flag: oper. 12 (COR1_TM2) em LE_REC2.	Flag: oper. 13 (COR2_TM1) em LE_REC2.	Flag: oper. 14 (COR2_TM2) em LE_REC2.
B9/235	Flag: oper. 15 (COR3_TM1) em LE_REC2.	Flag: oper. 16 (COR3_TM2) em LE_REC2.	Flag: oper. 17 (MIX_TM1) em LE_REC2.	Flag: oper. 18 (MIX_TM2) em LE_REC2.	Flag: oper. 19 (TM1_OUT) em LE_REC2.
B9/240	Flag: oper. 20 (TM2_OUT) em LE_REC2.	Flag: oper. 21 (TE1_OUT) em LE_REC2.	Flag: oper. 22 (TE2_OUT) em LE_REC2.	Flag: oper. 23 (TE3_OUT) em LE_REC2.	Flag: oper. 24 (WAT_TM1) em LE_REC2.
B9/245	Flag: oper. 25 (WAT_TM2) em LE_REC2.	Flag: oper. 26 (WAT_TE1) em LE_REC2.	Flag: oper. 27 (WAT_TE2) em LE_REC2.	Flag: oper. 28 (WAT_TE3) em LE_REC2.	"sem uso"
B9/250	"sem uso"	Flag: espera inicio em LE_REC2.	Evento 5 de LE_REC2.	Flag: esperando execução em LE_REC2.	Flag: operação finalizada em LE_REC2.

B10 - Arquivo de Bits (Inicialização da B3)

Arquivo de bits que armazena o estado inicial do arquivo B3. Esse arquivo é utilizado na função Inicializa para colocar o arquivo B3 em seu estado inicial.

Address	0	1	2	3	4
B10/00	0	0	0	1	0
B10/05	1	0	1	0	1
B10/10	0	1	0	1	0
B10/15	1	0	1	0	1
B10/20	0	1	0	1	0
B10/25	1	0	1	0	1
B10/30	0	1	0	1	0
B10/35	1	0	1	0	1
B10/40	0	1	0	1	0
B10/45	1	0	1	0	1
B10/50	0	1	0	1	0
B10/55	1	0	1		

B11 - Arquivo de Bits (Inicialização da B9)

Arquivo de bits que armazena o estado inicial do arquivo B9. Esse arquivo é utilizado na função Inicializa para colocar o arquivo B9 em seu estado inicial.

Address	0	1	2	3	4
B11/00	0	0	0	0	1
B11/05	1	0	0	0	0
B11/10	0	1	1	0	0
B11/15	0	0	0	1	0
B11/20	0	0	0	0	1
B11/25	0	0	1	1	0
B11/30	0	0	1	0	0
B11/35	0	1	0	0	0
B11/40	1	1	0	0	0
B11/45	1	0	0	0	1
B11/50	0	0	0	0	0
B11/55	0	0	0	0	1
B11/60	0	0	0	0	1
B11/65	0	1	1	0	0
B11/70	0	0	0	1	1
B11/75	0	0	0	0	0
B11/80	1	0	0	0	0
B11/85	1	0	0	1	0
B11/90	0	0	0	0	0
B11/95	1	0	0	0	0
B11/100	1	0	0	1	0
B11/105	0	0	0	0	0
B11/110	1	0	0	0	0
B11/115	1	0	0	1	0
B11/120	0	0	0	0	0
B11/125	1	0	0	0	1
B11/130	0	0	0	0	0
B11/135	0	0	1	0	0
B11/140	0	0	0	1	0
B11/145	0	0	0	0	1
B11/150	0	0	0	0	0
B11/155	1	0	0	0	0
B11/160	0	1			

B12 - Arquivo de Bits (Variáveis de inicialização)

Arquivo de bits que armazena o estado das variáveis de Inicializa (função que inicializa o estado do sistema).

Address	0	1	2	3	4
B12/00	Controle: (not) inicializa.	Flag: inicializa finalizado.	Flag: arquivos padrões copiados.	Evento 1 de Inicializa.	Evento 2 de Inicializa.
B12/05	Flag: sistema ligado.				

N13 - Arquivo de Inteiros (Operações da Receita 1)

Arquivo de inteiros que armazena as operações (em sua sequência de execução) da receita 1 a ser executada na planta de mistura. Esta intimamente ligada aos arquivos N15.

Arquivo de inteiros que armazena os parâmetros de volume das operações da receita 1 a ser executada na planta de mistura. Esta intimamente ligada aos arquivos N13.

Address	0	1	2	3	4
N13:00					
N13:01	0	1	2	3	4
N13:02					
N13:03					
N13:04					
N13:05					
N13:06					
N13:07					
N13:08					
N13:09					
N13:10					

N16 - Arquivo de Inteiros (Operações da Receita 2)

Arquivo de inteiros que armazena as operações (em sua sequência de execução) da receita 2 a ser executada na planta de mistura. Esta intimamente ligada aos arquivos N17 e N18.

Address	0	1	2	3	4
N16:00					
N16:05					
N16:10					

N17 - Arquivo de Inteiros (Parâmetros de Volume da Receita 2)

Arquivo de inteiros que armazena os parâmetros de volume das operações da receita 2 a ser executada na planta de mistura. Esta intimamente ligada aos arquivos N16.

Address	0	1	2	3	4
N17:00					
N17:05					
N17:10					

N18 - Arquivo de Inteiros (Parâmetros de Tempo da Receita 2)

Arquivo de inteiros que armazena os parâmetros de tempo das operações da receita 2 a ser executada na planta de mistura. Esta intimamente ligada aos arquivos N16.

Address	0	1	2	3	4
N18:00					
N18:05					
N18:10					

T19 - Arquivo de Temporizador (inicial)

Arquivo do estado inicial do arquivo T4 (temporizadores). O arquivo T19 é utilizado no procedimento de inicialização do sistema.

Address	EN	TT	DN	BASE	PRE	ACC	Descrição
T19:00	0	0	0	1	10	0	Usado em MIX TM1.
T19:01	0	0	0	1	10	0	Usado em MIX TM2.
T19:02	0	0	0	1	10	0	Usado em TE1 OUT.
T19:03	0	0	0	1	10	0	Usado em TE2 OUT.
T19:04	0	0	0	1	10	0	Usado em TE3 OUT.
T19:05	0	0	0	1	1	0	Usado em COR1 TM1.
T19:06	0	0	0	1	1	0	Usado em COR1 TM2.
T19:07	0	0	0	1	1	0	Usado em COR2 TM1.
T19:08	0	0	0	1	1	0	Usado em COR2 TM2.
T19:09	0	0	0	1	1	0	Usado em COR3 TM1.
T19:10	0	0	0	1	1	0	Usado em COR3 TM2.
T19:11	0	0	0	1	1	0	Usado em LE REC1.
T19:12	0	0	0	1	1	0	Usado em LE REC2.

C20 - Arquivo de Contador (inicial)

Arquivo do estado inicial do arquivo C5 (contadores). O arquivo C20 é utilizado no procedimento de inicialização do sistema.

Address	CU	CD	DN	OV	UN	PRE	ACC	Descrição
C20:00	1	0	0	0	0	10	0	Usado em COR1 TM1.
C20:01	1	0	0	0	0	10	0	Usado em COR1 TM2.
C20:02	1	0	0	0	0	10	0	Usado em COR2 TM1.
C20:03	1	0	0	0	0	10	0	Usado em COR2 TM2.
C20:04	1	0	0	0	0	10	0	Usado em COR3 TM1.
C20:05	1	0	0	0	0	10	0	Usado em COR3 TM2.

R21 - Arquivo de Controle (inicial)

Arquivo do estado inicial do arquivo R6 (palavras de controle). O arquivo C21 é utilizado no procedimento de inicialização do sistema.

Address	EN	EU	DN	EM	ER	UL	IN	FD	LEN	Descrição
R21:00	0	0	0	0	0	0	0	0	0	Usado em LE_VOLUME.
R21:01	0	0	0	0	0	0	0	0	0	Usado em LE_VOLUME.
R21:02	0	1	0	1	0	0	0	0	0	Usado em LE_REC1.
R21:03	0	1	0	1	0	0	0	0	0	Usado em LE_REC1.
R21:04	0	1	0	1	0	0	0	0	0	Usado em LE_REC1.
R21:05	0	1	0	1	0	0	0	0	0	Usado em LE_REC2.
R21:06	0	1	0	1	0	0	0	0	0	Usado em LE_REC2.
R21:07	0	1	0	1	0	0	0	0	0	Usado em LE_REC2.

9.1.2 Funções e Rotinas implementadas

A menos das funções *Principal* e *Inicializa*, todas as funções e rotinas foram concebidas utilizando a técnica do PFS/MFG. Por meio de uma metodologia de conversão, converteu-se os diagramas MFG na linguagem Ladder utilizada na programação do CLP-5 da Allen Bradley.

