

# Grafcet do Básico ao Avançado

## com Simatic S7 & TIA Portal

Eng. Luciano Cardoso Vasconcelos

Grafcet Básico com Simatic S7 & TIA Portal oferece aos leitores um ponto de partida para compreender sistemas de automação industrial, fundamentando conceitos de Grafcet. A capacidade de construir sistemas complexos é crucial para o desenvolvimento de um país. Este livro tem como objetivo desmistificar o processo de desenvolvimento de sistemas completos de automação industrial de processos - abrangendo coleta de dados, design, otimização, segurança e integração. O texto aborda toda a gama de conceitos relevantes para a engenharia geral em todos os setores e aplicações usando Grafcet. Os leitores aprenderão princípios básicos sobre o projeto de arquiteturas de programas de PLCs, estratégias de testes de plataforma emulando a planta, otimização de lógica, performance e muito mais. Além disso, considerações cruciais sobre sistemas em áreas como confiabilidade, privacidade, segurança e validação de soluções também são exploradas em profundidade. Em resumo, o livro se esforça para nivelar iniciantes e profissionais com conhecimento integrado que abrange o desenvolvimento de sistemas de PLC, usando o método de Grafcet facilmente acessíveis como veículo para transmitir conceitos universais necessários para a indústria.

### Table of contents

Prefácio .....	8
Por que escrevemos este livro .....	8
O que você precisa saber .....	9
Convenções de livros .....	9
Quer ajudar? .....	9
Contato .....	9
Colaboradores .....	9
Dedicação .....	10
Reconhecimentos .....	10
Colaboradores .....	10
Convenções .....	10
Direitos .....	12
Sobre o Livro .....	12
Visão Geral .....	12
Tópicos explorados .....	12
Quem deveria ler isto .....	12
Principais resultados de aprendizagem .....	13
Pré-requisitos para leitores .....	13

1. Introdução.....	14
1.1 Arquitetura do Sistema.....	14
1.1.1 Generalizando a Arquitetura do Sistema .....	14
1.2 Definição das Entidades .....	15
1.2.1 PLC.....	15
1.2.2 Driver .....	17
1.2.3 SCADA .....	18
1.3 Diferença entre as entidades de arquitetura do sistema.....	18
2. Hardware Siemens.....	19
2.1 Hardware de estudo (Simatic S7-300) .....	19
2.1.1 Módulos do Siemens - S7-300.....	20
2.1.2 CPU - S7-300.....	21
2.2 Ferramentas de programação.....	22
2.2.1 Tipos de línguas.....	22
2.2.2 Diagnóstico do processo .....	22
2.3 Distribuição dos dados do programa .....	22
2.3.1 Dados externos .....	22
2.3.2 Dados internos.....	23
2.3.3 Dados específicos internos .....	24
2.4 Resumo .....	24
3. Modularização .....	25
4. Organização .....	25
5. Endereçamento .....	25
5.1 Absoluto.....	25
5.2 Simbólico.....	25
5.3 Tabela de Símbolos .....	26
6. Configuração .....	28
6.1 Configuração de hardware .....	28
6.2 Endereçamento dos módulos S7-300 .....	28
6.3 Configuração Multi-rack.....	28
6.4 Propriedades da CPU .....	28
6.4.1 Cycle / Clock Memory .....	29
6.4.2 Geral ( Endereço MPI ) .....	29
6.5 Salvar e Download da configuração.....	29

6.6 Upload da configuração para a CPU.....	29
6.7 Propriedades da CPU.....	29
6.7.1 Startup ( Inicialização ).....	29
6.7.2 Memória retentiva.....	29
6.7.3 Protection.....	29
6.7.4 Diagnóstico / Relógio.....	29
6.7.5 Comunicação.....	29
6.8 Exercícios.....	29
7. Programação.....	29
7.1 Arquitetura de Blocos.....	29
7.2 Tipos de Blocos de Programa.....	30
7.3 Estrutura do Programa.....	30
7.4 Imagens de Processo.....	30
7.5 Execução Cíclica do Programa.....	30
7.6 Linguagens de programação ( LAD, FBD e STL ).....	30
7.6.1 Editores.....	30
7.6.2 Personalização dos editores.....	30
7.6.3 Criando um bloco no OB1.....	30
7.7 Operações com os blocos.....	30
7.8 Funções.....	30
7.9 Tipos de chamadas de Blocos.....	30
7.10 Exercícios.....	30
8. Lógicas Binárias.....	31
8.1 AND, OR.....	31
8.2 OU Exclusivo.....	31
8.3 Sensores e símbolos.....	31
8.4 Set e Reset.....	31
8.5 Flip-Flop.....	31
8.6 Bobina de atribuição intermediária.....	31
8.7 Detecção de flanco positivo e negativo.....	31
8.8 Detalhamentos dos registradores internos.....	31
8.8.1 Instruções que afetam o RLO.....	31
8.9 Exercícios.....	31
9. Controle do Programa.....	31

9.1 Jump incondicional.....	32
9.2 Jump condicional.....	32
9.3 Exercícios .....	32
10. Tipos de Dados.....	32
10.1 Inteiro .....	32
10.2 Duplo inteiro .....	32
10.3 Real .....	32
10.4 BCD.....	32
10.5 Formatos Numéricos.....	32
10.6 Armazenamento de dados .....	32
10.7 Operações de carga e transferência .....	32
10.8 Operações lógicas digitais.....	32
10.9 Funções de contador.....	33
10.9.1 Instruções básicas de contagem.....	33
10.10 Funções de temporizador.....	33
10.10.1 Formatos de tempo S5.....	33
10.10.2 Instruções básicas de temporização .....	33
10.11 Operações de comparação .....	33
10.12 Funções aritméticas Básicas.....	33
10.13 Operações de conversão .....	33
10.13.1 Inteiro para Real ( I -> DI ).....	33
10.14 Exercícios.....	33
11. Tipos Elementares.....	33
11.1 Inteiro .....	33
11.2 Duplo inteiro .....	34
11.3 Real .....	34
11.4 BCD.....	34
11.5 Formatos Numéricos.....	34
11.6 Armazenamento de dados .....	34
11.7 Operações de carga e transferência .....	34
11.8 Operações lógicas digitais.....	34
11.9 Funções de contador.....	34
11.9.1 Instruções básicas de contagem.....	34
11.10 Funções de temporizador.....	34

11.10.1 Formatos de tempo S5.....	34
11.10.2 Instruções básicas de temporização .....	34
11.11 Operações de comparação .....	34
11.12 Funções aritméticas Básicas.....	35
11.13 Operações de conversão .....	35
11.13.1 Inteiro para Real ( I -> DI ).....	35
11.14 Tipos de dados complexos .....	35
11.15 Exercícios.....	35
12. Defeitos e Diagnóstico.....	35
12.1 Categoria de Erros.....	35
12.2 Ferramentas de análise.....	35
12.3 Diagnóstico do sistema .....	35
12.4 Informações dos Módulos .....	35
12.5 Depurando o programa.....	35
12.6 Dados de referência.....	35
12.7 Estrutura do programa .....	36
12.8 Referência cruzada .....	36
12.9 Localização de instruções .....	36
12.10 Atribuições .....	36
12.11 Exercícios.....	36
13. Módulo analógicos.....	36
13.1 Módulos analógicos .....	36
13.2 Entrada analógica.....	36
13.3 Saída analógica .....	36
13.4 Tipos de medição.....	36
13.5 Propriedades dos módulos analógicos .....	36
13.6 Resolução do valor analógico .....	36
13.7 Endereçamento dos módulos analógicos.....	37
13.8 Convertendo os valores analógicos.....	37
13.9 Entrada analógico.....	37
13.10 Saída analógica .....	37
13.11 Exercícios.....	37
14. Introdução .....	37
14.1 Exemplos de Grafcet.....	37

14.1.1 “G7” de modo de operação .....	38
14.1.2 “G7” gestor do processo .....	39
15. Modularização.....	41
15.1 Gestão de I/O .....	41
15.1.1 Arquivos gerais externos dos programas.....	41
15.1.2 Gestor de indicação de status / entrada digitais do PLC .....	42
15.1.3 Gestor de cartas analógicas de entrada e saída.....	42
15.2 Gestão de programa .....	42
15.2.1 Gestor de mensagens para Supervisório .....	42
15.2.2 Gestor de mensagens para Supervisório .....	42
15.2.3 Gestor de interrupção cadenciada.....	43
15.3 Distribuição dos Arquivos dos Programas Principais.....	43
15.3.1 Gestores do modo dos programas principais .....	43
15.3.2 Gestor de intertravamento.....	43
15.3.3 Gestor de acionamentos .....	43
15.3.4 Gestor de defeitos .....	44
16. Organização .....	44
16.1 Chamadas dos blocos do programa.....	44
16.1.1 Chamadas dos programas principais .....	45
16.2 Quiz .....	46
Planta CIP .....	47
Objetivo .....	47
Descrição do Processo.....	47
Premissas e Exceção do projeto .....	48
Arquivos Fonte.....	49
Fontes dos Exercícios .....	49
Apêndice A: Configuração.....	49
A.1 OBs de Interrupção .....	49
A.1.1 Tipos de Interrupção OB .....	50
A.1.2 Interrupção Cíclica (OB30, OB38) .....	50
A.1.3 Interrupção de Hardware (OB40) .....	52
A.1.4 Interrupção por Tempo (OB20) .....	53
A.1.5 Interrupção de Inicialização (OB100) .....	53
A.1.6 Prioridade das Interrupções.....	53

A.1.7 Interrupções cíclicas no Portal TIA .....	54
A.2 OBs de Erros .....	56
A.2.1 OB121 - Erro de programação .....	56
A.2.2 OB122 - Erro de acesso periférico .....	56
A.3 References .....	56
Apêndice B: Ferramentas.....	57
B.1 Hardware Kits .....	57
B.1.1 Utilitários de Hardware .....	57
B.2 Ferramentas de Diagnóstico.....	57
B.2.1 Kits de Diagnóstico e Recuperação.....	57
B.3 IDEs e ambientes de desenvolvimento .....	57
Apêndice C: Instrumentação.....	58
C.1 Sistemas instrumentados de segurança (SIS).....	58
C.1.1 Lógica de votação .....	59
Apêndice D: Network.....	59
D.1 Endereços IP Privados ( Redes Locais ).....	60
D.1.1 Loopback - 127.0.0.0.....	60
D.2 Entidades de Rede.....	61
D.2.1 Roteadores.....	61
D.2.2 Switches .....	61
D.2.3 Default Gateway.....	61
D.3 Comandos úteis .....	61
D.3.1 ipconfig ( Windows ).....	61
D.4 Subnetting IPv4.....	62
D.4.1 Subnet Mask.....	62
D.4.2 Endereços Reservados .....	62
D.5 IPv6 .....	63
D.6 Troubleshooting.....	63
D.6.1 TCP/UDP open ports .....	63
D.7 Fontes de informação .....	64
D.7.1 Playlist - Your suck at subnetting .....	64
D.8 F.Q.A - Perguntas frequentes.....	64
D.8.1 Como você encontra o endereço IP dos seus dispositivos? .....	64
D.8.2 Como saber seu endereço IP.....	64

D.8.3 Desmistificando sub-redes.....	64
D.8.4 Gateway padrão.....	64
D.8.5 Desafio.....	64
D.8.6 Solução do desafio.....	64
D.8.7 Recapitulando.....	64
D.8.8 O que aconteceu com todos os endereços IP?.....	64
D.8.9 Quais são os intervalos das classes?.....	64
D.8.10 Quem divulgou todos esses endereços?.....	64
D.8.11 O que são redes sem classes.....	64
D.8.12 Isso me deixa louco (Classe D e E).....	64
D.8.13 O que é loopback.....	64
D.8.14 O que é o Ping?.....	64
D.9 References.....	64

## **Prefácio**

Bem-vindo ao Grafcet Básico com Simatic S7 & TIA Portal. Este livro é a sua porta de entrada para o mundo acelerado dos sistemas de automação industrial através do uso de PLCs.

Nosso objetivo é fazer deste livro de código aberto um esforço colaborativo que reúna insights de estudantes, profissionais e da comunidade mais ampla de profissionais da indústria. Queremos criar um guia completo que se aprofunde nos detalhes básicos dos sistemas de automação industrial e seus muitos usos.

*“Sabedoria é aplicar o conhecimento de forma sensata, com benevolência e comportamento ético.”*

Este não é um livro estático, mas dinâmico, um documento vivo. Estamos tornando-o continuamente atualizado para atender às necessidades em constante mudança deste campo dinâmico. Espere uma rica combinação de conhecimento especializado que o guiará pela complexa interação entre algoritmos e os princípios fundamentais que os fazem funcionar. Que este projeto seja uma fonte contínua para a comunidade.

## **Por que escrevemos este livro**

Estamos em uma época em que a tecnologia está sempre evoluindo. A colaboração aberta e o compartilhamento de conhecimento são os alicerces da verdadeira inovação. Esse é o espírito por trás do Grafcet Básico com Simatic S7 & TIA Portal. Estamos indo além do modelo tradicional de livro didático para criar um centro de conhecimento vivo.



O livro cobre princípios, algoritmos e estudos de caso de aplicações do mundo real, com o objetivo de fornecer uma compreensão profunda que o ajudará a navegar no cenário em constante mudança da indústria. Ao mantê-lo aberto, não estamos apenas tornando o aprendizado acessível; estamos convidando novas ideias e melhorias contínuas. Em suma, estamos construindo uma comunidade onde o conhecimento é livre para crescer e iluminar o caminho a seguir na tecnologia global de automação industrial.

## O que você precisa saber

Você não precisa ser um especialista em automação industrial para mergulhar neste livro. Tudo o que você realmente precisa é de um conhecimento básico de sistemas de automação industrial e uma curiosidade para explorar como hardware, software e linguagem de programação se unem. É aqui que a inovação acontece, e uma compreensão básica de como os sistemas funcionam será a sua bússola.

A proposta é desenvolver um projeto básico, aplicando o conhecimento compartilhado no livro, a fim de desmistificar para o leitor os métodos de desenvolvimento de aplicações em automação industrial utilizando quaisquer plataformas de PLC, independente do fabricante.

Pretendemos que esta seja uma experiência de aprendizado rica e gratificante.

## Convenções de livros

Para obter detalhes sobre as convenções usadas neste livro, consulte a seção [Convenções](#).

## Quer ajudar?

Se você estiver interessado em contribuir, você pode encaminhar um e-mail para [Eng. Luciano Vasconcelos](#).

## Contato

Tem dúvidas ou comentários? Sinta-se à vontade para enviar um e-mail diretamente para [Eng. Luciano Vasconcelos](#).

## Colaboradores

Um grande obrigado a todos que ajudaram a tornar este livro o que ele é! Junte-se a nós como colaborador!

## Dedicação

*A todos os jovens que se interessam por tecnologia e tenham a curiosidade de entender como as coisas funcionam. Mantenham essa chama acesa em seus corações para toda sempre.*

## Reconhecimentos

É uma jornada incrível montar este livro, abrangendo vários anos de trabalho árduo. A ideia inicial deste livro surgiu da falta de material didático sobre Grafcet, em português. Estamos profundamente gratos aos pioneiros neste assunto cujo trabalho inovador lançou as bases para este livro.

Em especial ao **engenheiro Luciano Cardoso Vasconcelos**, cuja crença no poder transformador das comunidades de código aberto e na orientação, e força de vontade para se dedicar a esse projeto.

Também devemos muito à equipe do GitHub. Vocês revolucionaram a forma como as pessoas colaboram, e este livro é um testemunho do que pode ser alcançado quando as barreiras à cooperação global são removidas.

A todos que se interessaram por este livro – obrigado! Escrevemos o mesmo pensando em vocês, na esperança de provocar reflexão, inspirar perguntas e talvez até acender uma centelha de inspiração. Afinal, de que adianta escrever se ninguém está lendo?

Por último, mas certamente não menos importante, nossos mais profundos agradecimentos aos nossos amigos, familiares, mentores e todas as almas gentis que nos apoiaram emocional e intelectualmente enquanto este livro se concretizava.

## Colaboradores

Estendemos nossos sinceros agradecimentos à todos os indivíduos que generosamente contribuíram com sua experiência, insights e tempo para aprimorar o conteúdo esse projeto.

Abaixo você encontrará uma lista de todos os colaboradores. Se você gostaria de contribuir com este projeto, consulte nossa página [GitHub](#).

