

altus

www.altus.com.br

APOSTILA DE
TREINAMENTO

**SCADA/HMI
BLUEPLANT**



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).

DIREITOS AUTORAIS

Nexto, MasterTool e BluePlant 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. Adicionalmente ao código fonte, todos os termos da licença, condições de garantia e informações sobre direitos autorais podem ser disponibilizadas sob requisição.

Sumário

Introdução	10
Um pouco de teoria... O que são sistemas SCADA?	11
O Curso MP5	12
Estruturação do Tutorial	13
Ementa do curso	13
Referências	16
Documentos relacionados	16
Avaliação do curso	16
Terminologia	16
Convenções Utilizadas	17
Suporte Técnico	18
Base de Conhecimento ALTUS	19
Cursos ALTUS	19
Revisões deste Tutorial	19
Training Box Nexto Xpress	20
1. Características Técnicas	1-1
Características Gerais do BluePlant	1-1
O que é o BluePlant?	1-1
Porque o BluePlant?	1-2
Quais São Seus Diferenciais?	1-2
Quais são as principais características do produto?	1-4
Características Avançadas do BluePlant	1-5
Software com Segurança Intrínseca	1-5
Mecanismo Gráfico Superior	1-5
Recursos de Manutenção, Teste e Diagnóstico Avançado	1-5
Servidores e Extensões .NET Incorporados	1-5
Banco de Dados em Tempo Real (Tags)	1-6
Linguagens .NET e Scripts	1-6
Alarmes e Segurança	1-6
Tendência e Historiadores	1-6
Dispositivos e Redes	1-6
Conjunto de Dados	1-7
Relatórios	1-7
Telas do Cliente	1-7
Objetos do <i>Runtime</i>	1-8
Isolação de Módulo	1-8
Ferramentas do <i>Runtime</i> e Diagnóstico	1-8
Implantação e Teste de Projeto	1-8
Drivers de Comunicação	1-9
Instalação do BluePlant	1-10
Licenças e Hardkey	1-16
2. Características Operacionais	2-1
Primeiros Passos com o BluePlant	2-1
Gerenciamento de Projeto	2-1
Abrindo ou Criando Projetos	2-3
Estudo Dirigido 2-1: visão geral das funcionalidades do BluePlant	2-3

Estudo Dirigido 2-2: primeiros passos para utilizar o BluePlant	2-4
Criando um Novo Projeto	2-4
Interface do Usuário	2-5
Itens do Menu Principal	2-7
Implementação de um Projeto no BluePlant	2-12
Criando Tags no Projeto	2-12
Incluindo Objetos na Tela Principal	2-15
Executando um Projeto	2-19
Estudo Dirigido 2-3: projeto de um sistema de supervisão	2-25
3. Editando a Aplicação	3-1
Menu Edit	3-1
Editando Tags	3-2
Edição de Objetos Tag (<i>Objects</i>)	3-2
Edição de Tipos Definidos pelo Usuário (<i>Templates</i>)	3-6
Edição de Níveis (<i>Assets</i>)	3-8
Edição de Historiador (<i>Historian</i>)	3-8
Tabelas do Historiador (<i>HistorianTables</i>)	3-10
Editando Configurações de Segurança	3-12
Edição de Usuários (<i>Users</i>)	3-12
Edição de Permissões (<i>Permissions</i>)	3-13
Edição de Políticas (<i>Policies</i>)	3-14
Visualização Usuários (<i>RuntimeUsers</i>)	3-15
Editando Dispositivos	3-16
Canais (<i>Channels</i>)	3-16
Pontos (<i>Points</i>)	3-20
Tipo de Acesso (<i>AcessType</i>)	3-24
Editando Alarmes	3-27
Grupos de Alarme (<i>Groups</i>)	3-27
Áreas de Alarme (<i>Areas</i>)	3-29
Itens de Alarme (<i>Items</i>)	3-30
Editando Banco de Dados	3-33
DBs	3-33
Estudo Dirigido 3-1: conexão com bases de dados	3-34
Tables	3-35
Queries	3-36
Files	3-37
Editando Scripts	3-39
Tasks	3-39
Classes	3-40
CodeEditor	3-41
Um pouco de teoria... Linguagem de Programação VB e C#	3-42
Linguagem VB	3-42
Linguagem C#	3-43
Expressions	3-45
Estudo Dirigido 3-2: envio de e-mail com autenticação	3-51
Editando Displays	3-51
Displays	3-51
Layouts	3-56
Resources	3-58
Editando Relatórios	3-60
TextEditor	3-62
Estudo Dirigido 3-3: edição da aplicação	3-63
4. Desenhando a Aplicação	4-1
Menu Draw	4-1
Ferramentas, Objetos e Componentes, Símbolos e Controles	4-2
Ferramentas de Seleção	4-3
Objetos Geométricos	4-4

Componentes de Tela	4-4
Entrada e Saída de Texto	4-4
Biblioteca de Símbolos	4-6
Controles Avançados	4-6
Alarmes e Tendências	4-13
Alarmes	4-13
Tendência (Legada)	4-14
Tendência	4-20
Estudo Dirigido 4-1: recursos Historian e TrendChart	4-22
Configuração Avançada	4-24
Grade de Dados	4-24
Barra de Ferramentas Horizontal	4-25
Configuração Dinâmica	4-26
CodeBehind	4-35
Símbolos Locais	4-36
Estudo Dirigido 4-2: desenho da aplicação	4-39
5. Implementando a Aplicação	5-1
Execução da Aplicação	5-1
Build	5-1
Test	5-3
Startup	5-4
Publish	5-5
UseCount	5-6
Dictionaries	5-9
Extensions	5-13
Informações da Aplicação	5-15
Project	5-15
Track	5-19
Notes	5-23
Ferramentas da Aplicação	5-24
Tstartup	5-24
Estudo Dirigido 5-1: atalho do TStartup e diferentes tipos de clientes	5-26
PropertyWatch	5-26
TraceWindow	5-27
ModuleInformation	5-29
Objetos em Tempo de Execução	5-31
Namespace Tag	5-33
Recursos Avançados	5-42
Linhas de Comando	5-42
Executando o BluePlant como um Serviço do Windows	5-45
TWebServer	5-45
Clientes Remotos	5-46
Instalando o Web Server no IIS	5-46
Estudo Dirigido 5-1: formatação de dados para comunicação	5-53
Estudo Dirigido 5-2: implementação da aplicação Mistura e Secagem	5-53
6. Aplicações, Avaliação e Encerramento	6-1
Cenários de Sistemas Típicos	6-1
Sistema Standalone	6-1
Sistema de Entradas e Saídas Distribuídas	6-1
Sistema Cliente e Servidor	6-2
Sistema de Servidor Redundante	6-3
Sistema de Controle	6-8
Sistema de Controle Distribuído e Distribuído Redundante	6-9
Sistema de Compartilhamento de Carga	6-10
Exercícios Dirigidos	6-10
Exercício 1: Display MENU	6-10
Exercício 2: Display MAIN	6-15

Exercício 3: Dispositivos	6-18
Exercício 4: Alarmes	6-20
Exercício 5: Histórico de Alarmes	6-23
Exercício 6: Tendência	6-25
Exercício 7: Banco de Dados	6-28
Exercício 8: Relatório	6-32
Exercício 9: Compilar, Publicar e Rastrear	6-34
Exercício 10: Programação VB.NET e C#	6-35
Situações de Aprendizagem Complementares	6-36
1. Texto Dinâmico em Botão	6-36
2. Comando e Monitoração via Pop-up	6-36
3. Utilização de uma biblioteca WPF externa	6-36
Estudo de Caso Livre	6-37
Avaliação do Treinamento	6-38
Parabéns...	6-38
Apêndice	A
Guia de Referência Rápida Comparativa de Comandos VB.NET X C#	A
Glossário	B

Figuras

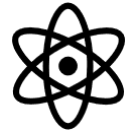
Figura 1-1. O BluePlant Altus	1-1
Figura 1-2. Tela de seleção do idioma do instalador	1-10
Figura 1-3. Tela do Contrato de Licença do BluePlant	1-10
Figura 1-4. Tela de informações referentes à instalação	1-11
Figura 1-5. Tela de seleção de tarefas adicionais	1-12
Figura 1-6. Tela de configurações da instalação	1-13
Figura 1-7. Tela de progresso da instalação de pré-requisito (Microsoft Visual C++)	1-14
Figura 1-8. Tela de Instalação do BluePlant	1-15
Figura 1-9. Tela de Encerramento de Instalação do BluePlant	1-16
Figura 2-1. Ícone do BluePlant	2-1
Figura 2-2. Tela de Gerenciamento de Projeto	2-1
Figura 2-3. Guia Servidor	2-2
Figura 2-4. Guia Licença	2-3
Figura 2-5. Criando um Novo Projeto	2-5
Figura 2-6. Interface do Usuário	2-6
Figura 2-7. Ícones Menu Editar	2-8
Figura 2-8. Menu Draw	2-9
Figura 2-9. Menu Run	2-10
Figura 2-10. Menu Info	2-11
Figura 2-11. Editando Tags do Projeto	2-12
Figura 2-12. Processo de Criação de Tags	2-13
Figura 2-13. Processo de Criação de Canal de Comunicação	2-14
Figura 2-14. Opções de Protocolos para Criação de um Canal	2-14
Figura 2-15. Inclusão e Configuração de Nós	2-15
Figura 2-16. Associação de Tags com Nós e Endereços de Protocolos	2-15
Figura 2-17. Inclusão de um Símbolo do Tipo Tacômetro	2-16
Figura 2-18. Associação de Tag a um Símbolo	2-17
Figura 2-19. Configuração da Tag Associado ao Símbolo	2-18
Figura 2-20. Inclusão de um Texto	2-19
Figura 2-21. Compilação do Projeto	2-20
Figura 2-22. Progresso da Compilação	2-20
Figura 2-23. Resultado da Compilação	2-20
Figura 2-24. Teste do Projeto	2-21
Figura 2-25. Execução do Teste do Projeto	2-22
Figura 2-26. Execução da Inicialização do Projeto	2-22
Figura 2-27. Status da Inicialização do Projeto	2-23
Figura 2-28. Publicação do Projeto	2-24
Figura 2-29. Caminho da Publicação do Projeto	2-24
Figura 3-1. Edição de objetos da Tag	3-2
Figura 3-2. Edição de Templates Tag	3-7
Figura 3-3. Edição de Nível (Assets)	3-8
Figura 3-4. Configurações da Tabela do Historiador	3-9
Figura 3-5. Edição de Historiador	3-9
Figura 3-6. Edição de Perfis de Usuários	3-12
Figura 3-7. Edição de Permissões de Segurança	3-13
Figura 3-8. Edição de Políticas de Segurança	3-14
Figura 3-9. <i>RuntimeUsers</i>	3-15
Figura 3-10. Edição de Canais de Comunicação	3-16
Figura 3-11. Edição de um Novo Canal de Comunicação	3-18
Figura 3-12. Edição do Nó para o Canal de Comunicação	3-19
Figura 3-13. Parametrização da Estação Primária	3-20

Figura 3-14. Edição de Pontos	3-21
Figura 3-15. Exemplo de Parametrização do Endereço	3-22
Figura 3-16. Exemplo de Uso do Modificador BIT	3-23
Figura 3-17. Conversão de Escala	3-24
Figura 3-18. Tipo de Acesso	3-25
Figura 3-19. AccessType	3-25
Figura 3-20. Grupos de Alarmes	3-27
Figura 3-21. Audit Trail	3-29
Figura 3-22. Áreas de Alarmes	3-30
Figura 3-23. Itens de Alarmes	3-30
Figura 3-24. Criação de Novo Database	3-33
Figura 3-25. Edição de Tabelas de Dados	3-35
Figura 3-26. Edição de Queries	3-36
Figura 3-27. Edição Arquivos de Bancos de Dados	3-37
Figura 3-28. Edição de Tarefas de Script	3-39
Figura 3-29. Classes de Script	3-41
Figura 3-30. CodeEditor	3-42
Figura 3-31. Tarefas de Scripts	3-47
Figura 3-32. CodeEditor das Tarefas de Scripts	3-48
Figura 3-33. Anexando depurador .NET	3-49
Figura 3-34. Editor de Código os Botões para Anexar e Desanexar depurador	3-49
Figura 3-35. Watch para variáveis dotNET	3-50
Figura 3-36. Edição de Displays	3-52
Figura 3-37. Permissões no <i>Runtime</i>	3-53
Figura 3-38. Configurando Múltiplos Monitores	3-54
Figura 3-39. Configurações dos Clientes Rich	3-55
Figura 3-40. Edição de Layouts	3-57
Figura 3-41. Edição de Recursos	3-58
Figura 3-42. Configuração de Relatórios	3-60
Figura 3-43. Edição de Relatório	3-61
Figura 3-44. Editor de Texto	3-62
Figura 3-45. Edição e Salvamento no Editor de Texto	3-62
Figura 3-46. Configuração de um Relatório	3-62
Figura 4-1. Menu Draw e Barra de Ferramentas Vertical	4-1
Figura 4-2. Menu Draw e Barra de Ferramenta Horizontal	4-2
Figura 4-3. Barra de Ferramentas Vertical	4-3
Figura 4-4. Objeto TextOutput	4-5
Figura 4-5. Objeto TextIO	4-5
Figura 4-6. Configuração de Page Selector	4-7
Figura 4-7. Configuração do XPS Viewer	4-8
Figura 4-8. Configuração de Painel Circular	4-9
Figura 4-9. Configuração de Gráfico de Pizza	4-10
Figura 4-10. Configuração de Gráfico de Barras	4-11
Figura 4-11. Janela de Alarmes	4-13
Figura 4-12. Configuração da Janela de Alarme	4-13
Figura 4-13. Configuração da Janela de Tendência	4-15
Figura 4-14. Cursor Settings	4-16
Figura 4-15. Eixo Y	4-16
Figura 4-16. Eixo X	4-17
Figura 4-17. Configuração da Caneta	4-17
Figura 4-18. Configuração de Estilo da Linha	4-18
Figura 4-19. SPC Settings	4-19
Figura 4-20. Opções Avançadas	4-19
Figura 4-21. Configuração da Janela de Tendência	4-20
Figura 4-22. Configuração da Janela do DataGrid	4-24
Figura 4-23. Configuração Dinâmica	4-27
Figura 4-24. CodeBehind	4-35

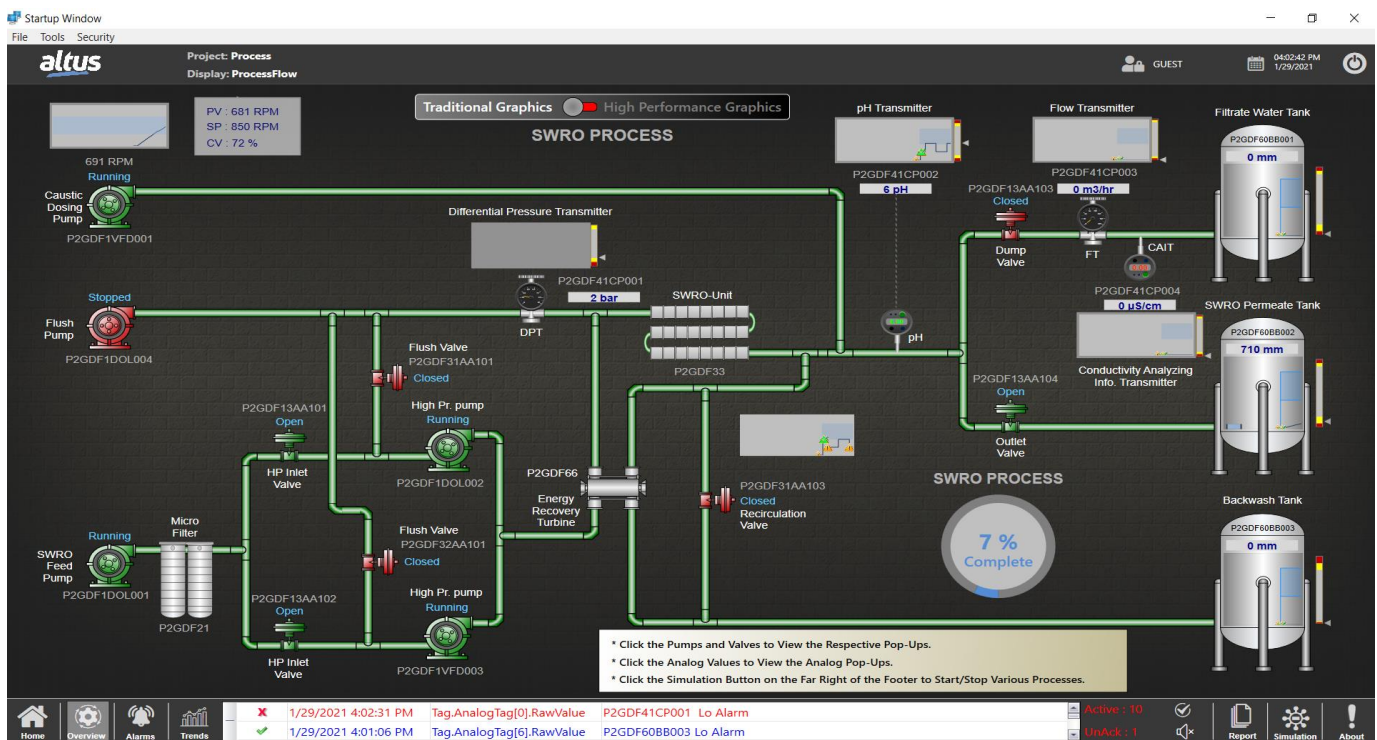
Figura 4-25. Menu Contexto do Símbolo	4-36
Figura 4-26. Criando um Símbolo	4-37
Figura 4-27. Configuração Dinâmica	4-37
Figura 4-28. Abrir Symbol Library	4-38
Figura 5-1. Guia Messages	5-1
Figura 5-2. Guia History	5-2
Figura 5-3. Menu Test	5-3
Figura 5-4. Menu Startup	5-4
Figura 5-5. Menu Publish	5-5
Figura 5-6. Menu UseCount	5-7
Figura 5-7. Menu CrossReference	5-8
Figura 5-8. Menu UnusedObjects	5-9
Figura 5-9. Menu Localization	5-10
Figura 5-10. Menu EnumerationSets	5-11
Figura 5-11. Configuração dos parâmetros	5-12
Figura 5-12. Menu Categories	5-12
Figura 5-13. Menu Import	5-13
Figura 5-14. Menu Export	5-14
Figura 5-15. Guia Version	5-15
Figura 5-16. Guia Settings	5-16
Figura 5-17. Ajuste do Tamanho Padrão para novos Layouts	5-17
Figura 5-18. Ajuste do Tamanho Padrão para novas Páginas	5-17
Figura 5-19. Ajuste de módulos dentro do TServer	5-17
Figura 5-20. Ajuste do protocolo OPC Server	5-17
Figura 5-21. Ajuste do Security	5-17
Figura 5-22. Guia Redundancy	5-18
Figura 5-23. Exemplo de Configuração de Redundância de Servidores	5-19
Figura 5-24. Guia Tables do Menu Info, Opção Track	5-20
Figura 5-25. Guia Changes do Menu Info, Opção Track	5-21
Figura 5-26. Guia Releases do Menu Info, Opção Track	5-22
Figura 5-27. Guia Notes do Menu Info Opção Notes	5-23
Figura 5-28. Janela de Startup	5-25
Figura 5-29. PropertyWatch	5-27
Figura 5-30. TraceWindow	5-28
Figura 5-31. Configurações	5-29
Figura 5-32. Informações do Módulo	5-30
Figura 5-33. Namespaces em Diálogos	5-32
Figura 5-34. Namespaces em Scripts	5-33
Figura 5-35. Desativar o Modo Protegido	5-46
Figura 5-36. Habilitar Aplicações do Navegador XAML	5-46
Figura 5-37. Procedimento de Instalação para o IIS 7.x	5-47
Figura 5-38. ServiceModelReg.exe	5-47
Figura 5-39. Chaveamento de Funcionalidades do Windows	5-48
Figura 5-40. Funcionalidades do Windows	5-48
Figura 5-41. Ativação HTTP	5-49
Figura 5-42. Mapeamento da Extensão	5-49
Figura 5-43. Utilitário InstallTWebServer	5-49
Figura 5-44. Informação para serviços	5-50
Figura 5-45. ProjectServer	5-50
Figura 5-46. Serviço iDataPanel	5-51
Figura 5-47. Diretório Virtual	5-52
Figura 5-48. Acrescentar Diretório Virtual	5-52
Figura 5-49. Autenticação	5-52
Figura 6-1. Sistema Stand Alone	6-1
Figura 6-2. Sistema de Entradas e Saídas Distribuídas	6-2
Figura 6-3. Sistema Cliente e Servidor	6-3
Figura 6-4. Sistema Servidor Redundante Terceira Máquina	6-4

Figura 6-5. Sistema Servidor Redundante Banco de Dados Local	6-4
Figura 6-6. Sistema Servidor Redundante com Redundância nos Dispositivos	6-5
Figura 6-7. Mecanismo de funcionamento do Servidor Redundante	6-6
Figura 6-8. Sistema de Controle	6-8
Figura 6-9. Sistema de Controle Distribuído	6-9
Figura 6-10. Sistema de Controle Distribuído Redundante	6-9

Introdução



Este Tutorial fornece um suporte didático-pedagógico ao treinamento presencial da Série Altus BluePlant, possibilitando ao usuário um primeiro contato orientado com a solução definitiva para supervisão e sistemas de aquisição de dados e controle - explorando suas características básicas, assim como as funcionalidades de aquisição de dados, interface gráfica, relatórios, acesso a bancos de dados externos e scripts, de maneira que o aluno possa especificar e configurar interfaces de operação para aplicações que demandem recursos gráficos diferenciados. Esta seção contém as divisões do tutorial e as simbologias e convenções utilizadas.



Um pouco de teoria... O que são sistemas SCADA?



Os sistemas SCADA, podem ser definidos como uma tecnologia que permite que seja monitorada e rastreada a informação de um processo produtivo ou instalação física. Tais informações são coletadas através de equipamentos de aquisição de dados e, em seguida, manipuladas, analisadas, armazenadas e, posteriormente, apresentadas ao usuário. Estes sistemas são tipicamente compostos por estações terminais mestres (MTU), unidades remotas como: CLP, RTU ou IED, dispositivos de campo como: sensores e atuadores e toda a tecnologia de comunicação envolvida para interconexão desses componentes. Dentro das funcionalidades dos sistemas SCADA, encontram-se:

- Configuração da interface de comunicação, das variáveis de entrada e de saída e dos operadores que terão acesso ao sistema;
- Construção de sinóticos, que representam graficamente os processos industriais;
- Geração de gráficos de tendências, que acompanham a evolução das variáveis do sistema;
- Geração de alarmes, que avisam ao usuário do sistema quando uma variável ou condição do processo de produção está fora dos valores previstos;
- Geração de históricos, que registram os eventos relevantes;
- Produção de relatórios a partir dos dados da planta, dos alarmes ocorridos em um intervalo de tempo e dos acessos ao sistema por parte dos operadores;
- Geração de scripts ou programas, que desenvolvem alguma lógica para o controle do processo;
- Criação de receitas, que configuram os equipamentos da planta para que essa passe a produzir determinado(s) produto(s); e
- Geração de informação para níveis gerenciais.

Estas funcionalidades tornam a maioria dos sistemas SCADA, hoje instalados, uma parte essencial da estrutura de gestão das informações corporativas e do chão de fábrica, os quais são vistos pela gerência, não simplesmente como ferramentas operacionais, mas como recurso importante de informação. Fonte: <https://repositorio.ufsc.br/xmlui/handle/123456789/95357>.

O Curso MP5



O curso MP5 contempla a Série Altus BluePlant - a solução definitiva para supervisão e sistemas de aquisição de dados e controle. O BluePlant atende requisitos de alto desempenho, amplos recursos de conectividade, interface gráfica de usuário extremamente rica e poderosa além de mecanismos avançados de aquisição de dados em tempo real. A seleção de drivers incorporados no BluePlant, a capacidade de engenharia distribuída, a redundância e o suporte OPC garantem uma experiência nova e única. Desenvolvido no Microsoft Windows Presentation Foundation (WPF), a tecnologia do BluePlant permite obter melhores resultados com as placas gráficas atuais, resultando em aplicações de desempenho excepcional.

O BluePlant inclui também as funcionalidades padrão encontradas nesse tipo de produtos, tais como interação com servidores de banco de dados (SQL, PI, Oracle, Sybase, Informix e outros), barramentos de rede personalizados, módulo de notificação de eventos e servidor de alarme fácil de utilizar, componente de registro e relatório, servidor de histórico avançado, gerenciamento de lógica de negócios, bem como suporte para clientes locais e remotos, também sendo executado em computadores, web, tablets e smartphones.

Os objetivos desse curso são:

- Fazer conhecer as funcionalidades básicas da Série Altus BluePlant no que se refere a aplicações típicas, modelos, recursos, comunicação e instalação.
- Apresentar e utilizar a ferramenta BluePlant para o desenvolvimento de aplicações de Interface Homem Máquina (IHM), explorando seus principais recursos, tais como funções de aquisição de dados, interface gráfica, relatórios, acesso a bancos de dados externos e scripts.

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 duração de 24 horas e demanda como pré-requisito a participação prévia do aluno no Curso MP1 — Introdução à Automação e certa experiência em aplicativos Windows. Além disso, para explorar algumas funções mais avançadas de scripts é necessário também o conhecimento do Microsoft .NET Framework e C# ou VB.NET.

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 sistema e enfatizando a configuração, edição e programação do mesmo. Ele é dividido em capítulos numa estrutura padronizada conforme esquema mostrado a seguir.

- ▷ 1. Características Técnicas
- ▷ 2. Características Operacionais
- ▷ 3. Editando a Aplicação
- ▷ 4. Desenhando a Aplicação
- ▷ 5. Implementando a Aplicação
- ▷ 6. 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: Características Técnicas do BluePlant

Este capítulo descreve as características gerais e avançadas do BluePlant enfatizando suas funcionalidades, recursos, ferramentas e diferenciais do produto. Na parte final é detalhado o procedimento de instalação do software. Tópicos abordados:

- ▲ 1. Características Técnicas
 - ▷ Características Gerais do BluePlant
 - ▷ Características Avançadas do BluePlant
 - ▷ Instalação do BluePlant

Capítulo 2: Características Operacionais do BluePlant

Este capítulo aborda os principais itens envolvidos em um projeto do BluePlant iniciando com os passos básicos, passando pelos itens do menu principal e culminando com as principais ações associadas a uma aplicação, tais como, criação de Tags, inclusão de objetos e execução do projeto. Tópicos abordados:

- ▲ 2. Características Operacionais
 - ▷ Primeiros Passos com o BluePlant
 - ▷ Itens do Menu Principal
 - ▷ Implementação de um Projeto no BluePlant

Capítulo 3: Editando a Aplicação

Este capítulo explora o ambiente de edição do BluePlant, o qual fornece acesso a todas as funcionalidades necessárias para configurar o projeto, o que inclui Tags, segurança, dispositivos, alarmes, bancos de dados, scripts, displays e relatórios. Tópicos abordados:

- ▲ 3. Editando a Aplicação
 - Menu Edit
 - ▷ Editando Tags
 - ▷ Editando Configurações de Segurança
 - ▷ Editando Dispositivos
 - ▷ Editando Alarmes
 - ▷ Editando Banco de Dados
 - ▷ Editando Scripts
 - ▷ Editando Displays
 - ▷ Editando Relatórios

Capítulo 4: Desenhando a Aplicação

Este capítulo detalha o ambiente de telas, o qual fornece acesso a todas as ferramentas de diagramação para compilar as telas da aplicação. Este ambiente inclui as seguintes opções: tela, códigos e símbolos, por meio do menu Draw. Tópicos abordados:

- ▲ 4. Desenhando a Aplicação
 - Menu Draw
 - ▷ Ferramentas, Objetos e Componentes, Símbolos e Controles
 - ▷ Alarmes e Tendências
 - ▷ Configuração Avançada

Capítulo 5: Implementando a Aplicação

Este capítulo detalha os passos necessários para executar uma aplicação editada e desenhada (tópicos dos capítulos anteriores). Ele inclui ainda considerações sobre informações da aplicação. Além disso, são explorados recursos da aplicação como ferramentas e objetos em tempo de execução e configurações avançadas do BluePlant. Tópicos abordados:

- ▲ 5. Implementando a Aplicação
 - ▷ Execução da Aplicação
 - ▷ Informações da Aplicação
 - ▷ Ferramentas da Aplicação
 - ▷ Objetos em Tempo de Execução
 - ▷ Recursos Avançados

Capítulo 6: Aplicações, Avaliação e Encerramento

Este capítulo descreve os cenários típicos de utilização do BluePlant para atender demandas que contemplam as áreas de produção, utilidades e manufatura. Além disso, são propostos exercícios dirigidos e situações de aprendizagem complementares. O tutorial finaliza com a proposição de um estudo de caso livre para consolidação das competências desenvolvidas no treinamento. Tópicos abordados:

- ▲ 6. Aplicações, Avaliação e Encerramento
 - ▷ Cenários de Sistemas Típicos
 - ▷ Exercícios Dirigidos
 - ▷ Situações de Aprendizagem Complementares
 - ▷ Estudo de Caso Livre
 - ▷ Avaliação do Treinamento

Referências



1. Site Altus: www.altus.com.br.
2. INTERNET (Sites diversos).
3. Documentos relacionados à Série Altus BluePlant (conforme indicado na sequência).

Documentos relacionados

Para obter informações adicionais sobre a Série BluePlant podem ser consultados outros documentos (manuais e características técnicas) além deste. Estes documentos encontram-se disponíveis em sua última revisão em www.altus.com.br.

Cada produto possui um documento denominado Característica Técnica (CT), onde se encontram as características do produto em questão. Adicionalmente o produto pode possuir Manuais de Utilização (os códigos dos manuais são citados na CT).

Aconselha-se os seguintes documentos como fonte de informação adicional:

- CT124000 - Características Técnicas BluePlant;
- NAP151 - Utilização do Tunneller OPC;
- NAP154 - Utilização de Base de Dados com BluePlant;
- NAP155 - Utilização de Base de Dados MySQL com BluePlant;
- NAP156 - Utilização de Base de Dados Microsoft SQL Server com BluePlant;
- NAP157 - Redundância de Servidores SCADA com BluePlant.

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...



SUPORTE & DOWNLOADS

Home > Suporte & Downloads

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

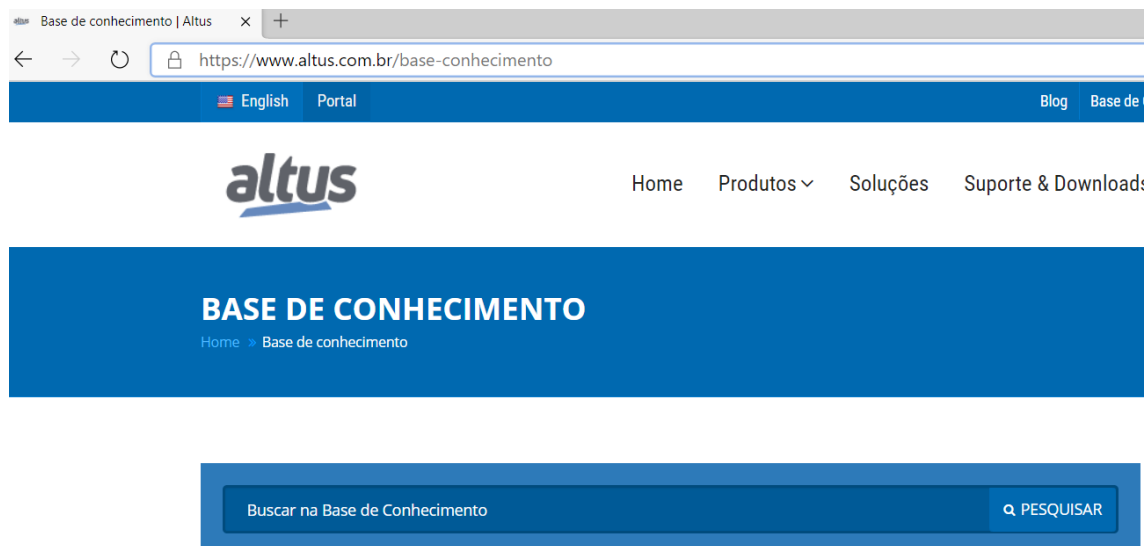
Aqui você encontra tutoriais, dicas e outros materiais de apoio criados por nossos especialistas para qualificar a sua experiência com os produtos e soluções Altus!

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 COMPANYY, 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	Outubro/12	Emissão do Documento
B	Fevereiro/21	Revisão Geral do Documento

Training Box Nexto Xpress



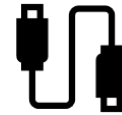
Na execução desse curso é utilizada a Maleta de Treinamento Nexto Xpress conforme mostrado na figura a seguir, a qual é um sistema didático completo e inovador, apresentando recursos que permitem ao usuário simular elementos presentes nos sistemas de automação industrial, predial e de máquinas, sendo excelente ferramenta para empresas deste setor e para instituições de ensino aplicar os novos conceitos da Indústria 4.0 e Internet das Coisas (IoT). Os produtos Training Box Nexto Xpress são integrantes da Série Training Box desenvolvida especialmente para prover soluções completas e compactas para uso em laboratórios de ensino e pesquisa. Kit didático compacto e resistente, proporciona facilidade no transporte e armazenamento.

Ela é composta por um Controlador Programável (CP) XP340 da Série Nexto como unidade central, o qual disponibiliza uma porta Ethernet com diversos protocolos, um canal serial RS-485 com MODBUS RTU, uma interface CAN, uma porta USB, 16 entradas digitais 24 Vdc, 16 saídas digitais a transistor, 5 entradas analógicas tensão/corrente, 2 entradas analógicas RTD, 4 saídas analógicas tensão/corrente e suporte ao Web Server, que permite criar IHMs e/ou telas de supervisão para serem acessadas no computador ou em dispositivos móveis. Finalmente, possui circuito dedicado que permite realizar simulações de controle e distúrbio em laços PID.



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

1. Características Técnicas



Características Gerais do BluePlant



O que é o BluePlant?

O BluePlant da Altus é a solução definitiva para supervisão e sistemas de aquisição de dados e controle.

O BluePlant atende requisitos de alto desempenho, amplos recursos de conectividade, interface gráfica de usuário extremamente rica e poderosa além de mecanismos avançados de aquisição de dados em tempo real. A seleção de drivers incorporados no BluePlant, a capacidade de engenharia distribuída, a redundância e o suporte OPC garantem uma experiência nova e única. Desenvolvido no Microsoft Windows Presentation Foundation (WPF), a tecnologia do BluePlant permite obter melhores resultados com as placas gráficas atuais, resultando em aplicações de desempenho excepcional.

O BluePlant inclui também as funcionalidades padrão encontradas nesse tipo de produtos, tais como interação com servidores de banco de dados (SQL, PI, Oracle, Sybase, Informix e outros), barramentos de rede personalizados, módulo de notificação de eventos e servidor de alarme fácil de utilizar, componente de registro e relatório, servidor de histórico avançado, gerenciamento de lógica de negócios, bem como suporte para clientes locais e remotos, também sendo executado em computadores, web, tablets e smartphones.



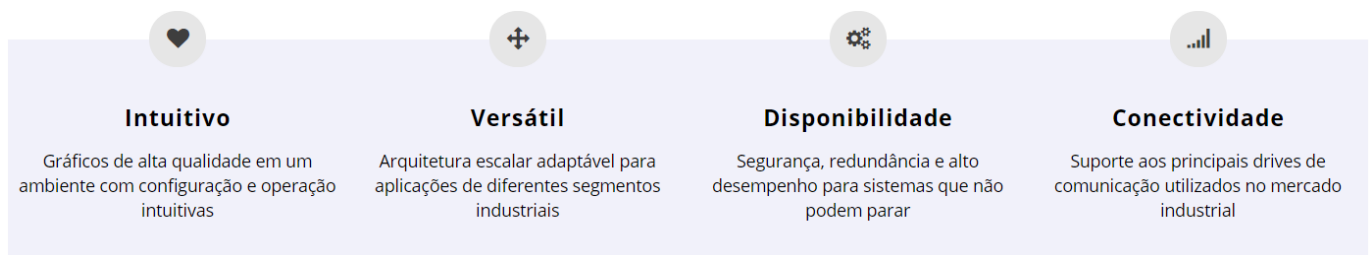
Figura 1-1. O BluePlant Altus

Porque o BluePlant?

A reputação da Altus está associada à excelência no fornecimento de sistemas de automação e produtos para controle de processos, como Controladores Programáveis (CPs) e Unidades Terminais Remotas (UTRs), oferecendo desempenho superior, tecnologia no estado da arte e características como redundância, alteração online, troca a quente, alta conectividade e outras funcionalidades avançadas. Esta vasta experiência em sistemas de automação industrial serviu de base para o desenvolvimento deste software SCADA/IHM. A experiência e o portfólio diversificado de produtos garantem à Altus uma posição-chave no fornecimento de soluções completas em automação.

Solução para supervisão, controle e aquisição de dados, o BluePlant alia design, desempenho e alta conectividade em um sistema rico em recursos e funcionalidades. Com visual atraente e intuitivo, o supervisório possui uma arquitetura cliente-servidor com inúmeras opções de quantidades de Tags de comunicação, além de simulador e ferramentas de depuração.

Por ter elevada capacidade de adaptação, o sistema pode ser aplicado em qualquer tipo de projeto de diferentes segmentos da indústria, como Predial, Agroindústria, Saneamento, Transportes, Siderurgia, Alimentício, Mineração, entre outros tantos. O software supervisório também permite que usuários possam se conectar remotamente para desenvolver sua programação ou trabalhar de forma colaborativa com vários desenvolvedores simultaneamente. Esse recurso agiliza o desenvolvimento da aplicação e dá mais confiabilidade e segurança ao sistema.



Quais São Seus Diferenciais?

Economia e agilidade para o comissionamento de projetos

O BluePlant disponibiliza, de forma integrada, painéis de controle dedicados para a análise e verificação de problemas e otimização de desempenho. Através dos painéis de usuário, pode-se verificar intervalos de tempo e consumo de processamento de drivers, módulos e outros componentes que estão sendo utilizados. Estas ferramentas permitem maior precisão nos diagnósticos e redução do investimento em horas de engenharia para o comissionamento e depuração do sistema de aquisição e supervisão em projetos de automação e controle de processos.

Velocidade de desenvolvimento e integração de telas

O sistema gráfico do BluePlant utiliza a tecnologia Microsoft WPF para criação de interfaces de usuário com mapeamento de valores e Tags em tempo real, um diferencial em relação aos outros supervisórios do mercado. O sistema conta com uma biblioteca de símbolos pré-definidos que aceleram a criação de telas de sinóticos, permitindo a criação de telas gráficas, cenários e representações de plantas. Aliado ao seu ambiente moderno, diversos recursos no design de telas são contemplados, como capacidade para objetos em 3D e multi-touch, que reconhece um ou mais pontos de contato com a superfície, zoom inteligente com o scroll do mouse, gráficos de chart, pizza, entre outros.

O supervísório também permite a integração de bibliotecas de figuras em diversos formatos na aplicação com fundo de imagem transparente, agilizando o desenvolvimento de telas.

Integração com plataformas externas

O BluePlant possibilita a criação de arquivos de historiador em bancos de dados externos, como Microsoft SQL Server ou Oracle, para uso em diversas aplicações da indústria. O supervísório permite que dados sejam salvos com base na alteração de valores de dados ou disparadores para registro de dados.

O software também possui uma opção exclusiva de faixa de tempo que impede o registro de dados com uma estampa de tempo menor do que um valor pré-definido, permitindo a criação de bancos de dados mais compactos e otimizados. A estampa de tempo permite usar o valor fornecido diretamente pelo dispositivo de controle ou de campo, garantindo maior precisão no evento. A organização das amostras permite incluir ou remover Tags do registro sem perder a compatibilidade com os últimos dados. Além disso, um objeto completo do tipo gráfico de tendências também é fornecido para a visualização de dados online e históricos.

Arquitetura modular e alta disponibilidade para processos críticos

Sua arquitetura modular pode ser configurada em diversos sistemas, bastando ao usuário escolher a opção que melhor se encaixe com a sua aplicação. A topologia cliente-servidor do supervísório permite a configuração de cada um dos módulos em diferentes computadores e nas quantidades exatas para sua aplicação. Além disso, sua plataforma moderna e nativa em 64 bits oferece também suporte a sistemas operacionais de 32 bits, suporte à redundância e tecnologia OPC.

Seu sistema de servidores redundantes permite que dois computadores distintos executem simultaneamente uma aplicação em uma topologia *hot-standby*. Toda a configuração de redundância é automática no próprio aplicativo, sendo necessário determinar os endereços de IP das estações primária e secundária. Nesta topologia, um computador em estado de espera (*standby*) é mantido em conjunto com o computador rodando a aplicação ativa. Se um erro de hardware ocorrer, o servidor em espera assume o controle automaticamente sem paradas no sistema ou perda de dados.

Quais são as principais características do produto?

O BluePlant executa nativamente em máquinas de 64 bits com .NET Framework 4. Há diferentes modelos de produtos para permitir a escolha da melhor solução de acordo com as necessidades. O BluePlant pode atender desde soluções de grande porte até aplicações embarcadas. Os modelos do BluePlant são compatíveis com o legado de computadores de 32 bits. As telas do cliente podem ser executadas em navegadores Web nos computadores com Windows e nos dispositivos móveis baseados em Windows.

	Características
BluePlant Enterprise	Projetado para gerenciamento de plantas de processo, inteligência de negócios (Business Intelligence - BI), painéis em tempo real, SCADA, IHM avançado, controle de processo e otimização. Permite clientes e aquisição de dados distribuídos. O tamanho da aplicação parte de 150 pontos de comunicação
BluePlant Lite	Projetado para painéis, computadores industriais, dispositivos embarcados e sistemas autônomos. Aplicado principalmente em interface de máquinas e pequenos projetos centralizados. O tamanho da aplicação varia de 150 até 1500 pontos de comunicação
BluePlant Express	Projetado somente para avaliação, e não para instalação em campo, limitada em 75 pontos de comunicação e uma hora de execução do <i>Runtime</i>
BluePlant Student	Projetado para escolas e universidades. O tamanho da aplicação varia de 150 até 1500 pontos de comunicação e uma hora de execução do <i>Runtime</i>

Tabela 1-1. Modelos de BluePlant

Para atender as necessidades de integradores de sistemas foram criadas soluções que permitem o desenvolvimento de projetos como no BluePlant Lite ou BluePlant Enterprise. Essas soluções para integradores de sistemas executam o *Runtime* por apenas uma hora para que possam ser realizados os testes do projeto. Após esse período, é necessário reiniciar o *Runtime*.

Todas as soluções para integradores de sistemas comunicam plenamente com os controladores programáveis através dos principais drivers de comunicação incorporados nos produtos.

O BluePlant é dividido em família/modelos de acordo com as funcionalidades e requisitos de cada aplicação. Desta forma é possível utilizar o modelo de BluePlant adequado ao tamanho da aplicação para otimizar o desempenho do sistema. Deve ser levado em consideração a quantidade total de Tags disponíveis, sendo esta quantidade 10 vezes a quantidade de pontos de comunicação. Os pontos de comunicação estão incluídos na quantidade total.



Pesquisa na Documentação: modelos do BluePlant

Consulte, na documentação do produto, a quantidade de pontos de comunicação e os produtos relacionados por modelo de BluePlant, bem como as soluções disponíveis para os integradores de sistemas.

Fonte: Manual de Utilização Série BluePlant – MU224000!

Características Avançadas do BluePlant



Software com Segurança Intrínseca

A característica de Segurança Intrínseca confere segurança e confiabilidade ao produto, uma das bases fundamentais para o desenvolvimento da plataforma BluePlant é que não há nenhum uso de código C ou C++, eliminando totalmente o risco de problemas com ponteiro e/ou exceções de memória. Cada tarefa de execução e processo, seja interna ou criada para ser executada no âmbito do BluePlant, é executada em seu próprio espaço alocado e "protegido" com controle de exceção interno, isolamento de memória, controle multitarefa e sincronização em tempo real. A metodologia de desenvolvimento de software empregada com VBA, VBScript, matemática e lógica proprietária, onde os potenciais problemas podem ser detectados em *Runtime*, foram eliminados e substituídos por linguagens compiladas do .NET, com validação completa dos scripts antes da execução pelo *Runtime* com proteção interna que confere desempenho superior, maior estabilidade operacional e segurança.

Mecanismo Gráfico Superior

Os gráficos no BluePlant são do tipo Windows Presentation Foundation (WPF) com sistema interno utilizando XAML. Isso permite a integração com mapas geoespaciais e modelos 3D. Os modelos 3D podem ser apresentados diretamente, bem como vinculados a dados dinâmicos com respostas associadas e comportamentos baseados em eventos e valores em tempo real. Um poderoso editor gráfico WPF está incluído no BluePlant. Os clientes Web dependem de XBAP (aplicativos baseados em navegador e Silverlight) de forma que não há nenhuma exigência para a instalação de qualquer componente externo ActiveX.