As rotinas implementadas foram:

- Principal
- Inicializa
- Le_Volume
- TB1_TM1
- TB1_TM2
- TB2_TM1
- TB2_TM2
- TM1_TE1
- TM1_TE2
- TM1_TE3
- TM2_TE1
- TM2_TE2
- TM2_TE3
- MIX_TM1
- MIX_TM2
- TE1_OUT
- TE2_OUT
- TE3_OUT
- COR1_TM1
- COR1_TM2
- COR2_TM1
- COR2_TM2
- COR3_TM1
- COR3_TM2
- TM1_OUT
- TM2_OUT
- WAT_TM1
- WAT_TM2
- WAT_TE1
- WAT_TE2
- WAT_TE3
- LE_REC1
- LE_REC2

9.1.1 Rotina Principal

Rotina principal do programa de controle da célula de mistura. Contém as funções que podem ser executadas mediante o acionamento dos bits de controle.

Devido a simplicidade dessa função, não foi necessária a sua modelagem em PFS/MFG.

Rung 2:0

Programa Principal

Rotina principal do programa da celula de mistura. Contem as funcoes que podem ser executadas mediante o acionamento dos bits de controle.

Daniel de Almeida Okino
Gustavo Soares de Arruda Borges
Out/98.

```

| Botao |Botao |
| ligar |emergencia|inicializa Sistema
| sistema |acionado |finalizado ligado
| B3 | B3 | B12 B12
+----] [-----]/[-----] [-----] +----( )-----+
| 0 | 1 | 1 | | 5 | |
| | | | | |reiniciali|
| | | | | |za quando |
| | | | | |desligado |
| | | | | | B12 |
+----(L)-----+
| 6 |

```

Rung 2:1

```

| |reiniciali
| Sistema |za quando inicializa
| ligado |desligado finalizado
| B12 | B12 B12
+----]/[-----] [-----] +----(U)-----+
| 5 | 6 | | 1 | |
| | | | |not |
| | | | |inicializa|
| | | | | B12 |
+----(U)-----+
| 0 |
|reiniciali|
|za quando |
|desligado |
| B12 |
+----(U)-----+
| 6 |

```

Rung 2:2

```

| Inicializa
| +JSR-----+
+-----+JUMP TO SUBROUTINE+-----+
| Prog file number 3|
| Input par |
| Return par |
+-----+

```



```
^^^
|
|          TM1_TE3
|+JSR-----+
|++JUMP TO SUBROUTINE  ++
| |Prog file number  11|
| |Input par         |
| |Return par        |
|+-----+
|          TM2_TE1
|+JSR-----+
|++JUMP TO SUBROUTINE  ++
| |Prog file number  12|
| |Input par         |
| |Return par        |
|+-----+
|          TM2_TE2
|+JSR-----+
|++JUMP TO SUBROUTINE  ++
| |Prog file number  13|
| |Input par         |
| |Return par        |
|+-----+
|          TM2_TE3
|+JSR-----+
|++JUMP TO SUBROUTINE  ++
| |Prog file number  14|
| |Input par         |
| |Return par        |
|+-----+
|          MIX_TM1
|+JSR-----+
|++JUMP TO SUBROUTINE  ++
| |Prog file number  15|
| |Input par         |
| |Return par        |
|+-----+
|          MIX_TM2
|+JSR-----+
|++JUMP TO SUBROUTINE  ++
| |Prog file number  16|
| |Input par         |
| |Return par        |
|+-----+
|          TE1_OUT
|+JSR-----+
|++JUMP TO SUBROUTINE  ++
| |Prog file number  17|
| |Input par         |
| |Return par        |
|+-----+
|
|vvv
vvv
```

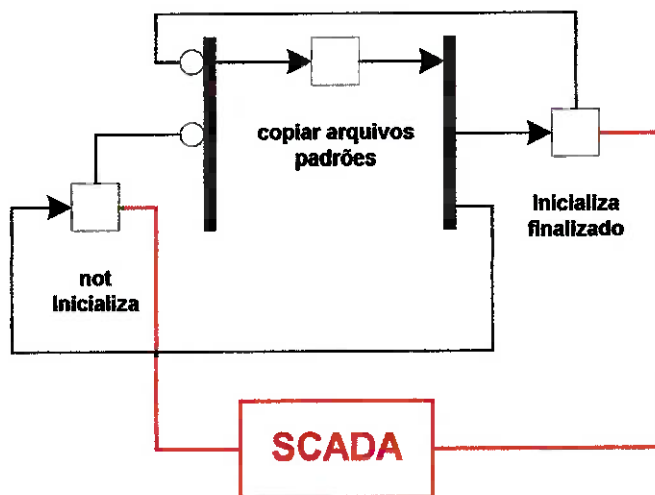
```
^^^
|
|           TE2_OUT
|+JSR-----+|
++JUMP TO SUBROUTINE ++|
||Prog file number 18||
||Input par
||Return par
|+-----+|
|           TE3_OUT
|+JSR-----+|
++JUMP TO SUBROUTINE ++|
||Prog file number 19||
||Input par
||Return par
|+-----+|
|           COR1_TM1
|+JSR-----+|
++JUMP TO SUBROUTINE ++|
||Prog file number 20||
||Input par
||Return par
|+-----+|
|           COR1_TM2
|+JSR-----+|
++JUMP TO SUBROUTINE ++|
||Prog file number 21||
||Input par
||Return par
|+-----+|
|           COR2_TM1
|+JSR-----+|
++JUMP TO SUBROUTINE ++|
||Prog file number 22||
||Input par
||Return par
|+-----+|
|           COR2_TM2
|+JSR-----+|
++JUMP TO SUBROUTINE ++|
||Prog file number 23||
||Input par
||Return par
|+-----+|
|           COR3_TM1
|+JSR-----+|
++JUMP TO SUBROUTINE ++|
||Prog file number 24||
||Input par
||Return par
|+-----+|
|
vvv
```

```
^^^
|
|          COR3_TM2
|+JSR-----+
++JUMP TO SUBROUTINE  ++
||Prog file number 25||
||Input par
||Return par
|+-----+
|          TM1_OUT
|+JSR-----+
++JUMP TO SUBROUTINE  ++
||Prog file number 26||
||Input par
||Return par
|+-----+
|          TM2_OUT
|+JSR-----+
++JUMP TO SUBROUTINE  ++
||Prog file number 27||
||Input par
||Return par
|+-----+
|          WAT_TM1
|+JSR-----+
++JUMP TO SUBROUTINE  ++
||Prog file number 28||
||Input par
||Return par
|+-----+
|          WAT_TM2
|+JSR-----+
++JUMP TO SUBROUTINE  ++
||Prog file number 29||
||Input par
||Return par
|+-----+
|          WAT_TE1
|+JSR-----+
++JUMP TO SUBROUTINE  ++
||Prog file number 30||
||Input par
||Return par
|+-----+
|          WAT_TE2
|+JSR-----+
++JUMP TO SUBROUTINE  ++
||Prog file number 31||
||Input par
||Return par
|+-----+
|
vvv
```

```
      ^^^                               ^^^|
      |                                 | | | | | |
      |           WAT_TE3              | |
      | +JSR-----+                   | |
      | ++JUMP TO SUBROUTINE ++        | |
      | ||Prog file number 32||        | |
      | ||Input par          ||        | |
      | ||Return par         ||        | |
      | +-----+                       | |
      |           LE_REC1              | |
      | +JSR-----+                   | |
      | ++JUMP TO SUBROUTINE ++        | |
      | ||Prog file number 33||        | |
      | ||Input par          ||        | |
      | ||Return par         ||        | |
      | +-----+                       | |
      |           LE_REC2              | |
      | +JSR-----+                   | |
      | ++JUMP TO SUBROUTINE ++        | |
      | ||Prog file number 34||        | |
      | ||Input par          ||        | |
      | ||Return par         ||        | |
      | +-----+                       | |
Rung 2:4
-----[END OF FILE]-----
```

9.1.2 Função Inicializa

Executa a inicializacao dos arquivos B3 e B9 do CLP assim que o CLP é colocado em modo de operacao. É controlada pelo bit B12/0.



Rung 3:4

```

|                                     not |
| Evento 2                           inicializa |
|   B12                               B12 |
| +---] [-----] ( )-----+
| |   4 |                                     0 | |
| |not | |                                     |
| |inicializa| |                               |
| |   B12 | |                               |
| +---] [-----+
|   0 | |
Rung 3:5
|
|-----[END OF FILE]-----+
|

```

9.1.3 Função Le_Volume

Executa a leitura dos sensores de nível dos tanques, através da leitura das placas analógicas.

Devido a simplicidade dessa função, não foi necessária a sua modelagem em PFS/MFG.