ICTINUS-Engenharia

## Convenções

Convenções adotadas para o para este livro on-line:

1. **Estrutura e organização claras:**

- **Esboços dos capítulos:** Cada capítulo inicia com um esboço que fornece uma visão geral dos tópicos abordados.
  - **Numeração sequencial:** Numeração sequencial para capítulos, seções e subseções para facilitar a referência.
2. **Idioma acessível:**
- **Glossário:** Inclui um glossário que define termos técnicos e jargões
  - **Terminologia Consistente:** Terminologia ao longo do livro para evitar confusão.
3. **Auxiliares de aprendizagem:**
- **Diagramas e Figuras:** Uso de diagramas, figuras e tabelas para transmitir visualmente conceitos complexos.
  - **Barras laterais:** Barras laterais para informações adicionais, e notas para fornecer contexto do mundo real ao conteúdo teórico.
4. **Elementos interativos:**
- **Exercícios e Projetos:** Exercícios e projetos ao final de cada capítulo para estimular o aprendizado ativo e a aplicação prática dos conceitos.
  - **Estudos de caso:** Estudos de caso para fornecer uma compreensão mais profunda de como os princípios são aplicados em situações do mundo real.
5. **Referências e leituras adicionais:**
- **Bibliografia:** Incluir uma bibliografia ao final de cada capítulo para leitores que desejam se aprofundar em temas específicos.
  - **Citações:** Estilo consistente para citações, aderindo a padrões acadêmicos reconhecidos como APA, MLA ou Chicago.
6. **Materiais de suporte:**
- **Recursos on-line complementares:** Links para recursos on-line complementares, como palestras em vídeo, webinars ou módulos interativos.
  - **Conjuntos de dados e repositórios de código:** compartilhe conjuntos de dados e repositórios de código para prática, especialmente para seções que tratam de algoritmos e aplicativos.
7. **Feedback e envolvimento da comunidade:**
- **Fóruns e grupos de discussão:** Fóruns ou grupos de discussão onde os leitores possam interagir, fazer perguntas e compartilhar conhecimentos.
  - **Processo de revisão aberto:** Processo de revisão aberto, solicitando feedback da comunidade para melhorar continuamente o conteúdo.
8. **Inclusão e acessibilidade:**
- **Formatos Acessíveis:** Garantir todo uso de mídia, recursos da web como, Youtube, QR code e outros recursos, para adaptar o livro a diferentes formatos.
9. **Índice:**

- **Índice abrangente:** Índice abrangente no final do livro para ajudar os leitores a localizar rapidamente informações específicas.

A implementação destas convenções contribui para a criação de um livro didático que seja abrangente, acessível e propício a uma aprendizagem eficaz.

## Direitos

Este livro é de código aberto e desenvolvido de forma colaborativa por meio do GitHub.

Todas as marcas comerciais e marcas registradas mencionadas neste livro são de propriedade de seus respectivos proprietários.

As informações fornecidas neste livro são consideradas precisas e confiáveis. No entanto, os autores, editores e editores não podem ser responsabilizados por quaisquer danos causados ou supostamente causados, direta ou indiretamente, pelas informações contidas neste livro.

## Sobre o Livro

### Visão Geral

Bem-vindo a este projeto colaborativo iniciado pelo Engenheiro Luciano Cardoso Vasconcelos. O objetivo é tornar este livro um recurso comunitário que ajude educadores e alunos a compreender a técnica de Grafcet. O livro será atualizado regularmente para refletir novos insights sobre Grafcet e métodos de ensino eficazes.

### Tópicos explorados

Este livro oferece uma visão abrangente de vários aspectos do método Grafcet. Os tópicos que nos aprofundamos incluem:

- Nivelamento das entidades de Automação Industrial
- Introdução ao método de Grafcet aplicado em PLC
- Segmentação de dados em PLC
- Blocos de Função

Ao terminar este livro, você terá uma compreensão básica de Grafcet. Você ganhará experiência prática por meio de tarefas baseadas em projetos.

### Quem deveria ler isto

Este livro foi feito sob medida para aqueles que são novos em automação industrial, controle de máquina e processos usando PLC. Começa com os conceitos básicos de Grafcet e avança para tópicos mais avançados relevantes para a comunidade de engenheiros e técnicos que trabalham na indústria. O livro é particularmente benéfico para:

- **Engenheiros de sistemas embarcados:** Para engenheiros no domínio de sistemas embarcados, este livro serve como um excelente guia de automação industrial ( OT ), ajudando-os a criar aplicativos programas estruturados em PLCs.
- **Estudantes de Ciência da Computação e Engenharia Elétrica:** Este livro é um recurso útil para estudantes que estudam ciência da computação e engenharia elétrica. Ele apresenta os métodos, algoritmos e técnicas usando Grafset, preparando-os para desafios do mundo real para construção de sistemas estruturados em PLCs.
- **Pesquisadores e acadêmicos:** Aqueles envolvidos em pesquisas em automação industrial, IoT, sistemas embarcados acharão este livro esclarecedor.
- **Profissionais da Indústria:** Se você trabalha em áreas como IoT, robótica, tecnologia vestível ou dispositivos inteligentes, este livro fornecerá o conhecimento necessário de chão de fábrica.

## Principais resultados de aprendizagem

Os leitores adquirirão habilidades na estruturação de programas em controladores lógicos programáveis, PLC, desenvolvimento e implantação. Especificamente, você aprenderá sobre:

- Conceitos fundamentais de PLCs
- Fundamentos de Grafset
- Técnicas de estruturação de programas de PLCs

Nosso objetivo é fazer deste livro um recurso abrangente para qualquer pessoa interessada no desenvolvimento de aplicações em PLCs. Ao concluir o livro, você estará bem equipado para projetar e implementar seus próprios projetos para controle de processos e máquinas industriais.

## Pré-requisitos para leitores

- **Habilidades básicas de programação:** Recomendamos que você tenha alguma experiência anterior em programação de PLCs, de preferência em PLCs Siemens S7. Uma compreensão de variáveis, tipos de dados e estruturas de controle facilitará o envolvimento com o livro.
- **Disposição para aprender:** O livro foi elaborado para ser acessível a um público amplo, com diversos níveis de conhecimento técnico. A disposição de se desafiar e de se envolver em exercícios práticos o ajudará a tirar o máximo proveito disso.
- **Disponibilidade de recursos:** Para os aspectos práticos, você precisará de um computador com Simatic Manager ou TIA Portal.

Ao atender a esses pré-requisitos, você estará bem posicionado para aprofundar sua compreensão de Grafset, participar de exercícios e até mesmo implementar aplicativos práticos.

# 1. Introdução

O material a seguir baseia-se no material didático de treinamento da [Sitrain](#).

## 1.1 Arquitetura do Sistema

Para se estruturar um sistema de automação para controle de uma planta ou máquina é importante entender a organização da plataforma escolhida para o desenvolvimento da aplicação.

Observe a figura abaixo, a figura da esquerda representa a arquitetura de hardware, software e infra-estrutura que controlam a planta representada na figura da direita. Muitas vezes não é de escolha do desenvolvedor da aplicação a escolha da arquitetura, ou seja a mesma é imposta por inúmeras razões, pelo cliente, força de contrato, custos e etc.

Descrever as varias redes existentes em automação industrial e usadas pela Siemens comumente. Explorar a pagina 1 – 27 do manual da Siemens S7-TIA1 e o livro de Hans Berder para explicar um pouco sobre as redes usadas pela Siemens. Usar ainda a pagina 1 - 28 para explicar sobre a linha de HMI da Siemens. Usar ainda a pagina 1 – 29. Usar ainda a pagina 1 – 30. Usar a pagina 1 -31 para explicar sobre os acionamentos de motores.

São inúmeros os fabricantes de PLCs no mercado e cada um com sua própria linguagem e características, com exceção das comuns a todos os PLCs. Imagina o trabalho que não daria reprogramar toda uma aplicação, grande é claro, feita em um PLC Rockwell para um PLC Siemens. Devido as diferenças que abordaremos brevemente abaixo, o trabalho seria tão grande quanto o tamanho da aplicação.

Adotando-se então um método de padronização do programa, ou melhor, desenvolvimento da aplicação, reduz o tempo de migrar a aplicação de uma plataforma para outra, focando esforços então em redesenhar as funcionalidades mais especificas da aplicação.

~~Antes de começarmos a detalhar as características dos PLC, quero fazer um breve exercício com você~~

### 1.1.1 Generalizando a Arquitetura do Sistema

Imagine a mesma figura da arquitetura do sistema acima, porem agora vamos tirar os fabricantes e vamos utilizar o que vou tomar a liberdade de chamar de entidades, acrescentando um elemento que está implícito na figura acima, mas que é de muita importância, o driver.

São três entidades, PLC, Driver e o sistema SCADA ou IHM.

Como vimos existem vários fabricantes de PLC e de sistemas SCADA<sup>1</sup>. Cada fabricante de SCADA possui um driver<sup>2</sup> para cada tipo de PLC, como pode ser observado na figura acima. Pode-se usar ainda solução na arquitetura, OPC cliente/server<sup>3</sup> na camada do driver para resolver a comunicabilidade entre fabricante. ~~, enfim aonde quero chegar é, está escolha também pode ser imposta e com certeza vai gerar muita dor de cabeça para o desenvolvedor ou responsável pela manutenção.~~

~~Acho que você já deve estar compreendendo onde quero chegar. Para continuarmos esse assunto preciso esclarecer que diferenças são essas.~~

## **1.2 Definição das Entidades**

### **1.2.1 PLC**

Antes, algumas características básicas do PLC entre fornecedores.

É a unidade central de processamento, ou seja onde é executado o programa de controle da sua planta. Fazendo uma analogia, a unidade central de processamento dos programas que são executados no seu laptop ou desktop de casa é um Pentium ou Core i7, algo assim. O PLC é onde é executado o programa também e o resultado é apresentado na memória do mesmo, nas suas interfaces de entrada e saída, os cartões e/ou remotas.

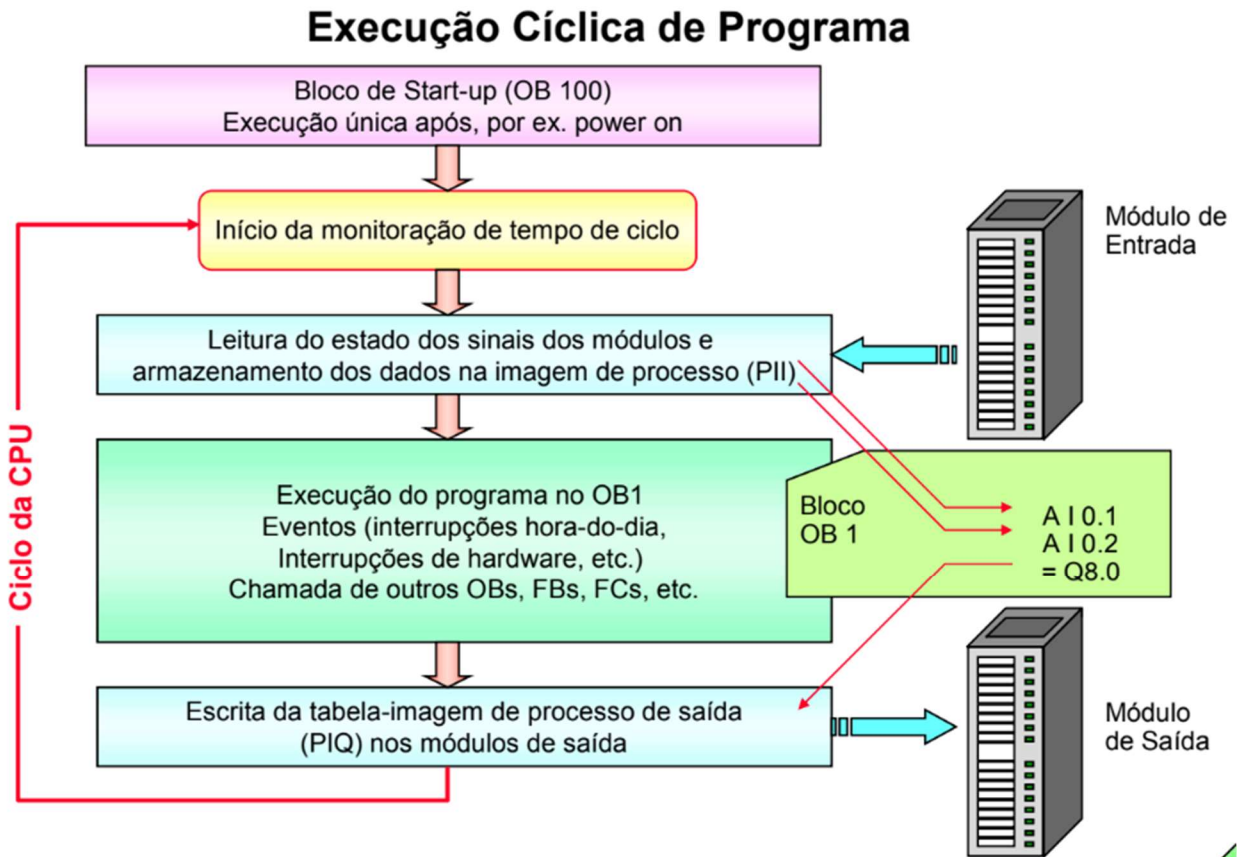
Abaixo, podemos ver o ciclo da CPU resumido, uma característica muito importante, até para depurar problemas de lógica e fundamental para compreender a construção estruturada de programas em PLC.

---

<sup>1</sup> SCADA – Acrônimo para supervisory control and data acquisition. É um sistema computadorizado para monitoramento e controle de processos industriais e de infraestrutura.

<sup>2</sup> Driver – Em nosso caso, software que permite o software supervisório acessar dados na memória do PLC. É o meio que o software supervisório faz uso para comunicar com o dispositivo do PLC.

<sup>3</sup> OPC – Acrônimo para OLE for process control. São padrões abertos especificados para comunicação de dados entre dispositivos de controle de diferentes fabricantes.

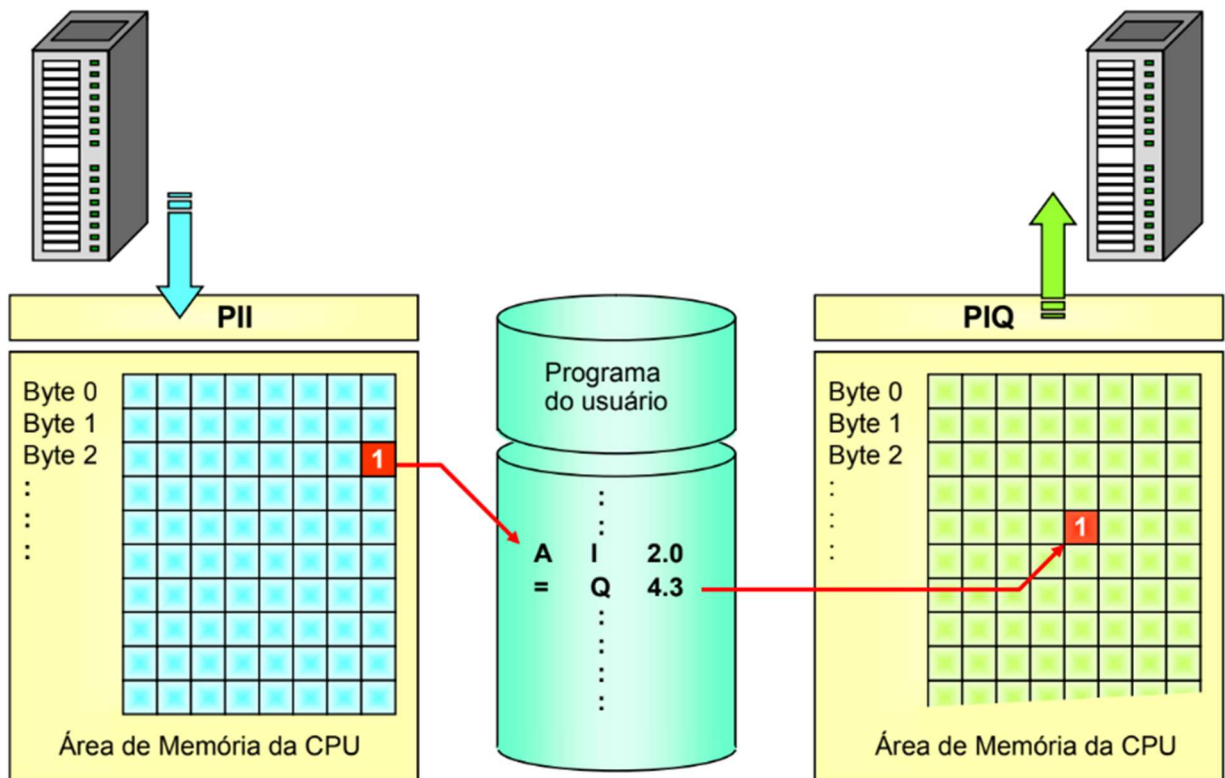


*Diagrama de execução cíclica do programa do PLC procedural. Nos vamos retornar a essa figura varias vezes ao longo do livro, assim a compreensão da mesma se torna muito importante.*

Analisando a figura, primeiro o firmware da CPU lê as entradas, o que chamamos de **PII (process-image input)** atualiza a tabela dos cartões de entrada na memória, executa o programa, o resultado do programa que afeta uma saída física é atualizado na tabela de saída na memória e só então o **PIQ (process-image output)** atualiza as saídas físicas nos cartões de saída conforme o ultimo estado da tabela de imagem das saídas. Segue, abaixo, uma outra figura que procura retratar esse ciclo.



## Imagens de Processo



*Imagem do processo do PLC.*

Isto não é uma verdade completa, alguns PLC mais atuais, Contrologix, se comportam diferente e também existe as aplicações específicas que utilizam recurso, funcionalidades, específicas para aplicações críticas. Prometo colocar algum caso mais para frente, mas esse não é nosso escopo.

O ciclo da CPU é uma característica comum entre os diferentes fabricantes desta entidade, PLC. Essas são características comuns entre PLC. Agora vejamos as diferenças entre dois fabricantes, Siemens e Rockwell, vamos estudar o caso de dois fabricantes desta mesma entidade, o PLC. Observe abaixo a tabela comparativa de diferenças entre o PLC da Rockwell e do PLC Siemens, como exemplo.

Tab 01, Mostrando o mapa de memória e endereçamento das mesmas.

