



[www.altus.com.br](http://www.altus.com.br)

# APOSTILA DE TREINAMENTO

## SÉRIE NEXTO

*BÁSICO*



---

Nenhuma parte deste documento pode ser copiada ou reproduzida sem o consentimento prévio e por escrito da Altus Sistemas de Automação S.A., que se reserva o direito de efetuar alterações sem prévio comunicado. Conforme o Código de Defesa do Consumidor vigente no Brasil, informamos, a seguir, aos clientes que utilizam nossos produtos, aspectos relacionados com a segurança de pessoas e instalações. Os equipamentos de automação industrial fabricados pela Altus são robustos e confiáveis devido ao rígido controle de qualidade a que são submetidos. No entanto, equipamentos eletrônicos de controle industrial (controladores programáveis, comandos numéricos etc.) podem causar danos às máquinas ou processos por eles controlados em caso de defeito em seus componentes e/ou de erros de programação ou instalação, podendo inclusive colocar em risco vidas humanas. O usuário deve analisar as possíveis consequências destes defeitos e providenciar instalações adicionais externas de segurança que, em caso de necessidade, sirvam para preservar a segurança do sistema, principalmente nos casos da instalação inicial e de testes. Os equipamentos fabricados pela Altus não trazem riscos ambientais diretos, não emitindo nenhum tipo de poluente durante sua utilização. No entanto, no que se refere ao descarte dos equipamentos, é importante salientar que quaisquer componentes eletrônicos incorporados em produtos contêm materiais nocivos à natureza quando descartados de forma inadequada. Recomenda-se, portanto, que quando da inutilização deste tipo de produto, o mesmo seja encaminhado para usinas de reciclagem que deem o devido tratamento para os resíduos. É imprescindível a leitura completa dos manuais e/ou características técnicas do produto antes da instalação ou utilização do mesmo. Os exemplos e figuras deste documento são apresentados apenas para fins ilustrativos. Devido às possíveis atualizações e melhorias que os produtos possam incorrer, a Altus não assume a responsabilidade pelo uso destes exemplos e figuras em aplicações reais. Os mesmos devem ser utilizados apenas para auxiliar na familiarização e treinamento do usuário com os produtos e suas características. A Altus garante os seus equipamentos conforme descritos nas Condições Gerais de Fornecimento, anexada às propostas comerciais. A Altus garante que seus equipamentos funcionam de acordo com as descrições contidas explicitamente em seus manuais e/ou características técnicas, não garantindo a satisfação de algum tipo particular de aplicação dos equipamentos. A Altus desconsiderará qualquer outra garantia, direta ou implícita, principalmente quando se tratar de fornecimento de terceiros. Os pedidos de informações adicionais sobre o fornecimento e/ou características dos equipamentos e serviços Altus devem ser feitos por escrito. A Altus não se responsabiliza por informações fornecidas sobre seus equipamentos sem registro formal. Alguns produtos utilizam tecnologia EtherCAT ([www.ethercat.org](http://www.ethercat.org)).

## **DIREITOS AUTORAIS**

Nexto e MasterTool são marcas registradas da Altus Sistemas de Automação S.A.

Windows é marca registrada da Microsoft Corporation.

## **NOTIFICAÇÃO DE USO DE SOFTWARE ABERTO**

Para obter o código fonte de componentes de software contidos neste produto que estejam sob licença GPL, LGPL, MPL, entre outras, favor entrar em contato através do e-mail [opensource@altus.com.br](mailto:opensource@altus.com.br). Adicionalmente ao código fonte, todos os termos da licença, condições de garantia e informações sobre direitos autorais podem ser disponibilizadas sob requisição.

# Sumário

Introdução	7
O Curso MP10	8
Estruturação do Tutorial	9
Ementa do curso	9
Referências	10
Documentos relacionados	10
Avaliação do curso	12
Terminologia	12
Convenções Utilizadas	12
Suporte Técnico	13
Base de Conhecimento ALTUS	13
Cursos ALTUS	14
Revisões deste Tutorial	14
Training Box Nexto	14
1. Hardware do CP Nexto	1-1
Características técnicas da Série Nexto	1-1
O que é o Nexto?	1-1
Porque o Nexto?	1-2
Quais São Seus Diferenciais?	1-2
Quais as suas características principais?	1-5
O que o Nexto traz de inovação?	1-6
Arquitetura da Série Nexto	1-9
Estudo Dirigido 1-1: módulos Nexto	1-10
Estudo Dirigido 1-2: arquiteturas Nexto	1-11
Um pouco de teoria... Histórico e características dos CPs	1-19
Diagnósticos e manutenção da Série Nexto	1-20
<i>One Touch Diag</i> (OTD)	1-21
Acessando pontos de E/S	1-23
Acessando a descrição do módulo e dos pontos de E/S	1-23
Visor Gráfico	1-24
Pesquisa na Documentação: visor gráfico	1-25
Pesquisa na Documentação: tempos de <i>One Touch</i>	1-26
Diagnósticos via LED	1-26
Pesquisa na Documentação: LEDs de diagnóstico e cão-de-guarda	1-26
Pesquisa na Documentação: LEDs dos conectores RJ45	1-27
Diagnósticos via WEB	1-27
Estudo Dirigido 1-3: diagnóstico da arquitetura Nexto	1-31
Diagnósticos via Variáveis	1-32
Diagnósticos via Função	1-32
<i>Diagnostic Explorer</i>	1-32
Log de Sistema	1-33
Não Carregar a Aplicação na Inicialização	1-33
Falha na Alimentação	1-33
Manutenção da Série Nexto	1-33
Solução de Problemas	1-34
Instalação da Série Nexto	1-36

---

Projeto Mecânico	1-36
Projeto Térmico	1-36
Projeto Elétrico	1-36
Instalação Mecânica	1-37
Instalação Elétrica	1-37
Instalação do Cartão de Memória	1-39
Instalação da Arquitetura	1-40
Remoção de Módulos no Bastidor Principal	1-41
Conexão com a Rede Ethernet	1-41
Conexão com Redes Seriais	1-42
Pesquisa na Documentação: interfaces	1-44
Estudo Dirigido 1-4: conexões RS-485 na Série Nexto	1-45
Estudo Dirigido 1-5: conexões RS-422 na Série Nexto	1-46
Um pouco de teoria... O meio físico RS485 e o Profibus	1-47
2. Software do CP Nexto	2-1
Características do MasterTool IEC XE	2-1
Linguagens de Programação IEC 61131-3	2-3
Versões do MasterTool IEC XE	2-3
Informações Gerais sobre o Programador	2-3
Funcionalidades do MasterTool IEC XE	2-3
Instalação	2-3
Projeto	2-4
Perfis de Projeto	2-4
Objetos de Recurso	2-4
Objetos de Programação	2-4
Janela das POUs	2-5
Dispositivos, Árvore de Dispositivos	2-5
Aplicação	2-6
Task Configuration	2-7
Geração de Código e Alteração Online	2-7
Monitoração	2-7
Depuração	2-7
Segurança	2-8
Interface do Usuário do MasterTool IEC XE	2-9
Janelas, Visualizações e Janelas do Editor	2-10
Customizando a Interface do Usuário	2-11
Organizando Barras de Menu e Barras de Ferramentas	2-11
Zoom	2-11
Interface do Usuário no Modo Online	2-11
Menus e Comandos Padronizados	2-12
Memória de Arquivos de Usuário	2-13
Gerenciamento de Usuários e Direitos de Acesso	2-13
Editores	2-13
Primeiros Passos com o MasterTool IEC XE	2-15
Iniciando o MasterTool IEC XE	2-15
Acrescentando Módulos	2-21
Criando POUs	2-22
Criando Tarefas	2-24
Configurando Tarefas	2-26
Vinculando uma POU a uma Tarefa	2-28
Configurando a UCP	2-29
Bibliotecas	2-29
Inserindo uma Instância de Protocolo	2-30
Compilando um Projeto	2-36
Simulando um Projeto	2-37
Criação e Execução de Projetos	2-40
Declarar Variáveis na UserPrg	2-40

---

Digitar o Código no Corpo da UserPrg	2-41
Criar a POU de Programação Adicional (ST Bloco Funcional FB1)	2-42
Definir os Recursos para Executar e Controlar o Programa no Nexto	2-43
Executar e Monitorar a Aplicação no Nexto	2-43
Depurar uma Aplicação via Breakpoint e Varredura do Programa	2-46
Diagnósticos	2-47
Página de Diagnósticos	2-47
Ajuda	2-51
Ajuda Sensível ao Contexto	2-51
Estudo Dirigido 2-1: criação de projeto no MasterTool IEC XE	2-51
Considerações gerais sobre a configuração do sistema	2-55
Estudo Dirigido 2-2: configuração do barramento lógico do Nexto	2-56
Pesquisa na Documentação: configuração padrão do sistema operacional	2-58
3. Programação do CP Nexto	3-1
Elementos Básicos da Aplicação	3-1
Unidade de Organização de Programa	3-1
Lista de Variáveis Globais - GVL	3-2
Gerenciamento de Bibliotecas	3-2
Operadores	3-2
Operandos	3-3
Declaração de Variáveis	3-3
Tipos de Variáveis	3-4
Linguagem de Programação Diagrama Ladder (LD)	3-4
Comandos Básicos do LD	3-7
Criação de POUs	3-11
POU em LD	3-11
Estudo Dirigido 3-1: aplicação para controle de um portão automático	3-14
Utilização de blocos funcionais definidos pelo usuário	3-18
Estudo Dirigido 3-2: criação de um bloco funcional	3-23
Estudo Dirigido 3-3: funções de diagnóstico no Nexto	3-27
Um pouco de teoria... Programação Padronizada Multi-linguagens	3-30
4. Aplicações, Avaliação e Encerramento	4-1
Aplicação 1: Controle Discreto	4-1
Proposição da Atividade	4-1
Resolução da Atividade	4-2
Aplicação 2: Controle Contínuo	4-2
Proposição da Atividade	4-2
Resolução da Atividade	4-4
Avaliação do Treinamento	4-5
Parabéns...	4-5
Glossário	A

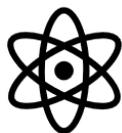
# Figuras

Figura 1-1. A UCP Nexto _____	1-1
Figura 1-2. A solução Nexto para disponibilidade total _____	1-4
Figura 1-3. Painel Frontal UCP Nexto _____	1-21
Figura 1-4. Visualização dos diagnósticos da UCP _____	1-22
Figura 1-5. Visor do Módulo _____	1-23
Figura 1-6. Tela de Status da UCP _____	1-25
Figura 1-7. Tela Inicial _____	1-28
Figura 1-8. Informações do Sistema _____	1-29
Figura 1-9. Diagnósticos do Sistema _____	1-30
Figura 1-10. Estado do Sistema _____	1-30
Figura 1-11. Inserção do Cartão de Memória na UCP _____	1-39
Figura 1-12. Retirada do Cartão de Memória _____	1-40
Figura 1-13. Instalação do Módulo no Bastidor _____	1-40
Figura 1-14. Desconexão do Barramento _____	1-41
Figura 2-1. Ambiente de programação MasterTool IEC XE _____	2-1
Figura 2-2. Janela dos POU's _____	2-5
Figura 2-3. Árvore de Dispositivos _____	2-6
Figura 2-4. Exemplo do MasterTool IEC XE Interface do Usuário _____	2-9
Figura 2-5. Linha Pontilhada no Menu e Barras de Ferramentas _____	2-11
Figura 2-6. Diálogo de Estado Online _____	2-12
Figura 2-7. Barra de Menu Padrão _____	2-13
Figura 2-8. Utilização da Memória de Arquivos de Usuário _____	2-13
Figura 2-9. Editores _____	2-14
Figura 2-10. Novo Projeto _____	2-15
Figura 2-11. Classificação do Projeto _____	2-16
Figura 2-12. Seleção dos Módulos de Hardware _____	2-17
Figura 2-13. Opções de Rede _____	2-18
Figura 2-14. Configuração de Pontos de E/S _____	2-19
Figura 2-15. Seleção do Perfil _____	2-20
Figura 2-16. Linguagem de Programação _____	2-20
Figura 2-17. Visualizando a Biblioteca _____	2-21
Figura 2-18. Acrescentando Módulos _____	2-21
Figura 2-19. Inserindo POU's _____	2-22
Figura 2-20. Classificando a POU _____	2-23
Figura 2-21. Editando a POU _____	2-24
Figura 2-22. Criando uma Tarefa _____	2-25
Figura 2-23. Nomeando a Tarefa _____	2-26
Figura 2-24. Configurando a Tarefa Criada _____	2-27
Figura 2-25. Vinculando POU's às Tarefas _____	2-28
Figura 2-26. Inclusão de uma Biblioteca no Projeto _____	2-29
Figura 2-27. Adicionando a Instância _____	2-30
Figura 2-28. Selecionando o Protocolo _____	2-31
Figura 2-29. Seleção do protocolo MODBUS Symbol RTU Slave _____	2-32
Figura 2-30. Adicionando a Instância _____	2-33
Figura 2-31. Selecionando o Protocolo _____	2-34
Figura 2-32. Selecionando o Protocolo MODBUS Ethernet _____	2-35
Figura 2-33. Compilando o Projeto _____	2-36
Figura 2-34. Mensagens da Compilação _____	2-37
Figura 2-35. Incluindo as Mensagens na Tela _____	2-37
Figura 2-36. Modo Simulação _____	2-38
Figura 2-57. Janela do Editor ST _____	2-41

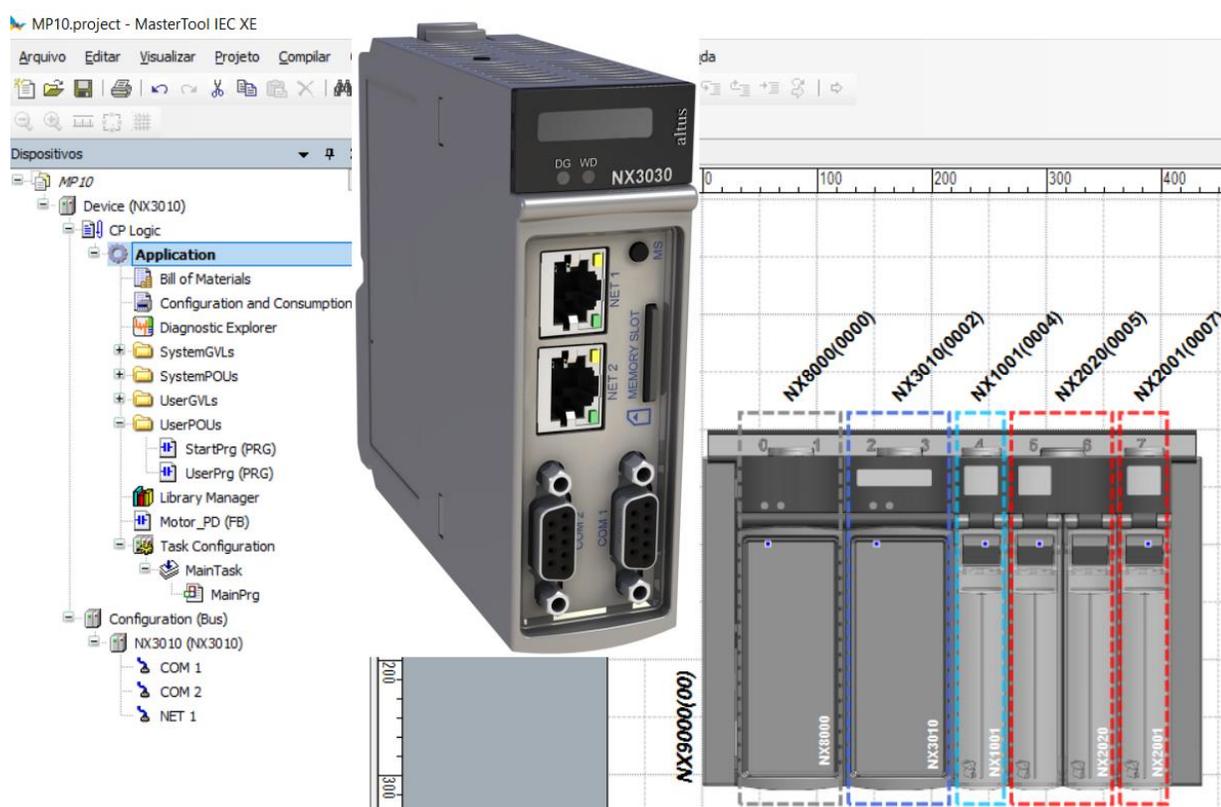
---

Figura 2-58. Diálogo Autodeclarar	2-42
Figura 2-59. Declaração da Variável “erg”	2-42
Figura 2-40. Expressões de Monitoração e Linhas de Código	2-45
Figura 2-41. Lista de Monitoração	2-45
Figura 2-42. Exibição do Valor Atual	2-46
Figura 2-43. Aplicação em “Stop”	2-46
Figura 2-44. Aplicação no Modo “Run”	2-46
Figura 2-45. Diagnostics Explorer	2-47
Figura 2-46. Variáveis de Diagnóstico	2-48
Figura 2-47. Variáveis de Diagnóstico Utilizando Drivers de Comunicação por Mapeamento Simbólico	2-49
Figura 2-48. Objeto Disables	2-50
Figura 2-49. Objeto ReqDiagnostics	2-50
Figura 2-50. Configuração do barramento	2-55
Figura 3-1. Instância “Inst1”	3-2
Figura 3-2. Rede LD	3-4
Figura 3-3. Comandos Editores FBD/LD/ID	3-6
Figura 3-4. Caixa de Ferramentas	3-6
Figura 3-5. Título, Comentário e Rótulo	3-7
Figura 3-6. Contato	3-7
Figura 3-7. Contato Negado	3-7
Figura 3-8. Bobina	3-8
Figura 3-9. Bobina Negada	3-8
Figura 3-10. Bobinas Set e Reset	3-8
Figura 3-11. Visualização Online do Editor LD	3-9
Figura 3-12. Diálogo Preparar Valor	3-9
Figura 3-13. Aplicação “Carro Automático para Máquina Operatriz”	3-11
Figura 3-14. Declaração das Variáveis de Alocação Direta (Entradas do CP)	3-12
Figura 3-15. Declaração das Variáveis de Alocação Direta (Saídas do CP)	3-12
Figura 3-16. Declaração das Variáveis Locais no Programa do Usuário (UserProg)	3-12
Figura 3-17. Edição das Redes (Implementação) do Programa de Usuário (UserProg)	3-13
Figura 3-18. Comentário das Redes	3-13
Figura 3-19. Inclusão da Nova POU na Árvore de Dispositivos (Etapa 1)	3-19
Figura 3-20. Criação da POU Tipo Bloco Funcional <i>Motor_RV</i> (Etapa 2)	3-20
Figura 3-21. Declaração das Variáveis (Etapa 3)	3-20
Figura 3-22. Edição do Corpo/Rede 1/ Comando à Direita (Etapa 4)	3-21
Figura 3-23. Edição do Corpo / Rede 2 / Comando à Esquerda (Etapa 5)	3-21
Figura 3-24. Inclusão das Variáveis Simbólicas (Etapa 6)	3-21
Figura 3-25. Alterações no Programa MainPrg (Etapa 7)	3-22
Figura 3-26. Chamada do Bloco Funcional <i>Motor_RV</i> no Programa MainPrg (Etapa 8)	3-22
Figura 4-1. Sistema de Controle Discreto	4-1
Figura 4-2. Sistema de Controle Contínuo	4-3

# Introdução



Este Tutorial fornece um suporte didático-pedagógico ao treinamento presencial CP-Nexto, possibilitando ao usuário um primeiro contato orientado com o equipamento e enfatizando a programação básica do mesmo na linguagem Diagrama Ladder via programador MasterTool IEC XE. O documento inclui ainda considerações sobre instalação do CP, conexões elétricas e configurações básicas. Esta seção contém as divisões do tutorial e as simbologias e convenções utilizadas. O objetivo é informar ao usuário como o documento pode ser consultado.



# O Curso MP10



O curso MP10 introduz a Série Nexto de controladores programáveis enfatizando seus diferenciais no que tange a um avançado sistema de automação capaz de controlar, de forma distribuída e redundante, complexos processos industriais, além de automatizar máquinas e linhas de produção de alto desempenho.



Os objetivos desse curso são:

- Fazer conhecer as principais características, as opções de arquitetura, o painel frontal da UCP e seus diagnósticos, bem como forma de montagem mecânica e as conexões elétricas para a instalação no sistema de automação.
- Apresentar a sequência básica de tarefas envolvidas no planejamento de um projeto de automação para aplicação no dispositivo através da utilização do ambiente de programação MasterTool IEC XE.
- Estabelecer considerações sobre sistemas de controle industriais, definir os conceitos básicos da norma IEC 61131-3 e descrever os itens comuns às linguagens de programação. A linguagem de programação adotada é o Diagrama LADDER.
- Detalhar as configurações do subsistema de entrada/saída e comunicação básica entre o Nexto e o ambiente de programação para a efetivação do DOWNLOAD e UPLOAD da aplicação.

Esse curso é destinado aos iniciantes em automação, clientes da Altus, potenciais usuários, profissionais técnicos e engenheiros de áreas afins, assim como todos os interessados em adquirir conhecimentos sobre automação.

O curso tem uma duração de 24 horas e demanda como pré-requisito, a participação prévia do aluno no Curso MP1 — Introdução à Automação.

O projeto didático do curso é concebido de forma a embasar o ensino direto, utilizando metodologias que propiciam a integração entre a teoria e a prática e favorecem a capacidade de construção e gestão do conhecimento e o autodesenvolvimento contínuo.

Os recursos didáticos associados (Tutorial, Slides e Documentação de Produto) contemplam elementos de instrução de alta qualidade pedagógica e suficientemente compreensíveis, dinâmicos e atrativos. No seu contexto está prevista a flexibilização do ensino para o desenvolvimento de habilidades de autoaprendizagem.

# Estruturação do Tutorial



O Tutorial fornece um suporte didático-pedagógico ao treinamento presencial, possibilitando ao aluno um primeiro contato orientado com o(s) equipamento(s) e enfatizando a programação do(s) mesmo(s).

O documento inclui ainda considerações sobre instalação, conexões elétricas e configurações básicas. Ele é dividido em capítulos numa estrutura padronizada conforme esquema mostrado a seguir.

- ▷ 1. Hardware do CP Nexto
- ▷ 2. Software do CP Nexto
- ▷ 3. Programação do CP Nexto
- ▷ 4. Aplicações, Avaliação e Encerramento

Alguns capítulos contêm Estudos Dirigidos cuja implementação permitirá ao aluno consolidar os conceitos abordados no Tutorial, aplicando-os em situações-problema. Os estudos dirigidos são exercícios teórico/práticos de aprofundamento e aplicação dos temas já tratados. O objetivo é que os alunos resolvam as tarefas propostas de modo relativamente independente, a partir das diretrizes definidas pelo Instrutor.

Outra estratégia de aprendizado prevista nos cursos é a pesquisa na Documentação de Produto, permitindo ao aluno uma familiarização com o acesso a um vasto acervo de manuais, características técnicas, notas de aplicação etc.

## Ementa do curso

### Capítulo 1: Hardware do CP Nexto

Este capítulo descreve o hardware da Série Nexto. Informa suas principais características, as opções de arquitetura, o painel frontal da UCP e seus diagnósticos, bem como forma de montagem mecânica e as conexões elétricas para a instalação no sistema de automação. Tópicos abordados:

- ▲ 1. Hardware do CP Nexto
  - ▷ Características técnicas da Série Nexto
  - ▷ Arquitetura da Série Nexto
  - ▷ Diagnósticos e manutenção da Série Nexto
  - ▷ Instalação da Série Nexto

### Capítulo 2: Software do CP Nexto

O capítulo versa sobre o software de programação MasterTool IEC XE e sua utilização. São fornecidas orientações sobre as configurações básicas para o seu correto funcionamento. O capítulo apresenta a sequência básica de tarefas envolvidas no planejamento de um projeto de automação para aplicação no dispositivo. São realizadas algumas considerações sobre sistemas de controle industriais e apresentados os conceitos básicos da norma IEC 61131-3. Além disso, são descritos os itens comuns às linguagens de programação. São introduzidos também os conceitos de Configurações, Recursos, Programas, Tarefas, Blocos funcionais, Funções, Variáveis, Caminhos de acesso, POU's e Forma hierárquica. A linguagem de programação adotada é o Diagrama LADDER. Tópicos abordados:

- ▲ 2. Software do CP Nexto
  - ▷ Características do MasterTool IEC XE
  - ▷ Funcionalidades do MasterTool IEC XE
  - ▷ Interface do Usuário do MasterTool IEC XE
  - ▷ Criação e Execução de Projetos
  - ▷ Diagnósticos
  - ▷ Ajuda

### Capítulo 3: Programação do CP Nexto

Este capítulo detalha as configurações do subsistema de entrada/saída e comunicação básica entre o CP e o ambiente de programação para a efetivação do DOWNLOAD e UPLOAD da aplicação. Tópicos abordados:

- ▲ 3. Programação do CP Nexto
  - ▷ Elementos Básicos da Aplicação
  - ▷ Criação de POUs

### Capítulo 4: Aplicações, Avaliação e Encerramento

O tutorial finaliza com a proposição de estudos de caso para consolidação das competências desenvolvidas no treinamento. Tópicos abordados:

- ▲ 4. Aplicações, Avaliação e Encerramento
  - ▷ Aplicação 1: Controle Discreto
  - ▷ Aplicação 2: Controle Contínuo
  - ▷ Avaliação do Treinamento

## Referências



1. Fonseca, M. Apostila de suporte de curso: Norma IEC 61131-3 para Programação de Controladores. ISA Distrito 4 – América do Sul, 2008.
2. INTERNET (Sites diversos).
3. Documentos relacionados à Série Nexto (conforme indicado na sequência)

### Documentos relacionados

Para obter informações adicionais sobre a Série Nexto podem ser consultados outros documentos (manuais e características técnicas) além deste. Estes documentos encontram-se disponíveis em [www.altus.com.br](http://www.altus.com.br). Cada produto possui um documento denominado Características Técnicas (CT), e neste documento encontram-se as características do produto em questão. Caso o produto possua mais informações, ele pode ter também um manual de utilização. Por exemplo, o módulo NX2020 tem todas as informações de características, utilização e de compra, na sua CT. Por outro lado, o NX5001 possui, além da CT, um manual de utilização. A tabela abaixo indica a lista de todos os documentos relacionados à Série Nexto.

<b>Código</b>	<b>Descrição</b>	<b>Idioma</b>
<b>CE114000</b> <b>CT114000</b> <b>CS114000</b>	Nexto Series – Technical Characteristics Série Nexto – Características Técnicas Serie Nexto – Especificaciones y Configuraciones	English Portuguese Spanish
<b>CE114100</b> <b>CT114100</b> <b>Cs114100</b>	Nexto Series CPUs Technical Characteristics Características Técnicas UCPs Série Nexto Especificaciones y Configuraciones UCPs Serie Nexto	English Portuguese Spanish
<b>CE114200</b> <b>CT114200</b> <b>CS114200</b>	NX8000 Power Supply Module Technical Characteristics Características Técnicas Fonte de Alimentação NX8000 Características Tecnicas del Fuente de Alimentación NX8000	English Portuguese Spanish
<b>CE114810</b> <b>CT114810</b> <b>CS114810</b>	Nexto Series Accessories for Backplane Rack Technical Characteristics Características Técnicas Acessórios para Bastidor Série Nexto Características Tecnicas del Cierres Laterales para el Bastidor	English Portuguese Spanish
<b>CE114902</b> <b>CT114902</b> <b>CS114902</b>	Nexto Series PROFIBUS-DP Master Technical Characteristics Características Tecnicas do Mestre PROFIBUS-DP da Série Nexto Características Técnicas del Módulo Profibus-DP Maestro	English Portuguese Spanish
<b>CE114903</b> <b>CT114903</b> <b>CS114903</b>	Nexto Series Ethernet Module Technical Characteristics Características Técnicas Módulo Ethernet Série Nexto Características Tecnicas del Modulo Ethernet Série Nexto	English Portuguese Spanish
<b>CE114700</b> <b>CT114700</b> <b>CS114700</b>	Nexto Series Backplane Rack Technical Characteristics Características Técnicas dos Bastidores da Série Nexto Características Tecnicas de los Bastidores de la Serie Nexto	English Portuguese Spanish
<b>CE114900</b> <b>CT114900</b> <b>CS114900</b>	NX4010 Redundancy Link Module Technical Characteristics Características Técnicas do Módulo de Redundância NX4010 Características Tecnicas del Módulo de Redundancia NX4010	English Portuguese Spanish
<b>MU216600</b> <b>MU216000</b>	Nexto Xpress User Manual Manual de Utilização Nexto Xpress	English Portuguese
<b>MU214600</b> <b>MU214000</b>	Nexto Series User Manual Manual de Utilização Série Nexto	English Portuguese
<b>MU214605</b> <b>MU214100</b>	Nexto Series CPUs User Manual Manual de Utilização UCPs Série Nexto	English Portuguese
<b>MU299609</b> <b>MU299048</b>	MasterTool IEC XE User Manual Manual de Utilização MasterTool IEC XE	English Portuguese
<b>MP399609</b> <b>MP399048</b>	MasterTool IEC XE Programming Manual Manual de Programação MasterTool IEC XE	English Portuguese
<b>MU214601</b> <b>MU214001</b>	NX5001 PROFIBUS DP Master User Manual Manual de Utilização Mestre PROFIBUS-DP Nexto	English Portuguese
<b>MU214608</b> <b>MU214108</b>	Nexto PROFIBUS-DP Head Utilization Manual Manual de Utilização da Cabeça PROFIBUS-DP Nexto	English Portuguese

## Avaliação do curso



A avaliação continuada e individualizada é uma prática corrente adotada pelo Instrutor no curso. Desta forma, as seguintes estratégias de avaliação são consideradas: acompanhamento do desenvolvimento do aluno (frequência, participação, postura, interesse e construção do conhecimento) durante o curso pelo Suporte Técnico e autoavaliação via resolução das tarefas propostas caracterizadas pelos Estudos Dirigidos, Pesquisa na Documentação e Aplicações.

## Terminologia



Neste Tutorial, as palavras “software” e “hardware” são empregados livremente, por sua generalidade e frequência de uso. Por este motivo, apesar de serem vocábulos em inglês, aparecerão no texto sem aspas.

As seguintes expressões podem ser empregadas no texto da Tutorial.

**CP:** Controlador Programável - equipamento composto por uma UCP, módulos de entrada e saída e fonte de alimentação.

**UCP ou CPU:** Unidade Central de Processamento é o módulo principal do CP, que realiza o processamento dos dados.

**MasterTool IEC XE:** identifica o programa para microcomputador executável em ambiente WINDOWS®, que permite o desenvolvimento de aplicativos para os CPs da Série Nexto.

## Convenções Utilizadas



Os símbolos utilizados ao longo deste manual possuem os seguintes significados:

- Este marcador indica uma lista de itens ou tópicos.

MAIÚSCULAS PEQUENAS indicam nomes de teclas, por exemplo, ENTER.

TECLA1+TECLA2 é usado para teclas a serem pressionadas simultaneamente. Por exemplo, a digitação simultânea das teclas CTRL e END é indicada como CTRL+END.

TECLA1, TECLA2 é usado para teclas a serem pressionadas sequencialmente. Por exemplo, a mensagem “Digite ALT, F10” significa que a tecla ALT deve ser pressionada e liberada e então a tecla F10 pressionada e liberada.

Maiúsculas GRANDES indicam nomes de arquivos e diretórios.

*Itálico* indica palavras e caracteres que são digitados no teclado ou vistos na tela. Por exemplo, se for solicitado a digitar *FACAO*, estes caracteres devem ser digitados exatamente como aparecem no manual.

**NEGRITO** é usado para nomes de comandos ou opções, ou para enfatizar partes importantes do texto.

As mensagens de advertência apresentam os seguintes formatos e significados:



### **PERIGO:**

**O rótulo PERIGO indica que risco de vida, danos pessoais graves ou prejuízos materiais substanciais resultarão se as precauções necessárias não forem tomadas.**

**CUIDADO:**

O rótulo CUIDADO indica que risco de vida, danos pessoais graves ou prejuízos materiais substanciais podem resultar se as precauções necessárias não forem tomadas.

**ATENÇÃO:**

O rótulo ATENÇÃO indica que danos pessoais ou prejuízos materiais mínimos podem resultar se as precauções necessárias não forem tomadas.

## Suporte Técnico



Formada por especialistas experientes e com grande conhecimento, a equipe de Suporte Técnico da ALTUS é altamente capacitada para prestar assistência técnica, preventiva e corretiva, para todos os produtos comercializados pela empresa...

**Precisa de ajuda? Fale com nosso Suporte Técnico!**

Formada por especialistas experientes e com grande conhecimento, nossa equipe de Suporte Técnico é altamente capacitada para prestar assistência técnica, preventiva e corretiva, para todos os produtos comercializados pela Altus. Estamos 100% disponíveis para resolver problemas, tirar dúvidas e lhe ajudar a otimizar o desempenho da sua aplicação.

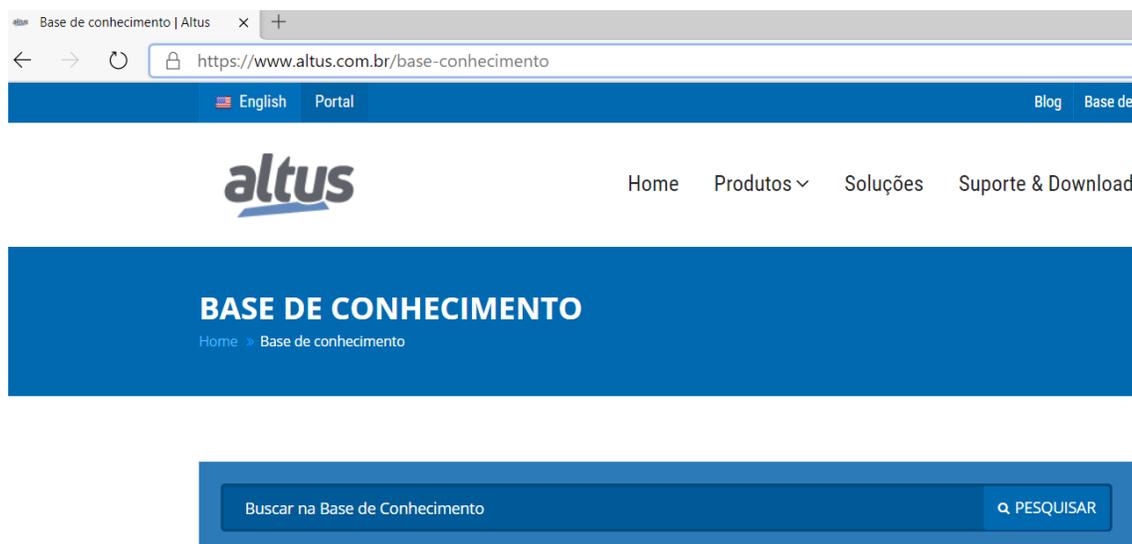
- Help Desk**  
Atendimento através do  
0800 510 9500 | +55 51 3589 9546  
De segunda à sexta - das 8h às 22h  
Sábados - das 8h às 17h  
Indisponível em domingos e feriados
- Downloads**  
Softwares, Documentação Técnica, Drivers, entre outros. Clique aqui e encontre as informações necessárias para você ter a melhor experiência com os produtos da Altus.
- Integradores**  
Conheça os profissionais treinados e certificados para utilizar a tecnologia Altus. Descubra o integrador mais próximo de você e garanta a qualidade do seu projeto!
- Base de conhecimento**  
Aqui você encontra tutoriais, dicas e outros materiais de apoio criados por nossos especialistas para qualificar a sua experiência com os produtos e soluções Altus!

Para mais informações acesse: <https://www.altus.com.br/suporte>.

## Base de Conhecimento ALTUS



No site ALTUS você encontra também vários tutoriais que auxiliam na implementação de aplicações específicas no espaço conhecido como Base de Conhecimento ALTUS (<https://www.altus.com.br/base-conhecimento>). Isso inclui: conexões, instalações, utilização de módulos função e configurações diversas...



## Cursos ALTUS



A Altus possui um calendário de treinamentos anual para profissionais da área de automação industrial interessados em desenvolver aplicações, realizar reformas e prestar manutenção em sistemas de controle de processos. Os cursos, que abordam as linhas de produtos da empresa, têm como objetivo introduzir conceitos sobre automação industrial e capacitar os participantes a desenvolver aplicações para controle de processos. Também são realizados cursos especiais e IN COMPANY, de acordo com a solicitação e necessidade dos clientes. Confira as opções de treinamento disponíveis no site ALTUS.

## Revisões deste Tutorial



O código de referência, a revisão e a data do presente documento estão indicadas na capa. A mudança da revisão pode significar alterações da especificação funcional ou melhorias no mesmo. A seguir as alterações correspondentes a cada revisão deste:

Revisão	Data	Descrição
A	Junho/12	Emissão do Documento
B	Abril/15	Revisão geral do documento
C	Setembro/20	Atualização do documento

## Training Box Nexto

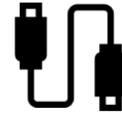


Na execução desse curso é utilizada a Maleta de Treinamento (TRAINING BOX) Nexto baseada no controlador programável de última geração Nexto Altus interligado a uma IHM da série X2 conforme mostrado na figura a seguir.



Para detalhes relacionados à funcionalidade da maleta de treinamento consulte o instrutor ou acesse o site Altus para DOWNLOAD da documentação associada.

# 1. Hardware do CP Nexto



## Características técnicas da Série Nexto



### O que é o Nexto?

Nexto é um poderoso e completo Controlador Programável (CP) destinado a atender as exigências de sistemas de controle de aplicações de pequena e alta complexidade. As UCPs oferecem diversas características interessantes, alta performance e arquitetura flexível permitindo, assim, o seu uso na automação de grandes processos e de máquinas de tempo crítico. Além disto, elas combinam conectividade com os mais atuais barramentos de campo industriais de alta velocidade.



Figura 1-1. A UCP Nexto

## Porque o Nexto?

A Série Nexto é uma poderosa e completa série de Controladores Programáveis (CPs) com características exclusivas e inovadoras. Devido a sua flexibilidade, design funcional, recursos de diagnóstico avançado e arquitetura modular, o CP Nexto pode ser usado para controle de sistemas em aplicações de médio e grande porte ou em máquinas com requisitos de alto desempenho.

A arquitetura da Série Nexto fornece uma grande variedade de módulos de entradas e saídas. Estes módulos, combinados com uma poderosa UCP de 32 bits e um barramento baseado em Ethernet de alta velocidade, atendem a muitas aplicações de usuário, tais como controle rápido de máquinas, complexas aplicações de processos distribuídos e redundantes ou até mesmo grandes sistemas de E/S para automação predial. Entre outras funcionalidades, a Série Nexto oferece módulos para controle de movimento, comunicação e interfaces com as mais conhecidas redes de campo.

A Série Nexto utiliza uma avançada tecnologia de barramento baseada em Ethernet de alta velocidade que permite que as entradas, saídas e informações processadas sejam compartilhadas entre todos os módulos do sistema. Os módulos de E/S podem ser facilmente distribuídos no campo e podem ser usados tanto como E/S locais quanto remotas, sem nenhuma perda no desempenho.

Adicionalmente, a Série Nexto apresenta uma ferramenta completa para programação, configuração, simulação e depuração da aplicação do usuário: o MasterTool IEC XE. Trata-se de um software flexível e de fácil utilização que fornece seis linguagens de programação definidas pela norma IEC 61131-3: Texto Estruturado (ST), Sequenciamento Gráfico de Funções (SFC), Diagrama de Blocos Funcionais (FBD), Diagrama Ladder (LD), Lista de Instruções (IL) e Gráfico Funcional Contínuo (CFC). O MasterTool IEC XE permite o uso de diferentes linguagens na mesma aplicação, fornecendo ao usuário uma poderosa forma de organizar a aplicação e reaproveitar códigos usados em aplicações anteriores.

Outros módulos da Série Nexto formam a solução Nexto Jet, a qual consiste em um conjunto ideal de entradas e saídas para aplicações de pequeno e médio porte, além de sistemas distribuídos. A solução apresenta módulos compactos de alto desempenho que são usados em conjunto com as UCPs, bastidores, módulos de comunicação e de redes de campo, além do software MasterTool IEC XE. Os módulos Nexto Jet agregam mais versatilidade e competitividade à consagrada Série Nexto, mantendo a flexibilidade, arquitetura modular e recursos de diagnósticos avançados.

## Quais São Seus Diferenciais?

### Alto Desempenho

Preemptivo e multitarefa, o sistema operacional das UCPs da Série Nexto permite sofisticadas aplicações e controle total da atuação do processador RISC PowerPC de 32 bits. Aliando desempenho e menor consumo de energia, o processador dispensa partes móveis e contribui para um maior MTBF (sigla em inglês para tempo médio entre falhas).

### Conectividade Total

Através das portas integradas nas CPUs e interfaces de comunicação, a Série oferece suporte de hardware e software aos protocolos de redes de campo MODBUS RTU, MODBUS TCP, MODBUS RTU/TCP, PROFIBUS-DP, PROFINET, IEC 60870-5-104 Servidor, EtherNet/IP, EtherCAT Master, OPC DA, OPC UA e MQTT. Ainda, em casos de necessidade o usuário pode desenvolver seus próprios protocolos, permitindo integração total com qualquer dispositivo industrial.

### **Disponibilidade**

Para aplicações que não podem parar, como sistemas de O&G, Açúcar & Etanol, Siderurgia, entre outros, a família Nexto conta com CPUs redundantes e troca a quente de seus módulos, recursos que praticamente eliminam a necessidade de parada do sistema para manutenção. A série também tem capacidade de expansão de módulos de E/S sem parada do sistema e endereçamento automático dos módulos nos bastidores.

### **Variáveis de Rede**

Inteligentes e versáteis, as unidades de processamento da Série possuem atributos que permitem a criação de variáveis de rede. Esse recurso possibilita ao usuário gerar ou modificar variáveis específicas em um CP e replicá-las para os demais controladores conectados à rede, o que diminui o tempo de manutenção e as horas de engenharia para desenvolvimento de sistemas.

### **Amigável com o Meio Ambiente**

A maioria dos módulos conta com proteção em seus componentes e placas eletrônicas (*conformal coating*), o que proporciona vida útil superior mesmo em ambientes hostis, e, ainda, não utiliza chumbo no processo de manufatura, compatível com a diretiva RoHS. Os produtos da Série possuem grande capacidade de memória retentiva, sem necessidade de baterias, e um relógio de tempo real (RTC) com longa autonomia. Isto é possível devido à tecnologia BFO (*Battery Free Operation*), que emprega componentes de hardware e algoritmos de software que eliminam a necessidade de baterias internas para manutenção das informações, reduzindo o impacto ambiental no descarte de elementos.

### **Arquitetura Escalar**

Baseado em Ethernet determinística, o barramento interno pode ser expandido em múltiplos bastidores remotos, sem perda de desempenho. Uma CPU pode controlar até 320 pontos E/S em um único bastidor e até 24 bastidores remotos podem ser conectados ao bastidor local.

A série é totalmente compatível com os serviços Web, como: Ambiente de criação de Páginas Web para supervisão de processos (*WebServer*); Servidor de páginas Web nativo para visualização dos diagnósticos e atualização do produto; SNTP para sincronismo de relógio; SNMP para suporte à gerenciamento de rede Ethernet TCP/IP.

### **Certificações Internacionais**

A alta qualidade dos controladores programáveis Nexto é atestada por renomados institutos tecnológicos de classe mundial. Além de serem concebidos para atender os requisitos das diretivas europeias (CE), os produtos também ostentam importantes certificados internacionais. Os equipamentos Nexto contam com os certificados UL, categoria NQAQ (regida pelas normas de segurança UL61010-1 e UL61010-2-201), concedido pela *Underwriters Laboratories*; DNV-GL, categoria *TYPE Approval*, fornecido pelo Grupo DNV-GL; e EAC, classes TR004 e TR020, requisito para países como Rússia, Bielorrússia e Cazaquistão.

### **Expansão de Barramento**

Alto desempenho e recursos avançados para sistemas que demandem um grande número de pontos de E/S.

### **Sistemas Distribuídos**

Maior alcance para a sua rede de controle, ideal para automações prediais, de refrigeração e saneamento.

## Disponibilidade Total

Redundâncias de CPU e troca de módulos a quente, perfeito para aplicações que não podem parar. A figura a seguir ilustra uma solução Nexto para disponibilidade total.

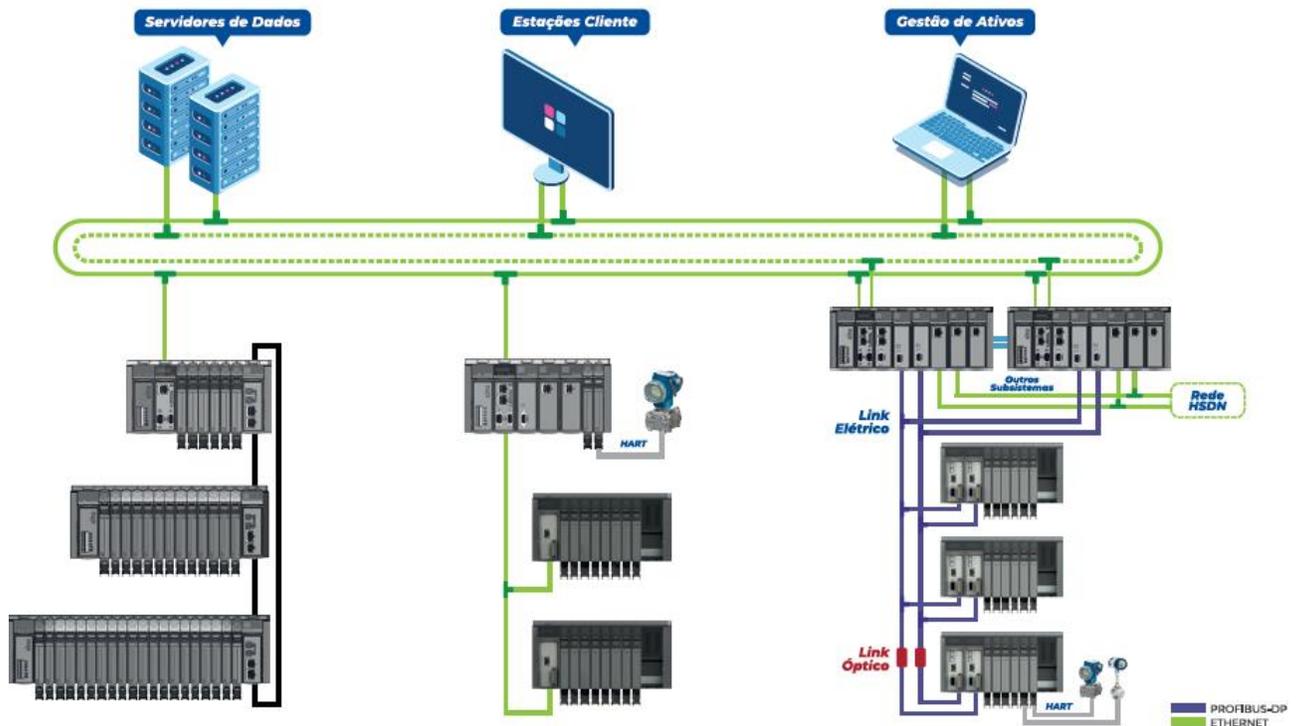


Figura 1-2. A solução Nexto para disponibilidade total

## Diagnósticos Sofisticados

O botão de diagnóstico presente nos produtos da série (exceto nos módulos da solução Nexto Jet e Xpress), possui a característica *One Touch Diag* (OTD) que apresenta informações avançadas do sistema, como curto-circuito nas saídas, endereço IP, tags alfanuméricas, entre outras, o que ajuda no comissionamento e dispensa a consulta a documentos técnicos durante a manutenção. Em conjunto com a *Eletronic Tag on Display* (ETD), a ferramenta permite a visualização de tags do módulo de E/S no visor gráfico da CPU.

## Armazenamento dos dados

O *Multiple Block Storage* (MBS) é uma característica que disponibiliza diferentes memórias para armazenamento de programas, códigos-fonte comentado, operandos, retentividade e registro de eventos. O recurso também permite o armazenamento de memória de massa em um cartão mini SD, função utilizada para arquivos de usuário, registro de dados da aplicação (*data logging*) e documentação de projeto através da característica *Onboard Full Documentation* (OFD).

## Armazenamento dos dados

A Série Nexto conta com uma ampla linha de módulos de E/S, interfaces de comunicação e módulos especiais. Seu design compacto e modular otimiza o espaço dentro do painel elétrico por meio do *Double Hardware Width* (DHW), característica que permite uma combinação de módulos com 18 ou 36 milímetros de largura. Os módulos (EPS), um mecanismo prático e seguro de inserção e extração de bornes que dispensa a utilização de ferramentas auxiliares.

## **Quais as suas características principais?**

### **UCPs**

As UCPs apresentam várias funções integradas, programação online, alta capacidade de memória e diversos canais seriais.

Estes dispositivos possuem um moderno processador de 32 bits que proporciona um excelente desempenho e a integração de diversas funcionalidades. Até mesmo a versão de UCP mais simples conta com porta Ethernet para programação, uso em redes TCP MODBUS e servidor web embarcado. As UCPs NX3010, NX3020 e NX3030 apresentam duas interfaces seriais para conexão de IHMs locais e redes MODBUS, e possuem uma posição de cartão de memória para hospedagem das páginas da web, código fonte da aplicação, entre outros arquivos do usuário. As UCPs NX3003, NX3004 e NX3005 apresentam uma interface serial para conexão com IHMs locais e redes MODBUS e não possuem interface para cartão de memória.

Todas as UCPs possuem páginas da web com status completos e lista de diagnósticos, atualização de firmware, entre outras características.

### **Módulos**

Os módulos apresentam E/S de alta densidade. A Série Nexto disponibiliza dois formatos de módulos – simples e duplos – o que permite uma melhor combinação de E/S para muitas aplicações onde a alta densidade e o tamanho de painel reduzido são requisitos. Cada módulo de E/S apresenta um visor para diagnósticos locais no qual é mostrado o estado de cada ponto de E/S. Há também os diagnósticos multifuncionais sobre o status dos módulos. Todas as informações de diagnóstico também podem ser acessadas remotamente pela UCP, cabeça de rede de campo ou pela ferramenta de configuração MasterTool IEC XE.

Os módulos Nexto de E/S com tampa frontal possuem etiquetas que podem ser utilizadas para identificar os módulos e os bornes.

### **Barramento de Alta Velocidade**

A arquitetura da Série Nexto apresenta um moderno barramento baseado em Ethernet. A alta taxa de transferência permite a atualização de grandes quantidades de entradas em um curto período de tempo. Além disto, aplicações de tempo crítico, como controle de movimento, são possíveis com esta tecnologia. Os módulos são endereçados e identificados automaticamente, evitando erros durante a configuração da aplicação e manutenção de campo. Os barramentos NX9000, NX9001, NX9002 e NX9003 fornecem características especiais que permitem redundância da UCP no mesmo bastidor ou em bastidores diferentes usando os Módulos de Link de Redundância: endereçamento e identificação de módulo automática; troca a quente (exceto NX9010); barramento serial baseado em Ethernet 100 Mbps; sincronização de tempo para atualização de E/S ou estampa de tempo precisa; solução de hardware de chip único.

### **Inserção & Remoção de Bornes**

A Série Nexto apresenta um mecanismo inovador patenteado para inserção e extração de bornes de módulos de E/S. Em muitas aplicações de automação, a densidade das E/S é alta, o que torna complexa a fiação de campo. No momento da manutenção, as complicações são ainda maiores devido à dificuldade de alcançar os fios. Algumas vezes é necessário remover alguns módulos próximos para acessar o fio desejado. Na Série Nexto, estes problemas são resolvidos combinando um borne especial com uma tampa frontal. Além da fácil remoção e inserção dos bornes, o design torna a aparência do painel atrativa e organizada.

### **Robustez**

O design da Série Nexto é extremamente robusto e permite o uso em aplicações onde há vibração mecânica. Aplicações de transporte ou locais em movimento são exemplos de situações que exigem esta função, a qual é fornecida sem o uso de parafusos ou montagem especial. O design de todo o produto foi desenvolvido de forma a oferecer esta possibilidade sem comprometer os procedimentos de instalação e manutenção.