Recursos de Manutenção, Teste e Diagnóstico Avançado

O sistema possibilita a troca de versões do projeto, permitindo que as aplicações em modo teste sejam executadas lado a lado, no mesmo servidor, com as aplicações no modo produção para validação e garantia da qualidade, incluindo a análise da utilização da UCP e das estatísticas de comunicação dos módulos do *Runtime* e redes fornecidas. As opções de hot-standby interna para redundância, locais alternativos de funcionamento e recuperação de desastres também estão incorporadas.

Servidores e Extensões .NET Incorporados

Além dos módulos incorporados para banco de dados em tempo real, SQL externo e conexões ERP, servidor de alarmes e eventos, servidor de históricos e emissão de relatórios, o BluePlant permite acesso completo ao Microsoft .NET Framework, permitindo customização e extensibilidade avançadas, sem o acréscimo de qualquer tipo de aplicação de terceiros ou ferramenta externa.

Além das características indicadas anteriormente, o BluePlant foi projetado a partir de um "campo verde" e foi criado totalmente sem empregar qualquer código herdado. É um aplicativo de código 100% gerenciado que permite utilizar e aproveitar o potencial pleno do Microsoft .NET Framework tanto atualmente, como futuramente. O BluePlant possui uma interface de configuração criada inteiramente a partir do Microsoft Windows Presentation Foundation Graphics (WPF) e suporta integralmente software como serviços (SaaS) implantação combinada com instalações locais, permitindo acessar e colaborar no desenvolvimento de projetos em todo o mundo com apenas um navegador de internet.

Banco de Dados em Tempo Real (Tags)

O BluePlant suporta os seguintes tipos de pontos de dados em tempo real: Digital (Booleano), Analógico (Inteiro, Double e Decimal), Mensagem de Texto, Contador, Temporizador e variáveis de data e hora. Além disso, podem ser definidos tipos próprios com vários níveis de encadeamento, Tags de referência e matrizes tridimensionais.

O banco de dados em tempo real garante, sem a necessidade de qualquer programação adicional, a sincronização de dados entre vários processos no servidor e múltiplas estações cliente. Um vasto conjunto de propriedades internas, tais como qualidade de dados, estampa de tempo, bloqueio de estado e valor bloqueado, permitem e simplificam a criação de aplicações.

Linguagens .NET e Scripts

BluePlant é um sistema SCADA que suporta as linguagens do Microsoft .NET em completa integração com o Microsoft .NET Framework. Os scripts e lógicas do projeto podem ser escritos em C# ou VB.NET, e um conversor de linguagem integrado permite alternar dinamicamente o código criado entre as linguagens.

No ambiente do BluePlant, é possível compilar, realizar a referência cruzada dos objetos e acessar diretamente (usando o Intellisense) as classes .NET e objetos do projeto, incluindo alarmes, relatórios e nós de comunicação.

As linguagens .NET fornecem um ambiente mais poderoso quando comparado com o VBA ou VBScript, que são linguagens interpretadas e não compiladas. Essas tecnologias deixam muitos erros de programação que são encontrados somente quando projeto em VBA ou VBScript é executado em tempo real, resultando muitas vezes em consequências e resultados indesejados. O ambiente gerenciado do Microsoft .NET Framework dá suporte para localizar e se recuperar das exceções, proporcionando assim um ambiente altamente confiável para o sistema do *Runtime* e as aplicações.

Alarmes e Segurança

Podem ser definidos vários níveis de alarme para cada ponto ou Tag e toda uma gama de comportamentos, tais como registro, reconhecimento, exibição etc., os quais são pré-definidos para facilitar a configuração. O sistema de segurança pode definir níveis de acesso para cada objeto de tela. Condições de alarme e de segurança são automaticamente replicadas nas aplicações redundantes.

Tendência e Historiadores

O BluePlant permite criar arquivos de historiador em bancos de dados externos, tais como Microsoft SQL Server ou Oracle, ou mesmo usar o banco de dados SQL integrado. Ele possibilita salvar os dados com base na alteração de dados ou disparadores de grupo e tem uma opção exclusiva de faixa de tempo que impede o registro de dados com uma estampa de tempo menor do que um valor pré-definido, permitindo a criação de bancos de dados mais compactos. O acesso a um servidor de PI da OSIsoft também é possível.

O recurso de estampa de tempo pode utilizar um valor fornecido diretamente a partir da E/S remota, ao invés do computador, garantindo maior precisão no evento. A organização das amostras permite incluir ou remover Tags do registro sem perder a compatibilidade com os últimos dados. Um objeto completo do tipo gráfico de tendência também é fornecido para a visualização de dados online e históricos.

Dispositivos e Redes

O BluePlant é fornecido com um driver OPC DA para coleta de informações de dispositivos remotos. Além do OPC, o BluePlant também oferece suporte a drivers de comunicação

personalizados para acessar diretamente CPs, sistemas remotos de E/S, barramentos de campo padrões, laços únicos e múltiplos, scanners, leitores de código de barras, dispositivos RFID e monitores digitais.

A ferramenta de configuração de dispositivo pode importar bancos de dados de servidores OPC, arquivos de texto ou CSV. Se o dispositivo for compatível, automaticamente implementa sistemas multitarefa em redes TCP/IP ou cenários multi-seriais. A sintaxe de endereçamento segue a convenção de nomenclatura do dispositivo remoto, tornando a configuração e a manutenção muito mais fáceis. Um conjunto completo de ferramentas de desempenho e diagnóstico está incluído também.

Conjunto de Dados

O módulo de conjunto de dados incluído no BluePlant fornece uma interface de fácil operação para a troca de dados em tempo real com bancos de dados externos, XML, CSV ou arquivos de texto, bem como a possibilidade de acessar tabelas e consultas SQL.

Para os mais populares bancos de dados e fontes de dados (Microsoft SQL Server, Oracle, Arquivos CSV, Microsoft Access, PI, Firebird, Informix e Excel), o BluePlant fornece configurações pré-definidas que reduzem o gerenciamento de configurações a um clique do mouse. Qualquer banco de dados que ofereça suporte a ODBC, ADO.NET ou OLE-DB também pode ser acessado. Um banco de dados DB SQL integrado é fornecido como opção de banco de dados local para a aplicação.

Os dados coletados via conjunto de dados podem ser mapeados dinamicamente em tempo real para pontos/tags e podem ser usados em scripts ou relatórios ou ainda apresentados em telas usando um poderoso objeto visual de grade de dados.

Relatórios

BluePlant oferece suporte a serviços Web, XML e outras interfaces de troca de dados com o objetivo de fornecer dados para as ferramentas de relatórios externas. Em contraste com outros pacotes, onde os relatórios são necessariamente criados em outra ferramenta, o BluePlant tem seu próprio editor de relatório interno.

O editor de relatório permite a inclusão de texto dinâmico, símbolo gráfico dinâmico e gráficos, e pesquisas em banco de dados, em um editor completo e de fácil utilização. Os relatórios podem ser salvos nos formatos texto, HTML, PDF ou XPS e facilmente apresentados em clientes remotos e telas Web.

Telas do Cliente

O editor gráfico integrado no BluePlant utiliza a tecnologia WPF da Microsoft para permitir a criação de interfaces completas de usuário com mapeamento em tempo real de valores e Tags do processo: um poderoso e completo conjunto de animações dinâmicas também está incluído.

As telas são salvas internamente usando XAML, o qual fornece independência de resolução, isolamento do código e fácil extensibilidade. Uma biblioteca de símbolos, onde os símbolos também podem manter um vínculo dinâmico com a biblioteca, acelera criação de processos sinóticos. Todas as tecnologias do cliente oferecem suporte de servidor redundante.

Três tecnologias são utilizadas em clientes remotos:

- **BluePlant Visualizer Clients:** é executado como uma aplicação desktop e permite bloquear o chaveamento de tarefas do Windows (teclas CTRL+ALT+DEL e ALT+TAB do Windows). Isto é ideal para operadores/usuários de intranet com demandas de segurança
- **Web Smart Clients:** utiliza a tecnologia .NET Smart Client da Microsoft e em clientes remotos é possível instalar com um único clique e sem requisitos de administrador. A aplicação é atualizada automaticamente nos clientes remotos quando atualizado no servidor. O BluePlant

usa todo o potencial do computador remoto e ainda mantém as vantagens de uma instalação centralizada

- **HTML5:** é executado uma aplicação em qualquer navegador como se fosse uma página web. A aplicação é atualizada automaticamente nos clientes remotos quando atualizado no servidor. O BluePlant usa todo o potencial do computador remoto e ainda mantém as vantagens de uma instalação centralizada.

Objetos do *Runtime*

Mais avançada do que a maioria dos sistemas, onde é necessário criar Tags ou variáveis para todas as propriedades internas e customizar lógicas para projetos, o BluePlant permite que as aplicações acessem diretamente todos os objetos do *Runtime* criados no projeto.

Isso significa que Tags temporárias não são necessárias para gerenciar o status dos nós da rede do CP, do número total de alarmes em um grupo ou do número de linhas em um conjunto de dados. É possível acessar os objetos do *Runtime* (que representem um nó da rede), um grupo de alarme ou conjunto de dados e exibir as informações necessárias ou tomar uma ação diretamente através das propriedades integradas.

Isolação de Módulo

Para um melhor desempenho, segurança e confiabilidade, os módulos da UCP com maior consumo e sensibilidade, tais como scripts, conjuntos de dados, dispositivos (drivers de comunicação), relatórios e telas, são executados em seus próprios processos ou no domínio da aplicação em sua própria tarefa, independentemente do banco de dados em tempo real do servidor.

Além das vantagens descritas anteriormente, a arquitetura do BluePlant também permite a distribuição da aquisição de dados da aplicação, ou qualquer aplicação intensiva da UCP em diferentes computadores em um ambiente distribuído, proporcionando uma maior flexibilidade para implementar vários cenários redundantes e conseqüente simplificação da manutenção em campo.

Ferramentas do *Runtime* e Diagnóstico

A ferramenta de monitoração de propriedade permite verificar e simular valores em todos os módulos e objetos e também iniciar e parar todos os módulos individualmente. A ferramenta de janela de rastreamento automaticamente gera mensagens de sistema sobre eventos importantes no *Runtime* e pode ser facilmente estendida para emitir mensagens específicas relacionadas com eventos de script, alterações de dados de Tags/pontos ou ações do usuário. A ferramenta de informações do módulo é uma ferramenta de desempenho e perfil avançado que fornece informações internas de todo o ambiente do *Runtime*.

Implantação e Teste de Projeto

Antes de executar uma aplicação ou projeto, pode ser utilizado o exclusivo "Modo Teste" do BluePlant o qual pode executar o projeto ou a aplicação em um ambiente de teste seguro. No "Modo Teste", não é possível enviar comandos para os controladores remotos (apenas comandos de leitura são enviados), alarmes e historiadores salvam os dados em arquivos temporários e os bancos de dados externos de tempo real são acessados em modo somente leitura.

Após concluir com sucesso o teste, é preciso executar a opção "Inicializar" para obter a funcionalidade completa. Quando o projeto ou aplicação estão prontos para serem implantados no campo, deve ser utilizado o recurso "Publicar" para configurar as opções de redundância (se aplicável) e para criar uma cópia do projeto somente leitura, versão controlada, cópia do projeto para a instalação em campo.

Drivers de Comunicação

Estão disponíveis vários drivers para os principais fabricantes de CPs e sistemas de automação.



Pesquisa na Documentação: características técnicas do BluePlant

Consulte, na documentação do produto, a análise comparativa das características gerais dos modelos BluePlant, as características gerais comuns e o detalhamento de algumas características inovadoras do produto: Banco de Dados em Tempo Real (Tags), Linguagens .NET e Scripts, Alarmes e Segurança, Tendência e Historiadores, Dispositivos e Redes, Conjunto de Dados, Relatórios, Ferramentas e Drivers.

Fonte: Manual de Utilização Série BluePlant – MU224000!

Instalação do BluePlant



Para executar a instalação do software BluePlant é necessário realizar o download do arquivo de instalação a partir do site www.altus.com.br. Em seguida, devem ser fechados todos os programas em execução no PC. Clicar duas vezes no arquivo de instalação.

A primeira tela do instalador será a tela de seleção do idioma do instalador. Clicar em *OK* para continuar a instalação do BluePlant.

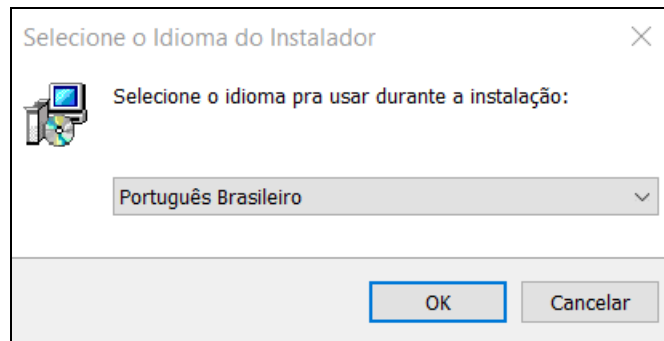


Figura 1-2. Tela de seleção do idioma do instalador

A tela do contrato de licença que aparecerá deve ser lida atentamente. Caso os termos da licença sejam aceitos, selecionar a opção para continuar a instalação. A tela de instalação a seguir irá aparecer, para prosseguir clicar no botão *Próximo*.

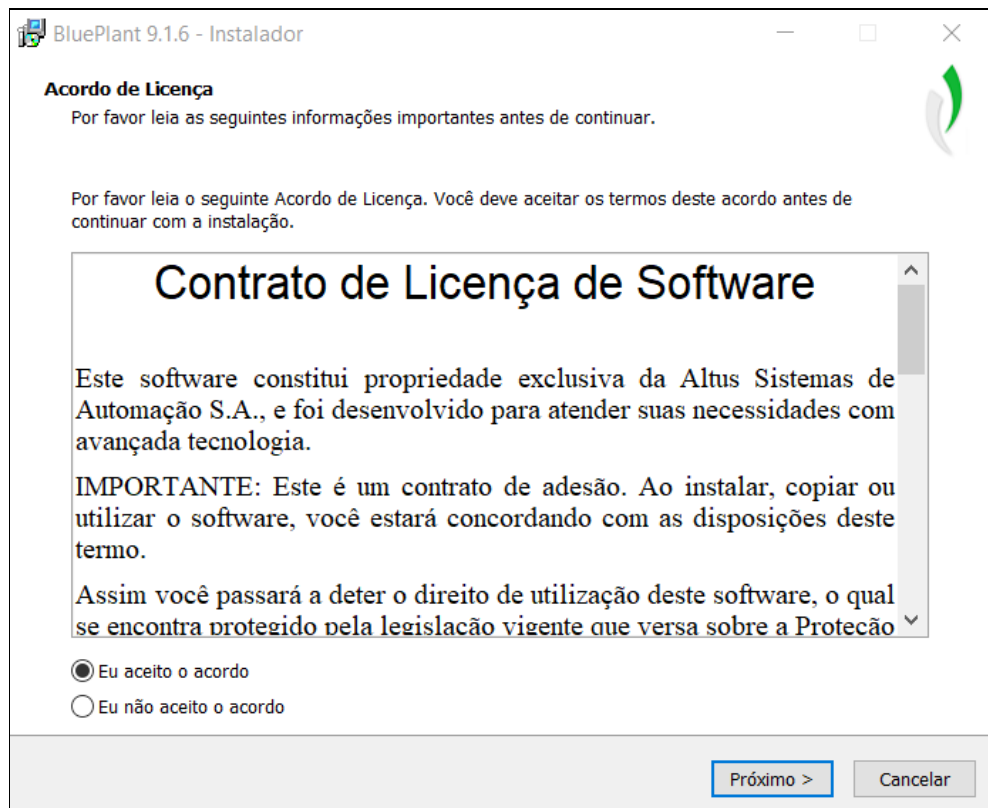


Figura 1-3. Tela do Contrato de Licença do BluePlant

Aparecerá a tela de informações referentes à instalação, clicar no botão *Próximo* para prosseguir.

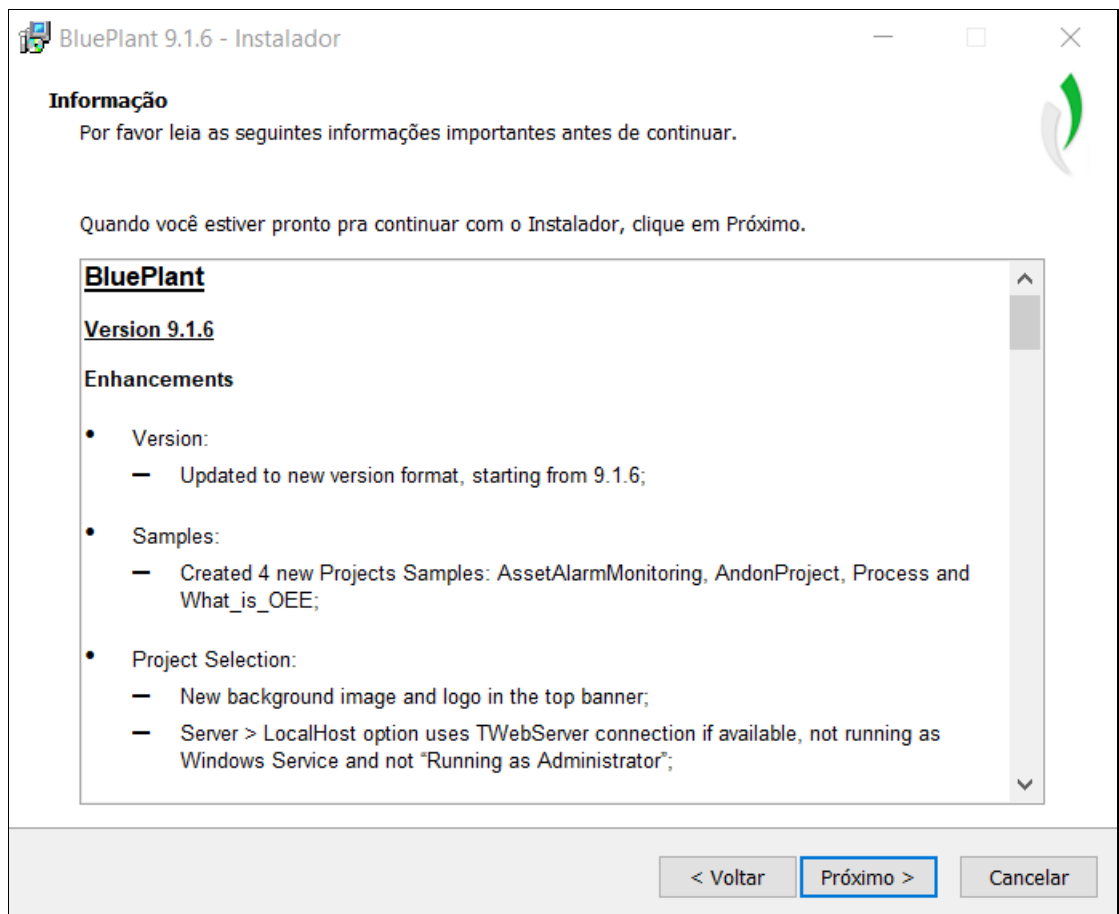


Figura 1-4. Tela de informações referentes à instalação

A próxima tela a aparecer (seleção de tarefas adicionais) permite selecionar a opção pela criação de um atalho na área de trabalho. A seguir clicar em *Próximo*.

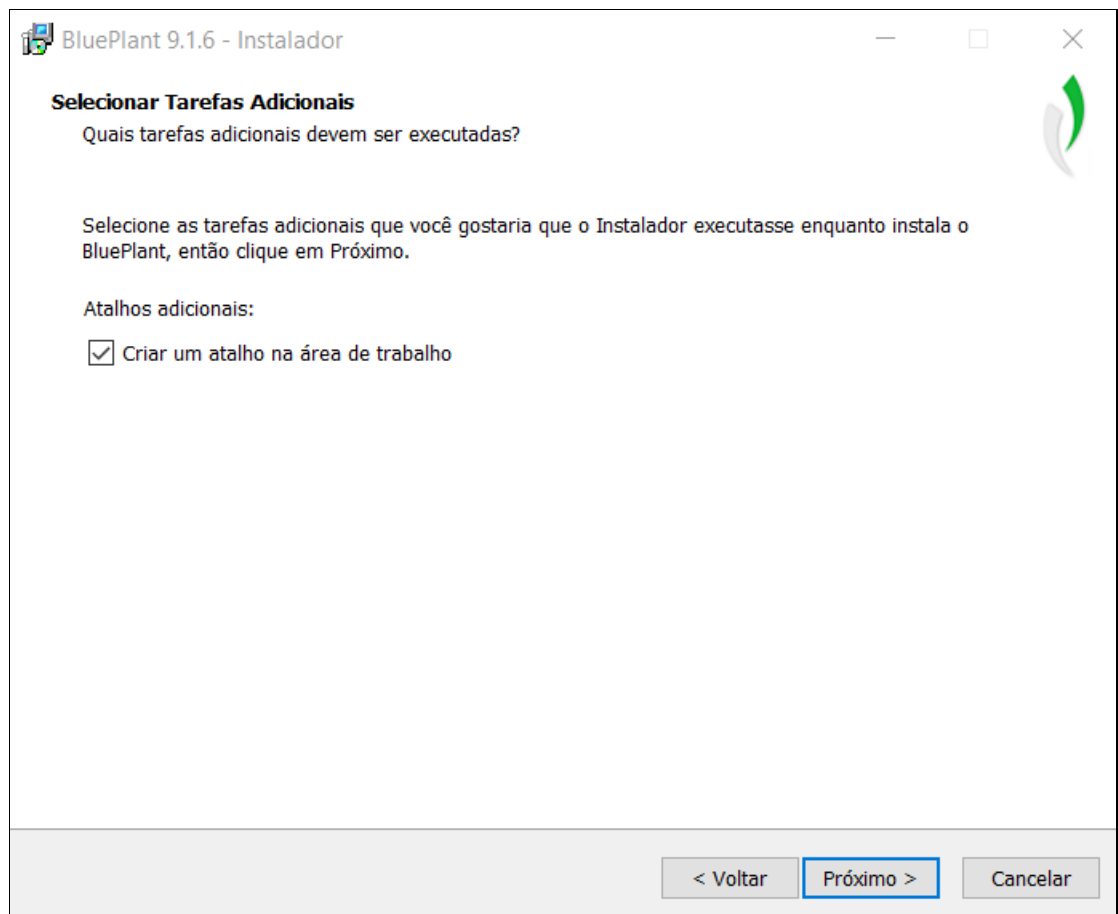


Figura 1-5. Tela de seleção de tarefas adicionais

A próxima tela sumariza as configurações a serem consideradas na instalação, tais como local de destino, tarefas adicionais e pré-requisitos. Ao apertar o botão *Instalar*, inicia a instalação do software.

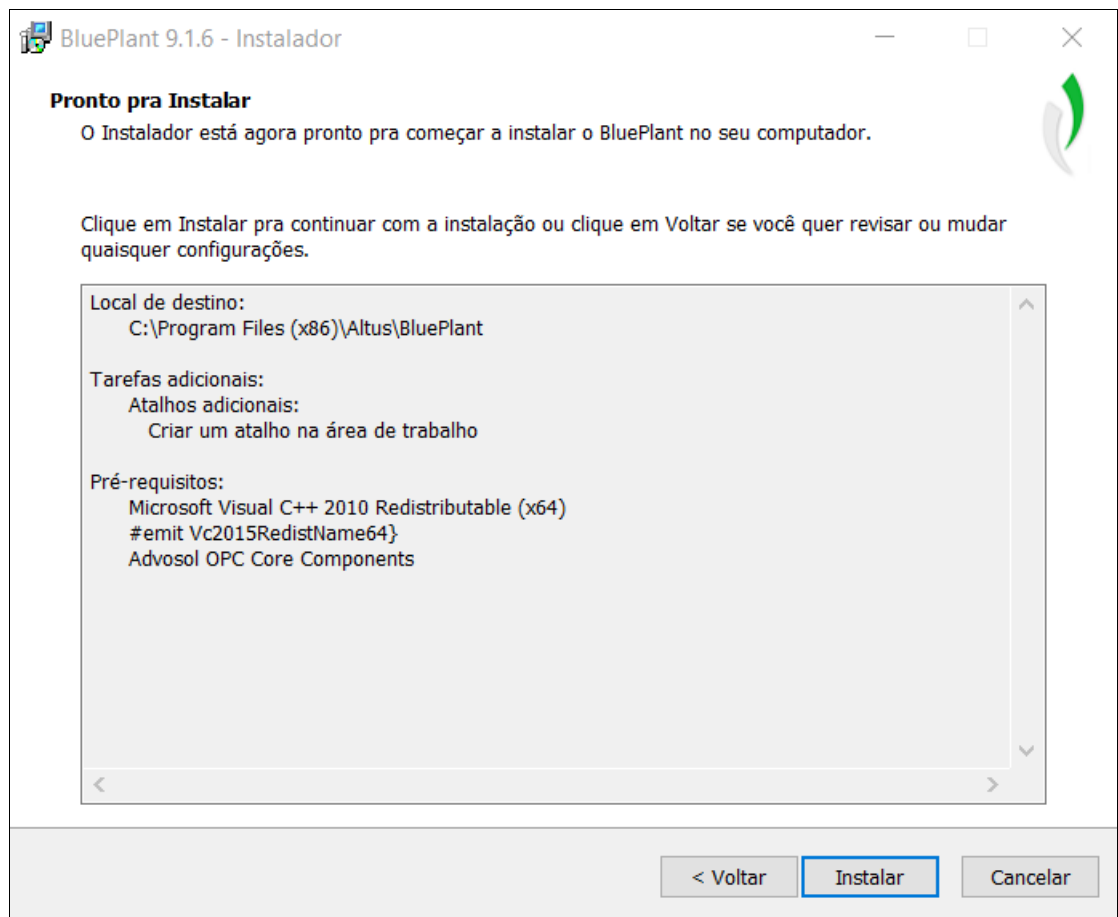


Figura 1-6. Tela de configurações da instalação

Normalmente na primeira instalação é necessário instalar os pré-requisitos do software, que são compostos basicamente pelo Microsoft Visual C++ e *Advosol OPC Core Components*. O progresso da instalação pode ser observado conforme exemplo mostrado na tela a seguir.

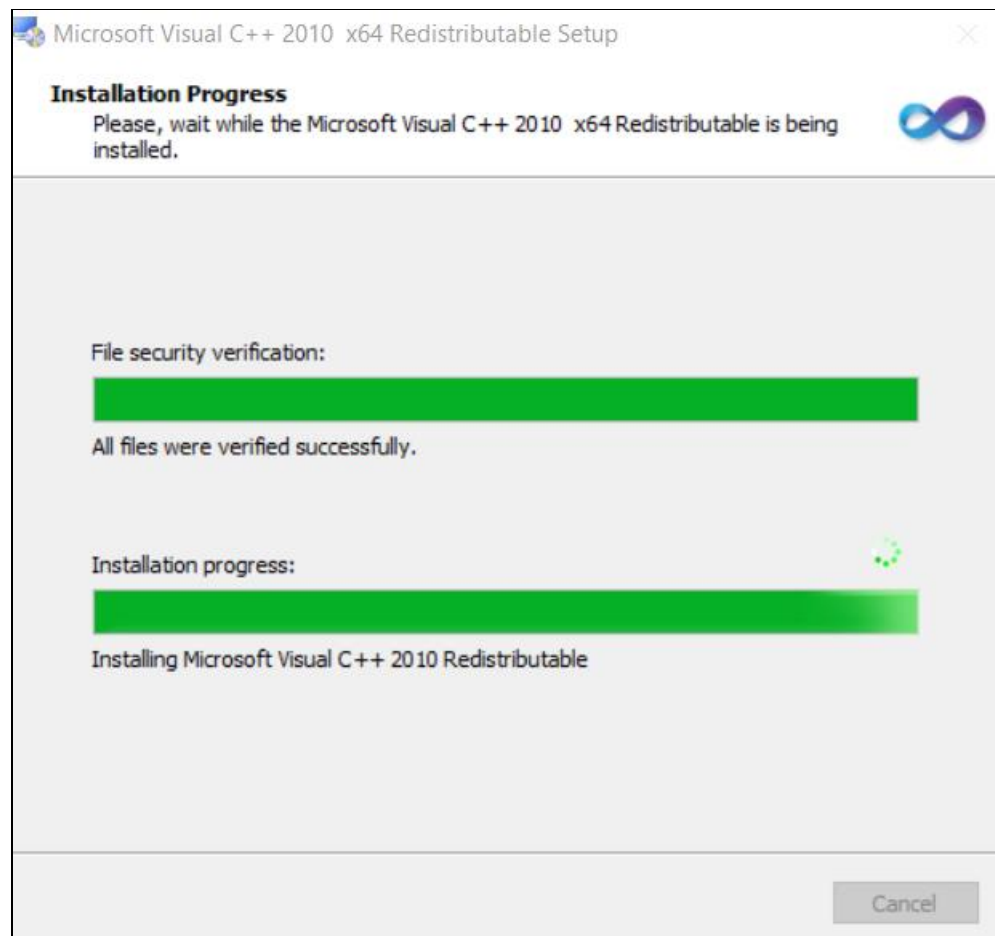


Figura 1-7. Tela de progresso da instalação de pré-requisito (Microsoft Visual C++)

Além do Microsoft Visual C++ é instalado o software *Advosol OPC Core Components*. Este software oferece um suporte completo na plataforma .NET para todos os principais padrões OPC. OPC é um conjunto de especificações normatizadas. Tais padrões resultaram da colaboração de vários dos principais fornecedores de automação em todo o mundo trabalhando em cooperação com a Microsoft. A especificação definiu um conjunto padrão de objetos, interfaces e métodos para uso em controle de processos e aplicações de automação para facilitar a interoperabilidade. Atualmente, existem centenas de clientes e servidores OPC Data Access.

Após a conclusão da instalação do *Advosol OPC Core Components*, inicia a instalação do BluePlant. Aguardar enquanto os arquivos necessários são instalados no computador. Isso pode levar alguns minutos dependendo da configuração do computador.

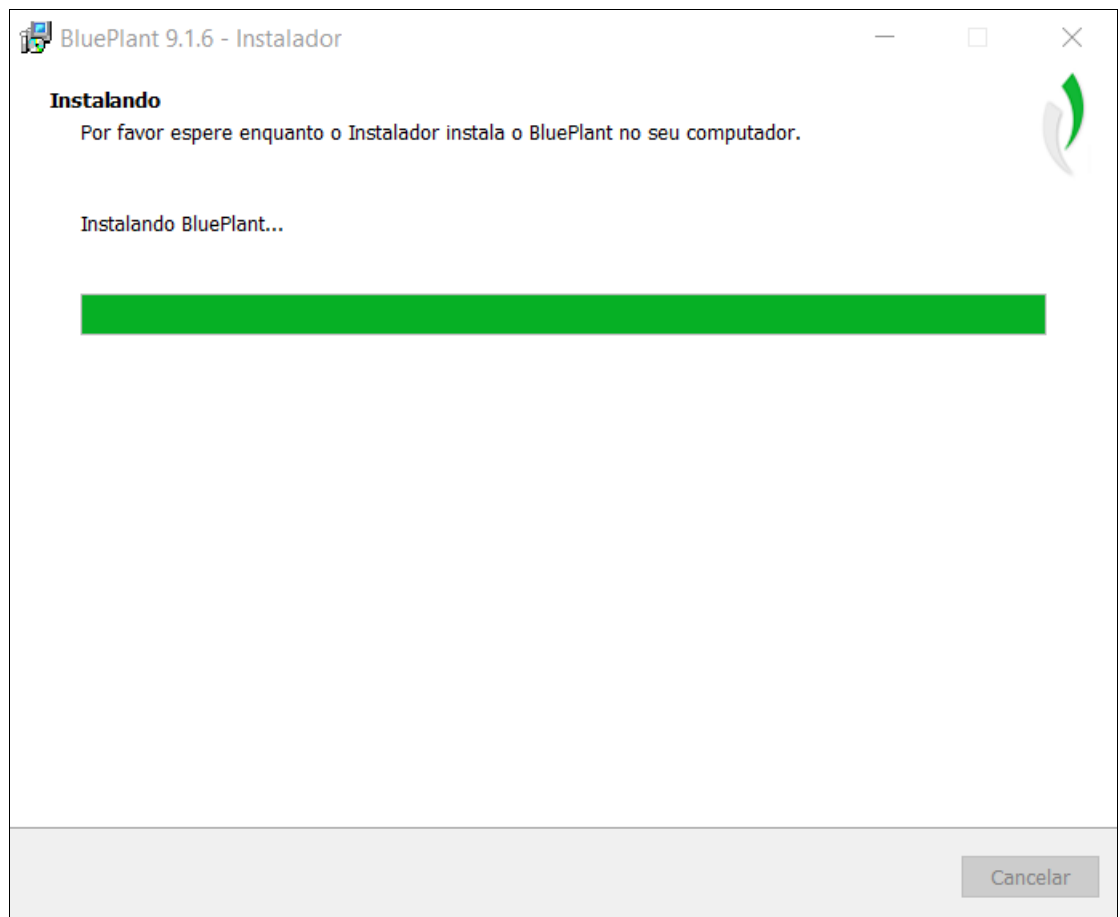


Figura 1-8. Tela de Instalação do BluePlant

Após a conclusão da instalação, a tela a seguir será exibida. Selecionar a opção de reinício desejada e clicar no botão *Concluir* para finalizar o procedimento de instalação.

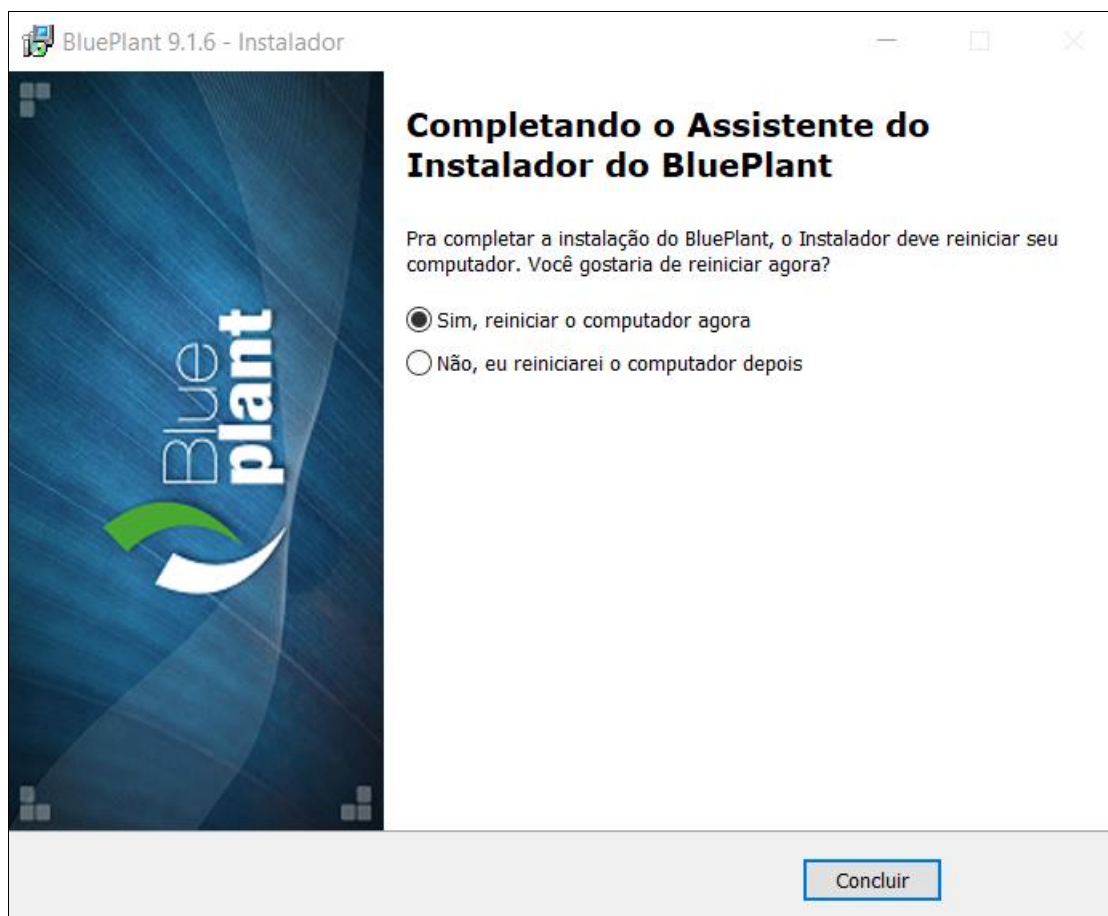


Figura 1-9. Tela de Encerramento de Instalação do BluePlant

O BluePlant está instalado e pronto para usar. Para executá-lo, clicar no atalho "BluePlant", criado durante a instalação, no Menu Iniciar.

Licenças e Hardkey

O BluePlant possui alguns modelos onde suas características são habilitadas através de licenças presentes em uma *hardkey*. A *hardkey* é um meio físico onde estão presentes as licenças requisitadas e para que o BluePlant libere as funcionalidades presentes na *hardkey*, a mesma deve ser conectada ao servidor onde será executado o projeto.

Na documentação do produto podem ser conferidos os modelos BluePlant disponíveis e exceto o modelo Express, todos demais são habilitados por licenças presentes em *hardkey*.

Uma vez que a *hardkey* é conectada no servidor, as informações podem ser visualizadas através da guia *License*.

ATENÇÃO:

Devem ser tomados cuidados para que a *hardkey* não seja danificada, pois nestes casos se o acesso às informações presentes na *hardkey* for comprometido, as funcionalidades habilitadas não mais estarão disponíveis. Neste caso deve-se entrar em contato com o Suporte Técnico da Altus através do [site www.altus.com.br](http://www.altus.com.br) ou e-mail altus@altus.com.br.

2. Características Operacionais



Primeiros Passos com o BluePlant



Uma vez instalado o BluePlant no computador, um duplo clique no ícone apresentado abaixo, permite abrir a ferramenta.



Figura 2-1. Ícone do BluePlant

Gerenciamento de Projeto

Após a inicialização do BluePlant deve aparecer a janela de Gerenciamento do Projeto.

No centro da tela inicial, em sua parte superior, encontram-se as guias denominadas *Projects*, *Server* e *License*.

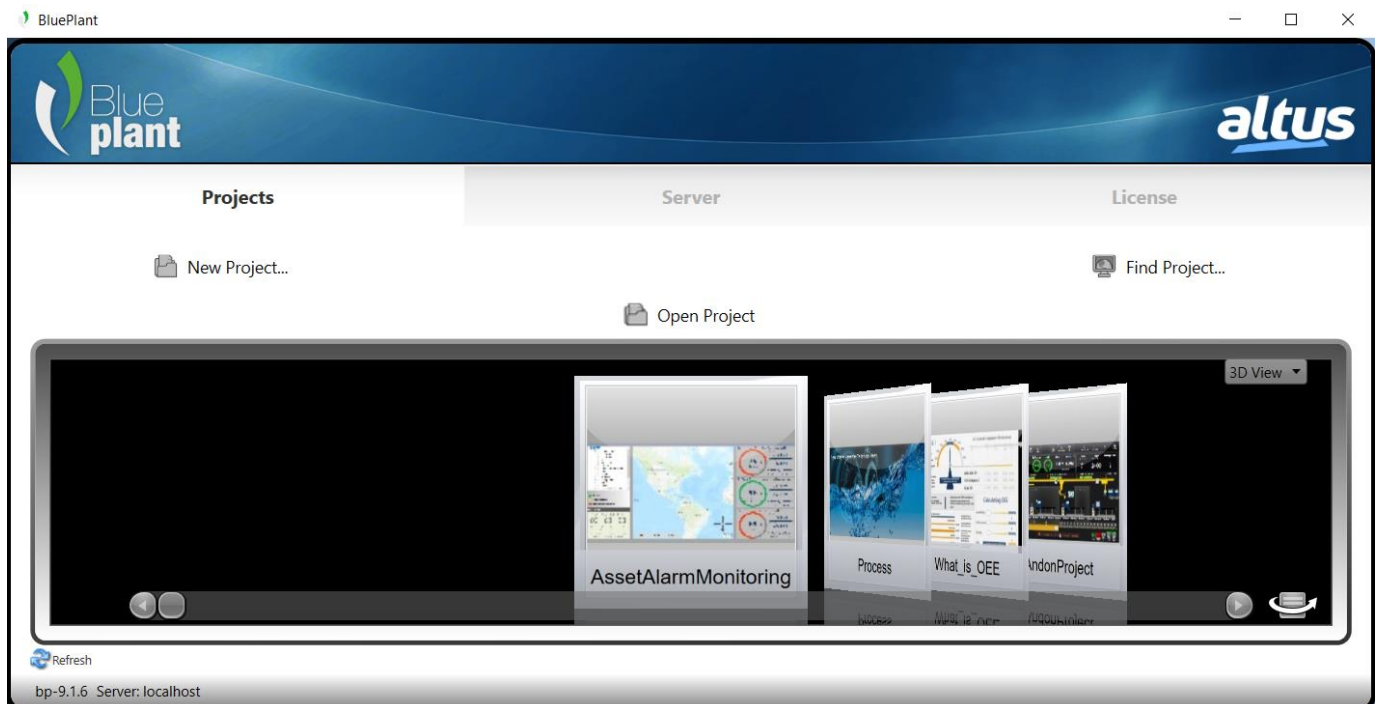


Figura 2-2. Tela de Gerenciamento de Projeto

Em seguida aparece a estrutura de menus padrão Web do BluePlant. Este design permite um acesso mais eficiente ao leque de ferramentas, reduzindo a quantidade de cliques. Além disso, uma tabela lista os projetos que podem ser acessados.

Na guia “*Projects*”, estão os ícones para criar um projeto *New Project*, para abrir um projeto selecionado *Open Project...* e para localizar outros projetos *Find Project...* que não estão localizados no diretório padrão do BluePlant.

A guia *Server* tem a aparência mostrada na figura a seguir onde duas opções estão disponíveis: *Localhost* e *Remote*. A primeira significa que o projeto será executado localmente e a segunda que o projeto será executado a partir de um servidor remoto.

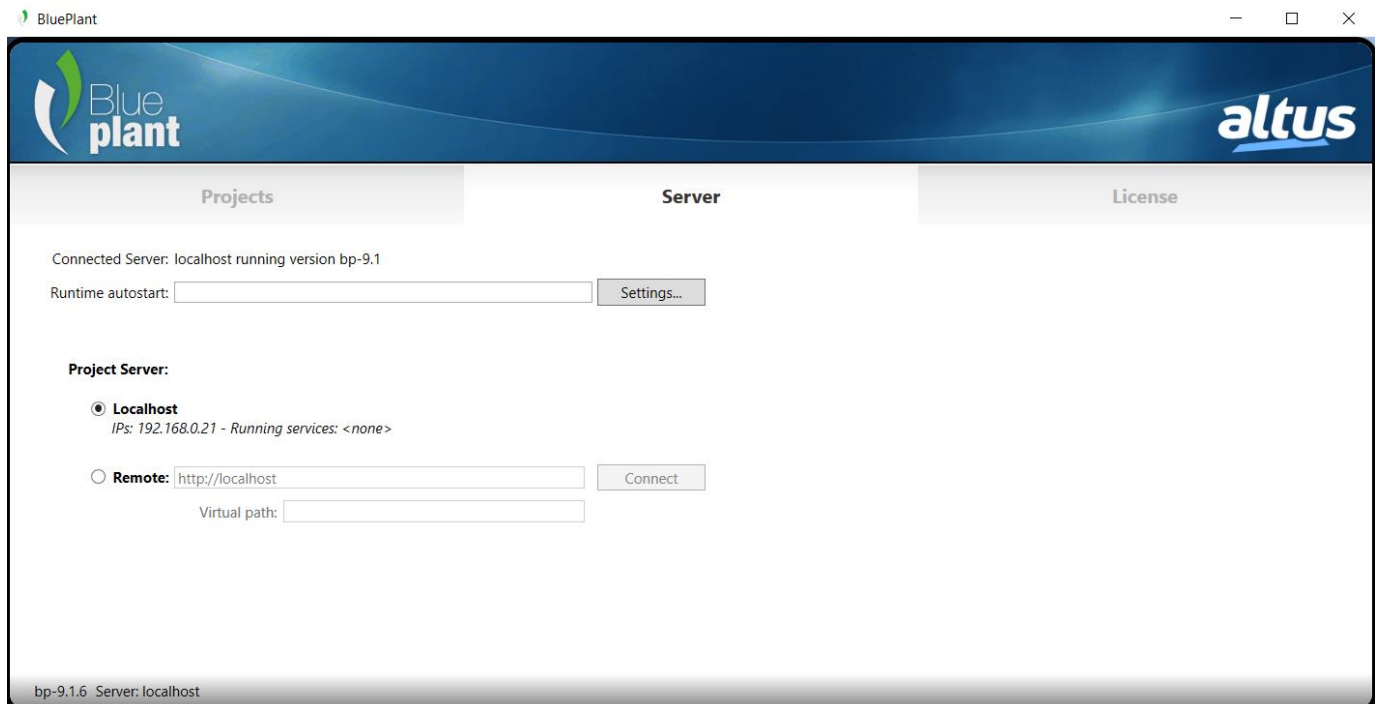


Figura 2-3. Guia Servidor

A terceira guia refere-se à licença do produto, incluindo o modelo, a versão do produto, família, número de série, código do produto entre outras informações presentes na hardkey como pode ser visto na figura a seguir.