Rung 4:0
 Funcao Le_Volume
 Executa a leitura dos sensores de nivel dos tanques, atraves da leitura das
 placas analogicas.
 Daniel de Almeida Okino
 Gustavo S. de Arruda Borges
 Out/98.

```

|                                     Le placas                                     |
|                                     analogicas                                 |
|                                     +BTR-----+                               |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|BLOCK TRNSFR READ      +- (EN) -+
| Rack                  00|
| Group                 3+- (DN) |
| Module                0|
| Control Block        N7:13+- (ER) |
| Data file            N7:18|
| Length               13|
| Continuous           Y|
+-----+-----+-----+-----+-----+-----+

```

```

Rung 4:1
|                                     primeira                                    |
|                                     conversao                                 |
|                                     +FAL-----+                               |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| N7:13 R6:1           | FILE ARITH/LOGICAL+- (EN) -+
+---] [---]/[---]-----+-----+-----+-----+-----+-----+-----+
| 13 DN                | Control          R6:0|
|                       | Length             13+- (DN) |
|                       | Position           0|
|                       | Mode              ALL+- (ER) |
|                       | Dest              #N7:31|
|                       | Expression         473|
|                       | #N7:18 AND 4095 |
+-----+-----+-----+-----+-----+-----+

```

```

Rung 4:2
|                                     segunda                                    |
|                                     conversao                                 |
|                                     +FAL-----+                               |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| R6:0                 | FILE ARITH/LOGICAL+- (EN) -+
+---] [---]-----+-----+-----+-----+-----+-----+-----+
| DN                   | Control          R6:1|
|                       | Length             13+- (DN) |
|                       | Position           0|
|                       | Mode              ALL+- (ER) |
|                       | Dest              #N7:0|
|                       | Expression         25|
|                       | #N7:31 * #F8:0 |
+-----+-----+-----+-----+-----+-----+

```

```

Rung 4:3
| N7:13                                                         N7:13 |
+---] [---]-----+-----+-----+-----+-----+-----+-----+-----+
| 12                                                           15 |

```

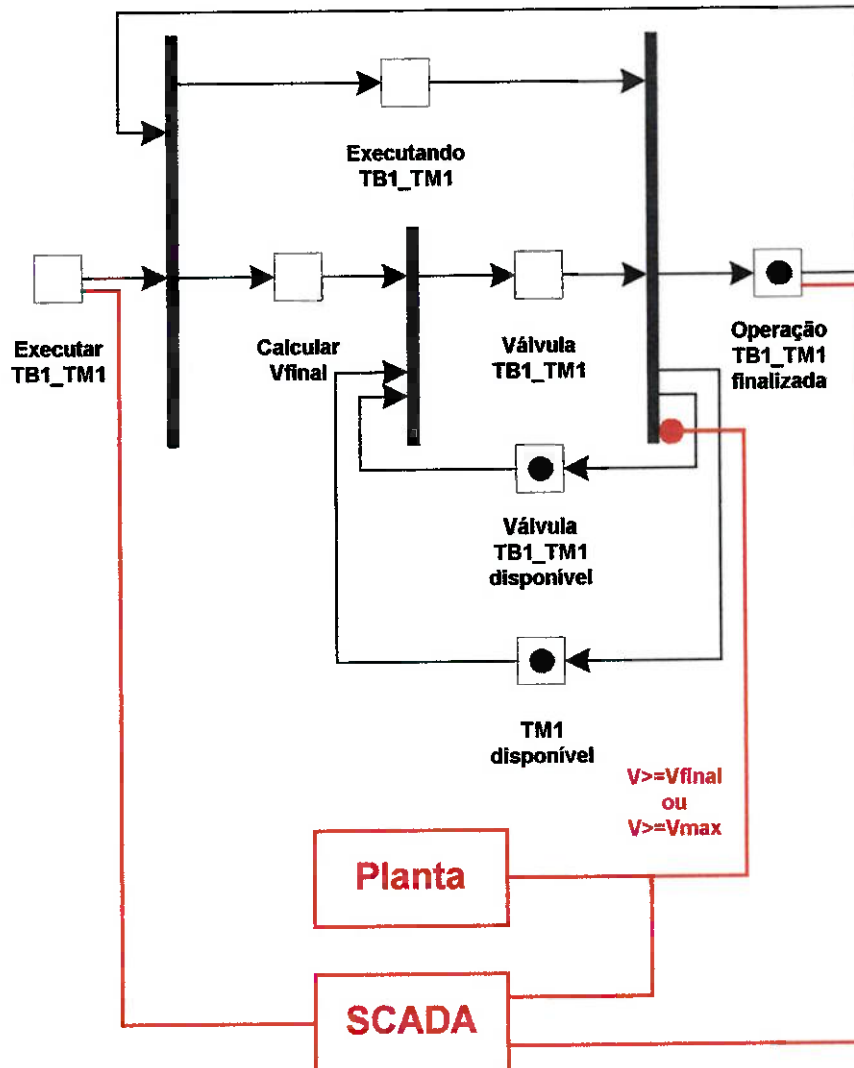
```

Rung 4:4
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     [END OF FILE]-----+-----+-----+-----+

```

9.1.4 Função TB1_TM1

Executa a passagem de "N7:44" litros de líquido do tanque de base TB1 para o tanque de mistura TM1.



Rung 5:0

Funcao TB1_TM1

Executa a passagem de "N7:44" litros de liquido do tanque de base TB1 para o tanque de mistura TM1.

Daniel de Almeida Okino
 Gustavo Soares de A. Borges
 Out/98.

```

| executar |                                     executar | |
| TB1_TM1 |Evento 1 |                                     TB1_TM1 |
| B3      | B9      |                                     B3      |
+----] [-----]/[-----]-----+-----+-----+
|      2      |      2      |                                     2      |

```

Rung 5:1

```

| operacao |
| executar | TB1_TM1 | executando | Vfinal | |
| TB1_TM1 | finalizada | TB1_TM1 | calculado | Evento 1 |
| B3      | B3      | B9      | B9      | B9      |
+----] [-----] [-----] / [-----] / [-----] +-----+-----+
|      2      |      3      |      0      |      1      |                                     2      |

```

Rung 5:2

```

| Evento 1 |                                     executando |
| B9      |                                     TB1_TM1 |
+----] [-----]-----+-----+-----+
|      2      |                                     0      |
| executando |
| TB1_TM1 |Evento 3 |
| B9      | B9      |
+----] [-----] / [-----] +-----+-----+
|      0      |      3      |

```

Rung 5:3

```

| Evento 1 |                                     Vfinal |
| B9      |                                     +CPT-----+
+----] [-----]-----+-----+-----+
|      2      |                                     +COMPUTE +-----+
| Vfinal | | Dest N7:45 | |
| calculado |Evento 2 | | Expression |
| B9      | B9      | | N7:0 + N7:44 |
+----] [-----] / [-----] +-----+-----+
|      1      |      6      | | Vfinal |
|                                     | calculado |
|                                     | B9      |
|                                     |-----+-----+
|                                     |      1      |

```

Rung 5:4

```

| Vfinal | TM1 | valvula | |
| calculado | disponivel | disponivel | TB1_TM1 |
| B9      | B9      | B9      | O:006 |
+----] [-----] [-----] [-----] / [-----] +-----+-----+
|      1      |      4      |      5      |      10      |                                     6      |

```

```

Rung 5:5
|
|      Evento 2
|      B9
|-----] [-----+-----+-----+
|      6
| |valvula |
| |TB1_TM1 |Evento 3
| |O:006   |B9
| +----] [-----]/[-----+
|      10      3
|

```

```

Rung 5:6
|
| executando|valvula      Volume      operacao |valvula |
| TB1_TM1  |TB1_TM1      TM1      TB1_TM1 |TB1_TM1 |TM1
| B9       |O:006      +CMP-----+ B3      B9      B9 >
| +----] [-----] [-----+COMPARE +----]/[-----]/[-----]>
|      0      10      |Expression      3      5      4 >
|      |      |Expression
|      |      |N7:0 >= N7:45
|      +-----+
|

```

```

|
|      Evento 3
|      B9
|<----( )-----+
|<      3
|

```

```

Rung 5:7
|
|      Evento 3
|      B9
|-----] [-----+-----+-----+
|      3
| |operacao |
| |TB1_TM1 |
| |finalizada|Evento 1
| |B3      |B9
| +----] [-----]/[-----+
|      3      2
|

```

```

Rung 5:8
|
|      Evento 3
|      B9
|-----] [-----+-----+-----+
|      3
| |valvula |
| |TB1_TM1 |
| |disponivel|Evento 2
| |B9      |B9
| +----] [-----]/[-----+
|      5      6
|