### **Troca a quente**

A característica de troca a quente permite a substituição de módulos sem que o sistema esteja desenergizado. A UCP mantém o controle de todo o processo e os módulos Nexto podem ser substituídos sempre que necessário.

#### **ATENÇÃO:**

Os módulos que fazem parte da solução Nexto Jet não possuem essa funcionalidade.

### **Alta Disponibilidade**

A Série Nexto oferece diversas arquiteturas diferentes de redundância, onde UCPs, Fontes de Alimentação e Interfaces de Rede de Campo podem ser montadas em uma aplicação redundante. Com esta flexibilidade o sistema pode ser ajustado desde sistemas simples sem nenhuma redundância até complexas e críticas aplicações onde a alta disponibilidade é essencial.

### **Diagnósticos Avançados**

Cada módulo contém seus próprios diagnósticos: as UCPs, Interfaces de Rede de Campo, Fontes de Alimentação e módulos de E/S apresentam vários diagnósticos disponíveis. Cada módulo tem um visor multifuncional que informa o status do mesmo. Cada módulo Nexto que conta com um botão na sua parte superior pode fornecer informações de diagnósticos diferentes para a equipe de manutenção. Estes diagnósticos podem ser monitorados no campo através de visores ou através da ferramenta de configuração. Alguns exemplos são:

- Módulo localizado em uma posição incorreta no bastidor;
- Ausência de fonte de alimentação;
- Curto circuito nas saídas;
- Visualização de tag e descrição de E/S no CP para módulos com OTD;
- Visualização de endereço IP.

### **Capacidades**

A característica de troca a quente permite a substituição de módulos sem que o sistema esteja desenergizado. A UCP mantém o controle de todo o processo e os módulos Nexto podem ser substituídos sempre que necessário. O maior bastidor pode conter até 24 módulos de 18 mm ou 12 módulos de 36 mm. Com esta arquitetura uma UCP única pode controlar 320 pontos de E/S usando apenas um bastidor. De acordo com o modelo da UCP, é possível expandir para até 25 bastidores utilizando-se os módulos de expansão de barramento.

### **O que o Nexto traz de inovação?**

A Série Nexto apresenta ao usuário várias inovações na utilização, supervisão e manutenção do sistema. A tabela a seguir mostra alguns destes recursos disponíveis nas UCPs da Série Nexto.

Tabela 1-1. Inovações da Série Nexto

<b>Battery Free Operation</b>	A Série Nexto não requer nenhum tipo de bateria para manutenção de memória e operação de relógio de tempo real. Esta funcionalidade é extremamente importante porque reduz a necessidade de manutenção do sistema e permite o uso em locais remotos de difícil manutenção. Além disso, esta característica é ambientalmente correta.	
<b>Easy Plug System</b>	A Série Nexto conta com um exclusivo método para conectar e desconectar bornes de E/S. Estes bornes são facilmente removíveis com um simples movimento e sem ferramentas especiais. Para conectar o borne novamente ao módulo, a tampa frontal auxilia o procedimento de inserção, encaixando o borne ao módulo.	
<b>Multiple Block Storage</b>	Diversos tipos de memória estão disponíveis nas UCPs da Série Nexto, oferecendo a melhor opção para cada necessidade. Estas memórias são divididas em memórias voláteis e memórias não voláteis. Para uso de memórias voláteis, as UCPs da Série Nexto oferecem variáveis de entrada de representação direta (%I), variáveis de saída de representação direta (%Q), variáveis de memória de representação direta (%M), memória de dados e memória de dados redundantes. Para aplicações que necessitam funcionalidades de memória não volátil, a Série Nexto possibilita a utilização de variáveis de representação direta de memória retentiva (%Q), memória retentiva de dados, variáveis de representação direta de memória persistente (%Q), memória persistente de dados, memória de programa, memória de código fonte, sistema de arquivo na UCP (Doc, pdf dados) e interface para cartão de memória.	
<b>One Touch Diag TM</b>	Esta é uma característica exclusiva dos CPs da Série Nexto. Através deste novo conceito, o usuário pode checar as informações de diagnóstico de qualquer módulo do sistema diretamente no visor gráfico da UCP, mediante apenas um pressionamento no botão de diagnóstico do respectivo módulo. A OTD é uma poderosa ferramenta de diagnóstico que pode ser usada offline (sem supervisor ou programador) e reduz os tempos de manutenção e comissionamento.	
<b>OFD – On Board Full Documentation</b>	As UCPs da Série Nexto têm a capacidade de armazenar a documentação completa do projeto na sua memória. Este é um recurso interessante para fins de backup e manutenção, já que informação completa fica armazenada em um único e seguro local.	
<b>ETD – Electronic Tag on Display</b>	Outra característica exclusiva apresentada pela Série Nexto é o ETD. Esta nova funcionalidade possibilita a verificação da tag de qualquer ponto ou módulo de E/S usado no sistema, diretamente no visor gráfico das UCPs. Juntamente com esta informação, o usuário pode também verificar a descrição. Este é um recurso extremamente útil durante a manutenção e resolução de problemas.	
<b>DHW – Double Hardware Width</b>	Os módulos da Série Nexto foram projetados para economizar espaço em painéis e nas máquinas. Por esta razão, a Série Nexto oferece duas diferentes larguras de módulos: largura dupla (com ocupação de 2 posições do bastidor) e largura simples (com ocupação de 1 posição do bastidor). Este conceito permite o uso de módulos de E/S compactos, com alta densidade de pontos de E/S, juntamente com módulos complexos, como UCPs, mestres de rede de campo e módulos de fonte de alimentação.	
<b>UCP de alta velocidade</b>	Todas as UCPs desta Série Nexto foram concebidas para fornecer ao usuário um excelente desempenho e atender a uma ampla gama de exigências nas aplicações. Por exemplo: as UCPs Nexto podem executar instruções de adição, multiplicação e subtração em menos de 15 ns para valores de tipo inteiro e em menos de 23 ns para valores de tipo real. Elas são igualmente capazes de executar 1000 laços PIDs em menos de 5 ms.	
<b>iF Product Design Award 2012</b>	A Série Nexto foi vencedora do iF Product Design Award 2012 no grupo industry + skilled trades. Este prêmio é reconhecido internacionalmente como um selo de excelência e qualidade, considerado o Oscar do design na Europa.	

### **Programação da UCP & Atualização de Firmware**

A Série Nexto permite a programação da UCP e a atualização do firmware através da porta Ethernet da UCP. Esta abordagem oferece algumas funcionalidades, tais como:

- Porta Ethernet multifuncional usada para compartilhar programação, intercâmbio de dados ponto a ponto, protocolo de dispositivo de terceiros na camada de aplicação, intercâmbio de dados das variáveis de rede etc.;
- Acesso direto às variáveis locais da UCP;
- Acesso remoto via interface Ethernet;
- Atualização de firmware via interface Ethernet.

# Arquitetura da Série Nexto



A Série Nexto é destinada a diferentes aplicações, que vão desde automações de pequenas máquinas de alta velocidade até a automação de processos de alta complexidade. Por esta razão, o sistema é muito flexível e modular, permitindo várias configurações diferentes sem comprometer o custo e o desempenho.

Os módulos de IOs Nexto e Nexto Jet não podem ser utilizados simultaneamente no mesmo barramento, ou seja, o uso de módulos misturados no mesmo barramento local/remoto não é válido, conseqüentemente a aplicação não será executada pela UCP escolhida.

Os principais componentes da arquitetura da Série Nexto são descritos a seguir.

## UCP

A UCP é responsável pela execução de todas as funções lógicas e de controle. O ciclo básico da UCP é composto por: leitura das entradas, execução dos algoritmos da aplicação e lógica, escrita nas saídas e execução de processos de comunicação com o sistema de supervisão e redes de campo.

## Módulo de Fonte de Alimentação (PSU)

O Módulo de Fonte de Alimentação fornece energia aos módulos instalados nos bastidores. Cada bastidor deve ter seu próprio Módulo de Fonte de Alimentação. Os requisitos de corrente da aplicação são mostrados na ferramenta de configuração.

## Barramento

Um sistema típico consiste em um bastidor local (UCP e seus módulos de E/S locais) e bastidores remotos (conjuntos de E/S remotos). Para o bastidor local, a arquitetura da Série Nexto oferece uma tecnologia de barramento Ethernet de 100 Mbps de tempo real e alta velocidade. Uma vez que é baseado em Ethernet, o barramento local pode ser facilmente estendido para bastidores remotos, usando-se cabos Ethernet (até 100 m) e dispositivos chamados de módulos de expansão de barramento.

Estes dispositivos convertem os sinais internos no padrão 100BASE-TX. O módulo de expansão de barramento pode ser usado em modo redundante, para obter uma arquitetura altamente confiável. Cada bastidor pode ter até 24 módulos e o sistema pode endereçar até 25 bastidores.

## Bastidores

Os bastidores apresentam um chassi de alumínio especial com uma placa de circuito impresso onde todos os módulos são conectados. Os módulos são montados diretamente no painel e oferecem alta imunidade contra interferência eletromagnética e descargas eletrostáticas (ESD) (se as regras de aterramento recomendadas forem executadas durante a fase de instalação).

## Módulos de E/S

Os módulos de E/S são conectados nos bastidores para a aquisição dos diferentes tipos de sinais de campo e envio dos mesmos para a UCP ou às cabeças de rede de campo. A Série Nexto suporta uma ampla variedade de tipos de E/S e faixas de operação, atendendo, assim, às típicas necessidades de um sistema de automação. Os módulos Nexto suportam troca a quente, ou seja, eles podem ser desconectados sem necessariamente desativar o sistema nem remover a energia, já os módulos que compõem a solução Nexto Jet não possuem essa funcionalidade. Devido às características de isolação, alguns módulos de E/S devem ser alimentados por fontes de alimentação externas.

### Cabeças de Rede de Campo

As cabeças de rede de campo conectam os módulos da Série Nexto a diferentes redes de campo. Elas podem comunicar-se com UCPs de diferentes fornecedores e suportam diversos protocolos, tais como MODBUS, PROFIBUS-DP e outros.

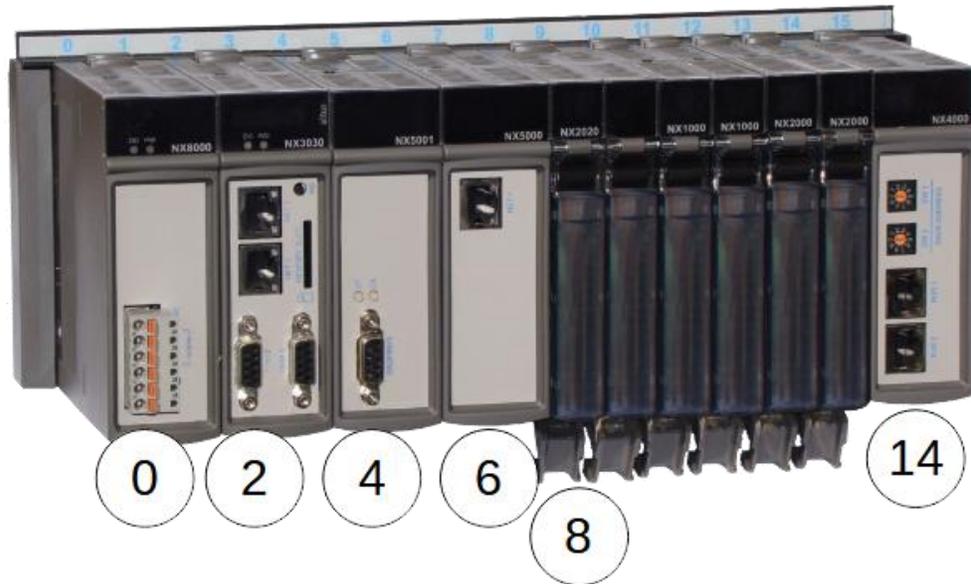
### Interface de Rede de Campo

As interfaces de rede de campo são nós mestres de barramentos de campo e permitem o acesso tanto a módulos remotos quanto a outros equipamentos do tipo utilizados na indústria como, por exemplo, PROFIBUS-DP, MODBUS e outros. As interfaces de rede de campo são conectadas em bastidores locais e necessitam de duas posições.



### Estudo Dirigido 1-1: módulos Nexto

Tomando como base a arquitetura da Série Nexto, identifique na figura abaixo o tipo de módulo a que se refere a indicação de posição no bastidor.



Resolução do Estudo Dirigido 1-1:

Posição	Módulo(s)
0	Fonte de alimentação
2	UCP
4	Mestre PROFIBUS DP
6	Módulo ETHERNET
8	Módulos de E/S
14	Módulo acoplador de barramento



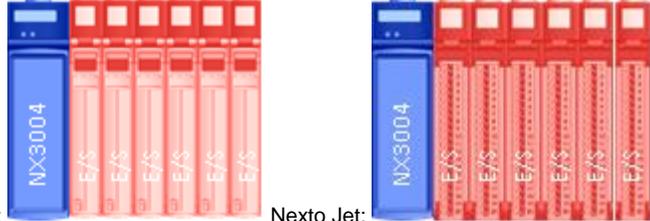
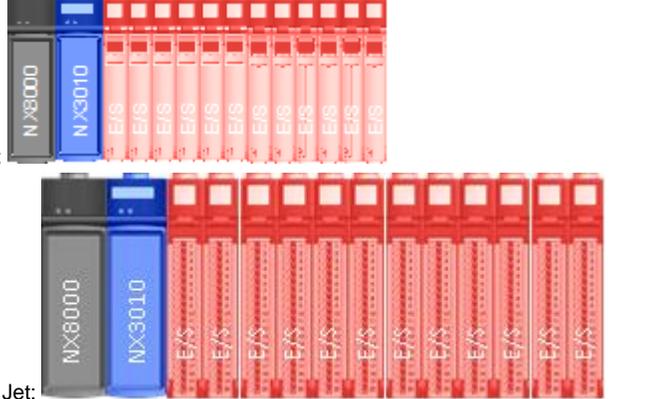
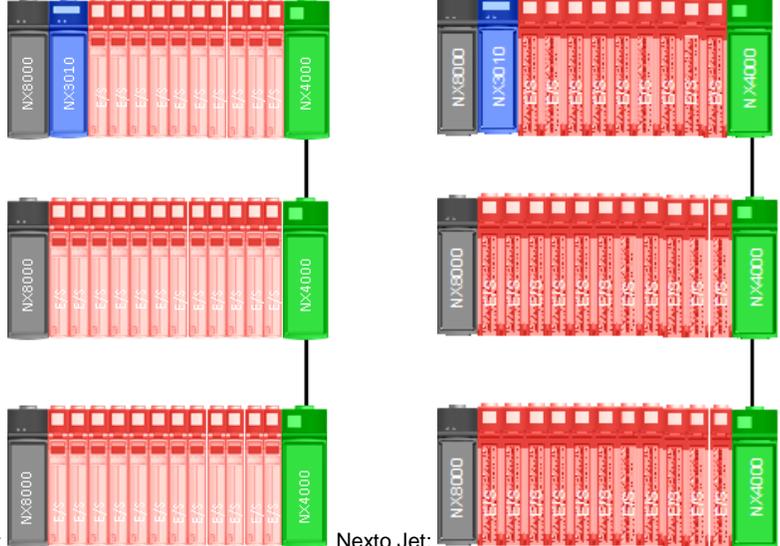
## Estudo Dirigido 1-2: arquiteturas Nexto

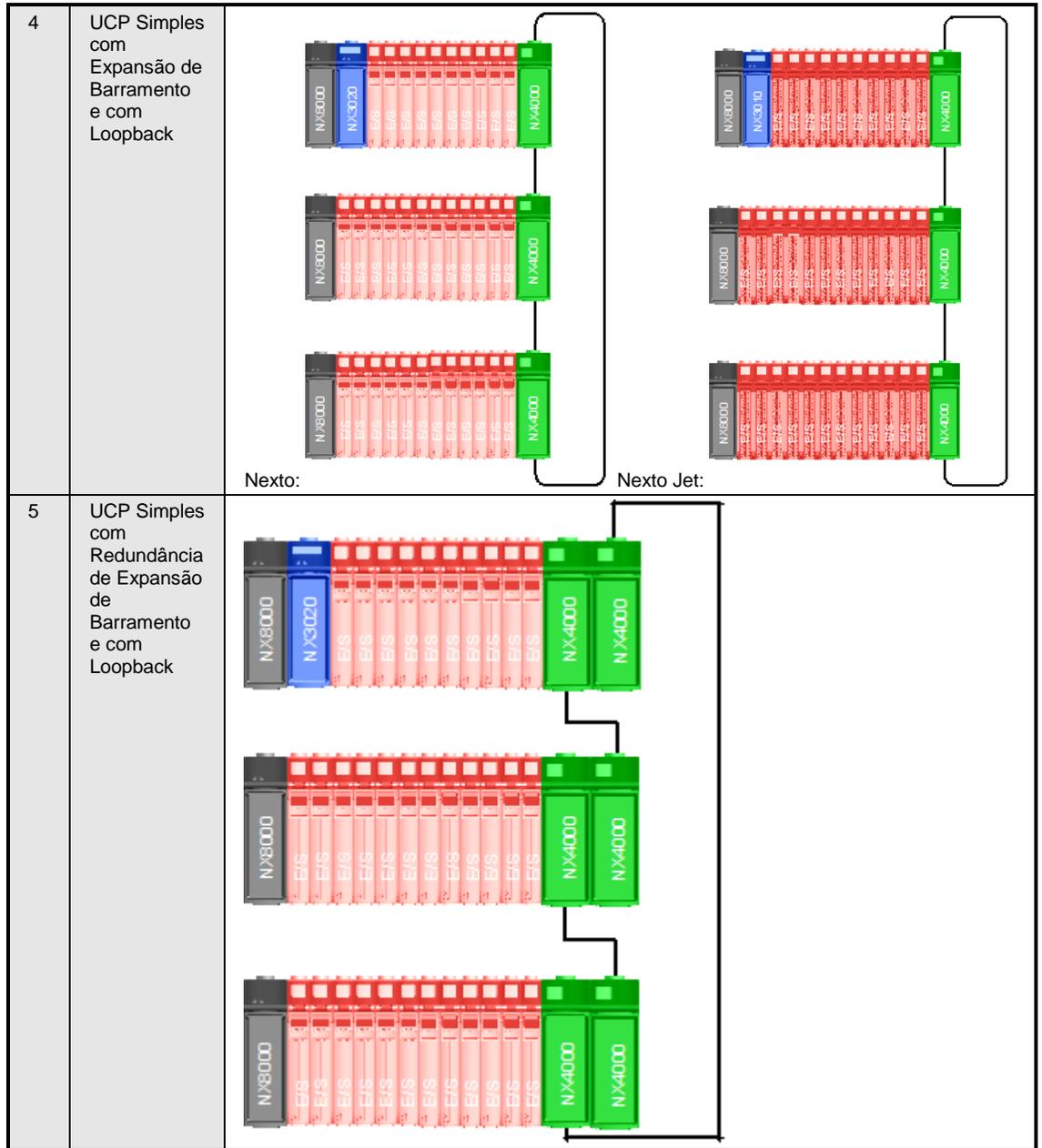
Correlacione as arquiteturas descritas na tabela com os itens listados à esquerda.

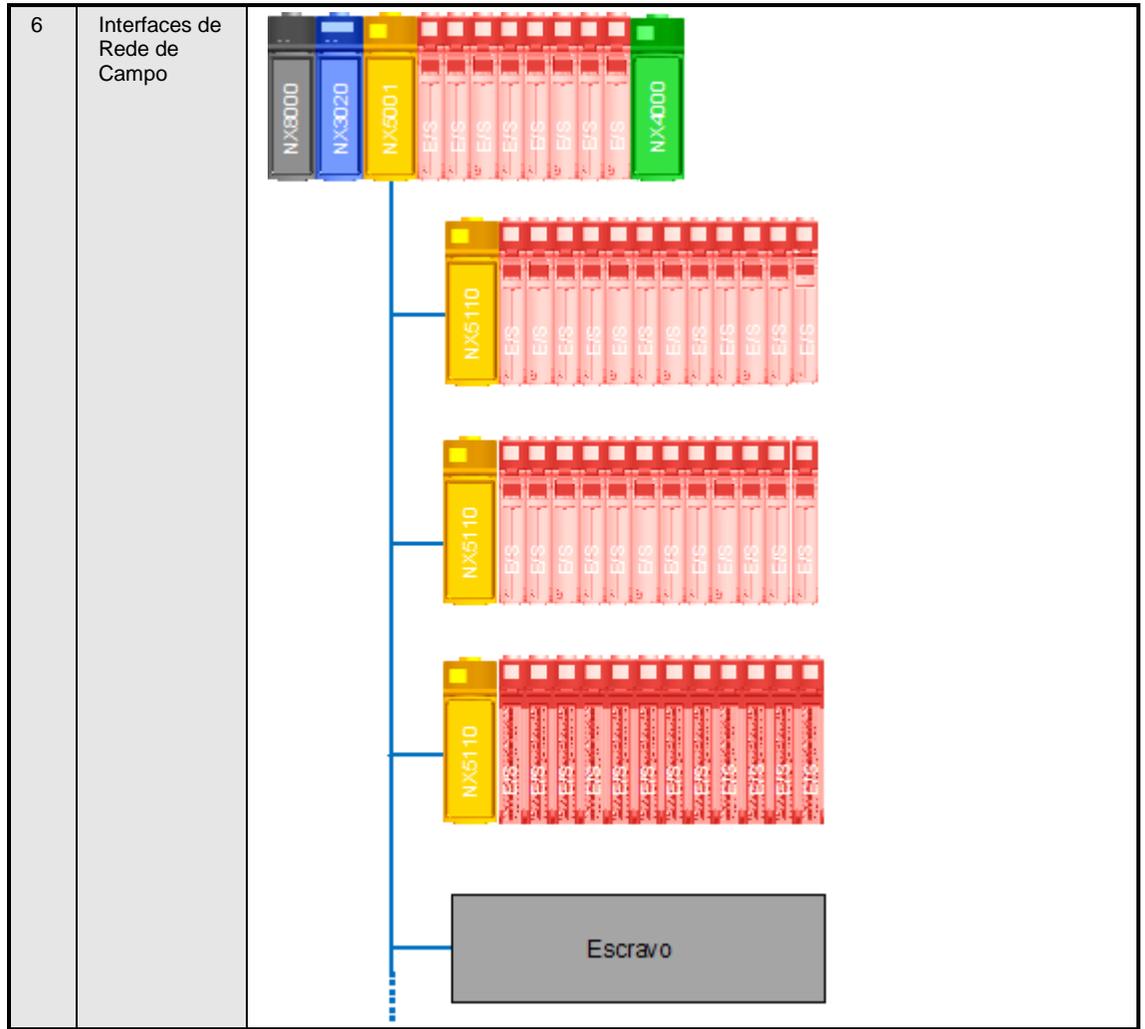
1	UCP Compacta	Esta arquitetura é baseada no uso de dois módulos de expansão de barramento por bastidor. Contando com dois módulos de expansão de barramento, o sistema apresenta uma disponibilidade elevada, pois suporta falha nos cabos de expansão de barramento ou no próprio módulo de expansão de barramento. Esta arquitetura se destina a sistemas onde a manutenção é crítica e o sistema precisa estar disponível por longos períodos de tempos. Nesta arquitetura, os bastidores devem ser montados com os módulos de expansão de barramento localizados lado a lado nas últimas posições. Observe que há portas de módulos de expansão de barramento não utilizadas, as quais devem ser deixadas desconectadas.
2	UCP Simples	Esta arquitetura é baseada na utilização de cabeças de rede de campo MODBUS para acessar redes de distribuição de E/S remotas.
3	UCP Simples com Expansão de Barramento	Esta arquitetura é baseada em um único bastidor chamado bastidor local. Este bastidor é composto por uma UCP, um módulo de fonte de alimentação e pelos módulos de E/S exigidos para a aplicação. A ordem dos módulos deve seguir as regras de configuração apresentadas na ferramenta de configuração. Esta arquitetura deve ser usada em pequenas aplicações como automação de máquinas. Ela pode ser implementada também com módulos Nexto Jet.
4	UCP Simples com Expansão de Barramento e com Loopback	Esta arquitetura apresenta interfaces de rede com protocolos baseados em Ethernet. Há duas interfaces de rede para cada finalidade: uma rede de controle para comunicação entre UCPS e uma rede de supervisão para uso com SCADA e OPC. Ambos <i>half-clusters</i> devem ter duas interfaces para cada rede, para formar um sistema de redundância completo com UCP, interface de rede e redundância do meio físico.
5	UCP Simples com Redundância de Expansão de Barramento e com Loopback	Esta arquitetura explora as necessidades de aplicações compactas, contando com uma UCP com fonte de alimentação integrada (NX3003, NX3004 ou NX3005), bastidor de 8 posições e módulos de entrada e saída que permitem a redução de espaço e custo em seu projeto. São indicadas para serem utilizadas em automação de máquinas. É importante frisar que os módulos Nexto e Nexto Jet não podem ser utilizados simultaneamente no mesmo barramento.
6	Interfaces de Rede de Campo	Esta arquitetura é baseada na utilização de interfaces de rede de campo para acessar redes de distribuição de E/S remotas e outros dispositivos de terceiros.
7	Interfaces de Rede de Campo com Redundância	Esta arquitetura é baseada em um bastidor local (onde está localizada a UCP) e bastidores remotos. A comunicação entre o bastidor local e os remotos também é feita através dos módulos de expansão de barramento. A única diferença é que a porta de dados de saída no último módulo de expansão de barramento é conectada à porta de dados de entrada do módulo de expansão de barramento do bastidor local. Esta arquitetura permite ao sistema manter o acesso de E/S mesmo em caso de uma falha nos cabos que interligam os bastidores. A UCP detectará o cabo danificado, redefinirá os caminhos dos dados internos para contornar esta falha e gerará um diagnóstico ao usuário. Esta função, além de ser interessante para uma manutenção rápida do sistema energizado, aumenta a sua disponibilidade geral. Quando esse tipo de arquitetura é utilizado, é importante lembrar que em caso de uso de módulos Nexto, somente este tipo de módulo pode ser utilizado no bastidor local e/ou na expansão. Isso vale no caso de uso de módulos Nexto Jet. Esta

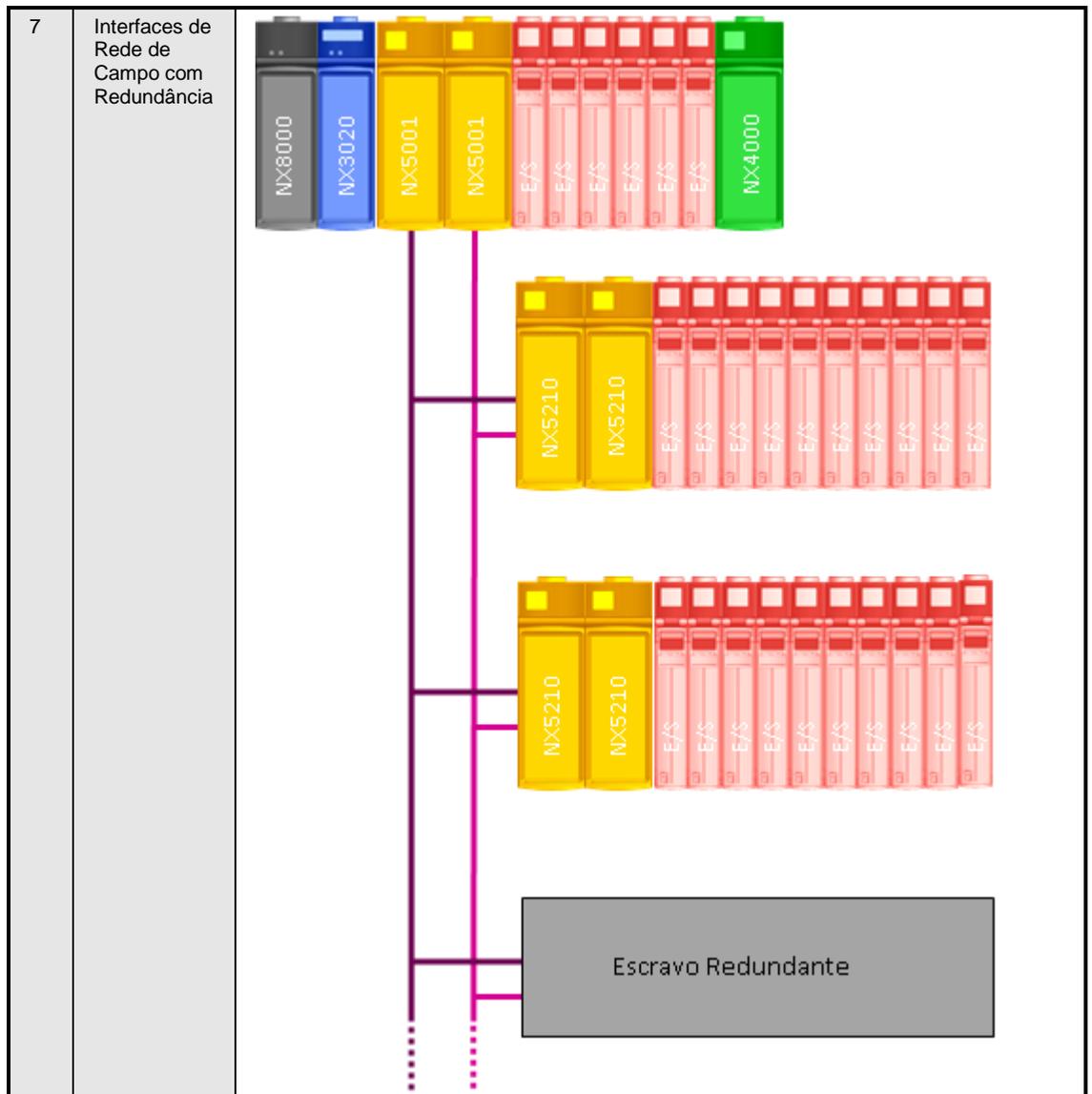
		arquitetura destina-se a médias e grandes aplicações onde o número de pontos de E/S é alto e há necessidade de maior disponibilidade.
8	<b>Cabeça de Rede de Campo MODBUS</b>	Um CP redundante é composto no mínimo de dois <i>half-clusters</i> idênticos, onde cada <i>half-cluster</i> é constituído dos seguintes módulos: bastidor onde os módulos são inseridos, que podem ser NX9000, NX9001, NX9002 e NX9003; fonte de alimentação NX8000; UCP NX3030 e o módulo NX4010.
9	<b>Redundância de UCP</b>	Esta arquitetura é baseada em um bastidor local (onde está localizada a UCP) e bastidores remotos. A comunicação entre o bastidor local e os remotos é feita através do módulo de expansão de barramento. Cada bastidor remoto necessita de seu próprio módulo de fonte de alimentação e de um módulo de expansão de barramento. A distância entre cada módulo de expansão de barramento pode ser de 100 metros usando-se os cabos blindados Ethernet padrão CAT5. Os módulos de expansão de barramento apresentam duas portas RJ45, sendo uma delas para os dados de entrada e a outra para os de saída. Neste exemplo de aplicação, o módulo de expansão do bastidor local está conectado usando-se apenas um cabo e deixando a porta dos dados de entrada aberta. No último bastidor remoto, a porta dos dados de saída está aberta. Os bastidores remotos entre eles apresentam ambas as portas conectadas: uma porta conectada ao bastidor anterior e a outra ao bastidor seguinte. Cada módulo de expansão de barramento contém uma chave para selecionar o endereço do bastidor. Cada bastidor deve ter um endereço único. Quando esse tipo de arquitetura é utilizado, é importante lembrar que em caso de uso de módulos Nexto, somente este tipo de módulo pode ser utilizado no bastidor local e/ou na expansão. Isso vale no caso de uso de módulos Nexto Jet. Esta arquitetura destina-se a médias e grandes aplicações onde há um alto número de pontos de E/S.
10	<b>Configuração Mínima de um CP Redundante (Sem utilização do Painel PX2612)</b>	Para aplicações críticas, a Série Nexto oferece redundância de UCPs. O único modelo de UCP com esta característica é a UCP NX3030. Estas UCPs podem estar posicionadas em diferentes bastidores (chamados de <i>half-clusters</i> ). Nesta arquitetura, o sistema terá um controlador executando a tarefa de controle (controlador principal) e outro permanecendo em espera com o status do sistema atual para que possa realizar um <i>switchover</i> (evento onde a UCP em espera torna-se ativa) no caso de uma falha no controlador principal. Isto significa que processos críticos deixam de ser afetados por eventuais falhas de hardware no sistema de controle. Os resultados são aumento na produtividade, minimização do tempo de inatividade e menor tempo de manutenção. A comunicação entre os controladores é feita no final de cada ciclo da UCP através de dois links de redundância de alta velocidade. No centro de um sistema redundante de dois <i>half-clusters</i> há um par de UCPs redundantes. Os links de redundância – entre duas UCPs – apresentam dois canais, de forma que a ocorrência de falhas em um canal não afeta o desempenho do sistema. A UCP ativa executa a aplicação de usuário e controla as E/S remotas. A UCP em espera coloca-se como um segundo recurso, pronto para assumir o controle, caso seja necessário. A UCP em espera é conectada à UCP ativa através de um link de alta velocidade presente no Módulo de Link de Redundância. Este módulo deve estar localizado ao lado direito da UCP, em posições do bastidor que suportem tal funcionalidade. Em caso de uma falha inesperada que ocorra com a UCP ativa, o sistema em espera alterna-se automaticamente, alterando a execução do programa da aplicação e o controle das E/S para a UCP em espera com o contexto de dados atualizado. Uma vez alterada, a UCP em espera torna-se a UCP ativa. A configuração dos dois sistemas deve ser idêntica, assim como os módulos das UCPs devem estar localizados nas mesmas posições em cada <i>half-cluster</i> . Após a energização, uma das UCPs opera como ativa e a outra permanece em estado de espera. A UCP ativa atualizará o status do sistema da UCP em espera ao final de cada ciclo. Assim a UCP em espera estará sempre atualizada de acordo com o último status das E/S e os resultados da execução do programa na UCP ativa. Esta aplicação é de fácil configuração e dispensa programação especial ou parametrização.
11	<b>Redundância de UCP e Interfaces de Rede</b>	Esta arquitetura é baseada na utilização de interfaces de rede de campo, com a diferença de que se usam duas interfaces de rede de campo para acessar a mesma rede. Uma vez que apresenta duas interfaces, a rede torna-se redundante e fornece um sistema com maior disponibilidade. Ressalta-se que os módulos que compõem a solução Nexto Jet não suportam qualquer tipo de redundância, portanto seu uso não é permitido em arquiteturas como as descritas nessa seção.

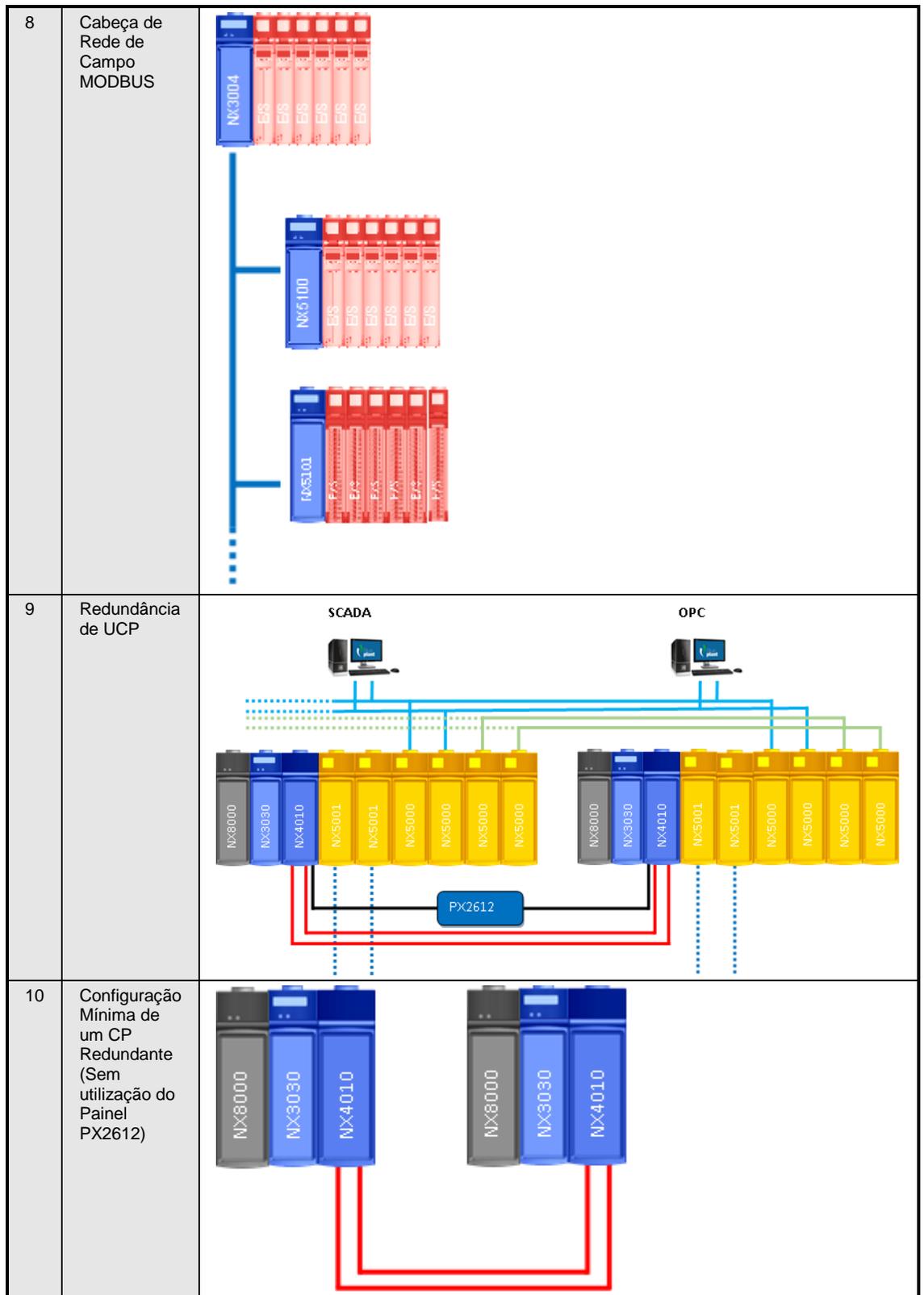
 NOTA: para auxiliar no desenvolvimento do estudo dirigido estão ilustradas, na sequência, as arquiteturas típicas da Série Nexto.

I	Nome	Ilustração da arquitetura
1	UCP Compacta	 <p>Nexto:  Nexto Jet: </p>
2	UCP Simples	 <p>Nexto:  Nexto Jet: </p>
3	UCP Simples com Expansão de Barramento	 <p>Nexto:  Nexto Jet: </p>











Resolução do Estudo Dirigido 1-2: correlação das colunas.

1	UCP Compacta	5
2	UCP Simples	8
3	UCP Simples com Expansão de Barramento	2
4	UCP Simples com Expansão de Barramento e com <i>Loopback</i>	11
5	UCP Simples com Redundância de Expansão de Barramento e com <i>Loopback</i>	1
6	Interfaces de Rede de Campo	6
7	Interfaces de Rede de Campo com Redundância	4
8	Cabeça de Rede de Campo MODBUS	10
9	Redundância de UCP	3
10	Configuração Mínima de um CP Redundante (Sem utilização do Painel PX2612)	9
11	Redundância de UCP e Interfaces de Rede	7

## Um pouco de teoria... Histórico e características dos CPs



No contexto da automação de "complexidade média", os elementos fundamentais para sua realização física são os Controladores Programáveis. No fim da década de 1960, os circuitos integrados permitiram o desenvolvimento de sistemas computacionais que foram logo utilizados para controle on-line de processos industriais. Em 1969, surgiram os primeiros controladores baseados numa especificação da General Motors, resumida a seguir: facilidade de programação; facilidade de manutenção com conceito *plug-in*; alta confiabilidade; dimensões menores que as dos painéis de relés, para redução de custo; envio de dados para processamento centralizado; preço competitivo; sinais de E/S de 115 Vca; expansão em módulos e mínimo de 400 palavras na memória. Na década de 1970, os controladores passaram a ter microprocessadores e a serem denominados Controladores Programáveis (CPs). Na década de 1980, houve aperfeiçoamento das funções de comunicação dos CPs, sendo então utilizados em rede. Algumas características dos CPs:

- Linguagens de programação de alto nível, caracterizando um sistema bastante amigável com relação ao operador. Depois de concluído e depurado, o programa pode ser transferido para outros CPs, garantindo confiabilidade na sua utilização.
- Simplificação nos quadros e painéis elétricos. Toda a fiação do comando fica resumida a um conjunto de entradas e saídas. Como consequência, qualquer alteração necessária torna-se mais rápida e barata.
- Confiabilidade operacional. Uma vez que as alterações podem ser realizadas através do programa necessitando de muito pouca ou de nenhuma alteração da fiação elétrica, a possibilidade de haver erro é minimizada, garantindo sucesso nos desenvolvimentos ou melhorias a serem implantadas.
- Os controladores podem realizar uma grande variedade de tarefas de controle através de funções matemáticas, controle da qualidade e informações para relatórios. Sistemas de gerenciamento de produção são bastante beneficiados com a utilização dos controladores.
- Comunicação em rede. Através de interfaces de operação, controladores e computadores em rede permitem coleta de dados e um enorme intercâmbio de troca de dados em relação aos níveis da pirâmide de automação.

Fonte: Engenharia de automação industrial. Cícero Couto de Moraes, Plínio de Lauro Castrucci.

# Diagnósticos e manutenção da Série Nexto



Cada módulo contém seus próprios diagnósticos: as UCPs, Interfaces de Rede de Campo, Fontes de Alimentação e módulos de E/S apresentam vários diagnósticos disponíveis. Cada módulo tem um visor multifuncional que informa o status do mesmo.

Além disto, cada módulo Nexto que conta com um botão na sua parte superior pode fornecer informações de diagnósticos diferentes para a equipe de manutenção. Estes diagnósticos podem ser monitorados no campo através de visores ou através da ferramenta de configuração. Alguns exemplos são:

- Módulo localizado em uma posição incorreta no bastidor;
- Ausência de fonte de alimentação;
- Curto-circuito nas saídas;
- Não há configuração para um módulo que necessite ser configurado para operação normal;
- Visualização de tag e descrição de E/S no CP para módulos com OTD;
- Visualização de endereço IP;

A documentação de cada módulo indica todos os diagnósticos presentes.

Nesta seção será abordada a funcionalidade OTD que está presente nos módulos da Série Nexto, exceto os que fazem parte da solução Nexto Jet. Tal funcionalidade permite o acesso de tag, diagnósticos e descrição de todos os módulos e pontos de E/S através das teclas de diagnósticos, visor do módulo em questão e visor da UCP.

A figura a seguir detalha os elementos que constituem o painel frontal da UCP Nexto, por exemplo, o qual concentra – além das interfaces de comunicação e de cartão de memória – o visor gráfico e a tecla de diagnóstico. Estes últimos fazem parte de uma importante ferramenta para monitoração completa do sistema, incluindo diagnósticos gerais e suporte à manutenção.

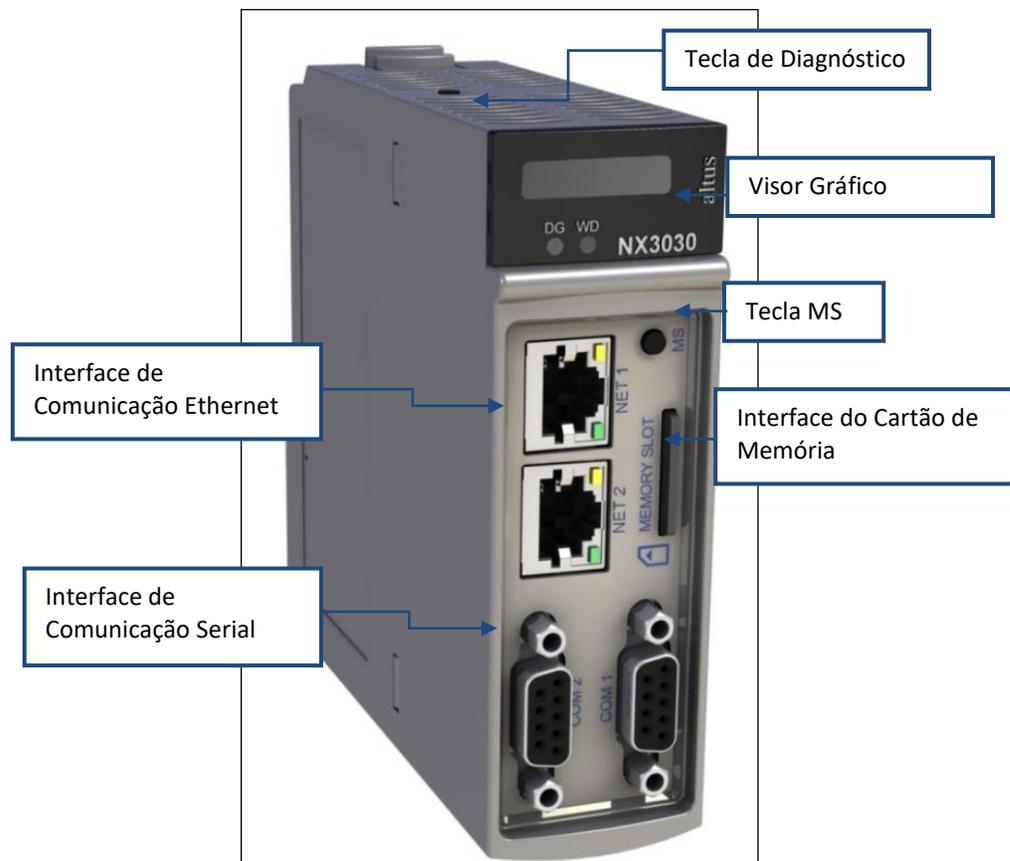


Figura 1-3. Painel Frontal UCP Nexto

### ***One Touch Diag (OTD)***

Independentemente do que está sendo mostrado no visor da UCP, após um pressionamento curto no botão de diagnóstico de um determinado módulo será mostrado no visor da UCP a tag e os diagnósticos ativos do respectivo módulo. Estes dados serão mostrados no visor da UCP na ordem indicada na figura a seguir.

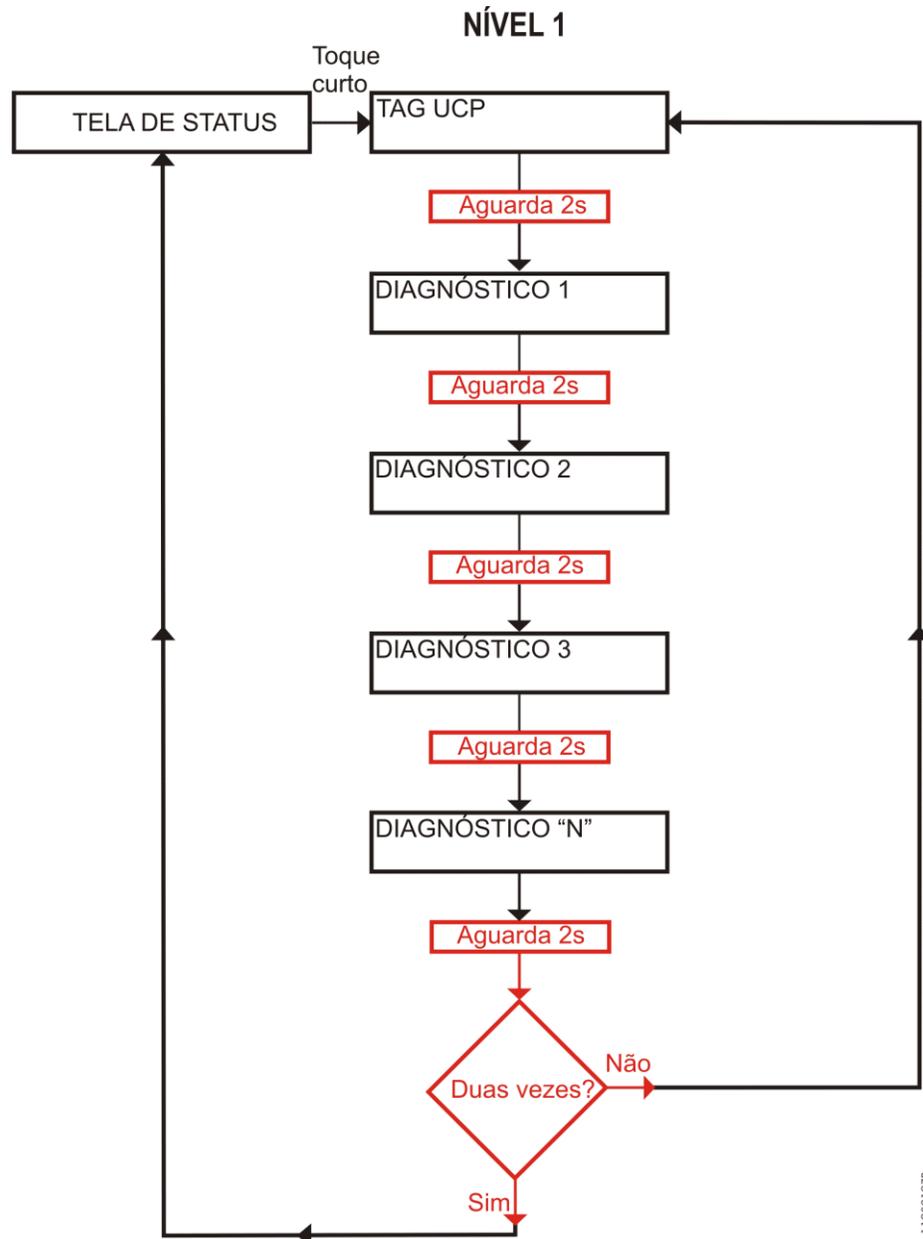


Figura 1-4. Visualização dos diagnósticos da UCP

É possível identificar que um determinado módulo está em modo diagnóstico quando os dois segmentos do visor do módulo estiverem piscando conforme mostrado na figura a seguir. A quantidade de segmentos depende da quantidade de pontos de cada módulo de E/S.

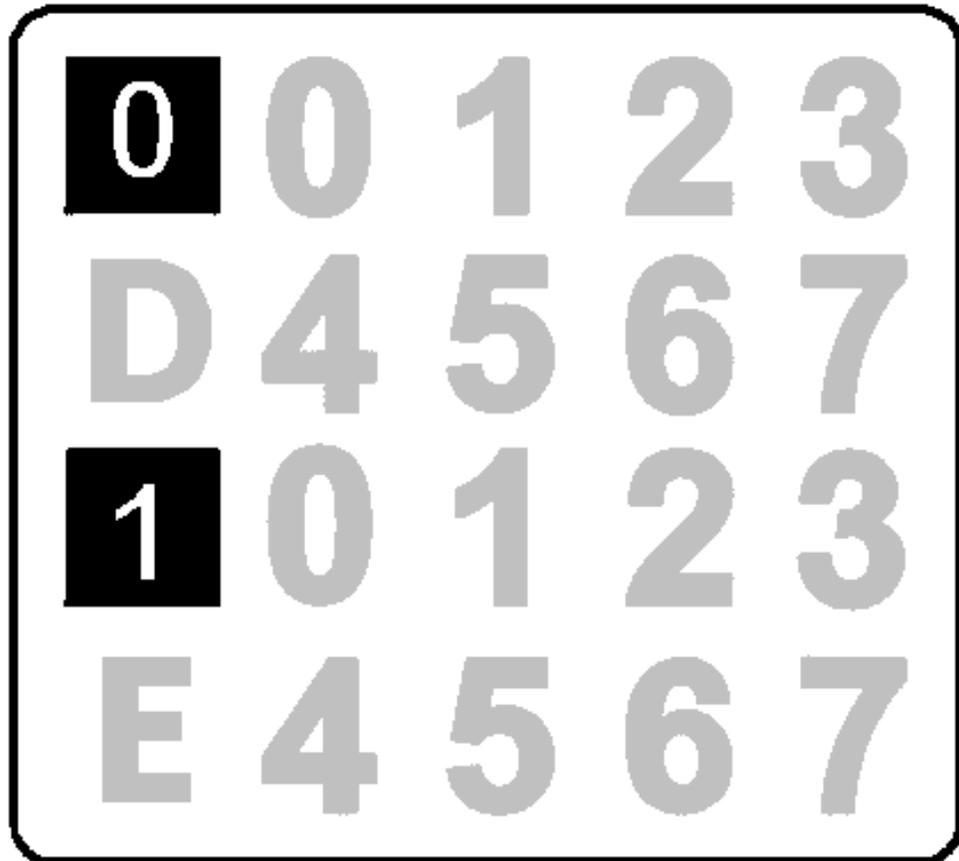


Figura 1-5. Visor do Módulo

### Acessando pontos de E/S

Após entrar no modo diagnóstico, o próximo pressionamento curto do botão de diagnósticos do módulo em questão selecionará o primeiro ponto de E/S, neste momento o visor apagará a indicação de diagnósticos individuais ativos e indicará qual é o ponto de E/S que está selecionado. Para a seleção do próximo ponto de E/S basta um novo pressionamento curto no botão de diagnósticos do respectivo módulo. Quando o último ponto de E/S estiver selecionado um novo pressionamento curto no botão de diagnóstico irá finalizar o modo de diagnóstico.