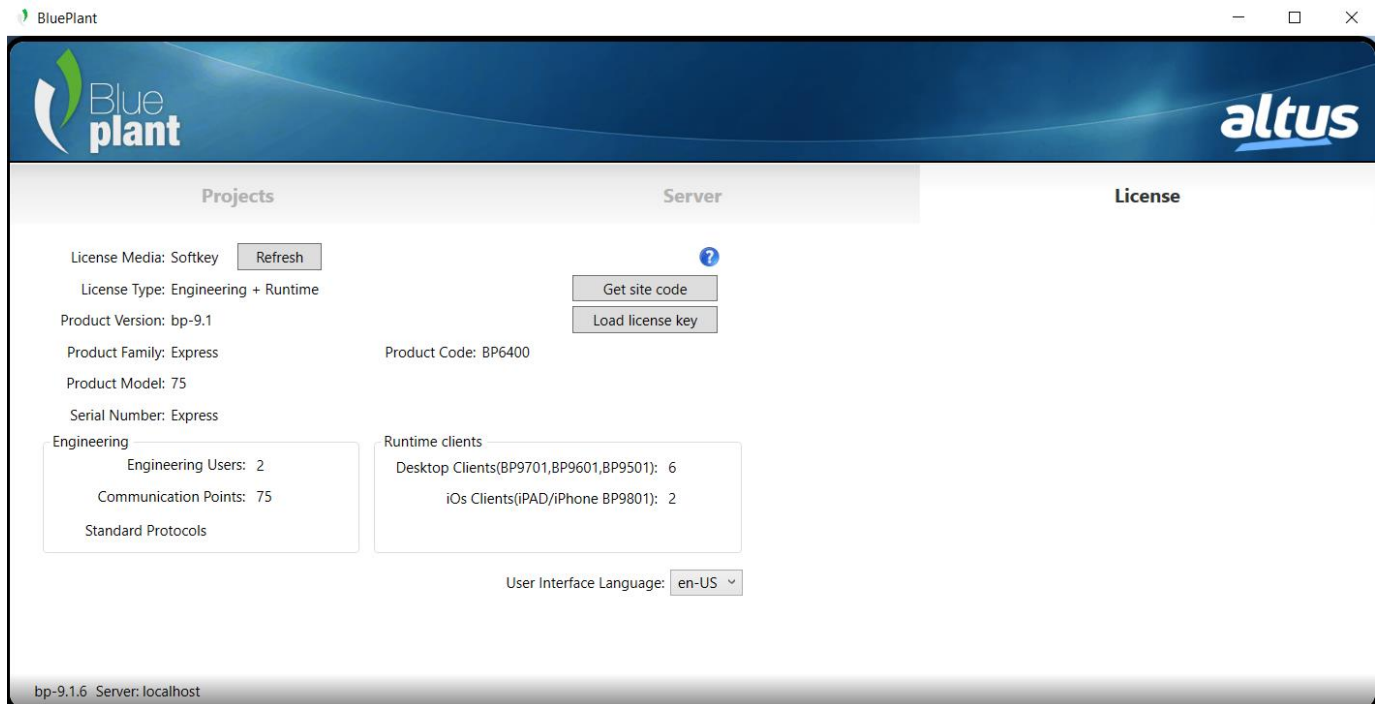


Figura 2-4. Guia Licença

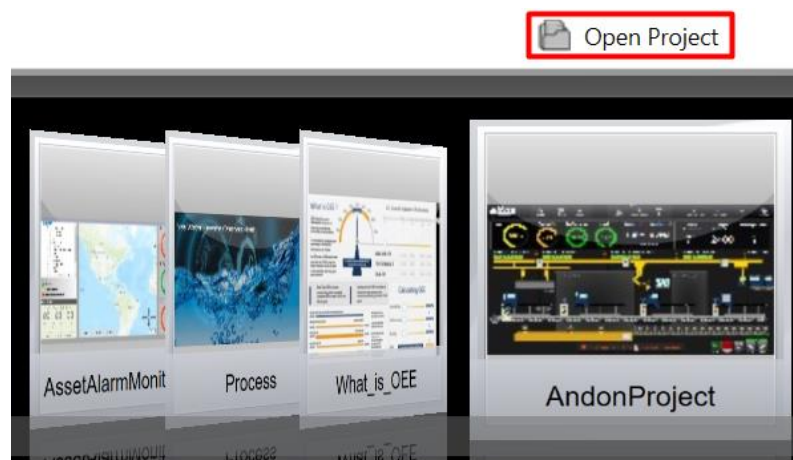
Abrindo ou Criando Projetos

Esta seção apresenta uma sequência de operações para a criação de um projeto simples ou abertura de um projeto existente. Junto com a ferramenta é disponibilizado um demo que permite uma visão simplificada dos recursos disponíveis no BluePlant. O detalhamento das funcionalidades ilustradas aqui será explorado no decorrer deste tutorial.



Estudo Dirigido 2-1: visão geral das funcionalidades do BluePlant

A partir da guia *Projects* no editor de projetos faça um “tour” virtual nos projetos de demonstração disponíveis.



DICA: consulte o Instrutor para orientações adicionais

Estudo Dirigido 2-2: primeiros passos para utilizar o BluePlant



Desenvolva, com o auxílio do instrutor, um projeto simples visando demonstrar ações básicas na utilização do software BluePlant, como: criação de um projeto, adição de Tags e desenvolvimento da tela do supervisor. Para isso, crie um projeto com duas Tags (uma do tipo *Digital* e outra do tipo *Text*), vinculando-as a objetos de tela para tornar possível a interação delas com o usuário.

📌 **DICA:** caso você queira saber mais sobre o desenvolvimento de um projeto no BluePlant, assista o Webinar a seguir. Nele é descrita a criação de um projeto no supervisor BluePlant, bem como a criação de Tags e a inserção de objetos na tela.



Criando um Novo Projeto

As configurações básicas do projeto a ser criado são definidas a partir da janela de abertura do BluePlant clicando em *New Project*.

Na janela seguinte, apresentada na figura a seguir, estão indicados os passos para criar um novo projeto.

Figura 2-5. Criando um Novo Projeto

Onde:

1. Indicação de novo projeto;
2. Definição do nome do projeto, descrição e diretório onde será gerado o projeto;
3. Configurações iniciais tais como: família e modelo do produto, plataforma, cultura e código (script) padrão;
4. Ajustes da resolução da tela;
5. Botão para criar um novo projeto e iniciar o desenvolvimento da aplicação.

O novo projeto criado está referenciado na lista da janela de abertura do BluePlant.

🔔 NOTAS:

1. O Campo Nome tem limite de no máximo 64 caracteres.
2. Campo Descrição tem limite de no máximo 256 caracteres.
3. O campo Nome não aceita os seguintes caracteres especiais: \:*/"?'<>_

Interface do Usuário

A configuração da interface do usuário opera como uma página da Web front-end onde se pode facilmente navegar entre os itens e editá-los. Os dados são salvos automaticamente no *back-end*, sem a necessidade de abrir/salvar/fechar os menus. A interface do usuário contempla, à esquerda, o menu principal no topo e o submenu logo abaixo o qual se altera quando se seleciona outro item do menu principal. Além disso, à direita, estão as guias da área de trabalho e, mais acima, uma barra de links rápidos e itens recentes. A figura a seguir ilustra a interface do usuário.

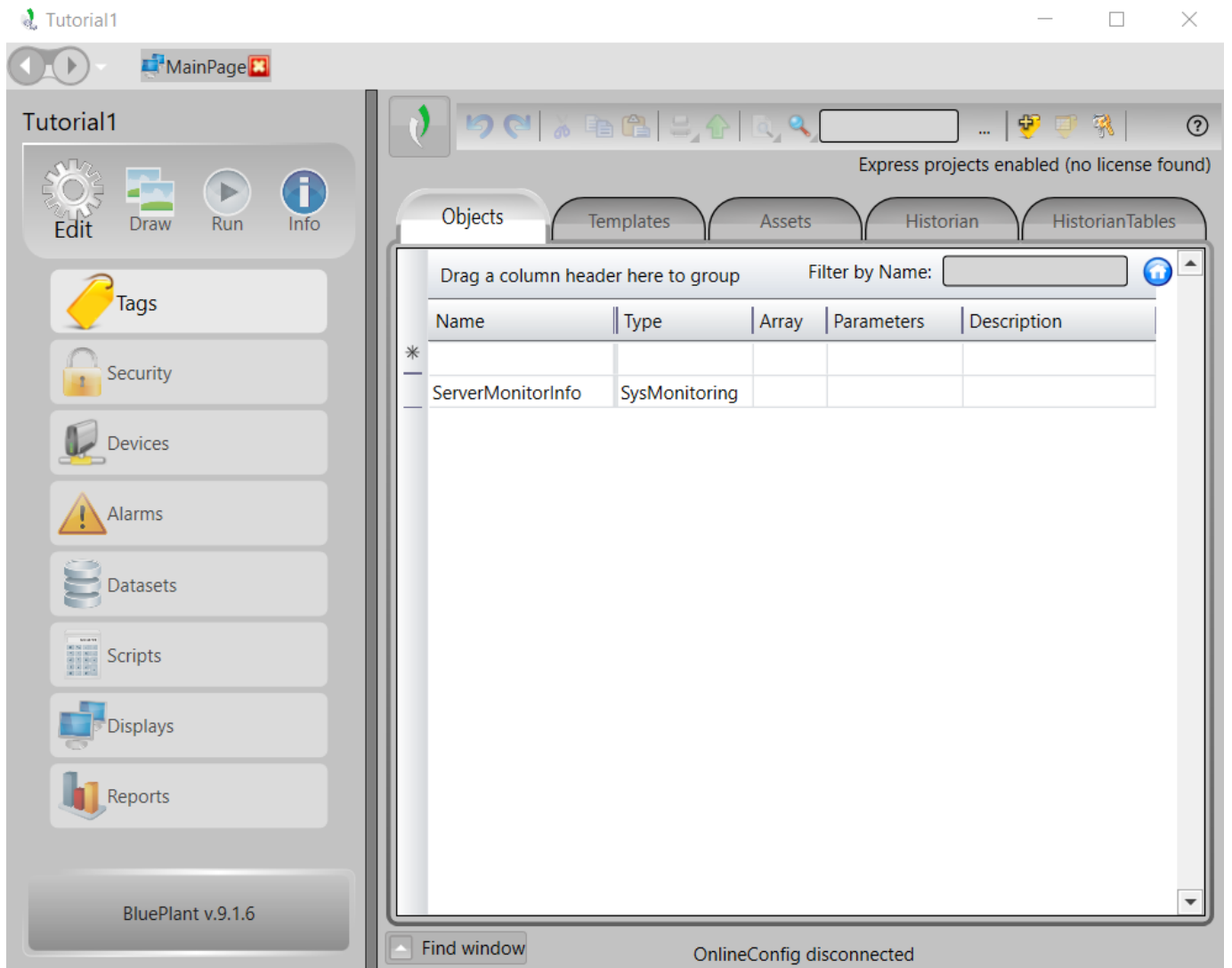


Figura 2-6. Interface do Usuário

Itens do Menu Principal



Menu Edit

Para configurar os elementos básicos do seu projeto a partir do Menu Principal clique no menu *Edit*. As funcionalidades associadas a esse menu estão detalhadas na sequência.

- *Tags*: define/Edita Tags a serem usadas no projeto;
- *Security*: define permissões para o usuário e níveis de acesso ao projeto;
- *Devices*: configura Protocolos de Comunicação e Aquisição de Dados do Dispositivo;
- *Alarms*: configura Tags de Alarme e Grupos de Alarme para gerar Alarmes sob determinadas condições;
- *Datasets*: define fontes de dados tais como Arquivos, Bases de dados, Tabelas específicas e Consultas pré-definidas;
- *Scripts*: configura Scripts .NET em VB.NET ou C-Sharp;
- *Displays*: seleciona telas gráficas dinâmicas para configuração;
- *Reports*: configura Relatórios que incluem símbolos gráficos atualizados nos formatos XPS, HTML, ASCII e Unicode.

A figura a seguir ilustra as funcionalidades desse menu.

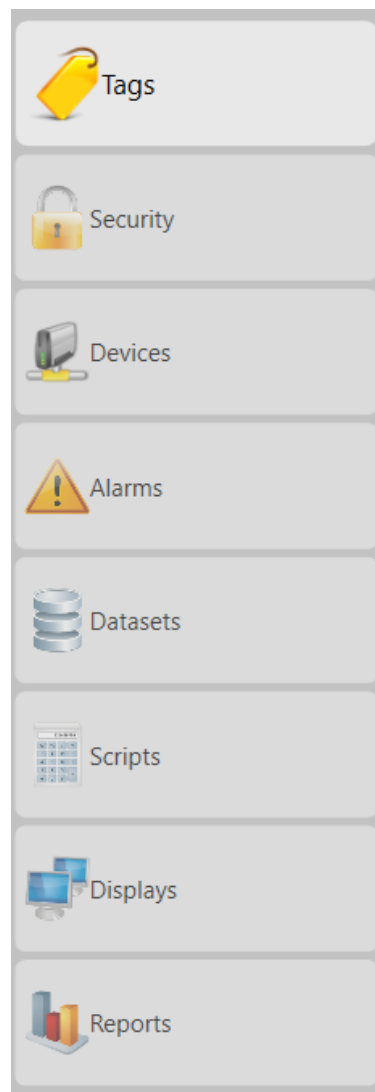
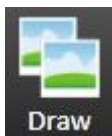


Figura 2-7. Ícones Menu Editar



Menu Draw

Para configurar telas gráficas dinâmicas clique no ícone *Draw* do menu principal.

Os parâmetros de Aparência ditam o estilo do pincel ao desenhar o objeto na tela, inclusive sua cor. A opção *Fill* permite especificar cores, gradientes ou outros objetos com os quais o gráfico pode ser preenchido.

As Configurações da Tela ditam os seus atributos. As telas podem ser configuradas como PopUps, Diálogos ou telas normais.

Assim como qualquer objeto gráfico as telas podem apresentar uma variedade de cores e padrões, bordas, títulos e outros.

Uma característica especial é que você pode especificar uma animação de transição entre as telas, da mesma forma que no PowerPoint.

A figura a seguir ilustra esse menu.

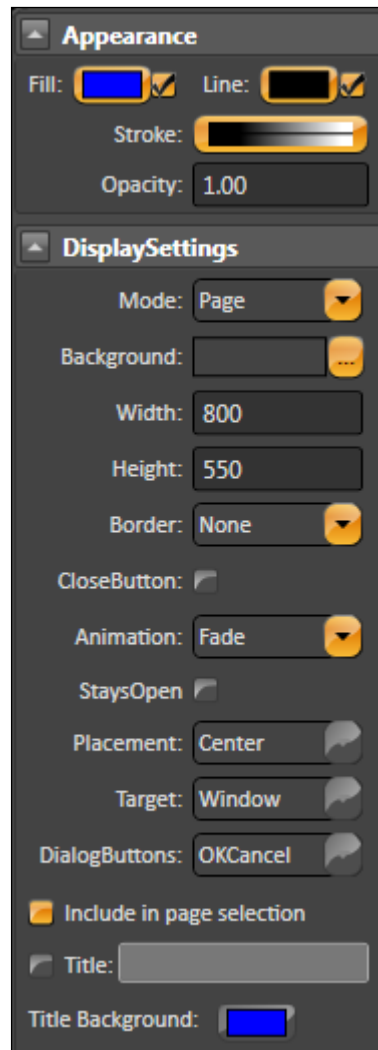


Figura 2-8. Menu Draw



Menu Run

O ambiente Run permite acesso a todos os parâmetros de execução do projeto. As funcionalidades a seguir estão contempladas:

- Prepara um Projeto para ser executado e o salva no histórico de compilação;
- Testa o projeto antes de publicá-lo para o uso no servidor;
- Especifica quais ferramentas de Diagnóstico e Módulos serão usados na execução do projeto;
- Publica o projeto para habilitar sua instalação final. Atualiza os números de Revisão de Publicação do projeto;
- Conta o número de Tags usadas no projeto. Fornece referência cruzada de objetos no projeto;
- Customiza seu projeto para o uso de outros idiomas nas Mensagens de Alarme e texto na tela;
- Importa dados de configuração já criados para Servidores OPC, qualquer tabela de configuração ou projeto inteiro.

A figura a seguir ilustra as funcionalidades desse menu.

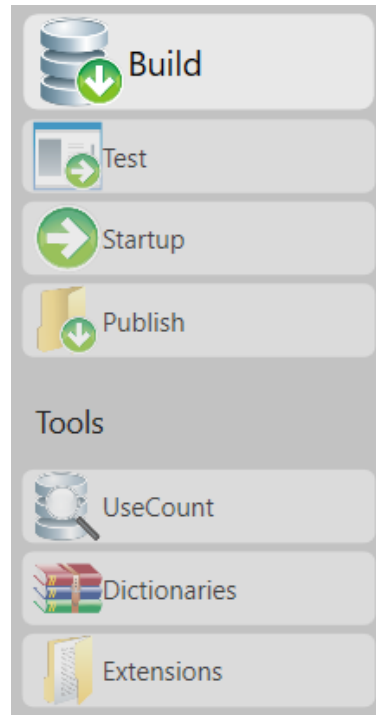


Figura 2-9. Menu Run



Menu Info

As seguintes funcionalidades estão disponíveis no menu Info:

- Fornece acesso aos detalhes do projeto atual: versão do produto e do projeto, localização, idioma local e informações de licença;
- Localiza informações relacionadas ao número de liberação do projeto e alterações no mesmo;
- Deixa notas arbitrárias para outros engenheiros do projeto;
- Fornece uma Guia de referência da ajuda do BluePlant;
- Exibe as informações específicas de licença. As guias de registro auxiliam no procedimento de registro online do produto.

A figura a seguir ilustra as funcionalidades desse menu.

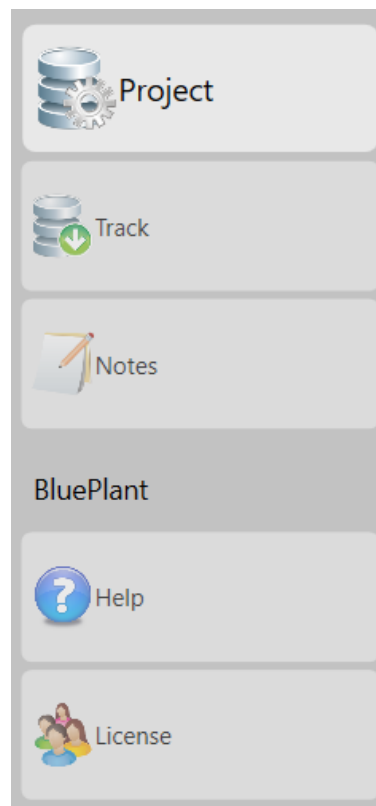


Figura 2-10. Menu Info

A seguir vamos explorar algumas funcionalidades básicas associadas a esses menus como criação de Tags, inclusão de objetos na tela e execução do projeto.

Cada sistema de supervisão gerencia certo número de objetos também denominados entidades que descrevem as variáveis de processo controladas e os elementos habituais do controle. A atividade de configuração de um supervisor compreende usualmente duas etapas:

1. Definir cada variável de processo na base de dados;
2. Definir sinóticos, gráficos e relatórios.

No sistema existem variáveis simples, primitivas e compostas, estas últimas formadas a partir das primeiras. Nesse contexto o nome da variável recebe o nome de Tag.

Na sequência vamos explorar algumas funcionalidades básicas associadas aos menus básicos do BluePlant: *Edit*, *Draw*, *Run* e *Info*, os quais contêm as ferramentas necessárias para o desenvolvimento de um projeto de um sistema de supervisão contemplando as etapas típicas indicadas anteriormente.

Implementação de um Projeto no BluePlant



Criando Tags no Projeto

No ambiente de edição é possível editar objetos como, por exemplo, Tags. A figura a seguir mostra o caminho para a edição de Tags. Neste exemplo, foram criadas duas Tags analógicas: o valor bruto de um medidor de potência elétrica a ser lido do controlador programável denominado de Power_PV e o valor de engenharia desta medição chamado de Power_ENG.

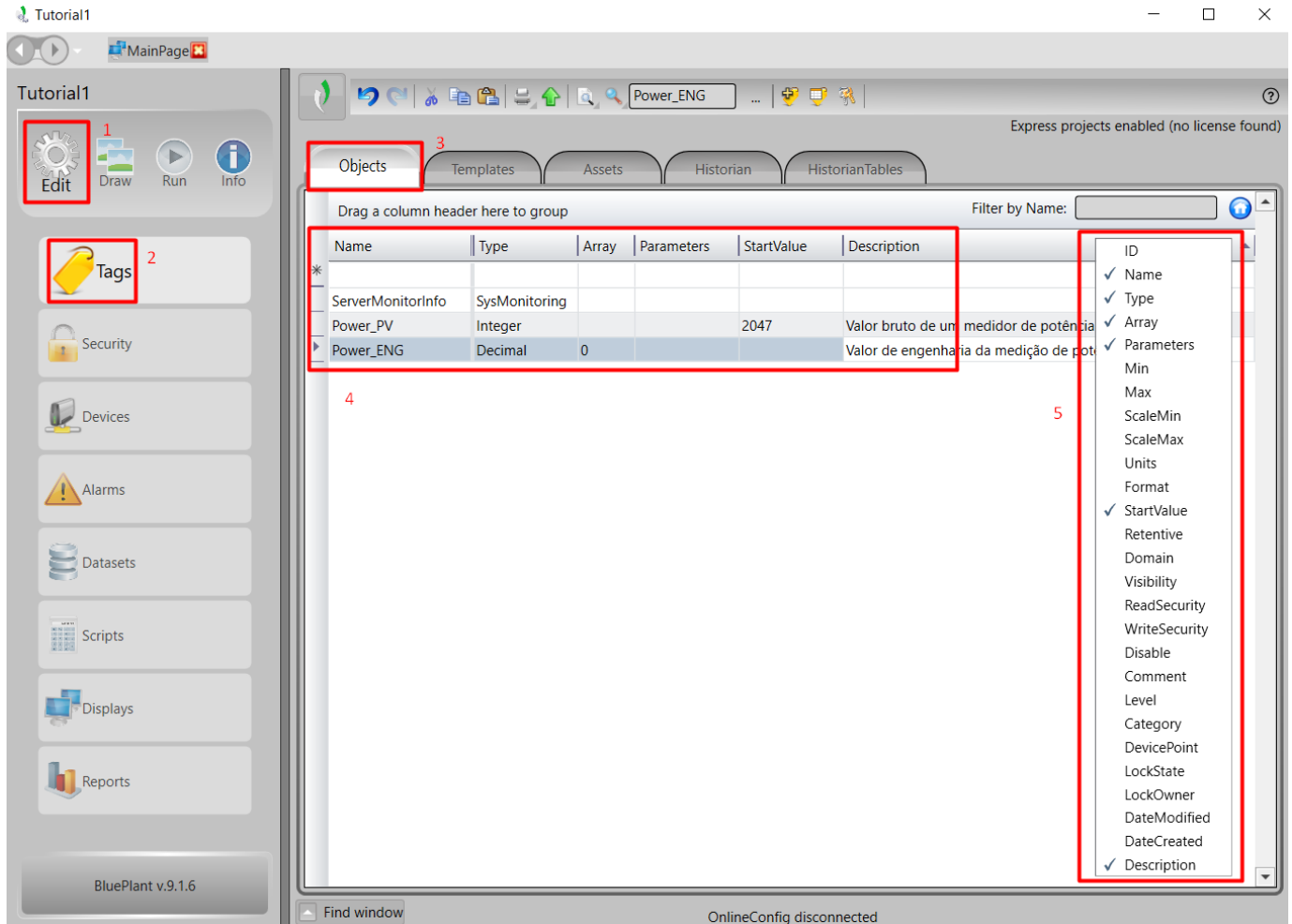
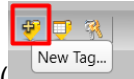


Figura 2-11. Editando Tags do Projeto

Os passos para a criação dessas Tags estão listados a seguir conforme numeração indicada na figura anterior.

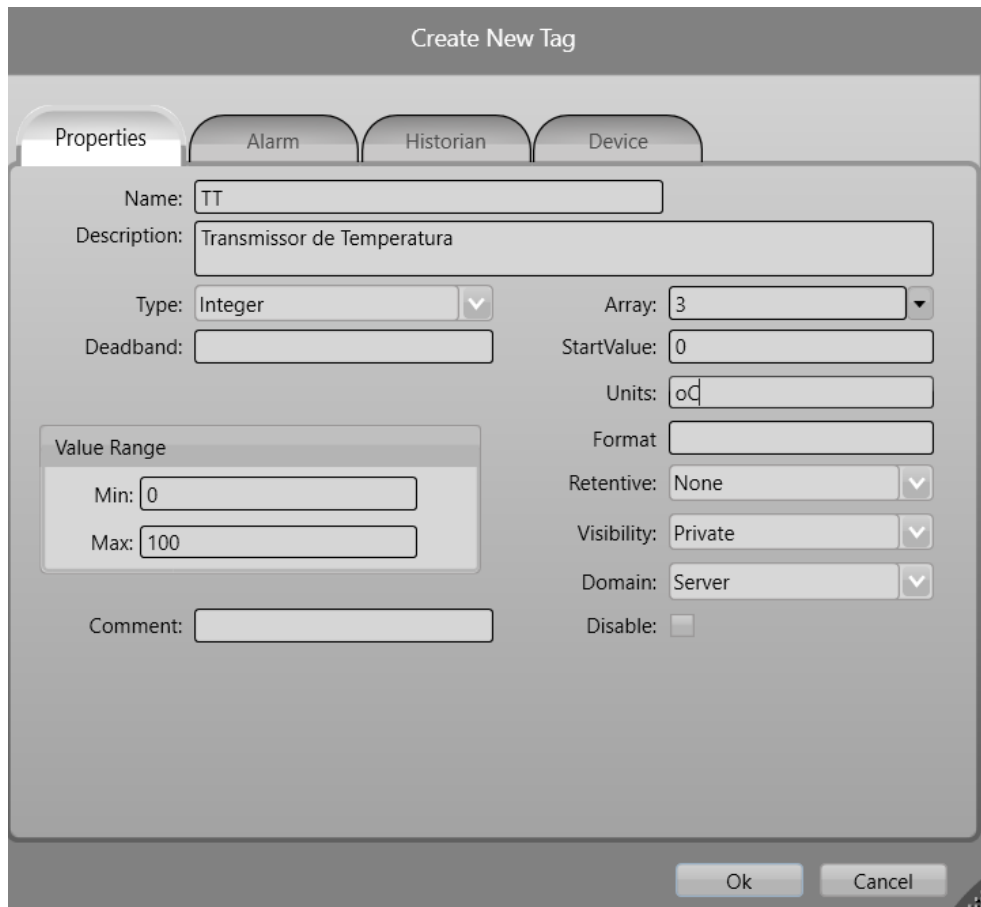
1. Acessar o menu *Edit*;
2. Selecionar a opção *Tags*;
3. Acessar a aba *Objects* na qual aparecerá a tabela com as Tags do usuário;
4. Digitar os dados relativos à nova Tag da aplicação na linha marcada com um asterisco (*). Para confirmar a inclusão na tabela deve ser pressionada a tecla ENTER;
5. Selecionar as opções de configuração das Tags através das colunas da tabela desejadas clicando com o botão direito do mouse no cabeçalho de uma coluna e marcando os itens para exibição.

Também é possível criar uma nova Tag acessando a barra de ferramentas no topo do ambiente



de edição. Basta clicar no ícone () e configurar os parâmetros da Tag.

A figura a seguir mostra o processo de criação de uma Tag vetor de uma dimensão, com três posições do tipo analógico inteiro a partir do Menu Editar. Uma Tag vetor é um conjunto de Tags com o mesmo nome, cuja identificação se dá através de índices. Neste exemplo, o acesso em outros locais da aplicação a essa Tag, se dá via sintaxe `TT[0]`, `TT[1]`, `TT[2]` e `TT[3]`.

The image shows a 'Create New Tag' dialog box with several tabs: 'Properties', 'Alarm', 'Historian', and 'Device'. The 'Properties' tab is active. It contains the following fields:

- Name: TT
- Description: Transmissor de Temperatura
- Type: Integer (dropdown menu)
- Array: 3 (dropdown menu)
- Deadband: (empty text box)
- StartValue: 0
- Units: °C
- Format: (empty text box)
- Value Range section with Min: 0 and Max: 100
- Retentive: None (dropdown menu)
- Visibility: Private (dropdown menu)
- Domain: Server (dropdown menu)
- Disable: (checkbox, unchecked)
- Comment: (empty text box)

At the bottom right, there are 'Ok' and 'Cancel' buttons.

Figura 2-12. Processo de Criação de Tags

Associando Tags a um protocolo comunicação

Para realizar a associação das Tags criadas a um protocolo de comunicação, os passos descritos a seguir devem ser executados.

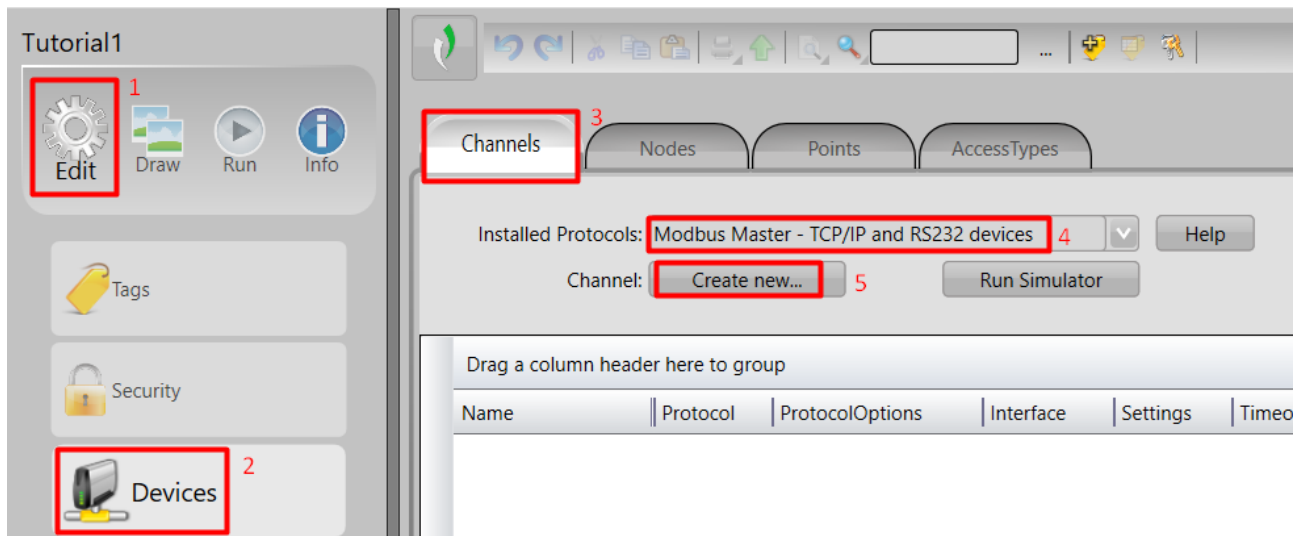


Figura 2-13. Processo de Criação de Canal de Comunicação

1. Acessar o menu *Edit*;
2. Selecionar a opção *Devices*;
3. Acessar a guia *Channels* na qual deve aparecer o campo com os protocolos disponíveis;
4. Selecionar no campo *Installed Protocols* o protocolo de comunicação a ser utilizado;
5. Para confirmar a criação do canal, clicar no botão *Create new...*

Uma vez o protocolo selecionado e confirmada a criação do canal, uma janela com as propriedades do protocolo deve aparecer, como pode ser visto na figura a seguir.

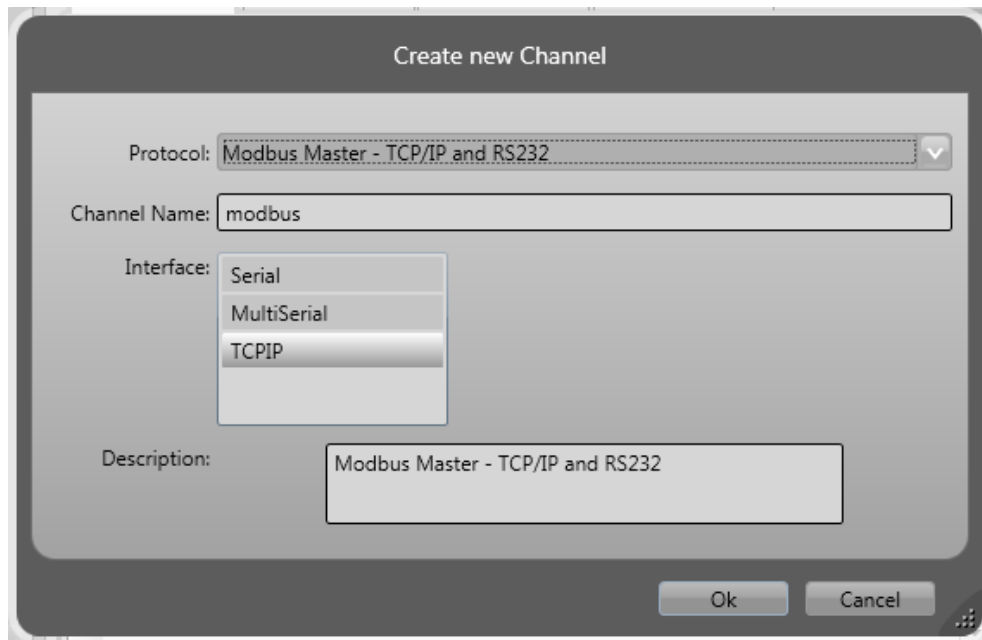


Figura 2-14. Opções de Protocolos para Criação de um Canal

Na guia *Nodes* são criados os nós de acordo com os canais existentes.

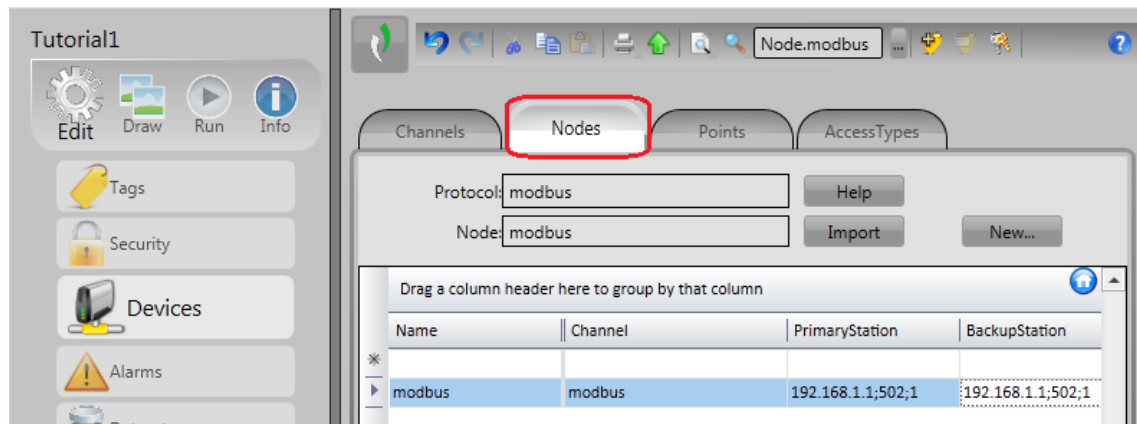


Figura 2-15. Inclusão e Configuração de Nós

Na guia *Points* são criadas as associações entre as Tags, o nó e os endereços do protocolo, além de configurações de tipos de acesso, escalas e tipos de dados.

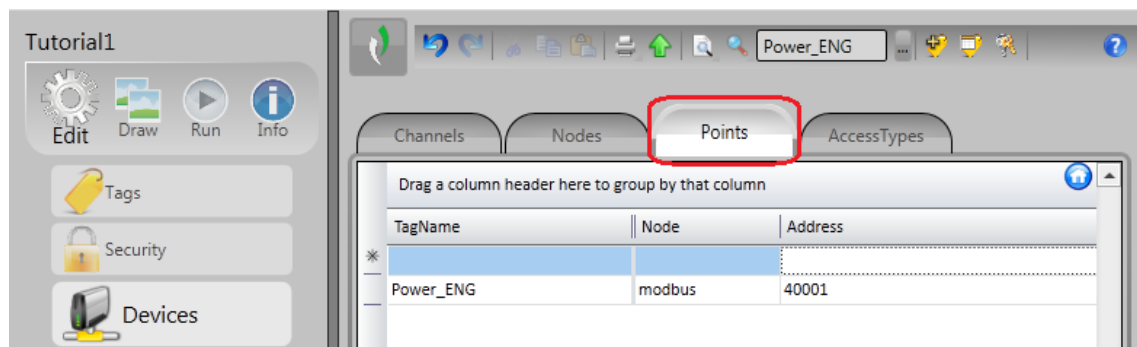


Figura 2-16. Associação de Tags com Nós e Endereços de Protocolos

Ao selecionar o campo *TagName* da guia *Points*, deve abrir uma janela onde estão todas as Tags criadas e que podem ser selecionadas para associar ao nó criado.



Incluindo Objetos na Tela Principal

No menu *Draw* é possível criar as telas que compõem a aplicação. As telas são formadas por objetos, tais como símbolos e textos.

Vamos explorar a inclusão desses dois tipos de objetos na tela principal padrão.

Inclusão de Símbolo

Para realizar a inclusão de um símbolo do tipo tacômetro na tela principal (MainPage), as seguintes etapas podem ser contempladas:

1. Selecionar Menu *Draw*;
2. Acessar a aba *Drawing*;
3. Abrir a biblioteca de símbolos;
4. Selecionar o símbolo do tacômetro, posicionando-o no local desejado;
5. Configurar as propriedades do símbolo.

Esses passos estão referenciados na figura a seguir conforme a numeração correspondente.

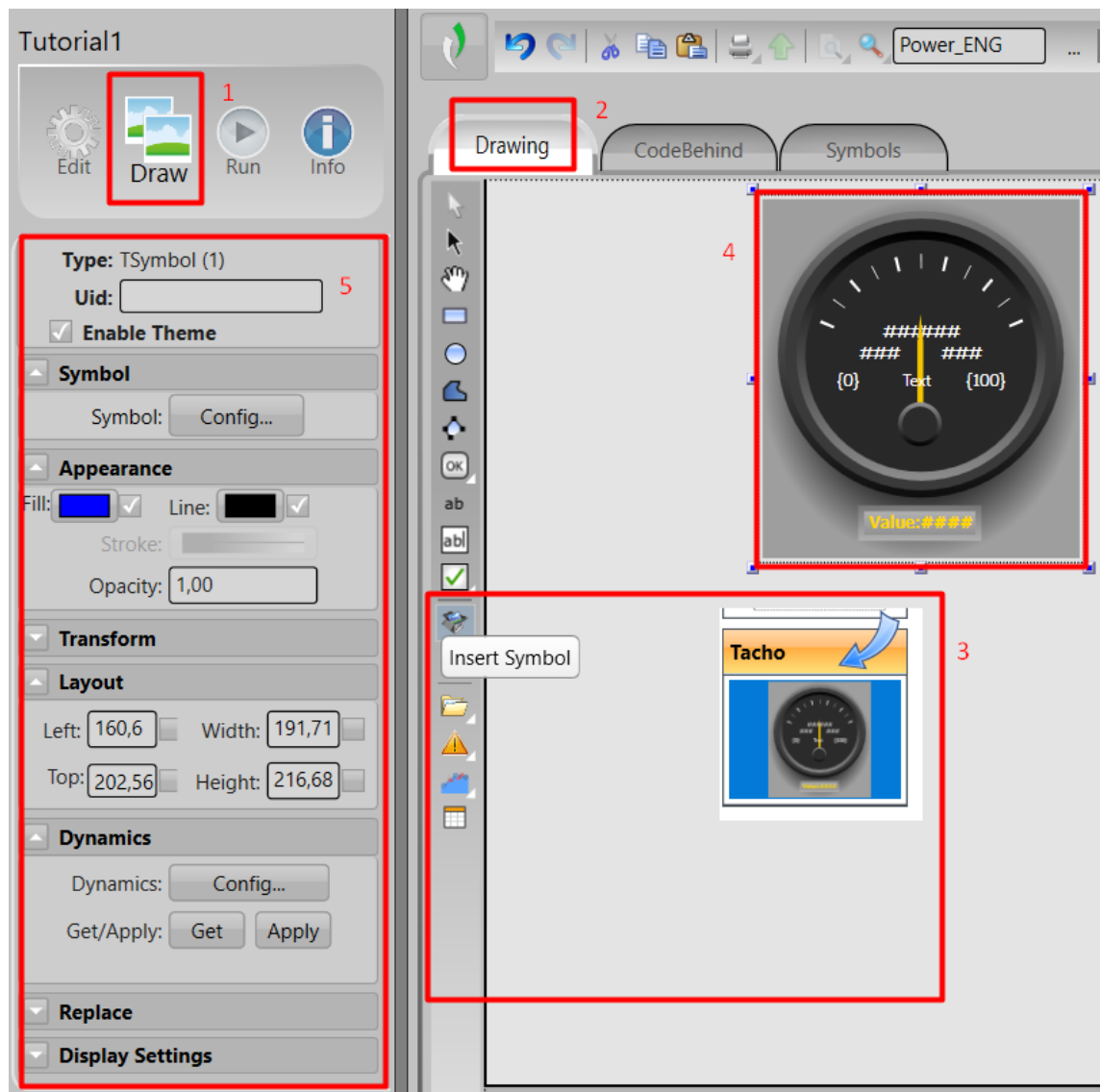


Figura 2-17. Inclusão de um Símbolo do Tipo Tacômetro

Para associar uma Tag ao símbolo, as seguintes etapas podem ser consideradas:

1. Clicar com o botão direito do mouse no símbolo;
2. Selecionar a opção *Symbol links*.

A figura a seguir mostra essas etapas de associação de Tag.

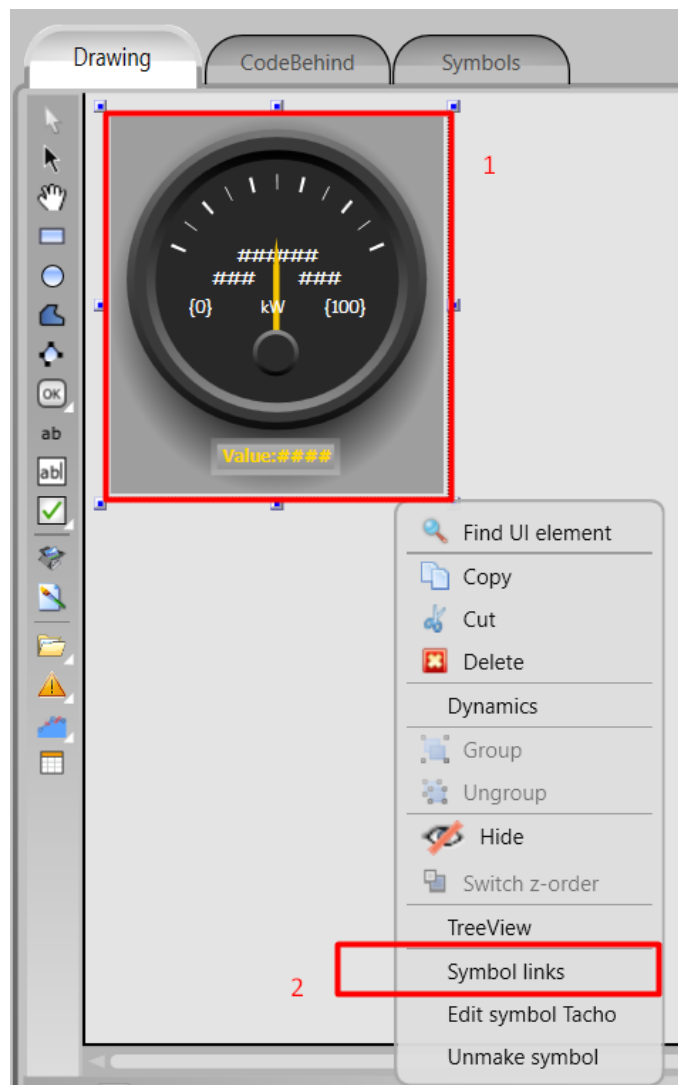


Figura 2-18. Associação de Tag a um Símbolo

Na janela seguinte é possível configurar a Tag e seus parâmetros conforme descrito nos passos a seguir:

1. Configurar as propriedades do símbolo (rótulo e valores máximo/mínimo) via duplo clique no símbolo;
2. Associar uma Tag ao símbolo no campo *RotateValue*;

Neste caso foi utilizado a Tag cliente *SimulationAnalog* para simular o movimento do ponteiro do medidor. A figura a seguir mostra essa configuração.