Antes de iniciar qualquer observação, vamos continuar nossa análise das entidades da arquitetura do sistema exemplificada acima. Por hora, destaco essa diferença como mais importante, que é o mapa de memória entre os fabricantes.

### 1.2.2 Driver

Analisando a figura abaixo, podemos dizer que esta entidade da arquitetura do sistema é o software que coleta as informações da memória do PLC e disponibiliza para a entidade superior, que no nosso caso é o sistema SCADA, que vamos falar mais abaixo.

### ***Em Construção***

As informações desejadas que são coletadas são pré-configuradas no driver pelo integrador conforme necessidade.

Conforme dito acima, os fabricantes de driver coletam as informações pré-configuradas nos PLCs e disponibilizam para o sistema de nível superior, como exemplo um sistema supervisório. O driver coleta as informações dos endereços de memória, ou melhor, faz acessos ao mapa de memória dos PLC, disponibilizando as informações em tabelas de dados então chamadas de base de dados, para operações de escrita e leitura.

O que muda entre os fabricantes de driver é a forma como você configura o driver, ou seja seu uso, parametrização. Pode-se entender então que o driver é uma caixa preta para o integrador, o mesmo configura os endereços que quer acessar na memória do PLC e não se preocupa em como a coleta é realizada, apenas formata o tipo de dado que deseja coletar. O que você precisa saber são as diferenças nos mapas de memória entre os fabricantes de PLC e como a memória está sendo utilizada pela aplicação para que configure corretamente o driver, este último sim, de responsabilidade do desenvolvedor, partilhada entre o integrador do PLC e do sistema supervisório.

#### **1.2.3 SCADA**

Entende-se por estação supervisório e aquisição de dados. Ou seja, é o computador que coleta as informações do PLC, fazendo uso de um driver e disponibiliza as mesmas de forma gráfica, uma interface, para o usuário do equipamento através de um software supervisório.

Vamos analisar a figura abaixo.

### ***Em Construção***

Você precisa entender a diferença entre vista e scada, assim entenderá melhor o que é scada. A estação vista mostra as mesmas informações que o sistema SCADA, porém a diferença é que não possui driver, o vista coleta as informações do SCADA e este último usa um driver para coletar as informações no PLC.

A diferença entre sistemas SCADA é restrita ao software supervisório de cada fabricante. Quero dizer, suas funcionalidade de histórico, recursos de edição de telas, quantidade de penas para construção de históricos, dentre outras. A aplicação SCADA fala com o driver, que é basicamente uma caixa-preta que é configurada para acessar o mapa de memória dos diferentes tipos de PLC.

### **1.3 Diferença entre as entidades de arquitetura do sistema**

Agora que já entendemos cada entidade separadamente, podemos perceber que o objeto central do desenvolvimento, o driver, tem a função de “traduzir” os tipos de PLC para a aplicação de nível superior que em nossa arquitetura é um software supervisório, ou seja, ler e escrever as informações no mapa de memória do PLC.

Esta observação por si só já aponta para o objeto central do escopo do livro que é o PLC. Já que o driver é uma entidade configurada de acordo com o mapa de memória do PLC, assim não tem influência em como o mesmo está distribuído. ~~mas em nosso caso não é modificável pelo integrador, ou seja o desenvolvedor do controle automático de processo da aplicação.~~

Agora imagine o PLC, se não for adotado nenhum tipo de organização para construção da aplicação de controle no PLC pelas pessoas que desenvolvem e realizam a manutenção da planta, cada desenvolvimento usaria uma distribuição diferente, confundindo as pessoas que viessem a realizar a manutenção e/ou suporte.

Assim, nota-se a importância de se adotar algum tipo de metodologia para desenvolvimento e padronização para os diferentes tratamentos dados pelo programa as n funcionalidades do mesmo e com isso reduzir custos de manutenção e de futuras modificações de melhorias da planta.

Logo o primeiro objeto de nosso estudo é o padrão de segmentação do mapa de memória do PLC.

Você perceberá a importância do trabalho de segmentação de memória a medida que for apresentado a metodologia de construção do programa, Grafcet. A sequência dos passos do método são fundamentais para um bom desenvolvimento também. Gradualmente os passos serão utilizados ao longo do livro, empregando-se um projeto pratico, assim a compreensão da técnica se dará por completa.

### *Em Construção*

## **2. Hardware Siemens**

### *Em Construção*

1. Visão geral da família Simatic S7.
2. Familiarizar com o S7-300, ferramentas de programação, módulos disponíveis, ferramentas básicas.

### **2.1 Hardware de estudo (Simatic S7-300)**

Para uma melhor compreensão de como é o método de estruturação do programa, precisamos adotar uma configuração de hardware básica e seguir o estudo com base nessa configuração.

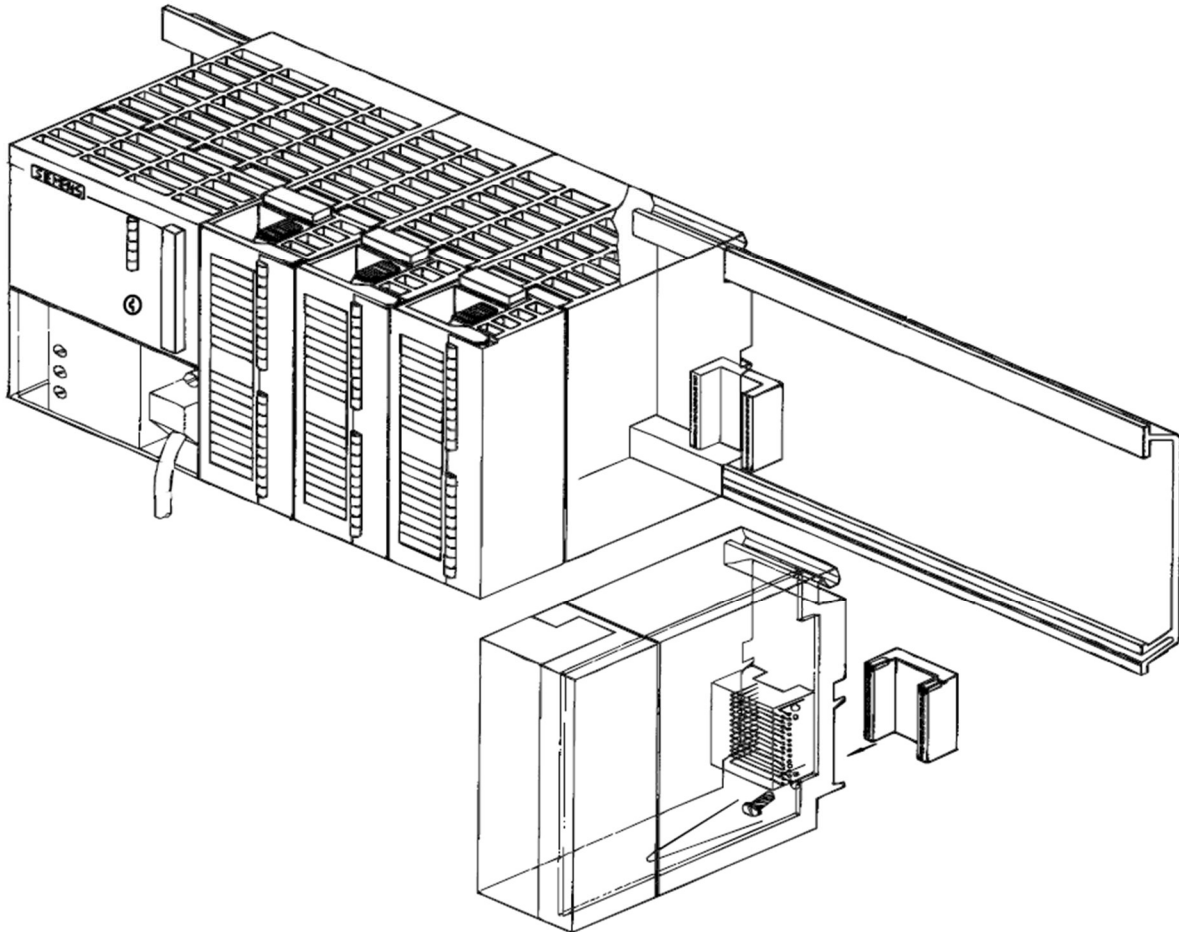
O método em si, compreendido, pode ser aplicado a qualquer arquitetura de sistema e fabricante de hardware, conforme especificação de memória de cada um.

Para nosso estudo, faremos toda a análise com base no PLC Siemens e faremos analogia, sempre que possível, com o PLC de médio porte da Allan Bradley. A família do hardware não importa muito, em geral a características dos PLCs de pequeno, médio e grande porte

das famílias de cada fornecedor é muito parecida. Refiro-me principalmente ao mapa de memória e ao set de instruções.

*Colocar a figura da página do S7-TIA1 — página 1-8 aqui e explicar resumidamente como funciona o hardware. Comentar as características que estão no slide do manual.*

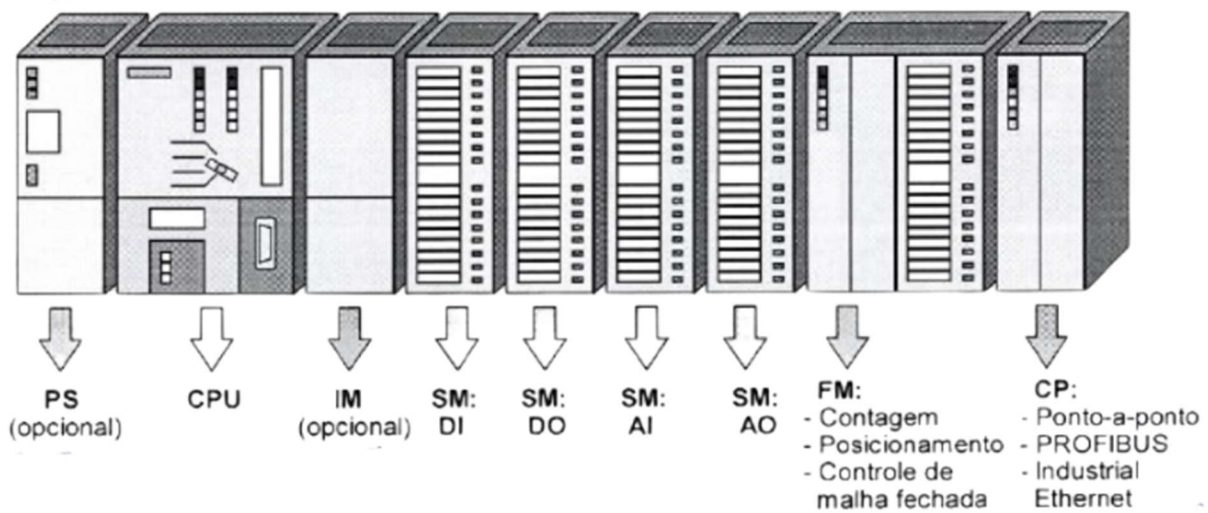
## S7-300



*Hardware S7-300*

### 2.1.1 Módulos do Siemens - S7-300

~~– Usar o manual da Siemens S7-TIA 1 página 1-9 para explicar cada módulo em especial. – Usar ainda o livro de automação de Hans Berger e outros manuais da Siemens para explicar os módulos e suas diferenças, essa aula serve para que o aluno compreenda o que cada módulo faz e ao que se destinam.~~



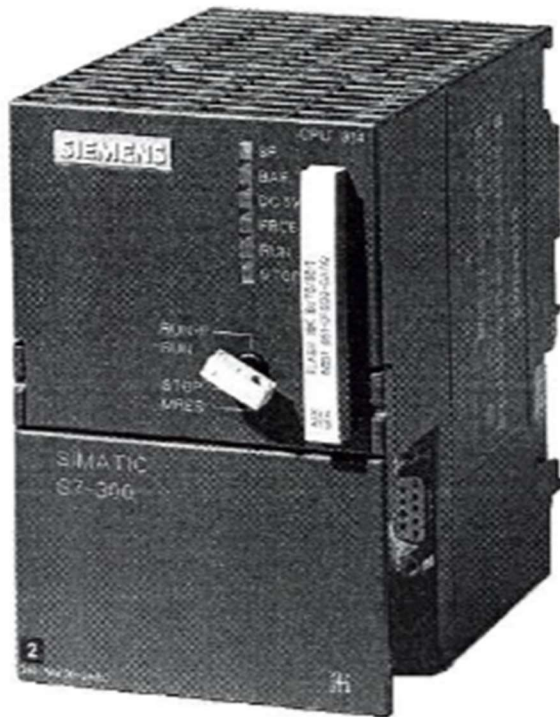
## Módulos do S7-300

### Em Construção

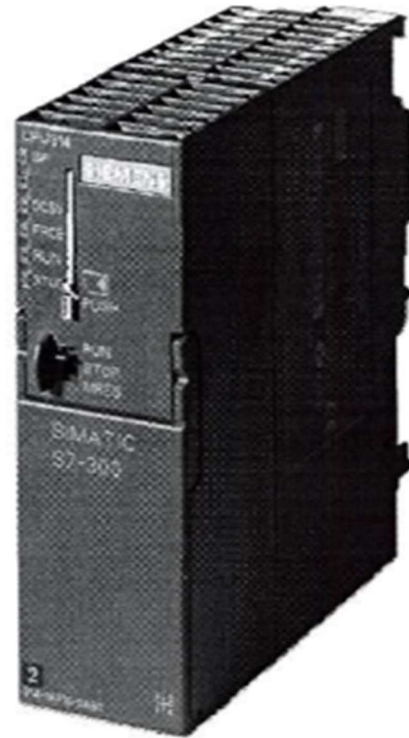
#### 2.1.2 CPU - S7-300

~~-Utilizar o manual da Siemens S7-TIA1 - página 1-10 e o livro de Hans Berger para explicar resumidamente cada CPU..~~

CPU 314 até Out. 2002



CPU 314 após Out. 2002



*CPU do S7-300*

## 2.2 Ferramentas de programação

*Comentar sobre a ferramenta de programação Simatic manager S7, usar o manual da Siemens S7 TIA1 páginas 1-14, 1-17, 1-18, 1-16, 1-19.*

### 2.2.1 Tipos de linguagens

*Comentar sobre os tipos de linguagens usando as páginas 1-20, 1-21, 1-22, 1-23.*

### 2.2.2 Diagnóstico do processo

*Comentar sobre o diagnóstico do processo usando as páginas 1-24, 1-25.*

## 2.3 Distribuição dos dados do programa

### 2.3.1 Dados externos

Primeiro, precisamos entender como é uma distribuição básica de dados do programa que provavelmente o desenvolvedor necessitará para realizar a integração do projeto.

Este é o primeiro passo para desenvolver uma boa solução usando grafcet. Veja as duas figuras abaixo, são as mesmas que você viu anteriormente, porém com mas detalhes para tratarmos o assunto de distribuição de dados.

### ***Em Construção***

Uma boa indicação de que dados poderíamos considerar como geral de programa está indicado na figura xa. Perceba que todo dado que é trocado entre o PLC e o supervisor e/ou IHM pode ser considerado como informação geral. Que informações seriam essas ?

1. **Entradas Digitais:** A indicação do status de chaves de fluxo, pressostatos, switch, ou mesmo das entradas dos cartões digitais do CLP.
2. **Alarmes:** Toda planta possui defeito, como exemplo, o térmico de uma bomba.
3. **Parâmetros de receita:** É comum nos depararmos com a necessidade de uma receita para operarmos uma planta industrial. Normalmente, são parâmetros de receita, setpoint de temperatura, de vazão, de temporizadores, dentre outras necessidades.
4. **Entradas e saídas analógica:** Uma planta industrial e/ou máquina muitas vezes possui transmissores de pressão, vazão, temperatura e outras. Essas informações normalmente são valores inteiros, como os indicados na figura da planta acima.
5. **Saídas digitais:** Existem também os comandos da planta, como comandos de acionamento de motores, válvulas, partida e parada de máquina e etc.

Por fim, observe a tabela de fluxo de dados abaixo.

### ***Em Construção***

Acabamos de identificar que dados gerais do programa, básicos, teremos que tratar quando formos segmentar o mapa de memória do PLC.

#### **2.3.2 Dados internos**

Para que você entenda o que podemos considerar como dados internos gerais, analise os dados novos que foram adicionados na figura abaixo.

### ***Em Construção***

Podemos concluir que são os dados cujo o escopo de uso é apenas interno ao programa e de uso comum no programa.

Contadores, temporizadores, bits de controle ( auxiliares ) são exemplos de dados comuns internos do programa.

Um dado interno pode sim ser transmitido para um dado exterior, para isso, por padrão, na técnica grafcet, seria necessário apenas copiar a informação do dado interno para a palavra externa, como a ilustração abaixo.

### ***Em Construção***

### 2.3.3 Dados específicos internos

Podemos dividir esses dados em dois tipos, basicamente.