```

Rung 5:9

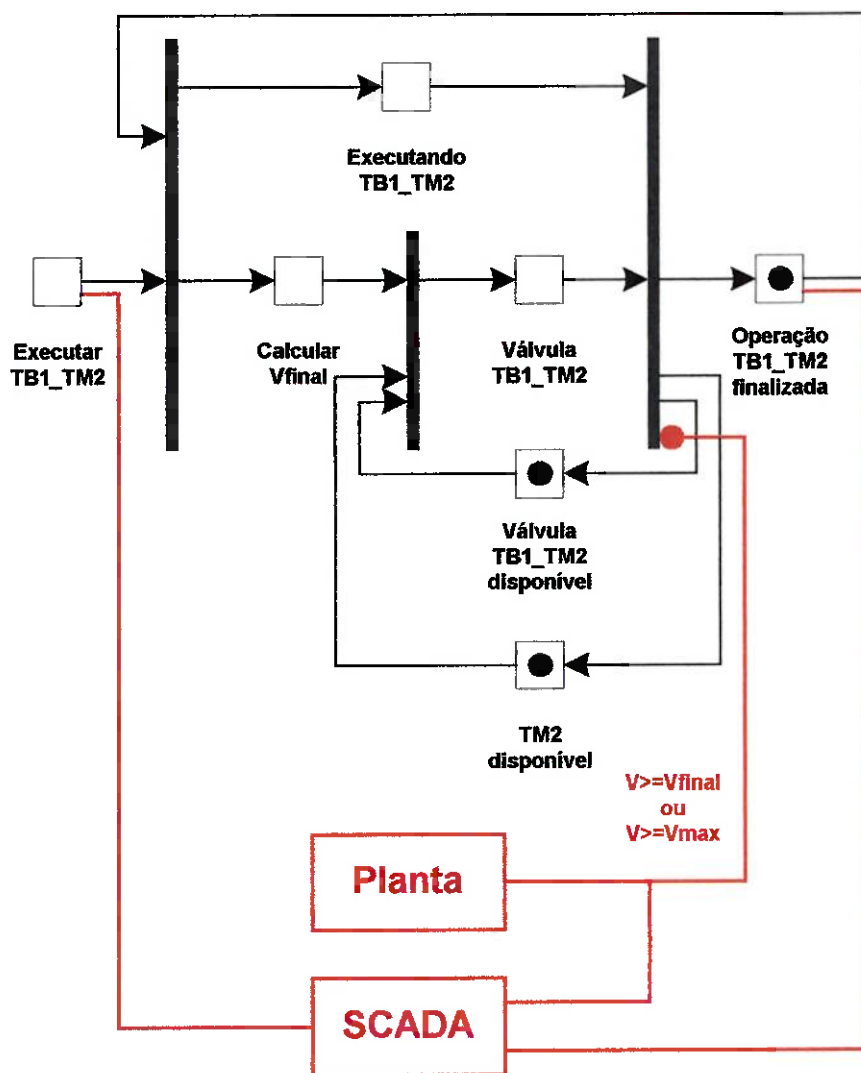
```

|                                     TM1 |
|                                     disponivel |
|      Evento 3                          B9 |
|      B9 |-----|-----|-----|-----| |
|      3 |                                     ( ) |
|      |                                     4 |
|      | TM1 |-----|-----|-----|-----|
|      | disponivel | Evento 2 |
|      | B9 |-----|-----|-----|-----|
|      |-----|-----|-----|-----|
|      4 |                                     6 |
Rung 5:10
|-----|-----|-----|-----|-----|
|                                     [END OF FILE] |
|

```

9.1.5 Função TB1_TM2

Executa a passagem de "N7:46" litros de líquido do tanque de base TB1 para o tanque de mistura TM2.



Rung 6:0

Funcao TB1_TM2

Executa a passagem de "N7:46" litros de liquido do tanque de base TB1 para o tanque de mistura TM2.

Daniel de Almeida Okino
Gustavo Soares de A. Borges
Out/98.

```

| executar |                                     executar | |
| TB1_TM2 |Evento 1 |                                     TB1_TM2 |
| B3      |B9      |                                     B3      |
+-----+ [-----] / [-----] +-----+ ( ) +-----+
| 4      | 7      |                                     4      |

```

Rung 6:1

```

| operacao |
| executar | TB1_TM2 | executando | Vfinal | |
| TB1_TM2 | finalizada | TB1_TM2 | calculado | Evento 1 |
| B3      | B3      | B9      | B9      | B9      |
+-----+ [-----] / [-----] / [-----] / [-----] +-----+ ( ) +-----+
| 4      | 5      | 8      | 9      | 7      |

```

Rung 6:2

```

| Evento 1 |                                     executando |
| B9      |                                     TB1_TM2 |
+-----+ [-----] +-----+ ( ) +-----+
| 7      |                                     8      |
| executar |
| TB1_TM2 |Evento 3 |
| B9      | B9      |
+-----+ [-----] / [-----] +-----+
| 8      | 10     |

```

Rung 6:3

```

| Evento 1 |                                     Vfinal |
| B9      |                                     +CPT-----+
+-----+ [-----] +-----+ +COMPUTE +-----+
| 7      |                                     ||Dest   N7:47|| |
| Vfinal |                                     ||      30||
| calculado |Evento 2 |                                     ||Expression ||
| B9      | B9      |                                     ||N7:1 + N7:46 ||
+-----+ [-----] / [-----] +-----+
| 9      | 13     |                                     Vfinal |
|                                     calculado |
|                                     B9      |
+-----+ ( ) +-----+
|                                     9      |

```

Rung 6:4

```

| Vfinal | TM2 | valvula | | |
| calculado | disponivel | disponivel | TB1_TM2 |
| B9      | B9      | B9      | O:007 | Evento 2 |
+-----+ [-----] [-----] [-----] / [-----] +-----+ ( ) +-----+
| 9      | 11     | 12     | 12     | 13     |

```

```
Rung 6:5
|
|
|      Evento 2
|      B9
|-----] [-----] ( )-----+
|      13
| |valvula | |
| |TB1_TM2 |Evento 3 |
| | O:007  | B9 |
| +----] [-----]/[-----+
|      12      10
```

```
Rung 6:6
|
| executando|valvula      Volume      operacao [valvula |
| TB1_TM2  |TB1_TM2      TM2      TB1_TM2 |TB1_TM2 |TM2
| B9       | O:007      +CMP-----+      B3 | B9 | B9 >
|-----] [-----] [-----+COMPARE +----]/[-----]/[-----]/[----->
|      8      12 |Expression      5      12      11 >
| |N7:1 >= N7:47 |
| +-----+
|
```

```

|
|
|      Evento 3
|      < B9
| <----( )-----+
| <      10
|
```

```
Rung 6:7
|
|
|      Evento 3
|      B9
|-----] [-----] ( )-----+
|      10
| |operacao | |
| |TB1_TM2 |
| |finalizada|Evento 1 |
| | B3      | B9 |
| +----] [-----]/[-----+
|      5      7
```

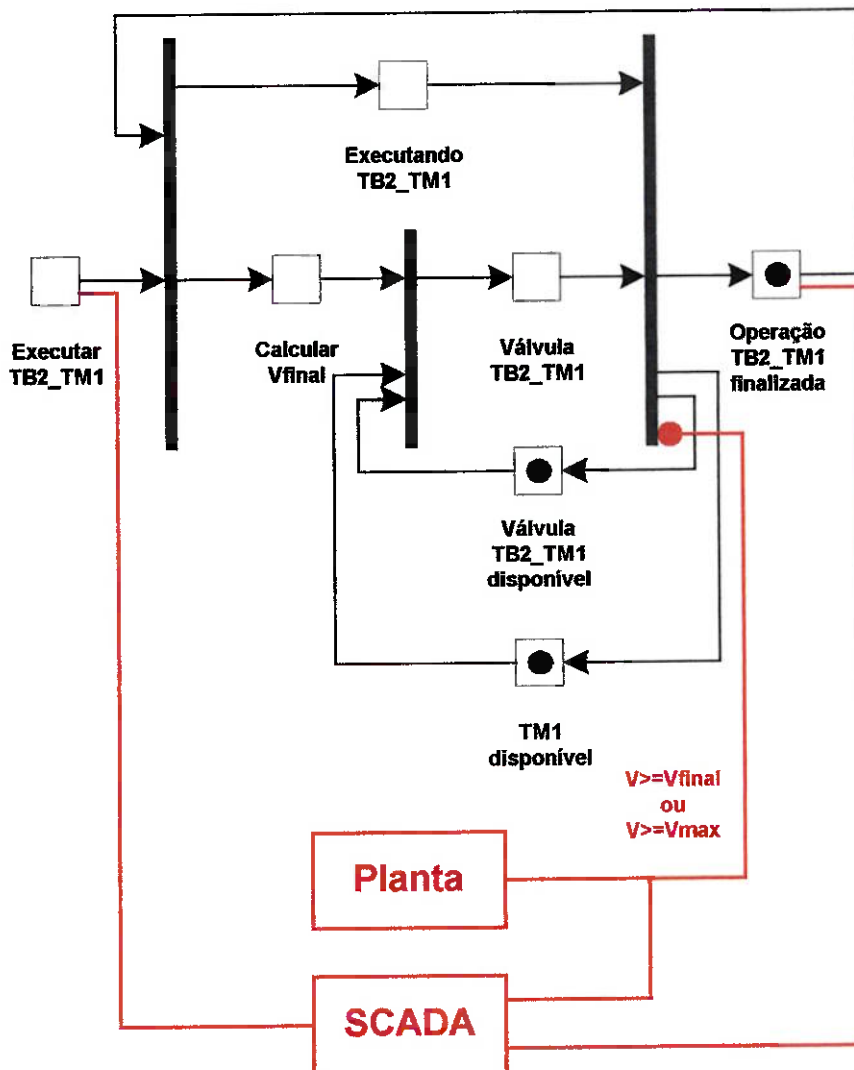
```
Rung 6:8
|
|
|      Evento 3
|      B9
|-----] [-----] ( )-----+
|      10
| |valvula | |
| |TB1_TM2 |
| |disponivel|Evento 2 |
| | B9      | B9 |
| +----] [-----]/[-----+
|      12      13
```