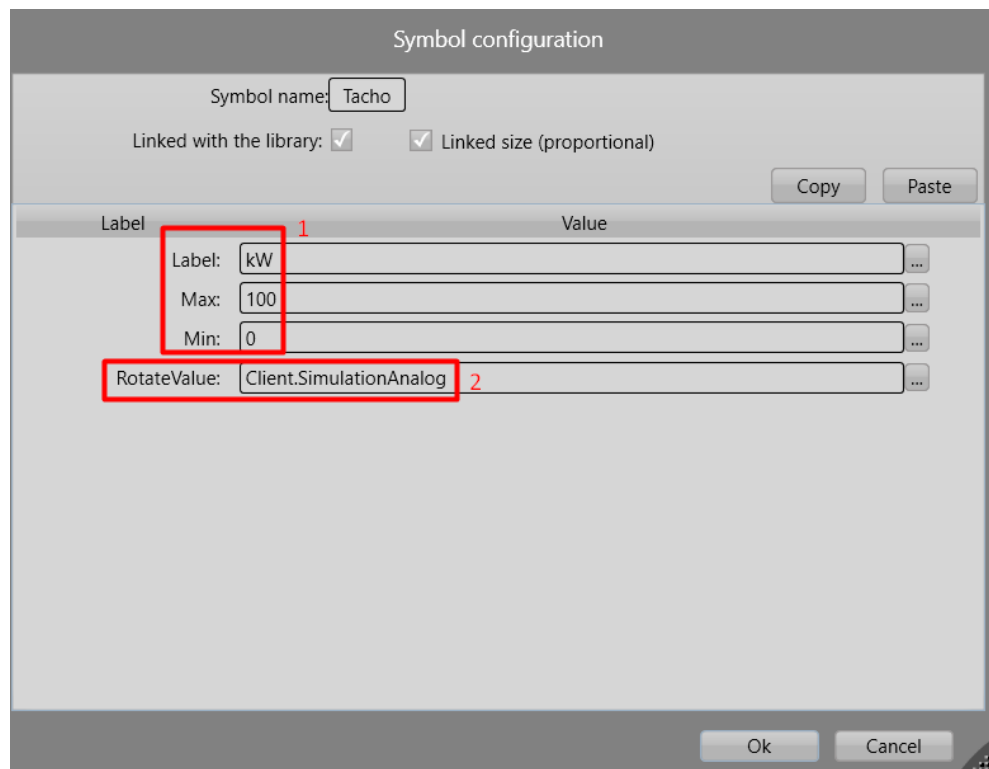


Figura 2-19. Configuração da Tag Associado ao Símbolo

Inclusão de Texto

No menu *Draw* para incluir um texto na tela podemos proceder conforme segue:

1. Selecionar a opção de *Text Output*;
2. Posicionar o texto no local desejado;
3. Configurar as propriedades do texto.

A figura a seguir ilustra esta sequência.

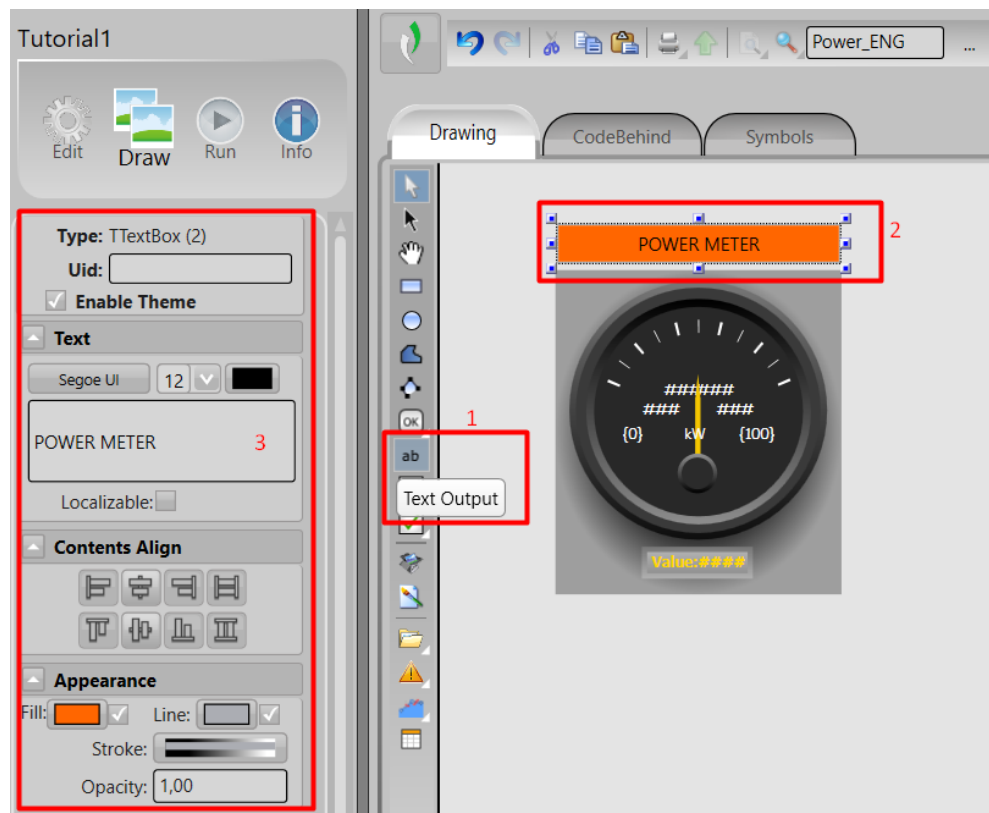


Figura 2-20. Inclusão de um Texto



Executando um Projeto

O ambiente de execução (*Run*) fornece acesso a todos os recursos de execução do projeto.

Os seguintes módulos estão disponíveis e são explorados nesta seção: *Build* (Compilar), *Test* (Testar), *Startup* (Iniciar) e *Publish* (Publicar).

Além disso, nesse menu estão incluídas as seguintes ferramentas: *UseCount* (Contagem de uso), *Dictionaries* (Dicionários) e *Extensions* (Extensões), as quais serão exploradas no decorrer desse manual.

Compilação do Projeto

A compilação verifica o projeto em relação a eventuais erros e otimiza o sistema para operação rápida e eficiente. O tempo necessário para compilar um projeto depende de seu tamanho e a capacidade de processamento do computador. Para compilar o projeto os seguintes passos servem como orientação:

1. Selecionar o menu *Run*;
2. Selecionar a opção *Build*;
3. Acessar a aba *Messages*;
4. Compilar o projeto;
5. Confirmar a compilação.

A figura a seguir ilustra o processo de compilação do projeto.

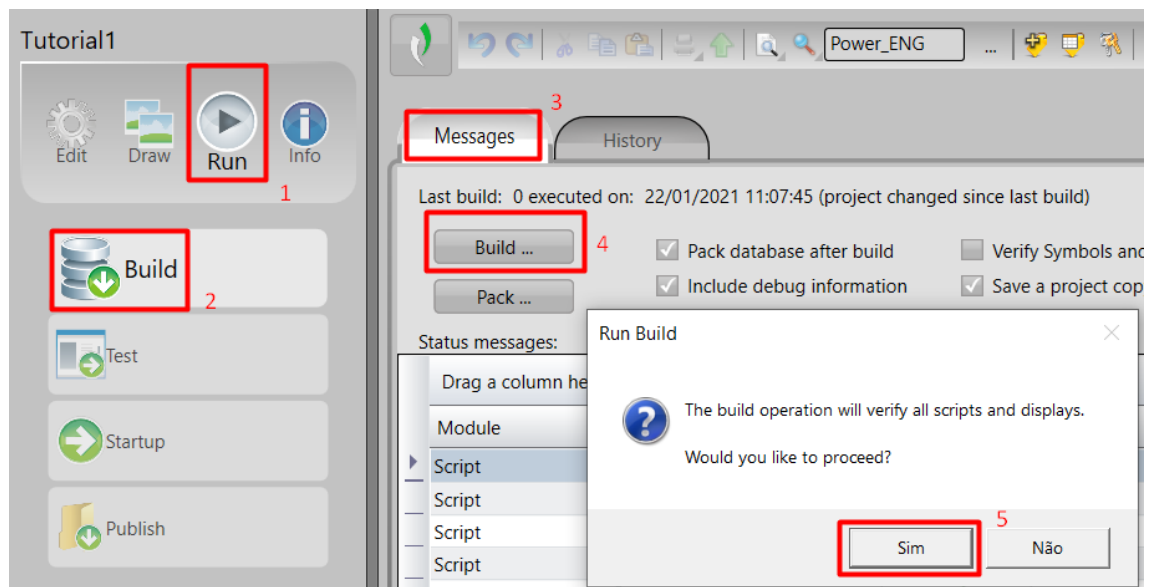


Figura 2-21. Compilação do Projeto

A figura a seguir mostra a janela de progresso da compilação.

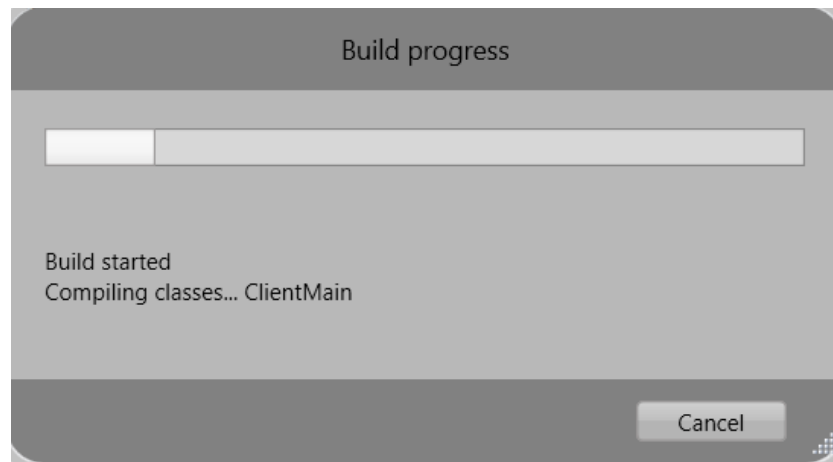


Figura 2-22. Progresso da Compilação

A figura a seguir ilustra o resultado da compilação e os erros, se existirem.

Status messages:

Drag a column header here to group by that column.

Module	Object ID	BuildStatus	BuildErrors	Location
Script	Class.ServerMain	✓	none	✓ ID
Script	Class.ClientMain	✓	none	✓ Module
Display	MainPage	✓	none	✓ Object ID
Display	SelectPage	✓	none	✓ BuildStatus
Display	LogOn	✓	none	✓ BuildErrors
				✓ Location

Figura 2-23. Resultado da Compilação

Teste do Projeto

A figura a seguir mostra os passos descritos para testar o projeto.

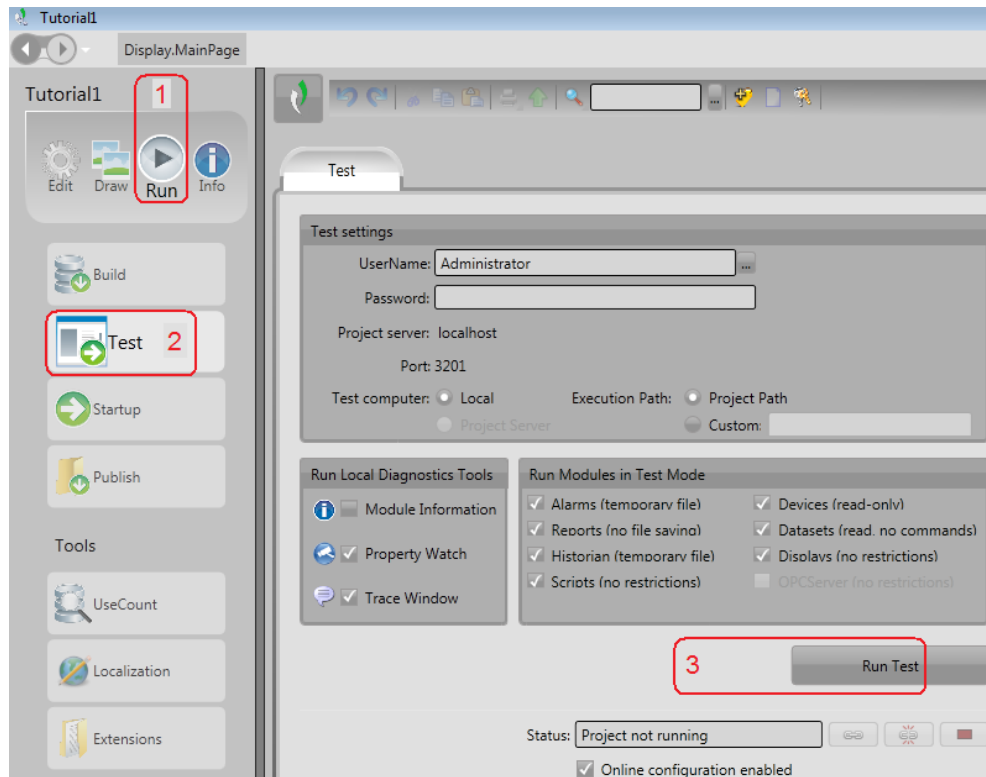


Figura 2-24. Teste do Projeto

1. Selecionar o menu *Run*;
2. Selecionar a opção *Test*, que executa o projeto em modo teste. É possível configurar o teste, as ferramentas de diagnóstico e os módulos a serem testados;
3. Iniciar o teste com as configurações definidas, confirmando através do botão *Run Test*;

A figura a seguir mostra o resultado da execução do teste do projeto.

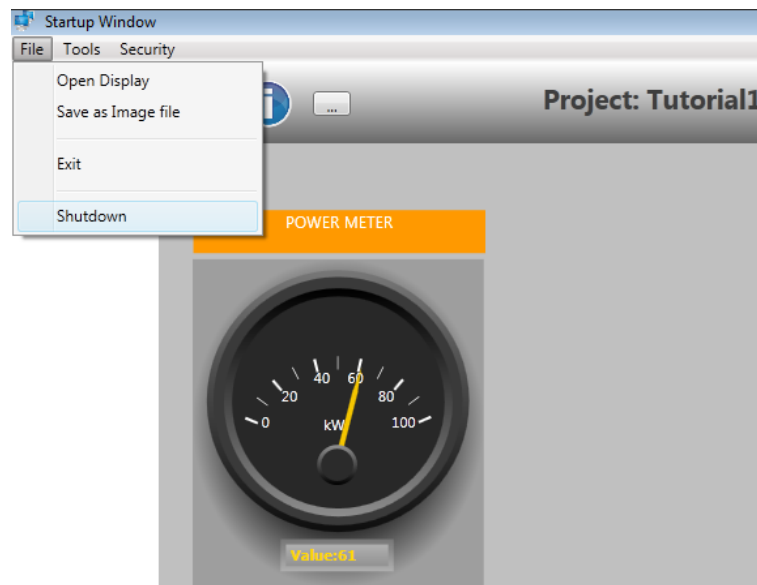


Figura 2-25. Execução do Teste do Projeto

Para finalizar o teste do projeto o usuário pode abrir o menu *File* e selecionar a opção *Shutdown* conforme representado na figura anterior.

Inicialização do Projeto

A figura a seguir mostra a sequência para inicialização do projeto.

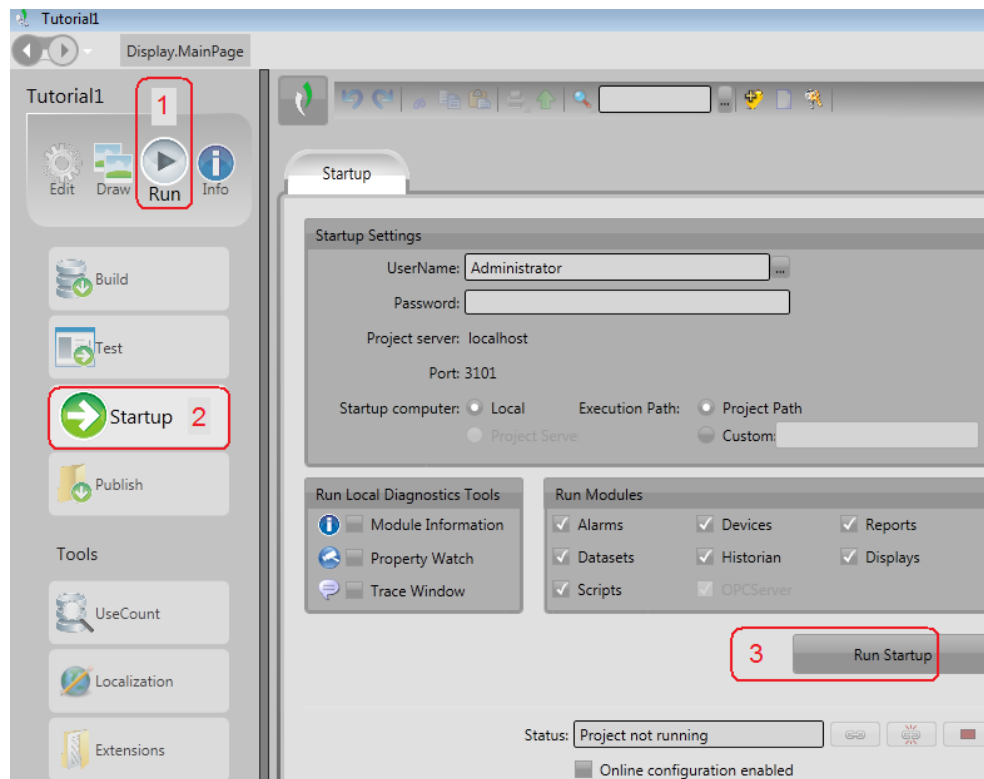


Figura 2-26. Execução da Inicialização do Projeto

1. Selecionar o menu *Run*;

2. Selecionar a opção *Startup*, que permite configurar a inicialização, as ferramentas de diagnósticos e os módulos a serem inicializados;
3. Iniciar a execução do projeto com as configurações definidas, através do botão *Run Startup*;

Juntamente com a visualização da tela principal do projeto será aberta uma janela mostrando o status e as mensagens de inicialização conforme pode ser observado na figura a seguir.

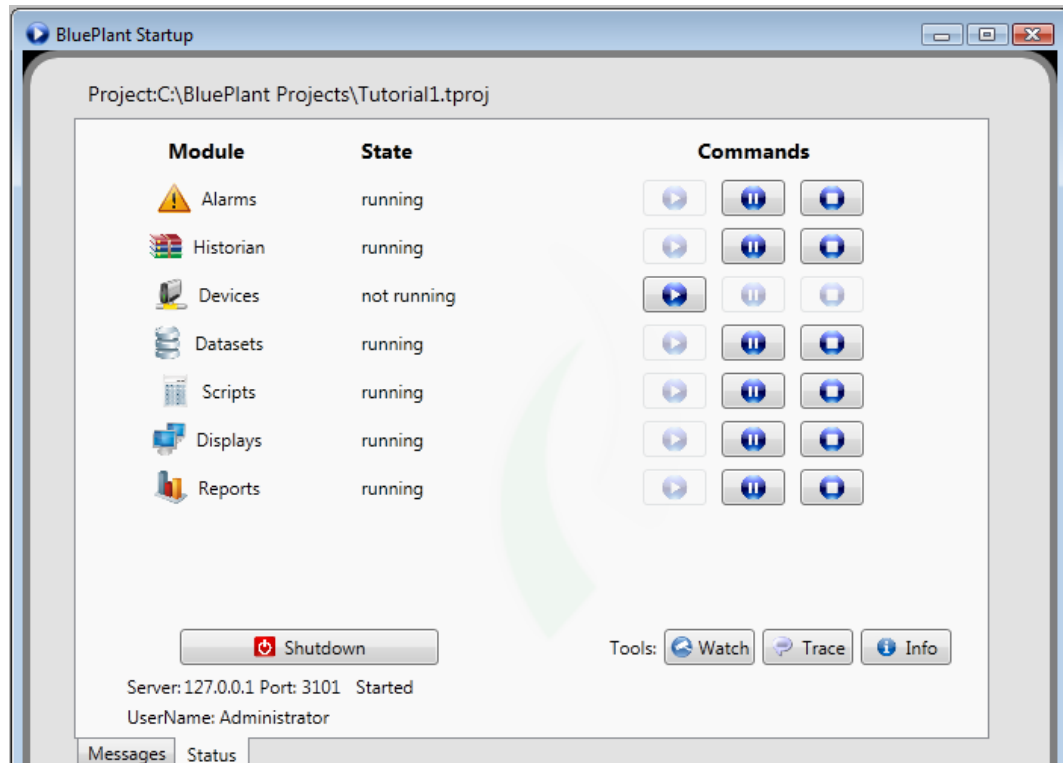


Figura 2-27. Status da Inicialização do Projeto

Publicação do Projeto

Quando o projeto está pronto para ser executado no campo, deve ser utilizado o recurso de publicação para configurar as opções de redundância (se aplicável) e para criar uma cópia do projeto na modalidade somente leitura com versão controlada, ou seja, uma cópia do projeto para a execução em campo. A extensão do arquivo gerado por ocasião da publicação é ".teng". A figura a seguir mostra a sequência para publicação do projeto.

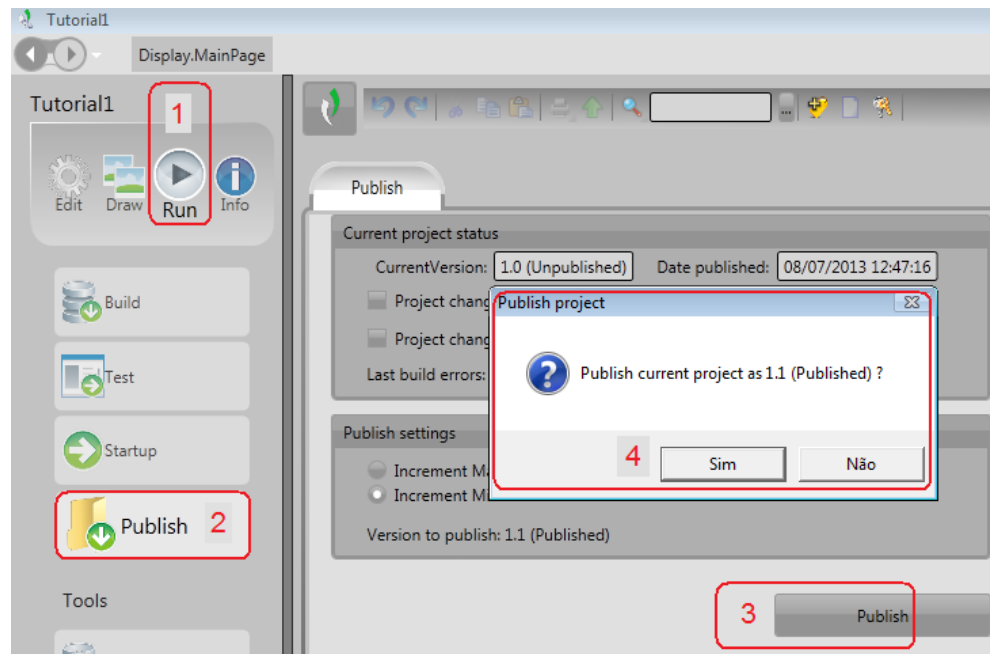


Figura 2-28. Publicação do Projeto

1. Selecionar menu *Run*;
2. Acessar a opção *Publish*, que permite acessar as configurações da publicação do projeto;
3. Clicar no botão *Publish*, que dispara a publicação do projeto com as configurações definidas;
4. Janela de confirmação para publicação do projeto.

Finalizada a publicação, uma janela similar à mostrada na Figura 2-29 será exibida indicando o caminho de publicação do projeto.

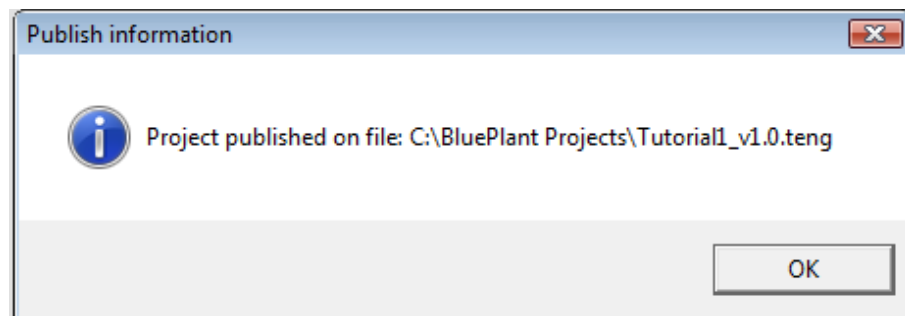


Figura 2-29. Caminho da Publicação do Projeto

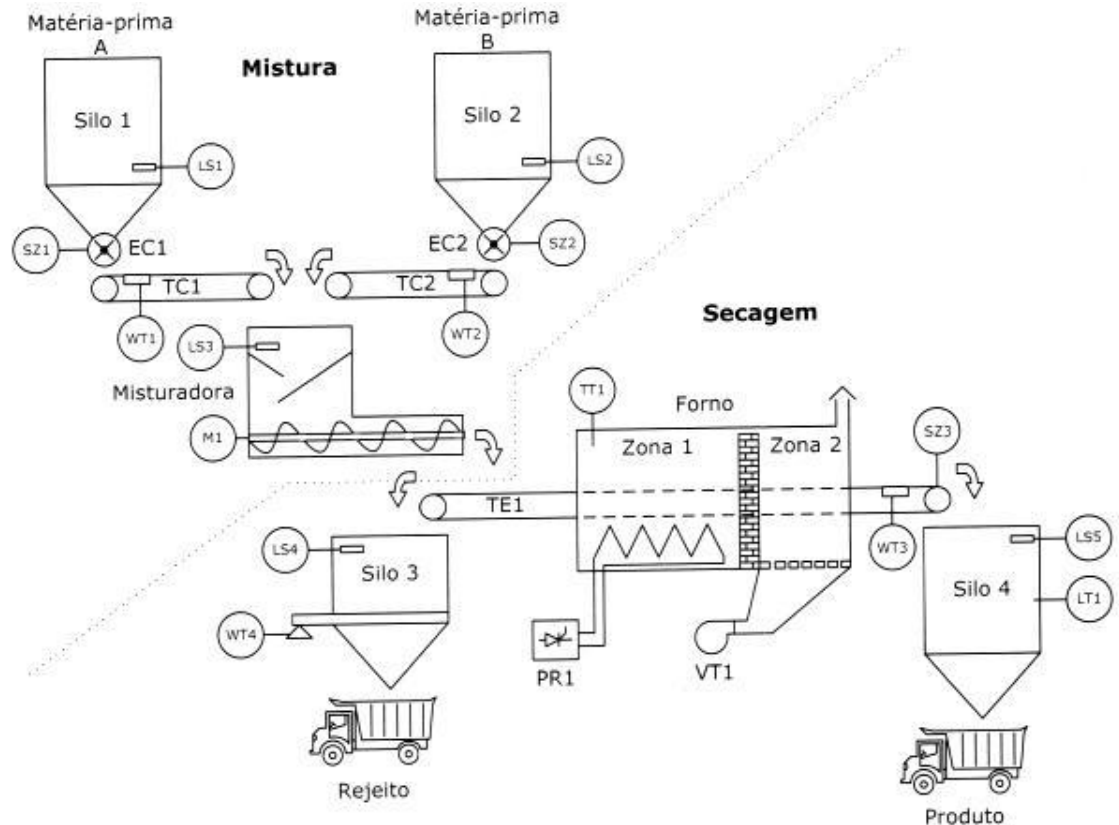
Estudo Dirigido 2-3: projeto de um sistema de supervisão



Processo de mistura e secagem de material

Leia com atenção a aplicação proposta e reflita sobre ela com o instrutor e colegas... Na sequência, crie um projeto no ambiente BluePlant configurando-o para atender a aplicação ora considerada.

O processo a ser automatizado consiste em duas unidades: Misturador e Secador. A figura a seguir mostra o diagrama de processo e instrumentação do sistema. Consulte o instrutor para orientações quanto à funcionalidade do mesmo.



Especificações do processo

Item	Especificação
Misturador	As eclusas fazem a retirada das matérias-primas dos silos numa taxa determinada pelos inversores, depositando-as nas correias transportadoras. Balanças dinâmicas pesam o material nas correias. As correias levam o material para mistura e homogeneização no misturador. Um eixo helicoidal transporta o material composto até a esteira de saída.
Secador	O secador é formado por uma esteira transportadora que conduz o material composto através do forno. Este último é composto de duas zonas (uma para aquecimento e outra para resfriamento e secagem). Uma balança totaliza a produção. O material resultante do processo é depositado no silo apropriado. A esteira transportadora tem é comandada por um acionamento reversível que permite também o ajuste de velocidade de transporte do material. Um outro silo com abertura manual recebe o rejeito da produção, o qual pode ser pesado por uma balança.



DICA: consulte o Instrutor para orientações adicionais

3. Editando a Aplicação



Os próximos capítulos detalham as ferramentas de configuração, *Runtime* e aplicativos do BluePlant. A seguinte estruturação foi considerada:

- Ferramentas de configuração: menus *Edit*, *Draw*, *Run* e *Info*;
- Ferramentas de *Runtime*: ferramentas da aplicação, objetos *Runtime* e linhas de comando;
- Ferramentas de aplicativos: telas em outros sistemas operacionais e serviço no Windows.

Neste capítulo vamos explorar o ambiente de edição do BluePlant, o qual fornece acesso a todas as funcionalidades necessárias para configurar o projeto.



Menu Edit

O ambiente de edição fornece acesso a todas as funcionalidades necessárias para configurar o projeto e é composto pelos itens listados na tabela a seguir.

Funcionalidade	Representação Gráfica
Edição de Tags	 Tags
Definições de Segurança	 Security
Configuração de Dispositivos	 Devices
Planejamento dos Alarmes	 Alarms
Banco de Dados	 Datasets
Linguagem Script	 Scripts
Projeto das Telas	 Displays
Configuração de Relatórios	 Reports

Tabela 3-1. Funcionalidades do Menu de Edit

Editando Tags



O menu *Edit*, opção *Tags* configura o banco de dados de Tags em tempo real.

Edição de Objetos Tag (*Objects*)

Use as Tags (e suas propriedades pré-definidas) incluídas na lista a seguir para configurar um banco de dados em tempo real. Tipos disponíveis podem ser estendidos e novos tipos criados via tabela *EditTagsUserTypes*. *Tag* no contexto de uma configuração de projeto refere-se a uma variável de processo. A Figura 3-1 a seguir ilustra a edição de objetos Tags.

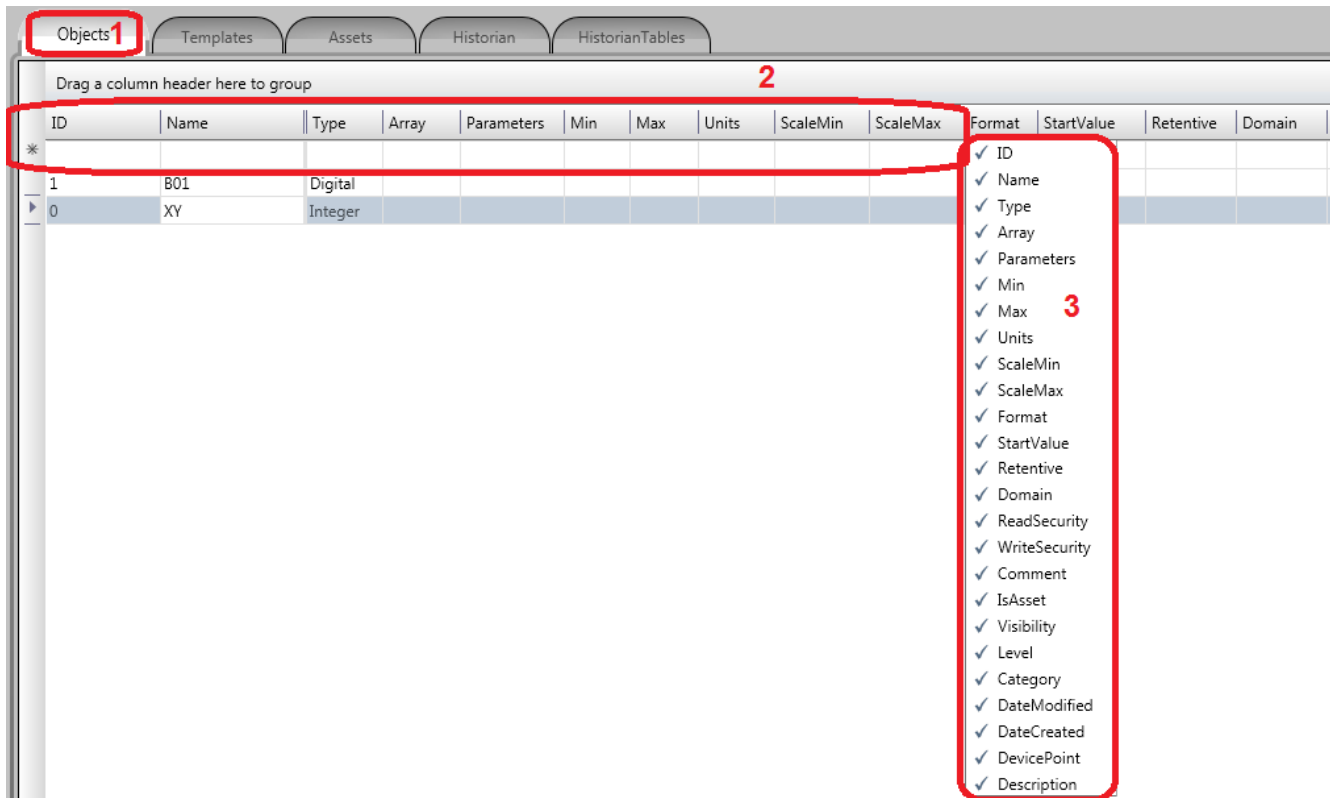


Figura 3-1. Edição de objetos da Tag

Os itens que compõem o menu de edição de objetos estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a guia *Objects*;
2. Digitar os dados relativos à nova Tag da aplicação na linha marcada com um asterisco (*);
3. Selecionar as opções de configuração das Tags. Clicar com o botão direito do mouse no cabeçalho de uma coluna e marcar os itens para exibição.

Na sequência são explicados cada um dos itens do menu de edição de objeto conforme ilustrado na figura anterior.

Name

Define o nome da Tag. Nesta coluna é possível criar ou editar Tags. O sistema avisará se o nome não é válido mostrando o contorno do campo em cor vermelha. Este campo aceita até 63

caracteres alfanuméricos, sendo aceita acentuação de letras e os seguintes caracteres especiais:
ç _

Type

Estabelece o tipo da Tag. Os tipos disponíveis são:

Tipo da Tag no BluePlant	Tipo em .NET Framework	Descrição
Digital	Boolean	0 ou 1
Integer	int32	Inteiro com sinal
Long	int64	Inteiro com sinal
Double	double	Ponto Flutuante
Decimal	decimal	Decimal
Text	string char[]	Texto
DateTime	DateTimeOffset	Data e hora
TimeSpan	TimeSpan	Time
Image	byte[]	Binary Image
Timer	int32	Temporizador
Guid	Guid	UniquelIdentifier
DataTable	System.Data.DataTable	System.Data.DataTable
Reference	-	Objeto referência apontando para outro objeto
UserTypes		Tipos definidos pelo usuário

Tabela 3-2. Tipos de Disponíveis para Tags



Pesquisa na Documentação: tipos disponíveis para Tags

Consulte, na documentação do produto, os intervalos de valores e as particularidades dos diversos tipos de Tags.

Fonte: Manual de Utilização Série BluePlant – MU224000!

Array

Quando não definido (em branco) a Tag não é do tipo matriz. Ao definir um Array com o valor N uma matriz de 0 até N posições é criada. Por exemplo, ao criar uma Tag Array de tamanho "5", a matriz é criada a partir da Tag[0] à Tag[5], significando que 6 elementos são criados. Esse método contempla dois estilos de programação: a indexação a partir do zero e a contagem a partir do 1. Em relação à extensão de Array por níveis de contas, as versões Lite e Express estão limitadas a matrizes unidimensionais. A versão Enterprise permite criar matrizes com até três dimensões.

Exemplo 1:

Campo Name: TagAnalog

Campo Array: 2

Cria 3 elementos: TagAnalog[0] TagAnalog[1] TagAnalog[2]. Sintaxe de acesso:

C#: TagAnalog[1]

VB.NET: TagAnalog(1)

Exemplo 2:

Campo Name: Temp

Campo Array: 1,2

Cria 6 elementos: Temp[0][0] Temp[0][1] Temp[0][2] Temp[1][0] Temp[1][1] Temp[1][2]. Sintaxe de acesso:

C#: Temp[1,2]

VB.NET: Temp(1,2)

Parameters

Define parâmetros da Tag de acordo com o tipo: *banda morta* para Tags *analógicas* e *ajustes* para Tags do tipo *contador* e *temporizador*. Quando criando *Tag de referência* (ou PONTEIROS) define a Tag de referência de destino na coluna *Tipo de referência*.

Min

Configura o valor mínimo admissível para a Tag. Esse valor somente é aplicado, caso seja habilitado essa consistência em *Info/Project/Setting/Limit values to Min-Max*

Max

Configura o valor máximo admissível para a Tag. Esse valor somente é aplicado, caso seja habilitado essa consistência em *Info/Project/Setting/Limit values to Min-Max*

Units

Define a unidade de engenharia da grandeza de uma medida que você deseja usar para essa Tag.

ScaleMin

Informa o valor de escala mínimo configurado no mapeamento desta Tag em um driver de comunicação (*Scaling*).

ScaleMax

Informa o valor de escala máximo configurado no mapeamento desta Tag em um driver de comunicação (*Scaling*).

Format

Especifica o formato de exibição de um valor. Exemplo 1: N1 (número com uma casa decimal). Exemplo 2: d (data abreviada).

Exemplo de formatos numéricos:

Especificação	Descrição
N0	Número sem casas decimais
N3	Número com 3 casas decimais
X	Hexadecimal (suportado apenas em tipos inteiros)
C	Moeda

Tabela 3-3. Exemplo de Formatos Numéricos

Exemplo de formatos DateTime:

Especificação	Descrição
T (somente)	Padrão de tempo no formato longo (equivalente a "HH:mm:ss")
d (somente)	Padrão de data abreviada (equivalente a "M/d/yyyy")(mês/dia/ano) (EUA)
dd	Representa o dia do mês como um número de 01 a 31
ddd	Representa o nome abreviado do dia da semana
dddd	Representa o nome completo do dia da semana
MM	Representa o mês como um número de 01 a 12
MMM	Representa o nome abreviado do mês
yy	Representa o ano como um número de dois dígitos
yyyy	Representa o ano como um número de quatro dígitos
hh	Representa a hora como um número de 01 a 12

HH	Representa a hora como um número de 00 a 23
mm	Representa o minuto como um número de 00 a 59
ss	Representa os segundos como um número de 00 a 59
fff	Representa os milissegundos como um número de 000 a 999

Tabela 3-4. Exemplo de Formatos DateTime

Exemplo 3 (padrão de tempo no formato longo): Formato = HH:mm:ss (hora:minuto:segundo).

Start Value

Estabelece o valor de inicialização da Tag. Quando deixado em branco nenhum valor de inicialização é aplicado.

Retentive

Especifica se as Propriedades e o Valor da Tag são retidos quando a aplicação é encerrada, sendo usados como valores de inicialização na próxima execução. As seguintes opções de configuração estão disponíveis:

- *None*: as propriedades e o valor da Tag não são salvos (não retentivo);
- *ValueOnly*: somente o valor da Tag é salvo quando modificado e pode ser usado como valor de inicialização na próxima execução;
- *Properties*: todas as propriedades da Tag são salvas, incluindo o valor, quando este é modificado para serem usadas na próxima inicialização;
- *PropertiesOnly*: as propriedades da Tag são salvas, mas o valor não.

Ao aplicar propriedades retentivas, as modificações são salvas no banco de dados <project>.retentiv. O uso de configurações retentivas para Tags cujos valores mudam rapidamente não é recomendado uma vez que isso pode piorar o desempenho da execução. Esta opção deve ser usada no caso de variáveis do processo que são consideradas críticas e/ou estão associadas a condições de segurança. Durante a edição é necessário pressionar a tecla ENTER na célula da tabela para a confirmação das configurações efetuadas.

Domain

Define o valor da Tag para o projeto inteiro ou um valor específico para cada tela cliente. As seguintes opções de configuração estão disponíveis:

- *Server*: o valor da Tag é consistido em todo o projeto e em todos os clientes;
- *Client*: o valor da Tag é local para cada computador remoto executando uma visualização do cliente (Web ou visor).

ATENÇÃO:

Tags do tipo cliente não devem ser usadas em módulos do servidor como Device, Alarm, Historian e ServerScripts, uma vez que seus valores são locais (restritos ao computador em execução) e não são propagados para clientes remotos.

A maioria das Tags em um projeto é definida como *Servidor*. Tags *Locais*, por sua vez, permitem valores diferentes em cada computador cliente. No entanto, pode-se usar Tags *Locais* em dados temporários específicos para computadores-cliente individuais. A situação mais comum para o uso de Tags *Locais* é quando dados temporários são necessários para gerenciar a Interface do usuário na tela.

ReadSecurity

Define as permissões de leitura da Tag.

WriteSecurity

Define as permissões de escrita da Tag.

Comment

Define opcionalmente qualquer comentário sobre esta Tag.

Visibility

Define a visibilidade do valor da Tag no servidor OPC para projetos remotos. As seguintes opções de configuração estão disponíveis:

- *Private*: define uma Tag visível somente no projeto local e par redundante;
- *Protected*: define uma Tag do tipo somente leitura visível no servidor OPC DA para projetos remotos e clientes OPC DA;
- *Public*: define uma Tag visível no servidor OPC DA para projetos remotos e clientes OPC DA.

Level

Define o nível que se deseja associar à Tag na aba Níveis (Levels ou Assets).

Category

Define uma ou mais categorias para este atributo.

DevicePoint



Informa o mapeamento desta Tag em um driver de comunicação.

Description

Define opcionalmente qualquer descrição sobre esta Tag.

Edição de Tipos Definidos pelo Usuário (Templates)

Uma Tag de tipo existente pode ser localizada no campo *User Custom Type*.

Clique  para apagar uma Tag de tipo estrutura existente e  para criar uma nova Tag de tipo estrutura.

Uma Tag de tipo do usuário pode ser usada da mesma forma que as Tags incorporadas.

Nas versões *Express* e *Lite*, as Tags de tipo do usuário podem ser usados somente na lista principal de Tags. A versão *Enterprise* permite a criação de um tipo do usuário com até quatro níveis.

Exemplo: ao criar um tipo do usuário PID com os seguintes membros, *setpoint* e *PV*, pode-se criar uma Tag denominada *loop* do tipo PID, sendo que a sintaxe para acesso ao seu valor é:
loop.setpoint e loop.PV

A figura a seguir ilustra a edição de tipos definidos pelo usuário.

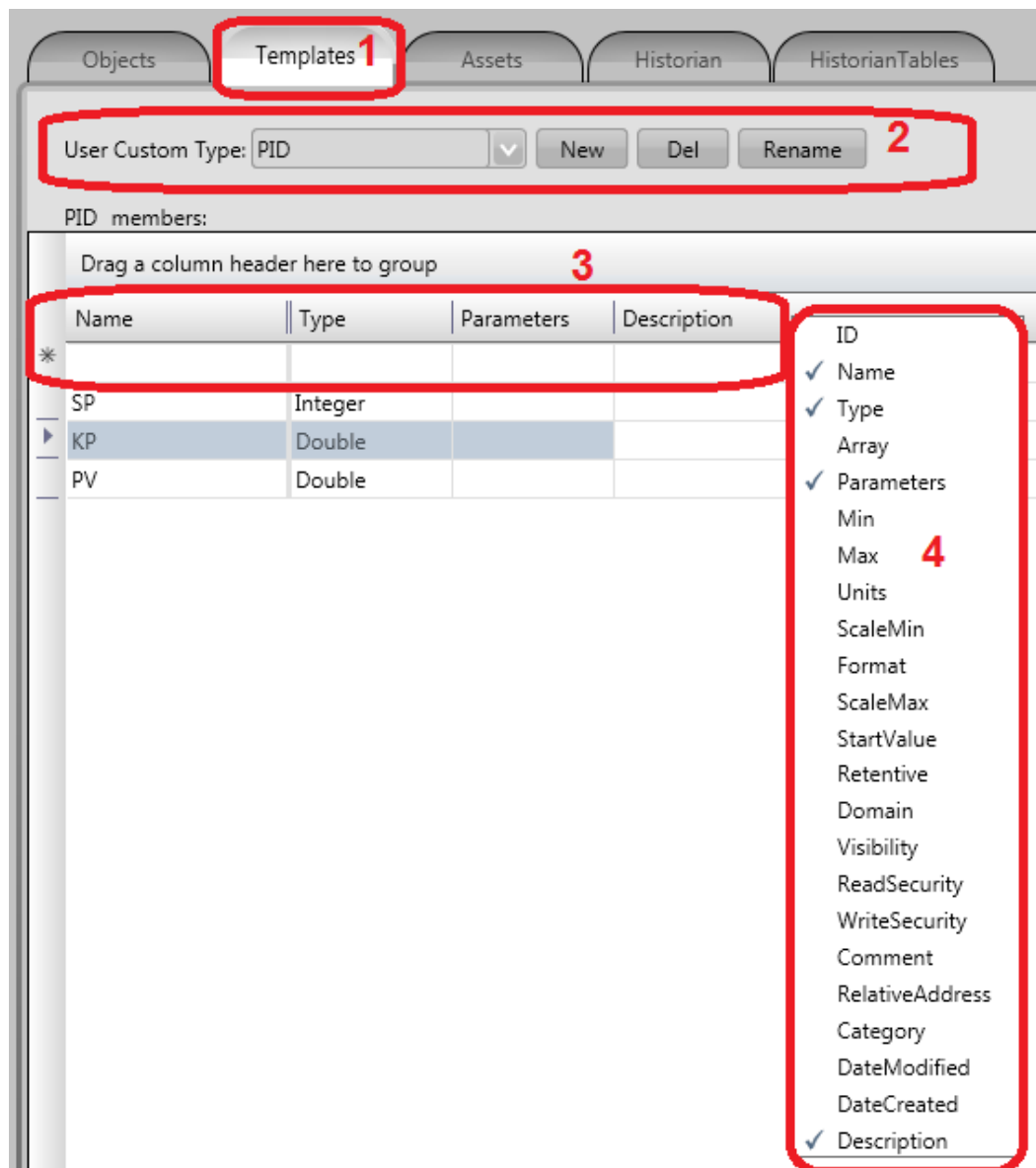


Figura 3-2. Edição de Templates Tag

Os itens que compõem o menu de edição de tipos estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a opção *Templates*;
2. Selecionar a opção *User Custom Type* clicando com o botão esquerdo do mouse em *New* para criar um novo elemento e em *Del* para excluir o elemento;
3. Digitar os dados relativos ao novo tipo na linha marcada com um asterisco (*);
4. Selecionar as opções desejadas de configuração da coluna *Type* da tabela clicando com o botão direito do mouse no cabeçalho de uma coluna e marcando os itens para exibição.