Ao acessar um ponto de E/S, o visor da UCP indicará a TAG do respectivo ponto de E/S e todos os diagnósticos ativos do mesmo da mesma forma como indicada no fluxograma mostrado acima.

Para módulos que não possuem pontos de E/S, um novo pressionamento curto no botão de diagnósticos, após a entrada no modo diagnóstico, finalizará a operação.

**ATENÇÃO:**

Os módulos Nexto Jet não possuem os diagnósticos por canal com acesso através do botão.

Essa funcionalidade é uma característica somente dos módulos Nexto.

### Acessando a descrição do módulo e dos pontos de E/S

Além da tag, os módulos e pontos de E/S podem possuir uma descrição. Indica-se utilizar a descrição quando existe a necessidade de adicionar uma informação extra, além da tag para um determinado módulo ou ponto de E/S como, por exemplo: "Módulo de leitura de temperaturas" para a descrição de um módulo ou "Acionamento bomba principal" para a descrição de um

determinado ponto de saída. O limite de caracteres para a tag, descrição da tag ou nome do módulo é de 255 caracteres.

Para alterar o nome e a descrição de cada módulo inserido na aplicação, clique com o botão direito sobre o módulo, no item “Propriedades”, na guia “Comum”, altere o nome ou a descrição sendo ambos limitados a 255 caracteres. Para alterar a tag, acesse a aba “Bus I/O Mapping” de cada módulo, e dê um duplo clique na coluna da tabela destinada à inserção da tag de cada ponto. Para alterar a descrição da tag, faça o mesmo na coluna destinada para descrição da tag, e insira a informação desejada.

Para acessar a descrição nos módulos Nexto, basta um pressionamento longo no botão de diagnóstico. Quando o módulo estiver indicando a tag e diagnósticos ativos do módulo, aparecerá no visor da UCP a descrição do módulo da mesma forma, quando estiver selecionado um ponto de E/S aparecerá no visor da UCP a descrição do determinado ponto de E/S.

**ATENÇÃO:**

É recomendável que a tag e o nome do módulo possuam apenas os caracteres alfanuméricos (maiúsculos ou minúsculos e o texto não iniciar com números). Para a descrição o recomendável é os caracteres alfanuméricos ( maiúsculos ou minúsculos), o espaço em branco e o caractere ponto “. A utilização de qualquer outro caractere diferente dos acima citados não é recomendada.

**ATENÇÃO:**

Quando utilizar a função ETD – Electronic Tag on Display, Diagnostic Explorer ou Web Server para visualizar a tag dos pontos de E/S, o nome da tag será truncado nos primeiros 24 caracteres após a string “Application.” do nome da tag. Ex. Para a tag "Application.UserPrg.MyTest.ON", somente a string “UserPrg.MyTest.ON” será visualizada no visor gráfico.

**ATENÇÃO:**

Quando utilizar a função ETD – Electronic Tag on Display, Diagnostic Explorer ou Web Server para visualizar a descrição da tag dos pontos de E/S, a descrição da tag será truncada nos primeiros 48 caracteres da descrição dada para a tag.

**ATENÇÃO:**

Quando utilizar a função ETD – Electronic Tag on Display, Diagnostic Explorer ou Web Server para visualizar o nome do módulo, o nome do módulo será truncado nos primeiros 24 caracteres da descrição dada para o nome do módulo.

Os módulos que fazem parte da solução Nexto Jet não possuem a funcionalidade ETD – Electronic Tag on Display. Esta é uma característica exclusiva dos módulos Nexto.

### Visor Gráfico

No visor gráfico disponível nas UCPs da Série Nexto podem ser reconhecidas as possíveis condições de erro, presença de componentes e de diagnósticos ativos. Além disso, é através do visor gráfico que todos os diagnósticos, inclusive dos módulos de E/S, são exibidos ao usuário. Na figura abaixo, é possível visualizar todos os caracteres disponíveis no visor gráfico da UCP Nexto e, a seguir, os seus respectivos significados.

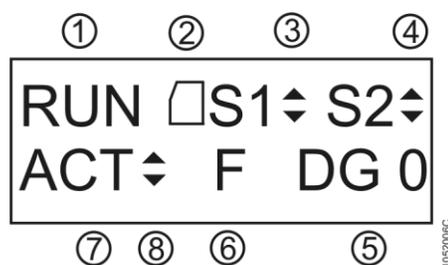


Figura 1-6. Tela de Status da UCP

Legenda:

1. Indicação do estado de operação da UCP. Caso a aplicação da UCP esteja em execução, o estado será RUN. Caso a aplicação da UCP esteja parada, o estado será STOP e quando estiver parada em marcas de depuração da aplicação o estado será BRKP.
2. Indicação da presença do Cartão de Memória.
3. Indicação de Tráfego na COM 1. A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados.
4. Indicação de Tráfego na COM 2. A seta para cima (▲) indica transmissão de dados e a seta para baixo (▼) indica recepção de dados.
5. Indicação da quantidade de diagnósticos ativos na UCP. Caso o número mostrado seja diferente de 0 (zero), existem diagnósticos ativos na UCP.
6. Indicação de variáveis forçadas na UCP. Caso o caractere F esteja exibido no visor gráfico, alguma variável está sendo forçada pelo usuário, seja ela variável simbólica ou variável de representação direta mapeada em uma variável simbólica.
7. Identificação do estado da redundância na UCP (mensagem válida somente na NX3030 em modo redundante). Caso a UCP seja o CP Ativo, a informação ACT será apresentada. Os demais estados possíveis são NCF (Não Configurado), STR (Inicializando), INA (Inativo) e SBY (Reserva).
8. Indicação de que está sendo executada a sincronização de projeto. A seta para cima (N) indica transmissão de dados do projeto e a seta para baixo (H) indica recepção de dados de um projeto.

Além dos caracteres descritos acima, as UCPs Nexto poderão apresentar algumas mensagens no visor gráfico, correspondentes a algum processo que está sendo executado no momento. Consulte o Manual de Utilização UCPs Série Nexto NX30xx para detalhamento das mensagens e suas respectivas descrições.



### Pesquisa na Documentação: visor gráfico

Consulte a tabela de mensagens do visor gráfico e suas respectivas descrições na documentação do produto.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

Tabela 1-2. Mensagens do Visor Gráfico

Mensagem	Descrição
...	



## Pesquisa na Documentação: tempos de *One Touch*

Consulte, na documentação do produto, a diferença entre os tempos de “sem pressionamento”, “pressionamento curto”, “pressionamento longo” e “tecla presa”.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

Tabela 1-3. Tempo de *One Touch*

Tipo Toque	Tempo Mínimo	Tempo Máximo	Condição para Indicação
...			

As mensagens exibidas no visor gráfico das UCPs Nexto, correspondentes aos diagnósticos podem ser consultadas no Manual de Utilização UCPs Série Nexto NX30xx – MU214100.

## Diagnósticos via LED

As UCPs da Série Nexto possuem um LED para indicação de diagnóstico (LED DG) e um LED para indicar ocorrência de cão-de-guarda (LED WD). Veja o significado de cada estado e suas respectivas descrições nas tabelas abaixo.



## Pesquisa na Documentação: LEDs de diagnóstico e cão-de-guarda

Consulte, na documentação do produto, a descrição dos Estados do LED de Diagnóstico e Cão-de-guarda.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

### DG (Diagnóstico)

Tabela 1-4. Descrição dos Estados do LED de Diagnóstico

Verde	Vermelho	Descrição	Causas	Prioridade
...				

### WD (Cão-de-guarda)

Tabela 1-5. Descrição dos Estados do LED de Cão-de-guarda

LED Vermelho	Descrição	Causas	Prioridade
...			



### NOTAS:

- ✓ Para limpar qualquer indicação de cão-de-guarda, o módulo deve ser desconectado da fonte de alimentação.
- ✓ Cão-de-guarda de software: no mínimo uma tarefa da aplicação de usuário leva mais que dois segundos para ser realizada.

### LEDs Conector RJ45

Os dois LEDs presentes nos conectores RJ45 (no caso da NX3010 um conector somente), identificados por NET 1 e NET 2, auxiliam o usuário na detecção de problemas na rede física

instalada, indicando a velocidade do LINK de rede e a existência de tráfego de comunicação com a interface.



### Pesquisa na Documentação: LEDs dos conectores RJ45

Consulte, na documentação do produto, o significado dos LEDs dos conectores RJ45.

Fonte: Manual de Utilização UCPs Série Nexto NX30xx – MU214100

Tabela 1-6. Significado dos LEDs Ethernet

Amarelo	Verde	Significado
...		

### Diagnósticos via WEB

Além das características apresentadas anteriormente, a Série Nexto possibilita ao usuário uma condição inovadora de acesso aos diagnósticos e estados de operação do sistema, através de uma página WEB.

A utilização, além de dinâmica, é bastante intuitiva e facilita as operações do usuário. Entre outras palavras, pode substituir o uso de um sistema supervisor quando o uso for restrito a verificação de status do sistema.

Para acessar a página WEB da UCP desejada, basta utilizar um navegador padrão (consulte o Manual de Utilização UCPs Série Nexto NX30xx para conferir as versões compatíveis) e digitar, na barra de endereço, o endereço IP correspondente a UCP (Ex.: <http://192.168.1.1>). Inicialmente, serão apresentadas as [Informações da UCP](#) na aba específica, conforme mostrado na próxima figura.

altus English | Español | Português

NEXTO

Informações da UCP Informações do Sistema Gerenciamento da UCP

**Informações da UCP**

Informações de 31/10/2014, 18:10:23. Para atualizar clique no ícone.

Modelo	NX3030
Tag	NX3030
Descrição	
Bastidores Configurados	1
Bastidor	0
Posição	2
Versão de Firmware	1.5.0.2
Versão do Bootloader	1.0.0.5
Versão do Processador Auxiliar	0.0.0.4
Estado da Aplicação	Run (Executando)
Diagnósticos Ativos	0
Valores Forçados	Não

Figura 1-7. Tela Inicial

A segunda aba apresenta as Informações do Sistema via visualização do bastidor (rack) ou da lista dos módulos presentes e a seleção se dá via opção no lado direito da tela. Quando não houver nenhuma aplicação na UCP, será exibida nesta página uma configuração com o maior Bastidor disponível e uma fonte de alimentação padrão, juntamente com a UCP conectada. Quando a visualização pelo Bastidor é utilizada, os módulos que têm diagnóstico ficam piscando e assumem a cor vermelha, conforme mostra a figura a seguir. Caso contrário será exibida uma lista dos módulos presentes no sistema, Tags correspondentes e número de diagnósticos ativos.

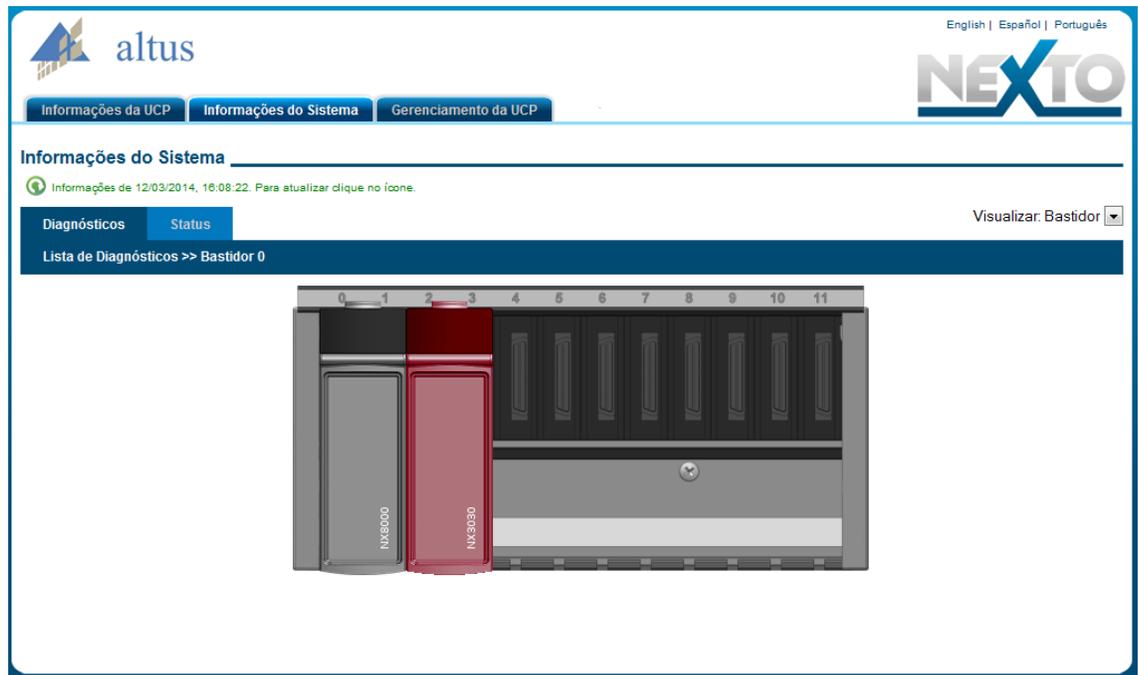


Figura 1-8. Informações do Sistema

Ao clicar no módulo com diagnóstico, no mesmo instante são mostrados os diagnósticos ativos do módulo, conforme mostra a figura.

**ATENÇÃO:**

Quando uma UCP for reiniciada e a aplicação entrar em exceção na partida do sistema, os diagnósticos não estarão válidos. É necessário corrigir o problema que gera a exceção da aplicação para que os diagnósticos sejam atualizados.

altus English | Español | Português

NEXTO

Informações da UCP | Informações do Sistema | Gerenciamento da UCP

Informações do Sistema

Informações de 12/03/2014, 16:11:09. Para atualizar clique no ícone.

Diagnósticos | Status

Lista de Diagnósticos >> Bastidor 0 >> NX3030 @ Posição 2

ID	Diagnóstico
00	Erro tecla OTD

Figura 1-9. Diagnósticos do Sistema

Caso seja selecionada a aba Status, será exibido na tela o estado de todos os diagnósticos detalhados, conforme mostra a figura a seguir.

altus English | Español | Português

NEXTO

Informações da UCP | Informações do Sistema | Gerenciamento da UCP

Informações do Sistema

Informações de 12/03/2014, 16:24:56. Para atualizar clique no ícone.

Diagnósticos | Status

Lista de Status >> Bastidor 0 >> NX3030 @ Posição 2

ID	Status
00	CONFIG - Não há erros de configuração no barramento
01	CONFIG - Não há módulos declarados ausentes no barramento
02	CONFIG - Não há módulos trocados no barramento
03	CONFIG - Não há módulos não declarados presentes no barramento
04	CONFIG - Não há módulos com diagnóstico(s) ativo(s)
05	CONFIG - Não há módulos em estado não funcional no barramento
06	CONFIG - Não há módulos com erro de parametrização no barramento
07	UCP - Hardware de acesso ao barramento operacional
08	UCP - Hardware interno operacional
09	UCP - Software interno em condição de execução normal
10	UCP - Interface do cartão de memória operacional
11	CONFIG - Configuração COM 1 sem erros
12	CONFIG - Configuração COM 2 sem erros
13	CONFIG - Configuração NET 1 sem erros

Figura 1-10. Estado do Sistema

Além disso, o usuário pode optar por três opções de idioma: português, inglês e espanhol. Basta alterar o menu superior direito para o idioma desejado.



Visualizar os LEDs de *status* presentes no painel (DG e WD) conforme a tabela de descrição de estado dos LEDs de diagnóstico. Forçar qualquer operando na programação através da tecla F7 e verificar a alteração na cadência do LED DG da UCP.

- Como executar o diagnóstico *one touch*?

Acione a tecla superior da UCP Nexto e verifique na tela de display da UCP as análises de diagnóstico.

- Como executar o diagnóstico via WEB?

Para acessar a página WEB da UCP desejada, basta utilizar um navegador padrão (Internet Explorer 7 ou superior, Mozilla Firefox 3.0 ou superior e Google Chrome 8 ou superior) e digite, na barra de endereço, o endereço IP correspondente a UCP (Ex.: <http://192.168.15.168>). Inicialmente, serão apresentadas as informações da UCP no navegador.

## Diagnósticos via Variáveis

As UCPs da Série Nexto possuem variáveis para indicação de diagnóstico. Existem estruturas de dados com os diagnósticos de todos os módulos declarados no barramento, mapeadas sobre variáveis de representação direta %Q, e definidas simbolicamente através da diretiva AT, na GVL *System\_Diagnostics* criada automaticamente pelo MasterTool IEC XE. Consulte o Manual de Utilização UCPs Série Nexto NX30xx para detalhamento do significado de cada bit dos diagnósticos da UCP. A tabela a seguir sumariza a divisão dos bytes/words de diagnóstico.

Tabela 1-7. Diagnósticos via Variáveis

Byte	Descrição
0 a 3	Diagnósticos resumidos da UCP.
4 a 560	Diagnósticos detalhados da UCP (NX3003)
4 a 558	Diagnósticos detalhados da UCP (NX3004, NX3005 e NX3010)
4 a 693	Diagnósticos detalhados da UCP (NX3020 e NX3030).

## Diagnósticos via Função

A função *GetTaskInfo* localizada na biblioteca *NextoStandard* permite a visualização de alguns parâmetros que não podem ser acessados de outra maneira. Essa função retorna informações sobre uma tarefa de uma determinada aplicação. Consulte o Manual de Utilização UCPs Série Nexto NX30xx para detalhamento dos parâmetros que devem ser repassados à função para que ela retorne as informações da aplicação, incluindo um exemplo de como a mesma pode ser chamada na aplicação.

## *Diagnostic Explorer*

O *Diagnostic Explorer* é a inclusão dos diagnósticos via WEB dentro do ambiente de programação MasterTool IEC XE, a fim de que o acesso seja mais rápido e objetivo. O acesso à característica ocorre de duas maneiras:

1. Acessando a opção *Diagnostic Explorer* na árvore de dispositivos, localizada na esquerda da tela do MasterTool IEC XE, colocando o IP correto no campo indicado (o usuário deverá estar conectado à UCP).
2. Clicando com o botão direito do mouse sobre o módulo, e selecionando "Diagnósticos", o *Diagnostic Explorer* será aberto, direcionando para a página de status do respectivo módulo.

Para maiores informações sobre Diagnostic Explorer, consulte o Manual de Utilização do MasterTool IEC XE - MU299048.

### Log de Sistema

O Log de Sistema é um recurso disponível no programador MasterTool IEC XE. É uma importante ferramenta de registro de processo, pois através dele é possível localizar eventos na UCP que podem indicar condições de erro, presença de componentes ou de diagnósticos ativos. Tais eventos podem ser visualizados em ordem cronológica com uma resolução de segundos, com uma capacidade de armazenamento de até mil entradas de log armazenadas na memória interna do CP e não podem ser removidos.

Para maiores informações sobre os Logs de Sistema, consulte o Manual de Utilização do MasterTool IEC XE - MU299048.

### Não Carregar a Aplicação na Inicialização

Caso seja necessário, o usuário pode optar por não carregar uma aplicação já existente na UCP durante sua inicialização. Para isto, basta energizar a UCP com a tecla de diagnóstico pressionada e mantê-la assim até ser exibida a mensagem na tela “APL. NÃO CARREGADA”. Após a inicialização da UCP e a exibição no visor da mensagem avisando que a aplicação não será carregada a UCP estará também em Modo Stop. Caso seja feito um login, o software MasterTool IEC XE indicará que não existe nenhuma aplicação na UCP. Para voltar a carregar a aplicação, a UCP deve ser reinicializada, ou um novo download da aplicação deve ser feito.

### Falha na Alimentação

A fonte de alimentação da Série Nexto (NX8000) possui um sistema de detecção de falha, de acordo com os níveis definidos em suas características técnicas. Existem duas formas de diagnosticar a falha:

1. Caso a fonte NX8000 seja energizada com tensão inferior ao limite mínimo exigido, será gerado um diagnóstico de falha de alimentação, o qual será reconhecido pela UCP e a mensagem “FALHA NA ALIMENTAÇÃO” será exibida em seu visor. Quando a alimentação estiver dentro dos limites estabelecidos, a UCP irá reconhecer e, automaticamente, será reiniciada com a aplicação do usuário. O diagnóstico ainda estará ativo para mostrar ao usuário que na última inicialização ocorreu falha na alimentação.
2. Caso a fonte NX8000 tenha uma queda de tensão para um valor inferior ao limite mínimo exigido e retornar para um valor acima em até 10 ms, a falha de alimentação não será reconhecida pela UCP e não será gerado o diagnóstico, pois o sistema se mantém intacto durante esse tempo. Porém, se a queda de tensão for superior a 10 ms, a mensagem “FALHA NA ALIMENTAÇÃO” será exibida no visor da UCP e o diagnóstico será ativado.

O usuário pode alterar o valor da variável atribuída à falha na alimentação para FALSE durante a execução da aplicação, facilitando a verificação e tratamento deste diagnóstico. O diagnóstico de FALHA NA ALIMENTAÇÃO já está previamente mapeado em uma região específica de memória, definida como Diagnósticos Detalhados da UCP. Dessa forma, basta utilizá-lo como uma variável global. O nome da variável está descrita na lista de diagnósticos detalhados da UCP.

### Manutenção da Série Nexto

Os diagnósticos abordados anteriormente estão relacionados principalmente ao monitoramento geral do funcionamento da Série Nexto e fornecem um suporte consistente à manutenção corretiva do sistema. Além disso, outros aspectos podem ser avaliados no contexto da manutenção, conforme indicado na sequência.

Se, ao energizar a UCP, a mesma não entrar em funcionamento, os seguintes itens devem ser verificados:

- A temperatura ambiente está dentro da faixa suportada pelos equipamentos?
- A fonte de alimentação do bastidor está sendo alimentada com a tensão correta?
- A fonte de alimentação é o módulo, inserido no bastidor, mais à esquerda (bastidor sendo visto de frente), seguido pela UCP da Série Nexto?
- Os equipamentos da rede, como hubs, switches ou roteadores, estão alimentados, interligados, configurados e funcionando corretamente?
- O cabo de rede Ethernet está devidamente conectado à porta NET 1 ou NET 2 da UCP Nexto e ao equipamento de rede?
- A UCP da Série Nexto está ligada, em modo de execução (Run) e sem diagnósticos relacionados ao hardware?

Se a UCP Nexto indica o estado Execução (Run), e, mesmo assim, não responde às comunicações solicitadas (sejam elas pelo MasterTool IEC XE ou através de protocolos), os seguintes itens devem ser verificados:

- A configuração dos parâmetros Ethernet da UCP está correta?
- O respectivo protocolo de comunicação está configurado corretamente na UCP?
- As variáveis que habilitam as relações MODBUS estão devidamente habilitadas?

Se nenhum problema for identificado, consulte o Suporte a Clientes Altus.

## Solução de Problemas

A tabela a seguir mostra os sintomas de alguns problemas com suas possíveis causas de problemas e possíveis soluções. Se o problema persistir, entre em contato com o Suporte Técnico da Altus.

Tabela 1-8. Tabela de Solução de Problemas

Sintoma	Possível Causa	Solução
<b>Não liga</b>	Falta de alimentação ou alimentado incorretamente.	Verificar se a UCP está conectada corretamente no bastidor. Desenergizar e retirar todos os módulos do barramento, menos a fonte de alimentação e a UCP. Energizar o barramento e verificar o funcionamento da fonte de alimentação, tanto a externa como a fonte conectada ao barramento. Verificar se a tensão de alimentação chega ao borne da fonte de alimentação Nexto, e também se a polarização está correta.
<b>Não comunica</b>	Mau contato ou mal configurado.	Verificar todas as conexões dos cabos de comunicação. Verificar as configurações das interfaces serial e ethernet no software MasterTool IEC XE.
<b>Não reconhece o cartão de memória</b>	Mal conectado ou desmontado	Verificar se o cartão de memória está conectado corretamente no compartimento. Verificar se o cartão de memória foi colocado do lado correto, conforme a indicação no painel frontal da UCP. Verificar se o cartão de memória não foi desmontado através do botão MS, localizado no painel frontal, visualizando a indicação no visor gráfico da UCP.

Por outro lado, aspectos relacionados à manutenção preventiva devem ser considerados para garantir uma operação satisfatória do hardware, conforme listado a seguir:

- Deve-se verificar, a cada ano, se os cabos de interligação estão com as conexões firmes, sem depósitos de poeira, principalmente os dispositivos de proteção.

- Em ambientes sujeitos a contaminação excessiva, deve-se limpar periodicamente o equipamento, retirando resíduos, poeira etc.
- Os varistores utilizados para a proteção contra transientes causados por descargas atmosféricas devem ser inspecionados periodicamente, pois podem estar danificados ou destruídos caso a energia absorvida esteja acima do limite. Em muitos casos, a falha pode não ser evidente ou facilmente visível. Em aplicações críticas, é recomendável a substituição periódica dos varistores, mesmo os que não apresentarem sinais visíveis de falha.

# Instalação da Série Nexto



## Projeto Mecânico

Estão disponíveis diferentes tamanhos de módulos na Série Nexto, dependendo do seu tipo. Os módulos de E/S de 18 mm e 36 mm Nexto, por exemplo, são usados em módulos de E/S que ocupam, respectivamente, uma e duas posições do bastidor. No caso da solução Nexto Jet existe um módulo específico para o primeiro caso. Os módulos de UCP, Interfaces de Rede de Campo, Fontes de Alimentação e Módulos Especiais têm largura padronizada de 36 mm.

Como elementos complementares ao projeto do painel cita-se a base de 2 posições para montagem em painel, o bastidor de 8 Posições (Sem Troca a Quente) e os bastidores de 8, 12, 16 e 24 posições.

Para informações adicionais referentes à profundidade do módulo montado no bastidor, espaçamento entre módulos/equipamentos do painel, dimensionamento da calha e montagem horizontal/vertical consulte o Manual de Utilização Série Nexto - MU214000.

## Projeto Térmico

Os equipamentos Altus são projetados para trabalhar a uma temperatura ambiente de até 60 °C. Portanto, esta deve ser a temperatura interna máxima do painel. Os seguintes cuidados devem ser observados no projeto do painel:

- Dimensionar painéis com volume interno suficiente para uma boa circulação de ar;
- Prever ventilação forçada ou trocadores de ar com o meio externo, caso necessário, para que não haja elevação da temperatura além do limite. Em casos críticos, recomenda-se o uso de equipamentos de refrigeração, para manter o equipamento operando dentro dos níveis de temperatura de operação;
- Distribuir de forma homogênea fontes de calor dentro do painel;
- Considerar a dissipação nos cabos que conduzem correntes mais elevadas para evitar superaquecimento interno às calhas.

### ATENÇÃO:

Para obter a máxima dissipação de cada módulo da Série Nexto, consulte o documento Características Técnicas do módulo em questão.

O Manual de Utilização Série Nexto - MU214000 detalha um método para calcular a temperatura interna do painel em função da sua dissipação e potência.

## Projeto Elétrico

Os controladores programáveis são fabricados atendendo normas mundiais, que estabelecem os níveis aceitáveis de condições ambientais e de ruído normalmente encontrados em processos industriais. É fundamental também que a instalação destes produtos siga regras de projeto convenientes, estabelecidas por normas de instalação. Problemas causados por interferências eletromagnéticas (EMI), tais como falhas de comunicação, falhas de execução de programa, ruído em variáveis analógicas, e até mesmo perda de programa, podem ser causadas por um projeto elétrico ou instalação deficientes.

O projeto elétrico dos CPs Altus deve respeitar a norma IEEE 518/1977, "*Guide for Installation of Electrical Equipment to Minimize Electrical Noise Inputs to Controllers External Sources*". Para

informações detalhadas referentes à alimentação do painel, distribuição dos cabos no painel, iluminação do painel, aterramento, interferência eletromagnética, blindagem, supressores de ruído, distribuição das alimentações fora do painel e proteção contra raios consulte o Manual de Utilização Série Nexto - MU214000.

### Instalação Mecânica

Para detalhamento referente à instalação mecânica da Série Nexto consulte o Manual de Utilização Série Nexto - MU214000. Os tópicos abordados lá são: fixação do bastidor (furação, montagem e remoção), inserção e remoção dos módulos, módulos de E/S e inserção e remoção da tampa de conector de bastidor.

As UCPs NX3003, NX3004 e NX3005 devem estar posicionadas na posição 0 do bastidor da Série Nexto. São necessárias duas posições sequenciais, isto significa que esta ocupará as posições 0 e 1 de um dado bastidor. Já, as UCPs da Série Nexto (NX3010, NX3020 e NX3030) devem ser inseridas na posição 2 do bastidor, logo após o Módulo Fonte de Alimentação.

### Instalação Elétrica

**PERIGO:**

**Ao realizar qualquer instalação em um painel elétrico, certifique-se de que a alimentação geral do armário esteja DESLIGADA.**

A correta fixação dos cabos das UCPs e dos módulos do sistema garante a segurança do equipamento e seu correto funcionamento. Para isso, devem ser verificados os seguintes pontos:

- Os cabos junto aos bornes de ligação do painel de montagem devem estar com conexão segura e firme;
- Os bornes de alimentação e aterramento das partes do sistema devem estar firmes e bem conectados, assegurando boa passagem de corrente;
- A conexão do terra dos equipamentos ao terra do painel de montagem deve estar firme e com a bitola de cabo correta, para garantir bom aterramento e imunidade a ruído.

Confira também se as tensões das alimentações estão dentro dos valores especificados nas características técnicas.

**ATENÇÃO:**

Onde houver alta tensão, colocar etiqueta de aviso e proteções que não permitam o fácil acesso.

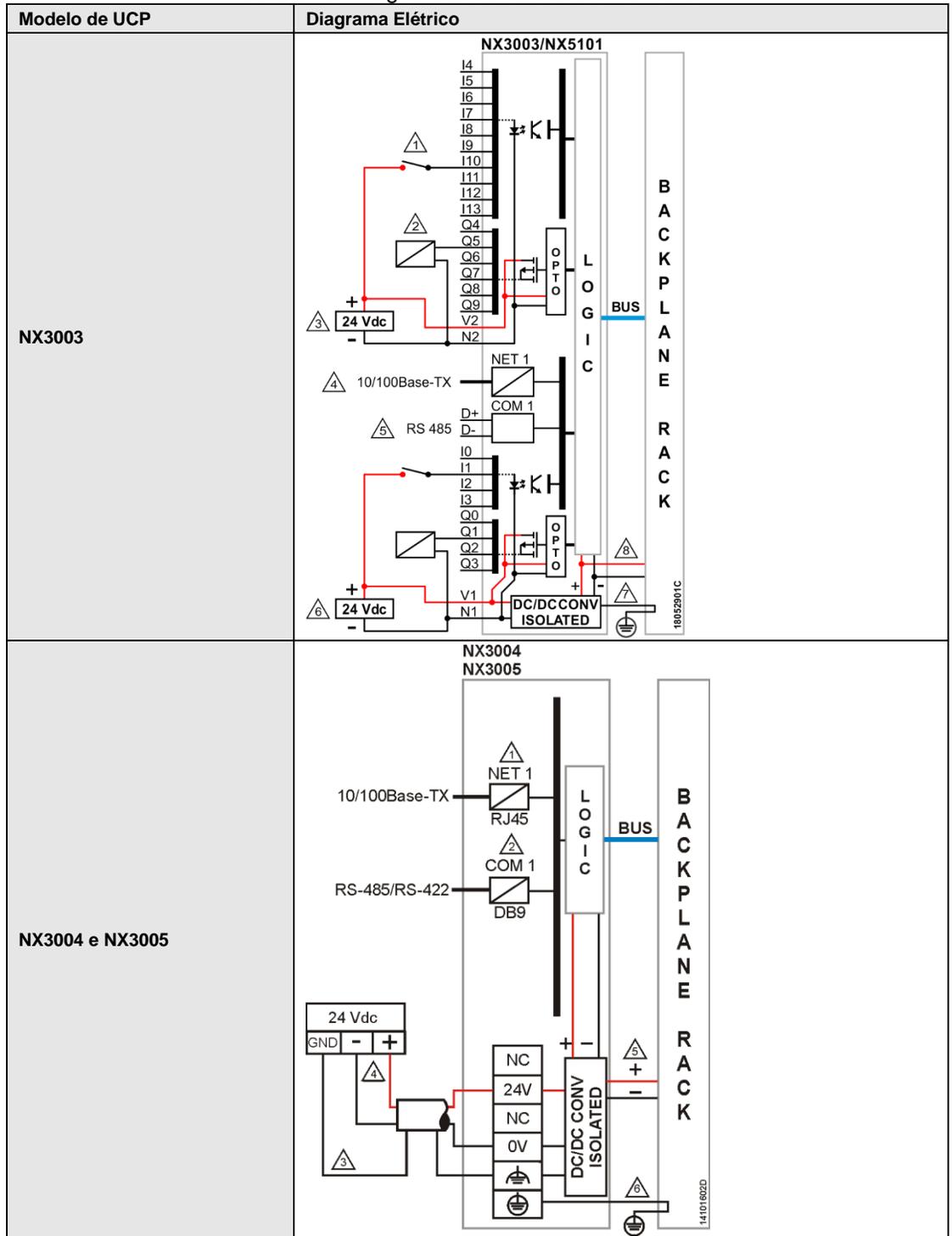
Recomenda-se verificar os fusíveis do sistema, certificando-se que eles estejam em bom estado e com valor e tipo correto, antes de energizar o sistema.

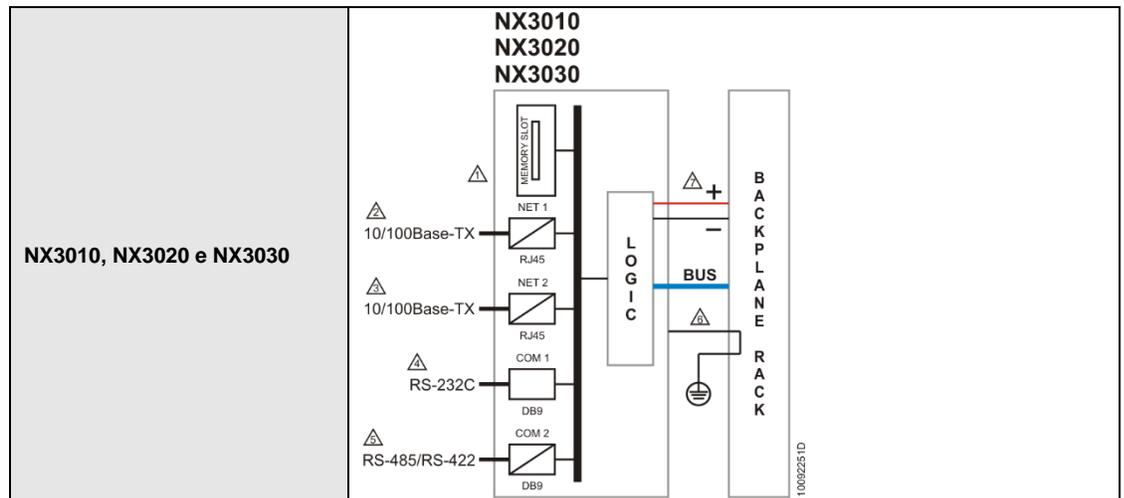


**PERIGO: nunca se deve substituir um fusível por outro de maior valor de corrente, sob pena de causar sérios danos ao equipamento.**

A alimentação das UCPs da Série Nexto é proveniente do Módulo Fonte de Alimentação, o qual fornece tensão às UCPs através da conexão ao bastidor, não necessitando de conexões externas. O aterramento do módulo é realizado através do contato entre a mola de aterramento do módulo e o bastidor. A tabela a seguir mostra o diagrama elétrico das UCPs da Série Nexto.

Tabela 1-9. Diagrama elétrico das UCPs Nexto





Para maiores informações consulte o Manual de Utilização UCPs Série Nexto NX30xx - MU214100.

### Instalação do Cartão de Memória

Inicialmente, deve-se ter atenção quanto à posição correta que o cartão de memória deve ser inserido. Existe um dos cantos se diferencia dos demais e o mesmo deverá ser utilizado como referência para a inserção correta do cartão. Sendo assim, o cartão de memória deverá ser colocado de acordo com o desenho localizado na parte frontal da UCP ou também como mostra a figura a seguir. Quando o cartão estiver instalado corretamente aparecerá um símbolo no visor gráfico da UCP.

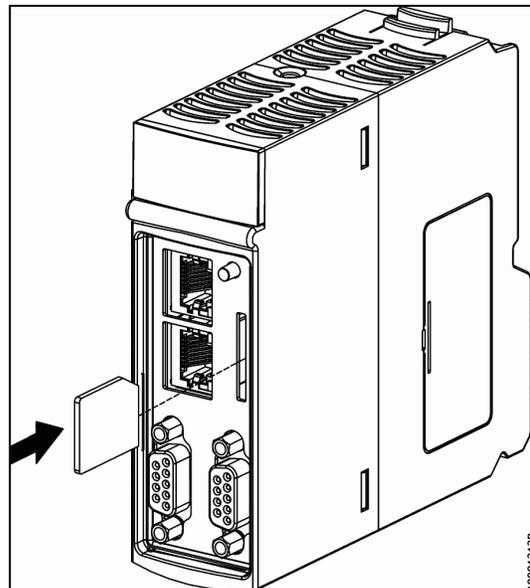


Figura 1-11. Inserção do Cartão de Memória na UCP

Para remover o cartão com segurança, pressione a tecla MS, espere que o símbolo do cartão desapareça do visor gráfico e, então, retire o cartão. Para que isso seja possível, pressione o cartão contra a UCP até que seja gerado um clique. Solte-o e retire-o do compartimento, conforme mostra a figura a seguir.

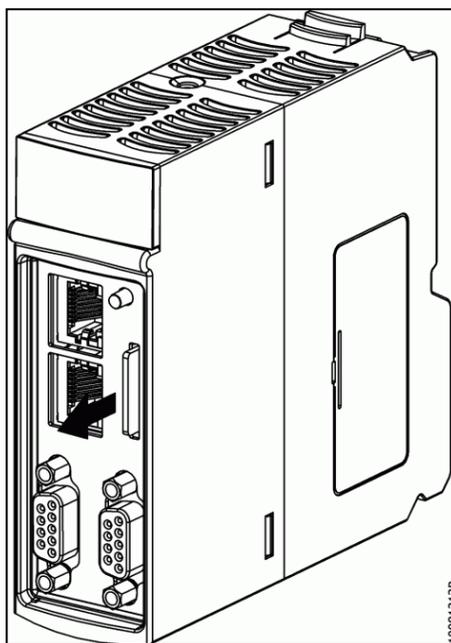


Figura 1-12. Retirada do Cartão de Memória

### Instalação da Arquitetura

A Série Nexto possui um exclusivo método para conectar e desconectar módulos do barramento, o qual não exige muito esforço do operador e garante a integridade da conexão. Para maiores informações sobre fixação dos produtos da Série Nexto, favor consultar o Manual de Utilização Série Nexto – MU214000.

A figura a seguir exemplifica a instalação de um módulo no bastidor. Primeiramente a parte inferior do módulo deve ser encaixada, pois ela serve como guia para a parte superior (localização do conector) ser acoplada com êxito.

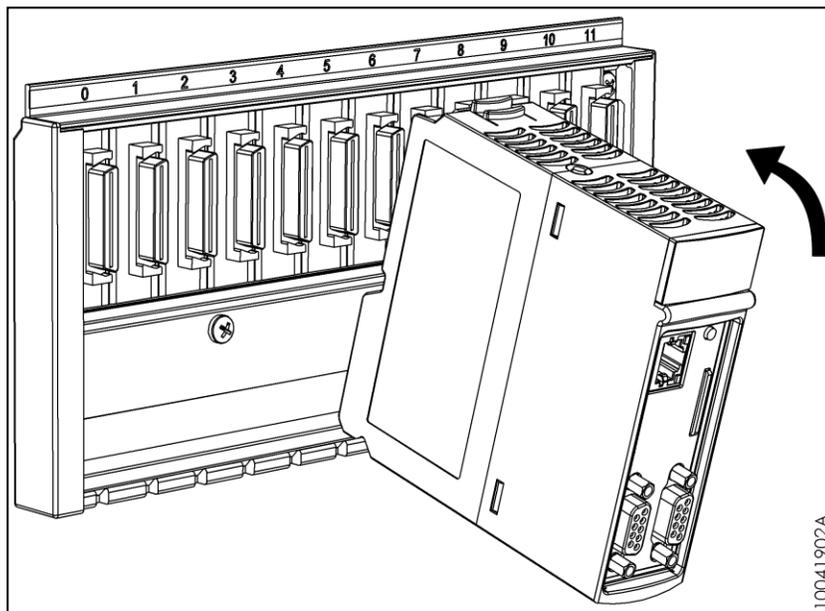


Figura 1-13. Instalação do Módulo no Bastidor

## Remoção de Módulos no Bastidor Principal

Para remover o módulo, repita o procedimento de inserção, porém inversamente. Primeiramente, pressione a trava de fixação (1) contra o próprio módulo a fim de destravá-lo do barramento e, então, puxe-o para frente, realizando um ângulo para baixo (2), conforme mostrado na figura a seguir.

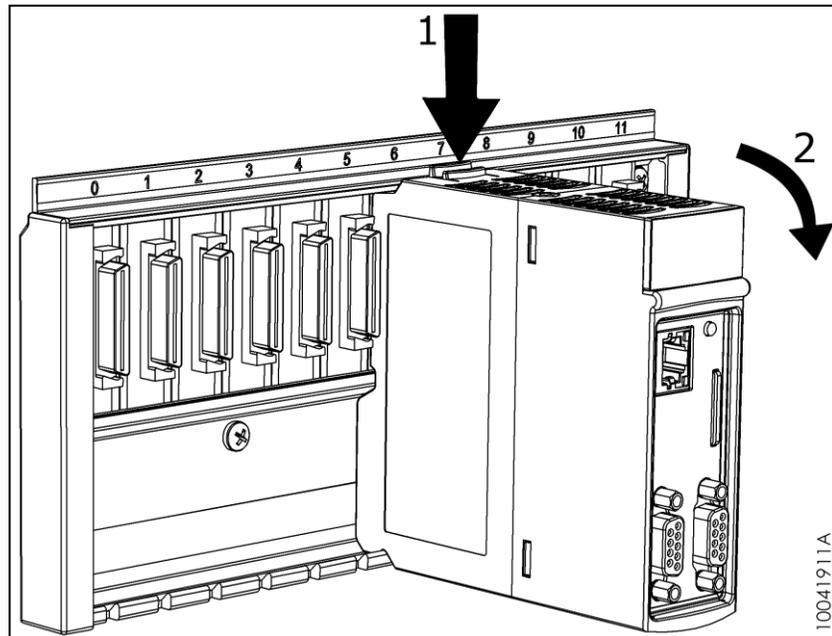


Figura 1-14. Desconexão do Barramento

## Conexão com a Rede Ethernet

As interfaces isoladas de comunicação NET 1 e NET 2 (somente para NX3020 e NX3030) possibilitam a conexão com uma rede Ethernet, onde a interface NET 1 é mais indicada para comunicação com o MasterTool IEC XE. Esta conexão utiliza cabos tipo par trançado (10/100Base-TX), sendo que a detecção da velocidade é realizada automaticamente pela UCP Nexto. Este cabo deve ter uma de suas extremidades ligadas à interface que se pretende utilizar e a outra ao HUB, switch, microcomputador ou outro ponto de rede Ethernet.

A interface de Ethernet NET 1 é utilizada para comunicação Ethernet e para configurar a UCP, para que isso seja possível, esta vem configurada de fábrica com parâmetros indicados a seguir.

Tabela 1-10. Conexão com a rede Ethernet

	NET 1
<b>Endereço IP</b>	192.168.15.1
<b>Máscara de subrede</b>	255.255.255.0
<b>Endereço de Gateway</b>	192.168.15.253

Os parâmetros Endereço IP e Máscara de Subrede podem ser visualizados no visor gráfico da UCP através do menu de parâmetros. Inicialmente, deve-se conectar a interface NET 1 da UCP a uma rede ou microcomputador com a mesma subrede para comunicação com o MasterTool IEC XE, onde os parâmetros de rede podem ser alterados.

A interface de Ethernet NET 2 disponível somente para as UCPs NX3020 e NX3030 é utilizada para comunicação Ethernet e vem configurada de fábrica conforme mostrado a seguir. A

visualização e o ajuste dos parâmetros de rede seguem as mesmas orientações da interface NET 1.

Tabela 1-11. Conexão com a rede Ethernet

	NET 2
<b>Endereço IP</b>	192.168.16.1
<b>Máscara de subrede</b>	255.255.255.0
<b>Endereço de Gateway</b>	192.168.16.253

A interface de Ethernet NET1/2 envia espontaneamente pacotes do tipo ARP, em broadcast, informando seu endereço de IP e MAC para todos os dispositivos interligados à rede. Estes pacotes são enviados durante o download de uma nova aplicação pelo software MasterTool IEC XE e na inicialização da UCP, quando a aplicação entra em modo Run. São disparados 5 comandos ARP com um intervalo inicial de 200 ms, dobrando o intervalo entre cada novo comando disparado, totalizando 3 s. Exemplo: o primeiro disparo ocorre no tempo 0, o segundo em 200 ms, o terceiro em 600 ms e assim até o quinto disparo no tempo 3 s.

As portas Ethernet das UCPs da Série Nexto possuem pinagem padrão, sendo a mesma utilizada, por exemplo, em computadores pessoais. A interface pode ser conectada em uma rede de comunicação através de um hub ou switch, ou então diretamente ao equipamento com o qual irá se comunicar. Neste último caso, devido as UCPs Nexto possuírem a característica de *Auto Crossover*, não se faz necessária a utilização de um cabo de rede denominado *crossover*, o qual é empregado na conexão de dois computadores, ponto a ponto, através da porta Ethernet. É importante ressaltar que se entende por cabo de rede, um par de conectores RJ45 machos interligados entre si por um cabo UTP ou ScTP, de categoria 5, sob a configuração direta ou *crossover*. O mesmo serve para interligar dois dispositivos com porta Ethernet. Normalmente estes cabos possuem uma trava de conexão que garante uma perfeita conexão entre o conector fêmea da interface e o conector macho do cabo. No momento da instalação, o conector macho do cabo deve ser inserido no conector fêmea do módulo até que se ouça um som característico ("click"), garantindo a atuação da trava. Para desconectá-los deve-se utilizar a alavanca presente no conector macho.

## Conexão com Redes Seriais

A interface não isolada de comunicação COM 1 das UCPs NX3010, NX3020 e NX3030 possibilita a conexão com uma rede RS-232C.

Por outro lado, a interface de comunicação na NX3003 denominada de COM 1 no configurador, utiliza os pinos D+ e D- indicados no painel frontal da UCP NX3003 e permite a conexão com uma rede serial RS-485. A terminação da rede serial é parametrizável via programador da UCP.

As interfaces isoladas de comunicação COM 1, para as UCPs NX3004 ou NX3005, e COM 2, para as UCPs NX3010, NX3020 ou NX3030, possibilitam a conexão com uma rede RS-485/422 e utilizam o conector DB9 fêmea para a conexão serial.

A tabela a seguir sumariza as conexões seriais típicas. Mais informações podem ser obtidas no Manual de Utilização CPs Série Nexto NX30xx - MU214100.

Tabela 1-12. Conexões seriais típicas na Série Nexto

Tipo de conexão serial	Diagrama
Interface RS-485 sem Terminação	
Interface RS-485 com Terminação Interna	
Interface RS-485 com Terminação Externa	
Interface RS-422 sem Terminação	
Interface RS-422 com Terminação Interna	
Interface RS-422 com Terminação Externa	



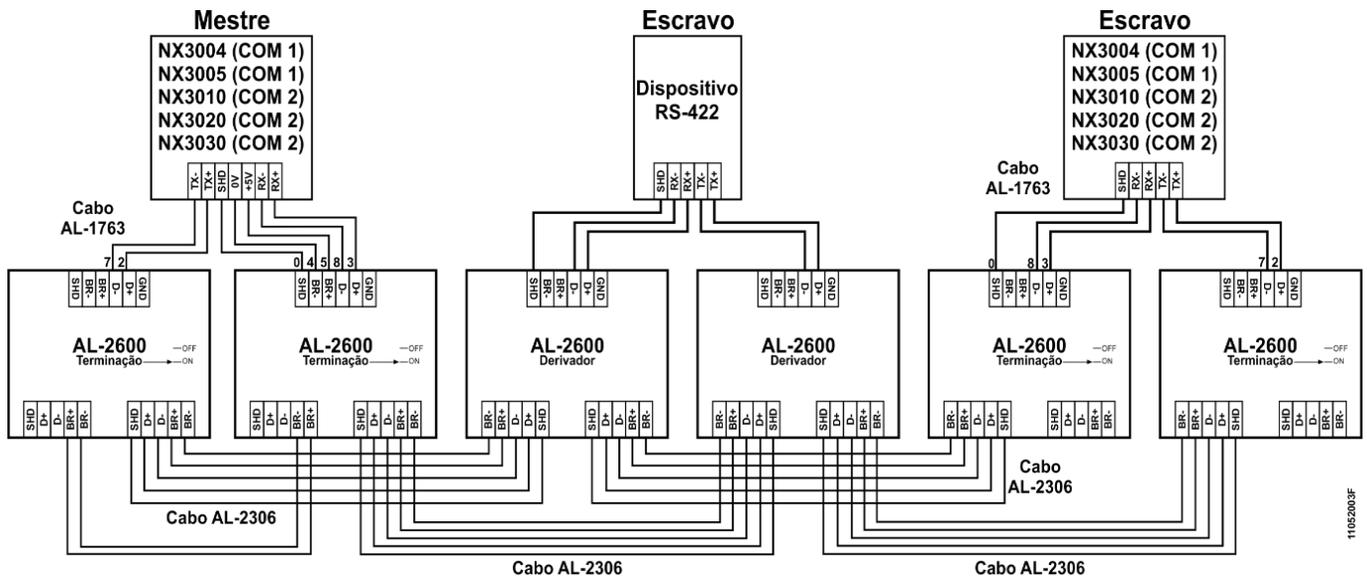
## **Pesquisa na Documentação: interfaces**

Consulte, na documentação do produto, as especificações das *Interfaces de Comunicação Ethernet*, *Interfaces de Comunicação Serial* e *Interface do Cartão de Memória*.

Fonte: Manual de Utilização Série Nexto – MU214000







## Um pouco de teoria... O meio físico RS485 e o Profibus



O padrão RS 485 é a tecnologia de transmissão mais frequentemente encontrada no Profibus. Sua aplicação inclui todas as áreas nas quais uma alta taxa de transmissão, aliada a uma instalação simples e barata, é necessária. Um par trançado de cobre blindado (shieldado) com um único par condutor é o suficiente neste caso.

A tecnologia de transmissão RS 485 é muito fácil de manusear. O uso de par trançado não requer nenhum conhecimento ou habilidade especial. A topologia por sua vez permite a adição e remoção de estações, bem como uma colocação em funcionamento do tipo passo-a-passo, sem afetar outras estações. Expansões futuras, portanto, podem ser implementadas sem afetar as estações já em operação.

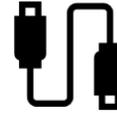
Taxas de transmissão entre 9.6 kbit/s e 12 Mbit/s podem ser selecionadas, porém uma única taxa de transmissão é selecionada para todos dispositivos no barramento, quando o sistema é inicializado.

Sempre que possível, consulte a EN50170 e a IEC60079-14 para as regulamentações físicas, assim como para as práticas de segurança em instalações elétricas em atmosferas explosivas.