Rung 6:9

```
|
|      Event 3      TM2
|      B9           disponivel
|      ( )          B9
+----] [-----+-----+-----+
|      10           |           11
| TM2              |           | |
| disponivel|Event 2 |           |
| B9         B9     |           |
+----] [-----]/{-----+
|      11          13
Rung 6:10
|-----[END OF FILE]-----+
|
```

9.1.6 Função TB2_TM1

Executa a passagem de "N7:48" litros de líquido do tanque de base TB2 para o tanque de mistura TM1.



```

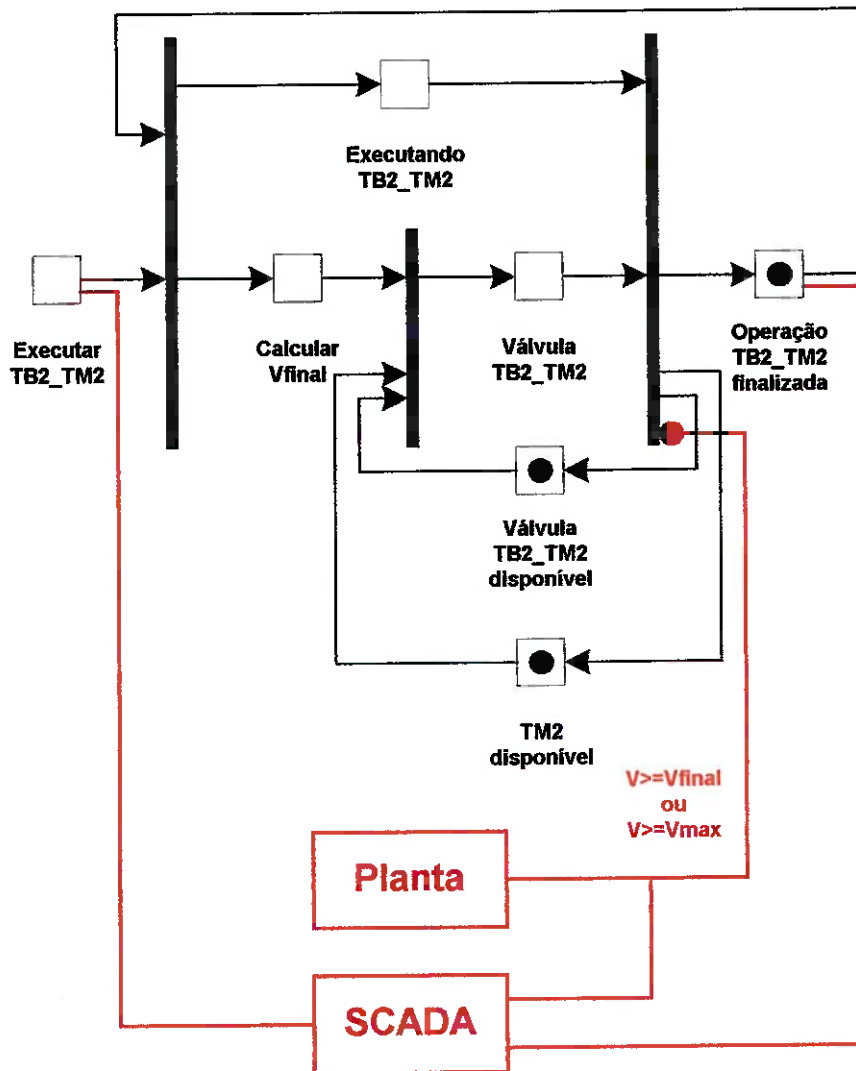
Rung 7:0
Funcao TB2_TM1
Executa a passagem de "N7:48" litros de liquido do tanque de base TB2 para o
tanque de mistura TM1.
Daniel de Almeida Okino
Gustavo Soares de A. Borges
Out/98.
| executar |                                     executar | |
| TB2_TM1 |Evento 1 |                                     TB2_TM1 |
| B3      |B9      |                                     B3      |
+-----+ [-----] / [-----] +-----+ ( ) +-----+
| 6      |14     |                                     6      |
Rung 7:1
|          |operacao | | | |
| executar |TB2_TM1 |executando|Vfinal |
| TB2_TM1 |finalizada|TB2_TM1 |calculado |Evento 1 |
| B3      |B3      |B9      |B9      |B9      |
+-----+ [-----] [-----] / [-----] / [-----] +-----+ ( ) +-----+
| 6      |7      |15     |16     |14     |
Rung 7:2
|          |                                     executando |
| Evento 1 |                                     TB2_TM1 |
| B9      |                                     B9      |
+-----+ [-----] +-----+ ( ) +-----+
| 14     |                                     15     |
|executando|
| TB2_TM1 |Evento 3 |
| B9      |B9      |
+-----+ [-----] / [-----] +-----+
| 15     |17     |
Rung 7:3
| Evento 1 |                                     Vfinal |
| B9      |                                     +CPT-----+ |
+-----+ [-----] +-----+ +-----+ +-----+ +-----+
| 14     |                                     ||Dest      N7:49|| | |
|Vfinal |                                     ||          81|| |
|calculado |Evento 2 |                                     ||Expression || |
| B9      |B9      |                                     ||N7:0 + N7:48 || |
+-----+ [-----] / [-----] +-----+ +-----+
| 16     |19     |                                     Vfinal |
|                                     calculado |
|                                     B9      |
+-----+ ( ) +-----+
|                                     16     |
Rung 7:4
| Vfinal |TM1 |valvula | | |
| calculado |disponivel|disponivel|TB2_TM1 |Evento 2 |
| B9      |B9      |B9      |O:006 |B9      |
+-----+ [-----] [-----] [-----] / [-----] +-----+ ( ) +-----+
| 16     |4      |18     |11     |19     |
    
```



```
Rung 7:9
|
|   Evento 3
|   B9
|-----] [-----+-----+-----+-----+-----+-----+
|   |   17
|   |   |
|   | TM1
|   | |
|   | |disponivel|Evento 2
|   | |   B9      |   B9
|   | +---] [-----]/[-----+
|   |   4          19
Rung 7:10
|
|-----[END OF FILE]-----+
|
```

9.1.7 Função TB2_TM2

Executa a passagem de "N7:50" litros se líquido do tanque de base TB2 para o tanque de mistura TM2.



Rung 8:0

Funcao TB2_TM2

Executa a passagem de "N7:50" litros se liquido do tanque de base TB2 para o tanque de mistura TM2.

Daniel de Almeida Okino
 Gustavo Soares de A. Borges
 Out/98.

```

| executar |                                     executar | |
| TB2_TM2 |Evento 1 |                                     TB2_TM2 |
|   B3   |   B9   |                                     B3   |
+-----+ [-----]/[-----] ( )-----+
|      8      |      20      |                                     8      |
    
```

Rung 8:1

```

| executar |operacao |                                     | | |
| TB2_TM2 |TB2_TM2 |executando|Vfinal |                                     |
| TB2_TM2 |finalizada|TB2_TM2 |calculado |Evento 1 |
|   B3   |   B3   |   B9   |   B9   |   B9   |
+-----+ [-----]/[-----]/[-----] ( )-----+
|      8      |      9      |      21      |      22      |      20      |
    
```

Rung 8:2

```

|                                     executando |
| Evento 1 |                                     TB2_TM2 |
|   B9   |                                     B9   |
+-----+ [-----] ( )-----+
|      20      |                                     21      |
| executando |
| TB2_TM2 |Evento 3 |
|   B9   |   B9   |
+-----+ [-----]/[-----]
|      21      |      23      |
    
```