~~Os dados que estão relacionados ao PLC que está sendo construído o programa e dados que estão relacionados com o método grafcet, que vamos detalhar mais para frente.~~

~~Como estamos detalhando todo o processo de estruturação de programa passo a passo, vou me limitar a mencionar, por enquanto, que existem dados específicos relacionados com o método grafcet.~~

~~Já que consideramos que estamos tratando o assunto com o PLC5 da Allan Bradley, podemos considerar dados específicos às palavras de controle tipo R, PD, MG e BT por exemplo. Essas palavras estão relacionadas com blocos de funções específicas disponíveis no set de instruções do PLC Allan Bradley.~~

~~Como disse antes, o livro é dedicado para como estruturar o programa. Estamos aprendendo o passo a passo para aplicar a técnica de estruturação grafcet, não o PLC necessariamente, logo recomendo que baixe os manuais do PLC5, por exemplo, e estude o mesmo.~~

~~Apenas para esclarecer a curiosidade, no PLC AB, tipos de dados MG são aqueles que estão relacionados com funções de transferência de dados ( MSG ), por exemplo transferência entre PLC em rede. Um PLC transfere informações para outro usando blocos que usam esse tipo de dados para configurá-los.~~

~~Dados tipos BT, são muito usados em funções usadas entre dispositivos AB, como um PLC e uma IHM AB.~~

~~Dados tipo PD são usados para configurar blocos PID.~~

~~Enfim, como pode perceber, teríamos que dedicar o livro a esse assunto. Você encontra fartos manuais para esclarecer dúvidas sobre PLC AB no site da Rockwell ([www.ab.com](http://www.ab.com)). Se baixar os manuais do set de instruções do PLC5, entenderá para que serve cada tipo de dados. Voltaremos a mencionar os tipos de dados quando iniciarmos a rascunhar o grafcet propriamente dito.~~

## 2.4 Resumo

O que realizamos nos itens anteriores foi o que chamamos de modularização.

Identificamos que existem indicação de status de chaves, pressostatos, informações de receita, indicação de variáveis de campo, alarmes, comandos e etc...

Identificamos também que esses são dados gerais de programa e que são encaminhadas para um supervisor ou mesmo uma IHM.

Após identificar essas informações o passo seguinte é modularizar o mapa de memória do PLC em que estamos trabalhando para estruturarmos o programa.



Igualmente importante, os dados gerais internos do programa são necessários devido a características específicas de cada PLC.

Antes de começarmos a trabalhar com o mapa de memória do PLC, precisamos entender como se divide os arquivos do PLC. Assim como os dados, os arquivos precisam ser estruturados de forma a padronizarmos a estrutura de chamadas e organização dos programas. Isto facilita a construção do mesmo e a manutenção do programa.

### 3. Modularização

*Em Construção*

### 4. Organização

*Em Construção*

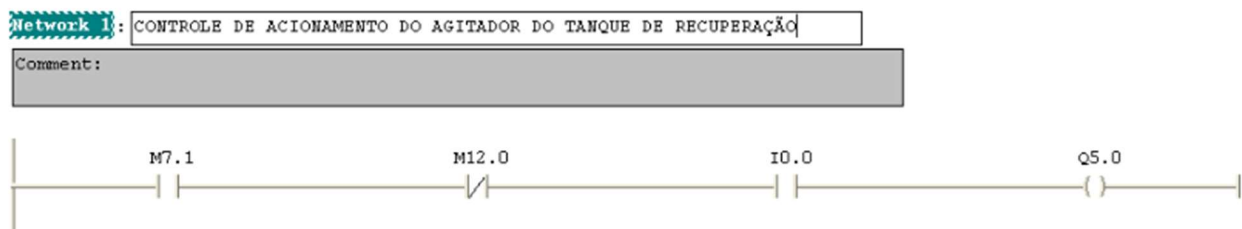
### 5. Endereçamento

*Em Construção*

#### 5.1 Absoluto

Entende-se por endereçamento absoluto, quando utilizamos ou melhor referenciamos o endereço de uma instrução diretamente a uma posição na memória do PLC, ou seja, não usando símbolos, porém a leitura e entendimento do programa torna-se de difícil leitura à medida que o mesmo aumenta em tamanho.

Veja abaixo exemplos de endereçamento absoluto.



*Endereçamento Absoluto*

*Em Construção*

#### 5.2 Simbólico

Orienta-se a utilização de comentários e símbolos em endereços absolutos de forma a tornar mais fácil o entendimento do programa por outras pessoas. Todas as informações textuais acrescentadas aos endereços absolutos de entrada e saída, temporizadores,

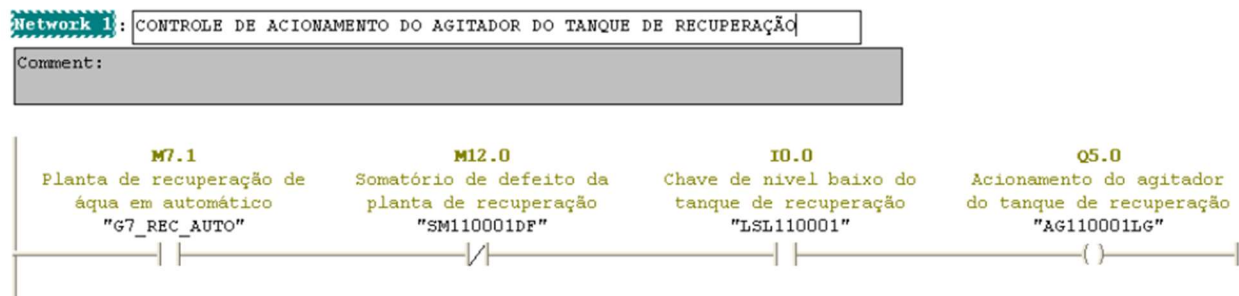
contadores, bit de memória, blocos de função e IO de periferia são armazenados dentro de uma tabela de símbolos administrada pelo próprio Simatic Manager.

Símbolo [Max. 24 Caract]	Endereço Absoluto	Tipo de Dado	Comentário [ Max. 80 Caracteres ]
G7_REC-AUTO	M7.1	BOOL	Planta de recuperação de água em automático.
SM110001DF	M12.0	BOOL	Somatório de defeito da planta de recuperação.
LSL110001	I0.0	BOOL	Chave de nível baixo do tanque de recuperação.
AG110001LG	Q5.0	BOOL	Acionamento do agitador do tanque de recuperação.

### Endereçamento Simbólico

#### Em Construção

Compare a linha acima, agora com símbolos e comentários de endereçamento. Acrescentando informações como símbolos, comentário de linha, título e comentário de instruções facilita o entendimento da lógica em análise, seja para a melhoria da mesma ou para a solução de problemas de manutenção.



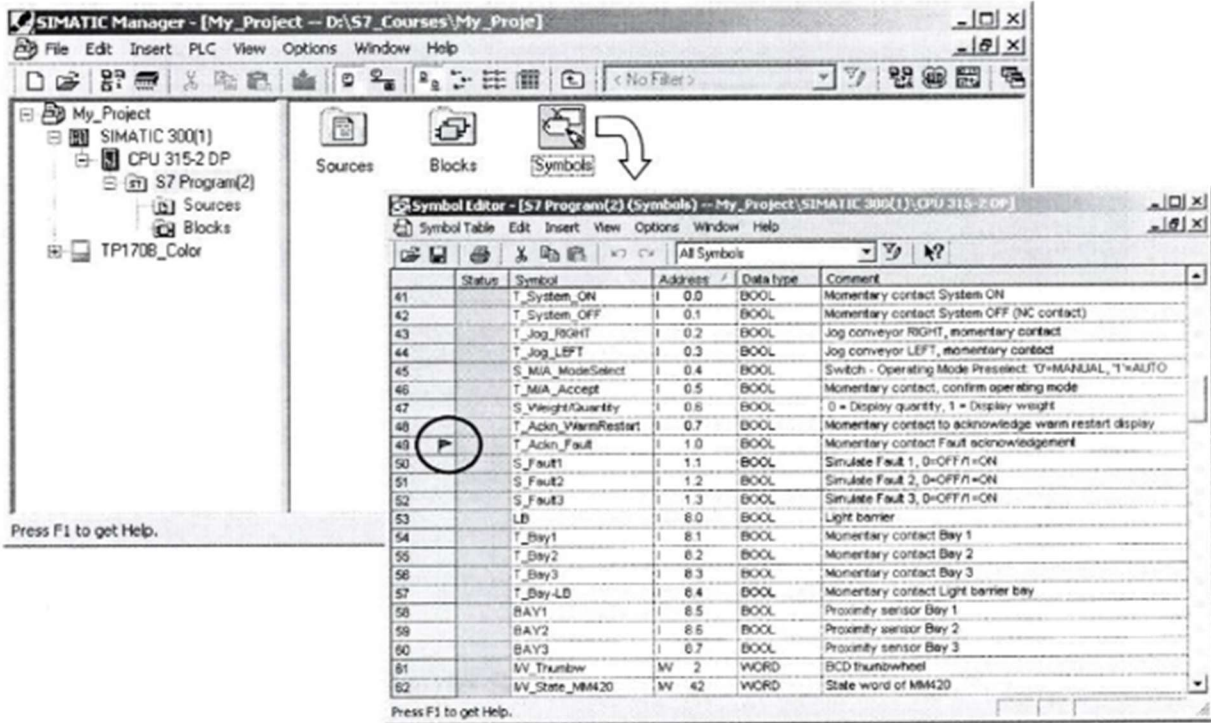
### Endereçamento Simbólico representado no Ladder

## 5.3 Tabela de Símbolos

Para se ter acesso a tabela de símbolos do projeto deve-se clicar em “S7 Program” na árvore do projeto conforme ilustrado abaixo. A tabela de símbolos é guardada pelo Simatic dentro desse diretório com o nome “Symbols”. Um clique duplo sobre este ícone abre a tabela de símbolos do projeto.

Qualquer alteração nessa tabela reflete automaticamente na atualização do programa.

A medida que os símbolos são digitados no editor LAD/STL/FBD, o Simatic atualiza automaticamente esta tabela, sem a interferência do programador, logo este é um repositório com as informações de comentários e símbolos do projeto.



*Tabela de Símbolos*

***Em Construção***

! Boas práticas na utilização de comentários

Comentar sobre a necessidade de se colocar o cabeçalho antes da primeira linha de cada bloco. Documentar principalmente as avaliações que houver no corpo de cada bloco.

OB1 : AGITADOR TANQUE RECUPERAÇÃO

```
-----  
Baustein      / module      / module      : FC112  
-----  
Bausteinbeschreibung:  Steuerung VG Zerhacker  
module description:   control VG chopper  
description du module:  
  
-----  
Version       / version       / version       : V1.0  
Datum         / date         / date         : 11.03.2002  
Programmierer / programmer    / programmeur   : Joachim Geiser  
-----  
Glatt D-79589 Binzen / D      +49-7621-6640  
      CH-4133 Pratteln / CH     +41-61-8264747  
-----  
V1.0 11.03.02 USED AT PROJECT "CN7315-7324 ROCHE BRAZIL"  
V0.1 19.10.2001 Original Version
```

**Network 1** : CONTROLE DE ACIONAMENTO DO AGITADOR DO TANQUE DE RECUPERAÇÃO

*Boas práticas: comentário de programa*

## 6. Configuração

*Em Construção*

### 6.1 Configuração de hardware

*Em Construção*

### 6.2 Endereçamento dos módulos S7-300

*Em Construção*

### 6.3 Configuração Multi-rack

*Em Construção*

### 6.4 Propriedades da CPU

*Em Construção*

#### **6.4.1 Cycle / Clock Memory**

*Em Construção*

#### **6.4.2 Geral ( Endereço MPI )**

*Em Construção*

#### **6.5 Salvar e Download da configuração**

*Em Construção*

#### **6.6 Upload da configuração para a CPU**

*Em Construção*

#### **6.7 Propriedades da CPU**

*Em Construção*

##### **6.7.1 Startup ( Inicialização )**

*Em Construção*

##### **6.7.2 Memória retentiva**

*Em Construção*

##### **6.7.3 Protection**

*Em Construção*

##### **6.7.4 Diagnóstico / Relógio**

*Em Construção*

##### **6.7.5 Comunicação**

*Em Construção*

#### **6.8 Exercícios**

*Em Construção*

### **7. Programação**

*Em Construção*

#### **7.1 Arquitetura de Blocos**

*Em Construção*

## **7.2 Tipos de Blocos de Programa**

*Em Construção*

## **7.3 Estrutura do Programa**

*Em Construção*

## **7.4 Imagens de Processo**

*Em Construção*

## **7.5 Execução Cíclica do Programa**

*Em Construção*

## **7.6 Linguagens de programação ( LAD, FBD e STL )**

*Em Construção*

### **7.6.1 Editores**

*Em Construção*

### **7.6.2 Personalização dos editores**

*Em Construção*

### **7.6.3 Criando um bloco no OB1**

*Em Construção*

## **7.7 Operações com os blocos**

*Em Construção*

## **7.8 Funções**

*Em Construção*

## **7.9 Tipos de chamadas de Blocos**

*Em Construção*

## **7.10 Exercícios**

*Em Construção*

## **8. Lógicas Binárias**

*Em Construção*

### **8.1 AND, OR**

*Em Construção*

### **8.2 OU Exclusivo**

*Em Construção*

### **8.3 Sensores e símbolos**

*Em Construção*

### **8.4 Set e Reset**

*Em Construção*

### **8.5 Flip-Flop**

*Em Construção*

### **8.6 Bobina de atribuição intermediária**

*Em Construção*

### **8.7 Detecção de flanco positivo e negativo**

*Em Construção*

### **8.8 Detalhamentos dos registradores internos**

*Em Construção*

#### **8.8.1 Instruções que afetam o RLO**

*Em Construção*

### **8.9 Exercícios**

*Em Construção*

## **9. Controle do Programa**

*Em Construção*

## **9.1 Jump incondicional**

*Em Construção*

## **9.2 Jump condicional**

*Em Construção*

## **9.3 Exercícios**

*Em Construção*

# **10. Tipos de Dados**

*Em Construção*

## **10.1 Inteiro**

*Em Construção*

## **10.2 Duplo inteiro**

*Em Construção*

## **10.3 Real**

*Em Construção*

## **10.4 BCD**

*Em Construção*

## **10.5 Formatos Numéricos**

*Em Construção*

## **10.6 Armazenamento de dados**

*Em Construção*

## **10.7 Operações de carga e transferência**

*Em Construção*

## **10.8 Operações lógicas digitais**

*Em Construção*



## **10.9 Funções de contador**

*Em Construção*

### **10.9.1 Instruções básicas de contagem**

*Em Construção*

## **10.10 Funções de temporizador**

*Em Construção*

### **10.10.1 Formatos de tempo S5**

*Em Construção*

### **10.10.2 Instruções básicas de temporização**

*Em Construção*

## **10.11 Operações de comparação**

*Em Construção*

## **10.12 Funções aritméticas Básicas**

*Em Construção*

## **10.13 Operações de conversão**

*Em Construção*

### **10.13.1 Inteiro para Real ( I -> DI )**

*Em Construção*

## **10.14 Exercícios**

*Em Construção*

## **11. Tipos Elementares**

*Em Construção*

### **11.1 Inteiro**

*Em Construção*

## **11.2 Duplo inteiro**

*Em Construção*

## **11.3 Real**

*Em Construção*

## **11.4 BCD**

*Em Construção*

## **11.5 Formatos Numéricos**

*Em Construção*

## **11.6 Armazenamento de dados**

*Em Construção*

## **11.7 Operações de carga e transferência**

*Em Construção*

## **11.8 Operações lógicas digitais**

*Em Construção*

## **11.9 Funções de contador**

*Em Construção*

### **11.9.1 Instruções básicas de contagem**

*Em Construção*

## **11.10 Funções de temporizador**

*Em Construção*

### **11.10.1 Formatos de tempo S5**

*Em Construção*

### **11.10.2 Instruções básicas de temporização**

*Em Construção*

## **11.11 Operações de comparação**

*Em Construção*

## **11.12 Funções aritméticas Básicas**

*Em Construção*

## **11.13 Operações de conversão**

*Em Construção*

### **11.13.1 Inteiro para Real ( I -> DI )**

*Em Construção*

## **11.14 Tipos de dados complexos**

*Em Construção*

## **11.15 Exercícios**

*Em Construção*

## **12. Defeitos e Diagnóstico**

*Em Construção*

### **12.1 Categoria de Erros**

*Em Construção*

### **12.2 Ferramentas de análise**

*Em Construção*

### **12.3 Diagnóstico do sistema**

*Em Construção*

### **12.4 Informações dos Módulos**

*Em Construção*

### **12.5 Depurando o programa**

*Em Construção*

### **12.6 Dados de referência**

*Em Construção*

## **12.7 Estrutura do programa**

*Em Construção*

## **12.8 Referência cruzada**

*Em Construção*

## **12.9 Localização de instruções**

*Em Construção*

## **12.10 Atribuições**

*Em Construção*

## **12.11 Exercícios**

*Em Construção*

## **13. Módulo analógicos**

*Em Construção*

### **13.1 Módulos analógicos**

*Em Construção*

### **13.2 Entrada analógica**

*Em Construção*

### **13.3 Saída analógica**

*Em Construção*

### **13.4 Tipos de medição**

*Em Construção*

### **13.5 Propriedades dos módulos analógicos**

*Em Construção*

### **13.6 Resolução do valor analógico**

*Em Construção*

## **13.7 Endereçamento dos módulos analógicos**

*Em Construção*

## **13.8 Convertendo os valores analógicos**

*Em Construção*

## **13.9 Entrada analógico**

*Em Construção*

## **13.10 Saída analógica**

*Em Construção*

## **13.11 Exercícios**

*Em Construção*

## **14. Introdução**

Inicialmente, detalharemos um pouco da história, algumas partes da norma IEC 848, como simbologia, sua representação em **LADDER** ou **STL**, regras que recomenda-se obedecer para evitar erros na execução do programa. Detalharemos ainda, boas práticas de construção de programa.