Para casos com mais de 32 estações ou para redes densas, devem ser utilizados repetidores. Segundo a EN50170, um máximo de quatro repetidores é permitido entre duas estações quaisquer. Dependendo do fabricante e das características do repetidor, é permitido instalar até nove repetidores em cascata. Recomenda-se não utilizar uma quantidade maior que a permitida, devido aos atrasos embutidos na rede e ao comprometimento com o slot time (tempo máximo que o mestre irá esperar por uma resposta do slave).

Fonte: <http://www.profibus.org.br/>

## 2. Software do CP Nexto



### Características do MasterTool IEC XE



O MT8500 – MasterTool IEC XE é o software para programação, configuração, diagnóstico e comissionamento da Série Nexto. Ele é um software programador de CP independente de dispositivo. Este produto oferece características para todas as etapas de desenvolvimento de um sistema de automação, começando por uma análise gráfica da topologia da arquitetura, passando por um ambiente de programação com suporte às linguagens da norma IEC 61131-3, com uma ferramenta de simulação realística, onde o usuário pode verificar o comportamento da aplicação antes de executá-la em um sistema real, e, finalmente, provê uma completa interface para visualização de diagnósticos e status. A figura a seguir ilustra o ambiente de programação MasterTool IEC XE.

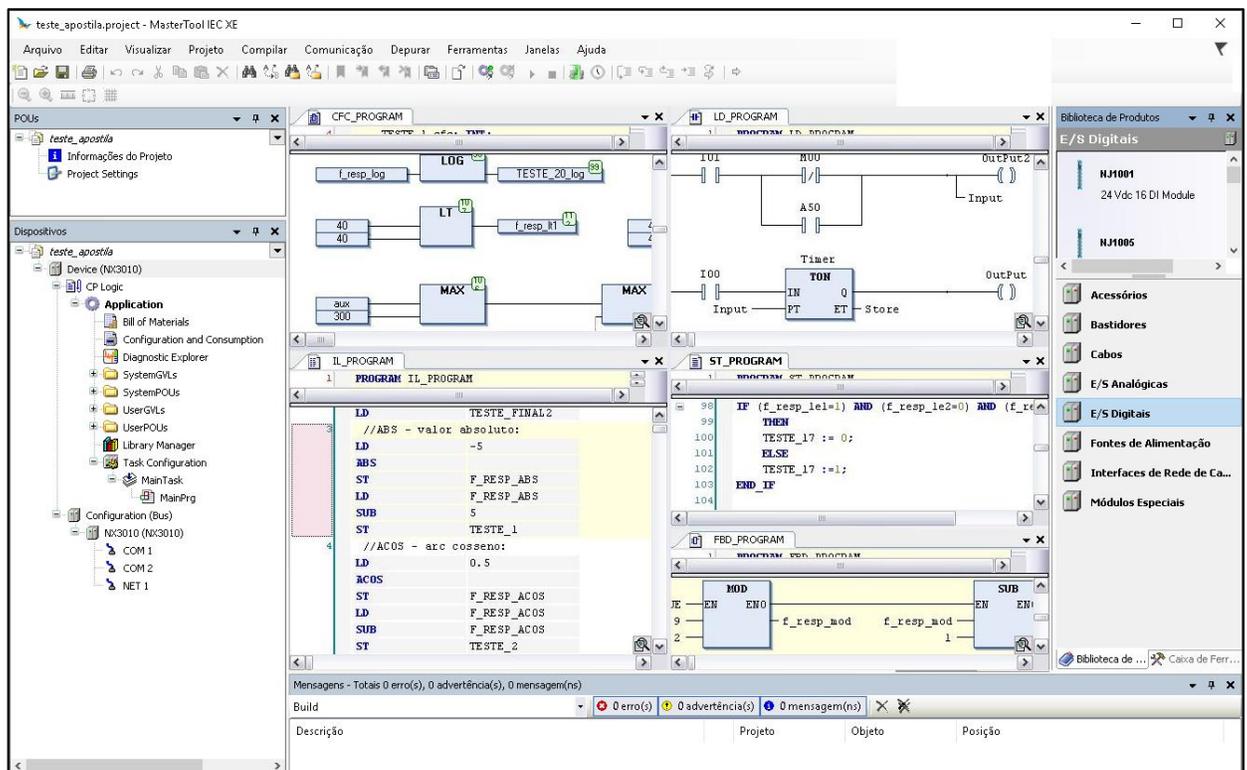


Figura 2-1. Ambiente de programação MasterTool IEC XE

As suas principais características estão resumidas na tabela a seguir.

Tabela 2-1. Características do MasterTool IEC XE

<b>Linguagens de Programação IEC 61131-3</b>	O MasterTool IEC XE oferece todos os editores definidos na norma IEC para desenvolvimento da aplicação: Texto Estruturado (ST), Sequenciamento Gráfico de Funções (SFC), Diagrama de Blocos Funcionais (FBD), Diagrama Ladder (LD), Lista de Instruções (IL) e Gráfico Funcional Contínuo (CFC). Todos os editores foram especialmente desenvolvidos para garantir um ótimo gerenciamento. Ideias e sugestões de usuários experientes foram incorporadas ao processo de desenvolvimento.
<b>Editores para Configuração de Projeto e de Hardware</b>	Com a ajuda de editores especiais, um projeto pode ser facilmente configurado no MasterTool IEC XE. A ferramenta gráfica permite uma rápida e amigável maneira de configurar o sistema. Adicionalmente, o usuário tem a visualização completa da arquitetura da aplicação com a posição física e as informações de cada módulo. A configuração das redes de campo e protocolos de comunicação padrão como PROFIBUS-DP e os protocolos MODBUS são integrados na ferramenta de programação. Esta característica permite que o usuário defina todos os parâmetros de configuração em um único lugar e não necessite usar diferentes ferramentas de software.
<b>Programação Orientada ao Objeto</b>	O aplicativo oferece uma programação orientada a objeto com as vantagens conhecidas das modernas linguagens de alto nível, como JAVA ou C++: classes, interfaces, métodos, herança, polimorfismo etc. Os blocos funcionais IEC são perfeitamente estendidos e as extensões estão disponíveis a todos os aspectos da engenharia. A programação orientada a objeto oferece grandes vantagens ao usuário, como por exemplo, quando se deseja reutilizar partes existentes de uma aplicação ou quando deseja trabalhar em uma aplicação com vários desenvolvedores.
<b>Funcionalidades: Online, Depuração e Comissionamento</b>	O código gerado a partir da aplicação é enviado para o dispositivo com um simples clique de mouse. A partir do momento em que o MasterTool IEC XE está online, são disponibilizadas várias funções importantes para realizar uma depuração rápida e eficiente, assim como para teste e comissionamento.
<b>Simulação</b>	Uma característica que permite ao usuário avaliar e testar diversas lógicas e algoritmos é a ferramenta de simulação. Este recurso permite que as aplicações do usuário sejam projetadas e testadas sem a necessidade de um controlador conectado. Isto é também interessante para treinamento, documentação e avaliação de casos de teste.
<b>Desenvolvimento de páginas Web (acessíveis através de protocolo HTTP)</b>	Além de permitir que o usuário crie a lógica responsável pelo controle de processos segundo a norma IEC61131-3, o MasterTool IEC XE também oferece um editor para o desenvolvimento de telas que permitem supervisão e operação do processo automatizado. Estas telas são desenvolvidas de maneira simples através de um editor gráfico que possui diversos objetos predefinidos e que são armazenados de forma vetorial juntamente com o projeto da lógica. Entre estes objetos se encontram formas geométricas, medidores de ponteiro e barras, controles para edição e visualização de variáveis, objetos que permitem carregar imagens entre outras funcionalidades. O editor das telas está integrado ao MasterTool IEC XE de forma que a interação destas com a lógica do CP é feita de forma transparente e intuitiva reduzindo significativamente o tempo de desenvolvimento e integração destas. Uma vez criadas as telas, elas são carregadas no CP em conjunto com o restante do programa e são acessadas utilizando um navegador web através do protocolo HTTP. Apesar das telas serem armazenadas e exibidas no formato web, não é necessário que os usuários possuam qualquer conhecimento sobre linguagens de programação, sendo toda a configuração feita de forma gráfica e intuitiva.
<b>Documentação de Usuário &amp; Arquivos de Ajuda</b>	Considerando que a programação do CP de acordo com as linguagens da norma IEC 61131-3 é uma tarefa complexa, o MasterTool IEC XE oferece um extenso arquivo de ajuda com várias dicas e descrições para orientar e servir como uma primeira base de dados de resolução de problemas para o usuário na criação de códigos de lógica ou no uso das características do software. Este arquivo de ajuda é disponibilizado em diferentes idiomas, de acordo com as opções de instalação. O MasterTool IEC XE também oferece suporte a múltiplos idiomas, permitindo ao usuário selecionar o seu idioma preferido a partir das opções disponíveis. A ideia é minimizar os possíveis problemas de entendimento decorrentes do uso de uma língua estrangeira. Como parte da documentação do usuário, o MT8500 pode imprimir documentos de aplicação tais como listas de materiais (BOM), POUs e parâmetros de configuração.
<b>Diagnósticos Avançados</b>	Uma das inovações chave da Série Nexto é o seu amplo suporte de diagnósticos. Esta ideia surgiu das demandas de extensas e complexas aplicações onde o correto uso de cada informação é fundamental para a manutenção, resolução de problemas e prevenção de potenciais problemas. Esta característica também está presente no MasterTool IEC XE onde o usuário, enquanto estiver conectado a uma UCP em execução, poderá acessar estruturas de diagnóstico completo através de janelas de monitoração e páginas da web.
<b>Visualização que utiliza o conceito de abas (tecnologia Docking View)</b>	A tecnologia <i>Docking View</i> permite ao usuário customizar o ambiente MasterTool IEC XE de acordo com suas necessidades pessoais. Adicionalmente, o usuário pode editar a estrutura do menu, as atribuições das teclas e a barra de ferramentas conforme desejado. Esta característica fornece uma interface de usuário amigável para maximizar a experiência com a ferramenta do software.

## Linguagens de Programação IEC 61131-3

Algumas características associadas às linguagens de programação são indicadas a seguir:

- Ao trabalhar em FBD, LD ou IL o usuário pode alternar entre estes editores;
- Os elementos de linguagens podem ser inseridos diretamente ou arrastados para o editor a partir de uma caixa de ferramentas;
- O MT8500 oferece um assistente de entrada inteligente e a funcionalidade autocompletar;
- As construções típicas de linguagem (declarações IF, laços FOR, classes de variáveis etc.) podem ser escondidas nos editores textuais;
- As construções de linguagem são criadas automaticamente (IF... END\_IF);
- O editor SFC pode ser usado conforme definido na norma ou na versão simplificada;
- Um tempo adequado de monitoramento para passos e para a funcionalidade de diagnóstico online também é disponibilizado no editor SFC;
- Os elementos no editor CFC podem ser encapsulados em macros.

O MasterTool IEC XE também oferece dois diferentes mecanismos de proteção da aplicação e características de segurança: Proteção da Propriedade Intelectual e Login Seguro no CP. A primeira permite proteger todo o projeto ou arquivos específicos dentro do projeto através da definição uma senha de acesso. Já o Login Seguro no CP provê uma maneira de proteger a aplicação do usuário de qualquer acesso não autorizado.

## Versões do MasterTool IEC XE

O software MasterTool IEC XE MT8500 possui quatro versões de distribuição, cada uma com um portfólio otimizado, de acordo com a necessidade do usuário. São elas:

1. **Lite:** software programador gratuito que permite a programação e carga de projetos de até 320 pontos de E/S no barramento Nexto;
2. **Basic:** software que permite a programação e carga de projetos de até 2048 pontos de E/S no barramento Nexto;
3. **Professional:** software programador para todas as UCPs da Série Nexto;
4. **Advanced:** software programador com ferramentas para aplicações de grande porte com redundância de *half-cluster*.

## Informações Gerais sobre o Programador

Para informações detalhadas sobre o aplicativo incluindo compatibilidade com os demais produtos, requisitos mínimos e recomendáveis, idiomas suportados e dados para compra consulte o Manual de Utilização MasterTool IEC XE MT8500 - MU299048.

# Funcionalidades do MasterTool IEC XE



A seguir, são apresentadas algumas funcionalidades disponíveis no MasterTool IEC XE.

## Instalação

Para realizar a instalação do software de desenvolvimento MasterTool IEC XE é necessário ter em mãos o CD-ROM de distribuição ou efetuar o download do arquivo de instalação no site

<http://www.altus.com.br>. Após, feche todos os programas que estejam em execução no seu computador e em seguida dê um duplo clique no arquivo de instalação e siga as orientações do aplicativo instalador.

Para desinstalar o programador e seus componentes ou para modificar a instalação atual, execute o arquivo de instalação.

## Projeto

Um projeto contém os objetos das POUs que compõem um programa do CP, assim como as definições dos objetos de recursos necessários para executar uma ou mais instâncias do programa (aplicação) em determinados sistemas-destino (CPs, dispositivos). Objetos de POUs podem ser gerenciados na janela de visualização das POUs ou na janela de visualização dos dispositivos, POUs criadas a partir do wizard, aparecem na janela visualização dos dispositivos, e os objetos de recursos específicos do dispositivo são gerenciados na janela de visualização dos dispositivos. Um projeto é salvo no arquivo <nome do projeto>.project. A configuração básica de um novo projeto (estrutura de menu, objetos pré-definidos) é determinada pelo modelo do projeto usado. Este modelo é escolhido na criação de um novo arquivo de projeto com o comando *Novo Projeto*.

## Perfis de Projeto

Um perfil de projeto no MasterTool IEC XE é um conjunto de regras, características comuns e padrões utilizados no desenvolvimento de uma solução de automação industrial, um perfil que influencia a forma de implementação da aplicação. Com a diversidade de tipos de aplicações suportadas pelo Runtime System (RTS) da Série Nexto, seguir um perfil é uma forma de reduzir a complexidade na programação. As aplicações podem ser criadas conforme um dos seguintes perfis:

- Simple
- Básico
- Normal
- Experiente
- Personalizado
- Perfil de Máquina

O MasterTool IEC XE pode disponibilizar inúmeros templates compatíveis para cada perfil definido para o RTS. Quando o usuário seleciona um template como modelo na criação de um projeto, a nova aplicação será desenvolvida conforme um determinado perfil, adotando as regras, características e padrões definidos pelo perfil associado ao template. Para mais informações sobre cada um destes perfis consulte o Manual de Utilização UCPs Série Nexto – MU214100.

## Objetos de Recurso

São objetos de dispositivos, aplicações, configurações de tarefas e são gerenciados na árvore de dispositivos ou no editor gráfico, dependendo do tipo do dispositivo. Ao inserir objetos, o hardware a ser controlado deve ser mapeado de acordo com determinadas regras.

## Objetos de Programação

Os objetos de programação (POUs) que podem ser instanciados no projeto (em todas as aplicações definidas no mesmo) devem ser gerenciados na janela das POUs. Alguns exemplos destes objetos de programação são: programas, funções, blocos funcionais, métodos, ações, definições de tipos de dados, entre outros. O instanciamento é feito ao chamar uma POU de

programa através de uma tarefa atribuída à aplicação. Os objetos de programação gerenciados na janela de dispositivos (atribuídos diretamente a uma aplicação) não podem ser instanciados por outra aplicação inserida abaixo.

#### Janela das POUs

Na janela *POUs* podem ser acrescentados POUs, arquivos externos, entre outros. Além de exibir objetos de configurações e informações de projeto conforme ilustrado na figura a seguir.

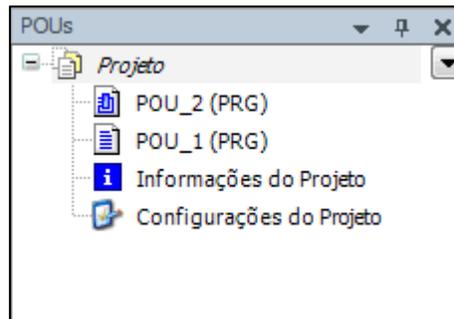


Figura 2-2. Janela dos POUs

#### Dispositivos, Árvore de Dispositivos

Na janela *Dispositivos* ("árvore de dispositivos") define-se o hardware onde a aplicação será executada. Cada "dispositivo" representa um hardware específico (destino). Exemplos: controlador, módulos de E/S, monitor. Ele é definido por uma descrição e deve ser instalado no sistema local para que possa ser inserido na árvore de dispositivos (conforme mostrado na figura a seguir).

O arquivo de descrição define as propriedades referentes à configuração, programação e possíveis conexões com outros dispositivos.

Na árvore de dispositivos são gerenciados os objetos necessários para executar uma aplicação no dispositivo (controlador, CP), incluindo aplicação, configuração de tarefas e tarefas. Entretanto, objetos de programação específicos (POUs, listas de variáveis globais e gerenciador de bibliotecas) podem - em vez de serem gerenciados como unidades instanciáveis globais de projeto na janela das POUs - serem gerenciados SOMENTE na árvore de dispositivos e, neste caso, estarem disponíveis apenas na sua aplicação ou nas suas "aplicações secundárias".

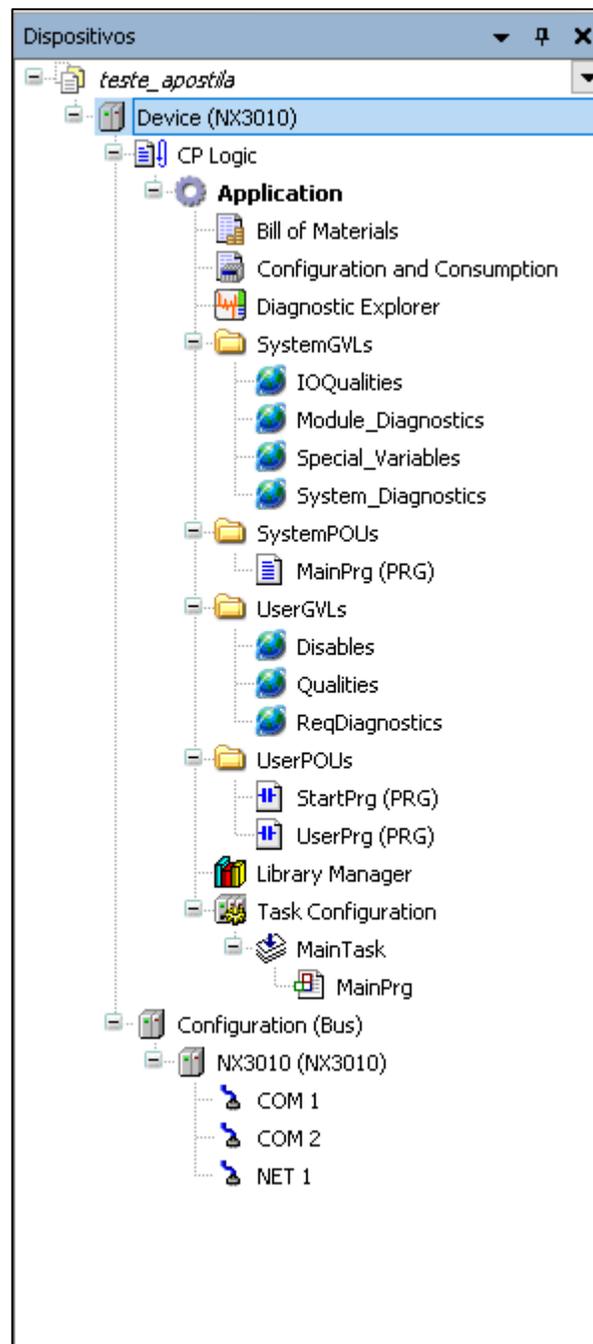


Figura 2-3. Árvore de Dispositivos

## Aplicação

Uma “aplicação” é o conjunto dos objetos necessários para executar uma instância específica do programa do CP em um determinado dispositivo de hardware (CP, controlador). Para isto, objetos “independentes” gerenciados na visualização das POU são instanciados e atribuídos a um dispositivo na janela visualização dos *Dispositivos*. Isto está em conformidade com a programação orientada a objetos. Entretanto, POU específicas da aplicação também podem ser utilizadas.

Uma aplicação é representada por um objeto de aplicação () na árvore de dispositivos inserido abaixo de um nó do dispositivo programável (PLC Logic). Os objetos que definem o “conjunto de recursos” da aplicação podem ser inseridos abaixo de um item da aplicação.

A aplicação padrão, “Application”, é criada junto com novos projetos criados a partir do modelo *Projeto MasterTool Padrão* ela é acrescentada à árvore de dispositivos abaixo do item Device e PLC Logic.

Uma parte essencial da aplicação é a Task Configuration que controla a execução de um programa (instâncias de POU ou POUs específicas da aplicação). Adicionalmente, podem estar atribuídos objetos de recursos, tais como listas de variáveis globais, bibliotecas, entre outros, os quais - ao contrário daqueles gerenciados na janela das POUs - somente podem ser usados pela aplicação específica e seus subitens.

A compatibilidade dos parâmetros da aplicação no CP com os parâmetros da aplicação do projeto é verificada no momento do login em um dispositivo (CP ou dispositivo de simulação). Em caso de incompatibilidade, é exibida uma mensagem apropriada.

## Task Configuration

A Task Configuration () define uma ou várias tarefas para controlar o processamento de um programa aplicativo.

Ela é um objeto de recurso essencial para uma aplicação e é inserido automaticamente ao criar um novo projeto a partir do modelo *Projeto MasterTool Padrão*. Uma tarefa pode chamar uma POU de programa específica da aplicação que esteja disponível na árvore de dispositivos, assim como o programa gerenciado na janela das POUs. Neste último caso, o programa do projeto global disponível será instanciado pela aplicação.

Uma task configuration pode ser editada no editor de tarefas, sendo as opções disponíveis específicas do dispositivo.

No modo online o editor de tarefas fornece uma visualização da monitoração e informações sobre os ciclos, tempos e status.

## Geração de Código e Alteração Online

O código fonte não será gerado até que o projeto da aplicação seja enviado para o dispositivo (CP ou dispositivo de simulação). A cada envio, as informações de compilação contendo o código e a ID de referência da aplicação carregada serão armazenadas em um diretório do projeto em um arquivo “<nome do projeto>.<nome do dispositivo>.<ID da aplicação>.compileinfo”. As informações de compilação serão apagadas quando os comandos *Limpar* e *Limpar Tudo* forem executados. Dependendo da alteração no projeto em execução no controlador, apenas os objetos modificados serão carregados para o mesmo.

## Monitoração

No modo online existem várias possibilidades de exibir os valores atuais das expressões de monitoração de um objeto no CP:

## Depuração

Para avaliar erros de programação use a funcionalidade de depuração do MasterTool IEC XE no modo online. Neste contexto, considere a possibilidade de verificar a aplicação no modo de simulação, isto é, sem a necessidade de conectar-se com um dispositivo de hardware real.

Breakpoints podem ser configurados em determinadas posições para forçar uma interrupção na execução. Algumas condições podem ser configuradas para cada breakpoint, como por exemplo, quais as tarefas associadas ou em quais ciclos o breakpoint deve atuar. Funções de passos estão

disponíveis para que um programa seja executado em passos controlados. A cada interrupção, os valores atuais das variáveis podem ser examinados. Uma pilha de chamadas pode ser visualizada na posição do passo atual.

#### **Segurança**

Através dos diálogos Configurações do Projeto e Propriedades do Objeto, são configurados e gerenciados o controle de acesso aos projetos e objetos, assim como o direito de executar determinadas ações em um projeto. Um projeto pode ser protegido por uma senha codificada. Os Direitos de Acesso referentes a objetos são sempre atribuídos a grupos de usuários específicos, e não a usuários únicos. Para cada objeto, pode ser definida uma lista de ações permitidas para cada grupo. Cada usuário, no entanto, pode ter uma senha própria. Dependendo do dispositivo pode haver também um controle de acesso relacionado a objetos e arquivos no CP.

# Interface do Usuário do MasterTool IEC XE



A interface de programação do usuário MasterTool IEC XE é estruturada através de “componentes” (veja abaixo na figura).

Esta aparência depende da estruturação das janelas que podem ser alteradas a qualquer momento pelo usuário (ações relacionadas à minimização/restauração, redimensionamento do tamanho ou fechamento de janelas).

A interface do usuário fornece menus e barras de ferramentas, organização de editores e objetos, janelas de monitoração e mensagens, além de uma linha de informação e status.

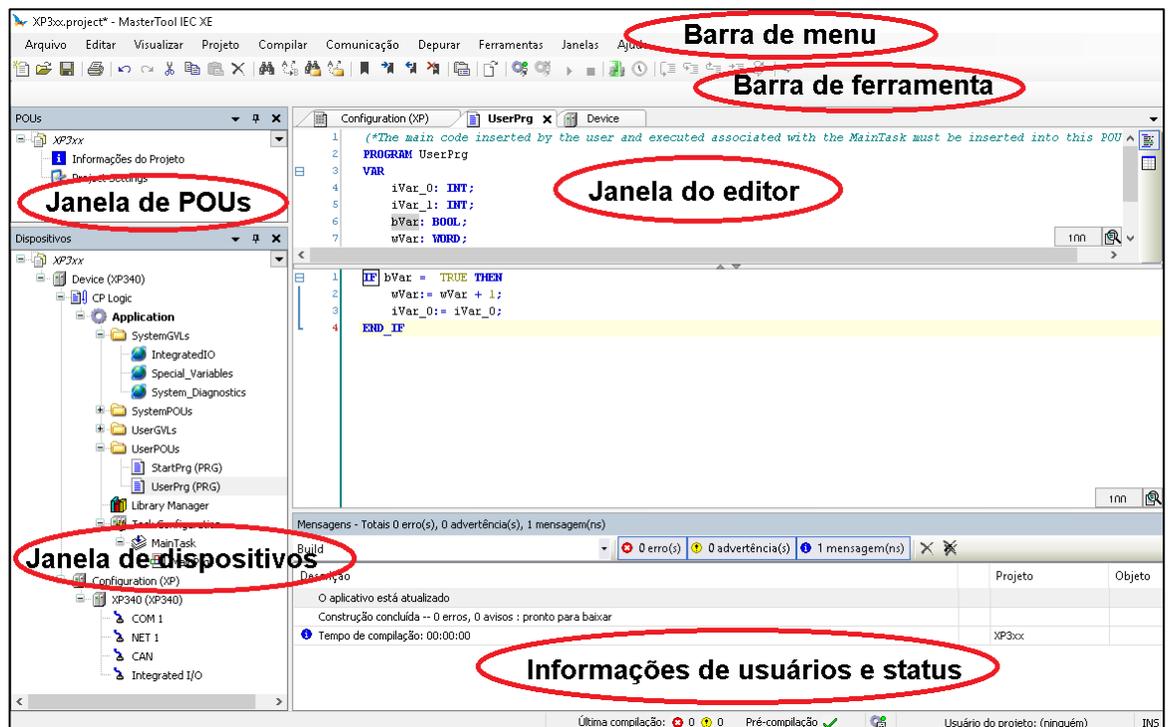


Figura 2-4. Exemplo do MasterTool IEC XE Interface do Usuário

Os componentes padrão são listados a seguir:

**Barra de menu:** fornece menus com todos os comandos atualmente disponíveis.

**Barra de ferramentas:** contém botões de ferramentas para todas as ferramentas atuais disponíveis.

**Janela de POUs:** organiza as unidades de programação (POUs, DUTs, etc.) de um projeto em uma estrutura de árvore (aberta no menu Visualizar).

**Janela de dispositivos:** organiza os objetos dos recursos dos dispositivos de um projeto em uma estrutura de árvore (aberta no menu Visualizar).

**Janela do editor:** cria um objeto específico no respectivo editor. No caso de editores de linguagem (por exemplo, Editores ST ou CFC), em geral, estas janelas estão na parte inferior, enquanto que no editor de declaração estão na parte superior. No caso de outros editores, esta janela também fornece diálogos (por exemplo, Editor de tarefas, Editor de dispositivos, Editor de UCP). O nome das POUs ou dos objetos dos recursos sempre são exibidos na barra de títulos. Os objetos podem ser abertos na janela do editor nos modos online ou off-line através do comando Editar Objeto.

Nos componentes abaixo, encontram-se mais informações sobre o projeto nos modos online ou off-line:

*Janela de Mensagens:* os comandos Precompilar, Compilar, Montar, Enviar mensagens, entre outros, são exibidos nesta janela.

*Janelas de Monitoração e visualizações online dos editores:* mostram uma visualização da monitoração de uma POU e uma lista de expressões de monitoração definida pelo usuário.

*Linha de informação e status:* a linha na borda inferior da interface do usuário fornece informações sobre o usuário conectado no momento. A posição do cursor e o status do modo de edição também serão exibidos se o usuário estiver trabalhando em uma janela do editor. No modo online, é indicado o status do programa.

*Usuário Atual:* cada projeto tem um usuário e gerenciamento de acesso. O usuário conectado no momento será nomeado na linha de status.

*Posição:* considerada a partir da margem superior esquerda da janela do editor em:

- Ln = número de linhas.
- Col = número de colunas (uma coluna inclui exatamente 1 espaço, caractere ou dígito).
- Ch = número de caracteres (neste contexto, um caractere pode ser um dígito ou caractere único, assim como também pode ser uma guia incluindo, por exemplo, 4 colunas).

Com um duplo clique em um dos campos é exibido o diálogo Ir para a linha, onde o usuário pode direcionar o cursor para outra posição.

*Status do modo de edição:* INS refere-se ao modo de inserção e OVR refere-se ao modo de sobrescrita. Com um duplo clique neste campo o usuário alterna entre as duas configurações.

*Informação no modo online:* status da aplicação no dispositivo conforme descrito a seguir:

- **RUN** = programa em execução.
- **STOP** = programa parado.
- **HALT ON BP** = programa interrompido em um breakpoint.
- Programa Carregado = o programa está carregado em um dispositivo.
- Programa Inalterado = o programa no dispositivo coincide com o programa do programador.
- Programa Modificado (alteração online) = o programa no dispositivo é diferente do programador e requer uma alteração online.
- Programa Modificado (envio completo) = o programa no dispositivo difere do programador e um envio completo se faz necessário.

## Janelas, Visualizações e Janelas do Editor

As janelas podem ser exibidas dentro ou fora da moldura da interface do usuário. Embora à primeira vista pareçam iguais, existem dois tipos de janelas:

Tipo 1: Algumas podem ser encolhidas para qualquer margem da moldura ou podem ser posicionadas na tela como janelas inteiras, independentemente da moldura da janela. Elas também podem estar "ocultas" - situação representada por uma guia na borda da moldura. Estas janelas exibem informações que não dependem de um único objeto do projeto, por exemplo, Mensagens, Dispositivos, POUs, Caixa de Ferramentas. Elas podem ser acessadas através dos comandos do menu *Visualizar* e são chamadas "visualizações". A maioria das visualizações inclui uma barra de ferramentas não configurável com botões para classificar, visualizar e pesquisar na janela.

Tipo 2: Outras abrem na visualização ou na edição de um objeto específico do projeto no respectivo editor. Estas janelas são exibidas em uma área do editor, ou dependendo das

configurações da interface do usuário. Estas janelas não podem ser ocultadas nem encolhidas na moldura da janela e podem ser acessadas através dos comandos do menu *Janelas*.

Tipos adicionais de janelas ou visualizações podem ser acrescentados através de componentes específicos do fornecedor.

## Customizando a Interface do Usuário

A aparência da interface do usuário, ou seja, a estruturação e a configuração dos seus componentes específicos, dependem dos itens abaixo:

Pré-configurações padrão para menus, funções de teclado e barras de ferramentas. As configurações padrão vêm instaladas no MasterTool IEC XE.

Propriedades de um editor definidas nos respectivos diálogos de Opções: estas pré-configurações podem ser igualmente substituídas pelo usuário e a configuração atual também será salva no sistema local.

Organização das visualizações ou das janelas de edição no projeto, feitas pelo usuário: as posições atuais serão salvas com o projeto.

## Organizando Barras de Menu e Barras de Ferramentas

A barra de menu está sempre posicionada no topo da interface do usuário, entre a barra de título e as janelas de visualização. Uma barra de ferramentas pode ser posicionada na mesma área que a barra de menu (fixa) ou como uma janela independente em qualquer lugar da tela.

Para reposicionar uma barra, clique com o cursor na linha pontilhada na extremidade final esquerda da barra e, mantendo-o pressionado, conduza a barra até a posição desejada.



Figura 2-5. Linha Pontilhada no Menu e Barras de Ferramentas

❗ **NOTA:** Nas janelas de visualização, como por exemplo, POUs, Dispositivos ou Propriedades de elementos visuais, está disponível uma barra de ferramentas especial que fornece botões para classificação, visualização e localização. Esta barra não pode ser configurada.

## Zoom

Cada janela dos editores das linguagens de programação apresentam uma função de zoom (🔍), exceto no editor da linguagem ST. Este botão, localizado no canto inferior direito da janela, abre uma lista com os níveis permitidos: 25, 50, 75, 100, 150, 200, 400 %. A impressão refere-se à visualização em 100%. O usuário também pode definir o nível de zoom de acordo com o seu interesse, bastando para isto digitar o valor desejado no campo correspondente.

## Interface do Usuário no Modo Online

Assim que o login for realizado no projeto, todos os objetos abertos no modo off-line serão automaticamente visualizados no modo online.

Para abrir um objeto no modo online (já aberto no modo off-line), execute um duplo clique no mesmo, na janela das POUs ou dispositivos ou utilize o comando *Editar Objeto*.

Se este procedimento foi executado corretamente, o objeto será aberto no modo online. Caso contrário, por exemplo, se haviam várias instâncias do objeto selecionado (blocos funcionais, etc.) contidas no projeto, o diálogo *Selecionar Estado Online <nome do objeto>* será exibido para que se possa escolher o que deve ser visualizado (uma instância ou a implementação básica do objeto). Neste diálogo, também se define o modo de exibição do objeto (modo online ou off-line).

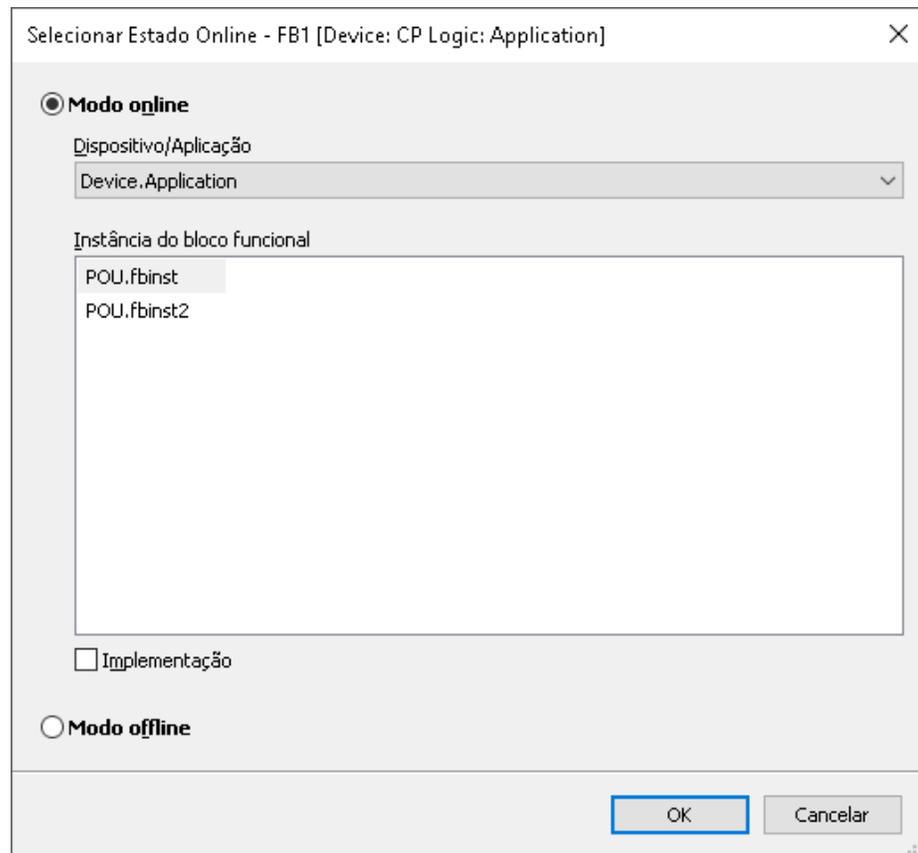


Figura 2-6. Diálogo de Estado Online

Nos campos *Dispositivo/Aplicação* veja o Dispositivo e a Aplicação aos quais o respectivo objeto está associado.

Para abrir a visualização online do objeto, ative a opção *Modo Online* e pressione *OK*. Para a visualização off-line, ative a opção *Modo Offline*.

Se o objeto é um bloco funcional, no campo *Instância do bloco funcional* haverá uma lista de todas as instâncias atuais usadas na aplicação. Neste caso, o usuário pode:

Selecionar uma das instâncias e ativar os modos online ou off-line, ou

Selecionar a opção *Implementação*, que, independentemente da instância selecionada, abrirá a visualização da implementação básica do bloco funcional. A implementação não tem efeito para objetos não instanciados.

Para mais informações sobre visualizações online dos editores específicos, consulte a descrição dos respectivos editores.

A barra de status fornecerá informações sobre o status atual da aplicação.

## Menus e Comandos Padronizados

Veja a seguir uma visão geral sobre a estrutura dos menus principais e comandos.

Os comandos especiais para um determinado editor estarão disponíveis no menu correspondente quando este editor for aberto (exemplo: ao editar um objeto no editor SFC, o menu *SFC* será acrescentado à barra de menu).

Figura



Figura 2-7. Barra de Menu Padrão

## Memória de Arquivos de Usuário

As UCPs da Série Nexto possuem uma área de memória destinada ao armazenamento de dados de uso geral, ou seja, o usuário poderá gravar diversos arquivos de projeto na memória da UCP utilizada. Essa área de memória irá variar de acordo com o modelo de UCP utilizada.

Essa funcionalidade é acessada com um duplo-clique sobre o item *Device* e selecionando a aba *Arquivos*, conforme mostram as indicações da figura a seguir.

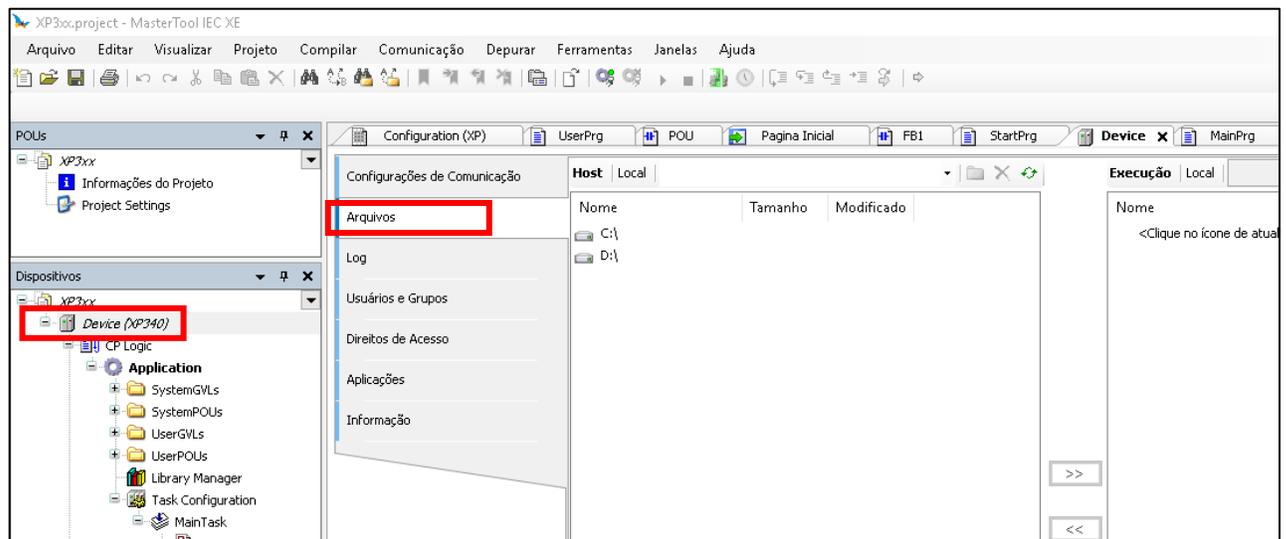


Figura 2-8. Utilização da Memória de Arquivos de Usuário

## Gerenciamento de Usuários e Direitos de Acesso

Fornecem funções para definir contas dos usuários e configurar os direitos de acesso ao projeto. Observe que o gerenciamento de usuários específico do dispositivo deve ser suportado para controlar os direitos de acesso no sistema de arquivos do CP e objetos durante a execução. Os direitos para acessar os objetos do projeto via ações especificadas são atribuídos somente a grupos e, portanto, cada usuário deve pertencer a um grupo.

## Editores

O MasterTool IEC contempla vários editores para configuração de objetos, dispositivos, parametrização de rede e outros tipos de ajustes. Cada tipo de editor possui as suas peculiaridades. Entretanto algumas características são gerais e válidas para todos os editores.

Nos editores sempre que existirem campos numéricos sendo configurados estes campos possuem um limite mínimo e máximo de valor que depende da funcionalidade do campo. Por exemplo, um campo de time-out pode ter valor mínimo de 10 s e o valor máximo de 65535 s. Nesses casos não é possível configurar valores fora dessa faixa e esta consistência é executada durante a configuração. Para alguns campos numéricos esta consistência é executada em uma

das etapas de geração de código, isso também se dá quando existe dependência entre diferentes campos, como por exemplo entre o tempo de ciclo de uma tarefa e o seu valor de cão-de-guarda.

Já nos casos que parâmetros representam endereços de mapeamentos diretos nas áreas de %I, %Q e %M a consistência é executada apenas durante a geração de código. Como os projetos podem ser feitos para quaisquer modelos de UCP e estes possuem tamanhos de áreas diferentes os campos que representam faixas de endereço dentro destas áreas estão limitados entre 0 e 2147483647. Durante a geração de código os valores atribuídos na configuração são consistidos com os limites disponíveis para cada modelo de UCP e caso estejam fora destes limites uma mensagem de erro é gerada durante o processo.

Em módulos de entrada e saída a guia Mapeamentos de E/S não mostrará os mapeamentos com seus valores atuais quando ela se encontrar aberta e o comando Reset Origem for executado, exibirá <Bad> na coluna Valor Atual. Para que os valores sejam devidamente exibidos e atualizados deve-se realizar o comando Download (menu Comunicação), fechar e abrir a janela do editor do módulo. Para mais informações consulte Reset Origem e Download.

Os editores disponíveis no programador estão listados na sequência.

- ▷ Editor de Barramento
- ▷ Editor de Módulos de E/S Digital
- ▷ Bill of Materials
  - Configuration and Consumption
- Editores das Linguagens de Programação
- ▷ Editor de Declaração
- ▷ Editor de Dispositivos
- ▷ Editores para Comunicação OPC
  - Editor DUT
  - Editor FBD/LD/IL
  - Editor de Lista de Variáveis Globais
- ▷ Editor Library Manager
- ▷ Editor Recipe Manager
- ▷ Editor Recipe Definition
- ▷ Editor de Tarefas
- ▷ Editor Trace
- ▷ Editor de Lista de Monitoração
- ▷ Editores MODBUS
  - Editor PROFIBUS
  - Editor UCP
- Interfaces Seriais
- Interfaces Ethernet
- ▷ Editor do Controle PID
- ▷ Visualização

Figura 2-9. Editores

Para maiores informações sobre os comandos de menu, memória de arquivos de usuário, gerenciamento de usuários e direitos de acesso, bem como sobre os editores consulte o Manual de Utilização UCPs Série Nexto – MU214100.

## Primeiros Passos com o MasterTool IEC XE

O objetivo desta seção é indicar os passos básicos para a programação das UCPs da Série Nexto. Serão explorados os seguintes itens:

- Criar e executar projetos;
- Desinstalar, atualizar e reparar;
- Obter ajuda.

## Iniciando o MasterTool IEC XE

A partir do menu Iniciar do PC selecione a opção MasterTool IEC XE. A inicialização também pode ser feita através do ícone MasterTool IEC XE que está disponível no desktop após a instalação. Inicialmente, o usuário deverá criar um novo projeto no MasterTool IEC XE a partir do menu Arquivo e logo em seguida, Novo Projeto..., conforme mostra a figura a seguir.

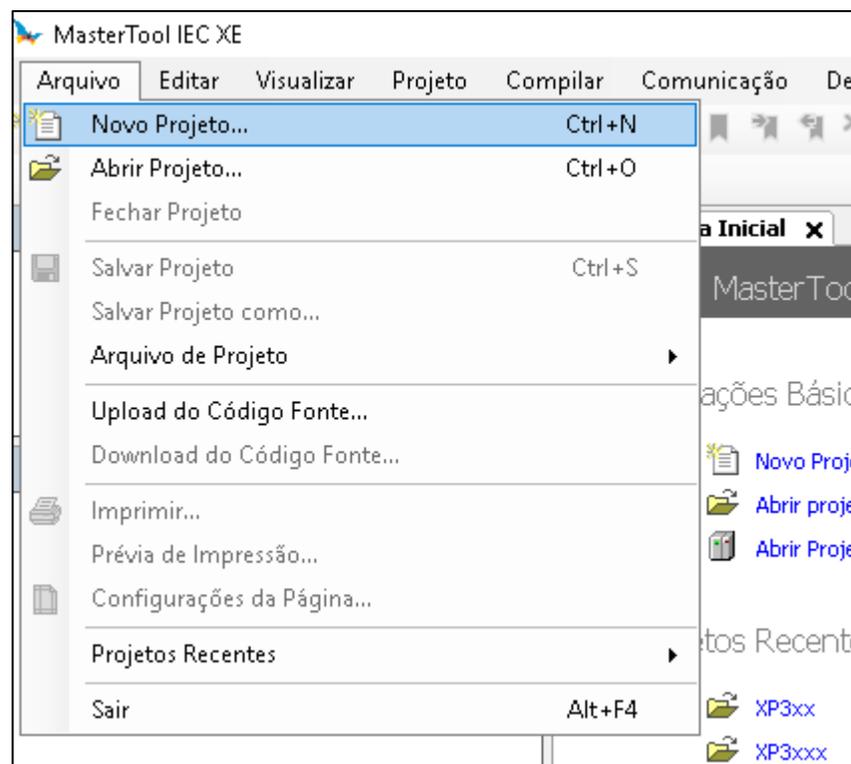


Figura 2-10. Novo Projeto

Após, uma janela será apresentada ao usuário, solicitando que ele selecione o tipo de projeto que deseja fazer e, em seguida, escreva um nome e a localização para armazenar o projeto no computador. Clicar em *OK* para prosseguir. A opção de cancelamento também está disponível nessa janela via botão *Cancelar*.

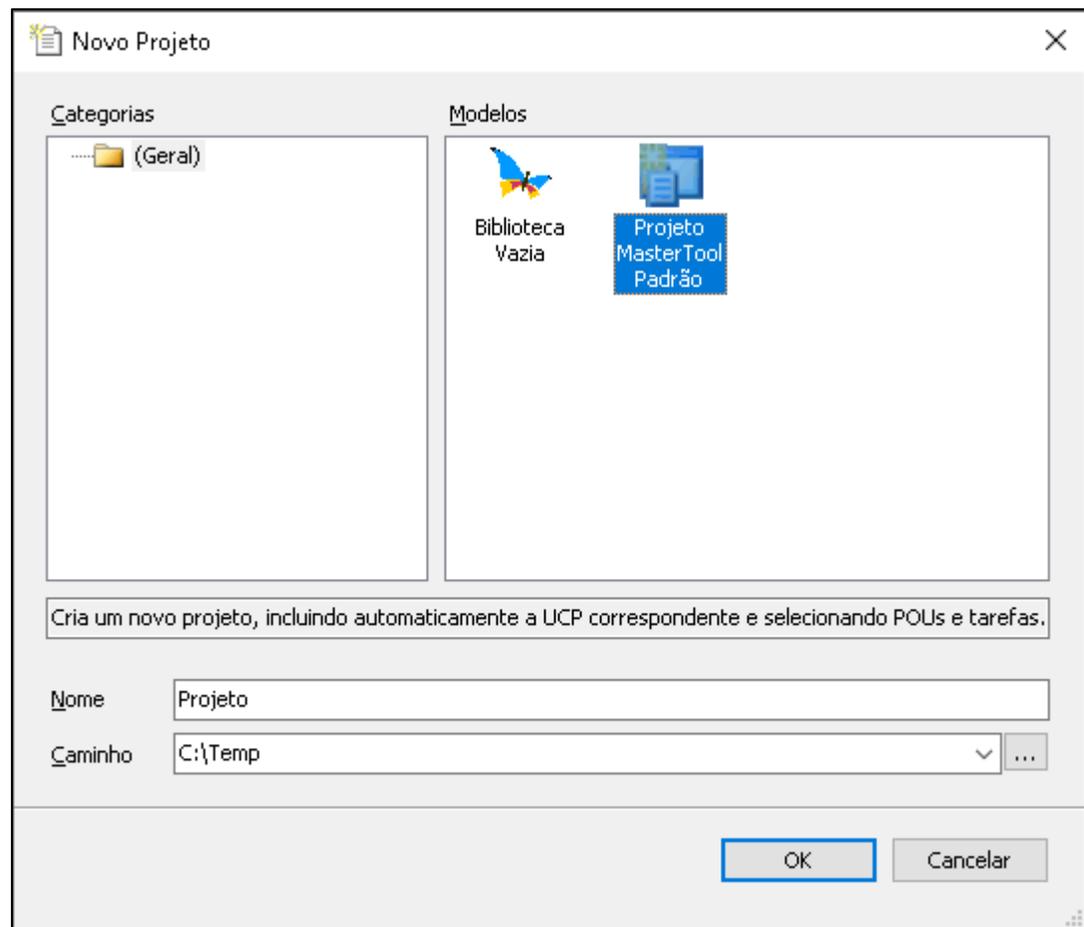


Figura 2-11. Classificação do Projeto

Escolhendo o modelo *Projeto MasterTool Padrão* será aberto um wizard para criação de projetos, onde o usuário deverá selecionar a UCP desejada, os módulos de hardware básicos que compõem o barramento, ou seja, o modelo de bastidor e de fonte de alimentação e a configuração de redundância. Nesse caso, será utilizada a UCP NX3030, um bastidor NX9000 e uma fonte de alimentação NX8000 e serão utilizadas as opções sem redundância de Half-Cluster e de expansão de barramento.

Projeto MasterTool IEC XE Padrão

Você está prestes a criar um novo Projeto MasterTool IEC XE Padrão. Selecione as seguintes opções e o assistente criará o projeto como você decidir.

Selecione o modelo do dispositivo:  
NX3010 (Altus S.A.) - CPU, 1 Eth., 2 Serial, Memory Card and Rack Expansion

Selecione o modelo do bastidor:  
NX9000 (Altus S.A.) - 8-Slot Backplane Rack

Selecione o modelo da fonte de alimentação:  
NX8000 (Altus S.A.) - 30 W 24 Vdc Power Supply Module

Selecione a configuração da redundância de Half-Cluster:

Selecione o modo de operação da redundância de Half-Cluster:

Selecione a configuração de comunicação OPC DA:  
Desabilitado

Selecione a configuração da redundância da expansão de barramento:  
Sem Redundância

Criar diretório para o projeto

< Anterior Próximo > Concluir Cancelar

Figura 2-12. Seleção dos Módulos de Hardware

O próximo passo é escolher as opções de rede. Nesta página é possível selecionar a quantidade de redes PROFIBUS e Ethernet. Também é possível definir se as redes de comunicação serão simples ou redundantes. No caso das redes Ethernet ainda é possível definir se estas irão gerar um chaveamento em caso de falha quando configurado com redundância de Half-Cluster. Neste exemplo não foi criada nenhuma rede além das interfaces de rede Ethernet da UCP.