Os campos relativos à adição de Tags dentro da aba *Templates*, marcados com os números 3 e 4 na figura anterior, possuem as mesmas características dos campos da aba *Objects*.

Edição de Níveis (Assets)

Se você tiver a versão Enterprise do BluePlant os Níveis ou (Assets) permitem configurar metadados adicionais para o seu projeto. Por exemplo, você pode organizar os objetos em seu projeto, tais como Tags, Devices e Alarmes, em uma hierarquia. Isto permite-lhe agrupar Tags que estão relacionadas entre si. Por exemplo, nas concessionárias de energia, usa-se, geralmente, para se identificar um “bay” a subdivisão do sistema elétrico em Regionais, Subestações e Setores de Tensão. Os objetos níveis existem justamente para esta definição hierárquica. A hierarquia pode refletir elementos como, UHEs, Subestações e Setores de Tensão. Define-se na aba Níveis (Assets) do espaço de trabalho das Tags. Para criar Níveis (Assets):

1. Vá para *Edit > Tags > Assets*;
2. Botão direito do mouse no nome do projeto e selecione *New Level*;
3. Digite um nome para o nível;
4. Clique com o Botão direito do mouse o novo nível e selecione *Insert Asset*: a janela Selecionar objeto é exibida, com todos os objetos, por tipo, no lado esquerdo;
5. Selecione o tipo de objeto a partir do lado esquerdo e do objeto que você deseja a partir do lado direito;
6. Clique em *OK*: o objeto torna-se um filho do nível selecionado.
7. Continue adicionando níveis filho ou irmão e inserção de ativos, conforme a necessidade. Se necessário, clique com o botão direito um nível para renomear ou apagar, ou clique com o botão direito em um Nível para excluí-lo;
8. Na aba Objetos, para Tags novas ou já existentes, selecione o novo nível na coluna *Level*.



Figura 3-3. Edição de Nível (Assets)

Edição de Historiador (Historian)

É possível configurar um banco de dados de Historiador para registrar alterações nas Tags. Para selecionar uma Tabela de Historiador existente pesquise no campo *Historian Tables*.

Clique para apagar uma Tabela de Historiador existente ou clique para configurar uma nova Tabela de Historiador. A figura a seguir mostra as configurações da Tabela do Historiador.

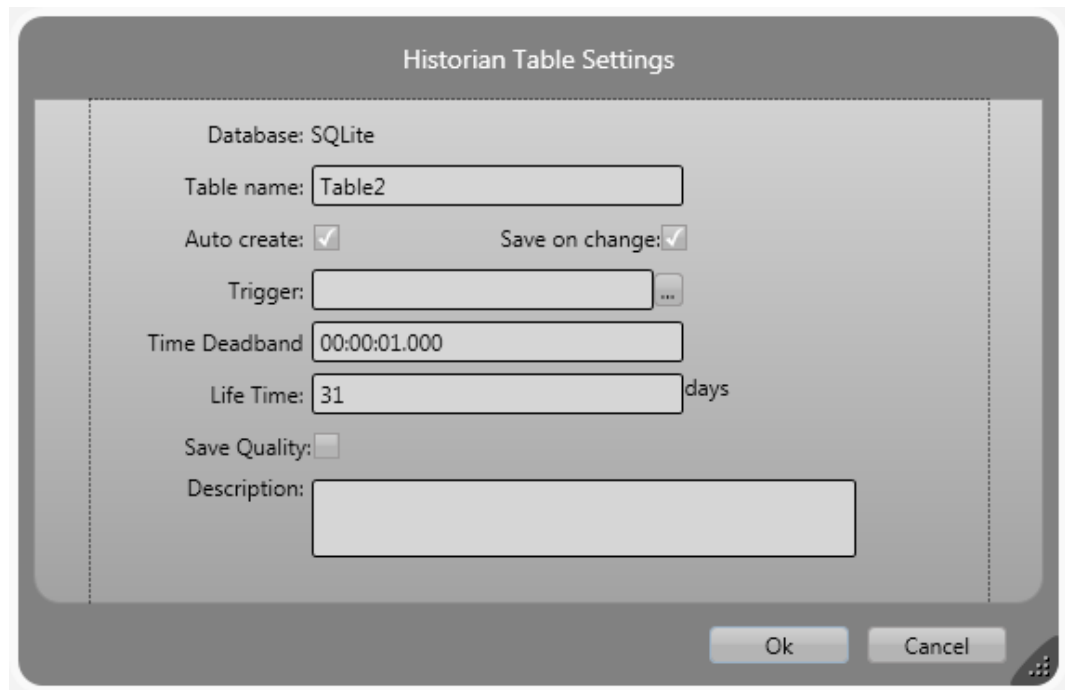


Figura 3-4. Configurações da Tabela do Historiador

O banco de dados onde as Tags são salvas é definido em EditDatasetDBs no DB chamado *TagHistorian*. A figura a seguir ilustra as configurações associadas ao historiador de Tags.



Figura 3-5. Edição de Historiador

Os itens que compõem a tabela de edição de historiador estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a guia *Historian* clicando com botão esquerdo do mouse;
2. Na opção *Historian Tables* clicar com o botão esquerdo do mouse em *New* para criar uma nova tabela e em *Del* para excluir uma tabela. O botão *Config* permite acesso às configurações da tabela do historiador;
3. Digitar os dados relativos à Tag cujas alterações desejam-se registrar na linha marcada com um asterisco (*).

Na sequência são explicados cada um dos itens do menu de edição de historiador conforme ilustrado na figura anterior.

TagName

Indica o nome da Tag. Nesta coluna é possível inserir ou remover Tags do historiador.

DeadBand

Configura a banda morta do Historiador. O evento de salvamento no registro é disparado somente se a Tag tem seu valor alterado acima da banda morta especificada. Exemplos:

- Banda morta configurada 10. Valor atual 20. Todos os novos valores (10 ou mais) maiores do que o valor anterior são adicionados ao registro do historiador;
- Valor alterado para 35 (diferença em relação ao valor anterior que era 20:15). Resultado: valor adicionado ao registro do historiador;
- Valor alterado para 50 (diferença em relação ao valor anterior que era 35: 15). Resultado: valor adicionado ao registro do historiador;
- Valor alterado para 55 (diferença em relação ao valor anterior que era 50: 5). Resultado: valor não é adicionado ao registro do historiador.

Deviation

Se o *SaveOnChange* estiver definido nas configurações da tabela e a Tag sofrer uma alteração maior do que o parâmetro *Deviation* (valor anterior e atual > *Deviation*), o valor será salvo imediatamente (não aguarda o próximo *TimeSpan*).

RateOfChange

Se a Tag *RateOfChange* (unidades de engenharia por segundo) for maior que o parâmetro *RateOfChange* especificado e *SaveOnChange* estiver habilitado, a tabela será salva imediatamente (não aguarda o próximo *TimeSpan*).

DateModified

Registro da última edição desta Tag nesta tabela.


DateCreated

Registro da data da inserção desta Tag nesta tabela.

Historian Table

Define a tabela do banco de dados onde as Tags do Historiador são salvas.

Tabelas do Historiador (HistorianTables)

A aba *Historian Tables* apresenta a lista de tabelas do historiador criadas no projeto. Estas tabelas podem ser editadas clicando em  na aba *Historian*, conforme mostrado anteriormente, ou ainda nesta aba, clicando diretamente na linha da tabela que se quer editar. As opções de configuração são:

- *Table name*: nome da tabela do banco de dados;
- *Trigger*: a tabela é salva a cada alteração no disparo;
- *Time Deadband*: tempo que o sistema deve esperar depois de gerar um registro para uma Tag antes de gerar um novo registro para a mesma Tag. Use com a opção *Save on Change* para evitar a geração de registros em demasia na base de dados;
- *AutoCreate*: cria automaticamente uma tabela quando esta não é encontrada no banco de dados;

- *Save on change*: armazena dados na tabela cada vez que um das Tags associadas sofrer modificação;
- *LifeTime*: tempo em que devem ser retidos os dados históricos. Passado este tempo, as linhas mais antigas na tabela serão excluídas da base de dados. Para que os dados nunca sejam excluídos deixe este campo vazio ou digite um zero;
- *Save Quality*: Salva a qualidade da Tag no banco;
- *Description*: descrição definida pelo usuário para fins de documentação.

Editando Configurações de Segurança



O menu *Edit*, opção *Security* define as permissões do usuário e níveis de acesso ao projeto.

Edição de Usuários (Users)

Define os usuários do projeto e as permissões relacionadas. As IDs dos níveis Administrator, Guest e User apresentam atributos incorporados. Algumas permissões de usuário permanecem ativas independentemente da alteração de nome ou de permissões, conforme segue:

- Usuário *Administrator* (ID:2): a ID do Administrator permanece a mesma mesmo se o nome do usuário ou as permissões forem alteradas. O Administrator é o único usuário que pode apagar ou bloquear usuários e pode definir senhas para as interfaces do Banco de Dados.
- Usuário *Guest* (ID:0): esta ID é usada para usuários de login anônimo. Usuários guest não possuem senhas atribuídas. Entretanto, suas permissões podem ser alteradas. Quando o sistema é inicializado sem Usuário(s) definido (ou se é um login anônimo) se aplicam as permissões de segurança definidas para o usuário guest.

A figura a seguir mostra a edição dos perfis dos usuários no âmbito da segurança.

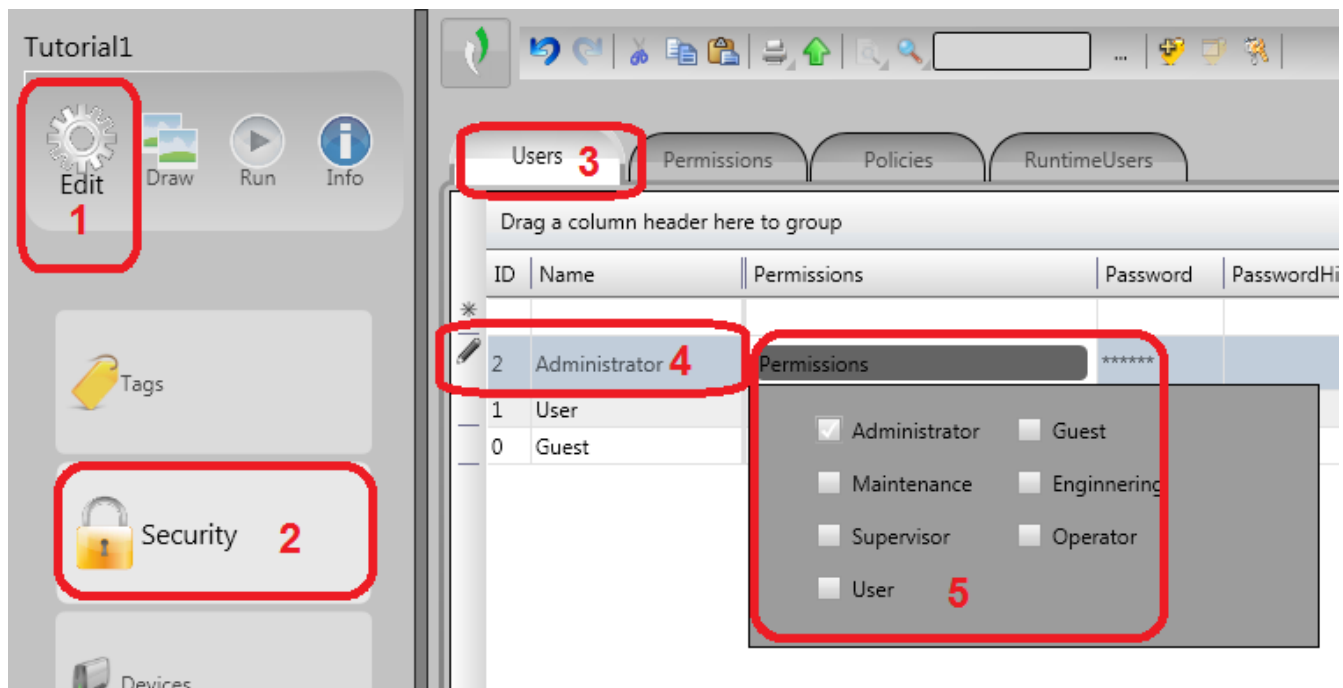


Figura 3-6. Edição de Perfis de Usuários

Os passos para edição/inclusão de perfis de usuários estão numerados na figura anterior e são descritos a seguir:

1. Selecionar o menu *Edit*;
2. Acessar o item *Security*;
3. Clicar na guia *Users* para acesso aos usuários do sistema;
4. Digitar os dados relativos aos usuários da aplicação na linha marcada com um asterisco (*);
5. Selecionar as permissões disponíveis que o usuário adicionado ou em edição pode ter no sistema.

ATENÇÃO:
É possível configurar uma senha e uma dica da senha para cada usuário.

Os campos de configuração desta aba são:

- *Name*: nome do usuário;
- *Permissions*: seleção do grupo de permissões para este usuário;
- *Password*: definição da senha para o usuário;
- *PasswordHint*: onde se insere a ajuda de senha;
- *Policy*: seleção da opção de política de uso para este usuário;
- *Blocked*: bloqueia o acesso do usuário;
- Deleted: bloqueia o acesso e marca o usuário como excluído, sem excluir o usuário;
- *Profile*: Endereço de e-mail, número de telefone e nome completo do usuário.

Edição de Permissões (Permissions)

Define a configuração e os privilégios de execução para cada grupo de permissão. A figura a seguir ilustra essa seleção.

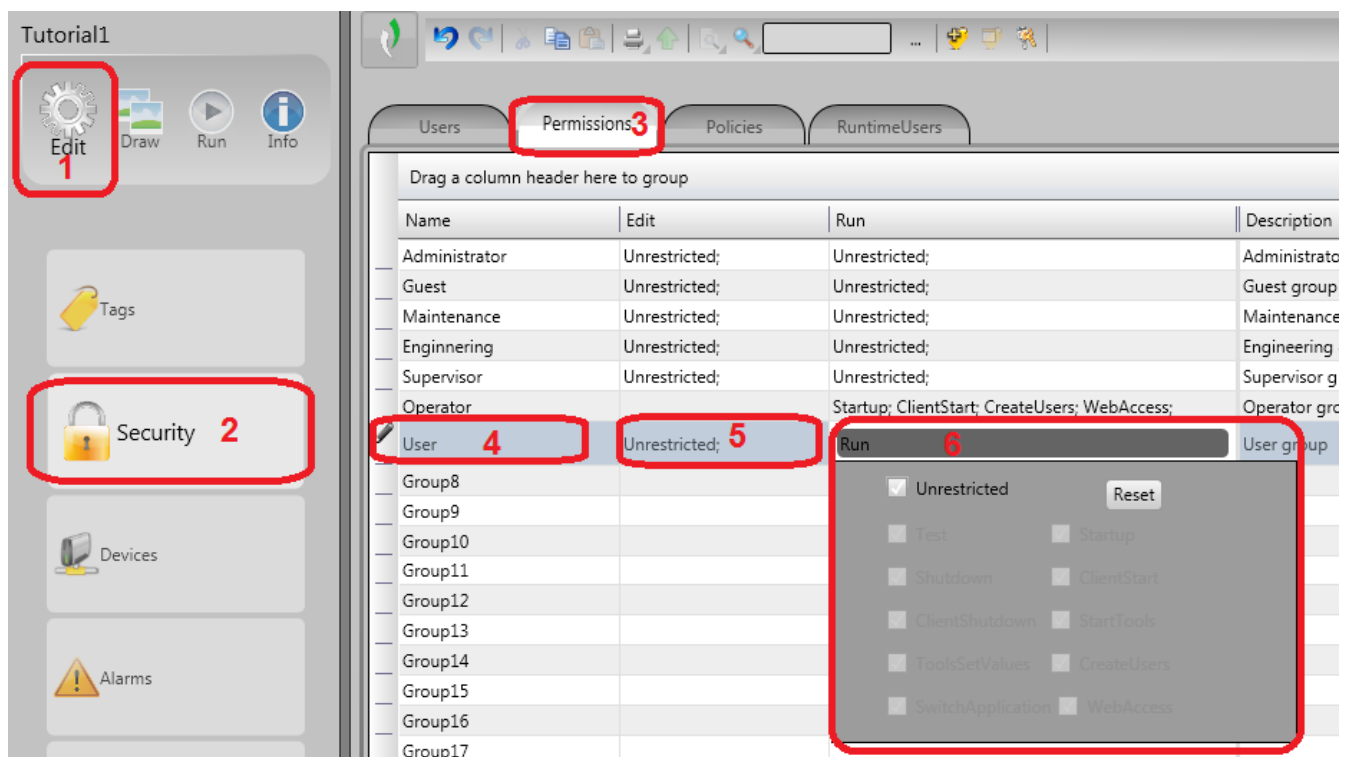


Figura 3-7. Edição de Permissões de Segurança

Os passos para editar as permissões existentes estão ilustrados na figura anterior e descritos a seguir:

1. Selecionar o menu *Edit*;
2. Acessar o item *Security*;
3. Clicar na guia *Permissions* para acesso às permissões do sistema;

4. Digitar os dados relativos às permissões dos usuários na linha marcada com um asterisco (*);
5. Configurar as permissões de edição na coluna *Edit*;
6. Configurar as permissões do *Runtime* na coluna *Run*.

ATENÇÃO:

A configuração das permissões de edição, serve para que os usuários configurados com esta permissão possam realizar determinados tipos de modificações no projeto. As configurações das permissões do *Runtime* servem para que os usuários configurados com esta permissão possam realizar determinadas ações durante a execução do projeto.

Os campos de configuração desta aba são:

Name: Estabelece o nome do grupo de permissão;

Edit: Define os privilégios de grupos de permissão para edição e desenho. As opções disponíveis são: Unrestricted (irrestrito), EditTags (Edição de Tags), Security (Segurança), Scripts (Scripts), Datasets (Banco de dados), Reports (Relatório), Publish (Publicação), Notes (Notas), Historian (Historiador), Alarms (Alarmes), Devices (Dispositivos), Displays (Telas), Startup (inicialização), Settings (Configurações) e CreateTags (Criação de novas Tags);

Run: Define os privilégios de Execução de Grupos de Permissão (Runtime). As opções disponíveis são: Unrestricted (irrestrito), Test (teste), Startup (inicialização), Shutdown (parada), ClientStart (inicialização do cliente), ClientShutdown (parada do cliente), StartTools (ferramentas de inicialização), ToolsSetValues (valores de ajuste das ferramentas de inicialização), CreateUsers (criação de usuários), SwitchApplication (alternar aplicação) e WebAccess (acesso via Web).

Edição de Políticas (Policies)

Define as regras de senha e outras políticas de segurança para o acesso e execução do projeto. A figura a seguir ilustra essa seleção.

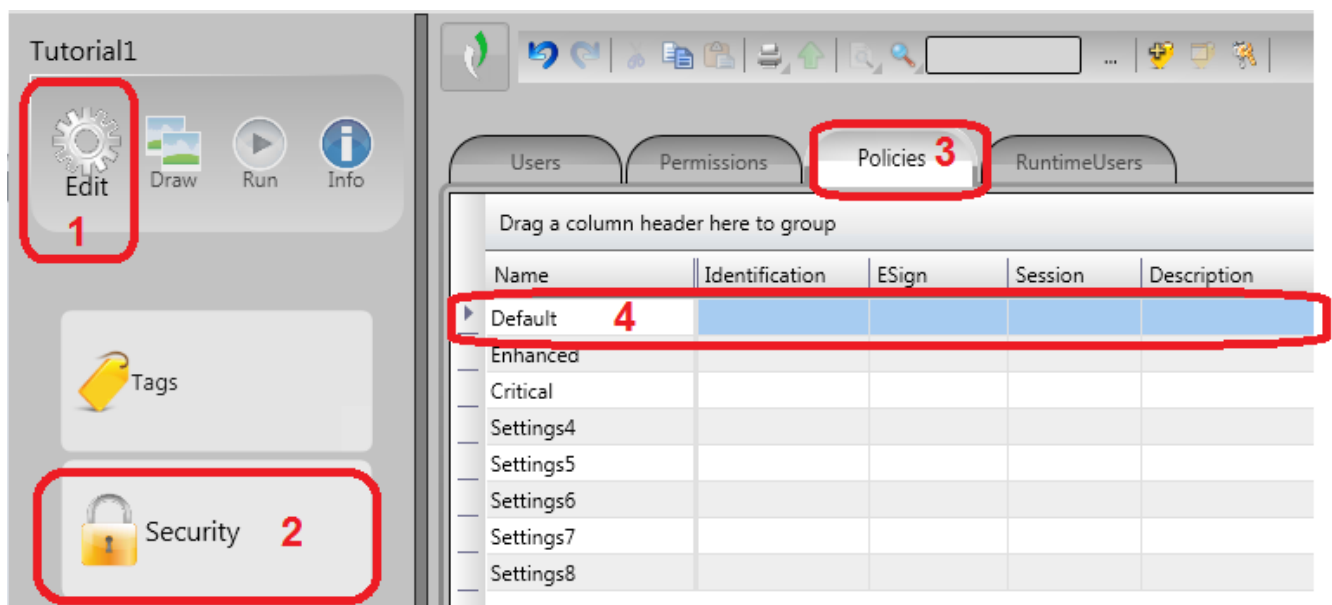


Figura 3-8. Edição de Políticas de Segurança

Os passos para editar as políticas de segurança existentes estão ilustrados na figura anterior e descritos a seguir:

1. Selecionar o menu *Edit*;

2. Acessar o item *Security*;
3. Clicar na guia *Políticas* para acessar as políticas de permissões do sistema;
4. Digitar os dados relativos às políticas de permissões dos usuários na linha marcada com uma seta (►).

Os campos de configuração desta aba são:

Name: nome da política;

Identification: definição das regras de senha, tanto para a edição de projetos como para acessar projetos em tempo de execução;

ESign: definição de um tempo limite, em minutos, para a sessão de *Runtime* (somente aplicável em tempo de execução);

Session: configura o comportamento da sessão após passado o tempo configurado em *ESign*. Esta definição somente faz o encerramento da sessão do usuário. O aplicativo continua a ser executado (somente aplicável em tempo de execução).

Visualização Usuários (RuntimeUsers)

Informa todos os usuários criados em tempo de execução.

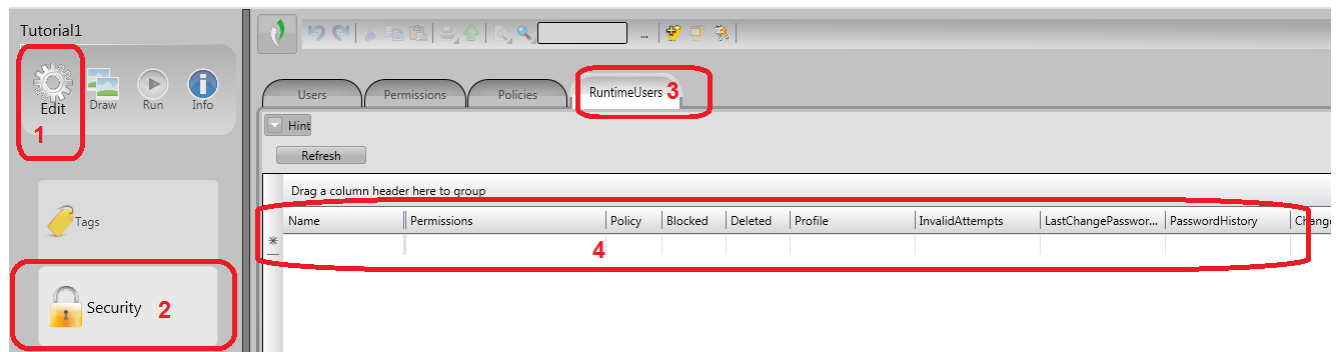


Figura 3-9. *RuntimeUsers*

Os passos para visualizar os usuários criados em tempo de execução são:

1. Selecionar o menu *Edit*;
2. Acessar o item *Security*;
3. Clicar na guia *RuntimeUsers* e clicar no botão *Refresh*.

Serão exibidos todos os usuários bem como suas configurações.

Editando Dispositivos



O BluePlant é fornecido com um driver OPC DA para coleta de informações de dispositivos remotos. Além do OPC, o BluePlant também oferece suporte a drivers de comunicação personalizados para acessar diretamente CPs, sistemas remotos de E/S, barramentos de campo padronizados, laços únicos e múltiplos, scanners, leitores de código de barras, dispositivos RFID e monitores digitais. Consulte o manual de suporte ao Dispositivo para mais informações sobre protocolo e programação de cada dispositivo.

ATENÇÃO:

Para mais informações sobre as configurações dos dispositivos (canais, nós e pontos de comunicação) deve ser acessado o documento de ajuda do dispositivo selecionado no campo *Installed Protocols* através do botão *Help* na guia *Channels*.

Canais (Channels)

A aba *Channels* configura os protocolos e os canais de comunicação. A figura a seguir ilustra essa seleção.

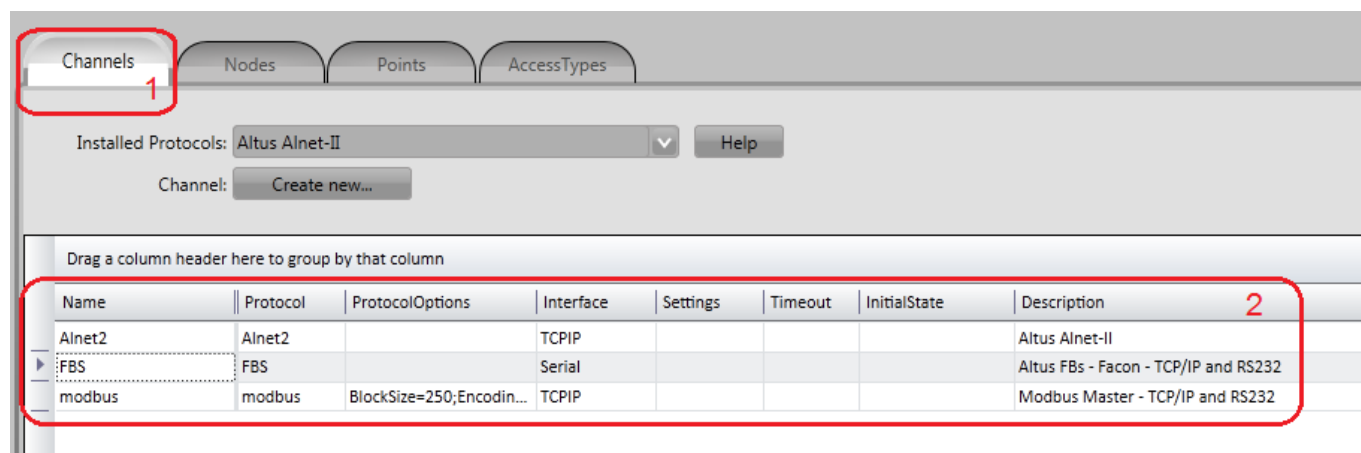


Figura 3-10. Edição de Canais de Comunicação

Os itens que compõem o menu de edição de canais de comunicação estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a opção *Channels*;
2. Observar que os dados relativos aos protocolos instalados aparecem na linha marcada com uma seta (▶).

Na sequência são explicados os itens do menu de edição de canais de comunicação conforme ilustrado na figura anterior.

Name

Estabelece o nome do canal.

Protocol

Define o protocolo de comunicação em execução em um Canal específico.

ProtocolOptions

Contempla a configuração específica do protocolo.

Interface

Indica a interface de comunicação para o canal. As interfaces disponíveis dependem do protocolo escolhido.

Settings

Inclui a definição das configurações da interface de comunicação. Cada tipo de interface possui ajustes específicos como, por exemplo, porta de comunicação, velocidade, bits de dados, bits de parada, paridade e sinais de controle. Um duplo clique na célula Settings permite acessar o menu suspenso com as configurações associadas à interface.

Timeout

Configurações de timeout para a interface de comunicação.

InitialState

Seleciona o estado inicial do canal na aplicação conforme opções indicadas a seguir:

- *Enabled*: o canal inicia o *Runtime* em execução;
- *Disabled*: o canal inicia o *Runtime* parado, deve ser colocado em execução durante a o *Runtime*;
- *Remote*: a opção Remote é utilizada quando está usando arquitetura distribuída e o *Channel* estará rodando numa outra máquina.

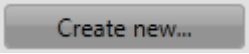
RemoteSettings

Define o endereço IP primário e backup dos servidores nos quais serão executados o canal.

DriverVersion

Apresenta a versão do driver do protocolo configurado no canal.

Criando um Canal de Comunicação

O botão  pode ser utilizado para criar um novo canal. A figura a seguir ilustra essa seleção.

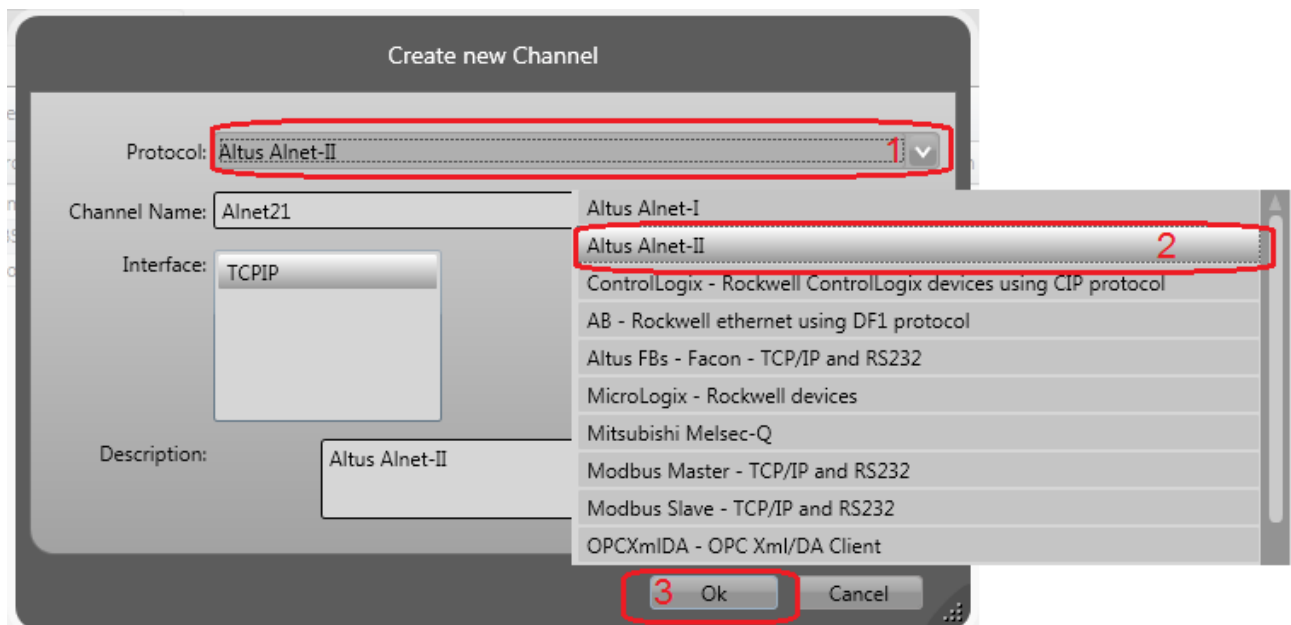


Figura 3-11. Edição de um Novo Canal de Comunicação

Na sequência será explicado o passo a passo para a criação de um novo canal de comunicação conforme exemplo ilustrado na figura anterior.

1. Selecionar a opção *Protocol* clicando com o botão esquerdo do mouse;
2. Selecionar nas opções existentes, aquela desejada. Neste caso: *Altus ALNET II*;
3. Finalizar clicando com o botão esquerdo do mouse no botão *Ok*.

Nós (Nodes)

No âmbito da computação, um nó é um ponto ou terminal na rede, onde uma mensagem pode ser criada, recebida ou transmitida. No contexto dos sistemas de supervisão operando em rede, o nó pode ser um dispositivo eletrônico ativo que está ligado a uma rede, e é capaz de enviar, receber ou transmitir informações através de um canal de comunicação. A figura a seguir mostra o menu de edição de nós para um determinado canal de comunicação. Um canal de comunicação que, conforme indicado anteriormente, está associado a um protocolo específico e pode conter um ou mais nós, caracterizando os dispositivos conectados a esse canal.

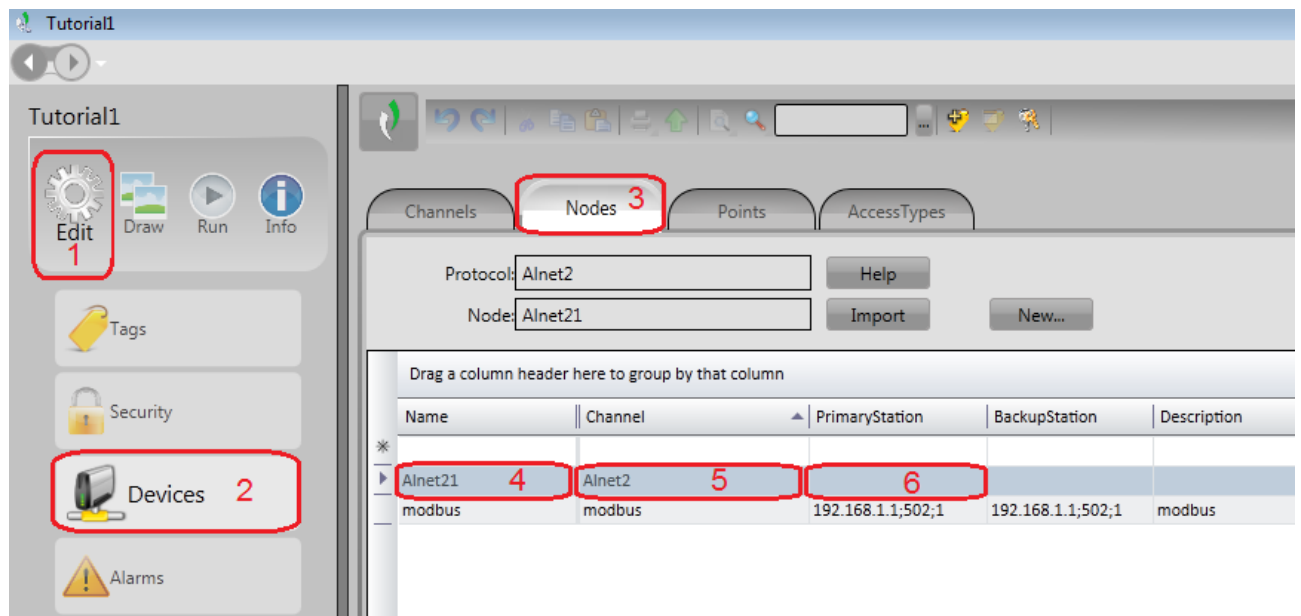


Figura 3-12. Edição do Nó para o Canal de Comunicação

Os passos para editar ou incluir os nós existentes estão ilustrados na figura anterior e descritos a seguir:

1. Selecionar o menu *Edit*;
2. Acessar o item *Devices*;
3. Clicar na guia *Nodes* para acesso às configurações dos nós;
4. Incluir ou editar o nome do nó na linha marcada com um asterisco (*);
5. Selecionar o *Channel* (protocolo) previamente adicionado no projeto;
6. Indicar através da coluna *PrimaryStation* qual o endereço do equipamento que estará relacionado ao nó.

Name

Indica o nome do nó.

Channel

Informa o canal de comunicação associado a este nó.

Primary Station

Referencia a estação do Nó. Define o Endereço IP, Porta e SlaveID. A sintaxe do campo da estação depende do protocolo. A figura a seguir mostra um exemplo de parametrização no caso do protocolo ALNET.

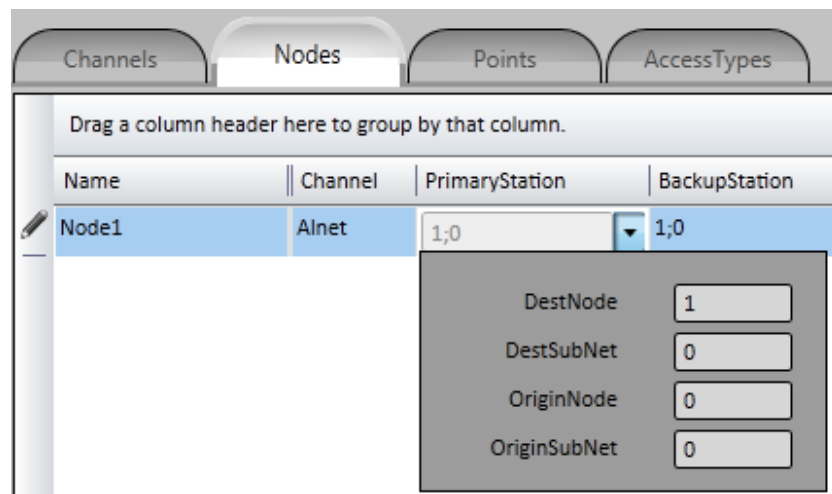


Figura 3-13. Parametrização da Estação Primária

Backup Station

Define o Endereço IP, Número de porta e SlaveID. Se este campo estiver definido, ao ocorrer uma falha de comunicação na estação principal, o sistema automaticamente tentará estabelecer uma comunicação com a estação de backup.

SyncDate

Apresenta a data da última sincronização executada pelo protocolo, válido para OPC.

SyncStation

Apresenta as configurações da estação que executou a sincronização de dados com este nó, válido para OPC.

SyncSettings

Apresenta as configurações usadas para a sincronização do nó, válido para OPC.

Pontos (Points)

Define os valores de aquisição de dados dos dispositivos de campo e mapeia os valores nas Tags.

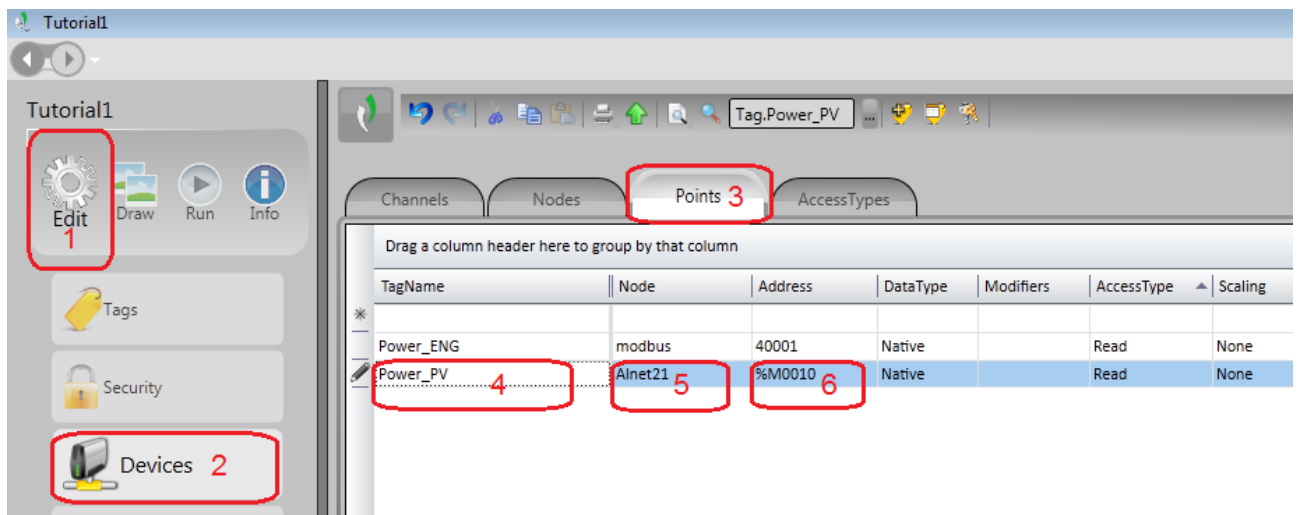


Figura 3-14. Edição de Pontos

Os passos para editar ou incluir os pontos de comunicação estão ilustrados na figura anterior e descritos a seguir:

1. Selecionar o menu *Edit*;
2. Acessar o item *Devices*;
3. Clicar na guia *Points* para acesso às configurações dos pontos de comunicação;
4. Incluir ou editar o nome da Tag existente (selecionável) na linha marcada com um asterisco (*);
5. Selecionar o nó (dispositivo configurado) previamente adicionado no projeto;
6. Indicar através da coluna *Address* qual o endereço do dado a ser recebido ou enviado do equipamento especificado.

TagName

Define a TagName a ser lida ou escrita no Dispositivo identificado.

Nodes

Define o Nó de comunicação associado ao ponto do dispositivo.

Address

Indica o endereço do ponto no dispositivo. A sintaxe do campo de endereço depende do protocolo. A figura a seguir ilustra um exemplo de parametrização do endereço (operando) no caso do protocolo ALNET.

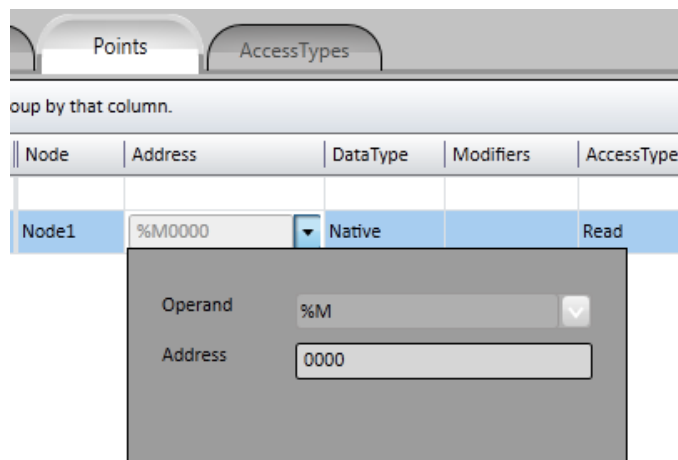


Figura 3-15. Exemplo de Parametrização do Endereço

Data Type

Define a conversão de dados aplicada aos dados de comunicação. Para a maioria dos protocolos pode-se utilizar a opção NATIVE, sendo assim, o sistema automaticamente tratará da conversão dos dados. Se um DataType diferente do NATIVE for selecionado, os padrões dos protocolos são desprezados, podendo haver truncamento dos dados.

Os tipos de dados possíveis são:

- Native (automático)
- Bit (binário – 1 bit)
- Byte (octeto – 8 bits)
- Char (menor unidade endereçável da máquina que pode conter caracteres básicos – 8 bits)
- Word (número inteiro – 16 bits)
- Short (número inteiro com sinal – 16 bits)
- Dword (número inteiro – 32 bits)
- Integer (número sem ponto decimal – 32 bits)
- Long (número inteiro com sinal – 64 bits)
- ULong (número inteiro sem sinal – 64 bits)
- BCD (número decimal codificado como binário – 8 bits)
- LBCD (Long BCD – 32 bits)
- Single (número de ponto flutuante com precisão simples – 32 bits)
- Real (número real – ponto flutuante – 64 bits)
- ASCII (codificação de caracteres de sete bits baseada no alfabeto inglês – 8 bits)
- Unicode (representação e manipulação de texto)
- OPCDateTime (data e hora padrão OPC)
- Timer (codificação de tempo)
- Counter (codificação de contagem)
- Control (tipo de dado de controle)

- SBCD (BCD com sinal – 8 bits)
- LSBCD (Long BCD com sinal – 32 bits)



Pesquisa na Documentação: compatibilidade de variáveis com MasterTool IEC XE

Consulte, na documentação do produto, a compatibilidade entre os tipos de Tags do BluePlant e as variáveis do programador MasterTool IEC XE, no caso do uso de protocolos de comunicação com a opção Native para os tipos de pontos.