Após este nivelamento, um projeto básico de uma planta de CIP será utilizado como proposta de desafio de construção do mesmo, seguindo as premissas apresentadas nesta parte II.

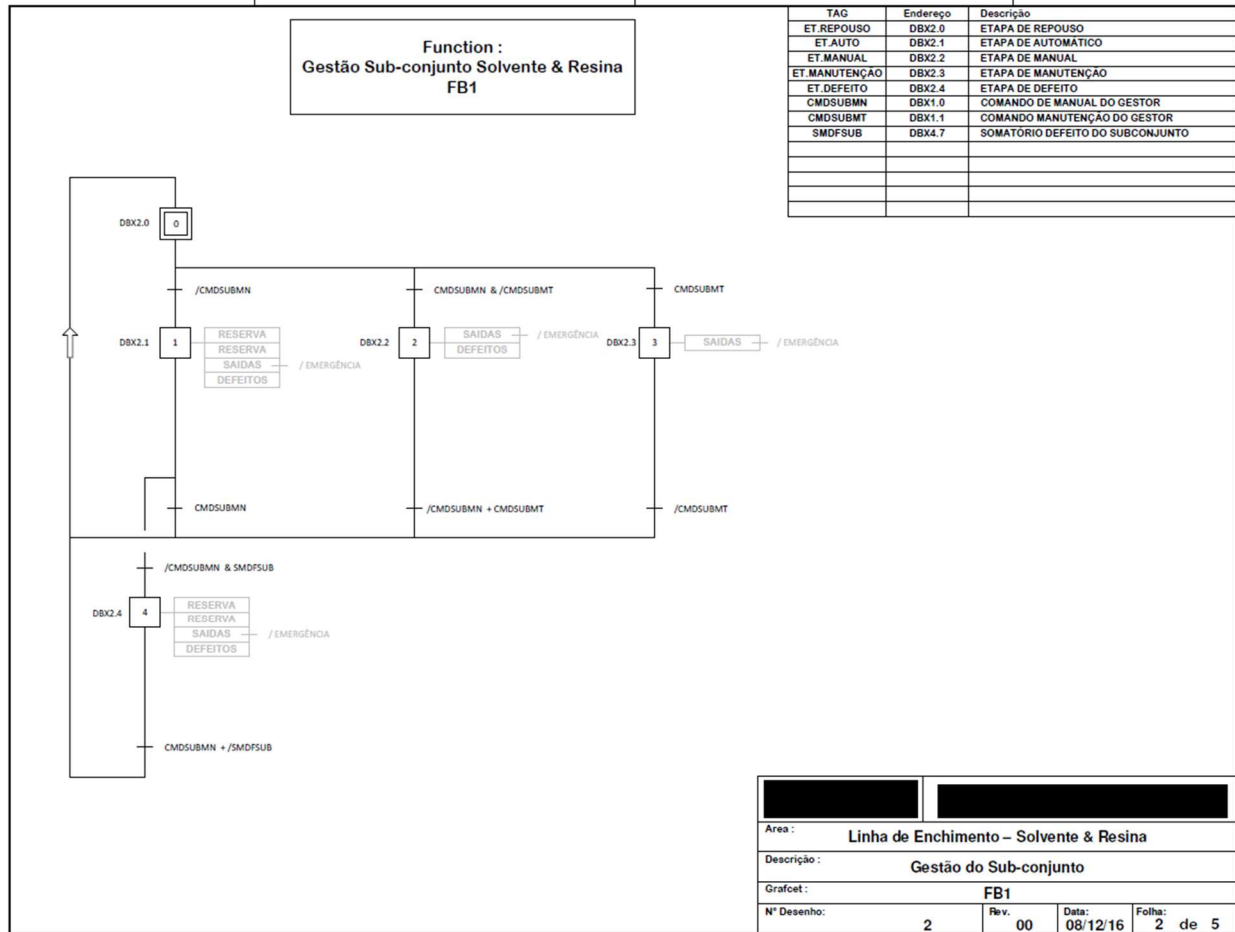
Importante ressaltar que os métodos de conversão de grafcet em seu equivalente LADDER ou STL aqui apresentados, não é a única forma de conversão. A linguagem STL do Siemens tem recursos suficientes para permitir a construção da lógica de diferentes formas.

*Em Construção*

### **14.1 Exemplos de Grafcet**

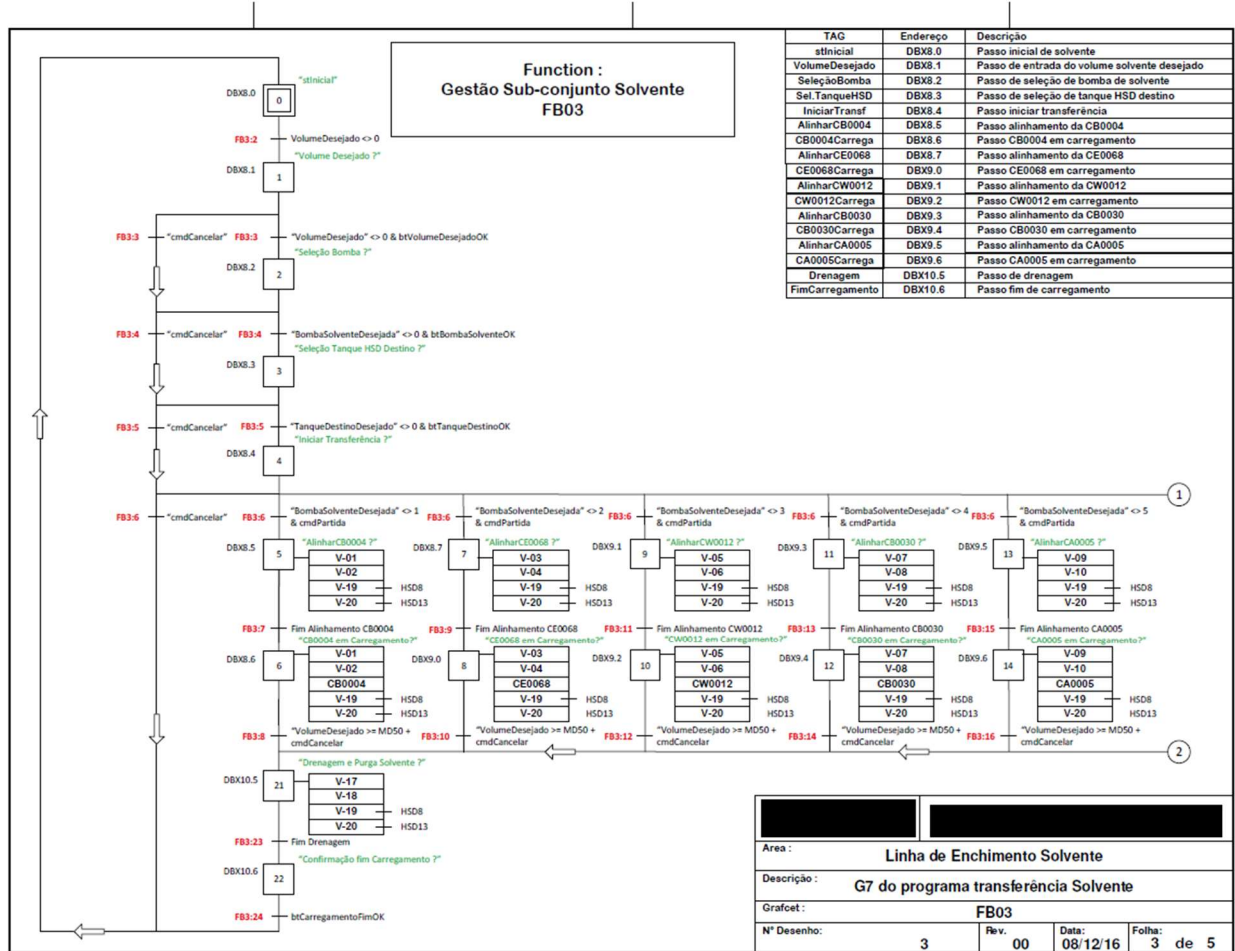
Abaixo, exemplos de representação de grafcets de gestão de modo de operação, bem como de controle do processo de uma planta de solvente e resina.

### 14.1.1 "G7" de modo de operação



*G7 de gestão de modo de operação, automático, manual e manutenção de uma planta de solvente e resina.*

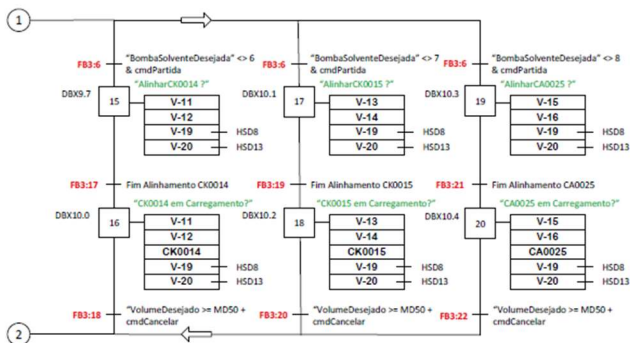
### 14.1.2 "G7" gestor do processo



G7 do programa gestor de controle da linha de enchimento de solvente.

**Function :  
Gestão Sub-conjunto Solvente  
FB03**

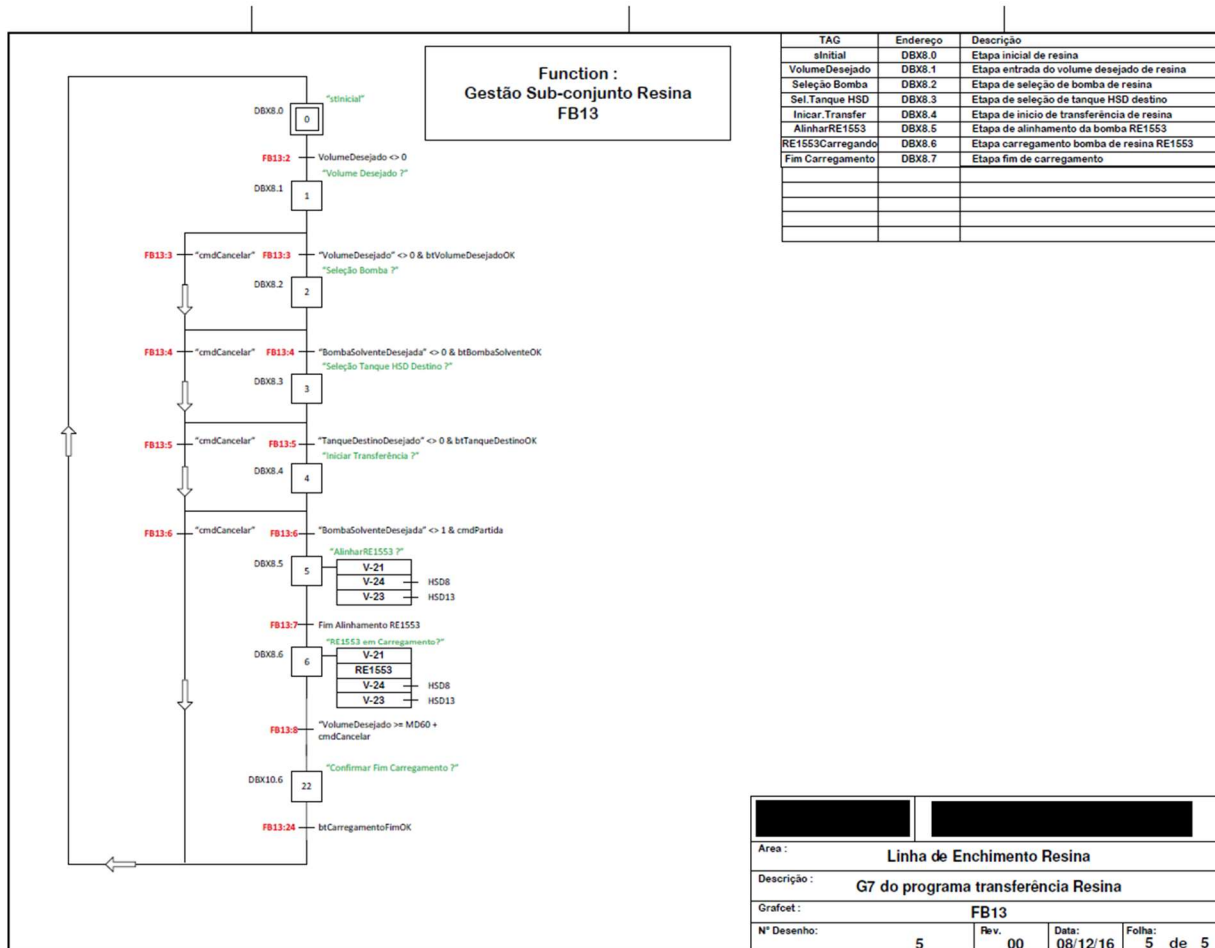
TAG	Endereço	Descrição
AlinharCK0014	DBX9.7	Alinhamento da bomba CK0014
CK0014Carrega	DBX10.0	CK0014 em carregamento de Solvente
AlinharCK0015	DBX10.1	Alinhamento da bomba CK0015
CK0015Carrega	DBX10.2	CK0015 em carregamento de Solvente
AlinharCA0025	DBX10.3	Alinhamento da bomba CA0025
CA0025Carrega	DBX10.4	CA0025 em carregamento de Solvente



Area : <b>Linha de Enchimento Solvente</b>			
Descrição : <b>G7 do programa transferência Solvente</b>			
Graficet : <b>FB03</b>			
Nº Desenho:	4	Rev.:	00
Data:	08/12/16	Folha:	4 de 5

*Continuação do programa gestor de controle de enchimento de solvente.*





*G7 do programa gestor de controle da linha de enchimento de resina.*

**! Confidencialidade**

Mantemos uma tarja preta em cima do nome do cliente atendido pela **ICTINUS**, afim de manter a confidencialidade do mesmo.

Usamos essa figura apenas como um exemplo de representação do grafcet de uma planta real.

## 15. Modularização

### Em Construção

#### 15.1 Gestão de I/O

##### 15.1.1 Arquivos gerais externos dos programas

Assim como os dados gerais dos programas, existem os arquivos gerais de programa. Vamos voltar a análise da figura abaixo:

## ***Em Construção***

Imagine as informações de forma modularizada.

É muito mais fácil reunir todos os dados gerais do programa em arquivos gestores diferenciados por funções.

### **15.1.2 Gestor de indicação de status / entrada digitais do PLC**

Normalmente não se configura o driver para ler as entradas digitais diretamente. O que se faz é tratar as entradas antes de usar no programa e *~~gerando .... transmitir* para o supervisor usar em arquivo de dados único.

Assim podemos pensar que é uma boa forma de tratar os dados de entrada com um único arquivo gestor.

### **15.1.3 Gestor de cartas analógicas de entrada e saída**

É comum termos que tratar os dados analógicos de entrada e saída do programa. Por exemplo, às vezes os cartões convertem os sinais analógicos para um número inteiro que pode ser representado em 12 bits, ou seja de 0 a 32767.

Assim, às vezes são dados tratamentos a essas informações para escalarmos para a unidade de engenharia que se quer trabalhar.

## **15.2 Gestão de programa**

*Reúne os programas, ou seja os graficets, do equipamento. Por exemplo, suas funcionalidades como, enchimento, agitação, esvaziamento, resfriamento e etc ...*

## ***Em Construção***

### **15.2.1 Gestor de mensagens para Supervisor**

Arquivo gestor utilizado para inicializar as variáveis para que garantir que as variáveis de programa tenham um valor inicial e o ressete do programa para a posição inicial sempre que o PLC é inicializado.

Esta medida evita acidentes, ou seja, é uma forma de garantir a segurança da planta sempre que o PLC é iniciado.

Este arquivo é executado apenas na primeira varredura do PLC.

### **15.2.2 Gestor de mensagens para Supervisor**

*Responsável por encaminhar o resultado de um cálculo para o supervisor com o valor indicado em um instrumento de campo, e etc...*

## ***Em Construção***

### **15.2.3 Gestor de interrupção cadenciada**

Arquivo utilizado para executar a parte do programa utilizada com as funções de interrupção disponíveis do PLC.

Muitos PLC têm interrupções de hardware e/ou tempo. Logo é necessário prever arquivos gestores para a execução dessas partes do programa.

As interrupções de hardware são conhecidas como PII e as de tempo são as STI ( selectable time interrupt ).

Explicar o que é first scan em apêndice A.

#### ***Em Construção***

### **15.3 Distribuição dos Arquivos dos Programas Principais**

Os programas principais estão associados aos equipamentos e todos os equipamentos possuem gestores de modo, de programas, intertravamentos, acionamentos e defeitos.

Podem existir outros, mas vamos trabalhar com eles.

#### ***Em Construção***

#### **15.3.1 Gestores do modo dos programas principais**

Os programas principais e/ou equipamentos possuem modos de operação manual / Automático, manutenção e defeito.

Esses modos geram a execução dos programas do equipamento, acionamentos da plantam, nos intertravamentos de segurança e operacional.

#### ***Em Construção***

#### **15.3.2 Gestor de intertravamento**

Reúne os intertravamentos dos equipamentos que vão agir diretamente nos programas e garantir o correto funcionamento do mesmo e/ou segurança operacional da mesma.

#### **15.3.3 Gestor de acionamentos**

Reúne todos os acionamentos do equipamento. Reunidos em um único arquivo facilita o gerenciamento de acidentes.