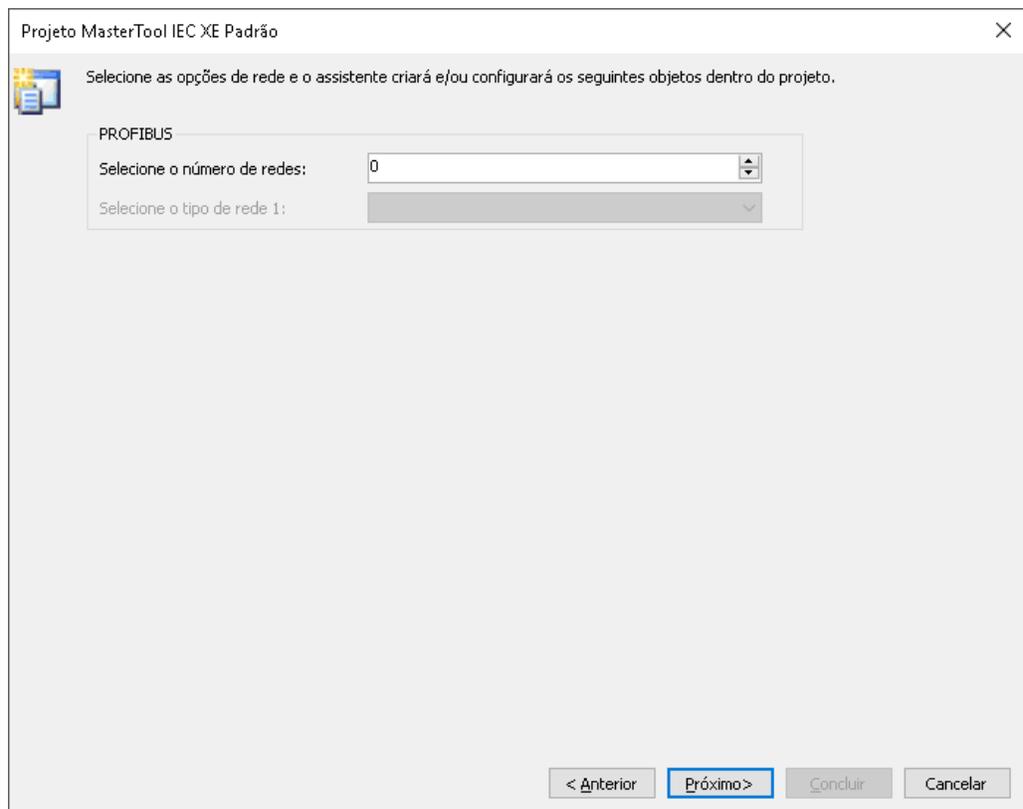


Figura 2-13. Opções de Rede

A página seguinte permite definir configurações de números de pontos de E/S que serão criados automaticamente com o projeto. Não é necessário conhecer os códigos dos produtos e basta inserir a quantidade de pontos da aplicação, pois o Wizard calcula a quantidade de módulo necessários e os adiciona. Neste exemplo não estão sendo criados pontos de E/S.

Projeto MasterTool IEC XE Padrão

Escolha a quantidade de pontos E/S que serão utilizados e o assistente criará os seguintes objetos dentro do projeto.

Escolha a solução para módulos E/S:

Módulos NX

Pontos de entrada digital: 0

Pontos de entrada analógica V/I: 0

Pontos de entrada analógica termopar: 0

Pontos de entrada analógica RTD: 0

Pontos de saída analógica V/I: 0

Pontos de saída digital a transistor: 0

Pontos de saída digital a relé: 0

Tipo de cabo de expansão:

Não Conectado

Quantidade de módulos E/S que serão utilizados: 0 de 128.

Quantidade de bastidores que serão utilizados: 1 de 9.

< Anterior Próximo > Concluir Cancelar

Figura 2-14. Configuração de Pontos de E/S

A seguir, o usuário deverá selecionar o perfil para o projeto, e a linguagem padrão para as POU's (programas). Nesse caso, será utilizado o perfil *Simples* (Single) e linguagem ST. Clicar em *Próximo* para prosseguir. A opção de cancelamento também está disponível nessa janela via botão *Cancelar*.

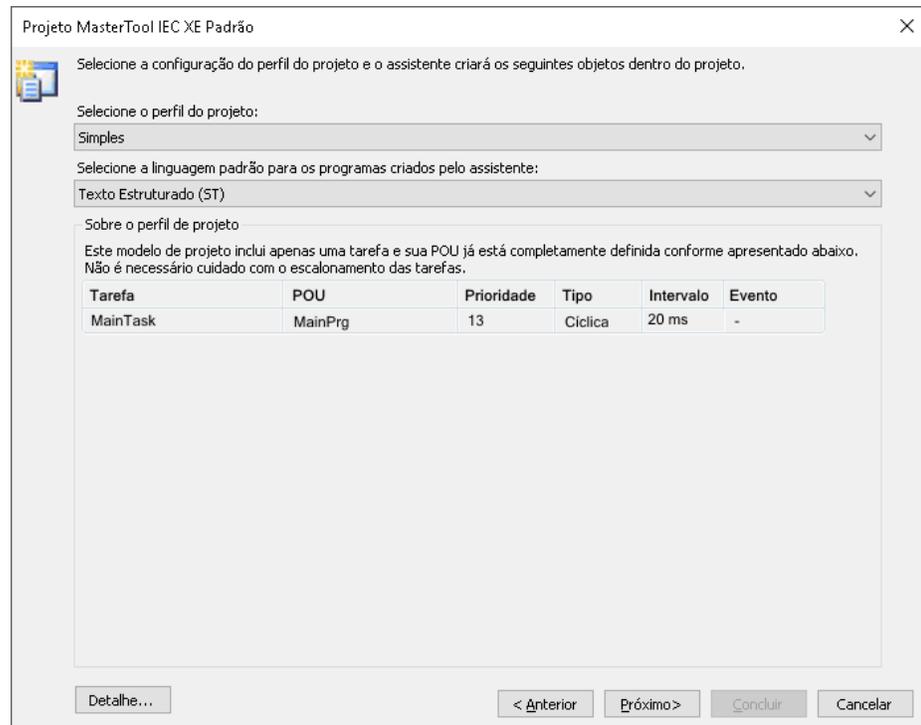


Figura 2-15. Seleção do Perfil

A próxima tela define a linguagem da POU criada pelo perfil selecionado. Como o perfil é *Simples*, existe apenas uma POU (MainPrg) e a linguagem ST foi mantida. Clicar em *Anterior* para voltar à tela antecedente, *Concluir* ou *Cancelar*.

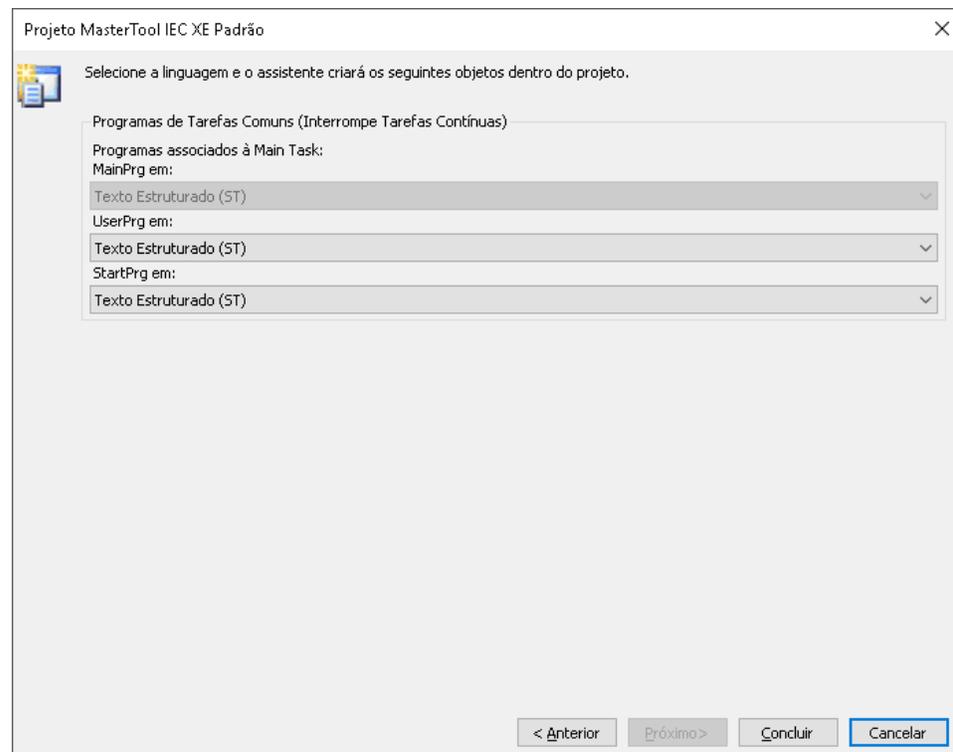


Figura 2-16. Linguagem de Programação

Ao pressionar o botão *Concluir*, o MasterTool IEC XE iniciará a criação do ambiente de desenvolvimento do projeto. Esse procedimento pode levar alguns segundos.

## Acrescentando Módulos

Por padrão, a UCP e os módulos de hardware selecionados na criação do projeto já são inseridos na configuração do sistema. Resta ao usuário incluir os outros módulos necessários.

Caso a aba *Biblioteca de Produto* não esteja disponível na tela do MasterTool IEC XE, a mesma deve ser incluída, através do menu *Visualizar*, clicando sobre o item *Biblioteca de Produtos*, conforme mostra a figura a seguir.

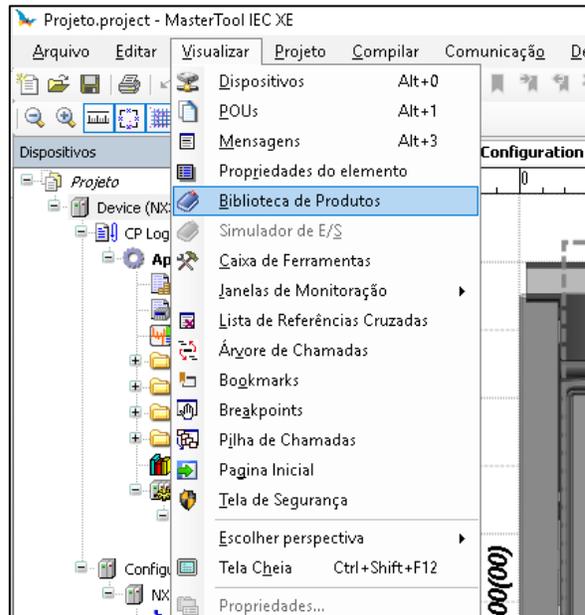


Figura 2-17. Visualizando a Biblioteca

Após abrir esta aba, basta selecionar o módulo a ser inserido no projeto e, mantendo pressionado o botão esquerdo do mouse, arrastá-lo para a área de configuração do barramento.

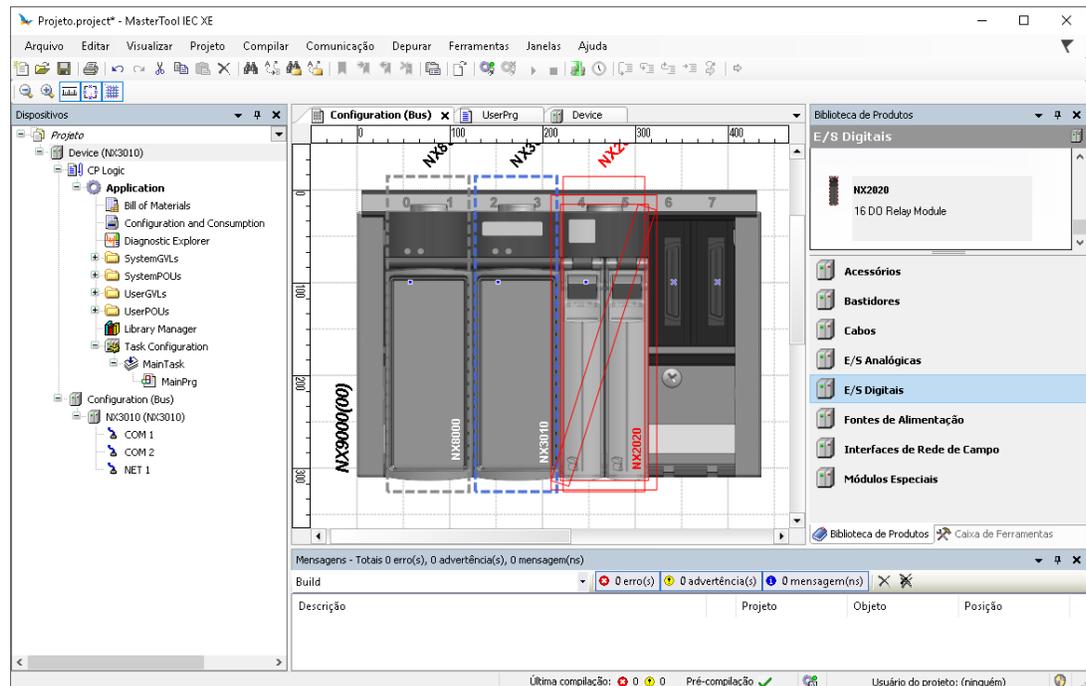


Figura 2-18. Acrescentando Módulos

## Criando POUs

Uma POU (Program Organization Unit, ou Unidade de Organização de Programa), é uma subdivisão do programa aplicativo que pode ser escrito em qualquer uma das linguagens disponíveis no software MasterTool IEC XE.

Com a criação do projeto através de um perfil selecionado, algumas POUs já são criadas, porém o usuário poderá criar quantas quiser, limitado pelo tamanho máximo da memória de programa.

Para inserir uma nova POU, basta clicar com o botão direito do mouse sobre Application (nome padrão criado para a aplicação), selecionar *Acrescentar Objeto* e, então, *POU...*, conforme mostra a figura a seguir.

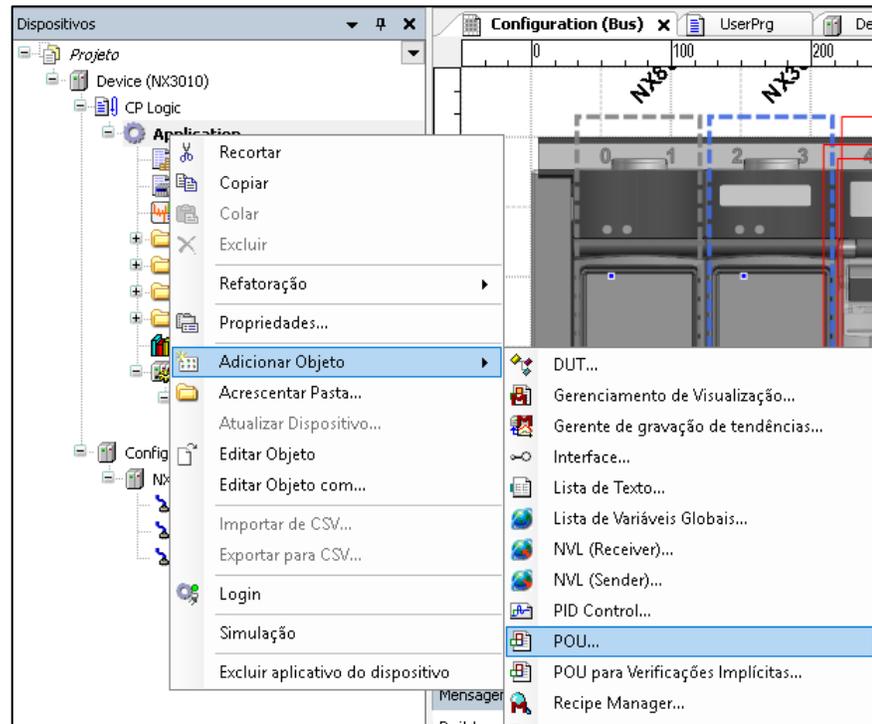


Figura 2-19. Inserindo POUs

Uma janela de configuração surgirá na tela, na qual o usuário deve colocar o nome da POU e selecionar o tipo e a linguagem que se deseja implementar. A seguir, deve clicar em *Abrir*.

Acrescentar POU

Criar uma nova POU (Unidade de Organização de Programa)

Nome  
POU

Tipo

Programa

Bloco funcional

Estender  ...

Implementar  ...

Final  Abstrato

Especificador de acesso  
▼

Linguagem de implementação do método  
Diagrama de Lógica Ladder (LD) ▼

Função

Tipo de retorno  ...

Linguagem de implementação  
Texto Estruturado (ST) ▼

Adicionar Cancelar

Figura 2-20. Classificando a POU

Para editar a POU basta selecionar a aba com o nome correspondente e iniciar o desenvolvimento da aplicação na linguagem escolhida anteriormente. O mesmo procedimento é válido para as POUs criadas automaticamente pelo perfil do projeto.

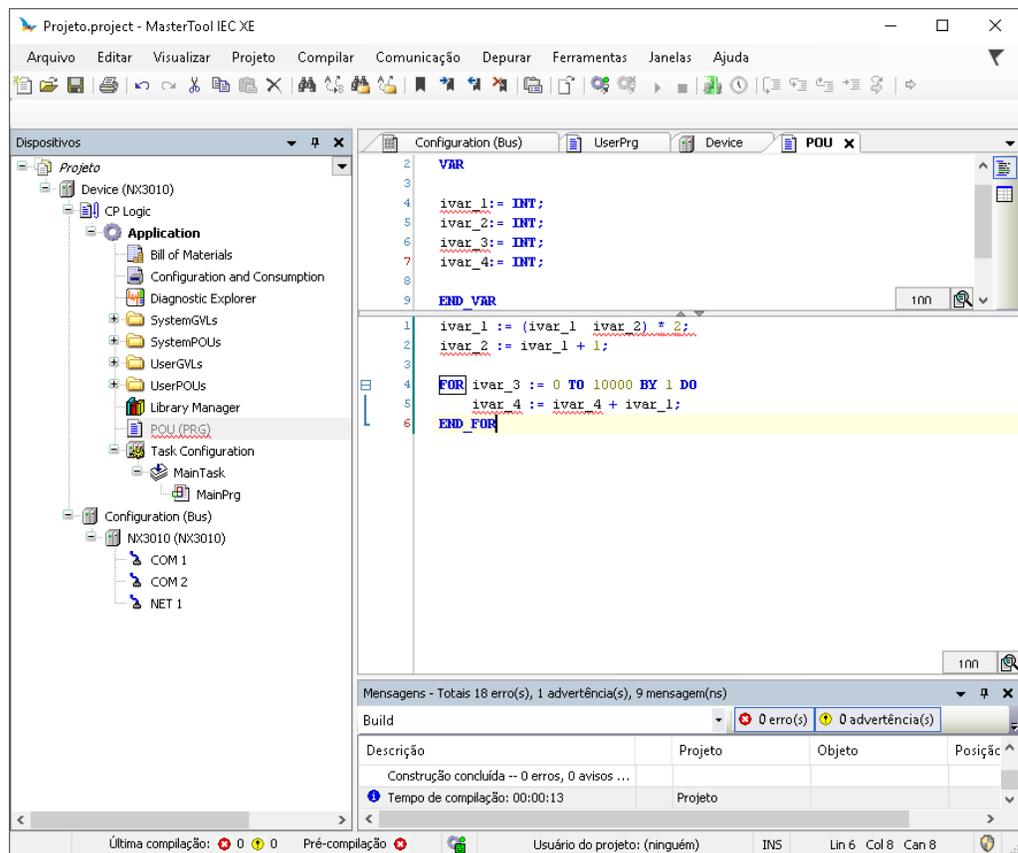


Figura 2-21. Editando a POU

## Criando Tarefas

Para que uma POU seja executada, ela deve estar vinculada a alguma tarefa. Esse mecanismo de escalonamento, denominado de Tarefa, é muito útil para sistemas de tempo real que definem a execução periódica ou em resposta a um evento (mudança de estado de alguma variável booleana). As tarefas controlam a execução de programas em diferentes taxas, em função das características da aplicação. A necessidade de executar programas em taxas diferentes tem por objetivo as exigências de tempo de resposta do processo sob controle e otimizar o uso da capacidade de processamento da UCP. Os controladores que fazem o uso de tarefas são denominados sistemas multitarefa.

Somente será permitida a criação de novas tarefas quando o perfil de projeto selecionado for o Personalizado (Custom), sendo que nos demais perfis as tarefas possíveis são criadas e configuradas automaticamente.

Desta forma, para incluir uma nova tarefa (caso o perfil selecionado permita), basta clicar com o botão direito do mouse sobre o objeto *Task Configuration*, selecionar *Acrescentar Objeto* e então *Tarefa...*, conforme mostra a figura a seguir.

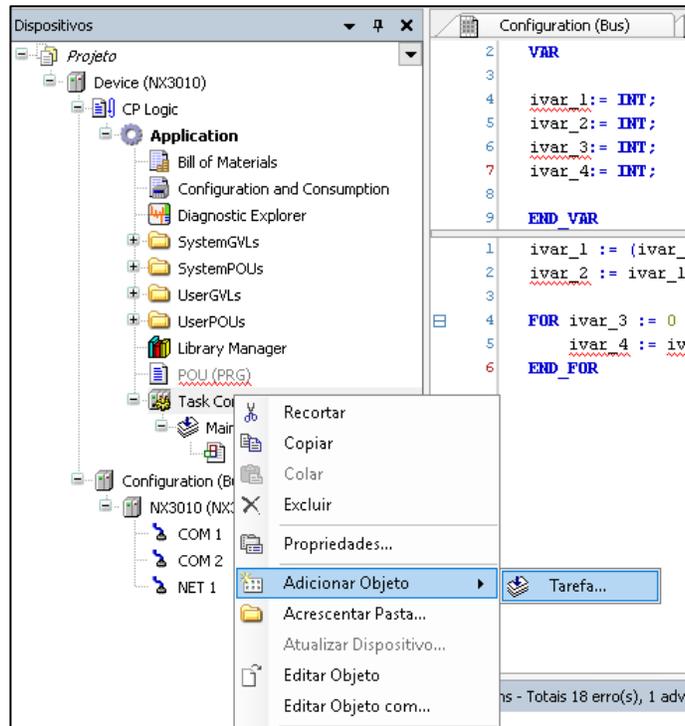


Figura 2-22. Criando uma Tarefa

Em seguida, surgirá uma tela para colocar o nome da tarefa. Após, clicar em *Abrir* para finalizar a criação da mesma.

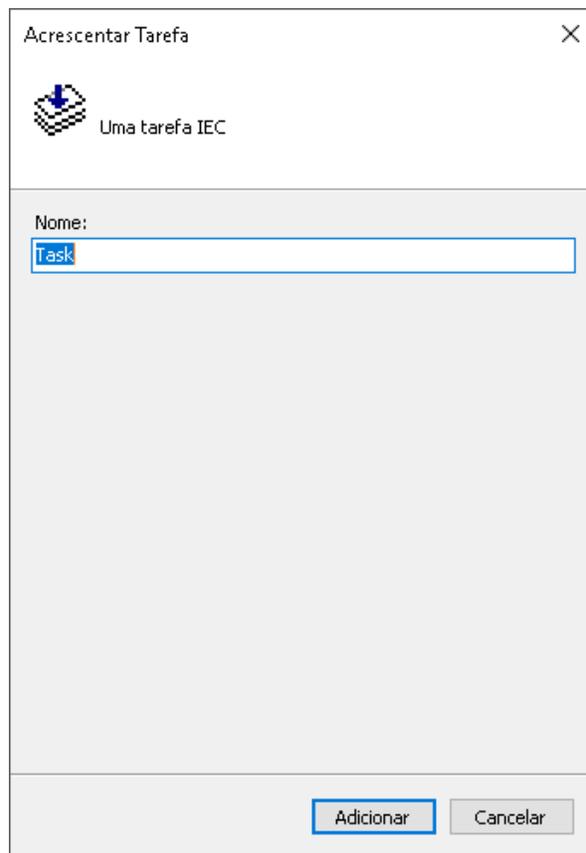


Figura 2-23. Nomeando a Tarefa

## Configurando Tarefas

Após abrir a tarefa, a janela de configuração surgirá para que o usuário defina e classifique o funcionamento da mesma.

O campo *Prioridade* (0..31) estabelece a prioridade em que a tarefa será executada na aplicação, sendo o 0 (zero) mais prioritário. Por exemplo, a *MainTask*, criada na maioria dos perfis de projeto, tem prioridade 13, ou seja, essa tarefa é considerada prioritária para o sistema.

O campo *Tipo* define qual o tipo e o método para a tarefa ser executada, sendo que podem ser selecionados os seguintes itens:

- *Cíclica*: a tarefa é executada ciclicamente, ou seja, é chamada a cada intervalo de tempo configurado no campo ao lado. Ex: t#20ms.
- *Evento*: a tarefa é executada quando a variável do tipo BOOL, configurada no campo ao lado, recebe uma borda de subida, ou seja, a variável passe de FALSE para TRUE.
- *Externa*: a tarefa é executada quando uma interrupção externa ocorre, a qual é configurada no campo ao lado.
- *Contínua*: a tarefa sempre é executada, de acordo com a sua prioridade, ou seja, tarefas com maior prioridade são executadas primeiramente.
- *Status*: a tarefa é executada quando a variável do tipo BOOL, configurada no campo ao lado, for verdadeira.

Além dos campos mencionados acima, ainda deve ser configurado o Intervalo (apenas para tarefas cíclicas), ou seja, o intervalo de tempo em que a tarefa é chamada para executar. O tempo máximo para a *MainTask* nos perfis Simples, Básico, Normal, Experiente e Personalizado é igual

a 750 ms e o tempo mínimo igual a 5 ms. No Perfil de Máquina o tempo máximo para a MainTask é igual a 100 ms e o tempo mínimo 5 ms. Recomenda-se configurar o intervalo da tarefa de no mínimo duas vezes o tempo de ciclo (execução) da mesma.

O cão-de-guarda da UCP é configurado para evitar o travamento das tarefas de usuário. O campo *Tempo*, define o tempo máximo permitido para a execução da tarefa. Caso a tarefa leve um tempo maior que o tempo de cão-de-guarda para ser executada, a aplicação irá para STOP e entrará em exceção por cão-de-guarda. O campo *Sensibilidade* refere-se a quantas vezes o tempo do cão-de-guarda deverá ser atingido para que o diagnóstico seja ativado. Caso o tempo de execução da tarefa atinja o valor do campo Sensibilidade multiplicado pelo campo Tempo, o diagnóstico também será indicado. Deve-se atentar ao fato de que o cão-de-guarda da UCP não é utilizado para proteger a aplicação do usuário de picos no tempo de execução e sim de travamentos. Portanto, seu tempo deve ser configurado com um valor alto, se comparado ao tempo de execução da tarefa a que está relacionado. O ideal é manter o tempo médio de execução das tarefas em, no máximo, 50% do tempo de cão-de-guarda. Assim, diminuem as chances de ocorrerem erros de cão-de-guarda por eventuais picos de tempo na execução da tarefa.

Visando proteger o sistema quanto a possíveis erros de configuração, o MasterTool IEC XE verifica em todas as tarefas cíclicas, durante a compilação, o cão-de-guarda (Cão-de-Guarda de Software) e os limites mínimo e máximo do intervalo da tarefa. É importante destacar que o usuário deverá ter cuidado ao alterar os valores pré-definidos pelos perfis de projeto, uma vez que alterações indevidas podem colocar em risco a execução do sistema. Então, recomenda-se utilizar os valores padrão.

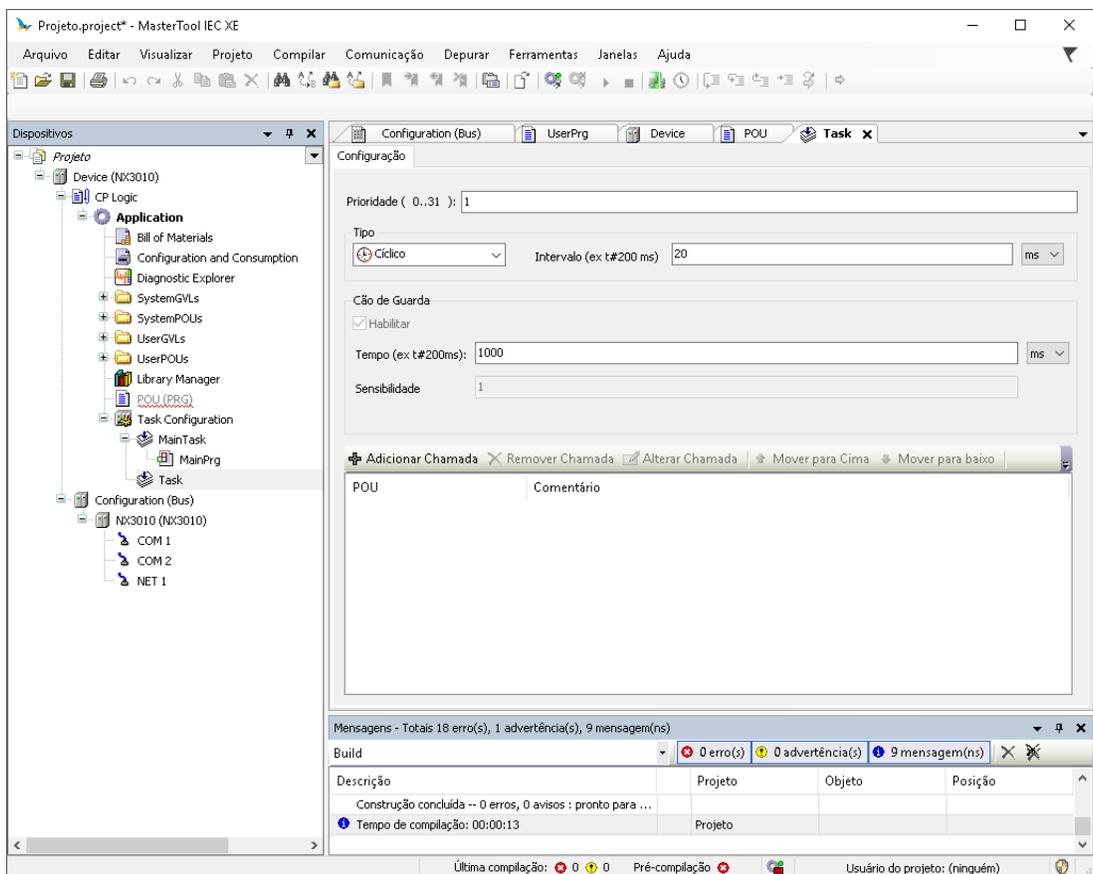


Figura 2-24. Configurando a Tarefa Criada

Abaixo, a tabela exibe as verificações realizadas pelo MasterTool IEC XE para a configuração do campo Intervalo das tarefas cíclicas, quando o campo Sensibilidade é igual a 1. Para o Perfil Personalizado não é feita consistência no intervalo da tarefa e no tempo de cão-de-guarda.

Tabela 2-2. Limites das tarefas

Tarefa	Tipo	Intervalo Mínimo	Intervalo Máximo
MainTask	Cíclica	5 ms	750 ms
CyclicTask	Cíclica	5 ms	2147483 ms
TimeInterruptTask00	Cíclica	500 us	2147483 ms

## Vinculando uma POU a uma Tarefa

Conforme mencionado anteriormente, para que uma POU seja executada na aplicação, ela deve estar associada a uma tarefa. Nos perfis de projeto (sem considerar o Personalizado), as POUs já estão associadas as suas respectivas tarefas. Porém, caso seja utilizado o perfil Personalizado (Custom) ou caso novas POUs sejam criadas, as mesmas devem ser vinculadas a tarefas.

Para associar uma POU criada basta acessar a tarefa desejada clicando duas vezes sobre a mesma na árvore de dispositivos e, em seguida, clicar sobre *Acréscenar*. Após, surgirá uma tela denominada *Assistente de Entrada*, na qual deverá ser selecionada a POU desejada, conforme mostra a figura a seguir.

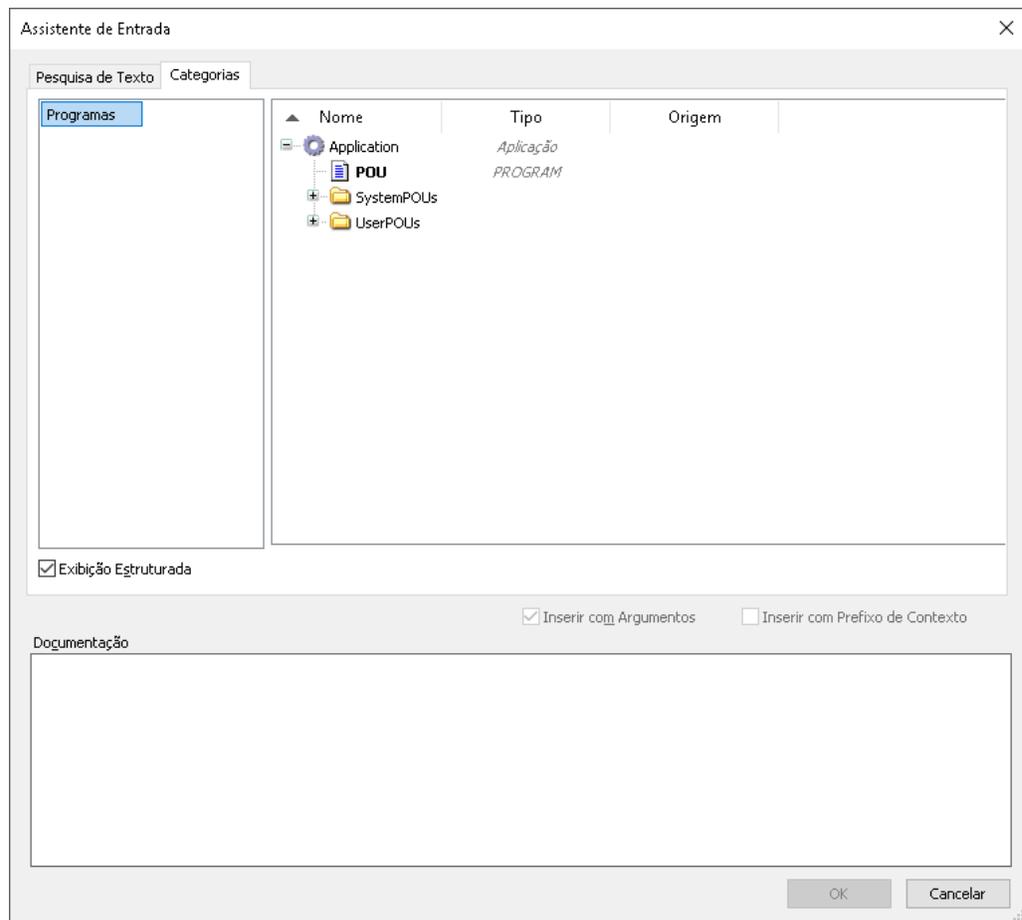


Figura 2-25. Vinculando POUs às Tarefas

O número máximo de tarefas que o usuário poderá criar somente está definido para o perfil Personalizado, ou seja, o único que tem essa permissão. Os demais já têm as suas tarefas criadas e configuradas. Para informações adicionais sobre a quantidade máxima

de tarefas IEC por UCP e perfil de projeto, consulte o Manual de Utilização UCPs Série Nexto (MU214100).

## Configurando a UCP

A configuração da UCP Nexto baseia-se em estruturar a área de diagnósticos, a área de memória retentiva e persistente, modo de troca a quente, entre outros parâmetros.

O usuário deverá dar dois cliques no ícone da UCP Nexto (localizado na árvore de dispositivos) e configurar os campos conforme descrito no capítulo anterior deste manual e no Manual de Utilização UCPs Série Nexto – MU214100, capítulo Configuração.

## Bibliotecas

Existem diversos recursos da ferramenta de programação disponibilizados através de bibliotecas. Sendo assim, os mesmos devem ser inseridos no projeto para que a sua utilização seja possível. O procedimento de inserção é bastante simples, sendo necessário selecionar o item *Library Manager*, disponível no menu da esquerda e selecionar *Acrescentar Biblioteca*, conforme mostra a figura a seguir.

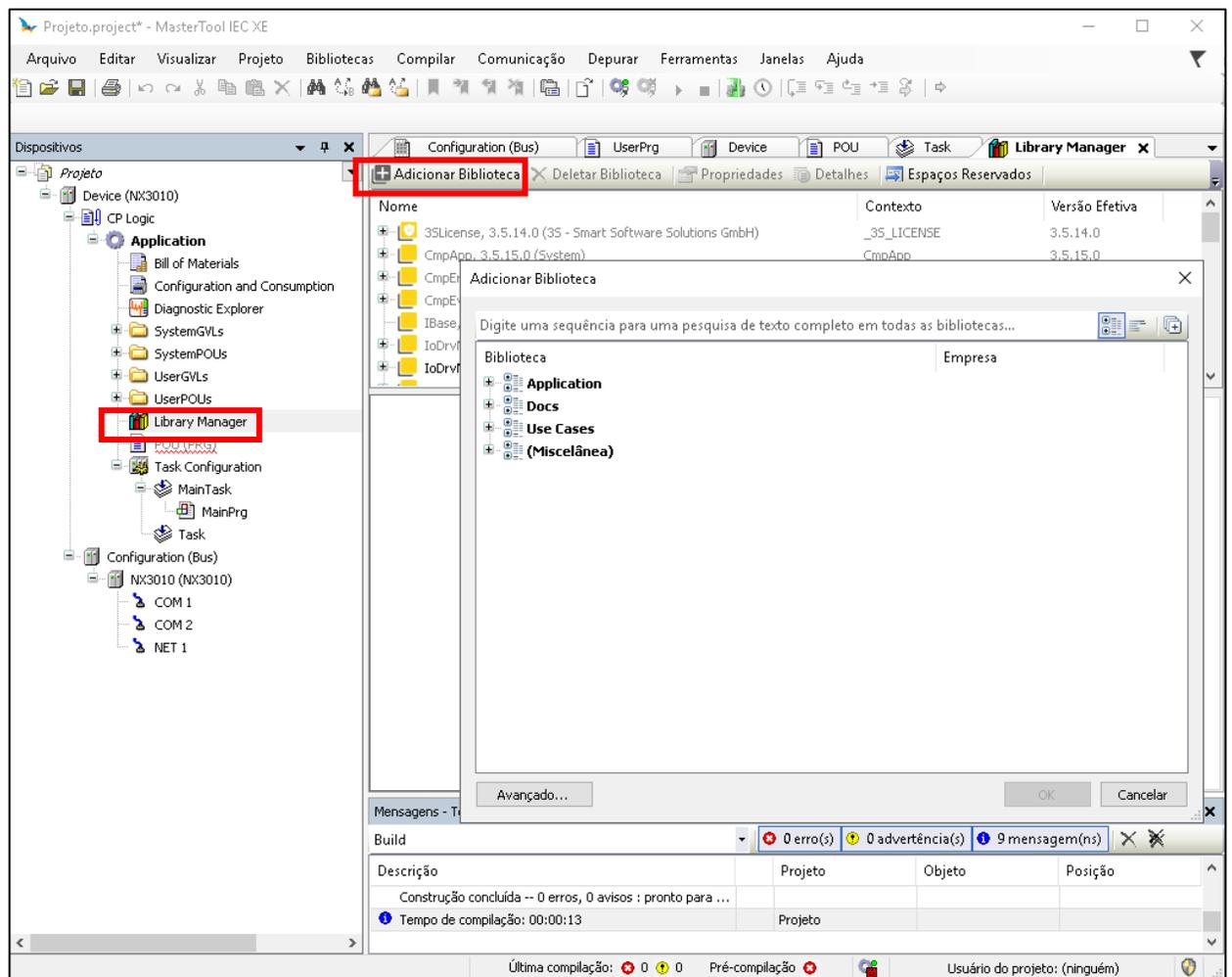


Figura 2-26. Inclusão de uma Biblioteca no Projeto

Na sequência, deve-se selecionar a biblioteca desejada para inclusão no projeto, pressionando em seguida, o botão *OK*.

**NOTA:** Dependendo das opções selecionadas nas opções de características esse diálogo pode ter mais ou menos opções.

## Inserindo uma Instância de Protocolo

As UCPs da Série Nexto disponibilizam protocolos, como o MODBUS. Basta adicionar e configurar a instância do protocolo desejado na interface de comunicação.

Na sequência, são descritos dois casos de inserção do protocolo MODBUS, sendo um na interface serial e outro na interface Ethernet.

### Inserção de uma instância do protocolo MODBUS RTU Escravo

O primeiro passo para configurar o MODBUS RTU, em modo escravo, é incluir a instância na COM desejada (nesse caso, COM 1). Clicar com o botão direito do mouse sobre a COM e selecionar *Acrescentar Dispositivo...*, conforme mostra a figura a seguir.

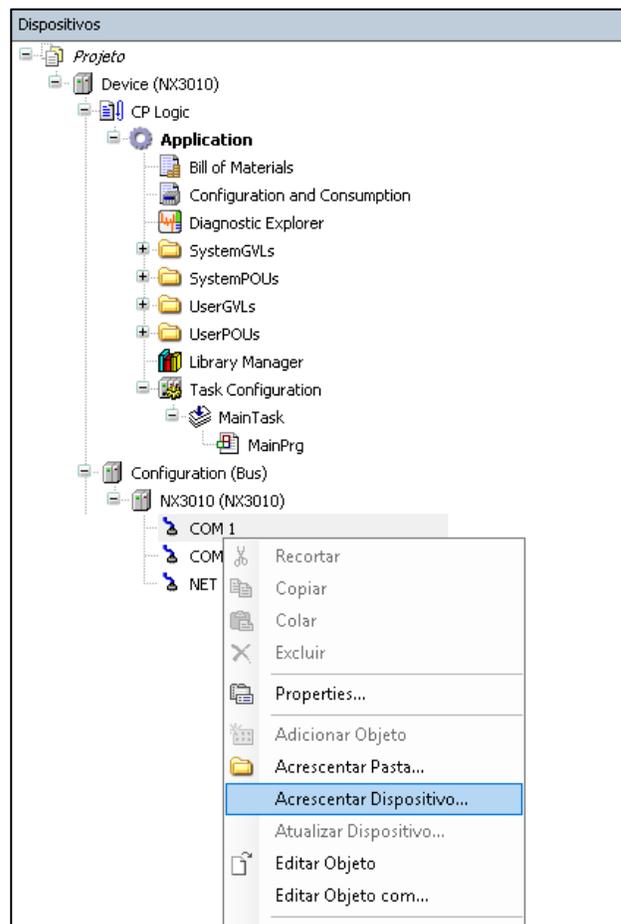


Figura 2-27. Adicionando a Instância

Após, surgirão na tela os protocolos disponíveis ao usuário. Nesse caso, selecionar o *MODBUS RTU Slave* e clicar em *Acrescentar*, conforme mostra a figura a seguir.

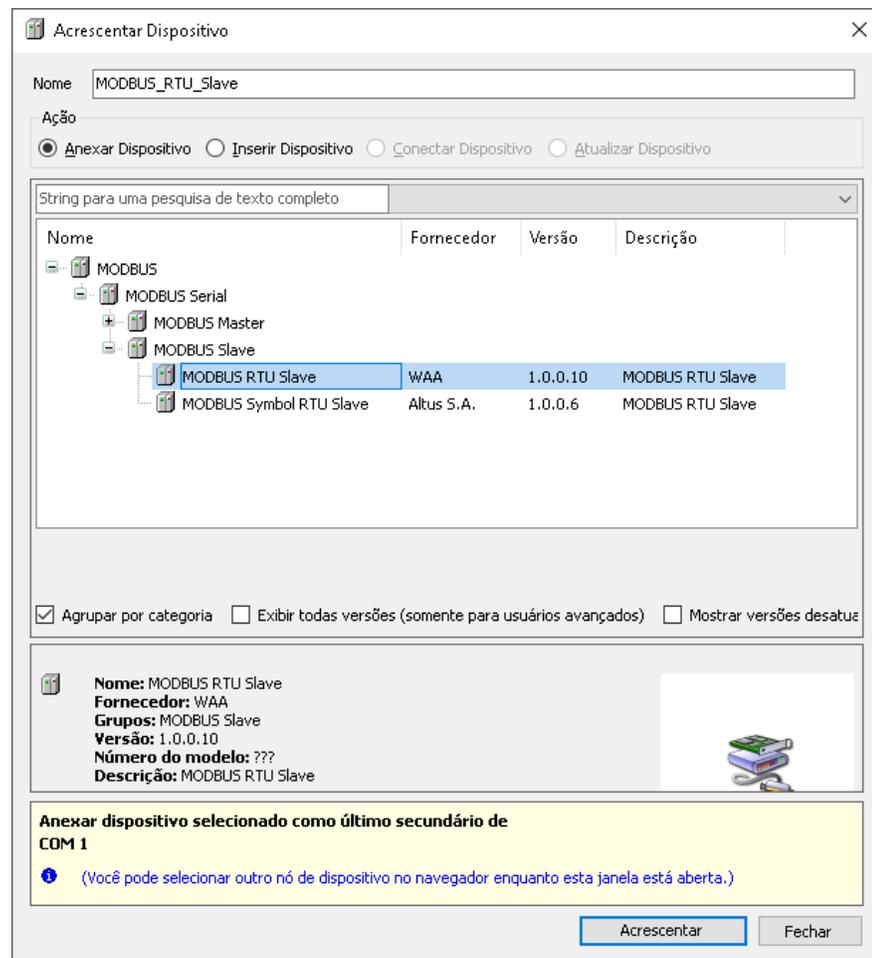


Figura 2-28. Selecionando o Protocolo

Há dois modos de configuração para este protocolo. Um faz uso de representação direta (%Q), no qual as variáveis são definidas pelo seu endereço. O outro, através de mapeamento simbólico, tem as variáveis definidas pelo seu nome.

Desta forma, como alternativa à configuração MODBUS RTU Slave por Representação Direta (%Q) mostrada anteriormente, pode-se selecionar a opção MODBUS Symbol RTU Slave, para configuração por Mapeamento Simbólico. A seguir, clicar em Acrescentar, conforme mostrado na sequência.

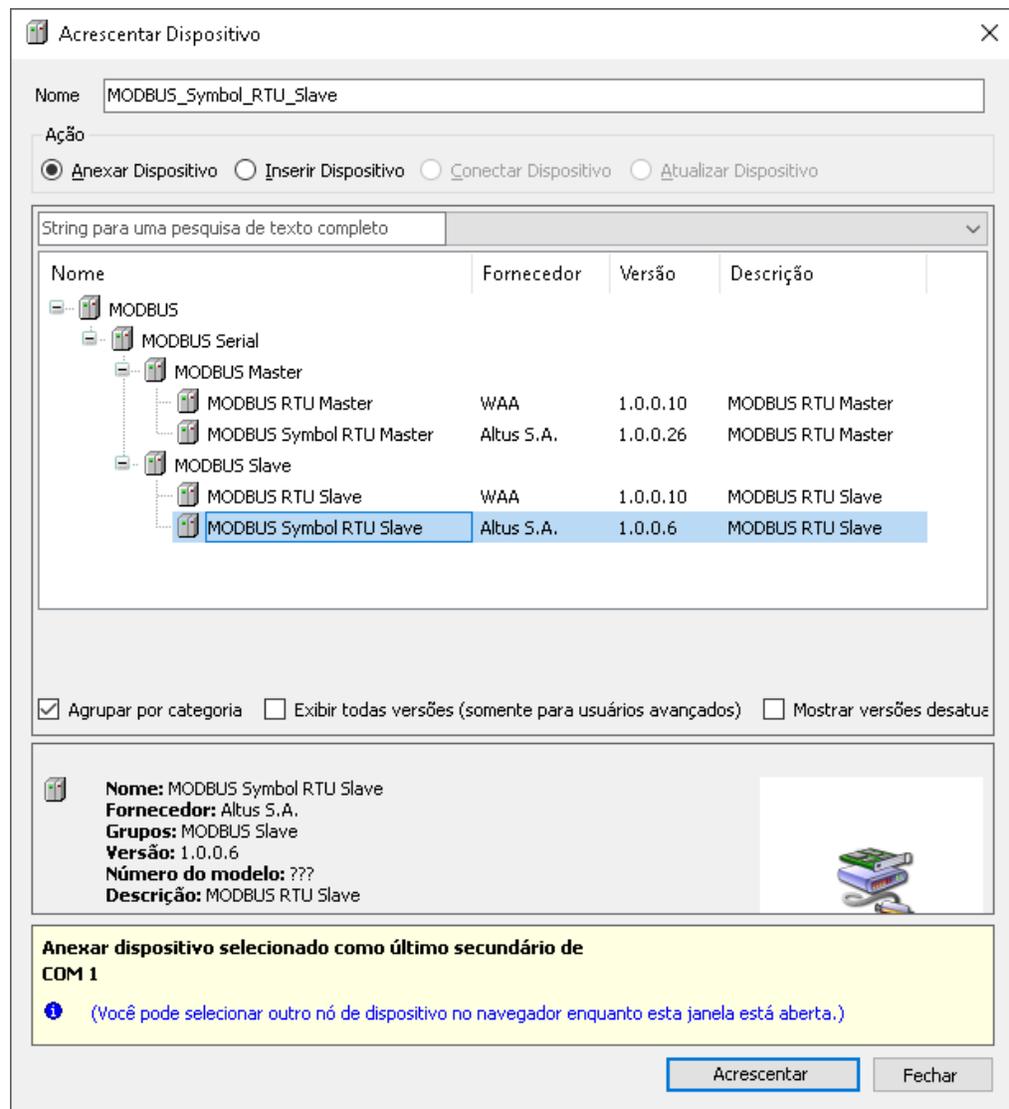


Figura 2-29. Seleção do protocolo MODBUS Symbol RTU Slave

### Configuração do Protocolo MODBUS Mestre

O Nexto também permite a configuração do protocolo MODBUS Mestre (a figura anterior ilustra essa possibilidade) nas opções de mapeamento simbólico e por representação direta (%Q). Essa configuração é abordada em detalhes no Módulo de Treinamento CP Nexto Avançado ALTUS MP10A, ou mesmo na documentação do produto.

### Inserção de uma instância do protocolo MODBUS Ethernet Cliente

O primeiro passo para configurar o MODBUS Ethernet, em modo cliente, é incluir a instância na NET desejada (nesse caso NET 1, pois a UCP NX3010 possui uma interface Ethernet). Clicar com o botão direito do mouse sobre a NET e selecionar *Acrescentar Dispositivo...*, conforme mostra a figura a seguir.

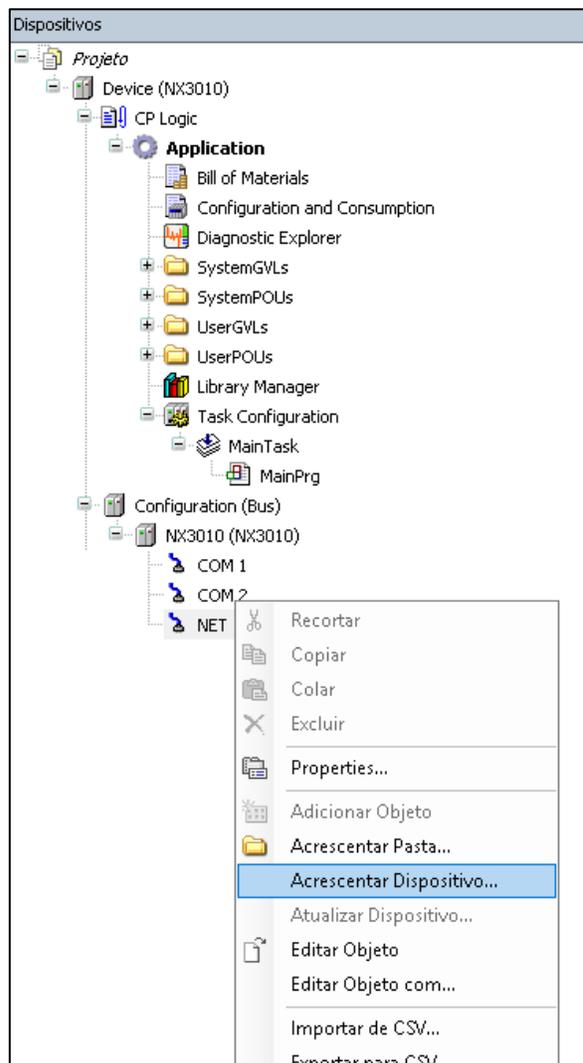


Figura 2-30. Adicionando a Instância

Após, surgirão na tela os protocolos disponíveis ao usuário. Nesse caso, selecionar o *MODBUS Client* e clicar em *Acrescentar*, conforme mostra a figura a seguir.