Fonte: Manual de Utilização Série BluePlant – MU224000!

Modifiers

Oferece opções de modificação do valor das Tags, alterando a posição de BYTES, WORDs e/ ou DWORDS dos dados que são usados na comunicação, podendo haver alterações no comportamento e na disponibilidade das opções de acordo com o protocolo. Os seguintes campos podem ser definidos:

- *Bit*: permite manipular o bit definido, usado para *Points* de tamanho maior que o tipo BIT. Aplicável para os protocolos ALNET, ALNETII e FBS. Exemplo: para acessar o endereço %S50.2 em ALNET se deve selecionar no campo *Address* o operando %S e o endereço 50. Após, no campo *Modifiers* digitar no campo *Bit* o valor 2.

Drag a column header here to group by that column							
TagName	Node	Address	DataType	Modifiers	AccessType	Scaling	
* AlnetIDigS[2]	Alnet	%S0050	Byte	Bit=2	ReadWrite	None	
AlnetIDigS[1]	Alnet	%S0050	Byte	Bit=1	Write	None	
AlnetIDigS[0]	Alnet	%S0050	Byte	Bit=0	Read	None	
AlnetIDigM[2]	Alnet	%M0000	Word	Bit=2,ByteSwap=False	ReadWrite	None	
▶ AlnetIDigM[1]	Alnet	%M0000	Word	Bit=1,ByteSwap=False	Write	None	
AlnetIDigM[0]	Alnet	%M0000	Word	Bit=0,ByteSwap=False	Read	None	
AlnetIDigE[0]	Alnet	%E0000	Byte	Bit=0	Read	None	
AlnetIDigA[2]	Alnet	%A0000	Byte	Bit=2	ReadWrite	None	
AlnetIDigA[1]	Alnet	%A0000	Byte	Bit=1	Write	None	
AlnetIDigA[0]	Alnet	%A0000	Byte	Bit=0	Read	None	

Figura 3-16. Exemplo de Uso do Modificador BIT

- *ByteSwap*: troca de ordem entre dois BYTES. Exemplo: Dado recebido: 16# 0123456789ABCDEF. ByteSwap: 16# 23016745AB89EFCD.
- *WordSwap*: troca de ordem entre duas WORDs. Exemplo: Dado recebido: 16# 0123456789ABCDEF. WordSwap: 16# 45670123CDEF89AB.
- *DWordSwap*: troca de ordem entre duas DWORDS. Exemplo: Dado recebido: 16# 0123456789ABCDEF. DwordSwap: 16# 89ABCDEF01234567.
- *Stringlength*: delimitação do comprimento da string (somente valido para Tags de formato Text).

AccessType

Define o comportamento de leitura e escrita para cada ponto.

Label

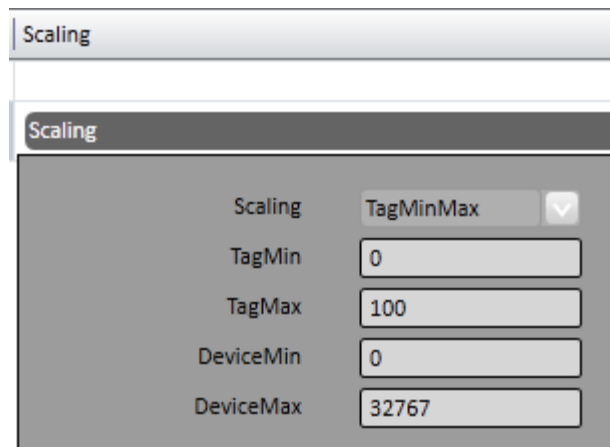
Adição de um rótulo para o ponto de comunicação.

Scaling

Estabelece a conversão de escala aplicada usando os dados de comunicação. As configurações típicas de conversão de escala são:

- *None* (nenhuma conversão);
- *TagMinMax* (escala baseada no valor mínimo e máximo da Tag);
- *Linear* (escala linear);
- *Equation* (escala via equação linear).

A figura a seguir ilustra a parametrização de conversão de escala baseada na opção TagMinMax.



Property	Value
Scaling	TagMinMax
TagMin	0
TagMax	100
DeviceMin	0
DeviceMax	32767

Figura 3-17. Conversão de Escala

Tipo de Acesso (AcessType)

Define as características do tipo de acesso comuns aos pontos do dispositivo. Podem ser configurados os tipos de acesso que controlam a leitura, escrita e outras configurações quando o aplicativo acessa dados do PLC ou IED, sempre de acordo com as possibilidades existentes no protocolo em questão. Como pode ser visto na figura a seguir, cujo modo de visualização selecionado é *CardView*, há três tipos de acesso padrão:

1. ReadWrite (leitura e escrita)
2. Write (escrita)
3. Read (somente leitura)

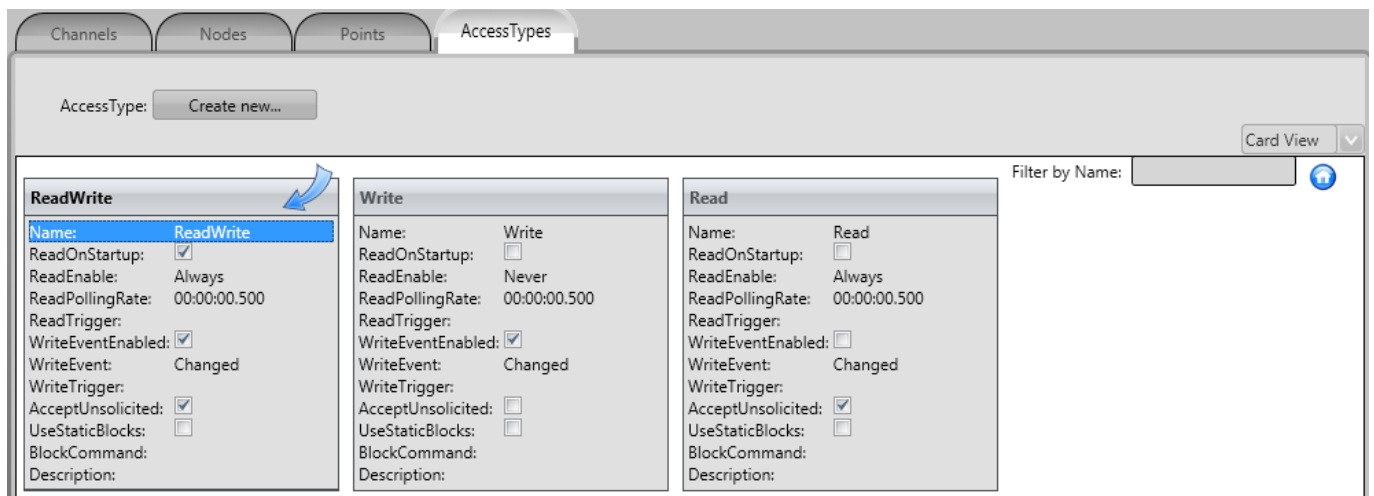


Figura 3-18. Tipo de Acesso

Clique em  para criar um novo AccessType.

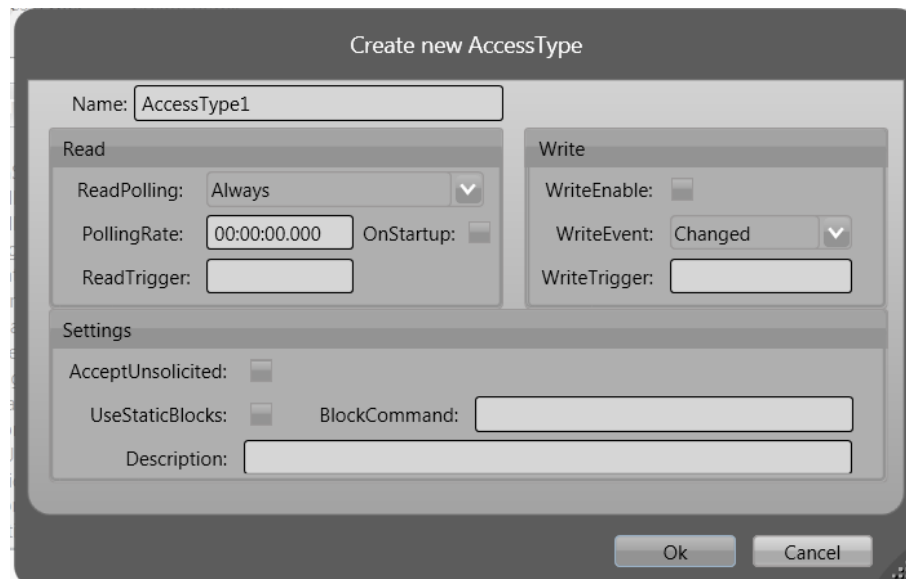


Figura 3-19. AccessType

Na sequência são explicados os principais itens do menu de tipo de acesso conforme ilustrado na figura anterior.

Name

Indica o nome do tipo de acesso.

ReadPooling

Define a leitura por pooling. Os atributos associados são:

- Always: leitura por polling ativa;
- Never: leitura por polling inativa;

- **OnDisplayOrServer**: o envio da requisição por parte do mestre só acontece durante a execução do display em que a Tag se encontra ou no Server.

PoolingRate

Define a taxa de leitura por pooling para cada *AccessType* se *ReadPooling* estiver selecionado.

OnStartup

Define o ponto de leitura na inicialização. Os atributos associados são:

- Selecionado: habilita a leitura na inicialização;
- Não selecionado: desabilita a leitura na inicialização.

ReadTrigger

A leitura é disparada quando o valor do objeto associado a esse campo for alterado.

WriteEnable

Habilita ou desabilita a escrita quando ocorrer um evento. Os atributos associados são:

- Selecionado: habilita a escrita;
- Não selecionado: desabilita a escrita.

WriteEvent

Configura o tipo de acesso de escrita ao ponto. Os atributos associados são:

- **Changed**: escreve quando a Tag vinculada é alterada;
- **ChangedUp**: escreve quando a Tag vinculada sofre um aumento de valor;
- **ChangedDown**: escreve quando a Tag vinculada sofre um decréscimo de valor.

WriteTrigger

A escrita é disparada quando o valor do objeto associado a esse campo for alterado.

AcceptUnsolicited

Define os atributos de aceite de mensagem não solicitada. Os atributos associados são:

- Selecionado: habilita mensagem não solicitada;
- Não selecionado: desabilita mensagem não solicitada.

UseStaticBlocks

Habilita ou desabilita o uso de *BlockCommand*.


BlockCommand

Deve ser preenchido com a definição de cada bloco que será criado. Deve ser verificado na documentação do driver se o mesmo permite o uso desta opção e quais são os valores válidos.

Editando Alarmes



Um Alarme pode ser configurado através da associação de uma Tag com um Grupo de Alarmes específico e um valor limite.

Para exibir um item de Alarme previamente configurado desenhe um objeto de alarme no ambiente de telas do seu projeto (Draw), inserindo o símbolo  (Alarm Window) a partir da barra de ferramentas esquerda.

Os métodos para o reconhecimento de Alarmes incluem objetos de alarme na tela, propriedades da Tag, grupos de alarme ou propriedades do item alarme, bem como os campos *AckAll* (global) e *AckMostPriority*.

NOTAS:

- **Reconhecer todos os alarmes:** pode-se usar a propriedade <Alarm.AckAll> que reconhece todos os alarmes configurados em um projeto com <Edit.Alarms.Items>;
- **Reconhecer alarme único ou alarme de alta prioridade:** a propriedade {Alarm.PriorityItem.UnAck} permite o reconhecimento do Alarme de alta prioridade configurado em <Edit.Alarms.Items> na coluna *Prioridade*. Se este for o único Alarme ou se este for de alta prioridade ele será reconhecido, mas somente se estiver Ativo ou Normalizado;
- **Reconhecer alarme específico:** para reconhecer um alarme específico, use a propriedade <Alarm.Items.IDxx.Unack>.

Grupos de Alarme (Groups)

Define o comportamento do tratamento de alarme comum para um grupo de itens de Alarme.

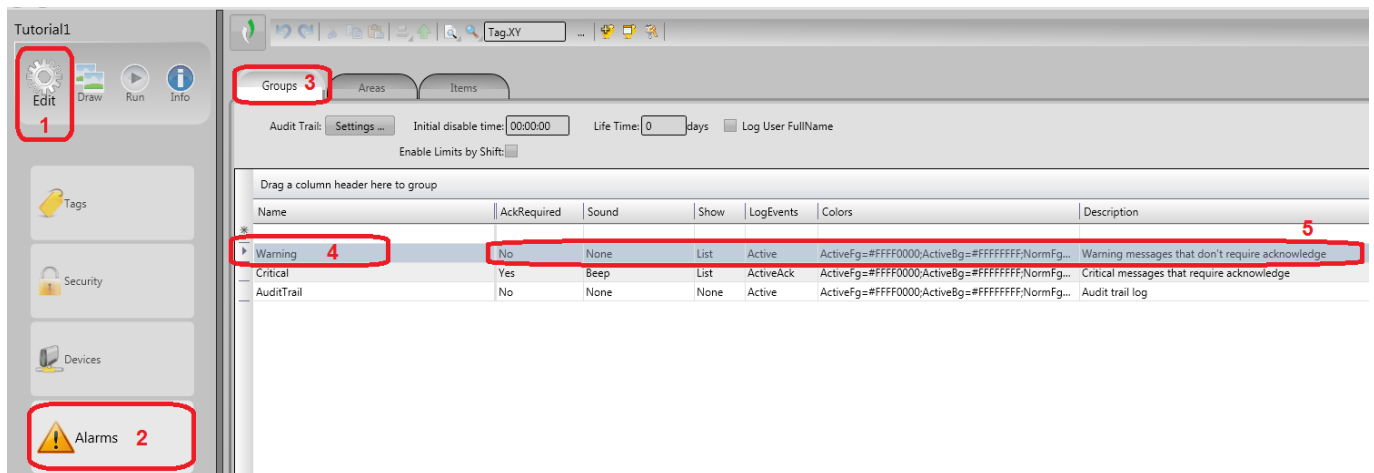


Figura 3-20. Grupos de Alarmes

Os passos para editar ou incluir os grupos de alarmes estão ilustrados na figura anterior e descritos a seguir:

1. Selecionar o menu *Edit*;
2. Acessar o item *Alarms*;
3. Clicar na guia *Groups* para acesso às configurações dos grupos de alarmes;
4. Incluir, linha marcada com asterisco (*), ou editar, linha marcada com a seta (▶), o nome do grupo de alarmes;

5. Selecionar as configurações do grupo especificadas nas colunas apresentadas na imagem acima.

Name

Indica o nome definido pelo usuário para o grupo de alarmes. Os grupos de alarme *Warning*, *Critical* e *SystemEvents* são incorporados e não podem ser excluídos, embora seus nomes e configurações possam ser alterados.

AckRequired

Configura as opções de reconhecimento para os alarmes do grupo. Os atributos associados são:

- *No*: Pontos de alarme definidos no grupo não requerem reconhecimento;
- *Yes*: Alarmes definidos no grupo requerem reconhecimento;
- *Comment*: Além de requerer reconhecimento, exibe uma tela para colocar um comentário sobre o motivo do reconhecimento.

AckTimeout

Define um tempo limite para o reconhecimento do alarme. Se o alarme não for reconhecido após o tempo especificado, o alarme volta para o estado ativo novamente.

AutoAckTime

Se o alarme não for reconhecido após o tempo especificado, o sistema reconhece o alarme automaticamente.

Sound

Habilita ou desabilita os sons do alarme, sendo estes alguns sons típicos, disponíveis no MS-Windows.

Show

Configura a exibição ou não do alarme (inclui a visualização no objeto alarmes online).

Log Events

Define o tipo de arquivamento de Historiador em eventos de Alarme. Os seguintes tipos são possíveis:

- *None* (sem log);
- *Active* (log quando o evento estiver ativo);
- *ActiveAck* (log quando o evento estiver ativo e for reconhecido);
- *ActiveNorm* (log quando o evento retorna ao normal);
- *All* (log em todas as condições acima).

¶ NOTA:

O banco de dados na qual as Tags são salvas é definido em *Edit/Dataset/DBs* no DB denominado *AlarmHistorian*.

Colors

Define a customização da tela de *Alarme* para cada linha de Alarme de acordo com seu grupo principal.

NotificationMethod

Nome de um método de alguma Classe de Script do tipo Server, que será chamado quando ocorrer qualquer alteração de estado de alarme neste grupo. O método tem que ter o protótipo:

```
void NotificationName (AlarmEventInfo[] info) .
```

Audit Trail

Seleciona quais eventos devam ser registrados nos arquivos de Audit Trail (registros para rastreamento de auditoria). Esta configuração é válida para os eventos de sistema que se deseja selecionar.

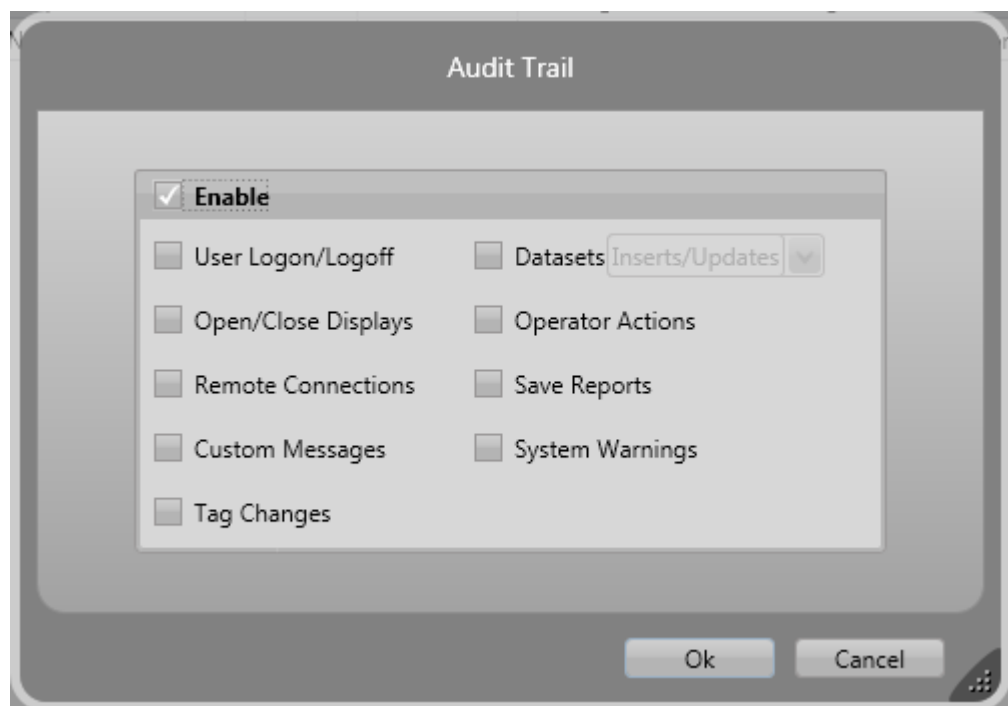


Figura 3-21. Audit Trail

Áreas de Alarme (Areas)

Outra funcionalidade disponível no tratamento de Alarmes é a configuração de Áreas Lógicas nas quais pode-se alocar os itens de Alarmes. Estas áreas lógicas são definidas no nível do projeto, e podem também ser definidas como Sub Áreas dentro de áreas, permitindo a configuração de agrupamentos lógicos dispostos hierarquicamente.

A facilidade serve para o tratamento em grupo de alarmes de uma mesma área. Pode-se então obter informações de quantos alarmes estão ativos ou reconhecidos em uma área ou sub área. Pode-se habilitar ou desabilitar todos os alarmes de uma área (incluindo suas sub áreas), ou apenas de sub áreas.

Para criar *Areas* siga o procedimento indicado a seguir:

1. Vá para *Edit>Alarms>Areas*;

2. Clique com o botão direito do mouse no nome do projeto e selecione *New Area*. Digite um nome para a Area;
3. Clique com o Botão direito do mouse a nova Area e selecione novamente *New Area* para criar uma sub-área;
4. Continue adicionando níveis filhos ou irmãos e inserção de ativos, conforme a necessidade.

Se necessário, clique com o botão direito uma Area para renomear ou apagar.

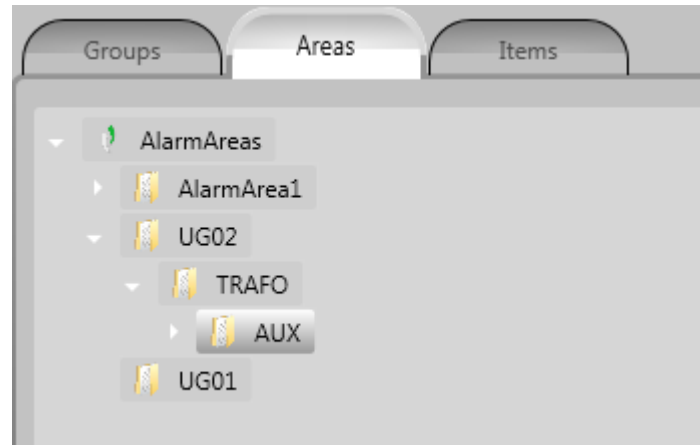


Figura 3-22. Áreas de Alarmes

Uma vez criada, a AREA ficará disponível na lista de existentes na coluna AREA da tabela de *Itens de Alarmes*. Uma AREA é um objeto específico que possui como propriedades, contadores de itens em alarme, reconhecidos, habilitados, desabilitados etc.

Itens de Alarme (Items)

Configura Tags para gerar alarmes sob condições definidas.

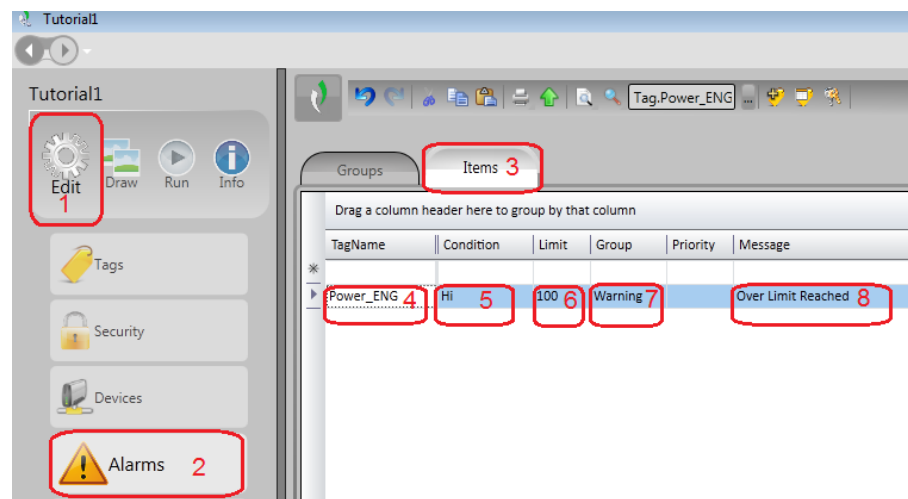


Figura 3-23. Itens de Alarmes

Os passos para editar ou incluir os grupos de alarmes estão ilustrados na figura anterior e descritos a seguir:

1. Selecionar o menu *Edit*;
2. Acessar o item *Alarms*;

3. Clicar na guia *Itens* para acesso as configurações dos itens de alarmes;
4. Incluir, linha marcada com asterisco (*), ou editar, linha marcada com a seta (►), o nome da Tag que gera o alarme;
5. Selecionar a condição de geração de alarme;
6. Configurar o limite que dispara o alarme;
7. Selecionar o grupo previamente criado com as características do alarme;
8. Inserir mensagem que será apresentada caso o alarme seja disparado.

Name

Define a *TagName* que será avaliado para gerar o Alarme.

Condition

Condição de avaliação para gerar alarmes. As opções disponíveis estão descritas na sequência.

- Hi: Tag \geq limite
- HiHi: Tag \geq limite (quando reconhecido automaticamente, reconhece o alarme Hi para a mesma Tag)
- Lo: Tag \leq limite
- LoLo: Tag \leq limite (quando reconhecido automaticamente, reconhece o alarme Lo para a mesma Tag)
- RateOfChange: Taxa de alteração da Tag \geq limite
- DeviationMinor: Valor absoluto (Tag - Setpoint) < limite (Setpoint definido na coluna Setpoint)
- DeviationMajor: Valor absoluto (Tag - Setpoint) > limite (Setpoint definido na coluna Setpoint)
- Equal: Tag = limite
- GreaterThan: Tag > limite
- GreaterEqual: Tag \geq limite
- LessThan: Tag < limite
- LessEqual: Tag \leq limite
- Changed: valor da Tag foi alterado
- ChangedUp: valor da Tag aumentou
- ChangedDown: valor da Tag diminuiu
- NotEqual: valor da Tag diferente do limite

Limit

Define o valor para avaliar as condições do alarme.

Group

Define o nome do Grupo no *EditAlarmGroups* onde o comportamento do item Alarme está especificado. Os grupos pré-definidos são:

- *Critical* (mensagens críticas que exigem reconhecimento);
- *SystemEvent* (log de eventos de sistema);
- *Warning* (mensagens de advertência que não exigem reconhecimento).

Priority

Define a Prioridade do Alarme. Este campo aceita valores de 0 a 999999999, onde o maior valor corresponde à prioridade mais alta. A mesma prioridade pode ser atribuída para mais de um evento de alarme.

Message

Define uma Mensagem, ou valores de Tag, a ser exibida quando ocorrer o Alarme. Para tanto acrescente o nome da Tag desejado na mensagem usando a notação de vinculação XAML entre chaves, conforme ilustrado no exemplo a seguir:

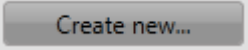
```
ValordaTag = {NomedadaTag.Value}
```

Editando Banco de Dados



Conforme mencionado anteriormente, o módulo de conjunto de dados (Datasets) incluído no BluePlant fornece uma interface de fácil operação para a troca de dados em tempo real com bancos de dados externos, XML, CSV ou arquivos de texto, bem como a possibilidade de acessar tabelas e consultas SQL. O banco de dados em tempo real garante, sem a necessidade de qualquer programação adicional, a sincronização de dados entre vários processos no servidor e múltiplas estações cliente. Um vasto conjunto de propriedades internas, tais como qualidade de dados, estampa de tempo, bloqueio de estado e valor bloqueado simplificam a criação de aplicações.

DBs

Através da guia DBs menu *Edit > Datasets > BDs* é possível acessar a configuração do banco de dados. A partir do botão  estão disponíveis as opções para criação de uma nova conexão, estas opções são apresentadas a partir de uma tela conforme mostrado na figura adiante.

Para criar a nova conexão é necessário escolher o nome da mesma, um *provider* dentro da lista de *providers* disponíveis no ambiente e o *database*. Podemos também criar uma descrição para a nova conexão.



Figura 3-24. Criação de Novo Database

Name

Estabelece o nome para a conexão com o DB.

Provider

Seleciona o fornecedor para o DB quando da criação de uma nova conexão de base de dados. Os fornecedores padrão são:

- Odbc Data Provider - Permite o acesso a um banco de dados ODBC através de um driver ODBC nativo;
- OleDb DataProvider - Fornece aplicativos para acesso aos dados armazenados em diversas fontes de informação;
- OracleClient Data Provider - Fornece acesso aos bancos de dados da Oracle;
- SqlClient Data Provider - É uma coleção de classes que podem ser usadas para acessar bancos de dados SQL Server;
- SQLite Data Provider - Fornece acesso aos bancos de dados da SQLite;
- PostgreSQL Data Provider - Fornece acesso aos bancos de dados da PostgreSQL.

Outros fornecedores podem ser criados adicionando-se modelos XML ao produto na subpasta *DBProviders* localizada na pasta de instalação do software BluePlant. Exemplos de provedores de dados: OdbcDataProvider, Firebird database, Microsoft Access Database, Microsoft Excel Database, ODBC using DSN, ODBC using fileDSN, Oracle Database e SQL Server Database.

Database

Seleciona o banco de dados quando da criação do objeto DB. A lista de base de dados disponíveis é criada dinamicamente baseada em fornecedores encontrados na subpasta *DBProviders* localizada na pasta de instalação do software BluePlant. As bases de dados mais comuns são: Firebird Database, CSV File Database, Microsoft Access Database, Microsoft Excel Database, Oracle Database, ODBC DSN, ODBC FILEDSN, SQL Server Database e SQL Server Database Windows Authentication.

ConnectionString

Esta coluna permite definir a string de conexão usada para comunicar-se com o banco de dados. Para tanto, digite o caminho da fonte de dados onde está localizado o arquivo do banco de dados.

Clique em  para testar se a fonte de dados foi localizada e está válida.

NameLogon

Define o nome do logon para conectar-se com o banco de dados.

Password

Senha solicitada para conectar-se com o banco de dados. Este campo pode ser editado somente pelo usuário Administrador.

Estudo Dirigido 3-1: conexão com bases de dados

Consulte a documentação do produto (Manual de Utilização Série BluePlant – MU224000) e, com o auxílio do instrutor, configure conexões com as seguintes bases de dados:

1. Excel via ODBC;
2. Excel via ODBC DSN;
3. Excel via OLEDB;
4. Oracle;



5. SQLServer.

Tables

A guia *Tables* permite acessar as tabelas de dados a partir das bases de dados configuradas (DBs). A Figura 3-25figura a seguir ilustra essa seleção.

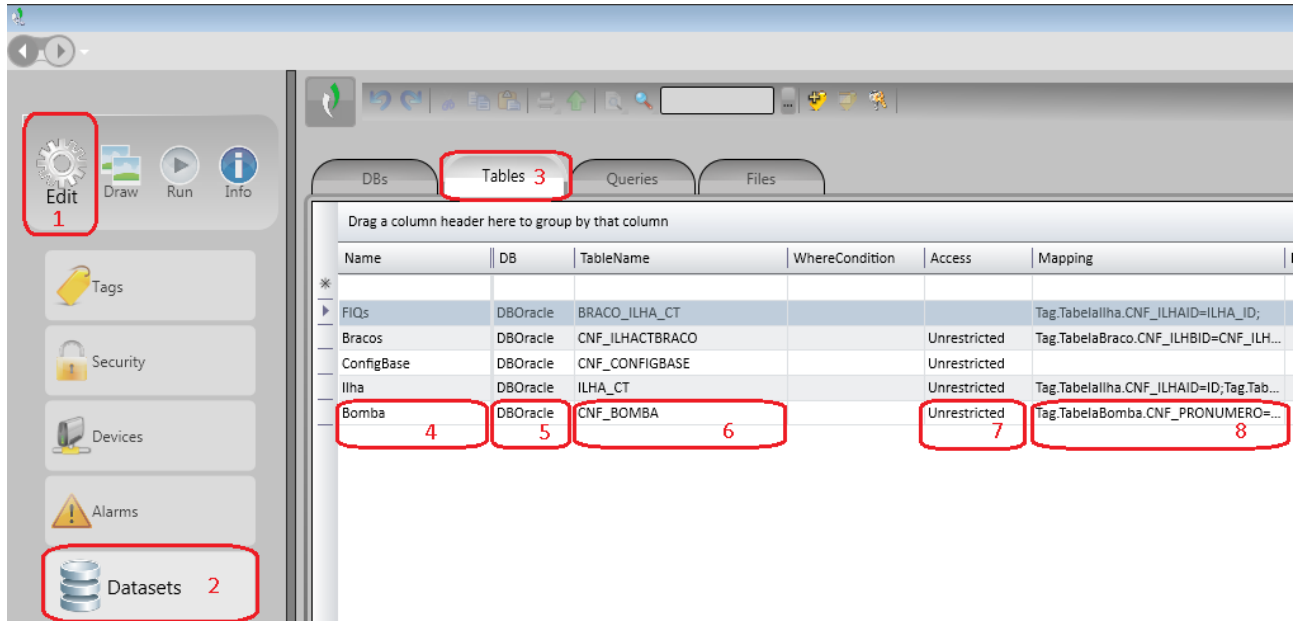


Figura 3-25. Edição de Tabelas de Dados

Os passos para editar ou incluir tabelas estão ilustrados na figura anterior e descritos a seguir:

1. Selecionar o menu *Edit*;
2. Acessar o item *Datasets*;
3. Clicar na guia *Tables* para acesso às configurações das tabelas;
4. Incluir ou editar o nome da tabela na linha marcada com um asterisco (*);
5. Selecionar qual conexão com o banco de dados previamente criada será utilizada;
6. Selecionar qual a tabela do banco de dados que irá comunicar;
7. Selecionar qual a permissão de acesso ao banco de dados para a tabela que configurada;
8. Configurar na coluna *Mapping*, as Tags relacionadas com as colunas da tabela do bando de dados.

Name

Indica o nome da Tabela usada nos objetos *Runtime*.

DB

Indica o nome da DB (conexão da Base de Dados) a qual pertence à tabela de dados.

TableName

Indica o nome da tabela assim como no banco de dados.

WhereCondition

Indica qual a condição em que a instrução SQL será executada.

Access

Define a política de segurança com operações permitidas para esta DataTable. As opções são:

- Read (leitura);
- Insert (inserção);
- ReadWrite (leitura e escrita);
- Unrestricted (irrestrita).

Mapping

Mapeia os conteúdos da tabela de dados com os objetos Tag. Ao executar os comandos de seleção e próximo, a primeira linha no resultado selecionado é aplicada às Tags. Então, ao executar o comando de atualização, os conteúdos da Tag são escritos na linha.

Queries

Esta guia mapeia as bases de dados configuradas (DBs) conforme mostrado na figura a seguir.

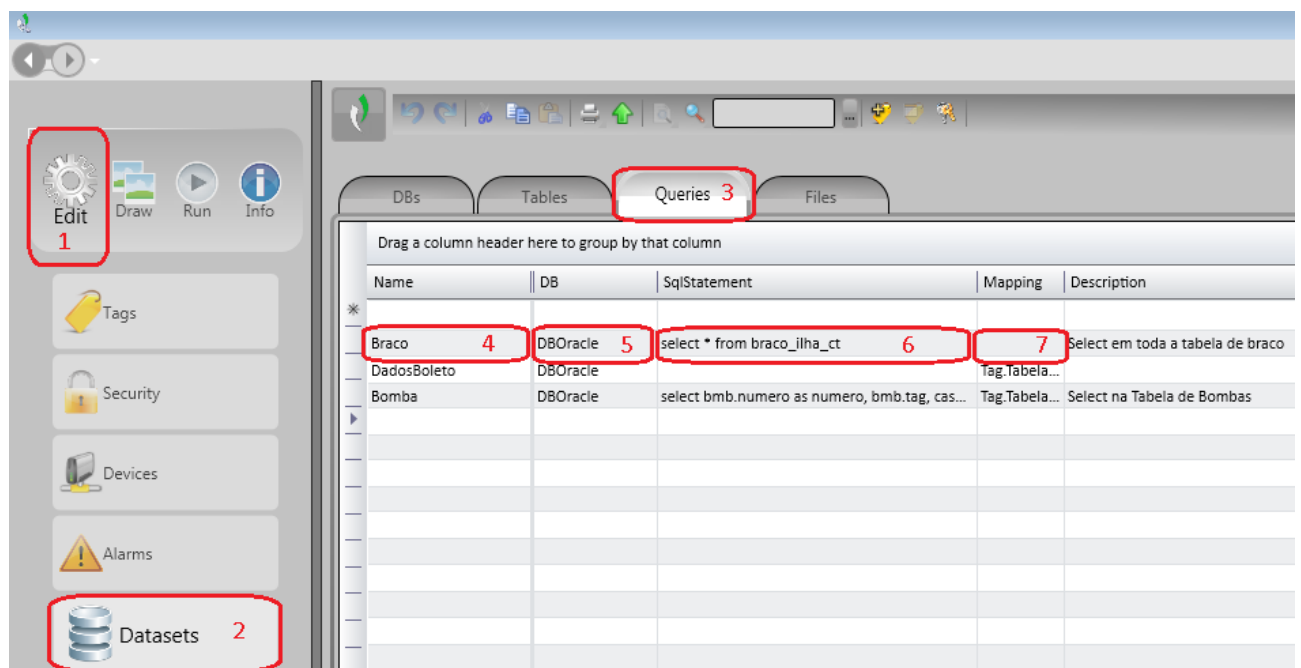


Figura 3-26. Edição de Queries

Os passos para editar ou incluir instruções SQL (queries) estão ilustrados na figura anterior e descritos a seguir:

1. Selecionar o menu *Edit*;
2. Acessar o item *Datasets*;
3. Clicar na guia *Queries* para acesso às configurações das instruções SQL;
4. Incluir ou editar o nome da instrução na linha marcada com um asterisco (*);
5. Selecionar qual conexão com o banco de dados previamente criado que será utilizada;

6. Configurar a instrução SQL na coluna *SqlStatement*;
7. Configurar na coluna *Mapping* as Tags relacionados às colunas da tabela do banco de dados.

Name

Nome para o objeto Query usado nos objetos de *Runtime*.

DB

Nome da conexão do banco de dados usada para o armazenamento dos queries.

SqlStatement

Instrução SQL utilizada para realizar consultas no banco de dados.

Mapping

Mapeia os conteúdos da tabela de dados em Tags. Ao executar o comando de seleção, a primeira linha no resultado é aplicada às Tags.

Files

A guia Files define o(s) arquivo(s) para a troca de informações com os bancos de dados conforme ilustrado na figura a seguir.

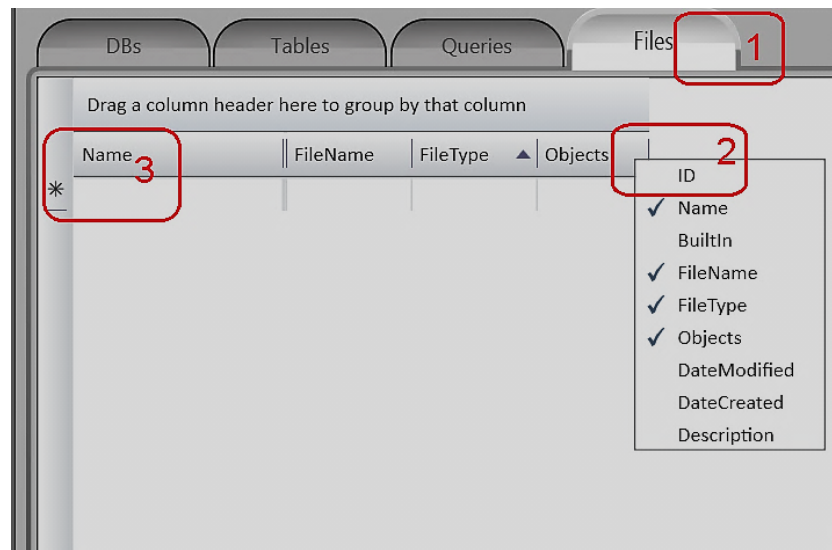


Figura 3-27. Edição Arquivos de Bancos de Dados

Os itens que compõem o menu edição de arquivos de bancos de dados estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a opção *Files* clicar com o botão esquerdo do mouse;
2. Clicar com o botão direito do mouse em um cabeçalho de coluna para visualizar os itens desejados;
3. Observar que os dados relativos aos arquivos irão aparecer nas linhas da tabela.

Na sequência são explicados os itens de edição de arquivos de bancos de dados conforme mostrado na figura anterior.

Name

Define o nome do arquivo usado nos objetos de *Runtime*.

FileName

Nome do arquivo e caminho. Exemplo: C:\BDs\file1.txt.

FileType

Caracteriza os tipos de arquivo. Os tipos de arquivo possíveis são:

- ASCII (codificação de caracteres de sete bits baseada no alfabeto inglês);
- Unicode (representação e manipulação de texto);
- XML (linguagem de marcação para a criação de documentos com dados organizados hierarquicamente).

Objects

Mapeia o conteúdo do arquivo com objetos Tag.

Editando Scripts



Um script pode ser executado na ocorrência de um evento de disparo ou ciclicamente através da definição de um intervalo de tempo para tal. As opções que compõem o menu de edição de scripts estão detalhadas na sequência.

Tasks

A figura a seguir enfatiza a aba *Tasks* para edição de tarefas de script. Existem quatro tarefas incorporadas:

1. *ServerStartup*: o script é executado na máquina do Servidor (TServer.exe) quando o projeto entra em execução;
2. *ServerShutdown*: o script é executado na máquina do Servidor na parada da operação;
3. *ClientStartup*: o script é executado em cada máquina cliente quando o TVisualizer.exe (módulo de telas) inicia sua execução;
4. *ClientShutdown*: o script é executado em cada máquina cliente quando o módulo de telas é encerrado.

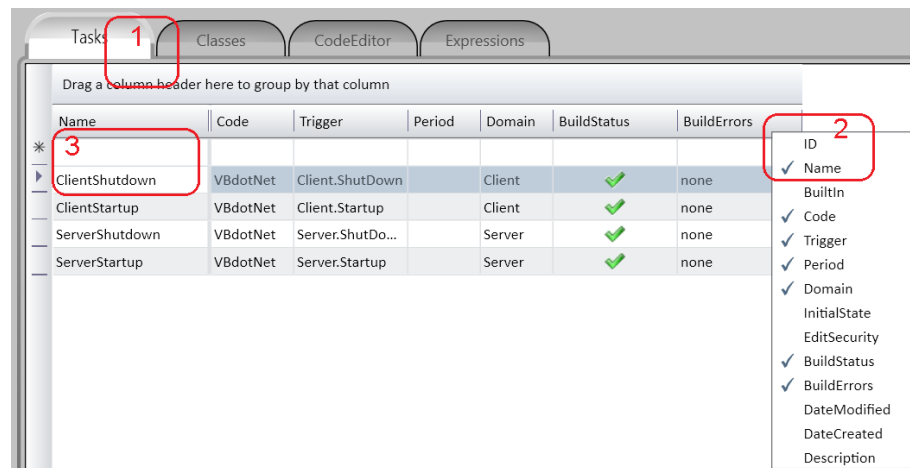


Figura 3-28. Edição de Tarefas de Script

Os itens que compõem o menu de edição de tarefas de script estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a guia *Tasks*;
2. Selecionar as opções de configuração das colunas da tabela desejadas clicando com o botão direito do mouse no cabeçalho de uma coluna e marcando os itens para exibição;
3. Observar: os dados relativos às tarefas de script irão aparecer nas linhas da tabela.

Na sequência são explicados os itens do menu edição de tarefas de script conforme mostrado na figura anterior.

Name

Indica o nome para o objeto Script.

Code

Define a linguagem do Script. O usuário pode selecionar entre VBdotNet ou CSharp.

Trigger

Tag ou objeto que dispara a execução da Tarefa. A tarefa é executada quando o valor do objeto muda.

Period

Período de tempo para executar uma Tarefa.

Domain

Define se o Script é executado na estação do Servidor ou em cada estação Cliente.

InitialState

Seleciona o estado inicial das tarefas:

- **Enabled:** a tarefa está pronta para ser executado desde o início;
- **Disabled:** a tarefa não está pronta para ser executado. Permite ativar a tarefa em condições específicas.

EditSecurity

Permite escolher o nível de segurança para a edição do código desta Tarefa. Ao clicar aparece a janela com as categorias de usuários que terão acesso para alterar esta TASK.

BuildStatus

Indica o status da última compilação: verde significa compilado com sucesso e vermelho indica erro de compilação

BuildErrors

Status de contagem de erro a partir da última compilação do Script. Atributos associados: ReadOnly (somente leitura).

Classes

Nesta opção estão listadas as classes definidas pelo usuário via biblioteca de métodos. A figura que segue mostra esse item do menu *Script*. Há duas *UserClasses* incorporadas:

1. *ServerMain*: biblioteca de métodos disponível para todas as tarefas do servidor e clientes;
2. *ClientMain*: biblioteca de métodos disponível para todos os clientes scripts, inclusive telas scripts.

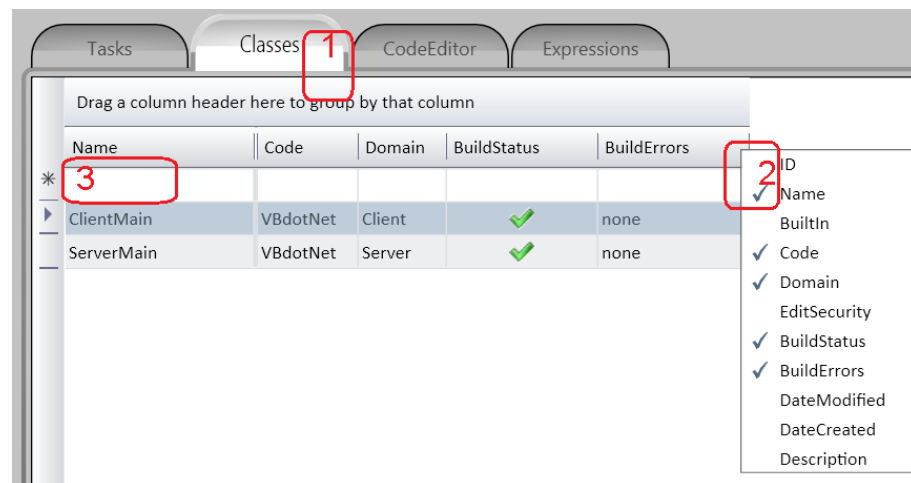


Figura 3-29. Classes de Script

Os itens que compõem o menu de edição de classes de script estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a guia *Classes*;
2. Selecionar as opções de configuração das colunas da tabela desejadas clicando com o botão direito do mouse no cabeçalho de uma coluna e marcando os itens para exibição;
3. Observar: os dados relativos às classes de script irão aparecer nas linhas da tabela.