Por exemplo, vamos supor que alguém aperte a parada de emergência do equipamento, podemos desabilitar a varredura do arquivo de acionamento associado e garantir via programa que todos os acionamentos sejam desabilitados, evitando assim acidentes.

Você verá como implementar isso mais a frente.

#### **15.3.4 Gestor de defeitos**

Reúne todos os defeitos do equipamento. Assim fica muito mais fácil gerir os defeitos do equipamento e mesmo realizar as manutenções do programa.

## **16. Organização**

### *Em Construção*

#### **16.1 Chamadas dos blocos do programa**

Vamos analisar novamente a figura abaixo:

### *Em Construção*

O programa do PLC é dividido em arquivos que por sua vez possuem programas que manipulam a memória do mesmo.

Estamos estruturando o programa de forma modularizada. Logo, vamos rever aos poucos o que vimos no capítulo 3.

Vimos que ao ligar o PLC existe um arquivo, na estruturação que estamos fazendo, que inicializa as variáveis do programa que são importantes para garantir uma posição inicial de partida do programa ou ressete de alarmes, enfim todo e qualquer dado que seja importante ser inicializado.

É correto pensar que poderíamos pensar na figura assim como :

### *Em Construção*

Seguindo o mesmo raciocínio, todas as informações importantes para a execução do mesmo devem ser executadas antes dos graficets de controle da planta. Continuando a mesma linha de raciocínio, se um valor de cartão analógico não for tratado antes da execução dos graficet, isto poderia causar o funcionamento errado das evoluções dos programas. Logo este também é um arquivo gestor que deve ser tratado antes dos arquivos gestores de programa, assim todo e qualquer arquivo gestor que incorra na mesma situação.

Assim, analise a mesma estrutura de chamada de arquivos com a sequência que iremos trabalhar.

### *Em Construção*

Observe o quadrado em separado do gestor de interrupção. Este não faz parte da execução procedural do programa por motivos óbvios. O mesmo é executado apenas quando ocorre a interrupção.

A outra parte em destaque é o gestor dos programas principais. Este deve ser analisado em separado conforme veremos.

Até aqui já conseguimos rascunhar as chamadas dos arquivos gestores de dados gerais do programa.

Esta organização não depende do programa, pois a organização é você quem constroi no PLC escolhido. É muito mais fácil dar manutenção em um programa modularizado por funcionalidade conforme acabamos de fazer, duvida ?

Bom, suponha que um valor analógico esteja mostrando um valor duvidoso ou diferente do que está sendo mostrado no campo, onde você vai procurá-lo no programa ? Onde ele está sendo escalado no programa ?

Com as informações que acabamos de estudar, você tem condições de responder essa pergunta, mas vamos te dar uma ajuda, está no arquivo gestor de cartas analógicas do programa.

### **16.1.1 Chamadas dos programas principais**

Volte ao capítulo 3, item 3.2.2. Quais são os gestores do programa principal ?

São:

- Gestores de condições iniciais;
- Gestores de programas;
- Gestores de intertravamentos;
- Gestores de acionamentos;
- Gestores de defeitos.

Vamos focar na organização das chamadas desses gestores.

Seguindo o mesmo princípio de organização de chamadas apresentado acima, devemos chamar todos os gestores que são condições iniciais antes dos gestores de programa. Logo :

#### ***Em Construção***

Após a verificação das condições iniciais, os gestores de programa podem ser chamados e os gestores de intertravamentos da planta podem ser chamados após os gestores de programa. Os gestores de acionamentos do equipamento dependem do estado do programa e dos intertravamentos da planta, logo devem ser chamados após os gestores de intertravamentos.

Por fim, os gestores de defeito do equipamento devem ser chamados por último como resultado de todas as operações anteriores.

Nosso diagrama de chamadas de arquivos fica assim :

#### ***Em Construção***

Até o momento não estamos em Grafcet propriamente dito, mas estamos modularizando as chamadas de arquivo de programa. Modularização aplicada a PLC e de forma amigável, gráfica, de fácil compreensão.

Esta técnica ajuda nas implementações dos programas no PLC, ou seja na construção e manutenção do mesmo. Grafcet faz uso dessa mesma técnica para um melhor entendimento e padronização dos programas.

Perceba que a estrutura dos chamados dos arquivos nada tem a haver com o PLC em que estamos estudando.

A sequência dos chamados modularizados conforme fizemos facilita a organização da estruturação geral do programa e a modularização que acabamos de ver pode ser aplicada para solucionar qualquer caso real em qualquer modelo de PLC.

*Agora já temos subsídios para começar a aplicar essa estrutura de forma prática no PLC em que estamos estudando e quando descrevermos a técnica grafcet, verá como fica fácil ...*

### ***Em Construção***

#### **16.2 Quiz**

Como fazer uma simulação de sinal, enganando o PLC na estrutura que acabamos de realizar?

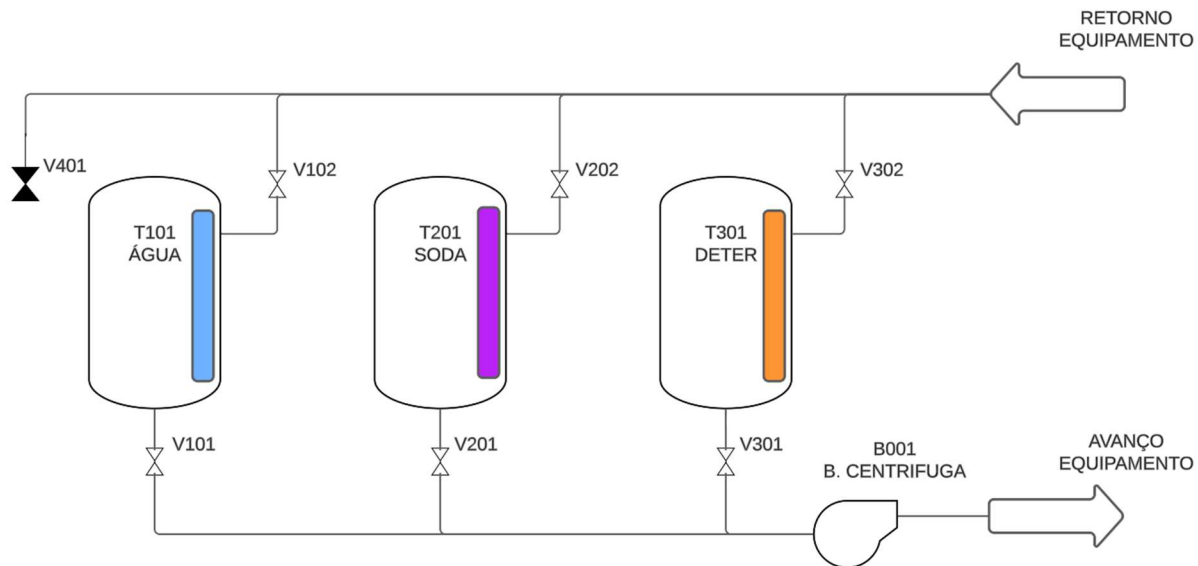
*A diferença principal é o plc, explicar o opc e voltar a questão da padronização com o estudo do principal no plc que é a memória do mesmo.*

*Observar que não necessariamente o supervisor tem que existir, pode ser uma IHM ou outro tipo de recurso, sinótico talvez.*

*Falar sobre os diferentes tipos de processos de batelada, dinâmico e fases e equipamentos.*

*Comentar sobre a aplicabilidades do grafcet, para o que ele serve, as novas aplicações.*

## Planta CIP



*Exemplo de uma planta de CIP de fornecimento de água, soda e detergente para limpeza de máquinas ou planta industriais, que deve obedecer uma determinada sequência pré-estabelecida, conforme descrito abaixo.*

### Objetivo

Automatizar a planta de fornecimento de CIP, ou seja, água, soda e detergente, em uma sequência pre-estabelecida, conforme descrição de processo abaixo.

A planta deve ter 3 modos de operação, sendo:

- **Modo Automático:** Neste modo, o programa controla os acionamentos da planta, obedecendo a sequência abaixo, uma vez recebido um pedido de CIP. Ainda neste modo, caso um dos equipamentos entre em falha, ou se o operador mudar o modo de operação, os acionamentos devem ser retirados da planta.
- **Modo Manual:** A planta deve obedecer comando do operador vindos de uma IHM ou Supervisório, como exemplo. Mesmo nesse modo, caso ocorra um alarme, devido ao não funcionamento de algum elemento final, o mesmo deve ser apresentado na IHM / Supervisório.
- **Modo Manutenção:** Modo utilizado pela departamento de manutenção da planta para fazer testes de acionamento da planta. Assumiremos que o mesmo não causa alarmes na planta.

### Em Construção

### Descrição do Processo

Uma explicação resumida do funcionamento da planta de CIP, e que serve como proposta de exercício para aplicação de grafset e seu entendimento prático.

- **Repouso:** Assumir que inicialmente, a planta de CIP encontra-se em repouso. Neste estado, todos os equipamentos, incluindo válvulas encontram-se desligados e o programa de controle de fornecimento de CIP está em sua posição de repouso, aguardando um pedido de fornecimento de CIP, vindo de outro equipamento, ou seja, o mesmo aguarda um sinal externo;
- **1st Etapa - Primeiro fornecimento de água:** A primeira etapa do programa, primeiro fornecimento de água potável, uma vez acionado após um pedido de CIP externo. Nesta etapa, assumiremos que as válvulas V101, V102 e a bomba B001, serão acionadas. O programa fica nesta etapa por 1 min, tempo este que deve ser parametrizado, antes de passar para a etapa seguinte. *Ver nota abaixo.*
- **2nd Etapa - Fornecimento de Soda:** Segunda etapa do processo, fornecimento de soda, devem ser acionados somente as válvulas V201, V202 e a Bomba B001. O programa fica nesta etapa por um tempo determinado por receita, como sugestão 1 min.
- **3rd Etapa - Fornecimento de Detergente:** Passo utilizado para neutralizar a soda em um caso real. Neste passo, somente a válvula V301, V302 e a Bomba B001 devem ser acionadas. A fase obedece também um tempo mínimo de 1 min, parametrizável.
- **4th Etapa - Fornecimento de água:** Segundo fornecimento de água, repetindo o acionamento da primeira etapa e obedecendo o tempo mínimo de 1 min, conforme condições anteriores.

Após a execução do tempo da última fase de água, o programa deve retornar para a posição de retorno e aguardar um novo pedido de CIP.

*Em Construção: considerar fazer um vídeo explicando o projeto*

### Premissas e Exceção do projeto

#### ⚠ Premissas e exceções do projeto

A fim de simplificar o programa de controle e descomplicar a implementação do mesmo, vamos assumir as seguintes premissas abaixo:

- **Golpe de aríete:** Vamos assumir que a planta não tem golpe de aríete.
- **Condutivímetro:** Neste exercício, não vamos considerar um condutivímetro para fazer as viradas do retorno para o tanque correspondente.
- **Emergência:** Caso seja batida a emergência da planta, ou seja, botoeira de emergência no campo por algum motivo, o programa deve parar sua execução no passo que estiver, desligar todas as saídas e sinalizar na IHM / Supervisório.
- **Tempo das fases / etapas:** O tempo de 1 min proposto no desafio do projeto é apenas uma sugestão, já que a ideia é rodar o programa simuladamente, e não é objetivo do projeto um tempo suficiente para garantir a limpeza de um equipamento real.
- **Válvula Dreno:** Esta proposta de projeto inicial não fará uso da válvula de dreno, uma vez que a lógica da mesma depende do uso do condutivímetro.



## Em Construção

## Arquivos Fonte

Arquivos fonte dos exercicios e exemplos propostos no Livro.

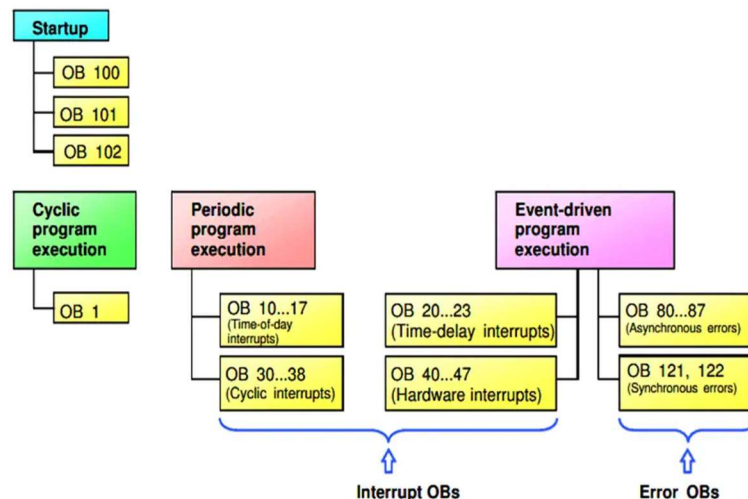
### Fontes dos Exercícios

No	Arquivo	Descrição	Especificação
1			

## Apêndice A: Configuração

Coletânea de informações sobre configuração de PLC, necessário para otimizar os recursos do PLC afim de atender as necessidades do projeto.

### A.1 OBs de Interrupção



*Árvore de Interrupções: Ilustração da árvore de interrupções da Siemens, como exemplo.*

Esta palavra não é exclusividade da Siemens, interrupções cíclicas são algumas rotinas específicas e críticas que o programador deseja que sejam executadas em um determinado horário e evento, por [`@intCiclicaOB30`].

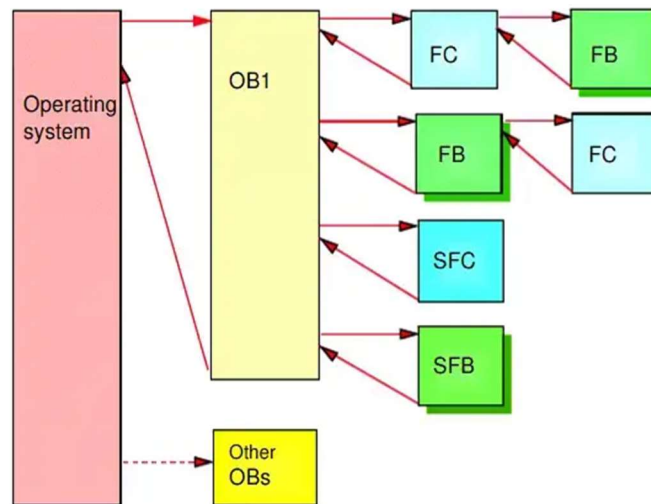
As interrupções não esperam que o programa principal termine seu ciclo. Uma vez que as interrupções ocorrem, ele para o OB principal ( **OB1** ) no ponto de interrupção e executa sua rotina, então o OB principal retoma o programa do ponto de interrupção novamente.

Dominar o uso do recurso de interrupção é importante para solucionar problemas práticos que somente o uso das mesmas resulta em solução confiável e eficiente.

### A.1.1 Tipos de Interrupção OB

Existem quatro tipos de interrupção OB usados, listados abaixo:

1. **Interrupção Cíclica:** OB30, OB38
2. **Interrupção de Erro de Hardware:** OB40
3. **Interrupção da Hora do Dia:** OB20
4. **Interrupção de Inicialização:** OB100



No CLP Siemens, para executar os programas do CLP é utilizado o Bloco Organizacional. No OB1 que é utilizado para executar lógica e chamamos outros blocos de programação deste OB1. Em muitas aplicações pequenas requer apenas o bloco OB1. Mas na aplicação grande, você também precisa usar outros blocos de organização que a Siemens oferece no software de programação.

### A.1.2 Interrupção Cíclica (OB30, OB38)

A interrupção cíclica é usada quando há um requisito de execução cíclica. A ideia por trás dessa interrupção é que você pode configurar uma delas para ser executada com determinados tempos cíclicos que independem do tempo do ciclo de varredura do seu programa principal.

As interrupções cíclicas no portal TIA geralmente têm a prioridade de (30: 38) que depende da CPU que você possui. Nem todas têm a mesma interrupção cíclica, algumas CPUs têm apenas (32: 35) outras CPUs podem ter outro grupo de interrupções do ciclo.

O programa principal OB1 para sua execução toda vez que uma interrupção cíclica é executada, e o programa volta ao ponto da chamada da interrupção no bloco principal ao final da execução da interrupção cíclica.

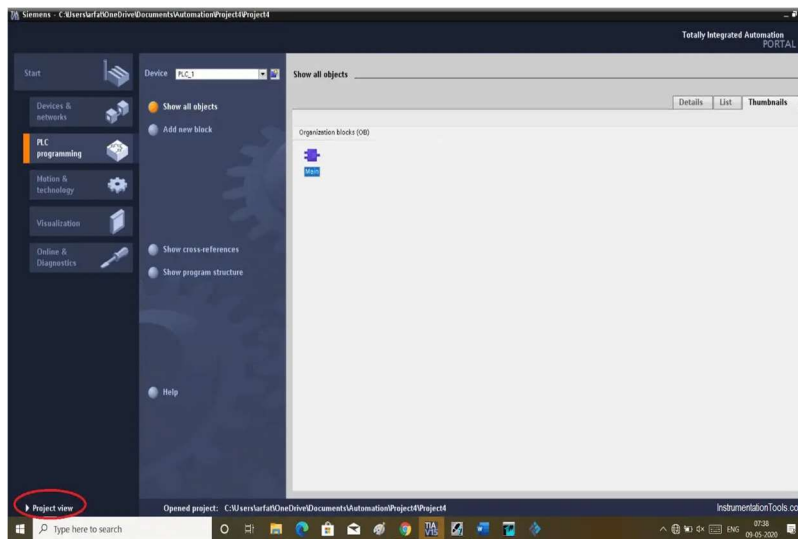
### Aplicações de interrupções cíclicas

1. **Controladores PID:** O tempo do ciclo de varredura principal pode afetar a precisão de suas ações PID, então você não deseja relacionar suas ações com o tempo do ciclo de varredura, então você empurra suas instruções PID para uma interrupção cíclica para ser executado periodicamente com determinados tempos cíclicos.
2. **Segurança:** Segurança está em primeiro lugar , portanto durante o seu processo você pode precisar de algumas ações de segurança que devem ser verificadas periodicamente em determinados horários.