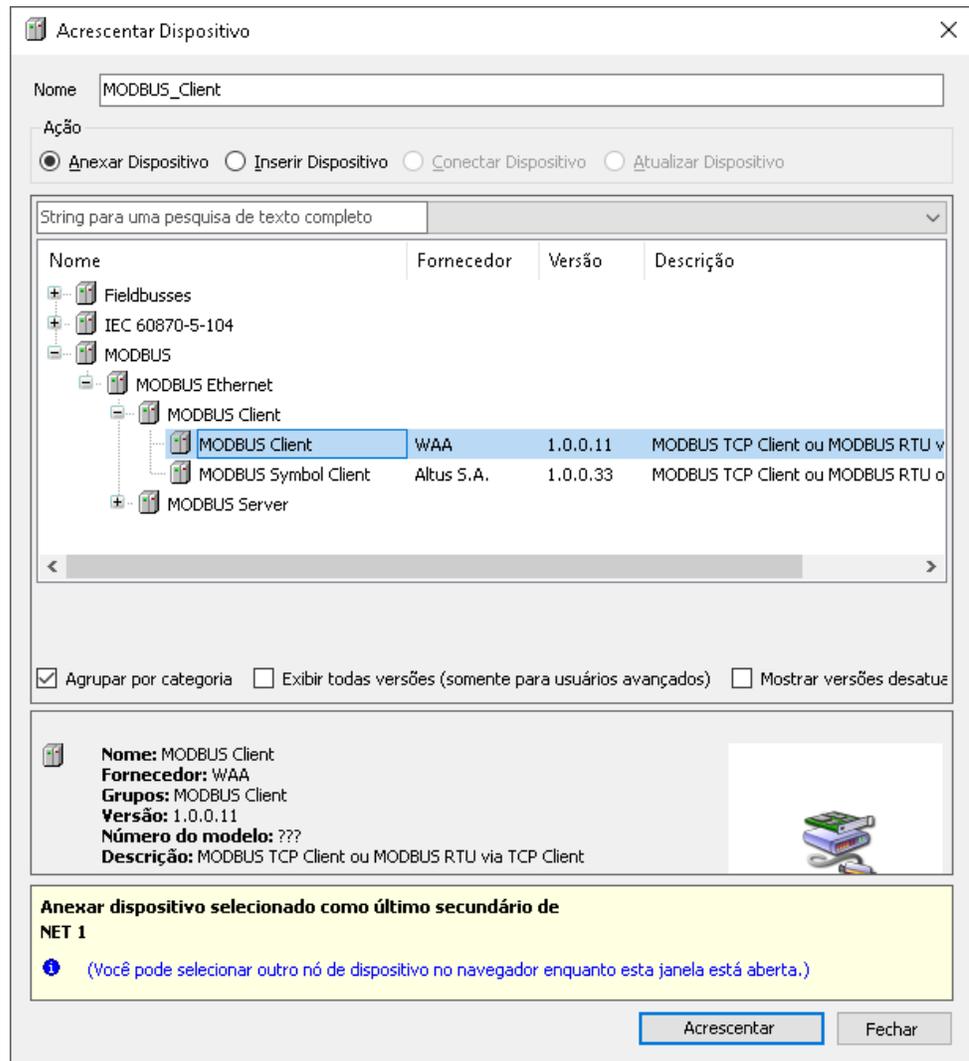


Figura 2-31. Selecionando o Protocolo

Como alternativa à configuração MODBUS Client por Representação Direta (%Q) mostrada anteriormente, pode-se selecionar a opção MODBUS Symbol Client, para configuração por Mapeamento Simbólico conforme ilustrado na figura a seguir.

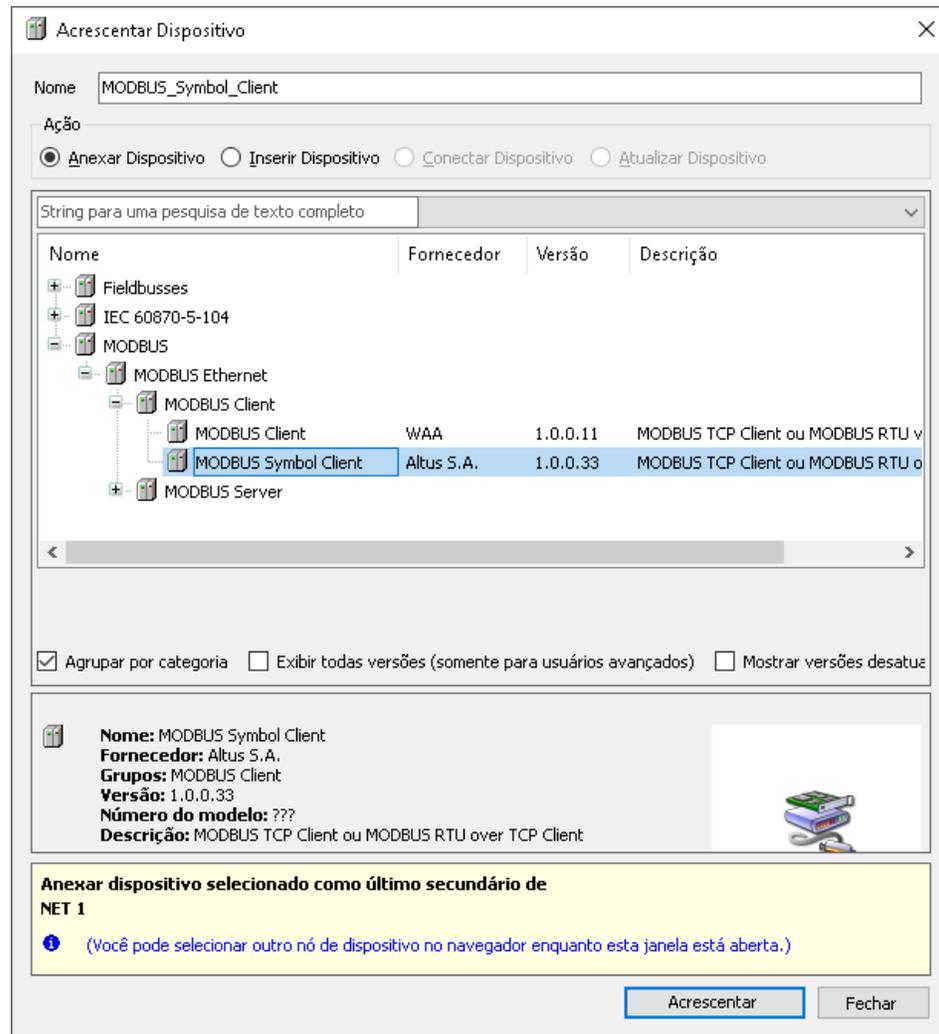


Figura 2-32. Selecionando o Protocolo MODBUS Ethernet

### Configuração do Protocolo MODBUS Ethernet Servidor

O Nexto também permite a configuração do protocolo MODBUS Ethernet Servidor (a figura anterior ilustra essa possibilidade) nas opções de mapeamento simbólico e por representação direta (%Q). Essa configuração é abordada em detalhes no Módulo de Treinamento CP Nexto Avançado ALTUS MP10A, ou mesmo na documentação do produto.

### Protocolo OPC DA na Série Nexto

Para comunicar com as UCPs da Série Nexto é possível utilizar a tecnologia OPC DA (Open Platform Communications Data Access). Esta plataforma de comunicação aberta foi desenvolvida para ser o padrão utilizado nas comunicações industriais. Baseado na arquitetura cliente/servidor, oferece inúmeras vantagens no desenvolvimento de projeto e facilidades na comunicação com os sistemas de automação.

O Servidor OPC DA é instalado juntamente com a instalação do MasterTool IEC XE e sua configuração é realizada dentro da ferramenta. Vale salientar que o OPC DA está disponível somente nas interfaces Ethernet locais das UCPs Nexto. Os módulos de expansão Ethernet não suportam essa funcionalidade.

Após a configuração do Servidor OPC DA os dados disponíveis em todos os CPs podem ser acessados via um Cliente OPC DA. Na configuração do Cliente OPC DA deve ser selecionado o nome do Servidor OPC DA correto. Neste caso o nome é CoDeSys.OPC.DA.

O protocolo OPC DA na Série Nexto é abordado em detalhes no Módulo de Treinamento CP Nexto Avançado ALTUS MP10A, ou mesmo na documentação do produto.

#### Protocolo OPC UA na Série Nexto

O protocolo OPC UA é uma evolução da família OPC. Independente de plataforma, foi concebido para ser o novo padrão utilizado nas comunicações industriais.

Baseado na arquitetura cliente/servidor, o protocolo OPC UA oferece inúmeras vantagens no desenvolvimento de projeto e facilidades na comunicação com os sistemas de automação.

Quando se trata do desenvolvimento de projetos, configurar a comunicação e trocar informações entre os sistemas é extremamente simples utilizando tecnologia OPC UA. Utilizando outros drivers, baseados em endereços, é necessário criar tabelas para relacionar as tags do sistema de supervisão e as variáveis do controlador programável. Quando as áreas de dados são alteradas, no decorrer do desenvolvimento do projeto, é necessário refazer os mapeamentos e novas tabelas com as relações entre as informações do CP com o sistema SCADA.

Após a configuração do Servidor OPC UA os dados disponíveis em todos os CPs podem ser acessados via um Cliente OPC UA. Na configuração do Cliente OPC UA deve ser selecionado o endereço do Servidor OPC UA correto. Neste caso o endereço `opc.tcp://endereço-ip-do-dispositivo:4840`.

O protocolo OPC UA na Série Nexto é abordado em detalhes no Módulo de Treinamento CP Nexto Avançado ALTUS MP10A, ou mesmo na documentação do produto.

#### Compilando um Projeto

A fim de realizar a verificação da aplicação criada, o usuário deve executar a compilação do projeto. Essa é a forma mais eficaz de se encontrar problemas ou receber avisos sobre alguns equívocos cometidos durante a configuração do produto e edição da aplicação. Para executar tal procedimento, basta acessar o menu *Compilar* e clicar na opção *Gerar Código*.

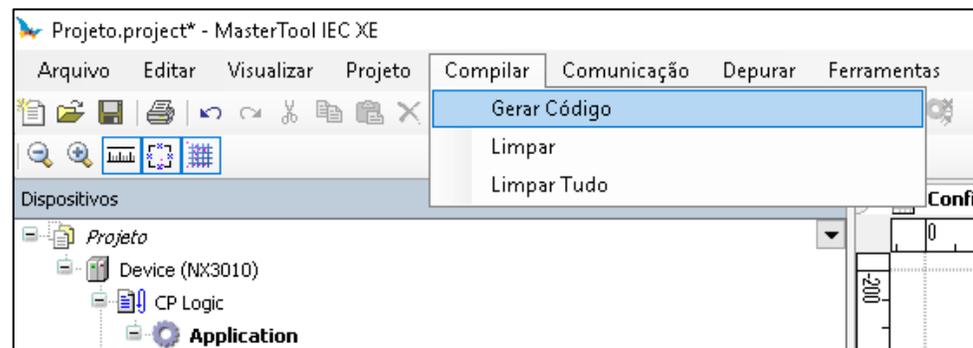


Figura 2-33. Compilando o Projeto

Depois do tempo de processamento, o qual vai variar de acordo com o tamanho da aplicação do usuário, os erros e mensagens de alerta, caso sejam necessários, serão mostrados conforme mostra a figura a seguir.

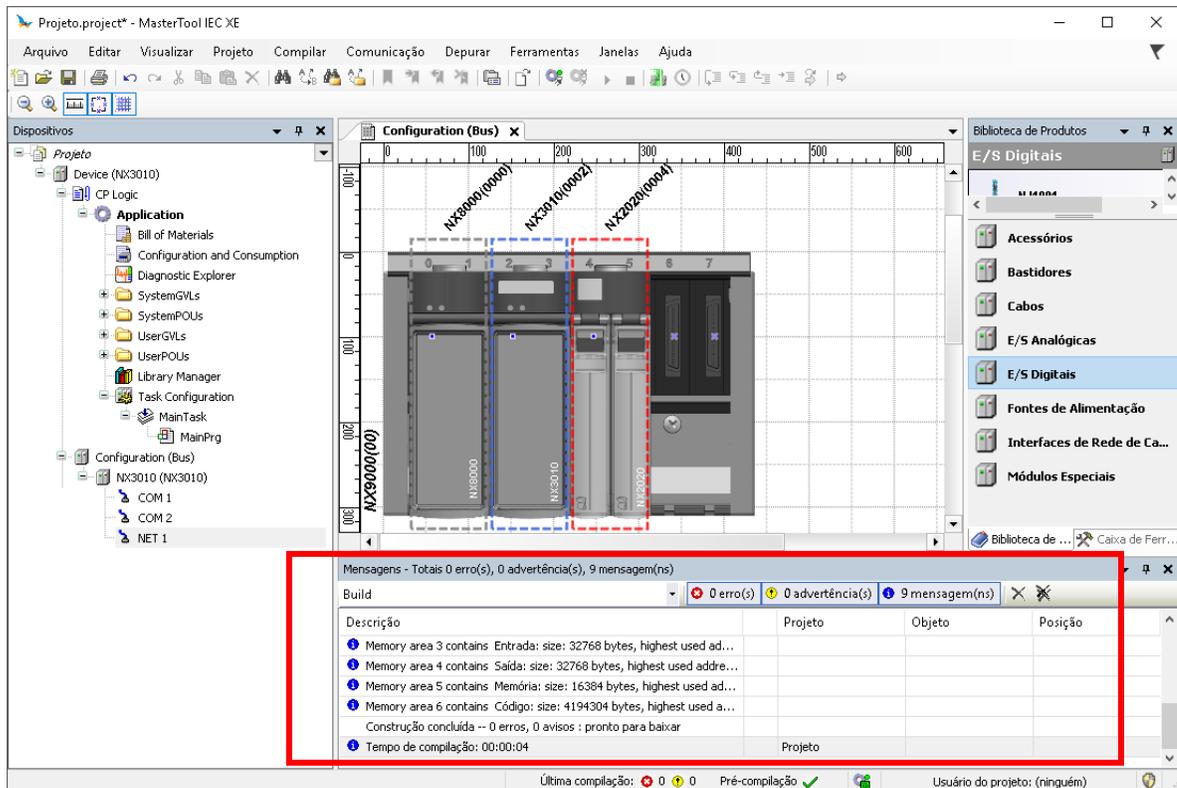


Figura 2-34. Mensagens da Compilação

Caso os erros e mensagens não estejam visíveis na tela, basta acessar o menu *Visualizar* e clicar em *Mensagens*, conforme mostrado na figura a seguir.

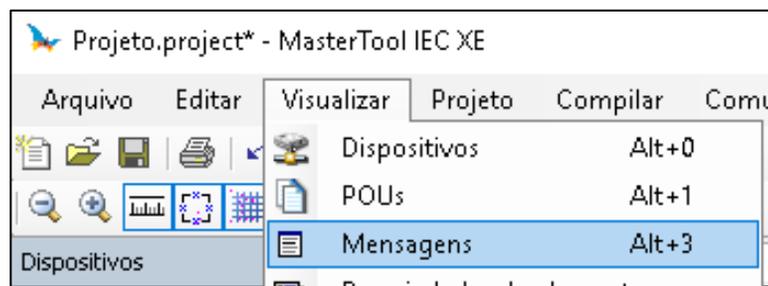


Figura 2-35. Incluindo as Mensagens na Tela

### Simulando um Projeto

O MasterTool IEC XE possui um importante recurso de simulação que permite ao usuário testar sua aplicação sem a utilização do equipamento, conferindo maior agilidade no desenvolvimento do programa. No entanto, alguns recursos específicos, que dependem do hardware das UCPs Nexto, não são possíveis de serem simulados.

Seguem abaixo, os recursos indisponíveis no modo de simulação:

- Relógio RTC
- Varredura do Barramento
- Módulos de E/S

- Interrupção de Barramento
- Portas Seriais
- Comunicação Ethernet
- Protocolos de comunicação como Modbus
- Interfaces PROFIBUS
- Escraves PROFIBUS
- Operações em cartão de memória
- Diagnósticos em variáveis
- Diagnostics Explorer
- Outras funções que acessem o hardware do CP

Por esta razão o modo simulação deve ser utilizado para testar a lógica da aplicação no que não depender de funções de acesso ao hardware. Estes recursos devem ser testados com o hardware para garantir o funcionamento da aplicação neste sentido.

Para alterar o MasterTool IEC XE para Modo Simulação é necessário selecionar esta opção no Menu Comunicação conforme figura a seguir. Após isso é exibido um aviso no barra inferior do MasterTool IEC XE que indica que a ferramenta está operando em Modo Simulação.

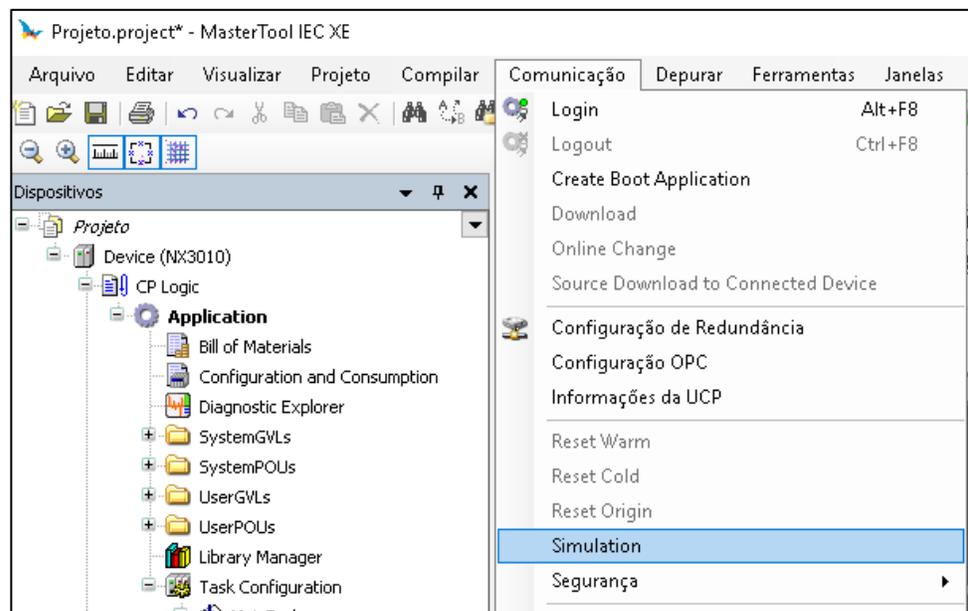


Figura 2-36. Modo Simulação

Em Modo Simulação a aplicação é executada em um dispositivo virtual no computador onde está instalado o MasterTool IEC XE. Por esta razão algumas características apresentadas estão relacionadas à arquitetura de hardware do computador e não das UCPs da Série Nexto. A principal característica neste sentido está relacionada ao formato dos dados nas áreas de memória de representação direta. O Modo Simulação trabalha com o formato little endian onde o primeiro endereço de memória é o menos significativo do dado. Por outro lado as UCPs da Série Nexto trabalham com o formato big endian onde o primeiro endereço de memória é o mais significativo do dado.

Neste caso um mesmo dado escrito por exemplo em %QD0, será escrito de forma diferente na simulação e nas UCPs da Série Nexto. Se o dado escrito for 16#1234ABDC a distribuição dos dados na memória do CP ficará da seguinte maneira:

```
%QW0 = 16#1234  
%QW2 = 16#ABCD  
%QB0 = 16#12  
%QB1 = 16#34  
%QB2 = 16#AB  
%QB3 = 16#CD
```

Para o mesmo dado escrito em %QD0 no Modo Simulação a distribuição dos dados na memória ficarão da seguinte maneira:

```
%QW0 = 16#ABCD  
%QW2 = 16#1234  
%QB0 = 16#CD  
%QB1 = 16#AB  
%QB2 = 16#34  
%QB3 = 16#12
```

Tendo em vista estas diferenças e para facilitar o desenvolvimento da aplicação, utilizando os recursos do MasterTool IEC XE e das UCPs da Série Nexto é recomendado o uso de variáveis simbólicas. Neste caso as diferenças entre o Modo Simulação e o comportamento com as UCPs da Série Nexto não são verificadas. Portanto a melhor prática é evitar o uso de variáveis de representação direta sempre que possível para evitar o retrabalho quando desenvolver uma lógica que será testada em simulação e depois carregada em uma UCP.

O modo Simulação pode ser utilizado para simular um projeto redundante, porém, terá as mesmas limitações citadas anteriormente, podendo ser testada apenas a lógica que não dependa do hardware. Neste caso, sempre serão executadas as POUs NonSkippedPrg e ActivePrg, como se o CP simulado fosse o CP Ativo.

Também é importante salientar que, em função das diferenças entre as arquiteturas dos dispositivos, o mesmo código gerado utilizando o dispositivo de simulação pode ter tamanhos nas áreas de dados e código diferentes daqueles gerados para uma UCP da Série Nexto.

# Criação e Execução de Projetos



Na descrição a seguir será mostrado como criar um projeto simples contendo um programa aplicativo, como carregá-lo para a UCP através de um Gateway Server e também como executá-lo e monitorá-lo. O sistema de tempo de execução do Nexto usado para este projeto exemplo é fornecido com as configurações do MasterTool IEC XE.

O programa exemplo será escrito em linguagem de Texto Estruturado (ST) e consiste em um programa (UserPrg) e um bloco funcional (FB1). UserPrg conterá uma variável de contador (ivar) e uma chamada de bloco funcional (FB1). FB1 obterá a entrada “in” a partir do UserPrg, adicionará “2” a esta entrada e escreverá o resultado na saída “out”. Out será lida pelo UserPrg.

Em linhas gerais, a sequência abaixo sumariza o procedimento de criação e execução do projeto.

1. Iniciar o MasterTool IEC XE;
2. Criar um projeto
3. Declarar variáveis na UserPrg;
4. Digitar o código no corpo da UserPrg;
5. Criar a POU de programação adicional (ST bloco funcional FB1);
6. Definir os recursos para executar e controlar o programa no Nexto;
7. Definir a aplicação ativa;
8. Configurar o canal de comunicação para o Nexto;
9. Compilar e carregar a aplicação no Nexto;
10. Iniciar a aplicação;
11. Monitorar a aplicação;
12. Depurar uma aplicação;
13. Configurar um breakpoint e realizar a varredura do programa.

## Declarar Variáveis na UserPrg

A POU “UserPrg” disponível na janela *Dispositivos* do editor de linguagem ST (parte central da interface do usuário MasterTool IEC XE) é automaticamente aberta com um duplo clique em seu item correspondente na árvore dos Dispositivos.

As POUs, em geral, sempre podem ser abertas na visualização do editor desta forma (clique duplo no item correspondente da árvore).

O editor ST consiste em uma parte de declaração (parte superior) e um “corpo” (parte inferior), separados por um divisor de tela móvel.

A parte da declaração mostra números de linha no canto esquerdo, o tipo e nome das POUs (PROGRAM UserPrg) e as palavras-chave “VAR” e “END\_VAR” para a declaração das variáveis.

O corpo é vazio, somente a linha número 1 é exibida na figura a seguir.

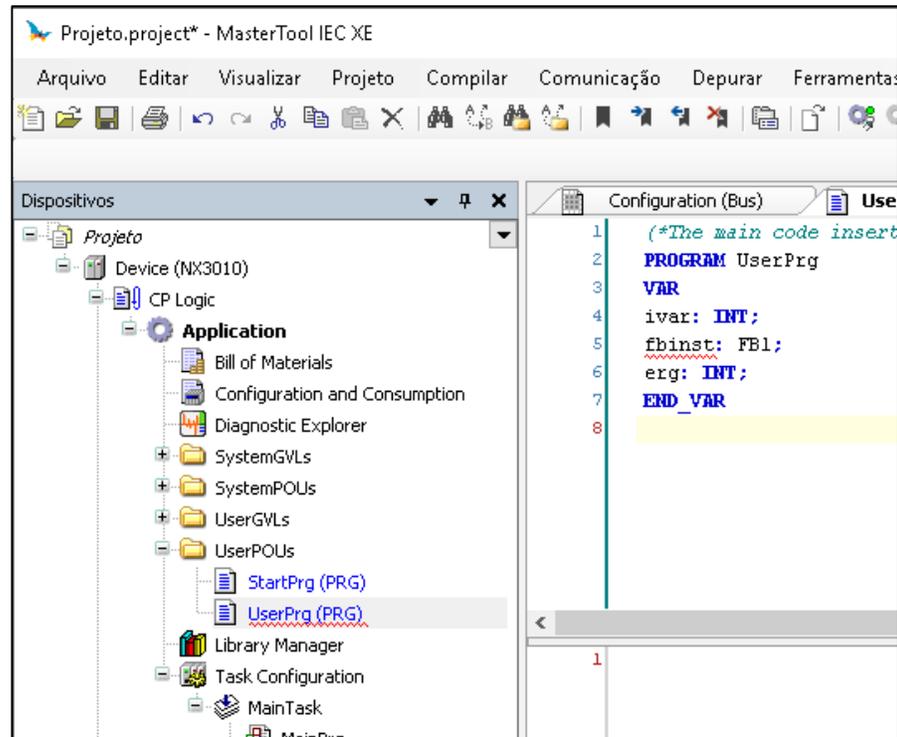


Figura 2-37. Janela do Editor ST

Na parte de declaração do editor posicione o cursor após VAR e pressione a tecla <ENTER>. Será inserida uma linha vazia para entrar com a declaração das variáveis “ivar” e “erg” do tipo INT e fbinst do tipo FB1:

```
PROGRAM UserPrg
VAR
ivar: INT;
fbinst: FB1;
erg: INT;
END_VAR
```

Outra opção é digitar diretamente uma instrução na parte de implementação do editor (corpo) e utilizar a função Autodeclarar.

### Digitar o Código no Corpo da UserPrg

```
ivar := ivar+1; // contador
fbinst(in:=11, out=>erg); // chama o bloco funcional do tipo FB1,
// com o parâmetro de entrada "in"
// a saída é escrita em "erg"
```

Outra possibilidade é usar a configuração da Autodeclaração. Sem nenhuma declaração anterior, entre com uma instrução no corpo do programa e pressione a tecla <ENTER>. Para cada variável não declarada encontrada na linha de implementação, o diálogo *Autodeclarar* se abre para realizar as configurações da declaração.

Figura 2-38. Diálogo Autodeclarar

O escopo e nome das variáveis, assim como a POU atual (objeto) serão preenchidos automaticamente. Digite o tipo desejado e o valor de inicialização de acordo com a declaração descrita anteriormente e adicione um comentário. Adicionalmente poderia ser inserido o endereço de uma variável de representação direta no campo *Endereço*.

Confirme o diálogo com *OK*. Isto fará com que a declaração de “erg” seja inserida na parte de declaração da POU com os comentários, conforme a figura a seguir.

```

(*The main code inserted here*)
PROGRAM UserPrg
VAR
  ivar: INT;
  fbinst: FB1;
  // result
  erg: INT;
END_VAR

```

Figura 2-39. Declaração da Variável “erg”

### Criar a POU de Programação Adicional (ST Bloco Funcional FB1)

O bloco funcional FB1 executa a seguinte expressão:  $out = in + 2$ .

Escolha o comando *Acréscimo Objeto* no menu *Projeto*.

Selecione *POU* na parte esquerda do diálogo *Acréscimo Objeto*. Insira o nome “FB1” para a POU e ative a opção *Bloco funcional* na seção *Tipo*.

Escolha *Texto Estruturado* para a linguagem de implementação.

Pressione o botão *Abrir* para confirmar as configurações do objeto.

Outra janela do editor será aberta para o novo bloco funcional FB1. Da mesma forma que em UserPrg, declare as seguintes variáveis:

```

FUNCTION_BLOCK FB1
VAR_INPUT

```

```

    in:INT;
END_VAR
VAR_OUTPUT
    out:INT;
END_VAR
VAR
    ivar:INT:=2;
END_VAR

```

Na parte de implementação do editor, digite:  
 out:=in+ivar;

## Definir os Recursos para Executar e Controlar o Programa no Nexto

### Iniciar Servidor de Gateway

O servidor de gateway é iniciado automaticamente na inicialização do sistema como um serviço. O ícone (🏠) na barra do sistema, indica que o servidor está rodando. O ícone (⌘) indica que o gateway está interrompido. Este ícone é parte do programa GatewaySysTray, disponível para controle e monitoramento do serviço de Gateway. Ele fornece um menu com comandos *Start Gateway* e *Stop Gateway*, permitindo ao usuário parar e reiniciar o serviço manualmente. O menu também inclui o comando *Exit Gateway SysTray* do controle de gateway, o qual encerra o programa GatewaySysTray (mas não o serviço de Gateway). O programa GatewaySysTray é iniciado automaticamente quando o Windows é iniciado, entretanto também pode ser inicializado manualmente através do menu *Programas*.

### Iniciar o Nexto (Simulação)

O dispositivo de simulação está disponível na inicialização do sistema. Na barra do sistema, os ícones (🏠) e (🏠) representam os status “parado” e “em execução” respectivamente. Se o sistema permitir, este dispositivo será iniciado automaticamente na inicialização do sistema. Caso contrário ele deverá ser iniciado manualmente via comando *Iniciar CP* (duplo clique no ícone). Este ícone é parte integrante do MasterTool IEC XE. Ele fornece um menu com os mesmos comandos *Iniciar* e *Parar* do Servidor de Gateway. O comando *Sair* do controle do CP encerra o programa GatewaySysTray (mas não o serviço de Gateway).

### Configurar um Canal de Comunicação com o Nexto

A conexão resultante aparecerá na linha seguinte à opção “Selecionar caminho na rede para o controlador Nexto”.

Execute um clique duplo no *Device* na janela *Dispositivos*. O diálogo *Device* será aberto com o subdiálogo *Configurações de Comunicação*. Configure a conexão entre o Nexto e o programador conforme indicado anteriormente no item *Localizando a Rede*.

## Executar e Monitorar a Aplicação no Nexto

### Compilar e Carregar a Aplicação no Nexto

Para verificar erros sintáticos na aplicação ativa, use o comando *Gerar Código* (menu *Compilar*).

NOTA: não será gerado nenhum código neste caso. Informações, avisos e mensagens de erro serão exibidos na janela de Mensagens localizada na parte inferior da interface do usuário (padrão).

Mesmo que esta verificação sintática não tenha sido feita antes, é possível realizar o login no Nexto. Certifique-se de que o Nexto está em execução (símbolo colorido na barra do sistema).

Use o comando *Login* (menu *Comunicação*). Se as configurações de comunicação foram configuradas corretamente a seguinte mensagem aparecerá (caso contrário será necessário corrigi-las):

“Não existe aplicação no Dispositivo. Você deseja criá-la e continuar com o envio?”

Confirme com *Sim* para iniciar a compilação e o envio da aplicação.

As mensagens de compilação serão exibidas na janela de *Mensagens*. Se o projeto foi criado corretamente, não deve ocorrer erro e a aplicação pode ser iniciada.

Após a criação do “projeto padrão” e do envio da aplicação conforme descrito nos passos anteriores, a mesma é iniciada no dispositivo.

#### **Iniciar a Aplicação no Nexto**

Execute o comando *Iniciar* no menu *Depurar*. O programa iniciará a execução e a sinalização RUN (em verde) será exibida na barra de status na parte inferior da interface do usuário.

#### **Monitorar a Aplicação**

Existem três possibilidades de monitoração das variáveis do programa aplicativo:

- Listas de monitoração
- Escrita e forçamento de variáveis
- Visualização online de POUs

#### **Abrir uma Janela de Instância do Programa**

A visualização de uma POU fornece todas as expressões de monitoração daquela instância em uma tabela na parte de declaração e - se ativada como monitoração em linha - também na parte de implementação.

Para abrir a visualização online dê um clique duplo na POU *UserPrg* na janela *Dispositivos* ou selecione o item e execute o comando *Editar Objeto* (menu de contexto).

Na parte inferior da visualização aparecerão as linhas de código conforme digitadas no modo off-line, acrescidas de pequenas janelas de monitoração em linha após cada variável, mostrando o valor atual. Na parte superior, uma tabela mostrará as expressões de monitoração da POU, que consistem nos valores atuais das variáveis da aplicação no Nexto.

Device.Application.UserPrg			
Expressão	Tipo	Valor	Valor preparado
ivar	INT	253	
+ fbinst	FB1		
erg	INT	13	

```

1  ivar 253 := ivar 253 +1; // contador
2  fbinst(in 11 :=11, out 13 =>erg 13); // chama o bloco funcional do tipo FB1,
3  // com o parâmetro de entrada "in"
4  // a saída é escrita em "erg"
5  RETURN

```

Figura 2-40. Expressões de Monitoração e Linhas de Código

### Escrever e Forçar Variáveis

É possível escrever ou forçar um Valor Preparado em uma variável no Nexto, ou seja, “ivar” assumirá este valor no início do próximo ciclo. Dê um duplo clique na coluna Valor Preparado, digite o valor inteiro desejado e saia do campo com <Enter> ou clicando com o mouse fora deste campo. Execute o comando Escrever Valores ou Forçar Valores (menu Depurar) para escrever ou forçar este valor no Nexto. O resultado aparecerá na coluna Valor.

### Usar as Visualizações de Monitoração

Janelas de expressões de monitoração podem ser usadas para configurar tabelas específicas de monitoração da aplicação para, por exemplo, a depuração do programa aplicativo.

A partir do menu *Visualizar* use o comando *Janelas de Monitoração, Monitorar 1* para abrir a janela de monitoração.

Na coluna *Expressão* clique na primeira linha da tabela para abrir o campo de edição. Digite o caminho completo para a variável a ser monitorada (“ivar”): “Device.Application.MainPrg.ivar”.

É recomendado usar o assistente de entrada via botão específico para tal. Feche o campo de edição com <ENTER>. O tipo da variável será incluído automaticamente.

Faça o mesmo para as demais variáveis. A lista de monitoração mostrada na próxima figura contém somente expressões da UserPrg, mas, obviamente, é possível listar várias variáveis do projeto. Note que para instâncias, por exemplo do bloco funcional FB1, basta digitar “Device.Application.MainPrg.fbinst”. As variáveis específicas destas aparecerão automaticamente e as linhas correspondentes podem ser abertas via símbolo “+”. O valor atual da variável é mostrado na coluna *Valor*.

Monitoração 1			
Expressão	Aplicação	Tipo	Valor
UserPrg.erg	Device.Application	INT	13
+ UserPrg.fbinst	Device.Application	FB1	
UserPrg.ivar	Device.Application	INT	4437

Figura 2-41. Lista de Monitoração

Se ainda não foi feito, selecione o objeto da aplicação e execute o comando *Iniciar* no menu de contexto. A aplicação será iniciada no Nexto e o valor atual será exibido na coluna *Valor*.

Monitoração 1			
Expressão	Aplicação	Tipo	Valor
UserPrg.erg	Device.Application	INT	13
UserPrg.fbinst	Device.Application	FB1	
in		INT	11
out		INT	13
ivar		INT	2
UserPrg.ivar	Device.Application	INT	5931

Figura 2-42. Exibição do Valor Atual

Também é possível realizar a escrita e o forçamento de valores.

Para desconectar-se do Nexto, execute o comando *Logout* no menu *Comunicação*.

## Depurar uma Aplicação via Breakpoint e Varredura do Programa

No modo online, os breakpoints podem ser configurados como posições de parada na execução do programa.

Quando o programa alcançar um breakpoint, o programa poderá ser executado em etapas. Em cada posição de parada, poderá ser visto o valor atual das variáveis nas visualizações de monitoração.

Selecione a linha 1 de UserPrg. Pressione a tecla <F9> (idem ao comando *Alternar Breakpoint*, menu *Depurar*). O breakpoint será indicado.

```

1  ● ivar 1277 := ivar 1277 + 1; // CONTADOR
2  fbinst(IN 11 :=11, OUT 13 =>ERG 13 ); // CHAMA O BLOCO FUNCIONAL DO TIPO FB1,
3  // COM O PARÂMETRO DE ENTRADA "IN"
4  // A SAÍDA É ESCRITA EM "ERG"RETURN

```

Figura 2-43. Aplicação em "Stop"

Uma aplicação em execução parará em um breakpoint:

```

1  ● ivar 1277 := ivar 1277 + 1; // CONTADOR
2  fbinst(IN 11 :=11, OUT 13 =>ERG 13 ); // CHAMA O BLOCO FUNCIONAL DO TIPO FB1,
3  // COM O PARÂMETRO DE ENTRADA "IN"
4  // A SAÍDA É ESCRITA EM "ERG"RETURN

```

Figura 2-44. Aplicação no Modo "Run"

A tecla <F8> permite percorrer os passos (equivalente ao comando *Passar Dentro* no menu *Depurar*) e, portanto, percorrerá também instâncias de blocos funcionais.

Para pular as etapas do bloco funcional, use <F10> (o mesmo que o comando *Passar Sobre*). Todo os valores da variável do Nexto serão exibidos.

O diálogo breakpoints também está disponível via comando *Breakpoints*, no menu *Visualização*. Aqui, os breakpoints configurados no momento podem ser visualizados e editados, assim como novos breakpoints também podem ser inseridos.

Observe também que as posições de breakpoint serão memorizadas ao fazer o logout. Elas serão indicadas por marcadores vermelho-claro.

# Diagnósticos



## Página de Diagnósticos

A página de Diagnósticos ou Diagnostics Explorer é a inclusão dos diagnósticos dos dispositivos via WEB dentro do MasterTool IEC XE, a fim de que o acesso seja mais rápido e objetivo. O acesso à característica ocorre de duas maneiras:

Acessar a opção Diagnostics Explorer na árvore de dispositivos (localizada na árvore de dispositivos). A página web dos diagnósticos serão carregados automaticamente ao abrir a tela, senão digite o IP correto no campo indicado na figura abaixo. Lembre-se que para a página de diagnósticos ser exibida, o usuário deverá estar com uma UCP definida como caminho ativo.

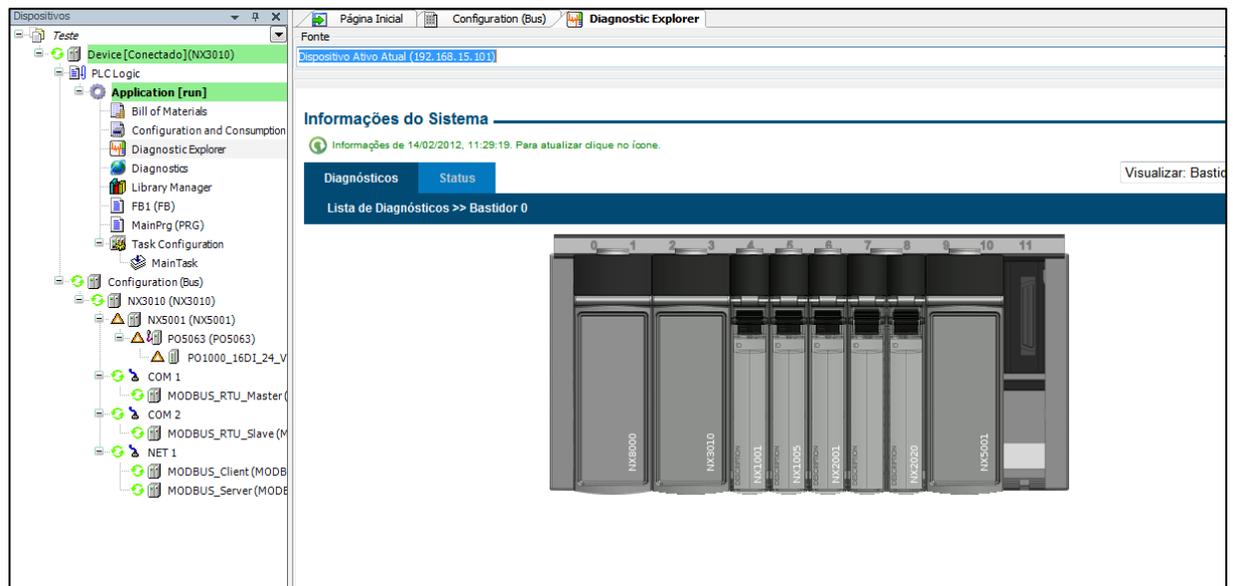


Figura 2-45. Diagnostics Explorer

Clicar com o botão direito do mouse sobre o módulo e selecionar Diagnósticos. O Diagnostics Explorer será aberto e direcionará para a página de status do respectivo módulo.

O objeto Diagnostics (localizado na árvore de dispositivos) é o local onde são exibidas as Variáveis Globais de diagnósticos de todos os dispositivos que estão sendo utilizados.

Cada dispositivo utilizado, possui variáveis de diagnósticos e as mesmas são exibidas conforme a tela abaixo.

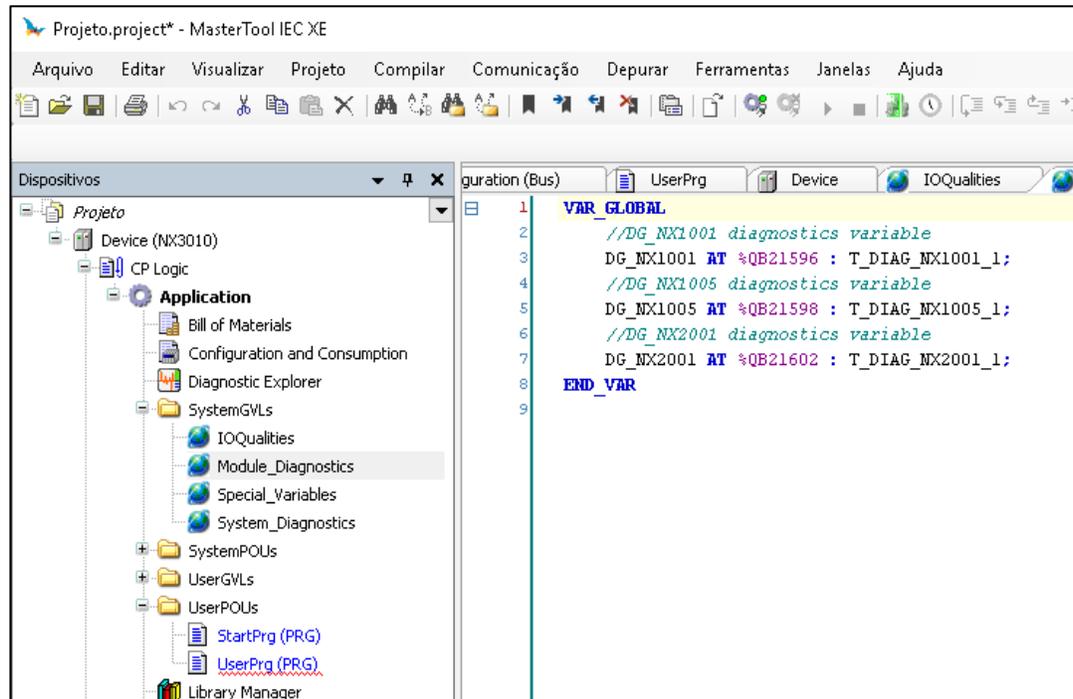


Figura 2-46. Variáveis de Diagnóstico

Ao ser removido algum dispositivo, suas variáveis de diagnósticos são automaticamente removidas.

Para os drivers de comunicação por Mapeamento Simbólico são criados outros dois objetos na árvore de dispositivos chamados *Disables* e *ReqDiagnostics*. Além disso, as variáveis de diagnóstico destes drivers são declaradas no objeto *Diagnostics*, não utilizando a diretiva AT na sua declaração, pelo fato de não existir relação entre uma variável simbólica e uma variável de representação direta para estes dispositivos. Este comportamento pode ser observado na figura a seguir.

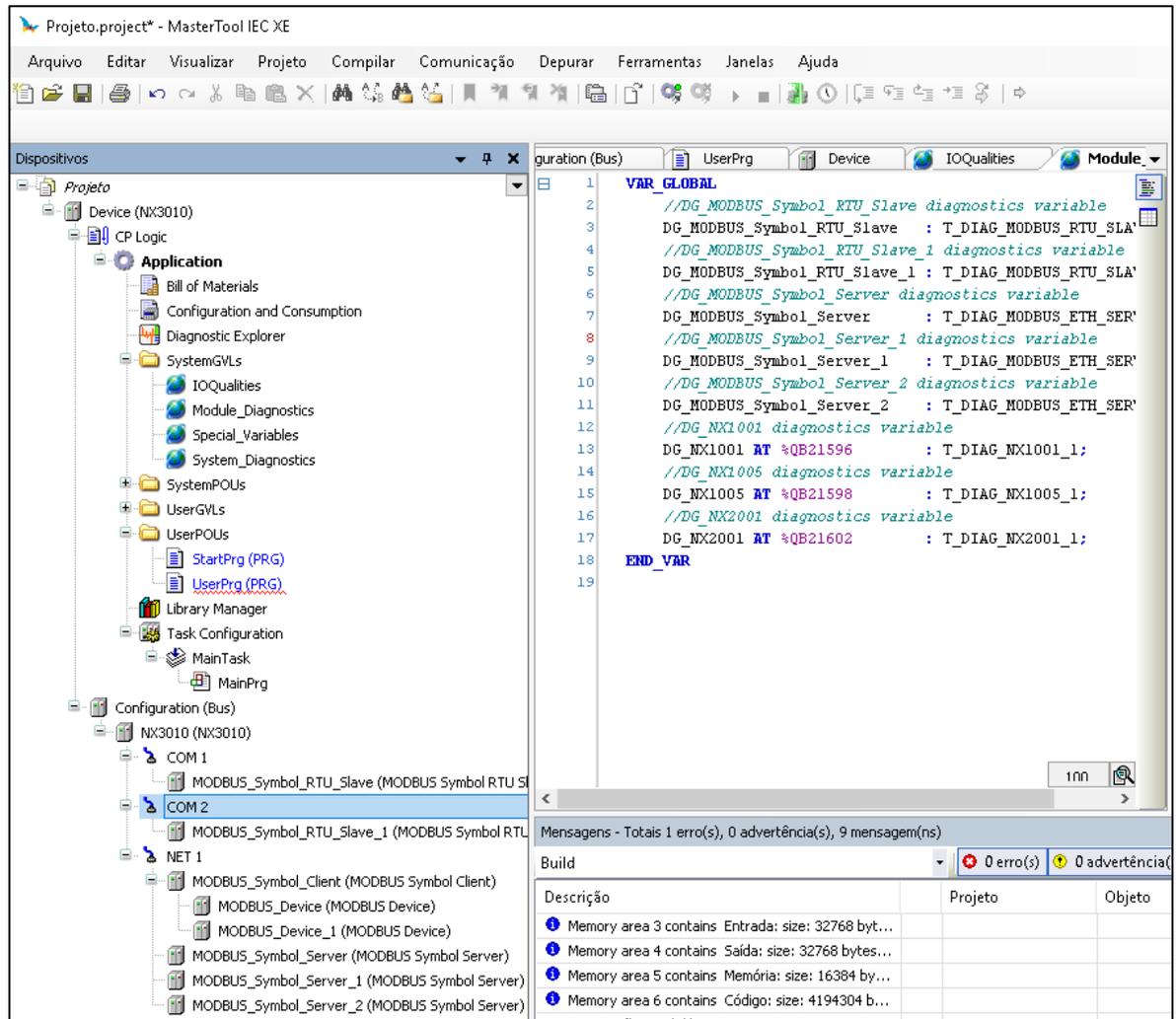


Figura 2-47. Variáveis de Diagnóstico Utilizando Drivers de Comunicação por Mapeamento Simbólico

O objeto *Disables* é utilizado para declarar as variáveis de desabilitação de requisições em drivers que utilizam mapeamento simbólico. Estas variáveis podem ser declaradas diretamente pelo usuário em qualquer POU ou GVL, mas também podem ser geradas automaticamente através do botão *Gerar Variáveis de Desabilitação* disponível na tela de configuração das requisições do driver. Se o objeto não existir no projeto ainda ele será criado após pressionar o botão. Após isso, se for necessário as variáveis declaradas neste objeto podem ser editadas.

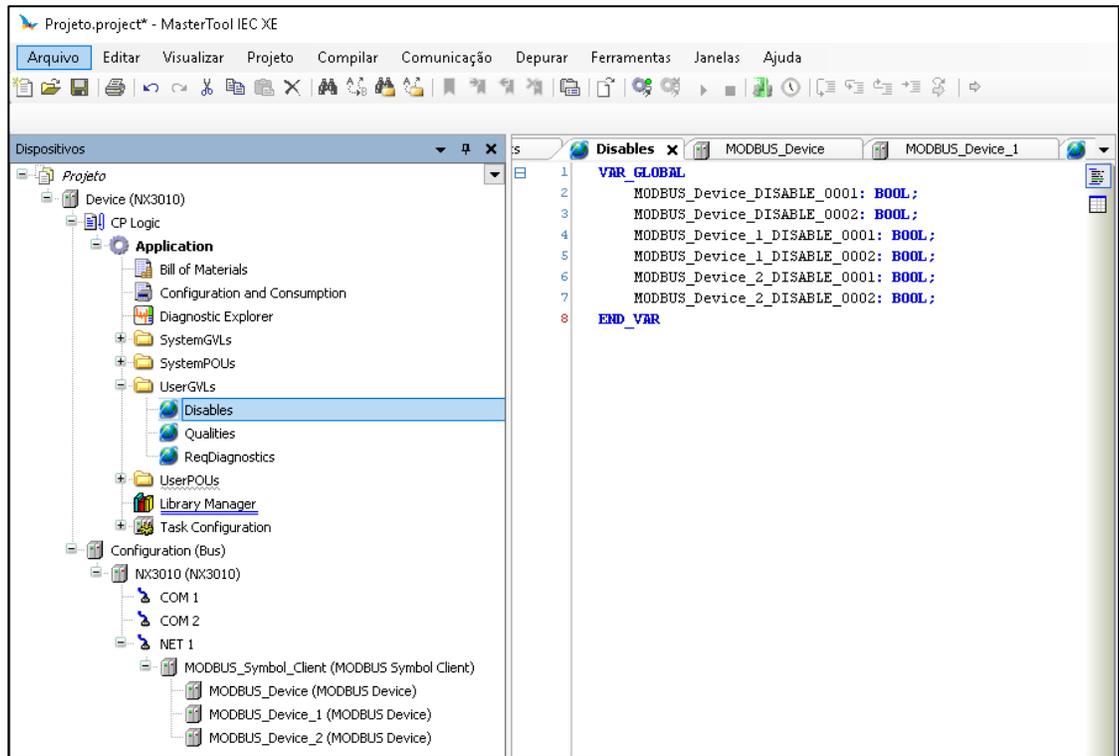


Figura 2-48. Objeto Disables

O objeto ReqDiagnostics é utilizado para declarar as variáveis de diagnóstico de requisições. Estas variáveis podem ser declaradas diretamente pelo usuário em qualquer POU ou GVL, mas também podem ser geradas automaticamente através do *botão Gerar Variáveis de diagnóstico* disponível na tela de configuração das requisições do driver. Se o objeto não existir no projeto ainda ele será criado após pressionar o botão. Após isso, se for necessário as variáveis declaradas neste objeto podem ser editadas.