Na sequência são explicados os itens do menu de edição de tarefas de script conforme mostrado na figura anterior.

Name

Indica o nome da classe do usuário.

Code

Indica a linguagem do Script. O usuário pode selecionar entre VBdotNet ou CSharp.

Domain

Define se os métodos nessa classe de usuário serão visíveis para os scripts do Servidor ou Cliente.

CodeEditor

Trata-se do painel editor de código das tarefas e classes de usuário conforme ilustrado na figura a seguir Figura 3-30 (item 1). As opções de linguagem de programação são: *VBdotNet* ou *CSharp*. A seleção da linguagem está ilustrada nessa mesma figura (item 2). Informações sobre a sintaxe das linguagens de programação podem ser obtidas no seguinte endereço <http://www.microsoft.com/net>.

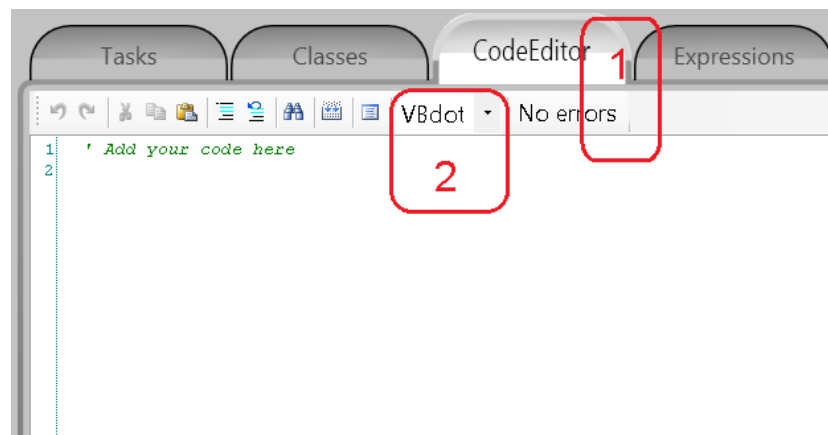


Figura 3-30. CodeEditor



Um pouco de teoria... Linguagem de Programação VB e C#

Linguagem VB

O Visual Basic (abreviado como VB) é uma linguagem de programação produzida pela empresa Microsoft, e é parte integrante do pacote Microsoft Visual Studio. Sua versão mais recente faz parte do pacote Visual Studio .NET, voltada para aplicações .Net. Sua versão anterior fez parte do Microsoft Visual Studio 6.0, ainda muito utilizado atualmente por aplicações legadas.

Um aperfeiçoamento do BASIC, a linguagem é dirigida por eventos (event driven), e possui também um ambiente de desenvolvimento integrado (IDE — *Integrated Development Environment*) totalmente gráfico, facilitando enormemente a construção da interface das aplicações (GUI — *Graphical User Interface*). O nome Visual Basic, é Derivado de:

- Basic — a linguagem de Programação
- Visual — o Nome do Pacote EX: Visual studio (Visual C++, Visual C#, Visual Basic .NET)

Visual Basic era muito usada em ambientes corporativos: uma pesquisa de 2005 indicou que 62% dos programadores usavam uma forma de Visual Basic, seguido de C++, JavaScript, C# e Java. Atualmente foi suplantada pelo Java.

Em suas primeiras versões, o Visual Basic não permitia acesso a bancos de dados, sendo, portanto, voltado apenas para iniciantes, mas devido ao sucesso entre as empresas — que faziam uso de componentes adicionais fabricados por terceiros para acesso a dados — a linguagem logo adotou tecnologias como DAO, RDO, e ADO, também da Microsoft, permitindo fácil acesso a bases de dados. Mais tarde foi adicionada também a possibilidade de criação de controles ActiveX, e, com a chegada do Visual Studio .NET, o Visual Basic — que era pseudo-orientada a objetos — tornou-se uma linguagem totalmente orientada a objetos (OO).

Existem várias linguagens derivadas, entre as quais:

- VBScript é a linguagem default (por definição) para Active Server Pages e pode ser usada no *scripting* (programação) de Windows e de páginas da Internet.
- Visual Basic .NET é a nova versão do Visual Basic, que é parte integrante da plataforma Microsoft .NET. Essa versão não é totalmente compatível com as versões anteriores, mas existe a possibilidade de converter códigos antigos, que após uma revisão podem ser usados no Visual Basic .NET. Para fins de comparação, essa linguagem usa o paradigma de Orientação a Objeto e você encontrará muita semelhança com o Java.
- *Visual Basic for Applications* (VBA) permite a criação de macros, e está integrado em todos os produtos da família de produtos Microsoft Office, e também em outros produtos de terceiros tais como Visio (agora pertencente à Microsoft) e WordPerfect Office 2002.

A partir de 2002 (a primeira versão da plataforma .Net) a linguagem Visual Basic mudou em vários aspectos ganhando muitos recursos utilizados anteriormente em outras linguagens como Java e C++ (herança, polimorfismo, etc.), porém continuou com a mesma sintaxe, mas por ser Orientado a Objetos as suas funções e métodos foram encapsulados em namespaces e classes.

Outras grandes melhorias com a plataforma .Net que fortaleceram a linguagem foi a possibilidade de programação para WEB (ASP.Net), dispositivos móveis, Windows Forms e mais recentemente Silverlight. Seu acesso a dados, que sempre foi um dos principais recursos, foi melhorado com o ADO.Net (baseado em XML) permite um acesso desconectado com o banco de dados. Com o lançamento do VB 2008 Express Edition, o XAML e seu design gráfico estrearam com novos recursos isso.

Exemplo de um código VB.NET...

```
Private Sub Form_Load()  
    ' Executa uma simples caixa de mensagem e escreve "Olá mundo!"  
    MsgBox "Olá mundo!"  
End Sub
```

Fonte: http://pt.wikipedia.org/wiki/Visual_Basic

Linguagem C#

Durante o desenvolvimento da plataforma .NET, as bibliotecas foram escritas originalmente numa linguagem chamada Simple Managed C (SMC), que tinha um compilador próprio. Mas, em Janeiro de 1999, uma equipe de desenvolvimento foi formada por Anders Hejlsberg, que fora escolhido pela Microsoft para desenvolver a linguagem. Dá-se início à criação da linguagem chamada Cool. Um pouco mais tarde, em 2000, o projeto .NET era apresentado ao público na Professional Developers Conference (PDC), e a linguagem Cool fora renomeada e apresentada como C#.

A criação da linguagem, embora tenha sido feita por vários programadores, é atribuída principalmente a Anders, hoje um Distinguished Engineer na Microsoft. Ele fora o arquiteto de alguns compiladores da Borland, e entre suas criações mais conhecidas estão o Turbo Pascal e o Delphi.

A Microsoft submeteu o C# à ECMA para uma padronização formal. Em Dezembro de 2001 a associação liberou a especificação ECMA-334 Especificação da Linguagem C#. Em 2003 tornou-se um padrão ISO (ISO/IEC 23270). Há algumas implementações em desenvolvimento, destacando-se a Mono, implementação open source da Novell, o dotGNU e o Portable.NET, implementações da Free Software Foundation, e o BDS 2008, implementação da CodeGear.

A Microsoft anunciou planos de adicionar o suporte a tipos parciais, generics e outras características. A padronização pela ECMA/ISO destas características foi solicitada, mas ainda não é parte da versão padrão da linguagem.

A linguagem suporta ponteiros através da palavra reservada unsafe (código não seguro), que é obrigatório. Seu uso não é aconselhável, e blocos de códigos que o usam geralmente requisitam permissões mais altas de segurança para poderem ser executados. As operações aritméticas são verificadas contra transbordamento de dados. C# também suporta as operações aritméticas básicas como adição, subtração, multiplicação e divisão. Esses símbolos são chamados operadores, e "operam" os valores das variáveis.

Um coletor de lixo também é suportado, um processo usado para a manutenção de memória. Com este recurso, é possível recuperar a zona de memória que um programa já não utiliza. Quando isto não ocorre pode acontecer a chamada perda de memória, um erro comum que pode levar ao término não desejado do programa em execução por esgotamento da memória livre.

Em C# não existe herança múltipla, ou seja, cada classe só pode herdar apenas outra classe e não mais do que uma, no entanto é possível simular herança múltipla utilizando interfaces. Através da herança reduz-se o código através da sua reutilização.

Os gabaritos não são suportados, mas a linguagem possui um suporte abrangente a generics. Nela podem se usar tipos genéricos para a maximização da reutilização de código, segurança de tipo, e desempenho. A utilização mais frequente é para a criação de classes. Pode criar as suas próprias interfaces genéricas, métodos, classes, eventos e delegates. As classes genéricas podem ser utilizadas para permitir acesso aos métodos usando tipos de dados específicos. Informações sobre os tipos usados em um tipo de dados genérico podem ser obtidas em tempo de execução por meio de reflexão.

Há três tipos de passagem de parâmetros em C#, por valor, por referência e por saída. Na passagem por valor é feita uma cópia do argumento da chamada do método para o parâmetro do mesmo. Isso significa que as alterações que ocorrem nas variáveis passadas por cópia dentro do método não se refletem fora dele. Na passagem por referência toda alteração feita na variável passada por referência dentro do método alterará também seu valor fora dele.

Por padrão todo objeto e vetor são passados por referência e toda estrutura e variáveis primitivas são passadas por valor. Para se forçar a passagem por referência, tanto a chamada do método como a declaração dele devem estar com os argumentos e parâmetros precedidos pelas palavras reservadas `ref` ou `out`. A primeira é usada para se manipular um parâmetro já inicializado antes da chamada do método, de forma a somente editá-lo. A segunda é usada para se inicializar um parâmetro durante a execução do método, retornando o resultado para o método que o chamou.

Ao contrário das outras linguagens de programação, nenhuma implementação de C# atualmente inclui qualquer conjunto de bibliotecas de classes ou funções. Mesmo assim, esta linguagem está muito vinculada à plataforma .NET, da qual obtém as suas classes ou funções de execução. O código é organizado num conjunto de espaços de nomes que agrupam as classes com funções semelhantes. Por exemplo, `System.Windows.Forms` contém o sistema Windows Forms; `System.Console` é usado para entrada/saída de dados.

Um nível de organização superior é fornecido pelo conceito de montador, que pode ser um simples arquivo ou múltiplos arquivos ligados juntos que podem conter muitos espaços de nomes ou objetos. Programas que precisam de classes para realizar uma função em particular podem se referenciar aos montadores como `System.Drawing.dll` e `System.Windows.Forms.dll` assim como a biblioteca core (conhecida como `microsoft.dll` na implementação da Microsoft).

Exemplo de um programa em C# que exibe o texto "Olá Mundo!" na tela...

```
using System;

namespace Teste
{
    class OlaMundo
    {
        static void Main()
        {
            Console.WriteLine("Olá Mundo!");
        }
    }
}
```

Fonte: <http://pt.wikipedia.org/wiki/Csharp>

Expressions

Utilizada para a criação de Expressões que também funcionam como uma linha simples de código que será executada em determinadas condições, fornecendo um resultado.

Além da tabela que pode ser atualizada por *Edit> Script> Expressions*, em vários outros lugares no espaço de Engenharia do Blueplant, pode-se digitar e usar expressões, utilizando a mesma sintaxe descrita aqui, para obter um valor resultado da expressão ou para configurar uma condição.

Expressions são meramente codificações na sintaxe VB.Net, como expressões aritméticas ou chamadas para métodos de classes definidas ou acessíveis no projeto. A facilidade *Intellisense* está orientada apenas para Tags e objetos de aplicação, não considerando a linguagem, mas as expressões são compiladas usando o compilador padrão VB.Net, então o que puder escrever sem erros de sintaxe no editor de código, também poderá escrever para o código das Expressões.

O Blueplant implementa algumas substituições automatizadas, como == para =, então a sintaxe de uma expressão é muito próxima de uma declaração em C#, mas sem a necessidade de adicionar o ";" no final.

Desta forma, tanto os programadores VB.NET como os de C # são capazes de usar os campos de expressão, sem diferenças.

Em expressões, não precisa colocar @ antes dos nomes de Tag. Precisa do @ no editor de código para diferenciar as Tags do projeto das variáveis dot NET. No entanto, como as expressões não têm Variáveis locais dot NET, então pode-se usar os objetos de projeto diretamente.

Para operadores aritméticos, deve-se usar os operadores padrão conforme descrito na documentação do dot NET.

Para permitir a avaliação de uma única linha, a linguagem dot NET dispõe do comando IIF, que atualmente é utilizado apenas com IF. O comando IF tem três parâmetros. Por exemplo:

```
IF (A, B, C)
```

O primeiro parâmetro é uma condição. Esse método retornará B se a condição A é verdadeira; E ele retorna C, se a condição for falsa. Exemplo:

```
IF(@Tag.A = 1, "Verdadeiro", "Falso")
```

Essa expressão irá retornar uma string de acordo com o valor da Tag.A.

Neste método dot NET, todos os três parâmetros são avaliados independentemente da condição. Exemplo:

```
IF(@Tag.A = 1, @Script.Class.Client.Func1(),
@Script.Class.Client.Func2())
```

Nesse caso, ambos Func1 e Func2 vão sempre ser executados. Apenas o valor de retorno será diferente com base no valor de Tag.A.

O método IF ou IIF precisa avaliar os parâmetros antes de chamar o método. Há muitas situações em que deseja executar apenas a função de acordo com o valor.

Para cálculos mais complexos, pode-se chamar uma classe que você deve antes criar na aba Classes.

Para configurar expressões:

1. Vá para *Edit>Scripts>Expressions*;
2. Digite ou selecione as informações, conforme necessário;
3. Clique em *Verify* para verificar a validade da expressão.

As configurações associadas são:

- *Object*: selecione uma Tag ou objeto existente;
- *Expression*: digite a expressão. A expressão pode ser uma expressão matemática básica, usar uma classe, ou ser uma expressão condicional;
- *Domain*: selecione onde a expressão é executada:
- *Client*: a expressão será executada em cada sistema cliente. Estas são expressões que se aplicam localmente (no computador do usuário), por exemplo, geração de relatórios;
- *Server*: a expressão será executada no servidor do sistema. Estas são expressões que se aplicam em toda a aplicação, ou seja, em nível global;
- *Execution*: selecione quando a expressão é executada:
- *OnChange*: a expressão é executada quando o valor de qualquer Tag na expressão muda;
- *TriggerOrPeriod*: a expressão é executada quando ocorre o evento de disparo configurado ou quando é expirado o intervalo de tempo definido no período;
- *ChangeOrStartup*: a expressão é executada quando o valor de qualquer Tag na expressão muda ou na partida do aplicativo;
- *Trigger*: digite ou selecione uma Tag ou objeto que será usado para disparar a execução da expressão. O disparo da execução ocorrerá quando o valor da Tag ou objeto se alterar.
- *DisableCondition*: digite ou selecione uma Tag ou objeto que será usado para desabilitar a execução da expressão.
- *BuildStatus*: somente leitura. Situação do código atualizado após clicar em *Verify*.
 - Marca Verde: expressão compilada sem erros;
 - Marca em X Vermelha: a Expressão contém erros.
- *BuildErrors*: somente leitura. Exibe todos os erros encontrados durante a última compilação.

ATENÇÃO:

Versões anteriores a 2018, disponibilizavam uma aba denominada *Bindings*. Projetos convertidos para a versão 2018, não terão mais essa aba e os *bindings* configurados serão movimentados para o *Expressions*.

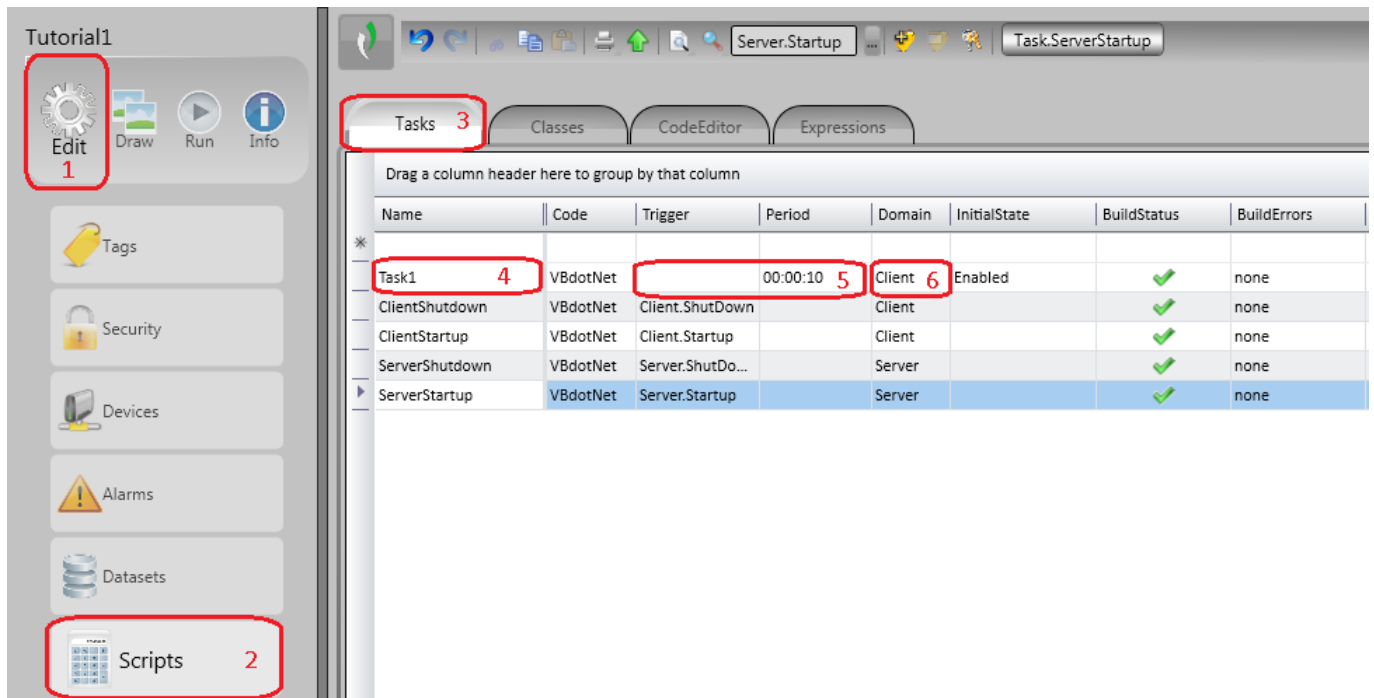
Utilizando Tarefas de Script

Figura 3-31. Tarefas de Scripts

Os passos para editar ou incluir uma tarefa de script estão ilustrados na figura a seguir e descritos na sequência:

1. Selecionar o menu *Edit*;
2. Acessar o item *Scripts*;
3. Clicar na guia *Tasks* para acesso às tarefas de scripts configuradas no sistema;
4. Incluir ou editar o nome da tarefa na linha marcada com um asterisco (*);
5. Configurar a variável de disparo ou o período que a tarefa de script será executada, através das colunas *Trigger* ou *Period*;
6. Selecionar em qual ambiente (cliente ou servidor) a tarefa será executada.

A figura a seguir apresenta os passos para a codificação da tarefa de script chamando o método de uma classe existente.

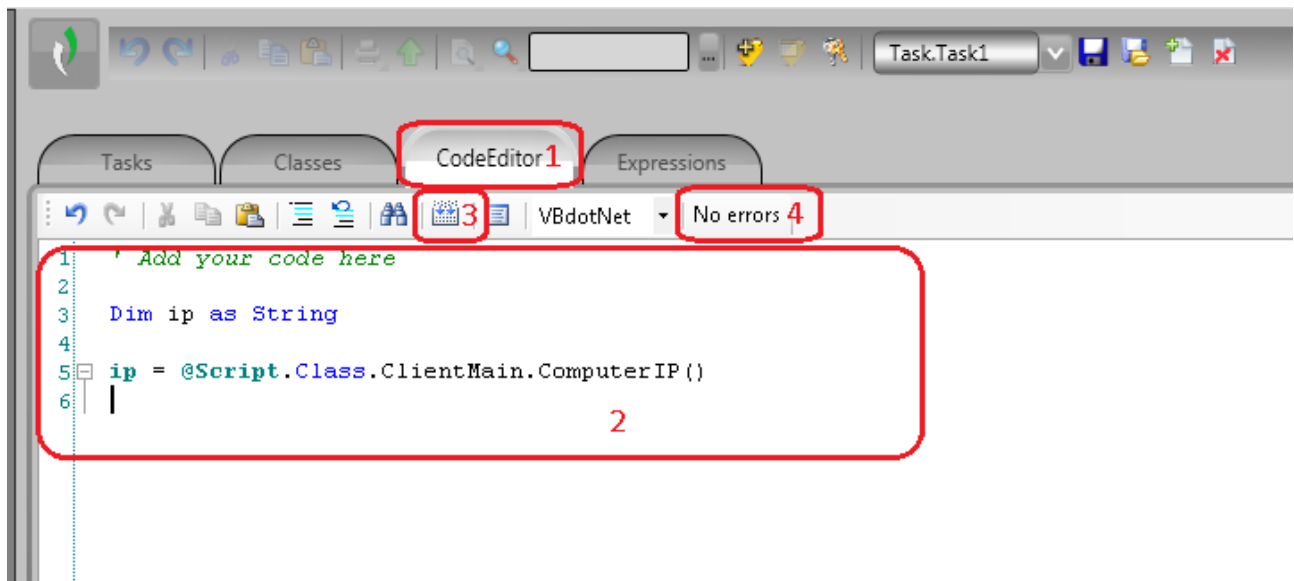


Figura 3-32. CodeEditor das Tarefas de Scripts

Na sequência, deve-se seguir o seguinte procedimento:

1. Clicar na guia *CodeEditor* para acesso ao ambiente de codificação;
2. Digitar o código no espaço demarcado;
3. Clicar no ícone para compilar o código;
4. Verificar a existência de erros no script codificado.

ATENÇÃO:

Para utilizar os *namespaces* no ambiente de codificação é obrigatório digitar o caractere @ antes do *namespace*, caso contrário será utilizado como uma variável local gerando possíveis erros na compilação.

Utilizando o Depurador .NET

O Blueplant tem um depurador dotNET integrado. Para usá-lo, é necessário que o computador local tenha arquivos de cache com os arquivos necessários para executar o depurador. As etapas para ativar o depurador são:

1. Em *Run>Build>Messages* selecionar a opção *Include debug information*;
2. Se necessário, salve novamente o código fonte que você deseja depurar, deste modo as informações de depuração serão criadas. Esta etapa só é necessária na primeira vez que abrir o projeto no computador. Depois disso, a compilação de “background” irá manter a geração das informações de depuração, à medida que se altera o código, para permitir o uso de pontos de interrupção e execução passo a passo.

Anexando o depurador dot NET

De modo a se ter uma sessão de depuração dotNET, o ambiente de Engenharia deve ser conectado ao ambiente do *Runtime* e o depurador dotNET anexado (*Attached*) ao processo do servidor ou processo do cliente.

Siga estes passos.

1. Durante a execução do projeto, seja em *Run>Test* ou *Run>Startup* habilitar a caixa de seleção de conexão;

- Se o projeto já estiver em execução, você pode ir para *Run>Test* ou *Run>Startup*, de acordo com o tipo de execução que você deseja anexar, e conectar-se ao sistema de execução, pressionando o botão de conexão nessas páginas;
- Abra qualquer script que tem informações de depuração e pressione o botão *Attach .NET Debugger*. Uma mensagem na parte inferior da área de trabalho de engenharia vai mostrar que uma sessão de depuração está ativa com os componentes do servidor ou os componentes do cliente do projeto em execução.



Figura 3-33. Anexando depurador .NET

Quando o depurador dotNET está *Anexado* o sistema irá parar nos pontos de interrupção definidos e irá parar automaticamente quando ocorrer qualquer .NET *Exception*.

A figura a seguir mostra o editor de código e os botões para anexar e desanexar depurador, executar, executar passo a passo e criar pontos de interrupção mostrados no retângulo vermelho. Na parte inferior informação de status da depuração.

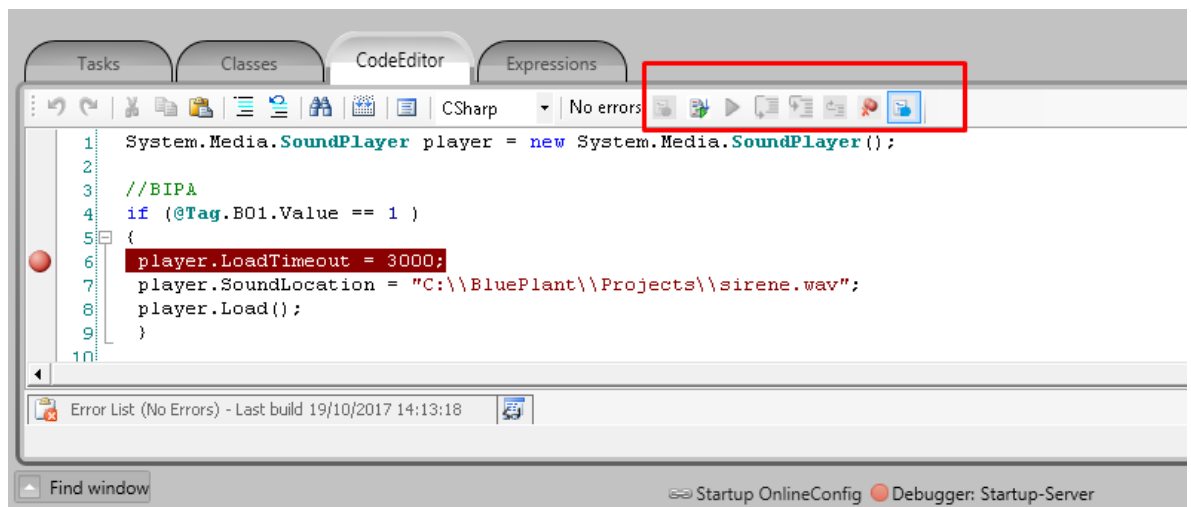


Figura 3-34. Editor de Código os Botões para Anexar e Desanexar depurador

Pontos de Interrupção, Passo a Passo e Watch

Para configurar um ponto de interrupção, abra o código desejado, selecione a linha e pressione o *Insert Breakpoint* na barra de ferramenta ou clique na barra vertical esquerda da janela de código.

Quando o sistema para em um ponto de interrupção, pode-se executar passo a passo pressionando os botões de execução passo a passo ou mandar seguir pressionando o botão continuar.

A fim de inspecionar variáveis locais do dotNET, Tags ou objetos do projeto, pode-se selecionar o texto no editor de script e, quando a execução for interrompida em um ponto de interrupção, o kit de ferramentas mostrará o valor atual da variável.

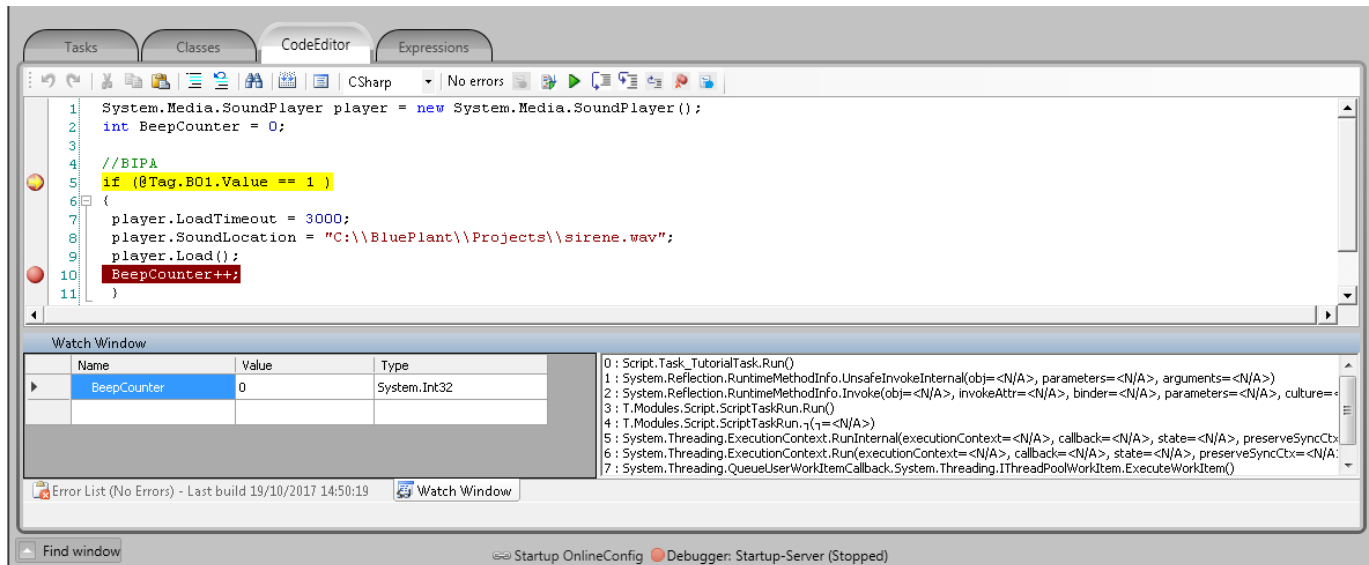


Figura 3-35. Watch para variáveis dotNET

Também pode-se adicionar variáveis locais do dotNET ou objetos do projeto na janela Watch do próprio Code Editor (parte inferior). Ao adicionar Tags ou objetos do projeto, precisa-se usar os símbolos @, exemplo @Tag.tag1, para que o sistema distinga entre objetos de projeto, e variáveis locais do dot NET. Tenha em mente que o Watch é atualizado apenas quando a execução é interrompida. Caso seja necessário ter os valores em tempo real para Tags e objetos pode-se abrir a ferramenta de diagnóstico *PropertWatch*.

Estudo Dirigido 3-2: envio de e-mail com autenticação



Com o auxílio do instrutor configure um script para envio de e-mail com autenticação para uso em diferentes situações: disparo de alarmes, falha de comunicação com PLC e outros a sua escolha.

📌 **DICA:** caso você queira saber mais sobre o envio de e-mail com autenticação no BluePlant, assista o Webinar a seguir. Nele é explicada a configuração de um script para envio de e-mail com autenticação para uso em diferentes situações: disparo de alarmes, falha de comunicação com PLC, entre outros.



Editando Displays



A opção *Displays* do menu *Edit* permite a configuração do módulo associado às telas. Esse módulo contempla o repositório das telas (*Displays*), seus modos de operação (*Layouts*) e imagens (*Resources*).

Displays

A guia *Displays*, quando selecionada, lista as telas que compõem o projeto. A visualização da lista de telas pode ser no formato de tabela ou cartão. O seletor *Card View/Table View* permite alternar o modo de visualização. A figura a seguir ilustra esse ambiente no formato de tabela.

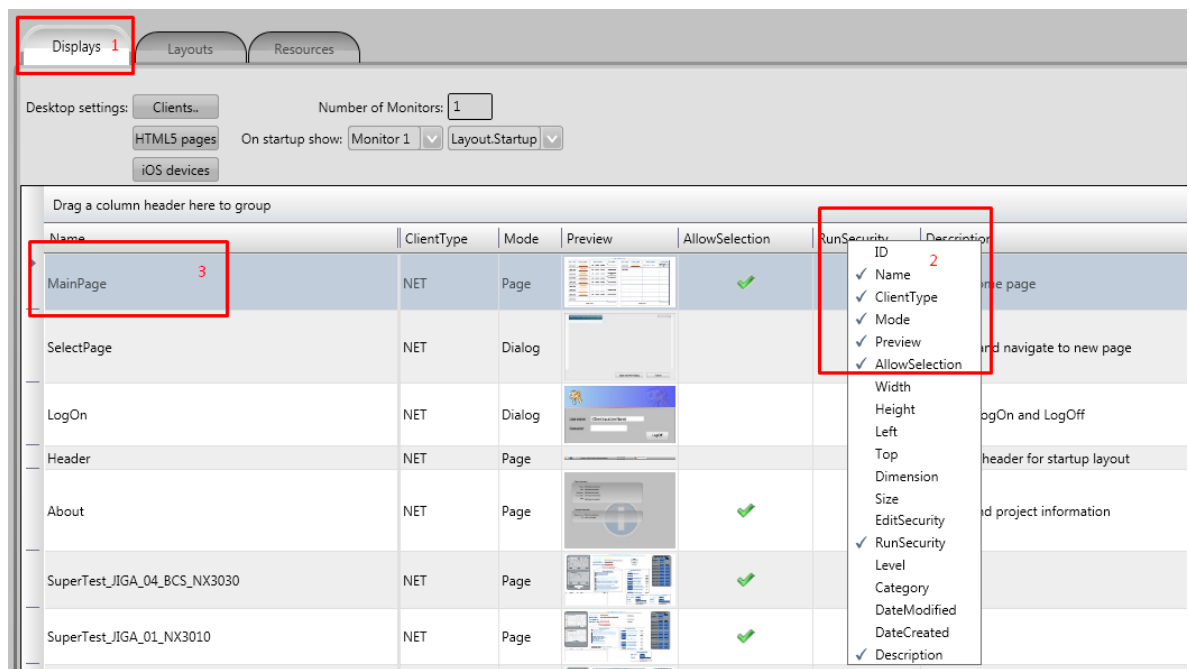


Figura 3-36. Edição de Displays

Os itens que compõem o menu de edição de telas estão numerados na figura anterior e são descritos na sequência:

1. Acessar à guia *Displays*;
2. Clicar com o botão direito do mouse em qualquer um dos cabeçalhos de coluna para selecionar as opções de configuração a serem exibidas;
3. Clicar na linha da tela para alterar as configurações.

Name

Nome da tela.

ClientType

Tipo de cliente que foi desenvolvido a tela. HTML5, iOS ou .Net.

Mode

Modo de operação da tela (Página, Diálogo ou Popup). Essas opções são detalhadas na sequência:

- *Page*: este é o modo de exibição padrão. Quando uma Página é aberta automaticamente, a última página do layout atual é fechada;
- *Dialog*: uma tela de Diálogo se abre como um *Window Modal Dialog*. Isto significa que os comandos em todas as outras telas abertas ficam desabilitados. Ao clicar em OK no diálogo o método On OK na tela CodeBehind é executado;
- *Popup*: uma tela de popup abre-se acima das outras telas. Ao abrir uma nova Página, por padrão, todas as telas de popup são fechadas.

Preview

Visualização da imagem da tela.

AllowSelection

Marca indicando se a tela está listada no objeto *DisplaySelection* quando em execução. *AllowSelection* é uma função utilizada pelo criador da aplicação/projeto para remover a seleção de página, teste e outras telas do operador padrão e assegurar que os itens removidos estejam disponíveis somente para contextos específicos da aplicação. O método incorporado para definir a seleção da tela ao rodar a aplicação é a função *PageSelector* que é configurada na guia *Editar>Telas>Telas*. Para impedir essa seleção pelo operador padrão remova o marcador da página ou da opção *AllowSelection* na coluna dos objetos.

EditSecurity

Define quais os tipos de usuários têm acesso à edição desta tela.

RunSecurity

Grupos de permissões de segurança exigidos para abrir esta tela durante o *Runtime*. Clique no campo *RunSecurity* para abrir a janela *Run Security Selection*. Após concluir as seleções clique em qualquer lugar na tabela para fechar a janela de seleção. Suas seleções aparecerão no campo *RunSecurity* da linha da tabela.

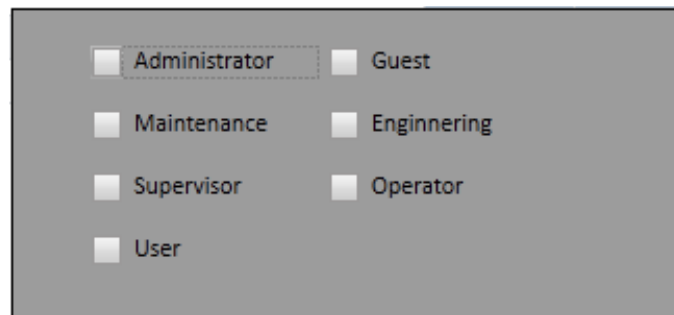


Figura 3-37. Permissões no *Runtime*

Configurando Múltiplos Monitores

Os passos para configurar múltiplos monitores estão ilustrados na figura a seguir e descritos mais adiante.

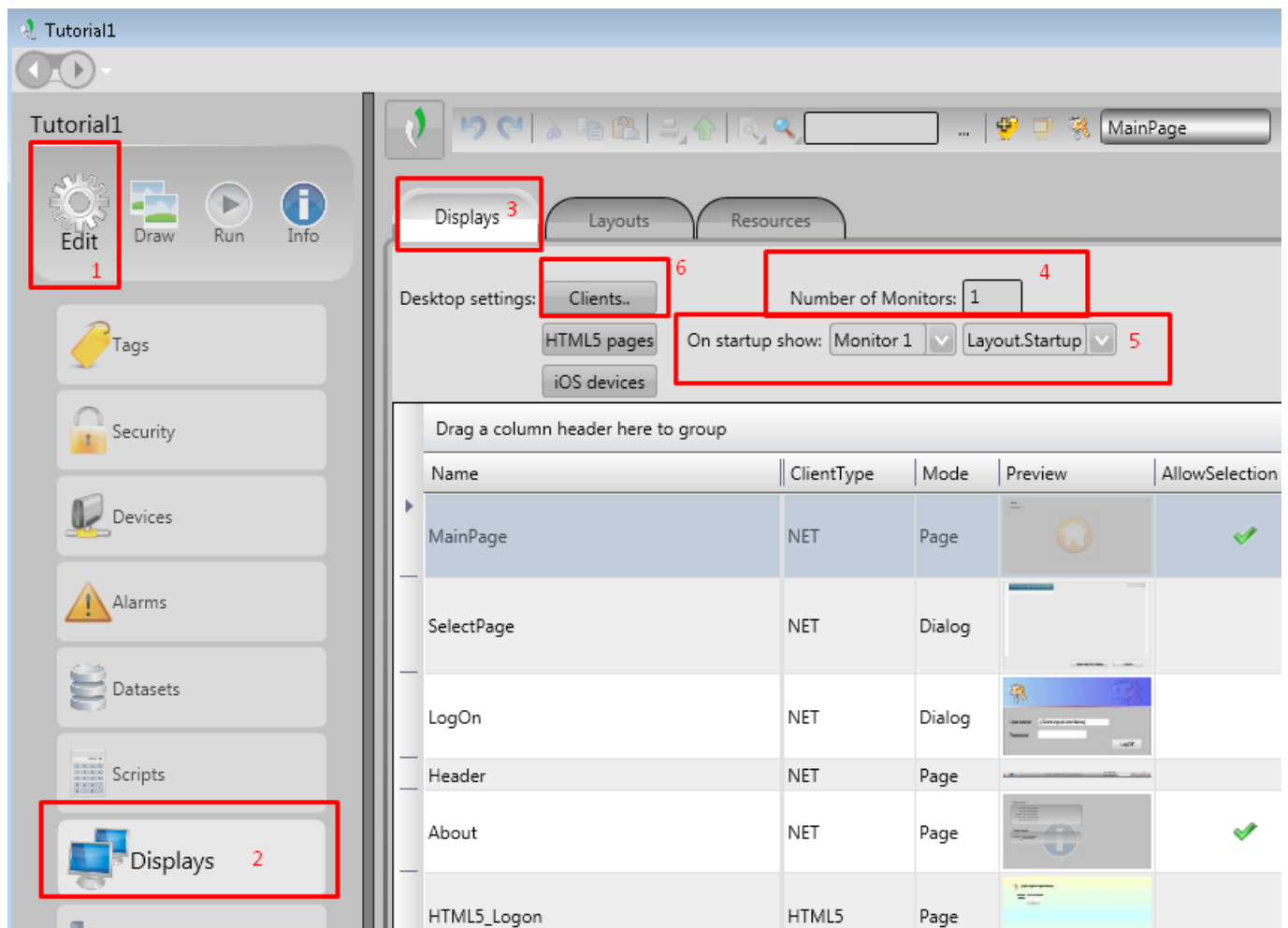


Figura 3-38. Configurando Múltiplos Monitores

1. Selecionar o menu *Edit*;
2. Acessar o item *Displays*;
3. Clicar na guia *Displays* para acesso às configurações das telas já existentes no projeto;
4. Configurar a quantidade de monitores que deseja utilizar;
5. Selecionar para cada monitor o layout correspondente que irá abrir quando executar o projeto;
6. Clicar no botão *Clients* a realizar a configuração dos Clientes *rich* no *Runtime* (uma configuração única para todos os clientes *rich*).

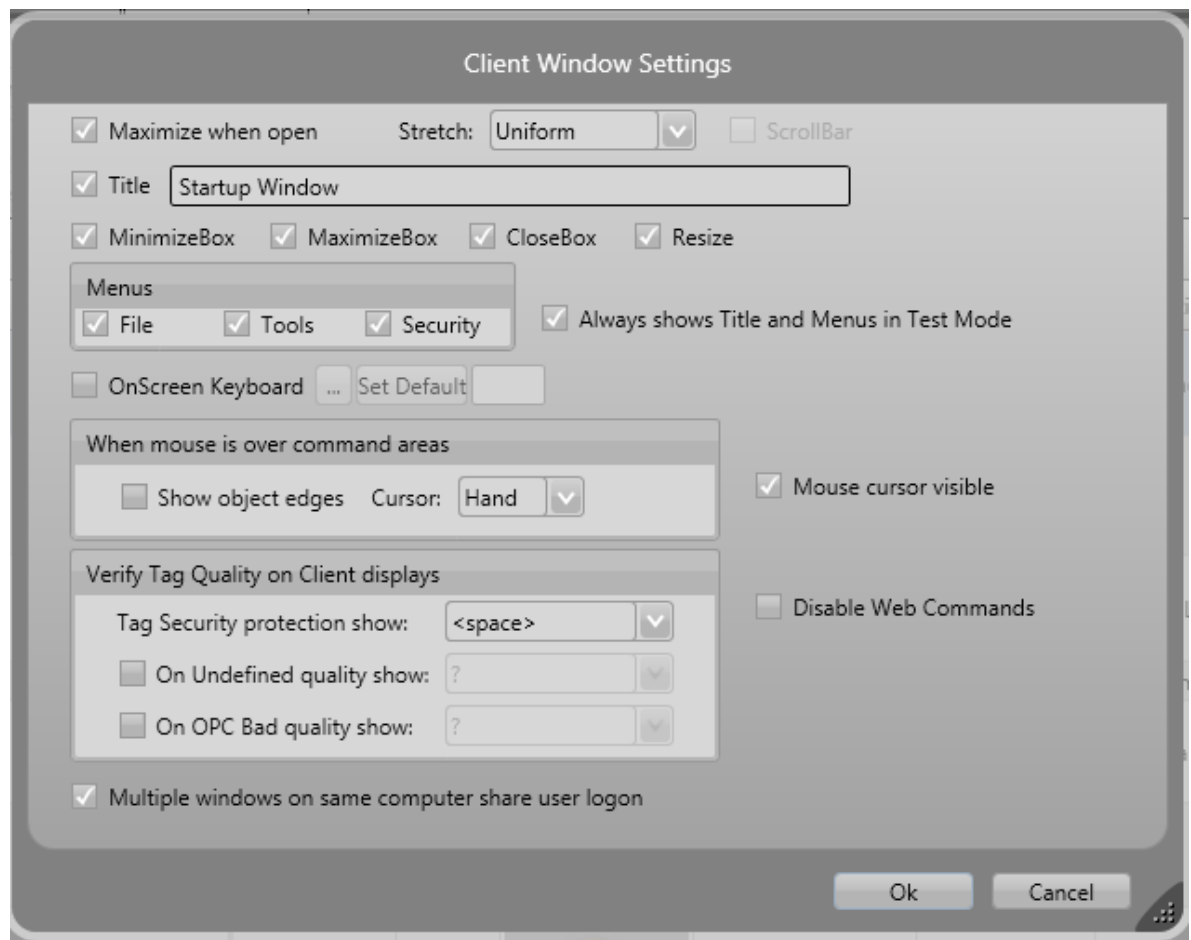


Figura 3-39. Configurações dos Clientes Rich





As configurações de *Clientes Rich* disponíveis são:

- **Maximize when open:** quando selecionado maximiza a janela do cliente para se ajustar ao monitor.
- **Stretch:** selecionar como o Layout vai preencher o monitor Cliente conforme indicado abaixo:
 - *None:* Layout não deve se redimensionar: Deve ser mostrado o tamanho exato que foi configurado para o layout. Neste caso recomenda-se selecionar a opção de incluir a barra de rolagem;
 - *Fill:* O layout será redimensionado para preencher completamente o espaço disponível no monitor do cliente., sem levar em conta a proporção do layout original;
 - *Uniform:* O layout será redimensionado proporcionalmente ao tamanho do monitor do cliente, mantendo a proporção do layout. Original;
 - *UniformToFill:* O layout redimensionado tanto para manter a proporção como também para preencher completamente o espaço disponível no monitor do cliente. Isto pode resultar em cortar algumas partes da disposição;
- **Scrollbar:** selecione para visualização da barra de rolagem na janela do cliente;
- **Title:** digite um título para a janela do cliente;
- **MinimizeBox:** selecione para exibir o botão padrão de minimizar;
- **MaximizeBox:** selecione para exibir o botão padrão de maximização;
- **CloseBox:** selecione para exibir o botão padrão de fechar;

- **ResizeBox:** selecione para exibir o puxador padrão de redimensionamento (canto inferior direito);
- **Menus:** Selecione quais menus devem ser mostrados (File, Tools e/ou Security);
- **OnScreen Keyboard:** Quando selecionado exibe um teclado numérico na tela.
- **Mouse Cursor Visible:** Quando selecionado o ponteiro do mouse se mantém visível.
- **Disable Web Commands:** Quando selecionado proíbe usuários do aplicativo de enviar comandos para o PLC
- **When Mouse Is Over Command Areas:** Selecione o que mostrar quando o mouse está sobre um comando (mostrar bordas do objeto) e cursor: altera o tipo de cursor.
- **Verify Tag Quality on Client Displays:**
 - **On undefined quality show:** Selecionar a opção e o caractere que deve ser mostrado quando a qualidade dos dados vindos do campo for indefinida.
 - **On OPC Bad quality show:** Selecionar a opção e o caractere que deve ser mostrado quando o servidor OPC indica que a qualidade dos dados é ruim.
- **Multiple Windows on same computer share user logon:** os usuários podem executar múltiplas instâncias do aplicativo no mesmo computador. Como o usuário pode precisar logar como um *User* diferente de vez em quando, se deve selecionar essa opção para alterar automaticamente o usuário conectado em todas as instâncias em execução do aplicativo no mesmo computador.