O tempo de interrupção padrão no PLC S7 é de 100 milissegundos. Mas você pode alterá-lo na visualização de Hardware.

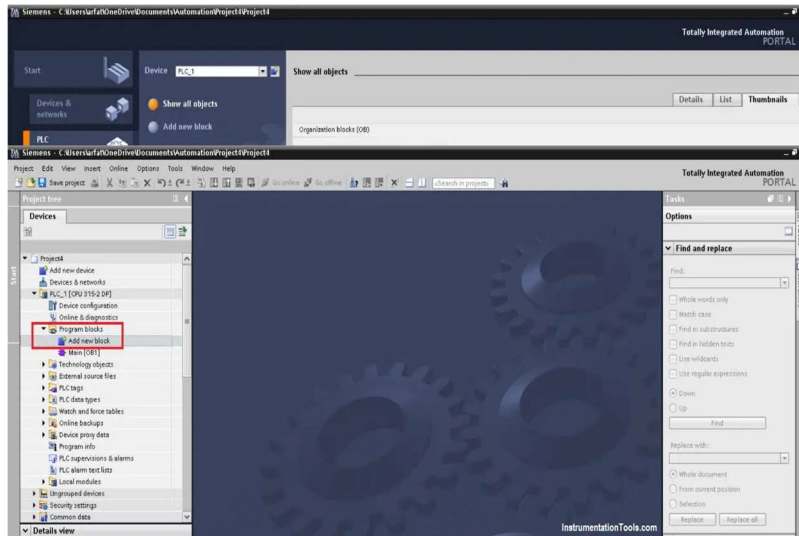
Para adicionar interrupção cíclica ao OB, abra o portal TIA e siga os passos abaixo.

### Passo 1:



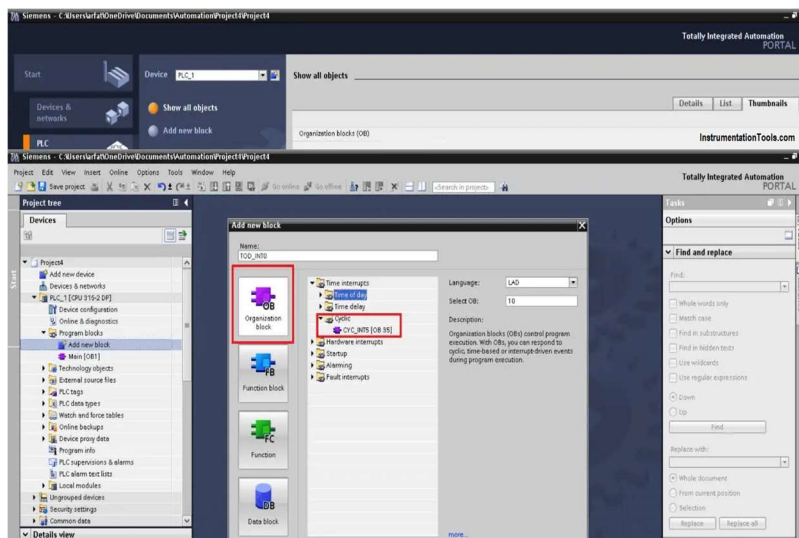
*crie um novo projeto e selecione o tipo de CPU. Clique em “visualização do projeto”.*

### Passo 2:



Na janela seguinte expanda “bloco de programa” e clique duas vezes em “Adicionar novo bloco”.

### Passo 3:



Uma nova janela aparecerá. Selecione “bloco organizacional” e você encontrará todos os tipos de interrupção de hardware disponíveis. Clique duas vezes em qualquer um para acessar o ambiente de programação.

#### A.1.3 Interrupção de Hardware (OB40)

Este tipo de interrupção OB é executada quando a CPU detecta algum erro no módulo de entrada, ou devido a circuitos residentes no módulo de entrada. É preciso que o módulo de entrada contenha o recurso de interrupção de hardware e o mesmo, OB40, seja configurado.

#### A.1.4 Interrupção por Tempo (OB20)

Este tipo de interrupção será executado com base em um determinado horário do dia para realizar uma tarefa.

É necessário configurar o OB20 para usar a interrupção por tempo. Neste bloco você pode especificar um tempo e escrever a lógica de execução da interrupção por atraso de tempo.

#### A.1.5 Interrupção de Inicialização (OB100)

Exemplos de chamada da interrupção de inicialização são, falha de energia, se o PLC entrar no modo de parada quando a energia estiver disponível, a transição da CPU de parada para operação será executada. O OB100 de inicialização será executado, o que redefinirá a entrada para não iniciar o equipamento automaticamente.

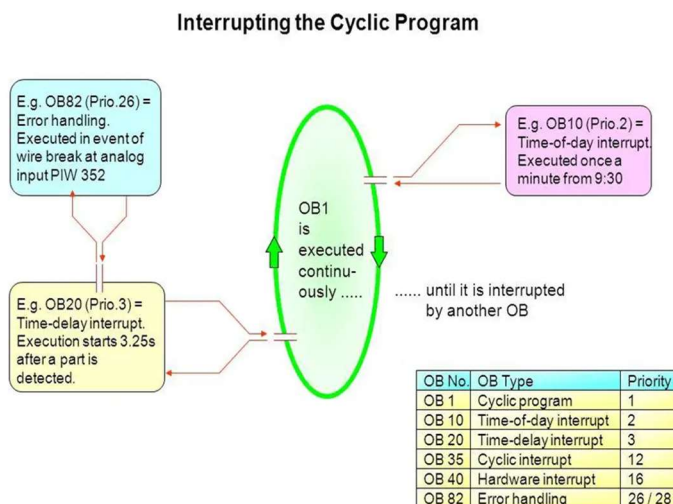
Este tipo de interrupção é utilizado para realizar uma inicialização da CPU de forma mais suave.

#### A.1.6 Prioridade das Interrupções

Uma observação muito importante que você deve estar ciente disso é a prioridade da interrupção que você está usando.

Cada interrupção no portal TIA possui um número denominado prioridade que se refere à disposição daquela interrupção entre todas as interrupções, supondo que duas interrupções ocorram ao mesmo tempo.

O PLC executará primeiro a maior prioridade e depois a Segunda interrupção (que tem menor prioridade) e depois voltará novamente para o OB Principal.

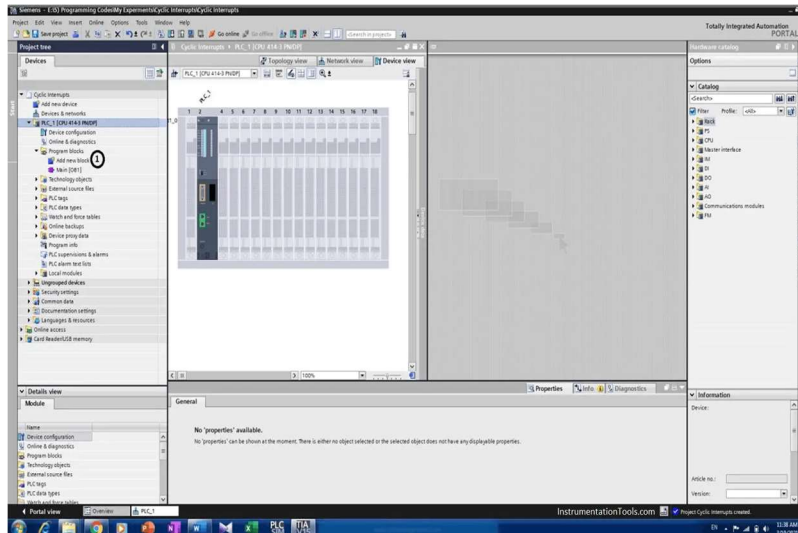


*Exemplo de interrupção do programa cíclico, OB1.*

## A.1.7 Interrupções cíclicas no Portal TIA

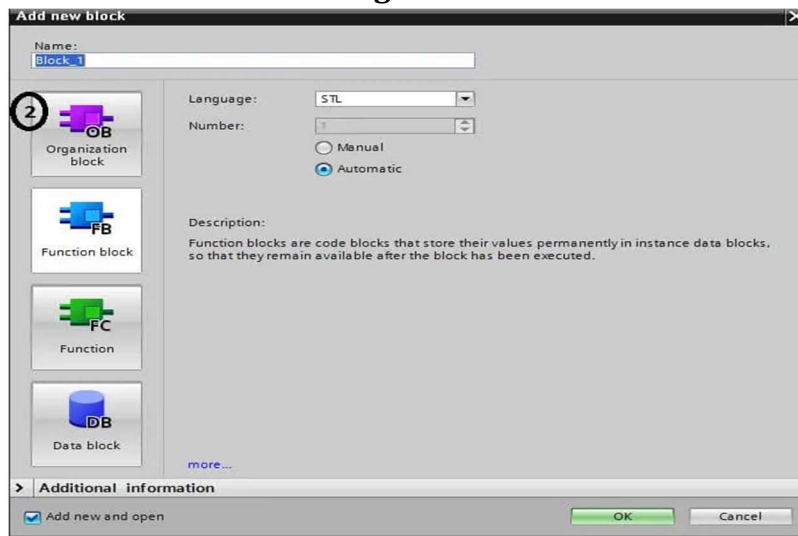
Uma observação muito importante que você deve estar ciente disso é a prioridade da interrupção que você está usando.

### 1. Selecione a CPU desejada:



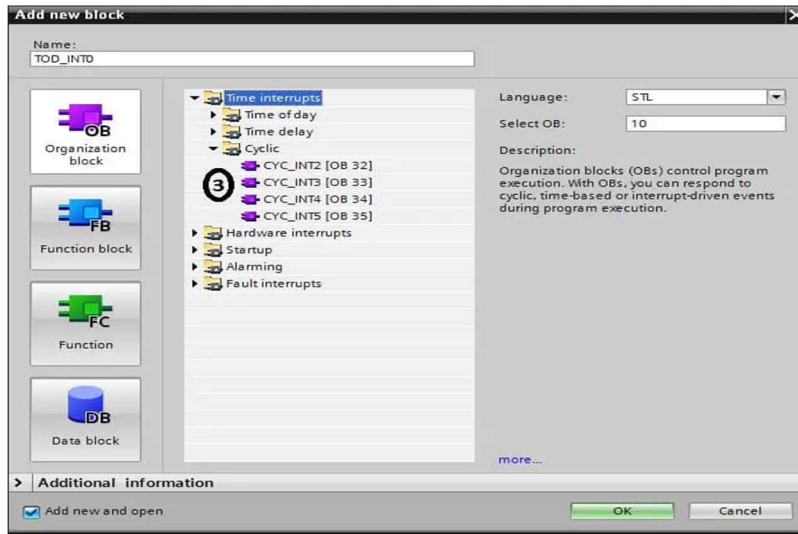
*Configurar um novo dispositivo.*

### 2. Selecione adicionar novo bloco organizacional:



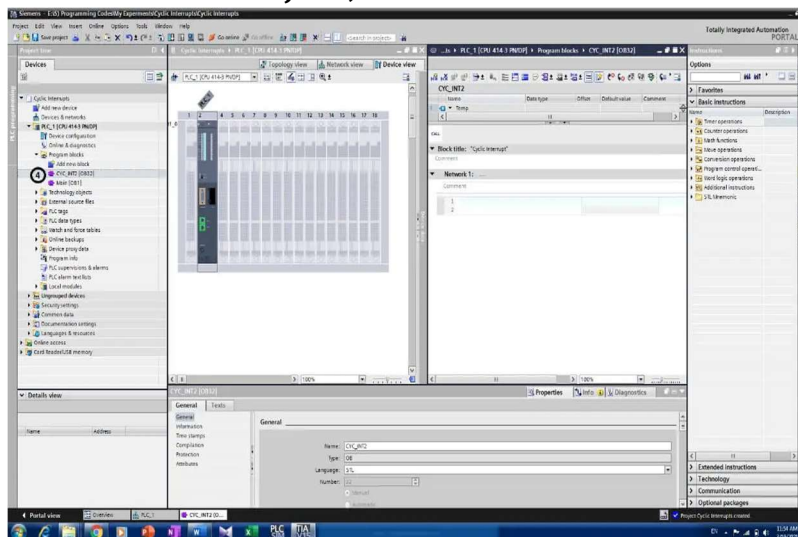
*Selecionar adicionar novo bloco organizacional.*

### 3. Selecione qualquer interrupção e pressione OK:



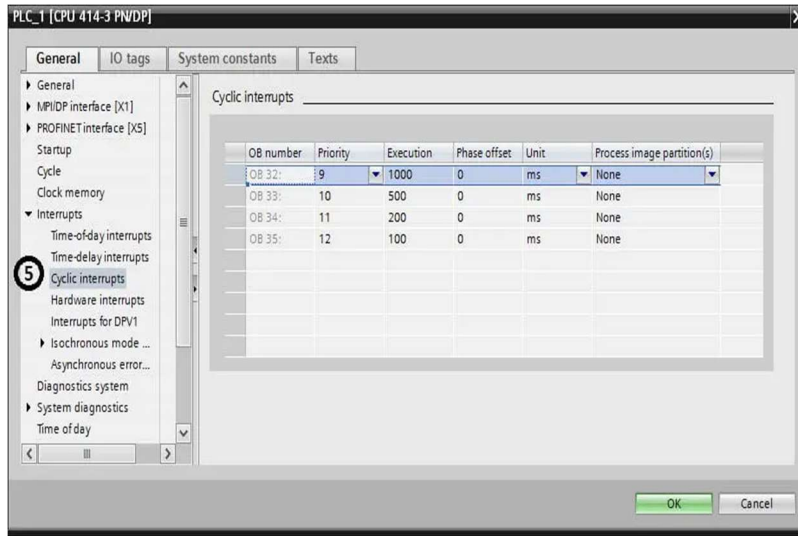
*Selecionar a interrupção desejada.*

4. Depois de selecionar o desejado, ele será mostrado nos blocos do programa:



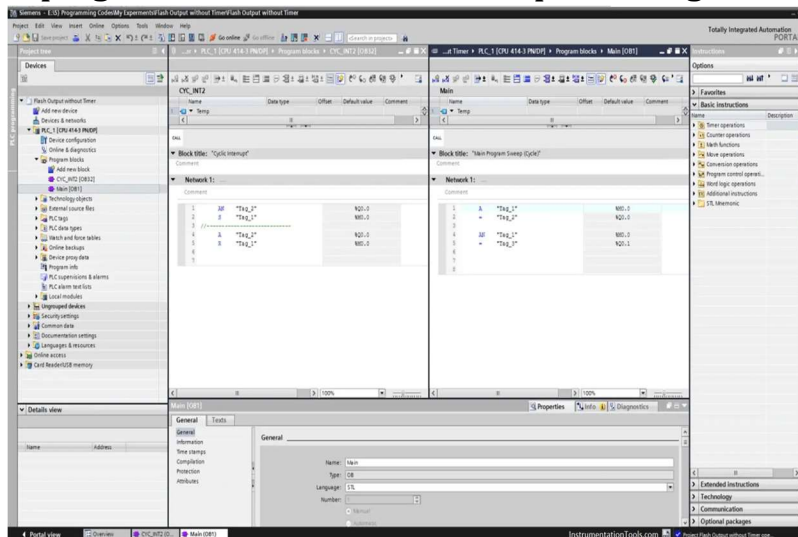
*Conferir o programa.*

5. Selecionar as propriedades da CPU para configurar o tempo de execução:



Configurar o tempo de execução.

## 6. Escrever o programa conforme mostrado na próxima figura.:



Escrever o programa.

### A.2 OBs de Erros

#### A.2.1 OB121 - Erro de programação

#### A.2.2 OB122 - Erro de acesso periférico

[https://www.youtube.com/embed/KclkwMxb2kg?si=M8Iw\\_S6YUakvydhw](https://www.youtube.com/embed/KclkwMxb2kg?si=M8Iw_S6YUakvydhw)

### A.3 References

[@intCiclicaOB30]



## Apêndice B: Ferramentas

Esta é uma lista de ferramentas e hardwares utilizadas em automação industrial.