Rung 8:3

```

| Evento 1 |                                     Vfinal |
|   B9   |                                     +CPT-----+
+-----+ [-----] +-----+
|      20      |                                     ||Dest      N7:51|| |
| Vfinal |                                     ||          104||
| calculado |Evento 2 |                                     ||Expression ||
|   B9   |   B9   |                                     ||N7:1 + N7:50||
+-----+ [-----]/[-----] +-----+
|      22      |      25      |                                     Vfinal |
|                                     |                                     calculado |
|                                     |                                     B9   |
|                                     |                                     +-----+ ( )-----+
|                                     |                                     22      |
    
```

Rung 8:4

```

| Vfinal |TM2 |valvula | |
| calculado |disponivel|disponivel|TB2_TM2 |
|   B9   |   B9   |   B9   |   O:007 |
+-----+ [-----] [-----] {-----}/[-----] ( )-----+
|      22      |      11      |      24      |      13      |      25      |
    
```

```

Rung 8:5
|
|   Evento 2
|   B9
|-----] [-----] ( )-----+
|   |   25
|   |valvula | |
|   |TB2_TM2 |Evento 3 |
|   | O:007   |B9      |
|   +----] [-----]/[-----+
|   |   13     |   23
|

```

```

Rung 8:6
|
| executando|valvula      Volume
| TB2_TM2  |TB2_TM2      TM2
| B9       |O:007      +CMP-----+
|-----] [-----] +COMPARE +----]/[-----]/[-----]/[-----]>
|   |   21     |   13 |Expression |   9   |   24   |   11   |>
|   |         |         |N7:1 >= N7:51|
|   +-----+
|

```

```

|
|   Evento 3
|   < B9
|   <---( )-----+
|   <   23
|

```

```

Rung 8:7
|
|   Evento 3
|   B9
|-----] [-----] ( )-----+
|   |   23
|   |operacao | |
|   |TB2_TM2 |
|   |finalizada|Evento 1 |
|   | B3       |B9      |
|   +----] [-----]/[-----+
|   |   9       |   20
|

```

```

Rung 8:8
|
|   Evento 3
|   B9
|-----] [-----] ( )-----+
|   |   23
|   |valvula | |
|   |TB2_TM2 |
|   |disponivel|Evento 2 |
|   | B9       |B9      |
|   +----] [-----]/[-----+
|   |   24     |   25
|

```

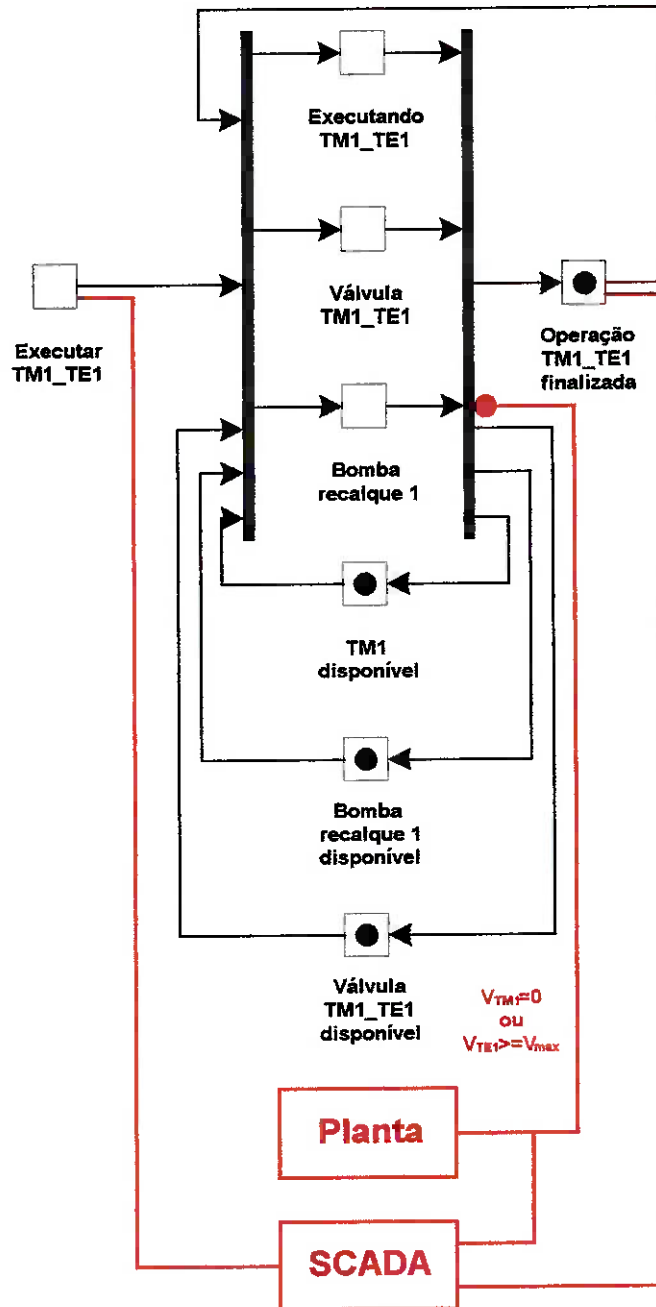
```

Rung 8:9
|
|
|      Evento 3
|      B9
|+----] [-----+-----+-----+-----+-----+-----+-----+-----+
|      23
| |TM2
| |disponivel|Evento 2
| |      B9      B9
| +----] [-----]/[-----+
|      11      25
Rung 8:10
|
|-----[END OF FILE]-----|
|

```

9.1.8 Função TM1_TE1

Executa a passagem de todo o conteúdo de líquido do tanque de mistura TM1 para o tanque de estoque TE1.



Rung 9:0

Funcao TM1_TE1

Executa a passagem de todo o conteudo de liquido do tanque de mistura TM1 para o tanque de estoque TE1.

Daniel de Almeida Okino
Gustavo S. de Arruda Borges
Out/98.

```

| executar |                                     executar | |
| TM1_TE1 |Evento 1 |                                     TM1_TE1 |
| B3      |B9      |                                     B3      |
+-----+ [-----]/[-----] ( )-----+
| 10      |26      |                                     10      |

```

Rung 9:1

```

|          |operacao |          |bomba   |valvula |
| executar |TM1_TE1 |TM1      |recalque 1|TM1_TE1 |executando|valvua
| TM1_TE1 |finalizada|disponivel|disponivel|disponivel|TM1_TE1 |TM1_TE1
| B3      |B3      |B9      |B9      |B9      |B9      |O:006 >
+-----+ [-----] [-----] [-----] [-----] [-----]/[-----]/[-----] >
| 10      |11      |4       |27      |28      |29      |17 >

```

```

|          |          |
| bomba   |          |
| recalque 1|Evento 1 |
| < O:013  |B9      |
| <-----+ [-----] ( )-----+
| < 15     |26      |

```

Rung 9:2

```

|          |          |          |          |
|          |Evento 1 |          |          |
|          |B9      |          |          |
+-----+ [-----] ( )-----+
|          |26      |          |          | |
| |executando|          |          |          |
| |TM1_TE1  |Evento 2 |          |          |
| |B9      |B9      |          |          |
| +-----+ [-----]/[-----]
|          |29      |30      |          |

```

Rung 9:3

```

|          |          |          | |
|          |Evento 1 |          |          |
|          |B9      |          |          |
+-----+ [-----] ( )-----+
|          |26      |          |          | |
| |valvula  |          |          |          |
| |TM1_TE1 |Evento 2 |          |          |
| |O:006   |B9      |          |          |
| +-----+ [-----]/[-----]
|          |17      |30      |          |