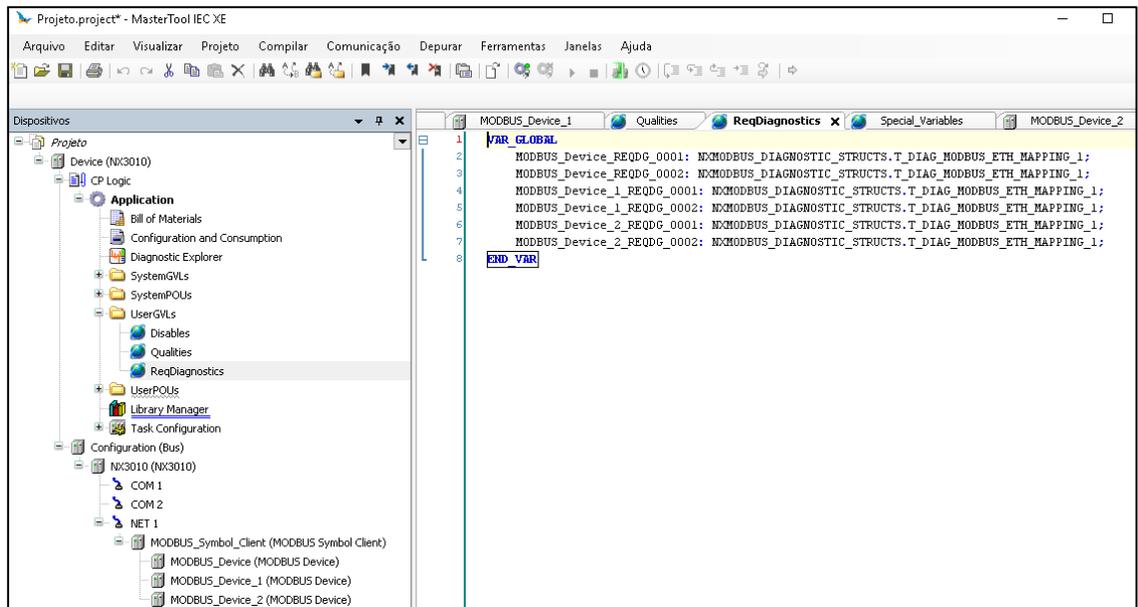


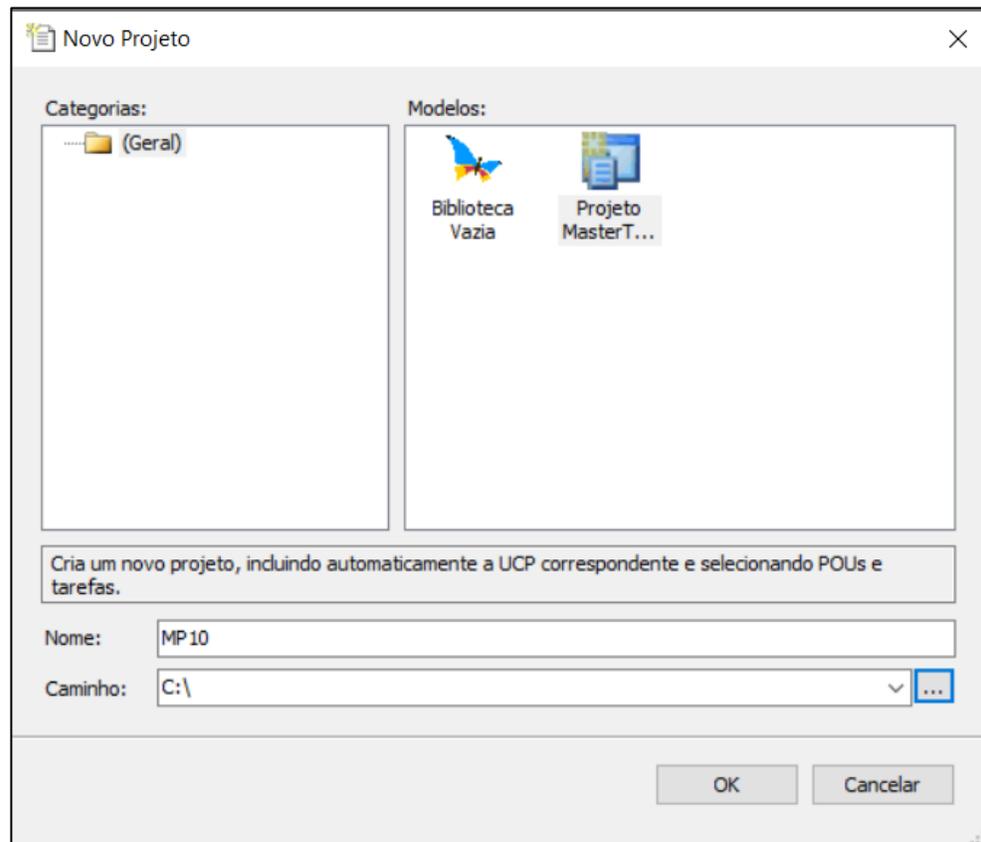
Figura 2-49. Objeto ReqDiagnostics



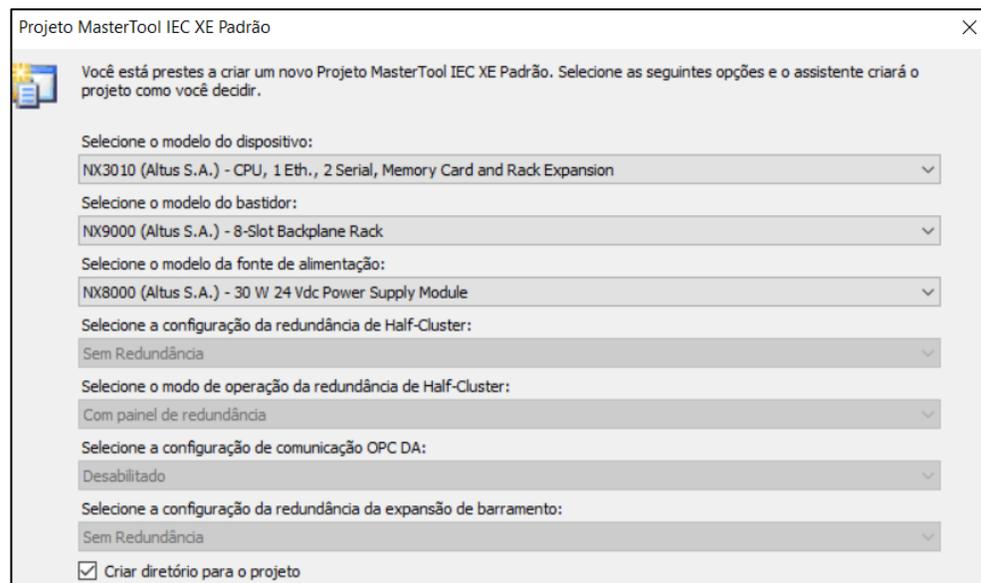


Resolução do Estudo Dirigido 2-1:

Etapa 1: escolha o tipo de projeto.



Etapa 2: escolha dos módulos de hardware (dispositivo, bastidor e fonte de alimentação).



E, na sequência, configure as interfaces Ethernet da UCP.

Etapa 3: escolha a quantidade de pontos que serão utilizados.

Projeto MasterTool IEC XE Padrão

Escolha a quantidade de pontos E/S que serão utilizados e o assistente criará os seguintes objetos dentro do projeto.

Escolha a solução para módulos E/S:  
Módulos NX

Pontos de entrada digital: 8

Pontos de entrada analógica V/I: 0

Pontos de entrada analógica termopar: 0

Pontos de entrada analógica RTD: 0

Pontos de saída analógica V/I: 0

Pontos de saída digital a transistor: 0

Pontos de saída digital a relé: 4

Quantidade de módulos E/S que serão utilizados: 1 de 128.  
Quantidade de bastidores que serão utilizados: 1 de 1.

Etapa 4: selecione o perfil do projeto e a linguagem padrão.

Projeto MasterTool IEC XE Padrão

Selecione a configuração do perfil do projeto e o assistente criará os seguintes objetos dentro do projeto.

Selecione o perfil do projeto:  
Simple

Selecione a linguagem padrão para todos os programas:  
Diagrama de Lógica Ladder (LD)

Sobre o perfil de projeto

Este modelo de projeto inclui apenas uma tarefa e sua POU já está completamente definida conforme apresentado abaixo. Não é necessário cuidado com o escalonamento das tarefas.

Tarefa	POU	Prioridade	Tipo	Intervalo	Evento
MainTask	MainPrg	13	Cíclica	20 ms	-

Etapa 5: selecione as linguagens dos programas de tarefas comuns e conclua o assistente.

Projeto MasterTool IEC XE Padrão

Selecione a linguagem e o assistente criará os seguintes objetos dentro do projeto.

Programas de Tarefas Comuns (Interrompe Tarefas Contínuas)

Programas associados à Main Task:

MainPrg em:  
Texto Estruturado (ST)

UserPrg em:  
Diagrama de Lógica Ladder (LD)

StartPrg em:  
Diagrama de Lógica Ladder (LD)

## Considerações gerais sobre a configuração do sistema

Os controladores da família Nexto são facilmente configurados no MasterTool IEC XE. A ferramenta gráfica permite uma rápida e amigável maneira de configurar o sistema. A configuração das redes de barramento de campo e protocolos de comunicação padrão como o PROFIBUS DP e os protocolos MODBUS são integrados na ferramenta de programação. Esta característica permite ao usuário definir todos os parâmetros de configuração na mesma plataforma. Deste modo, o usuário não precisa utilizar diferentes ferramentas de software para esta tarefa. Cada controlador pode conter uma ou várias aplicações. Isto significa que, se desejado, vários usuários podem trabalhar no mesmo controlador simultaneamente. O configurador de tarefas permite que o usuário compartilhe sua aplicação em diferentes camadas de tempos e funções que, após, serão sincronizadas com a camada de E/S. Adicionalmente, o usuário tem a visualização completa da arquitetura da aplicação com a posição física e as informações do módulo conforme ilustrado na figura a seguir para a aplicação desenvolvida anteriormente.

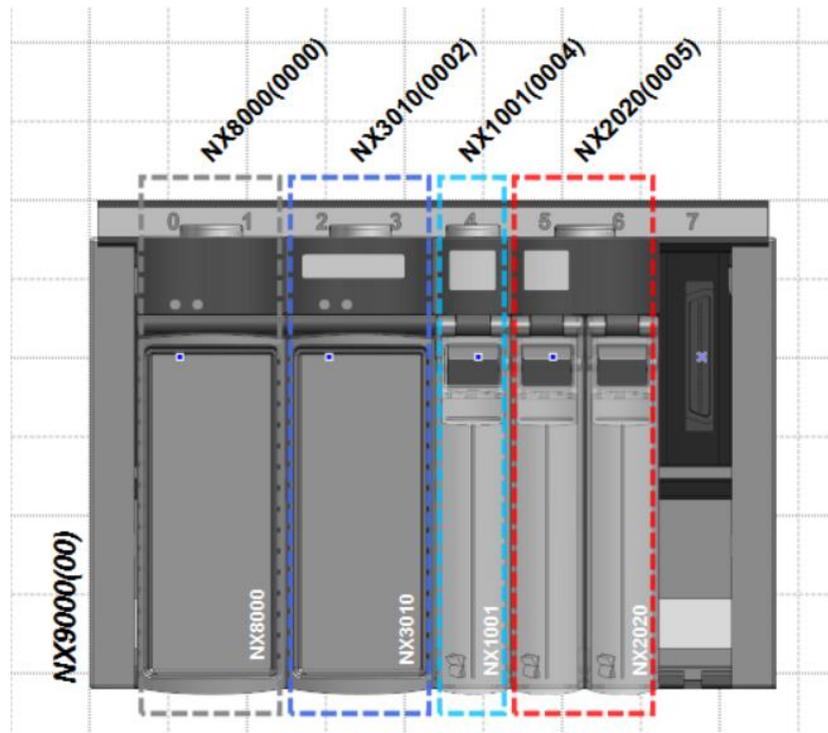


Figura 2-50. Configuração do barramento

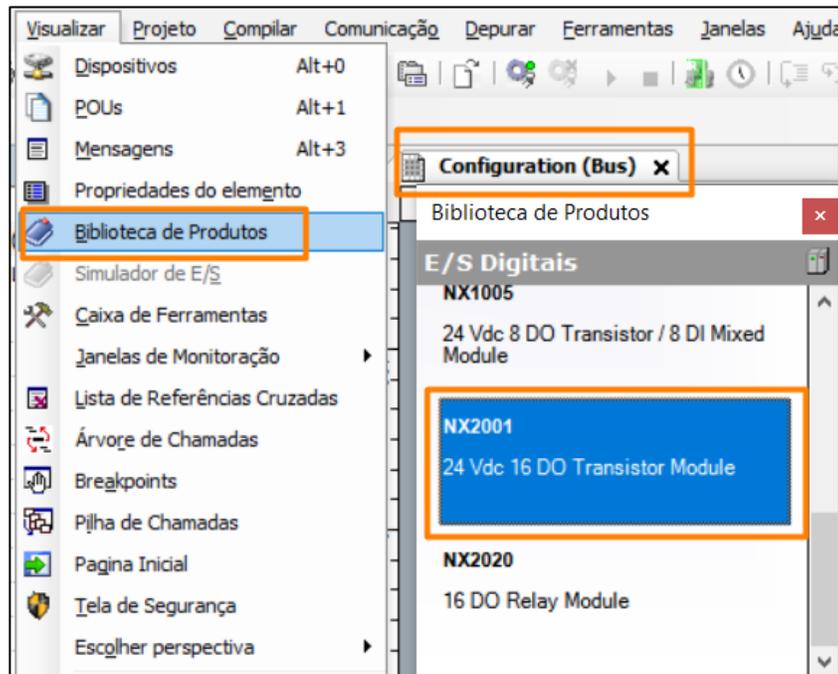
**Estudo Dirigido 2-2: configuração do barramento lógico do Nexto**

O objetivo deste estudo dirigido é praticar a configuração do barramento lógico da arquitetura Nexto por meio de um sistema prático de seleção dos módulos de interface de entrada e saída no MasterTool IEC XE, ou seja, como introduzir os módulos da arquitetura no barramento lógico?

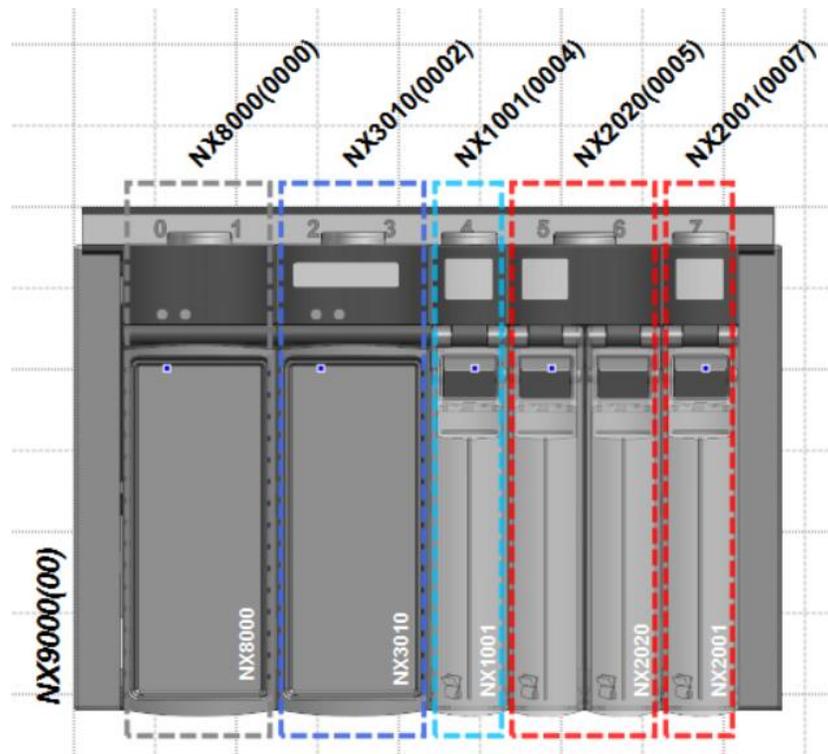
**ANOTAÇÕES**

Resolução do Estudo Dirigido 2-2: escolha através do mouse o módulo a ser introduzido no barramento lógico, na biblioteca de módulos, e arraste (mantendo selecionado com o botão

esquerdo) até a posição desejada. A figura a seguir mostra um exemplo de módulo a ser inserido (NX2001).



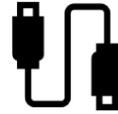
Procure visualizar e centralizar o marcador lógico do conector EURO localizado no bastidor com o (x) virtual do módulo. Após a centralização solte o botão esquerdo do mouse que o módulo virtual se afixará automaticamente na posição desejada. Após, a configuração, o sistema lógico da arquitetura deve ficar da maneira como está indicado na figura a seguir (seqüência da pré-configuração realizada anteriormente na aplicação via assistente).



### **Pesquisa na Documentação: configuração padrão do sistema operacional**

Consulte, na documentação do produto, a configuração padrão de fábrica utilizada pelo sistema operacional da série Nexto.

Fonte: Manual de Utilização Série Nexto – MU214000



## 3. Programação do CP Nexto

O MasterTool IEC XE é totalmente compatível com a IEC 61131-3 e contempla todas as linguagens de programação definidas na norma, incluindo a linguagem LADDER (LD) que utilizaremos nesse tutorial para programação do CP Nexto. Antes disso, vamos estudar alguns conceitos básicos comuns a todas as linguagens de programação da norma.

### Elementos Básicos da Aplicação



Uma aplicação é o conjunto dos objetos necessários para executar uma instância específica do programa no Nexto. É também chamada de Programa Aplicativo. Os principais elementos que constituem uma aplicação estão indicados na sequência.

#### Unidade de Organização de Programa

O termo POU é utilizado basicamente para todos os objetos usados para criar um programa do CP, e significa Unidade de Organização de Programa.

Uma POU pode ser uma função, bloco funcional, método, ação, interface, DUT ou ainda um arquivo externo de qualquer formato.

Cada POU consiste de uma parte de declaração (janela superior) e uma parte de implementação (janela inferior). O corpo é escrito em uma das linguagens de programação disponíveis, que são IL, ST, SFC, FBD, LD ou CFC.

#### Programa

Um programa é uma POU que retorna um ou vários valores durante a operação. Todos os valores são mantidos desde a última vez que o programa foi executado até a próxima execução. Um programa pode ser chamado por outra POU, mas a chamada de programa não é permitida em uma função. Não existem instâncias (reproduções/cópias) de programas.

#### Função

Uma função é uma POU que produz exatamente um elemento de dados (que pode consistir de um ou vários elementos, tais como campos ou estruturas) ao ser processada e cuja chamada em linguagens textuais pode ocorrer como um operador em expressões. Funções não contêm informações de estado, ou seja, a invocação de uma função com os mesmos argumentos (parâmetros de entrada) sempre produzirá os mesmos valores (saída). Por esta razão, as funções não devem conter variáveis globais e endereços.

#### Bloco Funcional

Um bloco funcional é uma POU que fornece um ou mais valores durante o processamento de um programa do CP. Ao contrário da função, os valores das variáveis de saída e as necessárias variáveis internas mantêm-se de uma execução do bloco funcional até a próxima execução.

Assim sendo, a chamada de um bloco funcional com os mesmos argumentos (parâmetros de entrada) nem sempre produz os mesmos valores de saída.

Um bloco funcional sempre é chamado através de uma instância, que é uma reprodução (cópia) do bloco funcional.

Cada instância tem o seu próprio identificador (nome da instância) e uma estrutura de dados contendo entradas, saídas e variáveis internas.

Assim como as variáveis, as instâncias são declaradas local ou globalmente e, por meio delas, o nome do bloco funcional é indicado como o tipo de dado de um identificador.

A figura a seguir ilustra a chamada da instância “inst1” do bloco funcional “fb”, o qual tem uma entrada (in1) e uma saída (out1).

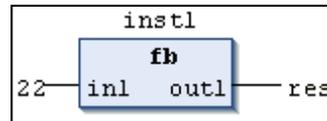


Figura 3-1. Instância “Inst1”

## Lista de Variáveis Globais - GVL

Uma Lista de Variáveis Globais, a GVL (🌐) é usada para declarar variáveis globais. Se uma GVL for definida na visualização das POU, as variáveis estarão disponíveis para todo o projeto. Se uma GVL estiver atribuída a uma determinada aplicação, as variáveis serão válidas dentro desta aplicação.

## Gerenciamento de Bibliotecas

Bibliotecas podem fornecer funções, blocos funcionais, tipos de dados, variáveis globais e até mesmo visualizações, que podem ser usadas em um projeto assim como em outras POU e variáveis definidas diretamente no projeto.

A expressão padrão para um arquivo de biblioteca é .library.

O gerenciamento das bibliotecas em um projeto é feito no Gerenciador de bibliotecas (📦), cuja instalação no sistema é feita previamente no diálogo do Repositório de bibliotecas. A biblioteca standard.library já vem instalada por padrão. Ela contém todas as funções e blocos funcionais necessários em conformidade com a norma IEC61131-3 assim como com as POU para o programador IEC.

## Operadores

Operadores são usados como funções em uma POU. Exemplos de operadores:

- Operadores de atribuição
- Operadores aritméticos
- Operadores de bitstring
- Operadores de deslocamento de bits
- Operadores de seleção
- Operadores de comparação
- Operadores de endereço
- Operadores de chamada
- Funções de conversão de tipo

- Funções numéricas
- Operadores adicionais à norma IEC
- Operadores de escopo adicionais à norma IEC

## Operandos

Os operandos são classificados como segue:

- Constantes (BOOL, TIME, DATE, TIME\_OF\_DAY, DATE\_AND\_TIME, Número, REAL/LREAL, STRING, Literais Tipados)
- Variáveis
- Endereços
- Funções

## Declaração de Variáveis

Para declarar variáveis utilizamos o editor de declaração, o qual consiste em um editor textual ou tabular usualmente disponível em combinação com os editores de linguagem. A classe (ou escopo, no diálogo de declaração) da variável a ser declarada é especificada pelas palavras-chave no âmbito de sua parte de declaração. A declaração de uma variável comum, por exemplo, aparece entre as palavras VAR e END\_VAR.

Sintaxe:

```
<IDENTIFICADOR> {AT <ENDEREÇO>}:<TIPO> {:=<INICIALIZAÇÃO>};
```

As partes entre chaves {} são opcionais. Por exemplo: a variável pode ser diretamente vinculada a um endereço definido usando a palavra-chave AT. Além disso, o valor de inicialização padrão é 0 para todas as declarações, porém valores de inicialização definidos pelo usuário podem ser adicionados na declaração de cada variável via operador de atribuição ":=".

O identificador é o nome da variável. Algumas regras referentes a indicadores:

- Não deve conter espaços nem caracteres especiais.
- Não diferencia maiúsculas e minúsculas, o que significa que VAR1, Var1 e var1 são todas a mesma variável.
- A\_BCD e AB\_CD são considerados diferentes identificadores, porém um identificador não deve conter mais de um sublinhado por linha.
- O número de caracteres do identificador (identificação) é ilimitado.

Cada identificador é atribuído a um tipo de dado que determina quanto de espaço de memória será reservado e quais tipos de valores ele armazena. Exemplos:

- BOOL
- Tipos de dados inteiros
- REAL / LREAL
- STRING / WSTRING
- Tipos de dados de tempo
- Tipos de dados definidos pelo usuário

Exemplo de declaração de uma variável:

```
CONTADOR:INT := 12; (* VARIÁVEL INTEIRA COM VALOR INICIAL 12 *)
```

## Tipos de Variáveis

### Variável de Representação Direta

Variável de representação direta significa que a variável pode ser acessada diretamente na memória utilizando o endereço desejado. Por exemplo: %QB0, %MW100.

### Variável Simbólica

Variáveis simbólicas são variáveis IEC criadas em POU's e GVL's durante o desenvolvimento do aplicativo, as quais não são endereçadas diretamente na memória.

### Variável de Entrada/Saída de Representação Direta

Variável de entrada/saída de representação direta pode ser utilizada para mapear pontos de entrada/saída analógicos ou digitais. Como referência, 8 pontos de entrada/saída digital podem ser representados por um byte e um ponto de entrada/saída analógica pode ser representado por dois bytes.

## Linguagem de Programação Diagrama Ladder (LD)

O Diagrama Ladder é uma linguagem de programação gráfica similar à estrutura de um circuito elétrico. Ele é utilizado para construir intertravamentos lógicos estruturados na forma de redes. Portanto, o LD é bastante útil para controlar a chamada de outras POU's.

O Diagrama Ladder consiste de uma série de redes, sendo cada uma delas limitada a sua direita e esquerda por linhas de corrente verticais (barras de energia). Uma rede contém um diagrama de circuito formado por contatos, bobinas, opcionalmente POU's adicionais (caixas) e por linhas conectoras. No lado esquerdo há uma série de contatos que transmitem a condição ON ou OFF da esquerda para a direita (o que corresponde aos valores booleanos TRUE e FALSE). Para cada contato é atribuída uma variável Booleana. Se esta variável for TRUE, a condição será transmitida da esquerda para a direita ao longo da linha conectora. Caso contrário, o valor OFF transmitirá ON. Assim sendo, as bobinas localizadas na parte direita da rede recebem o valor ON ou OFF vindo da esquerda para a direita (correspondentemente a TRUE ou FALSE para uma variável booleana atribuída).

Exemplo de uma rede LD:

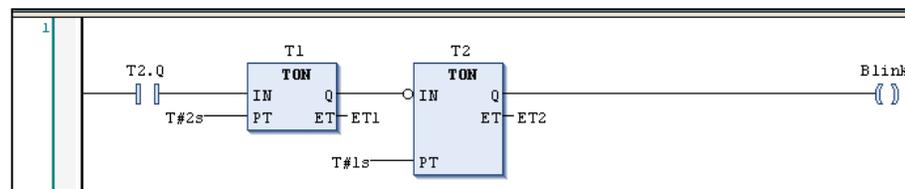


Figura 3-2. Rede LD

### Inserindo e Organizando Elementos

Elementos também podem ser diretamente arrastados com o mouse a partir da caixa de ferramentas para a janela do editor ou de uma posição para outra (arrastar&solta). Para isto, selecione o elemento com um clique de mouse, mantenha-o pressionado e arraste o mesmo para a respectiva rede na visualização do editor. Assim que a rede tiver sido alcançada, todas as possíveis posições de inserção serão indicadas por marcadores verdes. Ao posicionar o cursor

em um destes pontos, o marcador mudará para verde e, soltando o botão, o elemento poderá ser ali posicionado.

Os comandos Recortar, Copiar, Colar e Excluir, por padrão disponíveis no menu Editar, podem ser usados para organizar elementos. A cópia também pode ser feita via arrastar&soltar: selecione o elemento na rede com um clique de mouse e, mantendo-o pressionado, arraste-o para a posição destino. Assim que a posição for alcançada (marcador de posição verde), um símbolo de mais será acrescentado ao símbolo do cursor. Solte o botão do mouse e insira o elemento.

## Menu LD

Quando o cursor está localizado na janela do editor LD o menu deste está disponível na barra do menu e fornece os comandos para programação na visualização do editor atualmente configurado. A figura abaixo mostra os comandos comuns aos editores FBD/LD/IL.

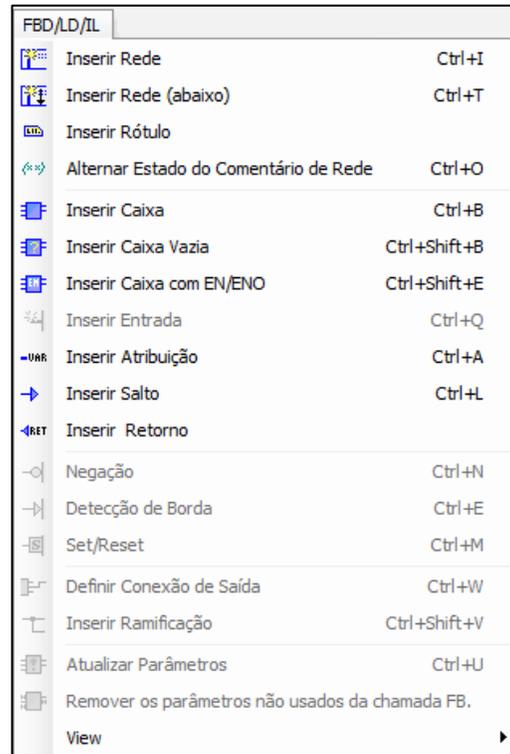


Figura 3-3. Comandos Editores FBD/LD/ID

O editor fornece uma caixa de ferramentas com elementos de programação a serem inseridos na janela do editor (arrastar&soltar). Por padrão, esta caixa de ferramentas pode ser aberta via comando Caixa de Ferramentas no menu Visualização. A figura a seguir mostra a caixa de ferramentas.

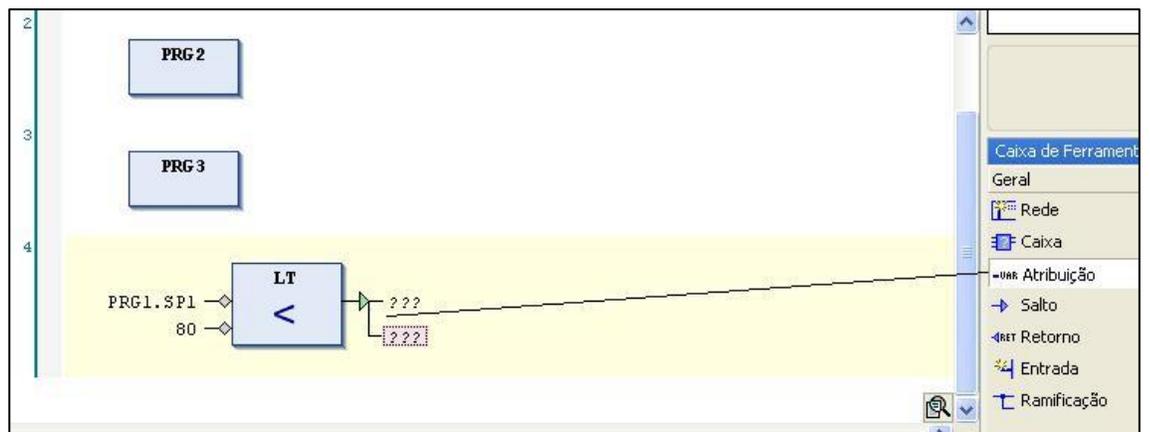


Figura 3-4. Caixa de Ferramentas

Em uma rede podem ser atribuídos, opcionalmente, um título, um comentário e um rótulo. A disponibilidade deste dois primeiros campos título e comentário pode ser ativada ou não no

diálogo Opções do Editor FBD, LD. Para acrescentar um rótulo, o qual pode ser endereçado por um salto, usa-se o comando Inserir rótulo. Veja esses itens na figura a seguir.

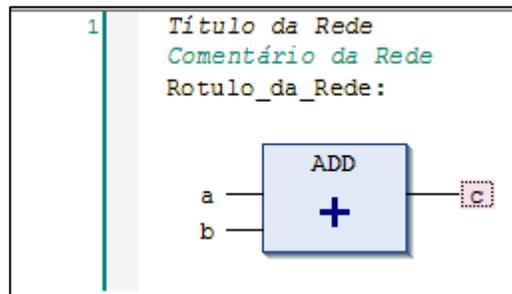


Figura 3-5. Título, Comentário e Rótulo

As redes podem ser configuradas em estado de comentário, o que faz com que a rede não seja processada, mas sim exibida e tratada como um comentário.

## Comandos Básicos do LD

### Contato

Cada rede LD contém um ou vários contatos na sua parte esquerda. Estes contatos são representados por duas linhas paralelas:

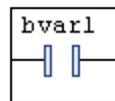


Figura 3-6. Contato

Contatos passam da condição ON (TRUE) ou OFF (FALSE) da esquerda para a direita.

Uma variável booleana é atribuída a cada contato. Se esta variável for TRUE, primeiramente a condição é transmitida da esquerda para a direita e a seguir para uma bobina na parte direita da rede. Caso contrário a conexão direita recebe o valor FALSE.

Múltiplos contatos podem ser conectados tanto em série quanto em paralelo. No caso de dois contatos paralelos somente um deles deve transmitir o valor TRUE para que a ramificação paralela transmita o valor TRUE. No caso de contatos conectados em série todos os contatos devem transmitir a condição TRUE para que o último contato transmita a condição TRUE.

Assim, a organização do contato corresponde tanto a um paralelo elétrico quanto a um circuito série.

Um contato também pode ser negado (barra no símbolo do contato):

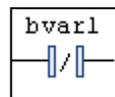


Figura 3-7. Contato Negado

Um contato negado transmite a condição de entrada (TRUE ou FALSE) somente se a variável Booleana atribuída for FALSE. Observe que a caixa de ferramentas fornece diretamente elementos de contatos negados.

Um contato pode ser inserido em uma rede LD através dos comandos:

- Inserir Contato

- Inserir Contato (Direita)
- Inserir Contato Paralelo (Acima)
- Inserir Contato Paralelo (Abaixo)

Estes comandos, por padrão, fazem parte do menu FBD/LD/IL. Alternativamente o elemento pode ser inserido via arrastar e soltar a partir da caixa de ferramentas ou a partir de outra posição no editor.

### Bobina

Ao lado direito de uma rede LD pode haver um número indeterminado de bobinas, as quais são representadas como segue.

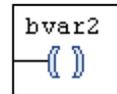


Figura 3-8. Bobina

Elas podem ser organizadas somente **em paralelo**. Uma bobina transmite o valor das conexões da esquerda para a direita e copia os mesmos para uma **variável Booleana** apropriada. Na linha de entrada podem estar presentes os valores ON (TRUE) ou OFF (FALSE). Contatos e bobinas também podem ser negados (barra no símbolo da bobina):

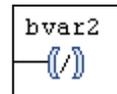


Figura 3-9. Bobina Negada

Neste caso o valor negado do sinal de entrada será copiado para a variável Booleana apropriada. Desta forma um contato negado somente será conectado se o valor Booleano apropriado for FALSE.

Uma bobina pode ser inserida em uma rede através do comando Inserir Atribuição, o qual, por padrão, faz parte do menu FBD/LD. Alternativamente o elemento pode ser inserido via arrastar e soltar a partir da caixa de ferramentas (elementos Ladder) ou a partir de outra posição no editor.

As bobinas do tipo Set (S) e Reset (R) permitem reter o estado a elas atribuído (TRUE ou FALSE). Veja um exemplo na figura abaixo.

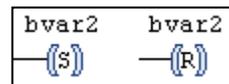


Figura 3-10. Bobinas Set e Reset

### Editor LD no Modo Online

No modo online o Editor LD fornece visualizações para a monitoração, escrita e forçamento das variáveis e expressões no CP. A funcionalidade de depuração (breakpoints, passo a passo etc.) também está disponível. A figura a seguir mostra um exemplo de visualização online do Editor LD.

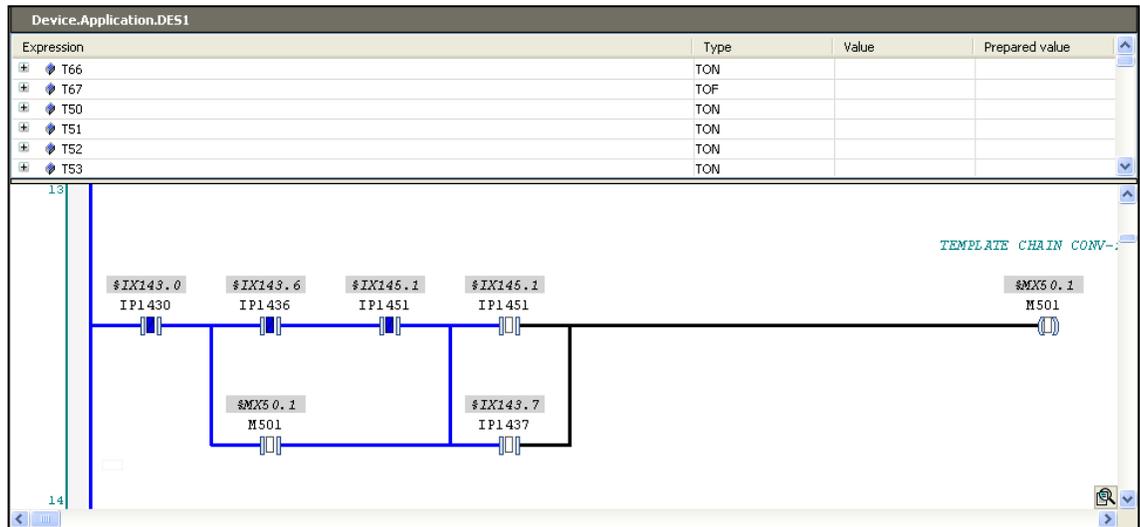


Figura 3-11. Visualização Online do Editor LD

### Forçamento/Escrita de Variáveis

No modo online é possível preparar um valor para forçamento ou escrita de uma variável tanto no editor de declaração quanto na parte de implementação. Na parte de implementação com um clique de mouse na variável, o seguinte diálogo se abrirá:

The 'Preparar Valor' dialog box contains the following fields and options:

- Expressão: ivar
- Tipo: INT
- Current value: 6
- O que você quer fazer?
  - Prepara um novo valor para a próxima operação de escrita ou força
    - 3
  - Remover a preparação com um valor.
  - Liberar o forçamento, sem modificar o valor.
  - Liberar o forçamento e restaurar a variável ao valor que tinha antes de forçá-la.

Buttons: OK, Cancelar

Figura 3-12. Diálogo Preparar Valor

Neste diálogo encontra-se nome da variável complementado pelo seu caminho na árvore de dispositivos (Expressão), seu tipo e seu valor atual. Ativando o item correspondente o usuário escolhe se deseja:

- Preparar um novo valor que deve ser inserido no campo de edição
- Remover um valor preparado,
- Liberar a variável atualmente forçada,
- Liberar a variável atualmente forçada e levá-la ao valor a ela atribuído antes do forçamento (reset).

A ação selecionada será efetivada com a execução do comando Forçar Valores (menu Online) ou através da tecla F7.

## Criação de POUs



Para acrescentar uma POU (POU), selecione o item correspondente na árvore das POU ou Dispositivos (uma aplicação, por exemplo), use o comando Acrescentar objeto no menu de contexto e selecione POU no submenu que aparecerá. O diálogo Acrescentar POU abrirá para que se configure o seu nome, tipo e linguagem de implementação.

### POU em LD

Para ilustrar a utilização de unidades de organização de programa vamos implementar nesta seção uma aplicação referente a um “Carro Automático para Máquina Operatriz” utilizando a linguagem LD. A figura a seguir ilustra esse sistema.

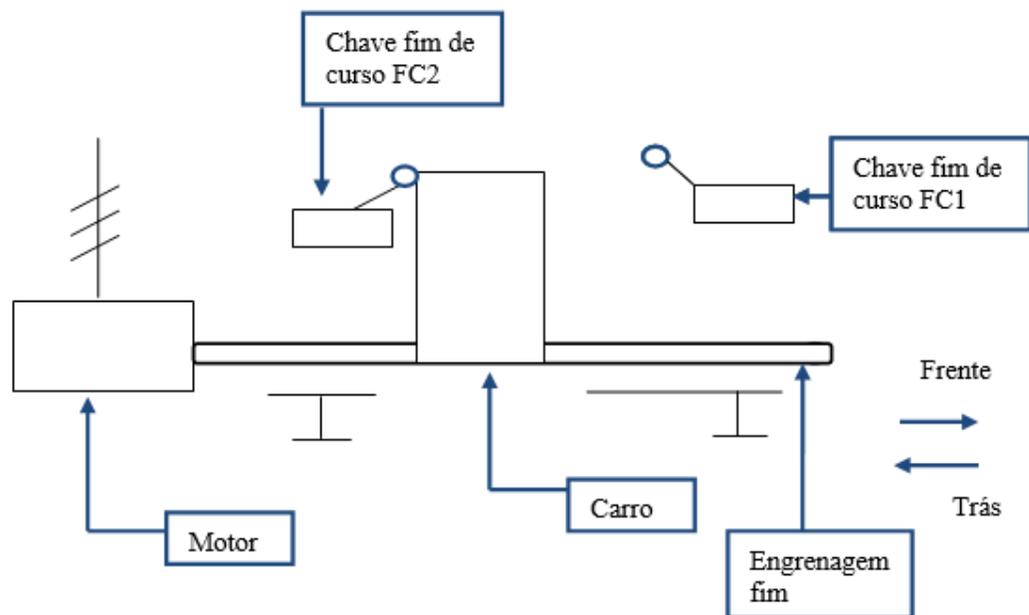


Figura 3-13. Aplicação “Carro Automático para Máquina Operatriz”

O sistema prevê dois modos de operação: automático e manual. No comando manual, pressionando-se o botão de avanço (S01) é energizado o contator de avanço do carro (K01). Por outro lado: S02 (botão de recuo) energiza K02 (contator de recuo). Neste modo, o avanço/recuo somente se processa pressionando-se o botão de parada (S03) antes de cada reversão. No comando automático, ao acionar o fim de curso de recuo (FCR) inicia a temporização de avanço (Temporizador\_A). Transcorrido um tempo de 10 s, o contator de avanço (K01) é energizado, movimentando o carro. Ao atingir o fim de curso de avanço (FCA) o motor para. Neste ponto inicia-se a temporização de recuo (Temporizador\_R). Transcorrido um tempo de 10 s, o contator de recuo (K02) é energizado, retornando o carro. Atingindo novamente o fim de curso de recuo (FCR) o ciclo se repete (contatores K01 e K02 desenergizados). O botão de STOP para tudo a qualquer momento.

A sequência para a criação da aplicação está indicada a seguir. Observe que as cinco primeiras etapas seguem o mesmo procedimento do estudo dirigido anterior.

Etapa 1: escolha do tipo de projeto.

Etapa 2: escolha dos módulos de hardware (dispositivo, bastidor e fonte de alimentação).

Etapa 3: escolha da quantidade de pontos que serão utilizados.

Etapa 4: seleção do perfil do projeto e a linguagem padrão.

Etapa 5: seleção das linguagens dos programas de tarefas comuns e conclusão do assistente.

Etapa 6: declaração das variáveis de alocação direta (entradas do CP).

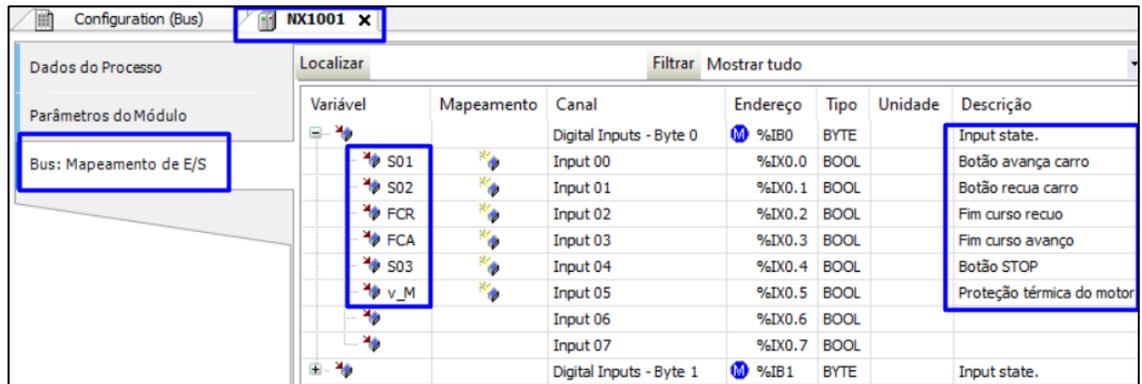


Figura 3-14. Declaração das Variáveis de Alocação Direta (Entradas do CP)

Etapa 7: declaração das variáveis de alocação direta (saídas do CP).

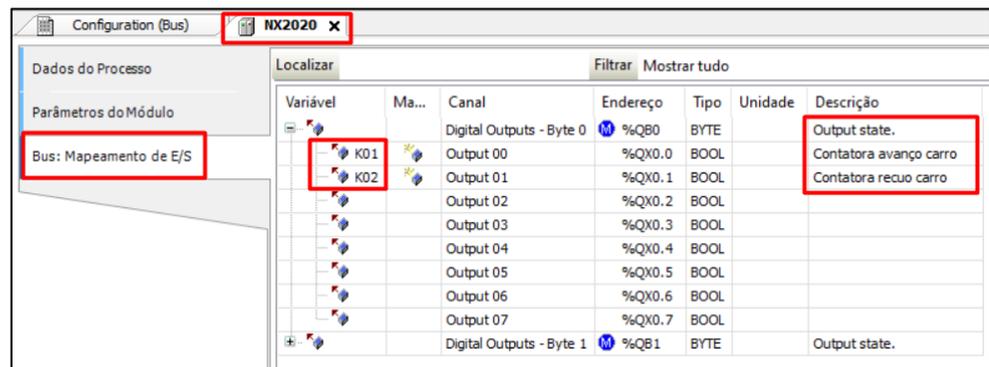


Figura 3-15. Declaração das Variáveis de Alocação Direta (Saídas do CP)

Etapa 8: declaração das variáveis locais no programa do usuário (UserProg).



Figura 3-16. Declaração das Variáveis Locais no Programa do Usuário (UserProg)

Etapa 9: edição das redes (implementação) do programa do usuário (UserProg).

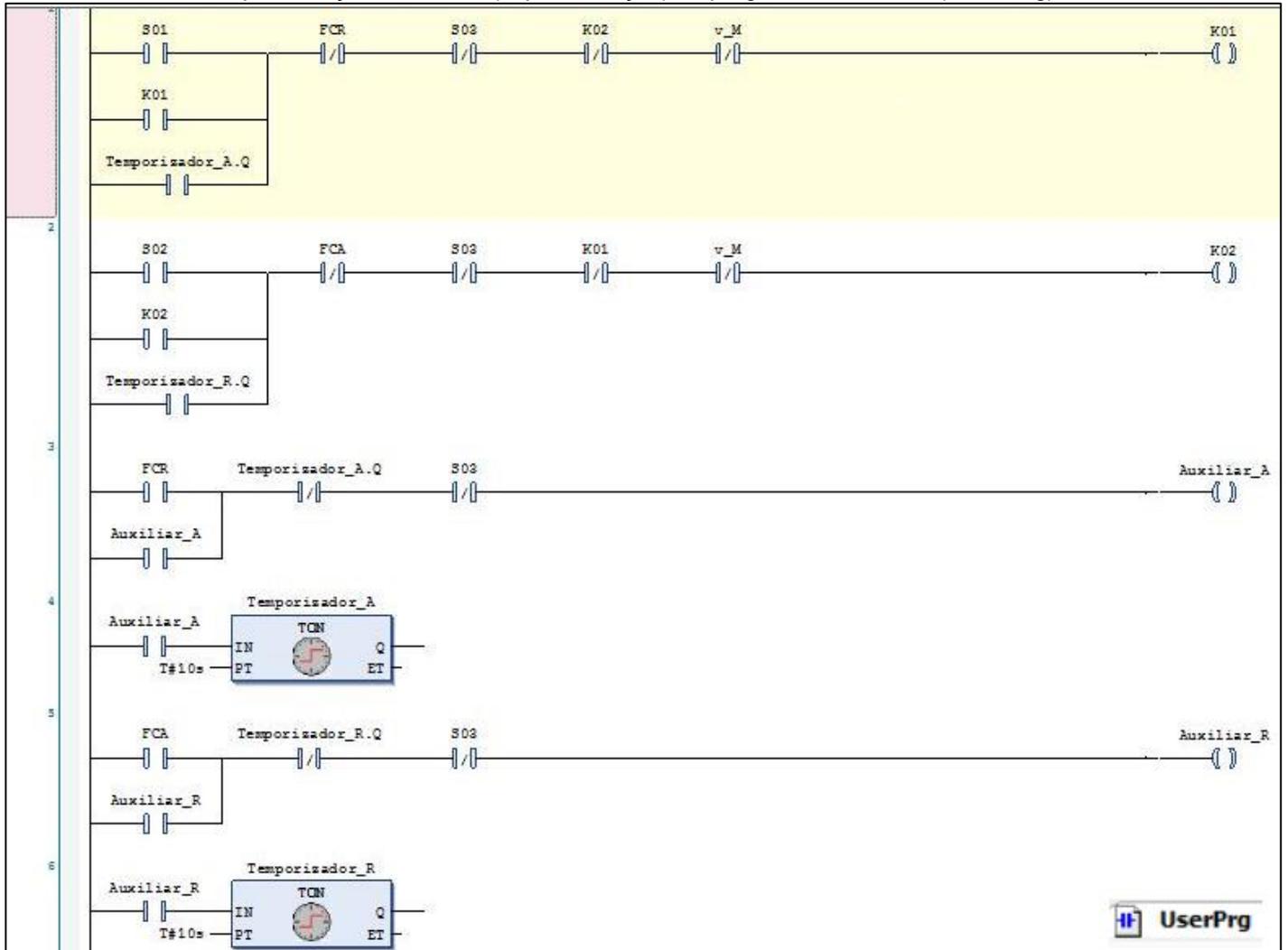


Figura 3-17. Edição das Redes (Implementação) do Programa de Usuário (UserProg)

Etapa 10: comentário das redes.

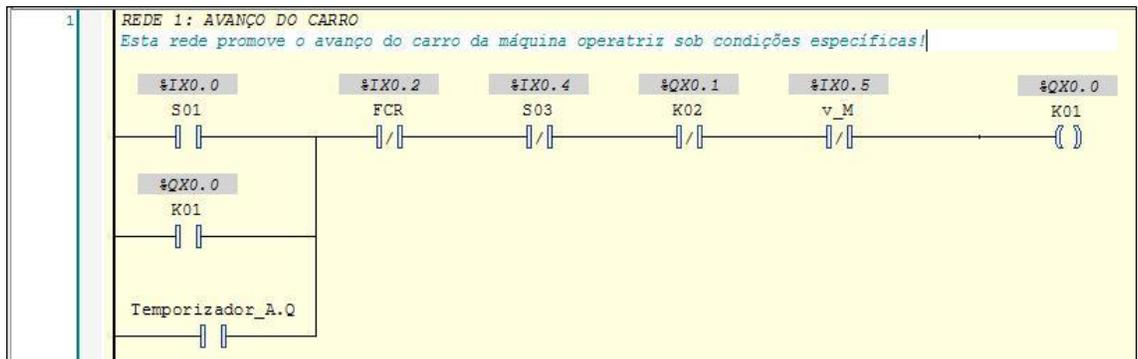


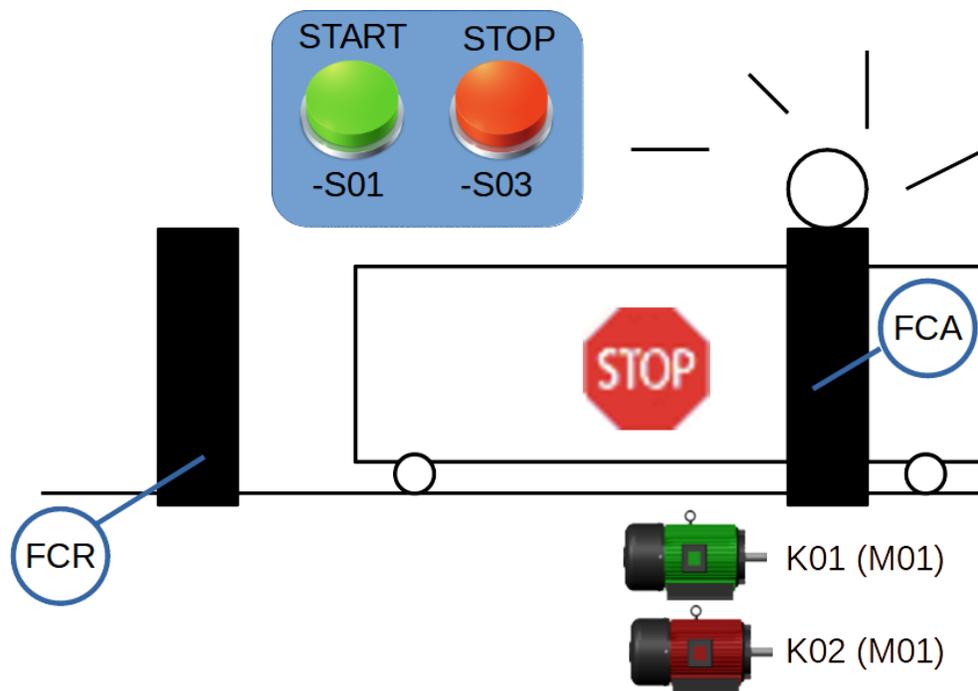
Figura 3-18. Comentário das Redes

### Estudo Dirigido 3-1: aplicação para controle de um portão automático



Desenvolva uma aplicação para controle de um “Portão automático com sistema de batente”, conforme figura e orientações abaixo. Trata-se de um sistema para abrir e fechar automaticamente um portão industrial. O funcionamento deste portão se dá de acordo com a operação descrita abaixo:

1. A bobina do contador K01 é energizada através do botão S01 de START (ou de um controle remoto). Esta ação movimenta a abertura do portão.
2. O portão parará quando o batente acionar mecanicamente a chave de fim de curso FCA.
3. O timer é ativado no período em que o portão permanece aberto. Finalizado este tempo a bobina do contador K02 de fechamento do portão é energizada.
4. Ao tocar a chave fim de curso FCR, o portão se fecha e o ciclo é concluído.
5. Um botão de STOP (S03) paralisa tudo a qualquer momento.



#### ANOTAÇÕES

ANOTAÇÕES

Resolução do Estudo Dirigido 3-1:

Etapa 1: escolha do tipo de projeto.

Etapa 2: escolha dos módulos de hardware (dispositivo, bastidor e fonte de alimentação).

Etapa 3: escolha da quantidade de pontos que serão utilizados.