### B.1 Hardware Kits

#### B.1.1 Utilitários de Hardware

No	Hardware	Processor	Características	Compatibilidade
1				

### B.2 Ferramentas de Diagnóstico

#### B.2.1 Kits de Diagnóstico e Recuperação

No	Ferramentas	Descrição	Casos de Uso
1	<a href="#">CloneZilla</a>	Programa de clonagem/imagem de partição e disco semelhante ao <a href="#">True Image®</a> ou <a href="#">Norton Ghost®</a> .	Ajuda fazer implantação do sistema, backup e recuperação bare metal.
2	<a href="#">Hirens CD</a>	Ferramentas utilizadas para resolver vários problemas relacionados ao computador. Notavelmente, não inclui nenhum software pirateado; em vez disso, contém exclusivamente software gratuito e legal.	Diagnóstico & Recuperação de PC
3	<a href="#">Medicat USB</a>	Seleção de ferramentas de diagnóstico e recuperação de computador fácil de usar.	Diagnóstico & Recuperação de PC
4	<a href="#">Venloy</a>	Ferramenta de código aberto para criar uma unidade USB inicializável para arquivos ISO/WIM/IMG/VHD(x)/EFI.	Solução USB inicializável
5	<a href="#">Splunk</a>	Software para pesquisar, monitorar e analisar dados gerados por máquina por meio de uma interface estilo web.	Diagnóstico de rede

### B.3 IDEs e ambientes de desenvolvimento

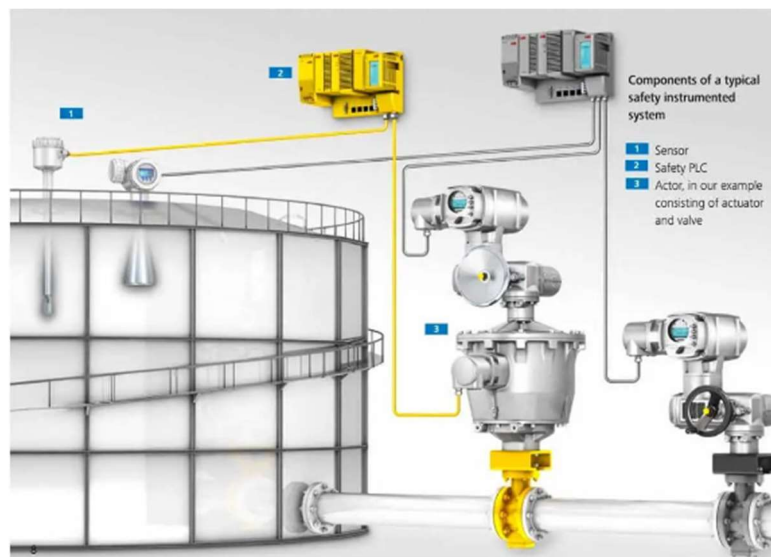
No	IDE/Ambientes de Desenvolvimento	Descrição	Características
1	<a href="#">TIA Portal</a>	Portal de Automação Totalmente Integrado. Ele oferece acesso irrestrito à gama completa de serviços de automação digitalizada, desde planejamento digital até engenharia integrada e operação transparente.	Ambiente de desenvolvimento de Automação para plataforma de PLC Siemens

## Apêndice C: Instrumentação

### C.1 Sistemas instrumentados de segurança (SIS)

Existem lógicas de votação 1oo1, 1oo2, 2oo2, 2oo3 etc<sup>4</sup>. na arquitetura do sistema instrumentado de segurança.

A arquitetura lógica de votação normalmente utilizada no instrumento de campo e/ou elementos de controle finais para atingir determinado **nível de integridade de segurança (SIL)** ou para atingir determinada redução de custos devido ao desligamento da plataforma. Em geral, quando devemos usar a arquitetura lógica de votação 1oo1, 1oo2, 2oo2 ou 2oo3?



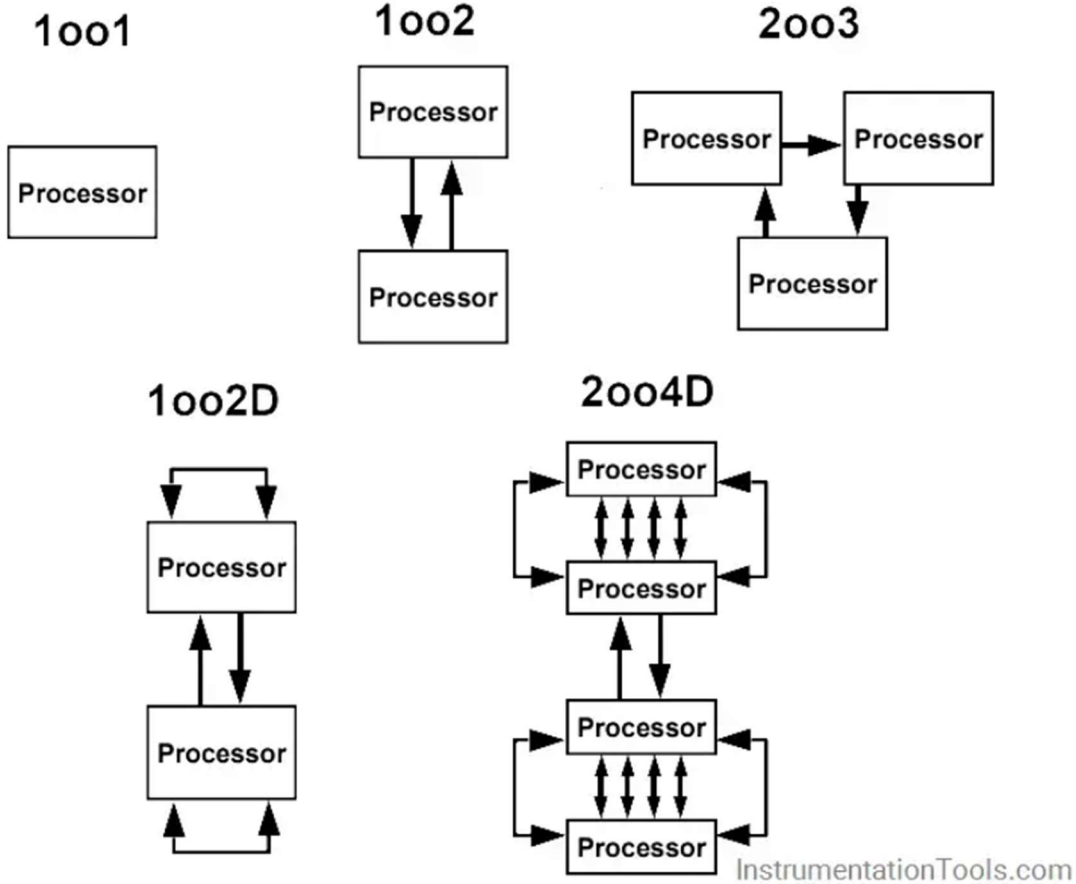
*Exemplo de SIS.*

***Em Construção***

---

<sup>4</sup> Em construção, <https://instrumentationtools.com/voting-logic-safety-instrumented-system/>

### C.1.1 Lógica de votação



*lógica de votação.*

Conforme mencionado acima, há dois propósitos pelos quais determinadas arquiteturas [lógicas de votação](#) foram escolhidas: o primeiro é atingir determinado SIL e, em segundo lugar, alcançar certa redução de custos devido ao desligamento espúrio da plataforma.

***Em Construção***

## Apêndice D: Network

Conjunto de dispositivos como computadores, servidores, telefones celulares, etc., que estão conectados e podem compartilhar informações e recursos entre si. As redes podem ser pequenas e locais (como uma rede doméstica) ou abranger uma grande área geográfica (como a Internet).

## D.1 Endereços IP Privados ( Redes Locais )

Os endereços reservados para redes locais, também conhecidos como endereços IP privados, são:

- 10.0.0.0 a 10.255.255.255
- 172.16.0.0 a 172.31.255.255
- 192.168.0.0 a 192.168.255.255

4.3 Billion IP Addresses  
 $2^{32} = 4,294,967,296$

RFC 1918  
Private IP Addresses

	Range	Subnet Mask
A	1.0.0.0 - 126.255.255.255	255.0.0.0
B	128.0.0.0 - 191.255.0.0	255.255.0.0
C	192.0.0.0 - 223.255.255.0	255.255.255.0
D	224.0.0.0 - 239.255.255.255	
E	240.0.0.0 - 255.255.255.255	

Private

A	10.0.0.0 - 10.255.255.255	255.0.0.0
B	172.16.0.0 - 172.31.255.255	255.255.0.0
C	192.168.0.0 - 192.168.255.255	255.255.255.0

not unique

192.168.1.0  
255.255.255.0

[@rfc1918Wiki; @rfc1918]

As classes de endereços IP são diferentes em termos de quantidade de redes e hosts que podem acomodar:

- **Classe A:** Este é o maior tipo de rede, pois pode acomodar mais de 16 milhões de hosts. Os endereços de Classe A variam de 1.0.0.0 a 126.0.0.0.
- **Classe B:** Estas redes podem acomodar até aproximadamente 65.000 hosts. Os endereços de Classe B variam de 128.0.0.0 a 191.255.0.0.
- **Classe C:** Estas redes são menores, acomodando até 254 hosts. Os endereços de Classe C variam de 192.0.0.0 a 223.255.255.0.
- **Classe D:** Esta classe é reservada para multicast. Os endereços variam de 224.0.0.0 a 239.255.255.255.
- **Classe E:** Esta classe é reservada para uso futuro ou experiências de pesquisa. Os endereços variam de 240.0.0.0 a 255.255.255.255.

### D.1.1 Loopback - 127.0.0.0

O range 127.0.0.0 é reservado para comunicações de **loopback**. Isso significa que é usado para **testar a rede de um computador**. Qualquer tráfego enviado para um endereço dentro deste range é redirecionado de volta para o próprio computador.

Exemplo, uma forma de testar se a porta Ethernet esta ok é fazer um loopback nela, conforme abaixo:<sup>5</sup>

```
ping 127.0.0.1
```

## D.2 Entidades de Rede

### D.2.1 Roteadores

Roteadores, também conhecidos como routers, são dispositivos que **encaminham pacotes de dados entre redes de computadores**. Eles têm a capacidade de “rotear” informações de um lugar para outro, fazendo conexões eficientes entre diferentes redes. Isso é crucial para a internet, onde os pacotes de dados podem ter que passar por várias redes antes de chegar ao seu destino final.

### D.2.2 Switches

Switches, também conhecidos como comutadores, são dispositivos de rede que conectam vários dispositivos em uma rede de computadores. Eles usam endereços MAC para encaminhar dados para o dispositivo de destino, permitindo a comunicação de dados de alta velocidade entre os dispositivos conectados. Ao contrário dos roteadores, que encaminham os dados para redes diferentes, os switches operam principalmente dentro de uma única rede.

### D.2.3 Default Gateway

O Default Gateway, ou Gateway Padrão, é um dispositivo de rede que serve como um ponto de acesso ou IP que os dispositivos usam para enviar informações para redes fora da sua rede local. Em geral, é o endereço do seu roteador na rede.

## D.3 Comandos úteis

### D.3.1 ipconfig ( Windows )

O **ipconfig** é um comando disponível em muitos sistemas operacionais baseados em Unix e em sistemas Microsoft Windows. Ele é usado para exibir as configurações atuais de rede do sistema, como endereços IP, máscaras de sub-rede, gateways padrão e muito mais.

#### ifconfig ( Linux )

O comando **ifconfig** é uma ferramenta de linha de comando que é usada para configurar, gerenciar e consultar as configurações de rede em sistemas operacionais **Unix/Linux**. Ele permite configurar (ou exibir) as interfaces de rede do sistema, como endereços IP, máscaras de sub-rede, entre outros.

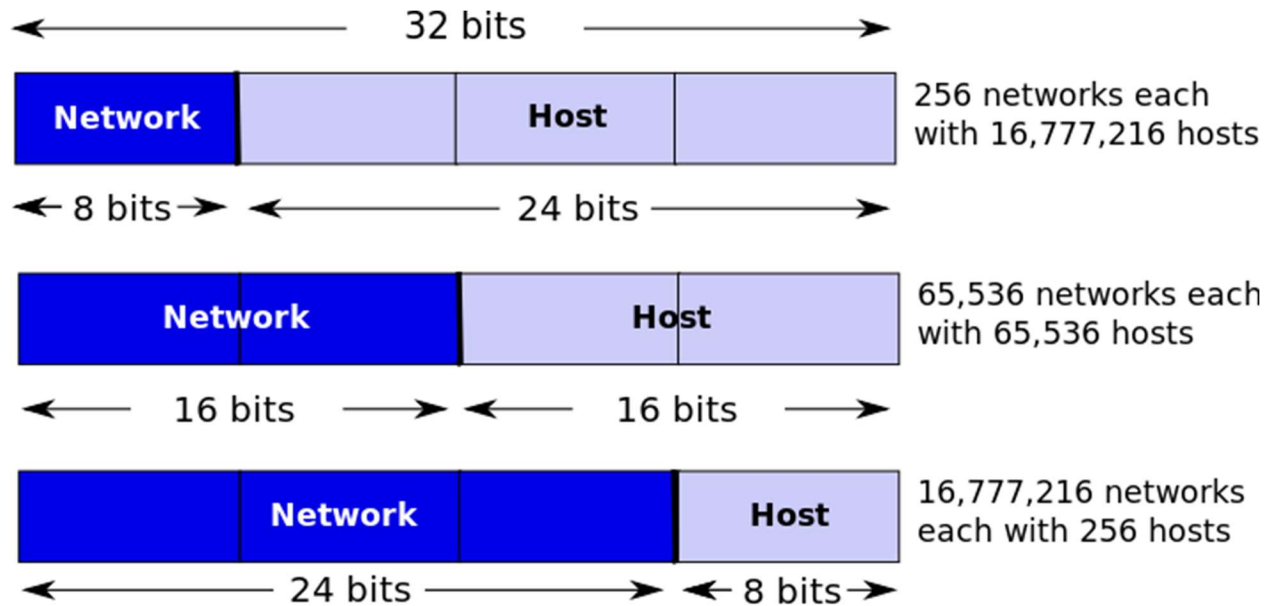
---

<sup>5</sup> Usar o CMD do Windows

## D.4 Subnetting IPv4

### D.4.1 Subnet Mask

É a máscara de sub-rede, é um número que define a separação entre a parte da rede e a parte do host em um endereço IP. Ela é usada para identificar a sub-rede a qual um endereço IP pertence.[@networkingPortion]



### Subnetting

#### Resumo CIDR do IPv4

MASK	>	128		64		32		16		8		4		2		1
HOST.NET	>	256		128		64		32		16		8		4		2

INCREMENTO: É o ultimo bit roubado.

```
/25 > .128  
/26 > .192  
/27 > .224  
/28 > .240  
/29 > .248  
/30 > .252  
/31 > .254  
/32 > .255
```

### D.4.2 Endereços Reservados

Os dois endereços IP reservados em cada sub-rede são o **endereço de rede** (primeiro) e o **endereço de broadcast** (ultimo).

## 💡 Endereços disponíveis

Logo a quantidade de endereços disponíveis são o total de endereços, menos 2 endereços, de rede e broadcast. Lembrando que um endereço, normalmente o primeiro é o endereço usado para o default gateway da rede.

Assim a quantidade disponível será o total da rede, menos 3 endereços.

## D.5 IPv6

<https://www.youtube.com/watch?v=QdrfX-2cDqs>

*Em Construção*

## D.6 Troubleshooting

### D.6.1 TCP/UDP open ports

Caso não tenha a sua disposição ferramentas de diagnóstico de rede como **Splunk**, é possível utilizar comandos do **PowerShell**, como o cmdlet **Test-NetConnection** para verificar se uma porta está disponível (aberta) em um computador remoto.

Você pode usar este cmdlet para verificar a resposta e a disponibilidade de um servidor remoto ou serviço de rede, **testar se a porta TCP está bloqueada por um firewall**, verificar a disponibilidade do ICMP e o roteamento.

Este comando, substitui várias ferramentas populares de administração de Test-NetConnectionrede, como ping, scanner de porta TCP, tracert, telnet e pathping

### ⚠️ Notas do comando Test-NetConnection

- **Restrição do comando:** comando usado para testar apenas a conectividade de uma porta TCP. Logo, não é possível usar o cmdlet para verificar a disponibilidade das portas UDP remotas.
- **Versão abreviada do comando Test-NetConnection:** TNC é a versão abreviada do mesmo.

#### D.6.1.1 Exemplo #1: Comando para verificar se porta remota esta aberta

1. Como exemplo, segue abaixo, comando<sup>6</sup> para verificar se a porta 25<sup>7</sup> está aberta no servidor remoto **ny-msg01**, e sua versão abreviada:

```
Test-NetConnection -ComputerName ny-msg01 -Port 25
```

---

<sup>6</sup> Executar no powershell

<sup>7</sup> TCP 25: Protocolo SMTP

#Versão abreviada

TNC ny-msg01 -Port 25

## *Em Construção*

### **D.7 Fontes de informação**

#### **D.7.1 Playlist - Your suck at subnetting**

<https://www.youtube.com/watch?v=5WfiTHiU4x8&list=PLIhvc56v63IKrRHh3gvZZBAGvsvOhwrRF>

### **D.8 F.Q.A - Perguntas frequentes**

#### **D.8.1 Como você encontra o endereço IP dos seus dispositivos?**

#### **D.8.2 Como saber seu endereço IP**

#### **D.8.3 Desmistificando sub-redes**

#### **D.8.4 Gateway padrão**

#### **D.8.5 Desafio**

#### **D.8.6 Solução do desafio**

#### **D.8.7 Recapitulando**

#### **D.8.8 O que aconteceu com todos os endereços IP?**

#### **D.8.9 Quais são os intervalos das classes?**

#### **D.8.10 Quem divulgou todos esses endereços?**

#### **D.8.11 O que são redes sem classes**

#### **D.8.12 Isso me deixa louco (Classe D e E)**

#### **D.8.13 O que é loopback**

#### **D.8.14 O que é o Ping?**

### **D.9 References**

[@networkingPortion]

[@rfc1918]

[@rfc1918Wiki]