```



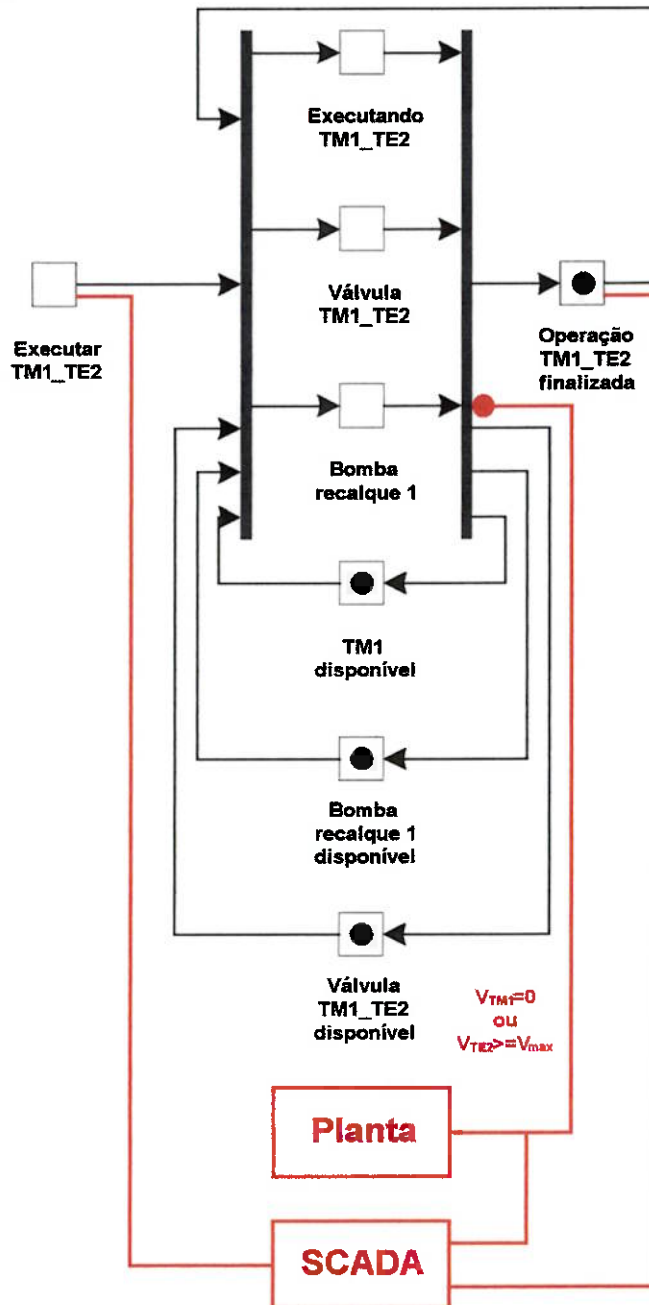
```

Rung 9:7
|
|          TM1
| Evento 2  disponivel
|   B9
| +-----] [-----+-----+-----+-----+-----+-----+-----+-----+
| |   30
| | TM1
| | disponivel|Evento 1
| |   B9       B9
| +-----] [-----]/[-----+
| |   4         26
Rung 9:8
|
|          bomba
|          recalque 1
| Evento 2  disponivel
|   B9
| +-----] [-----+-----+-----+-----+-----+-----+-----+
| |   30
| | bomba
| | recalque 1
| | disponivel|Evento 1
| |   B9       B9
| +-----] [-----]/[-----+
| |   27        26
Rung 9:9
|
|          valvula
|          TM1_TE1
| Evento 2  disponivel
|   B9
| +-----] [-----+-----+-----+-----+-----+-----+-----+
| |   30
| | valvula
| | TM1_TE1
| | disponivel|Evento 1
| |   B9       B9
| +-----] [-----]/[-----+
| |   28        26
Rung 9:10
|
|-----[END OF FILE]-----+
|

```

9.1.9 Função TM1_TE2

Executa a passagem de todo o conteúdo de líquido do tanque de mistura TM1 para o tanque de estoque TE2.



Rung 10:0

Funcao TM1_TE2

Executa a passagem de todo o conteudo de liquido do tanque de mistura TM1 para o tanque de estoque TE2.

Daniel de Almeida Okino

Gustavo S. de Arruda Borges

Out/98.

```

| executar |                                     executar | |
| TM1_TE2 |Evento 1 |                                     TM1_TE2 |
| B3      |B9      |                                     B3      |
+----] [-----] / [-----] ( )-----+
| 12      |31      |                                     12      |

```

Rung 10:1

```

| executar |operacao |          |bomba |valvula |
| TM1_TE2 |TM1      |recalque 1|TM1_TE2 |executando|valvula
| TM1_TE2 |finalizada|disponivel|disponivel|disponivel|TM1_TE2 |TM1_TE2
| B3      |B3      |B9      |B9      |B9      |B9      |O:007  >
+----] [-----] [-----] [-----] [-----] / [-----] / [-----] >
| 12      |13      |4       |27      |32      |33      |10     >

```

```

| bomba |
| recalque 1|Evento 1 |
< O:013   |B9      |
<-----] / [-----] ( )-----+
< 15      |31      |

```

Rung 10:2

```

|          |                                     executando |
| Evento 1 |                                     TM1_TE2 |
| B9      |                                     B9      |
+----] [-----] ( )-----+
| 31      |                                     33      | |
| |executando|
| |TM1_TE2 |Evento 2 |
| |B9      |B9      |
| +----] [-----] / [-----]
| 33      |34      |

```

Rung 10:3

```

|          |                                     valvula |
| Evento 1 |                                     TM1_TE2 |
| B9      |                                     O:007 |
+----] [-----] ( )-----+
| 31      |                                     10     | |
| |valvula |
| |TM1_TE2 |Evento 2 |
| |O:007   |B9      |
| +----] [-----] / [-----]
| 10      |34      |

```

Rung 10:4

```

|                                     bomba |
| Evento 1                             recalque 1 |
| B9                                     O:013 |
+-----] [-----] [-----] [-----] ( )-----+
| 31                                     15 |
| bomba |                               | |
| recalque 1 | Evento 2 |               |
| O:013   B9 |               |               |
+-----] [-----] / [-----] +-----+
| 15       34 |               |               |

```

Rung 10:5

```

|                                     bomba |
| executando | valvula | bomba | Volume | TM1 | bomba |
| TM1_TE2 | TM1_TE2 | recalque 1 | TM1 | disponivel | recalque 1 |
| B9 | O:007 | O:013 | +CMP-----+ | B9 | B9 |
+-----] [-----] [-----] [-----] [-----] / [-----] / [-----] +-----+
| 33 | 10 | 15 | || Expression || 4 | 27 |
| || N7:0 <= 0 || || ||
+-----+
| Volume |
| TE2 |
+-----+
| +CMP-----+ |
| +COMPARE |
| Expression |
| N7:9 >= N7:57 |
+-----+

```

```

valvula | operacao |
TM1_TE2 | TM1_TE2 |
disponivel | finalizada | Evento 2
< B9 | B3 | B9 |
<-----] / [-----] / [-----] ( )-----+
< 32 | 13 | 34 |

```

Rung 10:6

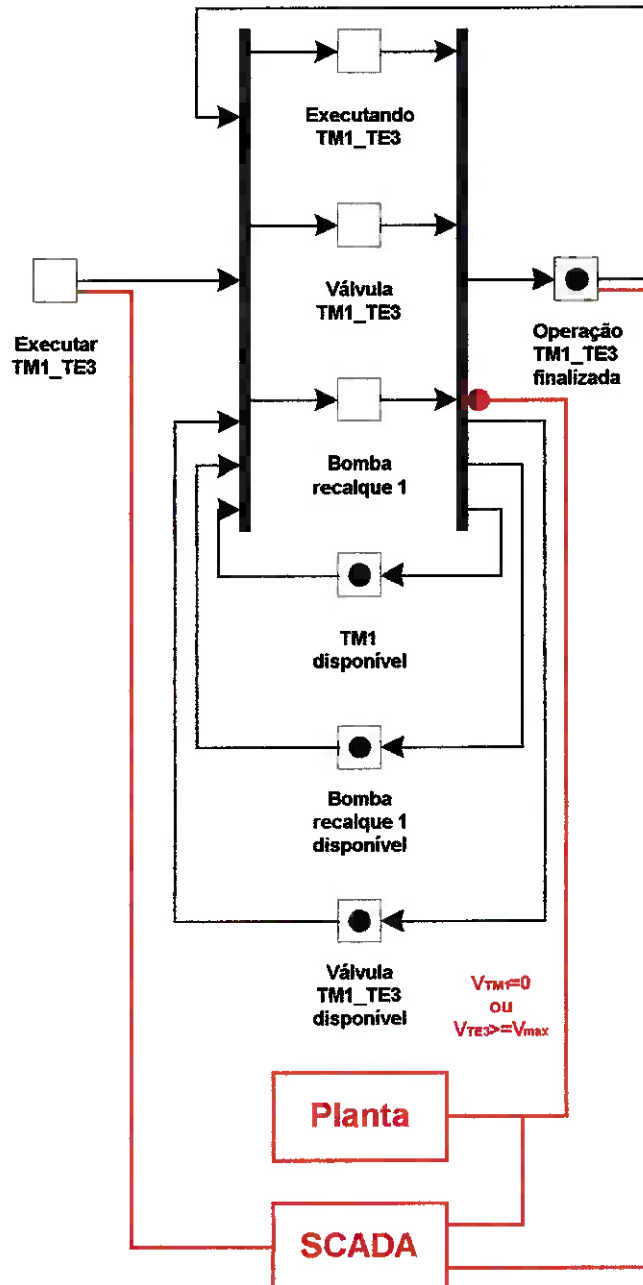
```

|                                     operacao |
|                                     TM1_TE2 |
| Evento 2                             finalizada |
| B9                                     B3 |
+-----] [-----] [-----] [-----] ( )-----+
| 34                                     13 |
| operacao |                               | |
| TM1_TE2 |                               |
| finalizada | Evento 1 |               |
| B3 | B9 |               |
+-----] [-----] / [-----] +-----+
| 13       31 |               |               |

```


9.1.10 Função TM1_TE3

Executa a passagem de todo o conteúdo de líquido do tanque de mistura TM1 para o tanque de estoque TE3.