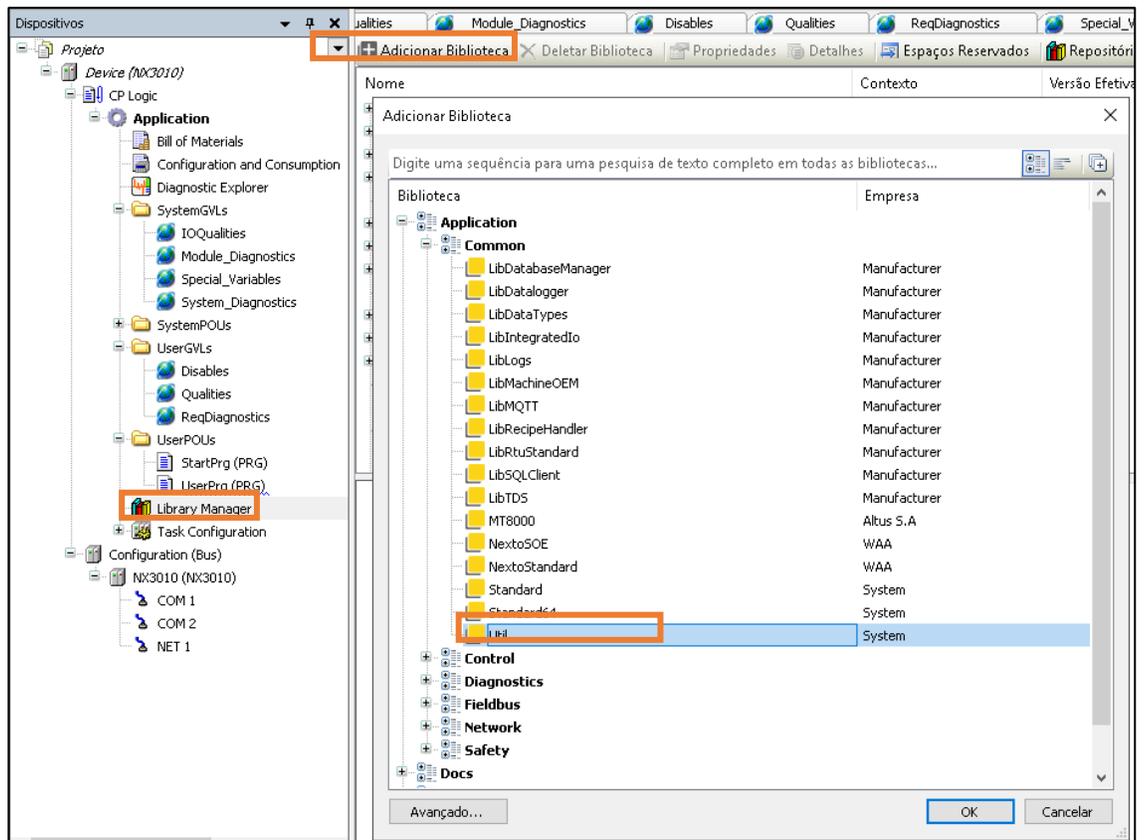
Etapa 4: seleção do perfil do projeto e a linguagem padrão.

Etapa 5: seleção das linguagens dos programas de tarefas comuns e conclusão do assistente.

Etapa 6: declaração das variáveis de alocação direta (entradas do CP).

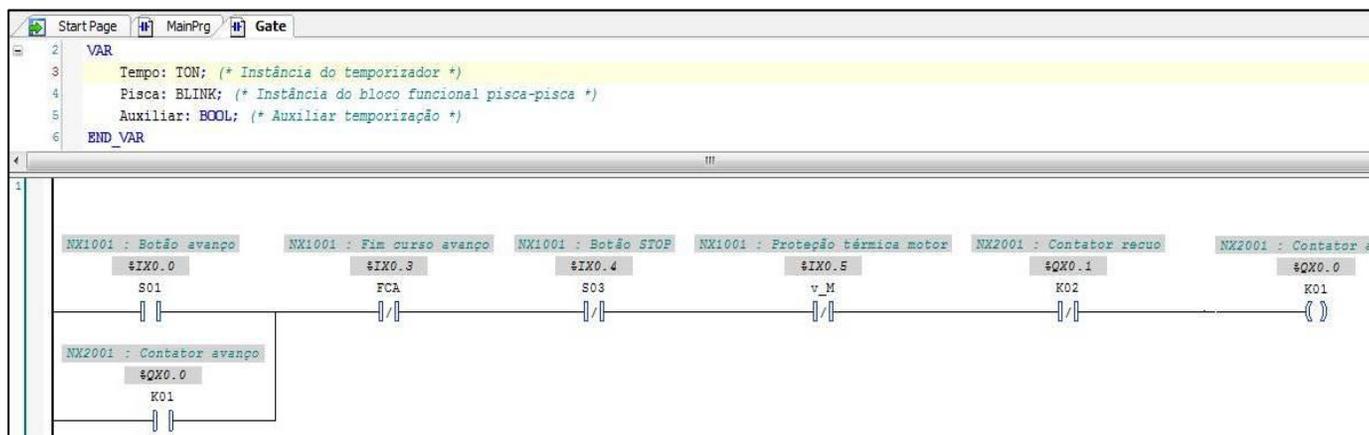
Etapa 7: declaração das variáveis de alocação direta (saídas do CP).

Etapa 8: inclusão da biblioteca "util.library" (para utilização do bloco funcional BLINK).

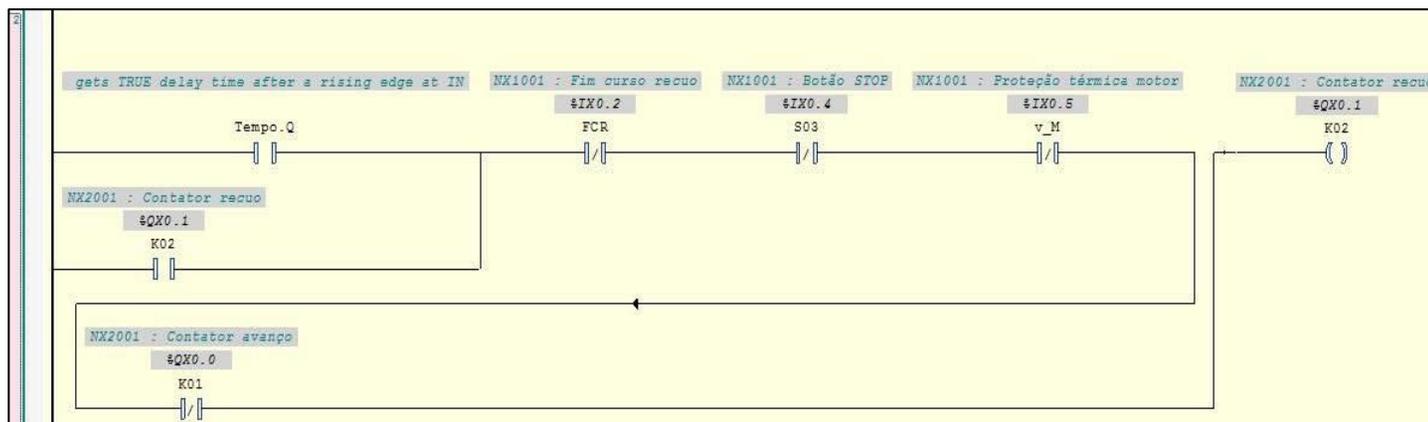


Etapa 9: edição das redes (implementação) do programa "GATE"

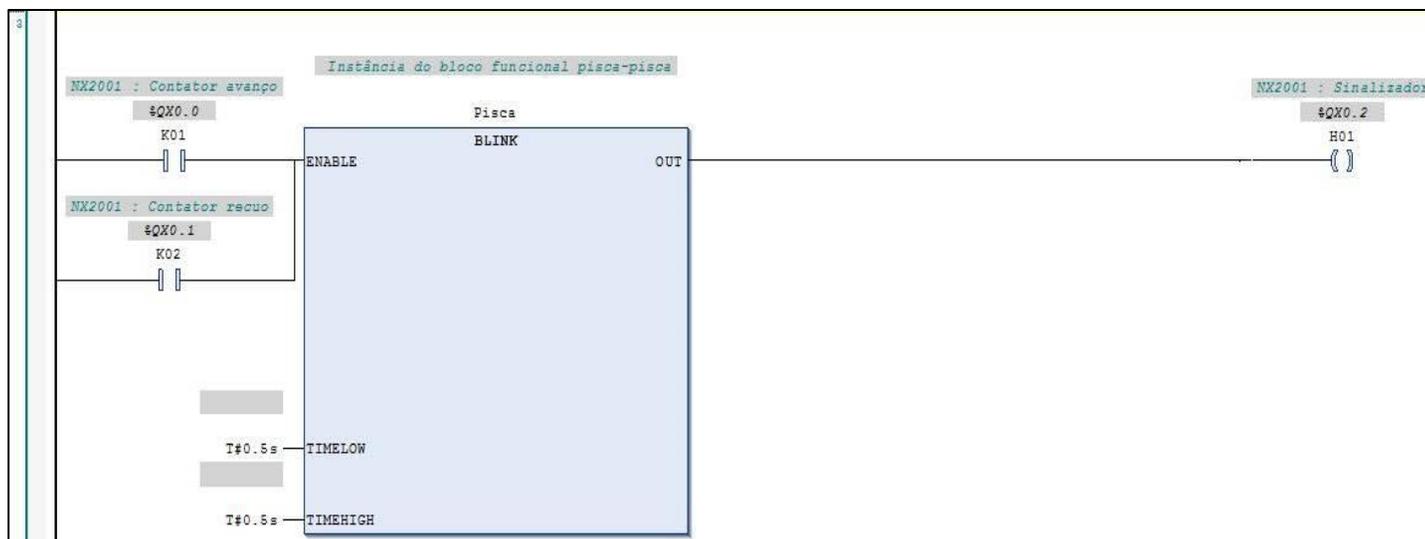
- Declaração das variáveis locais e rede 1



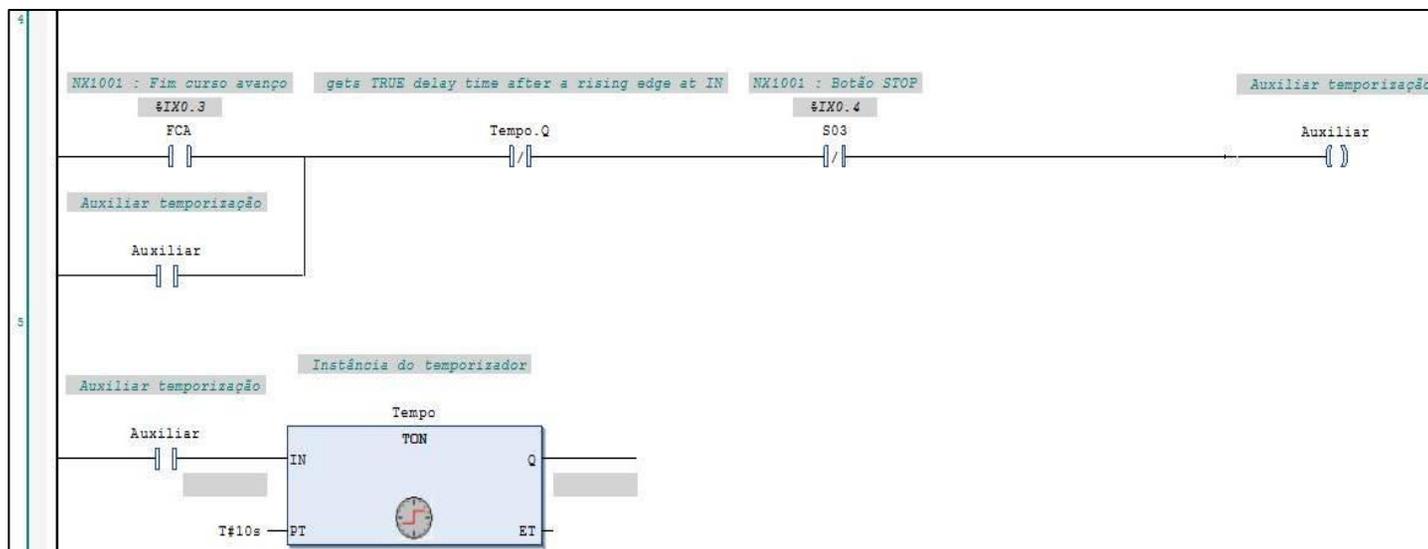
- Rede 2



## • Rede 3

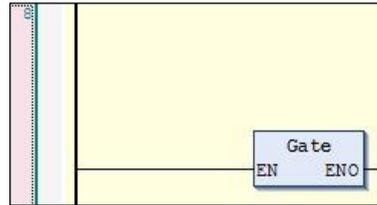


## • Rede 4



Etapa 10: comentário das redes.

Etapa 11: chamada do programa "GATE" no programa de usuário (UserProg)



### Utilização de blocos funcionais definidos pelo usuário

As POU's do tipo Bloco Funcional também podem ser criadas pelo usuário. Tais POU's são muito utilizadas para reutilização de código e permitem a estruturação de programas em conformidade com as diretrizes da norma IEC61131-3.

Para exemplificar a utilização desse tipo de POU vamos criar um bloco funcional para o controle de motores com partida reversível, utilizando a linguagem LD. O funcionamento do bloco funcional deve obedecer às seguintes diretivas:

- O motor elétrico pode ser acionado nos dois sentidos com reversão do sentido de rotação.
- O motor não deve entrar em funcionamento ou deve ser desligado no caso de sobrecarga no motor.
- O ligamento/desligamento do motor pode ser feito utilizando-se memórias biestáveis.
- Para evitar que o motor que se encontra em um sentido de rotação seja ligado em sentido contrário, deve ser incluído um intertravamento no circuito.
- O desligamento deve ser único.

Este Bloco Funcional será incluído no projeto "Carro Automático para Máquina Operatriz" desenvolvido anteriormente, o qual prevê a utilização de um motor reversível para movimentação do carro.

A utilização de blocos funcionais pode ser realizada seguindo a sequência descrita nas legendas das figuras a seguir.

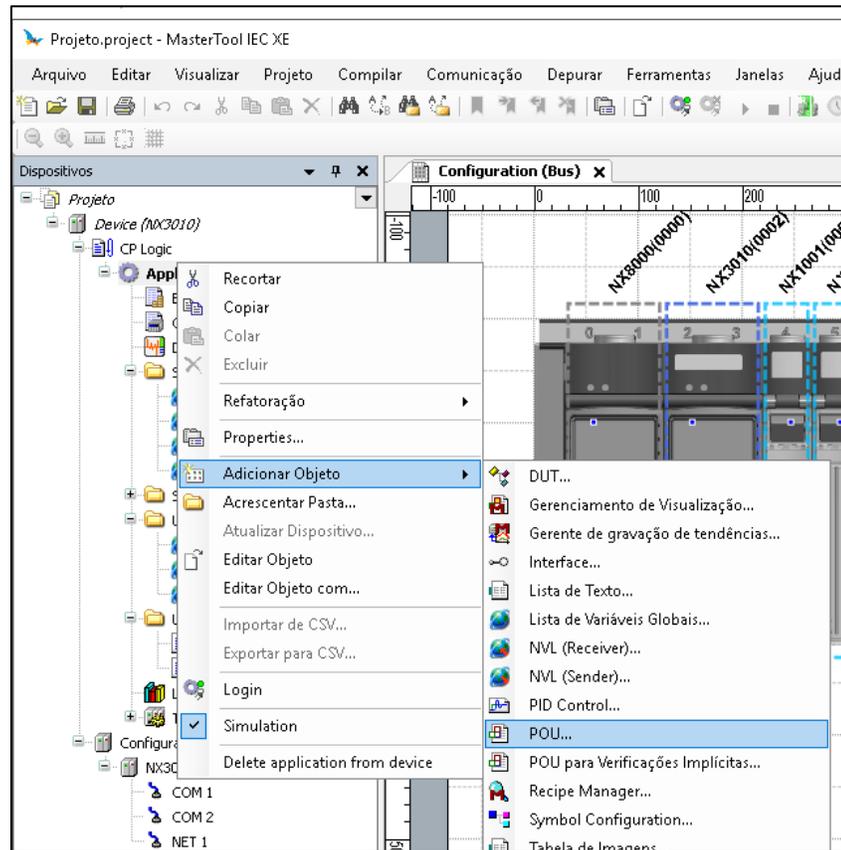


Figura 3-19. Inclusão da Nova POU na Árvore de Dispositivos (Etapa 1)

Acrescentar POU

 Criar uma nova POU (Unidade de Organização de Programa)

Nome  
Motor\_RV

Tipo

Programa  
 **Bloco funcional**  
 Estender  
 Implementar  
 Final     Abstrato

Especificador de acesso  
Diagrama de Lógica Ladder (LD)

Linguagem de implementação do método  
Texto Estruturado (ST)

**Função**  
 Tipo de retorno

Linguagem de implementação  
Diagrama de Lógica Ladder (LD)

Adicionar    Cancelar

Figura 3-20. Criação da POU Tipo Bloco Funcional *Motor\_RV* (Etapa 2)

```

1 FUNCTION_BLOCK Motor_RV
2 VAR_INPUT
3   LigD, LigE, Desl : BOOL; (* Comando do motor direita/esquerda/desliga *)
4   Sobrecarga      : BOOL; (* Sobrecarga no motor *)
5 END_VAR
6 VAR_OUTPUT
7   MotD, MotE : BOOL; (* Comando elétrico do motor direita/esquerda *)
8 END_VAR
9 VAR
10  MemoD : RS; (* Memoriza acionamento do motor à direita *)
11  MemoE : RS; (* Memoriza acionamento do motor à esquerda *)
12 END_VAR
  
```

Figura 3-21. Declaração das Variáveis (Etapa 3)

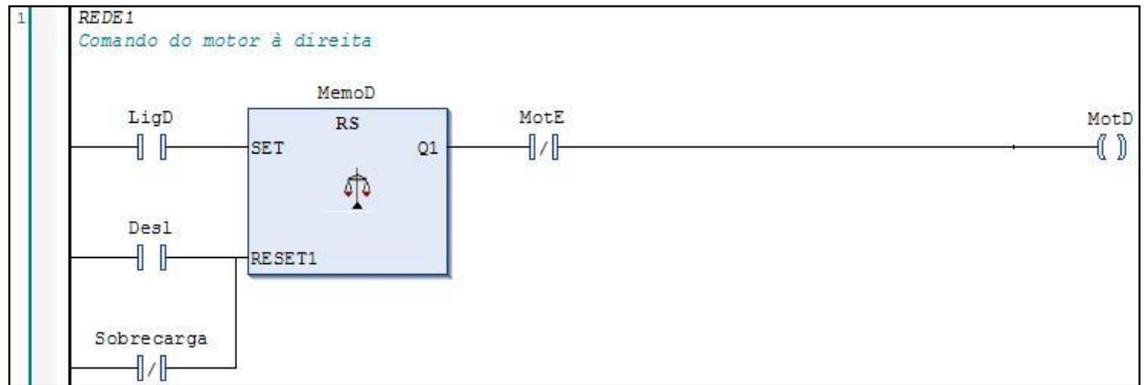


Figura 3-22. Edição do Corpo/Rede 1/ Comando à Direita (Etapa 4)

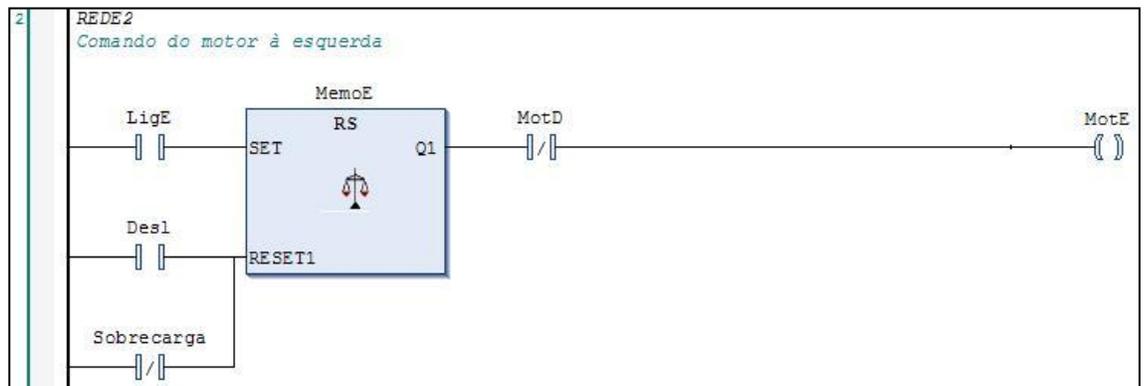


Figura 3-23. Edição do Corpo / Rede 2 / Comando à Esquerda (Etapa 5)

```

MainPrg
1  PROGRAM MainPrg
2  VAR
3      Temporizador_A : TON ;      (* Temporização de avanço *)
4      Temporizador_R : TON ;      (* Temporização de recuo *)
5      Auxiliar_A     : BOOL;      (* Variável auxiliar do avanço *)
6      Auxiliar_R     : BOOL;      (* Variável auxiliar do recuo *)
7      // Variáveis simbólicas adicionadas para comando do motor via bloco funcional Motor_RV...
8      Comando_A     : BOOL;      (* Comando de avanço do carro *)
9      Comando_R     : BOOL;      (* Comando de recuo do carro *)
10     Reversor      : Motor_RV;   (* Instância do bloco funcional de reversão do motor *)
11 END_VAR
    
```

Figura 3-24. Inclusão das Variáveis Simbólicas (Etapa 6)

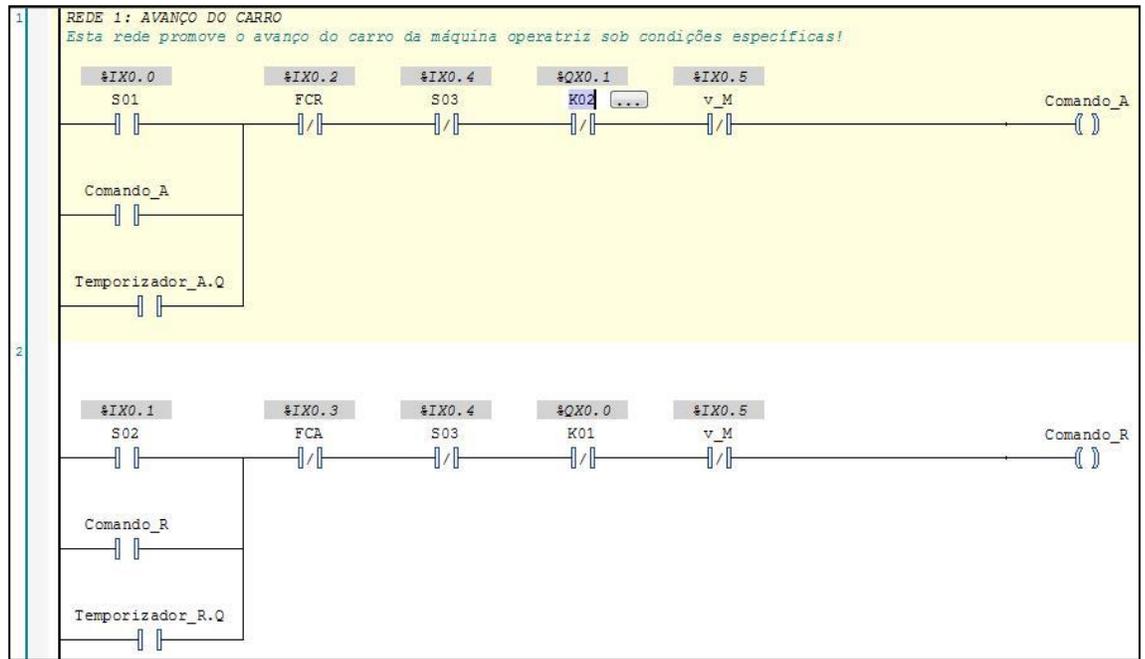


Figura 3-25. Alterações no Programa MainPrg (Etapa 7)

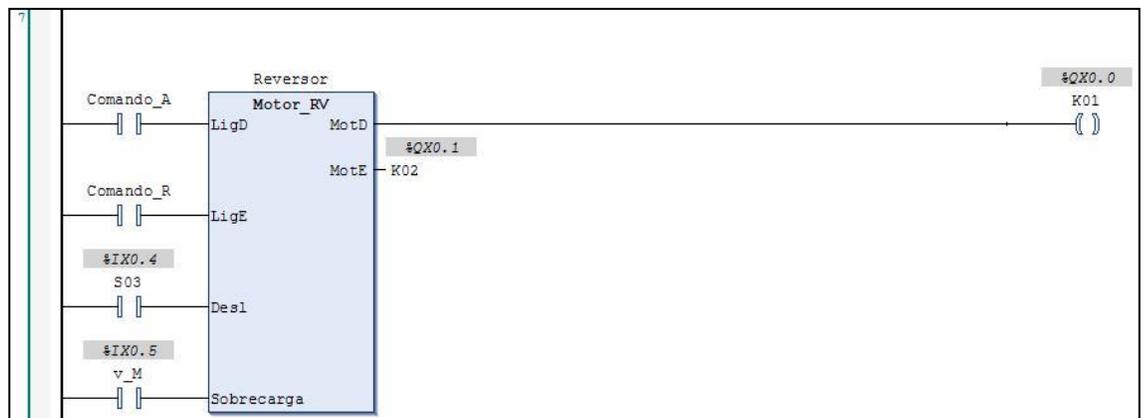


Figura 3-26. Chamada do Bloco Funcional Motor\_RV no Programa MainPrg (Etapa 8)

### Estudo Dirigido 3-2: criação de um bloco funcional



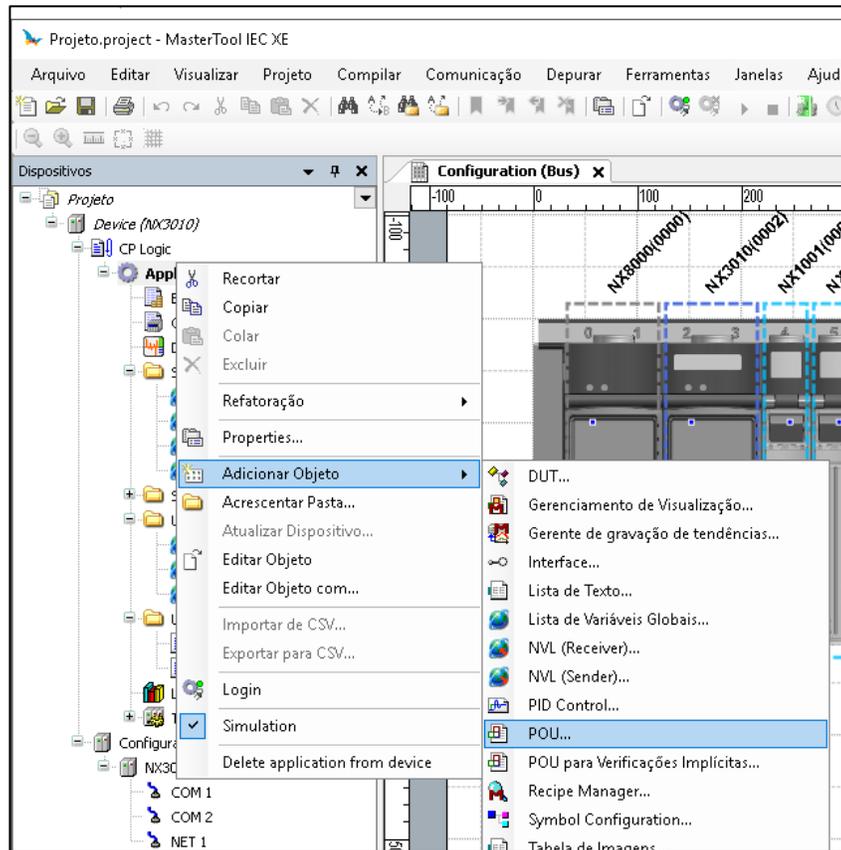
Criar um bloco funcional para o controle de motores com partida direta, utilizando a linguagem LD. O funcionamento do bloco funcional deve obedecer às seguintes diretrizes:

Deve ser previsto o controle do motor nos modos manual e automático via variável MAN\_Auto. Em manual (MAN\_Auto = TRUE) o motor pode ser comandado pelas entradas Liga/Desl. Em automático o motor é comandado (ligado/desligado) unicamente pela entrada Libera, a qual permite que lógicas externas possam ligar e desligar o motor de acordo com as necessidades de intertravamento. As saídas do bloco são: Ligar para comando do contator do motor e Falha que, em TRUE, indica algum defeito no motor (sobrecarga, problema na partida, etc.). A entrada "Funcionando" serve para detectar que o motor entrou em funcionamento após a saída "Ligar" ser acionada. Se esta entrada não estiver em TRUE 1,5 s depois do acionamento do motor - monitorado pela instância Retardo do bloco funcional TON - a saída de falha será acionada. A instância Pulso do bloco funcional R\_TRIG assegura que somente a transição do sinal será utilizada para ligar o motor em manual.

#### ANOTAÇÕES

Resolução do Estudo Dirigido 3-2:

Etapa 1: acrescentar POU



## Etapa 2: definir configurações do bloco funcional

Acrescentar POU ✕

 Criar uma nova POU (Unidade de Organização de Programa)

Nome  
Motor\_PD|

Tipo

Programa

**Bloco funcional**

Estender  ...

Implementar  ...

Final  Abstrato

Especificador de acesso  
 ▾

Linguagem de implementação do método  
Texto Estruturado (ST) ▾

Função

Tipo de retorno  ...

Linguagem de implementação  
Diagrama de Lógica Ladder (LD) ▾

## Etapa 3: declarar variáveis

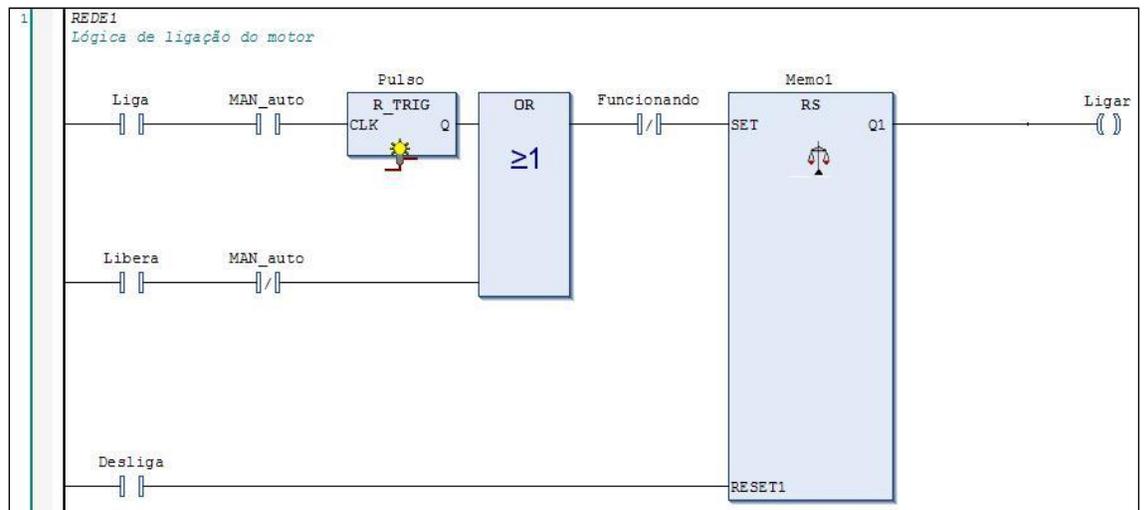
```

1  FUNCTION_BLOCK Motor_PD
2  VAR_INPUT
3      Liga, Desl : BOOL; (* Comando em Manual *)
4      Libera    : BOOL; (* Comando em Automático *)
5      Funcionando : BOOL; (* Motor funcionando *)
6      MAN_auto  : BOOL; (* Manual = TRUE & Automático = FALSE *)
7      Defeito   : BOOL; (* Defeito no motor *)
8      Sobrecarga : BOOL; (* Sobrecarga no motor *)
9  END_VAR
10 VAR_OUTPUT
11     Ligar : BOOL; (* Comando elétrico *)
12     Falha : BOOL; (* Motor em falha *)
13 END_VAR
14 VAR
15     Pulso : R_TRIG; (* Transição para ligar em manual *)
16     Mem01 : RS;     (* Memoriza acionamento *)
17     Retardo : TON; (* Retardo alarme partida *)
18     Desliga : BOOL; (* Lógica de desligamento *)
19 END_VAR
20

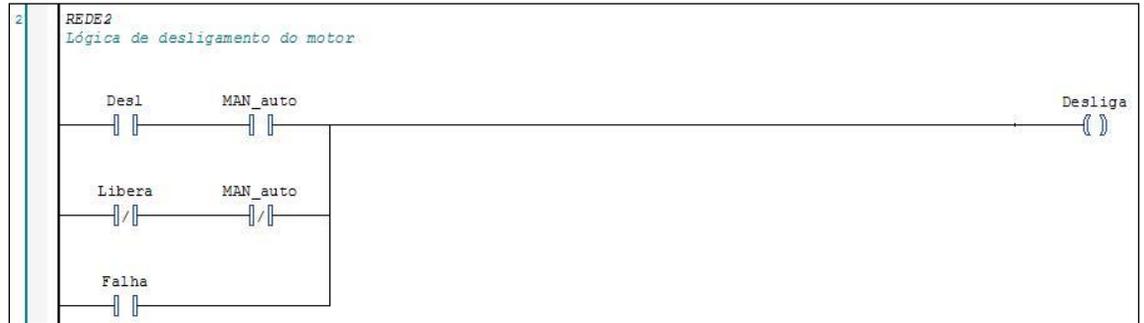
```

## Etapa 4: implementar bloco funcional

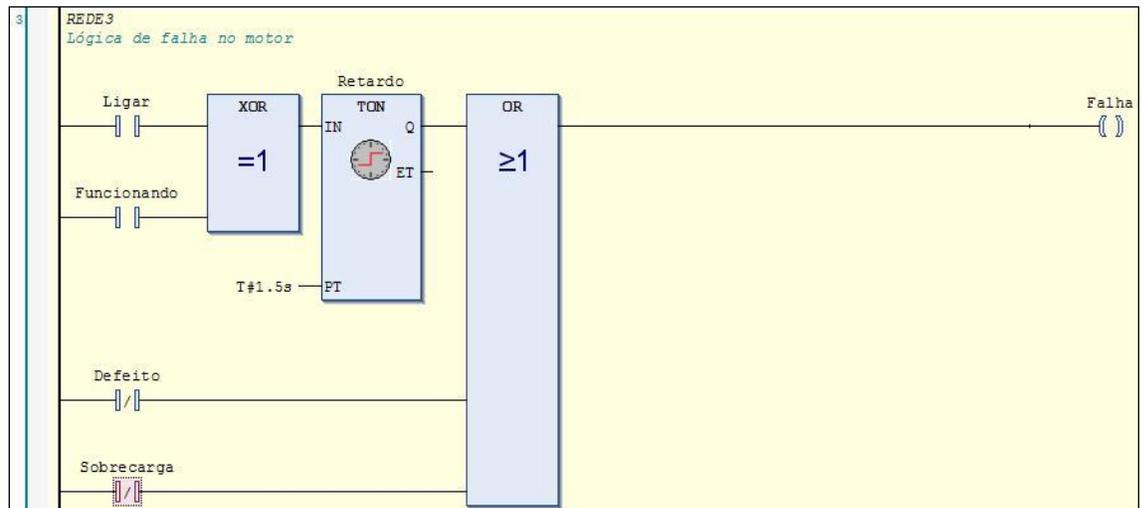
- Rede 1



- Rede 2



- Rede 3



### Estudo Dirigido 3-3: funções de diagnóstico no Nexto

Nesta aplicação veremos como utilizar Unidades de Organização de Programa do tipo Função através da programação de funções de diagnóstico no Nexto.

Estas funções proporcionam a visualização de alguns parâmetros que não podem ser acessados de outra maneira. As três funções sobre diagnósticos avançados estão localizadas na biblioteca Nexto e estão descritas abaixo.

ANOTAÇÕES

Resolução do Estudo Dirigido 3-3: consulte o Manual de Utilização Série Nexto – MU214000 e descreva a função dos parâmetros utilizados que estão indicados nas tabelas.

1. GetNoOfApplication: essa função retorna o número total de aplicações válidas que estão na UCP Nexto.



Parâmetros de entrada	Qual a sua função?
pwNoOfApp	

2. GetAppInfo: essa função retorna informações sobre a aplicação desejada.



Parâmetros de entrada	Qual a sua função?
psAppName	
pstAppInfo	

Parâmetros retornados	Qual a sua função?
byAppStatus	Status da Aplicação 1: RUN: _____ 2: STOP: _____ 3: HALT_ON_BP: _____ 4: _____ 5: _____
byInitStatus	Reporta o tipo de inicialização da aplicação: 1: Warm-Start: _____ 2: Cold-Start: _____
wColdStartCounter	
wWarmStartCounter	
wNumOfTasks	

3. GetTaskInfo: essa função retorna informações sobre uma tarefa de uma determinada aplicação.

```

GetTaskInfo
psAppName  POINTER TO STRING      ERRORCODE GetTaskInfo
psTaskName  POINTER TO STRING
pstTaskInfo POINTER TO stTaskInfo
    
```

Parâmetros de entrada	Qual a sua função?
psAppName	
psTaskName	
pstTaskInfo	

Parâmetros retornados	Qual a sua função?
dwCurScanTime	
dwMinScanTime	
dwMaxScanTime	
dwAvgScanTime	
dwLimitMaxScan	

## Um pouco de teoria... Programação Padronizada Multi-linguagens



Desde o advento dos controladores programáveis, muitas linguagens têm sido utilizadas para escrever programas para máquinas e processos.

O resultado desta falta de padronização acaba se refletindo na necessidade de treinamentos em diferentes equipamentos e formação de equipes de manutenção específicas em determinados fabricantes. A consequência direta, muitas vezes não percebida pelos usuários, é a perda de tempo e dinheiro.

Para atenuar este problema, um grupo formado pela organização internacional IEC (*International Electrotechnical Commission*) definiu uma norma para vários aspectos dos controladores, desde características do hardware, instalação, testes, comunicação e programação.

Especificamente a norma IEC61131-3 (parte 3) estabelece as principais características para programação de controladores. Estas características definem o modelo de software e cobre as 5 linguagens mais utilizadas em todo mundo: Function Block Diagram (FBD), Ladder Diagram (LD), Sequential Function Chart (SFC), Structured Text (ST) e Instruction List (IL).

Dentre as principais vantagens da norma podemos destacar a facilidade que o usuário tem em modularizar e estruturar a programação em elementos funcionais ou "POUs" (Program Organization Units), bem como poder definir a linguagem em que irá programar determinada parte do projeto, além de estar utilizando um ambiente de programação world-wide onde o usuário, aprendendo as linguagens da norma, poderá usar este conhecimento em diferentes ambientes de programação (fabricantes). Além disso, o modelo de software permite a reutilização de código através da utilização de biblioteca de blocos funcionais, facilitando o desenvolvimento, implantação e manutenção dos sistemas e aumentando a qualidade do software.

Os programas ou parte deles poderão ser usados entre os ambientes de programação através da importação e exportação de módulos.

# 4. Aplicações, Avaliação e Encerramento

## Aplicação 1: Controle Discreto



### Proposição da Atividade

Utilizando os conhecimentos desenvolvidos no treinamento implemente a automação do equipamento indicado a seguir.

Um equipamento para estampar peças plásticas é formado por um dispositivo de carregamento de peças (por gravidade), um cilindro A (alimentador), um cilindro B (estampador) e um cilindro C (extrator). Todos os três cilindros são de ação simples com retorno por mola, e têm seu avanço comandado pelas eletroválvulas Y1, Y2 e Y3 respectivamente. A máxima excursão de cada cilindro é monitorada pela atuação dos sensores S1, S2 e S3 do tipo *reed-switch*. A expulsão da peça é realizada por um sopro de ar comprimido, obtido a partir do acionamento da eletroválvula Y4, e efetivamente monitorada pela atuação do fotossensor (SOT). O funcionamento prevê como condição inicial que os cilindros não estejam avançados, ou seja, essa condição traduz que todas as eletroválvulas estejam desligadas. Assim, com a chave de partida (S0) acionada e estando a máquina na condição inicial, deve-se iniciar a operação. A sequência consiste em, primeiramente, colocar uma peça no molde, recuar o êmbolo do cilindro alimentador, prensar o estampo sobre a peça (deve-se aguardar um tempo de cinco segundos com a peça sendo prensada), atuar o extrator e o bico de ar para retirada da peça pronta. A figura a seguir ilustra o sistema.

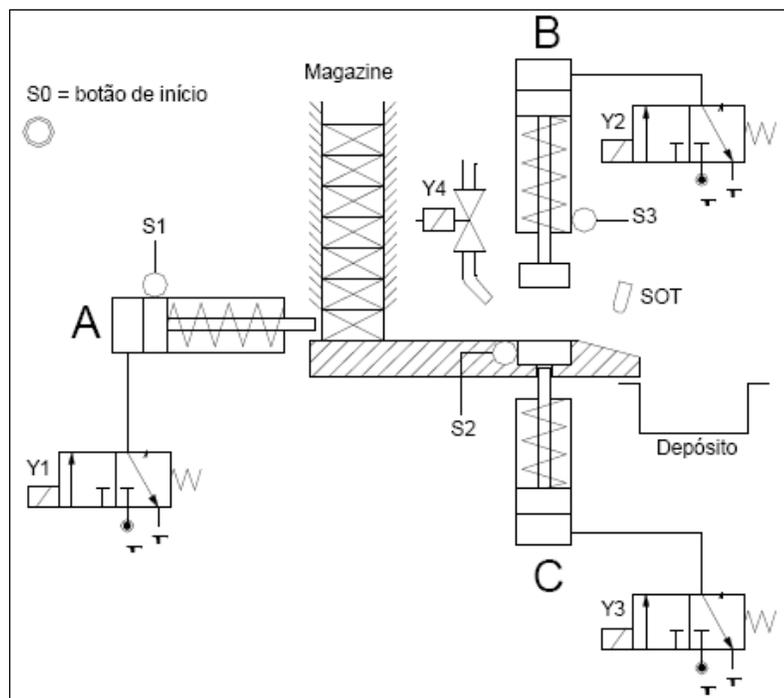


Figura 4-1. Sistema de Controle Discreto



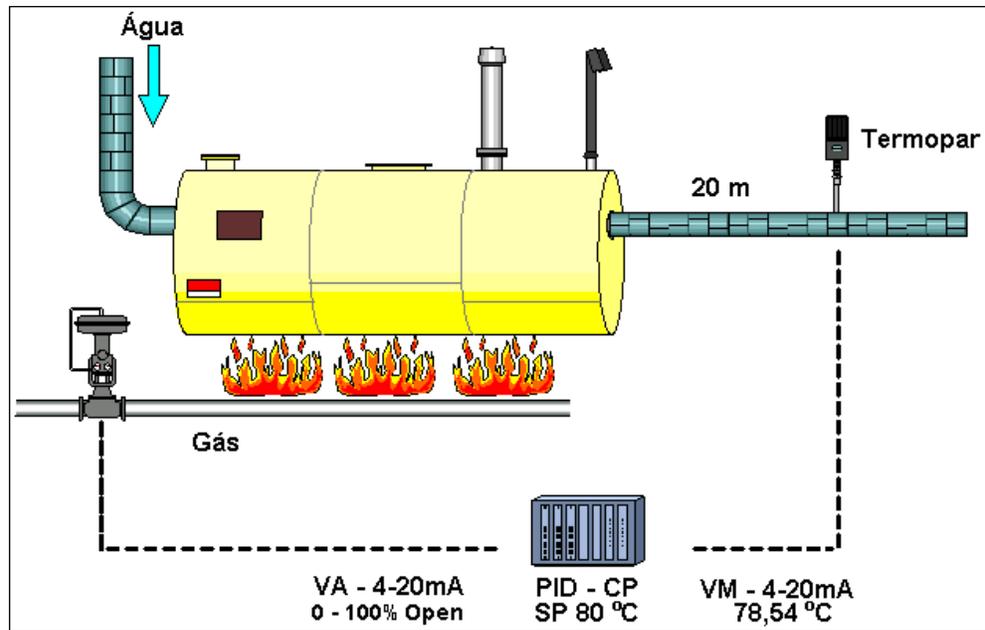


Figura 4-2. Sistema de Controle Contínuo

Na figura anterior, o transmissor de temperatura (Termopar) fica próximo ao consumo, o qual está situado a 20 metros do ponto de aquecimento da água. Processos como este são bons exemplos de como podem ser introduzidos “tempos mortos”. Isto porque a água aquecida no ponto de aquecimento leva algum tempo para percorrer a distância até o ponto de medição junto ao consumidor.

Assume-se que a água que chega a ponto de aquecimento sobre o queimador tem temperatura fixa de 30 °C.

Assume-se ainda que a vazão de água e a pressão do gás são constantes.

Algumas características do processo e dos seus elementos utilizados:

- A água aquecida deve ter sua temperatura programável entre 50 °C e 80 °C.
- O transmissor de temperatura (Termopar) tem saída de 4–20 mA, e se comporta de forma linear, de tal maneira que 4 mA correspondem a 30 °C e 20 mA correspondem a 130 °C.
- Para aumentar em 10 °C a temperatura da água, é necessário injetar 1m<sup>3</sup>/h de gás, este comportamento é linear.
- A válvula de gás se fecha com 4 mA, injetando 0 m<sup>3</sup>/h de gás. Por outro lado, com 20 mA, ela injeta 8 m<sup>3</sup>/h de gás, este comportamento é linear.

### ANOTAÇÕES




# Avaliação do Treinamento



Preencha a ficha de avaliação do treinamento.

## **Parabéns...**

Se você seguiu a sequência indicada no tutorial e realizou as tarefas propostas com sucesso, você atingiu plenamente os objetivos desse treinamento!!!

# Glossário

<b>Algoritmo</b>	Sequência finita de instruções bem definidas, objetivando à resolução de problemas.
<b>Árvore</b>	Estrutura de dados para configuração do hardware.
<b>Backup</b>	Cópia de segurança de dados.
<b>Barramento</b>	Conjunto de módulos de E/S interligados a uma UCP ou cabeça de rede de campo.
<b>Barramento local</b>	Conjunto de módulos de E/S interligados a uma UCP.
<b>Barramento remoto</b>	Conjunto de módulos de E/S interligados a uma cabeça de rede de campo.
<b>Bit</b>	Unidade básica de informação, podendo estar no estado 0 ou 1.
<b>Breakpoint</b>	Ponto de parada no aplicativo para depuração.
<b>Byte</b>	Unidade de informação composta por oito bits.
<b>Canal serial</b>	Interface de um equipamento que transfere dados no modo serial.
<b>Ciclado</b>	Modo de execução do CP passo-a-passo, onde cada passo é um ciclo do CP.
<b>Ciclo de varredura</b>	Uma execução completa do programa aplicativo de um controlador programável.
<b>Circuito de cão-de-guarda</b>	Circuito eletrônico destinado a verificar a integridade do funcionamento de um equipamento.
<b>Controlador programável</b>	Também chamado de CP. Equipamento que realiza controle sob o comando de um programa aplicativo. É composto de uma UCP, uma fonte de alimentação e uma estrutura de E/S.
<b>CP</b>	Veja controlador programável.
<b>Default</b>	Valor predefinido para uma variável, utilizado em caso de não haver definição.
<b>Diagnóstico</b>	Procedimento utilizado para detectar e isolar falhas. É também o conjunto de dados usados para tal determinação, que serve para a análise e correção de problemas.
<b>Download</b>	Carga de programa ou configuração no CP.
<b>E/S</b>	Veja entrada/saída.
<b>Entrada/saída</b>	Também chamado de E/S. Dispositivos de E/S de dados de um sistema. No caso de CPs, correspondem tipicamente a módulos digitais ou analógicos de entrada ou saída que monitoram ou acionam o dispositivo controlado.
<b>Escravo</b>	Equipamento ligado a uma rede de comunicação que só transmite dados se for solicitado por outro equipamento denominado mestre.
<b>Frame</b>	Uma unidade de informação transmitida na rede.
<b>Gateway</b>	Equipamento ou software para a conexão de duas redes de comunicação com diferentes protocolos.
<b>Hardware</b>	Equipamentos físicos usados em processamento de dados onde normalmente são executados programas (software).
<b>Hiperlink</b>	Atalho de navegação para uma nova página do help.
<b>IEC 61131</b>	Norma genérica para operação e utilização de CPs. Antiga IEC 1131.
<b>Interface</b>	Dispositivo que adapta elétrica e/ou logicamente a transferência de sinais entre dois equipamentos.
<b>Interrupção</b>	Evento com atendimento prioritário que temporariamente suspende a execução de um programa e desvia para uma rotina de atendimento específica
<b>kbytes</b>	Unidade representativa de quantidade de memória. Representa 1024 bytes.
<b>LED</b>	Sigla para light emitting diode. É um tipo de diodo semiconductor que emite luz quando estimulado por eletricidade. Utilizado como indicador luminoso.
<b>Linguagem de programação</b>	Um conjunto de regras e convenções utilizado para a elaboração de um programa.
<b>Local host</b>	Máquina, PC ou sistema que está em uso.
<b>Login</b>	Ação de estabelecer um canal de comunicação com o CP.
<b>Menu</b>	Conjunto de opções disponíveis e exibidas por um programa no vídeo e que podem ser selecionadas pelo usuário a fim de ativar ou executar uma determinada tarefa.
<b>Menu de Contexto</b>	Menu dinâmico com o conteúdo de acordo com o contexto atual.
<b>Mestre</b>	Equipamento ligado a uma rede de comunicação de onde se originam solicitações de comandos para outros equipamentos da rede.
<b>Módulo (referindo-se a hardware)</b>	Elemento básico de um sistema completo que possui funções bem definidas. Normalmente é ligado ao sistema por conectores, podendo ser facilmente substituído.
<b>Módulo de E/S</b>	Módulo pertencente ao subsistema de entradas e saídas.
<b>Multicast</b>	Disseminação simultânea de informação a um determinado grupo de nós interligados a uma rede de comunicação.

---

<b>Nó</b>	Qualquer estação de uma rede com capacidade de comunicação utilizando um protocolo estabelecido.
<b>Operandos</b>	Elementos sobre os quais as instruções atuam. Podem representar constantes, variáveis ou um conjunto de variáveis.
<b>PC</b>	Sigla para programmable controller. É a abreviatura de controlador programável em inglês.
<b>PDO</b>	Do inglês , é uma mensagem do protocolo CAN contendo os dados operacionais.
<b>Ping</b>	Do inglês , é um comando usado pelo protocolo ICMP que serve para testar a conectividade entre equipamentos e foi criado para o uso em redes com a pilha de protocolo TCP/IP.
<b>POU</b>	, ou Unidade de Organização de Programa, é uma subdivisão do programa aplicativo que pode ser escrito em qualquer uma das linguagens disponíveis.
<b>Programa aplicativo</b>	É o programa carregado em um CP, que determina o funcionamento de uma máquina ou processo.
<b>Protocolo</b>	Regras de procedimentos e formatos convencionais que, mediante sinais de controle, permitem o estabelecimento de uma transmissão de dados e a recuperação de erros entre equipamentos.
<b>RAM</b>	Sigla para random access memory. É a memória onde todos os endereços podem ser acessados diretamente de forma aleatória e com a mesma velocidade. É volátil, ou seja, seu conteúdo é perdido quando o equipamento é desenergizado, a menos que se possua uma bateria para a retenção dos valores.
<b>Rede de comunicação</b>	Conjunto de equipamentos (nós) interconectados por canais de comunicação.
<b>Reset</b>	Comando para reinicializar o CP.
<b>RUN</b>	Comando para colocar o CP em modo de execução.
<b>Set</b>	Ação para atribuir o estado de nível lógico alto para uma variável booleana.
<b>Software</b>	Programas de computador, procedimentos e regras relacionadas à operação de um sistema de processamento de dados.
<b>STOP</b>	Comando para congelar o CP em seu estado atual.
<b>Sub-rede</b>	Segmento de uma rede de comunicação que interliga um grupo de equipamentos (nós) com o objetivo de isolar o tráfego local ou utilizar diferentes protocolos ou meio físicos.
<b>Time-out</b>	Tempo preestabelecido máximo para que uma comunicação seja completada. Se for excedido procedimentos de retentiva ou diagnóstico serão ativados.
<b>Tooltip</b>	Caixa de texto com uma ajuda ou local onde pode-se entrar com a ajuda.
<b>UCP</b>	Sigla para unidade central de processamento. Controla o fluxo de informações, interpreta e executa as instruções do programa e monitora os dispositivos do sistema.
<b>UCP ativa</b>	Em um sistema redundante, a UCP ativa realiza o controle do sistema, lendo os valores dos pontos de entrada, executando o programa aplicativo e acionando os valores das saídas.
<b>Upload</b>	Leitura do programa ou configuração do CP.
<b>Visualização</b>	Conjunto de telas do CP.
<b>XML</b>	Do inglês, Extensible Markup Language, é um padrão para gerar linguagens de marcação.
<b>Zoom</b>	No contexto da janela de função do teclado, é utilizado para a troca de telas.