Rung 11:0

Funcao TM1_TE3

Executa a passagem de todo o conteudo de liquido do tanque de mistura TM1 para o tanque de estoque TE3.

Daniel de Almeida Okino
 Gustavo S. de Arruda Borges
 Out/98.

```
| executar |                                     executar | |
| TM1_TE3 |Evento 1 | TM1_TE3 |
| B3      |B9      | B3      |
+----] [-----]/[-----]-----+
| 14     | 35     | 14     |
```

Rung 11:1

```
|          |operacao |          |bomba   |valvula |
| executar |TM1_TE3 |TM1      |recalque 1|TM1_TE3 |executando|valvula
| TM1_TE3 |finalizada|disponivel|disponivel|disponivel|TM1_TE3 |TM1_TE3
| B3      |B3      |B9      |B9      |B9      |B9      |O:007  >
+----] [-----] [-----] [-----] [-----] [-----] [-----] [-----] [-----]
| 14     | 15     | 4      | 27     | 36     | 37     | 11    >
```

```
|
| bomba   |
| recalque 1|Evento 1 |
| < O:013 |B9      |
| <----]/[-----] [-----]
| < 15   | 35     |
```

Rung 11:2

```
|          |                                     executando |
| Evento 1 | TM1_TE3 |
| B9      | B9      |
+----] [-----]-----+
| 35     | 37     |
| executando|
| TM1_TE3 |Evento 2 |
| B9      |B9      |
| +----] [-----] [-----]
| 37     | 38     |
```

Rung 11:3

```
|          |valvula |
| Evento 1 | TM1_TE3 |
| B9      | O:007   |
+----] [-----]-----+
| 35     | 11     |
| valvula |
| TM1_TE3 |Evento 2 |
| O:007   |B9      |
| +----] [-----] [-----]
| 11     | 38     |
```

```

Rung 11:4
|
|      bomba
|      recalque 1
|      O:013
|      ( )
|      35 ----- 15
|      |bomba
|      |recalque 1|Evento 2
|      | O:013      B9
|      | +----] [-----]/[-----+
|      | 15          38

```

```

Rung 11:5
|
|      executando|valvula |bomba      Volume      TM1      bomba
|      TM1_TE3  |TM1_TE3 |recalque 1  TM1      disponivel|recalque 1
|      B9      O:007  O:013      +CMP-----+      B9      B9
|      +----] [-----] [-----] [-----++COMPARE      ++----]/[-----]/[----->
|      37      11      15      ||Expression      ||      4      27      >
|      ||N7:0 <= 0      ||
|      |-----+
|      |      Volume
|      |      TE3
|      |-----+
|      |-----+
|      |++COMPARE      ++
|      |Expression
|      |N7:10 >= N7:58
|      |-----+

```

```

valvula |operacao |
TM1_TE3 |TM1_TE3 |
disponivel|finalizada|Evento 2
< B9      B3      B9
<----]/[-----]/[------( )-----+
< 36      15      38

```

```

Rung 11:6
|
|      operacao
|      TM1_TE3
|      finalizada
|      B3
|      ( )
|      38 ----- 15
|      |operacao
|      |TM1_TE3
|      |finalizada|Evento 1
|      | B3      B9
|      | +----] [-----]/[-----+
|      | 15          35

```

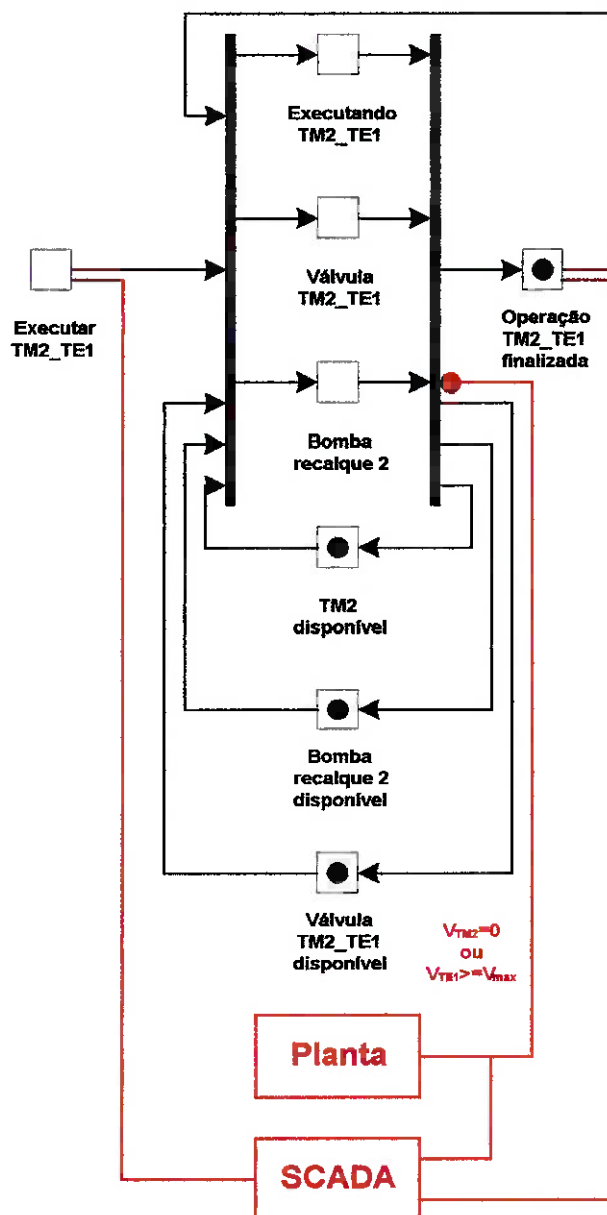
```

Rung 11:7
|
|      Evento 2      TM1
|      B9            disponivel
|      |            B9
| +----] [-----+-----+-----+-----+-----+-----+
| |      38          | |
| | TM1             |
| | disponivel|Evento 1 |
| | B9          B9   |
| +----] [-----+-----+
| |      4          35 |
|
Rung 11:8
|
|      Evento 2      bomba
|      B9            recalque 1
|      |            disponivel
|      |            B9
| +----] [-----+-----+-----+-----+-----+
| |      38          | |
| | bomba           |
| | recalque 1     |
| | disponivel|Evento 1 |
| | B9          B9   |
| +----] [-----+-----+
| |      27          35 |
|
Rung 11:9
|
|      Evento 2      valvula
|      B9            TM1_TE3
|      |            disponivel
|      |            B9
| +----] [-----+-----+-----+-----+-----+
| |      38          | |
| | valvula        |
| | TM1_TE3       |
| | disponivel|Evento 1 |
| | B9          B9   |
| +----] [-----+-----+
| |      36          35 |
|
Rung 11:10
|
|-----[END OF FILE]-----+
|

```

9.1.11 Função TM2_TE1

Executa a passagem de todo o conteúdo de líquido do tanque de mistura TM2 para o tanque de estoque TE1.



Rung 12:0

Funcao TM2_TE1

Executa a passagem de todo o conteudo de liquido do tanque de mistura TM2 para o tanque de estoque TE1.

Daniel de Almeida Okino
Gustavo S. de Arruda Borges
Out/98.

```

| executar |                                     executar | |
| TM2_TE1 |Evento 1 | TM2_TE1 |
|   B3   |   B9   |   B3   |
+----] [-----]/[-----] ( )-----+
|   16   |   39   |   16   |

```

Rung 12:1

```

|          |operacao |          |bomba |valvula |
| executar |TM2_TE1 |TM2      |recalque 2|TM2_TE1 |executando|valvula
| TM2_TE1 |finalizada|disponivel|disponivel|disponivel|TM2_TE1 |TM2_TE1
|   B3   |   B3   |   B9   |   B9   |   B9   |   B9   |O:011 >
+----] [-----] [-----] [-----] [-----] [-----]/[-----]/[-----] >
|   16   |   17   |   11   |   40   |   41   |   42   |   11   >

```

```

| bomba |
| recalque 2|Evento 1 |
| < O:013 |   B9   |
| <---]/[-----] ( )-----+
| <   16   |   39   |

```

Rung 12:2

```

|          |          |          |          |          |
| Evento 1 |          |          |          |          |
|   B9   |          |          |          |          |
+----] [-----] ( )-----+
|   39   |          |          |          |          | |
| |executando|          |          |          |          |
| |TM2_TE1 |Evento 2 |          |          |          |
| |   B9   |   B9   |          |          |          |
| +----] [-----]/[-----]
|   42   |   43   |

```

Rung 12:3

```

|          |          |          |
| Evento 1 |          |          |
|   B9   |          |          |
+----] [-----] ( )-----+
|   39   |          |          | |
| |valvula |          |          |
| |TM2_TE1 |Evento 2 |          |
| | O:011 |   B9   |          |
| +----] [-----]/[-----]
|   11   |   43   |

```