Layouts

Edita os layouts do projeto. Layouts são conjuntos de páginas (*DockPanels*) que se encaixam para definir o modo da tela durante a execução. A última tela listada no layout é aquela que será alterada sob o comando *Client.OpenDisplay()*. As demais que permanecem abertas normalmente contêm informações tais como menus de navegação, linhas de alarme e informações de aplicação global. O projeto pode ter apenas um layout, ou pode ser alterado dinamicamente através do comando *Client.OpenLayout()*.

Clique  para criar um novo layout. Clique  ou  para acrescentar ou excluir páginas no layout selecionado e  para navegar para cima e para baixo na lista da página. As dimensões do layout podem ser definidas usando os campos *Width* e *Height*, podendo alterar o fundo para uma cor ou imagem.

A figura a seguir ilustra a configuração dos layouts.

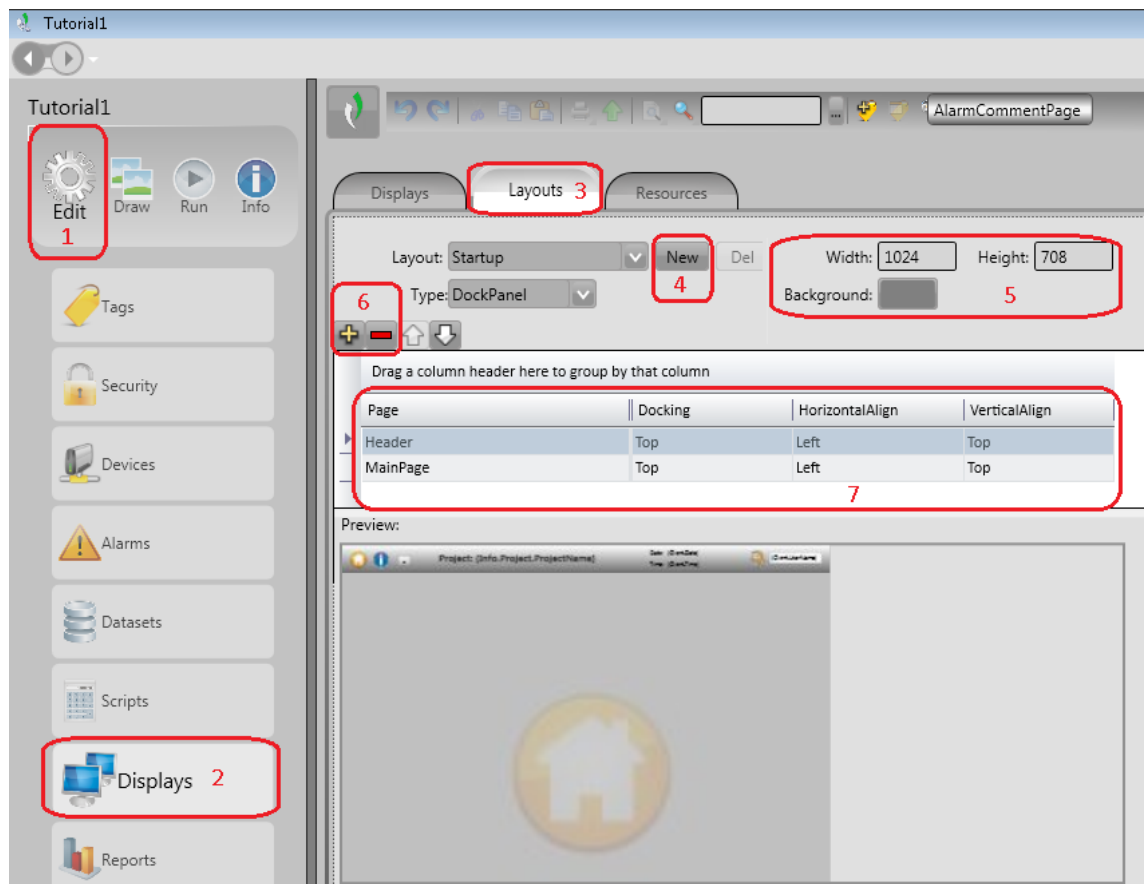



Figura 3-40. Edição de Layouts

Os passos para editar e incluir um novo layout estão ilustrados na figura anterior e descritos a seguir:

1. Selecionar o menu *Edit*;
2. Acessar o item *Displays*;
3. Clicar na guia *Layouts* para acesso as configurações dos layouts já existentes no sistema;
4. Clicar no botão *New* para adicionar um novo layout;
5. Configurar as dimensões e cor de fundo do layout correspondente;
6. Configurar a quantidade de telas que o layout irá apresentar utilizando os botões ;
7. Selecionar as telas e suas posições no layout.

Page

Nome da tela usada neste Layout. Somente telas do modo PÁGINA podem ser incluídas em layouts.

Docking

Define a posição de encaixe da tela. As opções possíveis são: Esquerda, Topo, Direita e Inferior. A posição de encaixe pode ser alterada clicando-se uma vez no campo da tabela e selecionando-se a posição desejada.

HorizontalAling

Define o alinhamento horizontal. As opções possíveis são: Esquerda, Centro e Direita.

VerticalAling

Define o alinhamento vertical. As opções possíveis são: Topo, Centro e Inferior.

Margins

Define as margens da tela dentro do DockPanel. Todas as medidas da tela no projeto (tamanho, largura etc.) são em unidades WPF (Windows Presentation Foundation).

Resources

Esta guia lista os recursos das telas. Para habilitar uma execução distribuída para as telas em *Runtime* e Clientes Web (ao usar imagens em telas e relatórios), importe a(s) imagem(ns) para os recursos do projeto através de `Import images...` em vez de selecionar um nome de arquivo. Um benefício adicional da guia Recursos é que quando a imagem é substituída no ResourceDictionary (o Nome do Recurso não é alterado) todas as referências a este recurso nas Telas e Relatórios são automaticamente atualizadas. A figura a seguir ilustra as três etapas do menu de edição de recursos.

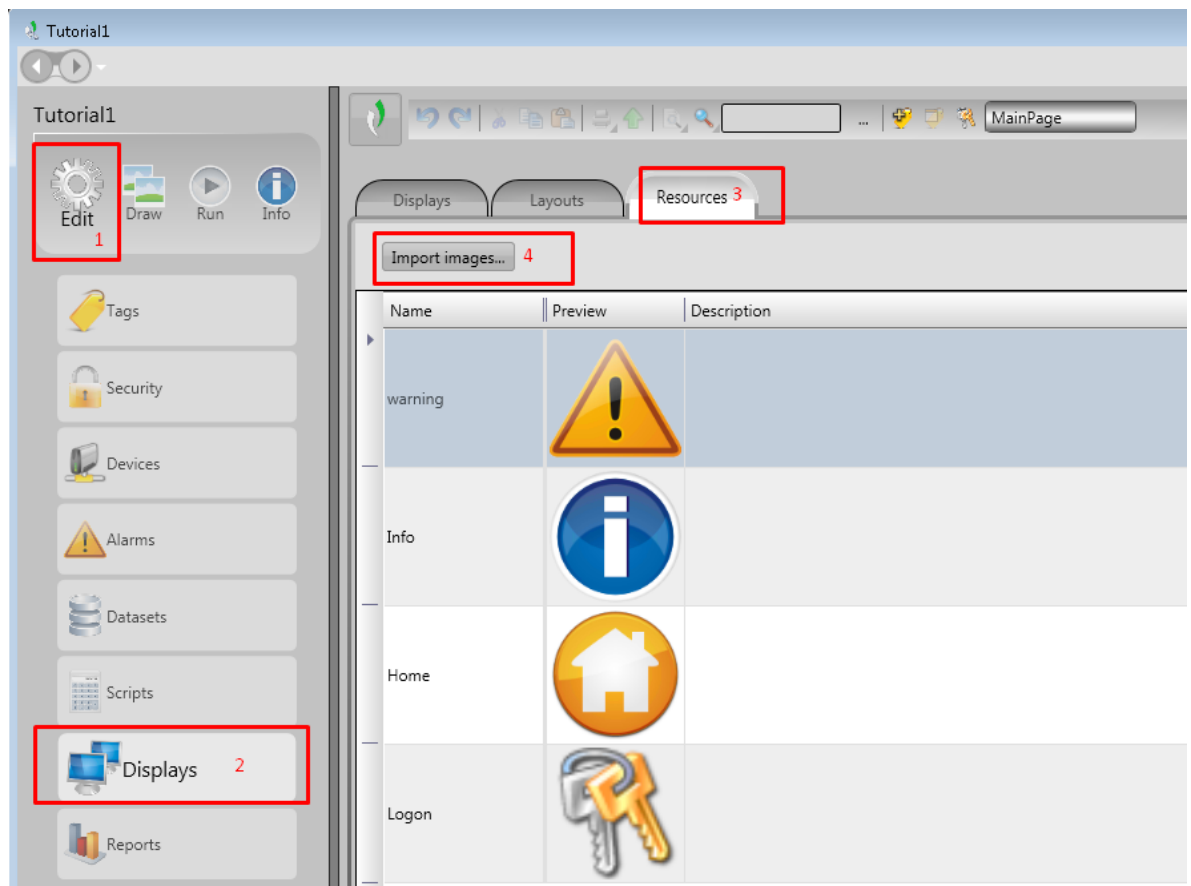


Figura 3-41. Edição de Recursos

Os passos para importar novas imagens estão ilustrados na figura anterior e são descritos a seguir:

1. Selecionar o menu *Edit*;
2. Acessar o item *Displays*;
3. Clicar na guia *Resources* para acesso a todas as imagens já importadas para o projeto;
4. Clicar no botão *Import Images...* para selecionar e importar a imagem.

Name

Nome para o objeto Recurso. Clique no campo da tabela para alterar o nome do objeto.

Preview

Visualização da imagem do Recurso.

Editando Relatórios



O editor de relatório permite a inclusão de texto dinâmico, símbolo dinâmico, gráfico, conjunto de dados e consulta de resultados em um editor completo e de fácil utilização. A configuração da guia *Reports* está ilustrada na figura a seguir.

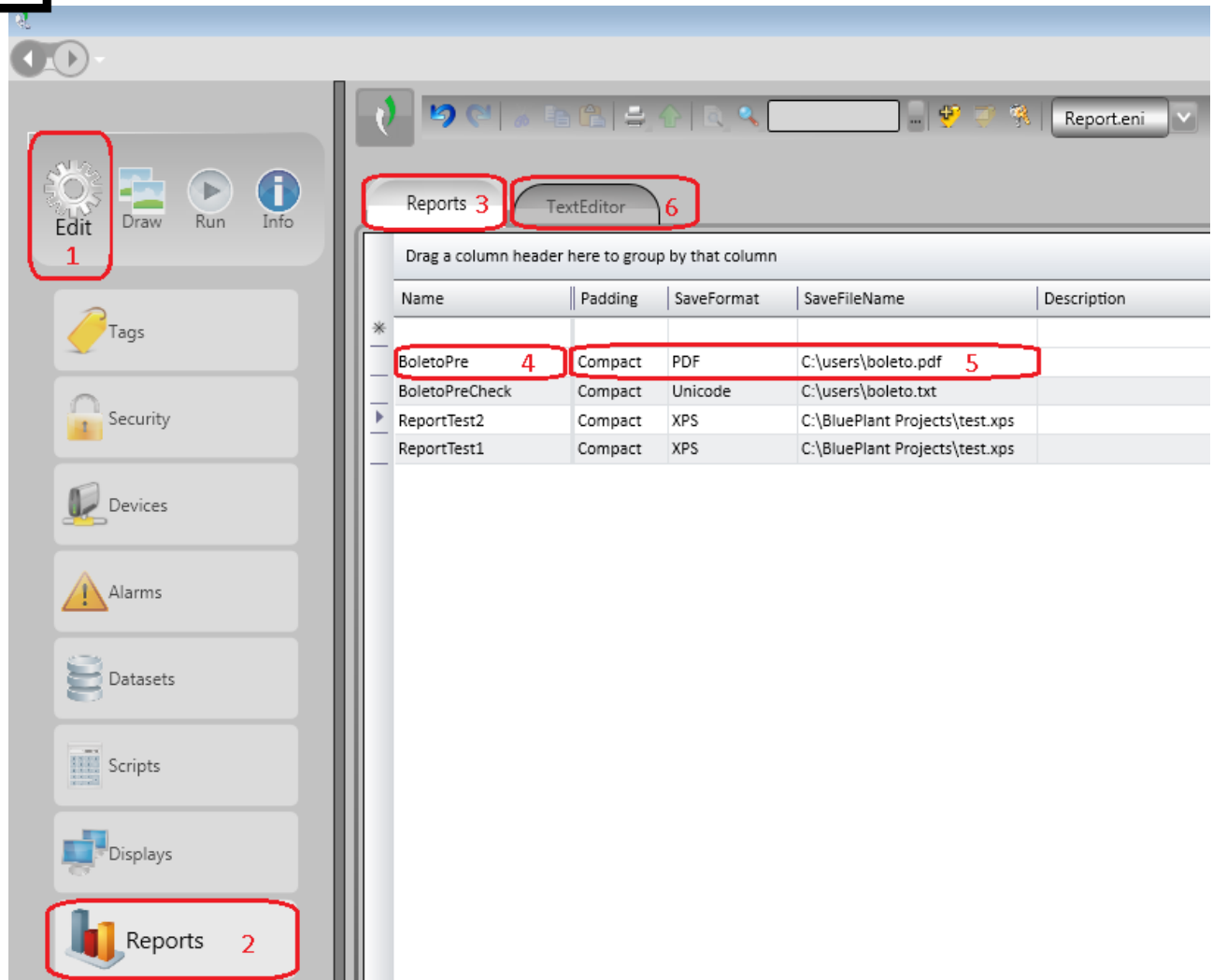


Figura 3-42. Configuração de Relatórios

Os passos para editar ou incluir um novo relatório estão ilustrados na figura anterior e são descritos a seguir:

1. Selecionar o menu *Edit*;
2. Acessar o item *Reports*;
3. Clicar na guia *Reports* para acesso a todos os relatórios previamente criados no projeto;
4. Incluir ou editar o nome do relatório na linha marcada com um asterisco (*);
5. Configurar as características do relatório como alinhamento dos dados, formato e nome do arquivo no qual será gerado o relatório;
6. Clicar na guia *TextEditor* para editar o formato do relatório. A figura a seguir detalha a formatação.

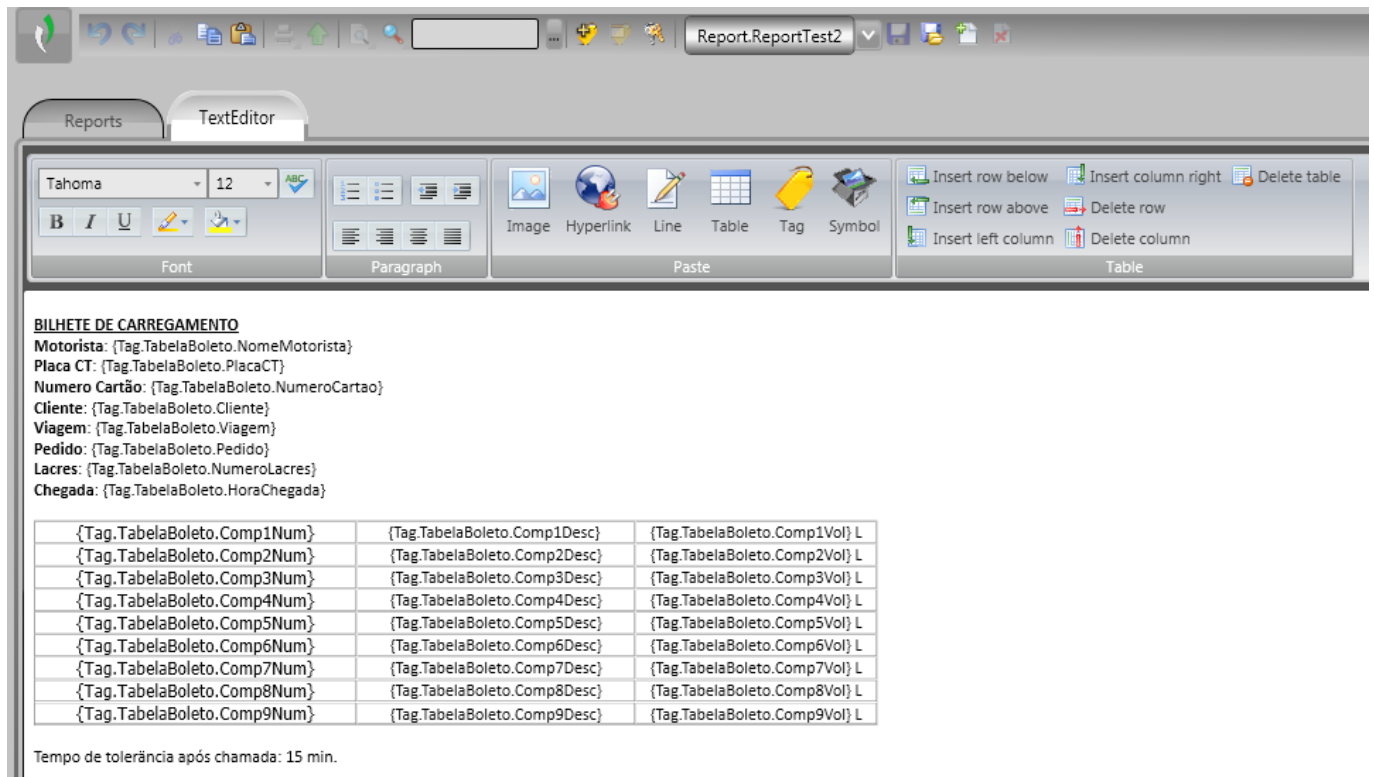


Figura 3-43. Edição de Relatório

Name

Indica o nome do relatório a ser gerado.

Padding

Define os ajustes ao substituir uma *TagName* por seus valores. As opções possíveis são: *Compact* (compacto), *PadRight* (ajuste à direita) e *PadLeft* (ajuste à esquerda). O ajuste usa o número de valor exato configurado no modelo do relatório ao criá-lo no *Runtime*. O valor do alinhamento à esquerda ou à direita dentro do espaço está também incluído.

SaveFormat

Define o formato do arquivo usado ao salvar o relatório. As opções possíveis são: XPS, HTML, Unicode, PDF e ASCII.

Save File Name

Define o nome do arquivo usado e onde será salvo o relatório. Este campo pode incluir os valores das Tags avaliados quando da geração de um alarme, por exemplo. Para tanto, acrescente o nome da Tag nesta célula da tabela usando a notação de associação XAML entre chaves. Exemplo:

C:\MyReport.XPS

C:\MyReport-Month{Server.Month}-Day{Server.Day}.XPS

TextEditor

Esta guia contempla um editor de texto para criação de relatórios conforme ilustrado na figura a seguir.

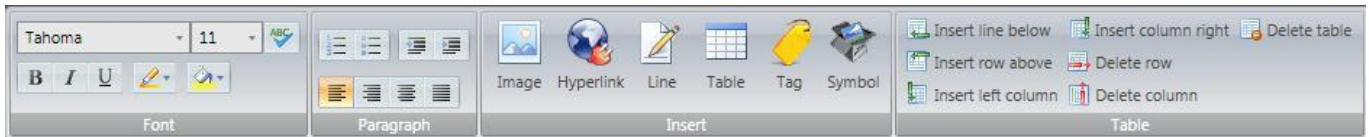


Figura 3-44. Editor de Texto

A figura a seguir mostra a edição e salvamento de um relatório com o auxílio do editor de texto embutido no software.

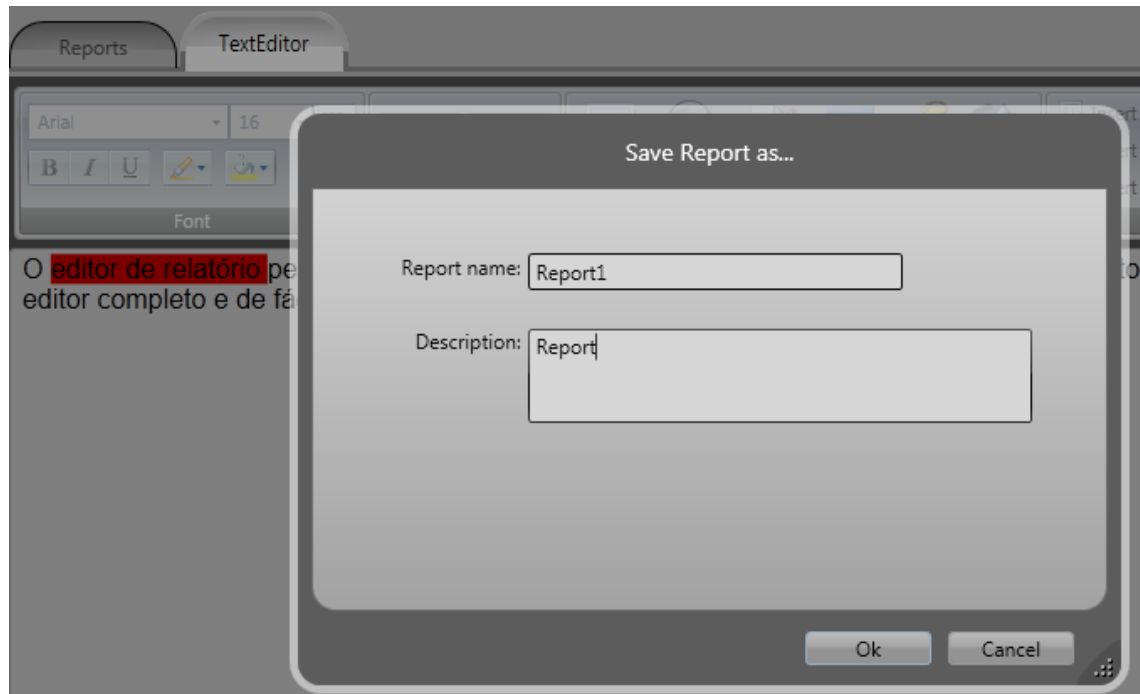


Figura 3-45. Edição e Salvamento no Editor de Texto

A figura a seguir mostra a consolidação do formulário editado na figura anterior com algumas configurações disponíveis no menu de edição de relatórios.

 A screenshot of the Reports menu. It shows a table configuration for a report. The table has five columns: Name, Padding, SaveFormat, SaveFileName, and Description. The first row is highlighted with a blue background and contains the values: Report1, PadLeft, PDF, Test_Report, and Report. Above the table, there is a text instruction: "Drag a column header here to group by that column."

Name	Padding	SaveFormat	SaveFileName	Description
Report1	PadLeft	PDF	Test_Report	Report

Figura 3-46. Configuração de um Relatório

Estudo Dirigido 3-3: edição da aplicação

Parte 1: planejamento da base de dados do processo



A partir do exposto anteriormente sobre o processo de Mistura e Secagem liste as variáveis relevantes (também chamadas de Tags) que devem fazer parte da aplicação BluePlant classificando-as como analógicas, digitais ou textuais (STRING), por exemplo. Considere que estas variáveis podem ser do tipo DEVICE (dados relacionados ao CP), DDE (dados provenientes de outro aplicativo e/ou computador) e MEMORY (dados locais no terminal de operação). Considere ainda os níveis mínimos de acesso que o operador deve ter a fim de estar habilitado a manipular o conteúdo da variável. Se desejado, uma tabela, conforme mostrado abaixo pode ser elaborada previamente à edição das Tags na aplicação.

Nome da variável	Descrição	Classificação (DEVICE, DDE ou MEMORY)	Tipo (Analogica, Digital, Textual...)	Segurança (Operação ou Manutenção)



Dica: Quando se planeja a base de dados da aplicação é interessante escolher para apresentação somente os dados essenciais, de maneira que o sistema de supervisão se torne conciso. Reflita sobre isso com o instrutor e a turma!

Parte 2: funcionalidades da aplicação

Seguindo as orientações do Instrutor acrescente à base de dados (Tags) existente na aplicação Mistura e Secagem outras funcionalidades de edição conforme mostrado na tabela a seguir.

Funcionalidade de Edição	Requisitos específicos
Segurança	...
Dispositivos	...
Alarmes	...
Conjuntos de dados	...
Scripts	...
Displays	...
Relatórios	...

Parte 3: definição das Tags da aplicação

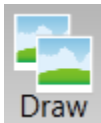
Seguindo as orientações do Instrutor defina as Tags principais da aplicação Mistura e Secagem e suas configurações baseado na análise realizada anteriormente.

Drag a column header here to group					
Name	Type	Array	Parameters	StartValue	Description
* DashboardPageTitle	Text				
SimulationStopConfirm	Integer				



4. Desenhando a Aplicação

O ambiente de telas fornece acesso a todas as ferramentas de diagramação para compilar as telas da aplicação. Este ambiente inclui as seguintes opções: tela, códigos e símbolos.



Menu Draw

A opção *Draw* permite o desenho de *Telas* e *Símbolos*. Use a barra de ferramentas vertical na tela para selecionar um componente ou use a ferramenta de seleção para mover, agrupar e redimensionar componentes. A figura a seguir mostra a barra de ferramentas vertical.

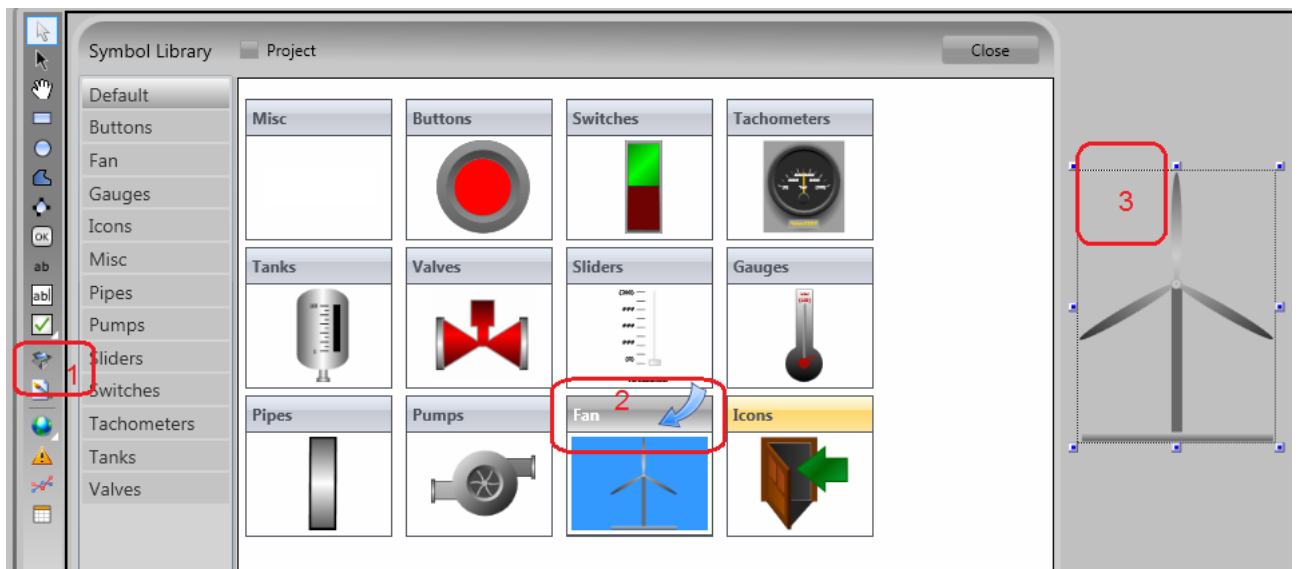


Figura 4-1. Menu Draw e Barra de Ferramentas Vertical

Os passos para a inclusão de um símbolo na tela estão numerados na figura anterior e são descritos na sequência:

1. Selecionar o símbolo a ser incluído, clicando uma vez no desenho correspondente na barra de ferramentas vertical. O símbolo será destacado;
2. Clicar e arrastar o símbolo para a tela;
3. Soltar o botão esquerdo do mouse para finalizar a inclusão do símbolo.

Ferramentas, Objetos e Componentes, Símbolos e Controles



A barra de ferramentas horizontal (localizada na parte inferior da tela) contém comandos para agrupar, combinar, alinhar e bloquear o(s) símbolo(s) selecionado(s). A seguir será descrito o uso da barra de ferramenta horizontal, bem como alguns comandos do menu de contexto conforme ilustra a figura a seguir.

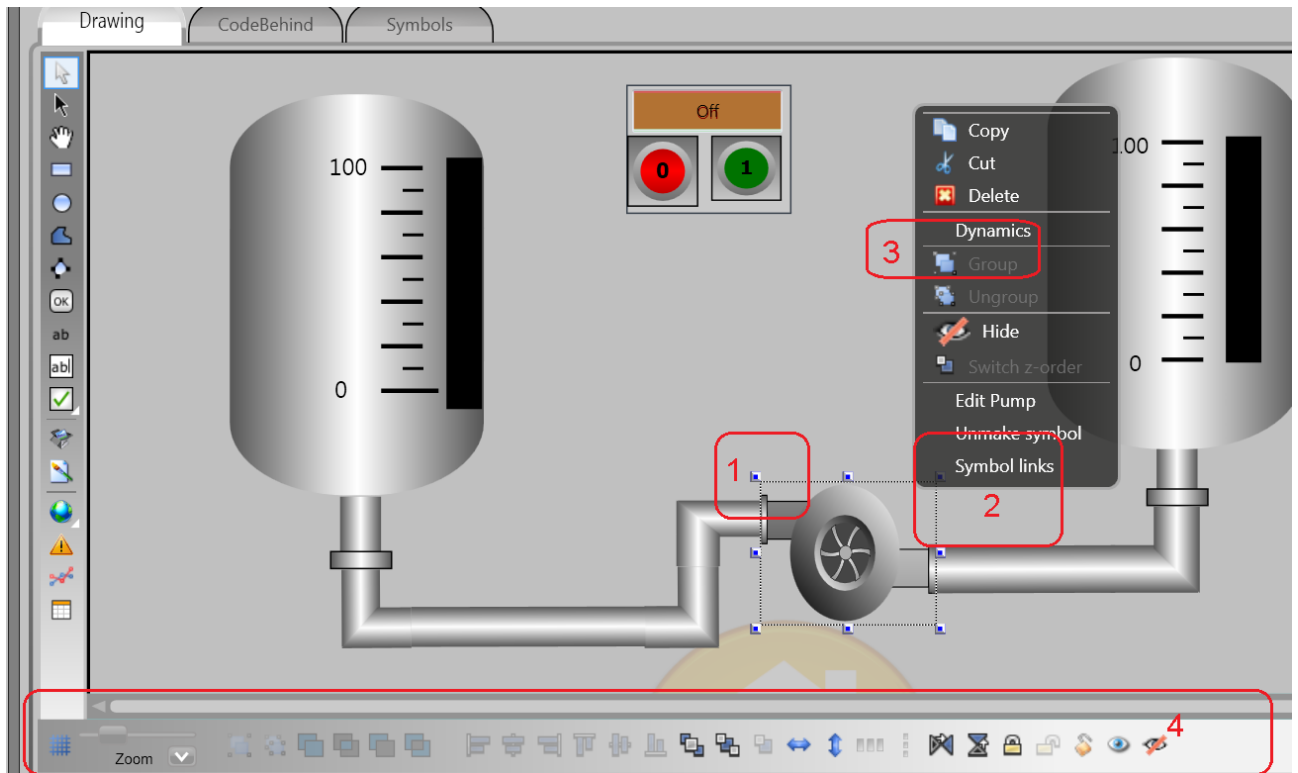


Figura 4-2. Menu Draw e Barra de Ferramenta Horizontal

A sequência de passos para a configuração das propriedades de um símbolo é:

1. Selecionar símbolo com o botão esquerdo do mouse. Para selecionar múltiplos símbolos use SHIFT+CLICAR COM BOTÃO ESQUERDO em cada componente desejado;
2. Clicar no símbolo com o botão direito do mouse para abrir o menu de contexto;
3. Selecionar a opção desejada no menu de contexto;
4. Aplicar as propriedades desejadas para o(s) símbolo(s) usando a barra de ferramentas.

Os itens que compõem a Barra de Ferramentas Vertical ilustrada na figura a seguir estão detalhados na sequência.

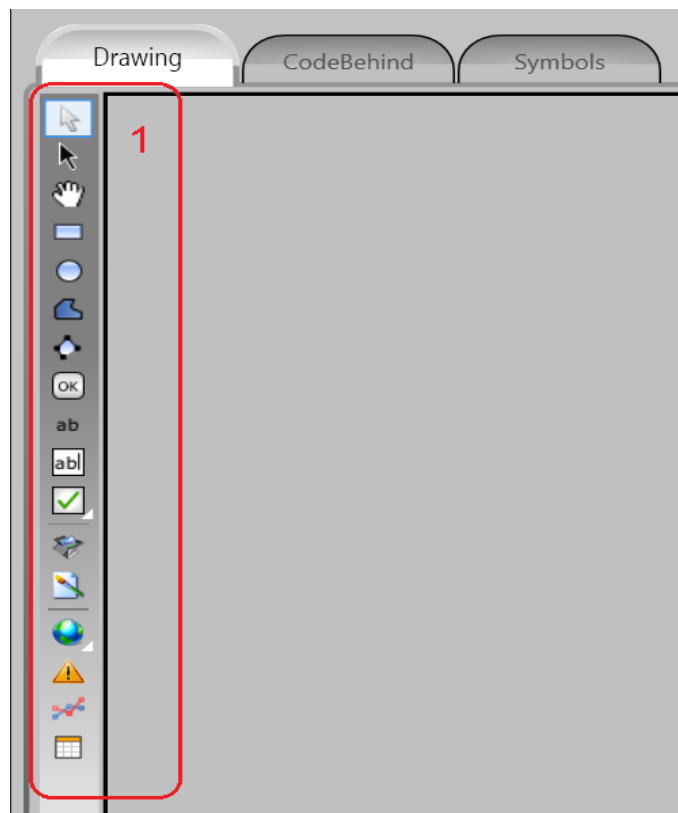



Figura 4-3. Barra de Ferramentas Vertical

Ferramentas de Seleção

Use as ferramentas de seleção para selecionar um objeto e modificar a área de visualização do desenho.

 Ferramenta de Seleção

Clique uma vez no objeto para selecioná-lo. A tecla CTRL+clique do mouse permite a seleção de vários objetos ao mesmo tempo, assim como grupos de objetos. Mantenha pressionada a tecla CTRL enquanto clica em cada objeto. Para alternar a visualização entre os vários objetos selecionados utilize a tecla SHIFT+clique do mouse. Clique em uma área aberta da tela e selecione um grupo de elementos, destacando os elementos desejados enquanto mantém pressionado o botão esquerdo do mouse. Clique duas vezes em um objeto para abrir a janela de configuração dinâmica que fornece as configurações para as propriedades do objeto dinâmico.


 Ferramenta de Seleção Direta


Use esta ferramenta para selecionar um objeto dentro de um grupo e modificar suas propriedades. Clique uma vez no objeto para selecioná-lo. É possível adicionar, remover e modificar os pontos em uma Polyline através desta ferramenta. Para mover o ponto, selecione-o com um clique e pressione o botão esquerdo do mouse. Arraste o ponto para a sua nova posição. Clique duas vezes em um ponto para adicionar um novo ponto adjacente ao ponto selecionado. Clique no botão direito do mouse em um ponto para excluir o ponto selecionado.


 Ferramenta Mão


Use a ferramenta de mão para modificar a janela de exibição. Clique uma vez na tela de fundo e, mantendo pressionado o botão esquerdo do mouse, mude a tela para a posição desejada.

Objetos Geométricos

 Cria um objeto Retângulo.

 Cria um objeto Elipse.


 Cria um objeto Polígono.

 Cria um objeto Polyline.


Clique com o botão direito do mouse para finalizar o uso de cada ferramenta.

Componentes de Tela

Estes objetos criam componentes de telas.

 Cria um objeto Botão.

Dê um clique com o botão direito no ícone do componente para acessar as ferramentas em um menu popup horizontal. Após a ferramenta ter sido selecionada no menu horizontal, ela torna-se a ferramenta padrão para aquela posição na barra vertical.


 Cria um objeto Botão com *LabelBox*.

 Cria um objeto *Checkbox*.


Dê um clique com o botão direito no ícone do componente para acessar as ferramentas em um menu popup horizontal. Após a ferramenta ter sido selecionada no menu horizontal, ela torna-se a ferramenta padrão para aquela posição na barra vertical.

 Cria um objeto *RadioButton*.

 Cria uma *ComboBox*.


 Cria uma *ListBox*.

 Cria uma *PasswordBox*.

 Cria um *MenuItem*.

 Cria um Controle *DatePicker*.

 Cria uma *DateTimeTextBox*.

 Cria um *Slider Control*

Entrada e Saída de Texto

Use as ferramentas de texto para criar objetos de entrada ou saída de texto.

ab Cria um objeto *TextOutput* (*TextBlock*) conforme ilustrado na figura a seguir.

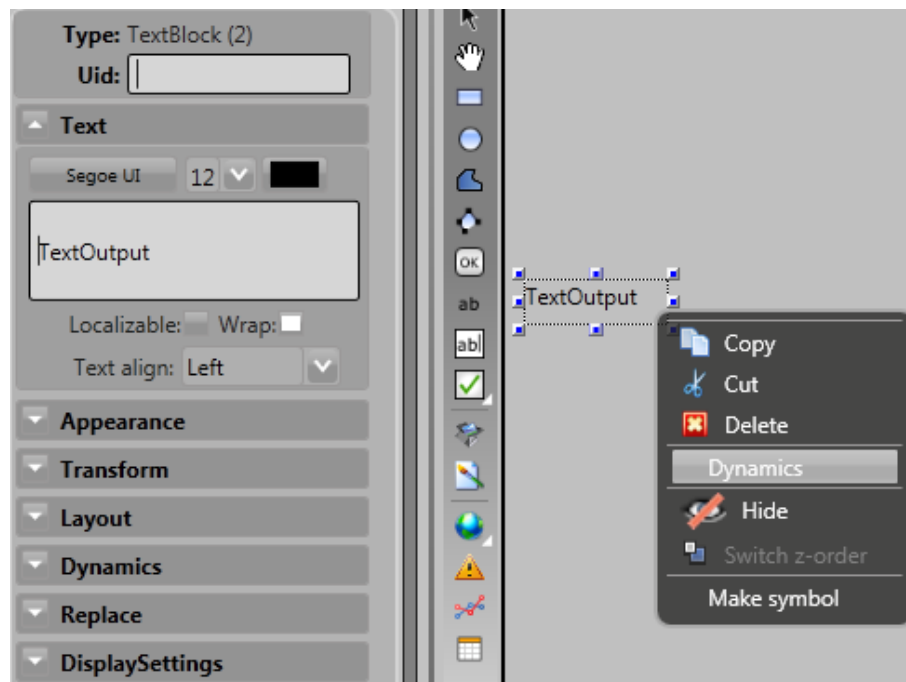


Figura 4-4. Objeto TextOutput

Um clique com o botão direito do mouse no objeto permite acessar o menu suspenso de edição associado à saída de texto, incluindo sua configuração dinâmica que também pode ser acessada através de um duplo clique no objeto.

abl Cria um objeto *TextIO* (*TextBox object*) conforme mostrado na figura a seguir.

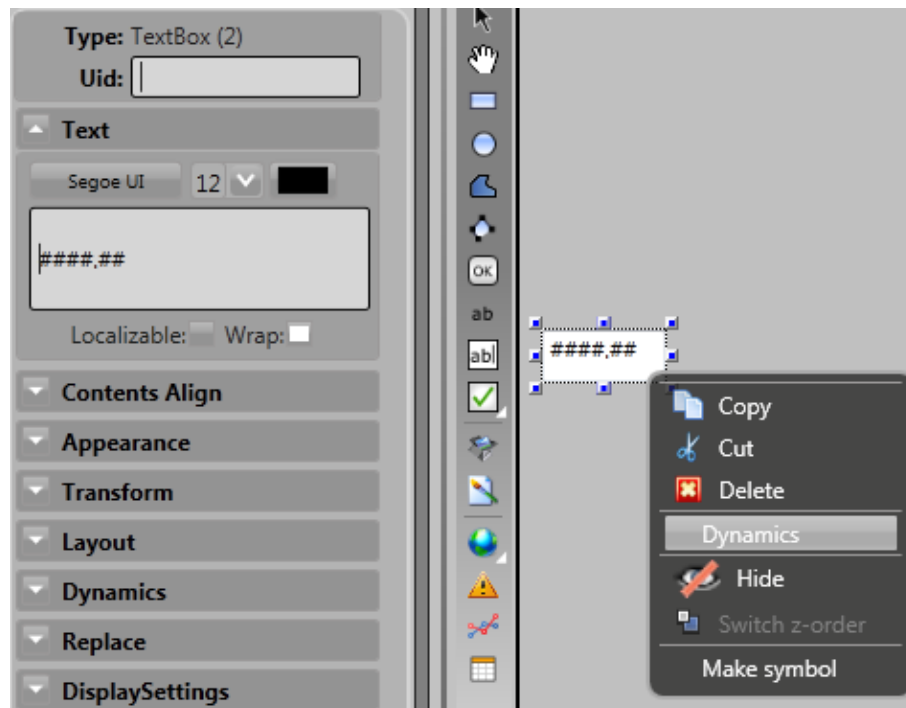


Figura 4-5. Objeto TextIO

Um clique com o botão direito do mouse no objeto permite acessar o menu suspenso de edição associado à entrada/saída de texto, incluindo sua configuração dinâmica que também pode ser acessada através de um duplo clique no objeto. Por exemplo, para vincular o *TextIO* com a Tag,

dê um clique duplo no objeto *TextIO* e, na janela de configuração dinâmica, escolha a dinâmica *TextIO*.

Biblioteca de Símbolos

Abre a biblioteca de símbolos.



Esta biblioteca inclui símbolos incorporados e definidos pelo usuário.

Esta biblioteca extra com símbolos.

Controles Avançados



Cria um objeto *WebBrowser*.

Dê um clique com o botão direito no ícone do componente para acessar as ferramentas em um menu popup horizontal. Após a ferramenta ter sido selecionada no menu horizontal, ela torna-se a ferramenta padrão para aquela posição na barra vertical.



Cria um objeto *ReportViewer*.



Cria um objeto *XpsViewer*.

Cria um objeto *PdfViewer*.



Cria um objeto *ChildDisplay Component*. Este objeto não permite que a mesma tela seja aberta mais de uma vez.



Inclui um componente ( *PieChart*, *MapControl*, *Report Preview*, *Circular Panel*,



Bar Chart, *Advanced PieChart*,  *Calculator* ou componente WPF)



Cria um objeto *PageSelector*.



Cria um componente *Assets*



Cria um componente *Assets Remote* (utilizado apenas para *PI Historian*)

Page Selector

O objeto *Page Selector* vai fazer uma varredura nos displays disponíveis e apresentá-los em uma lista. Para configurar a troca de display: dê um duplo clique sobre o objeto, aparecerá uma tela de configuração, selecione a opção *Action* e configure a opção *OpenDisplay*.

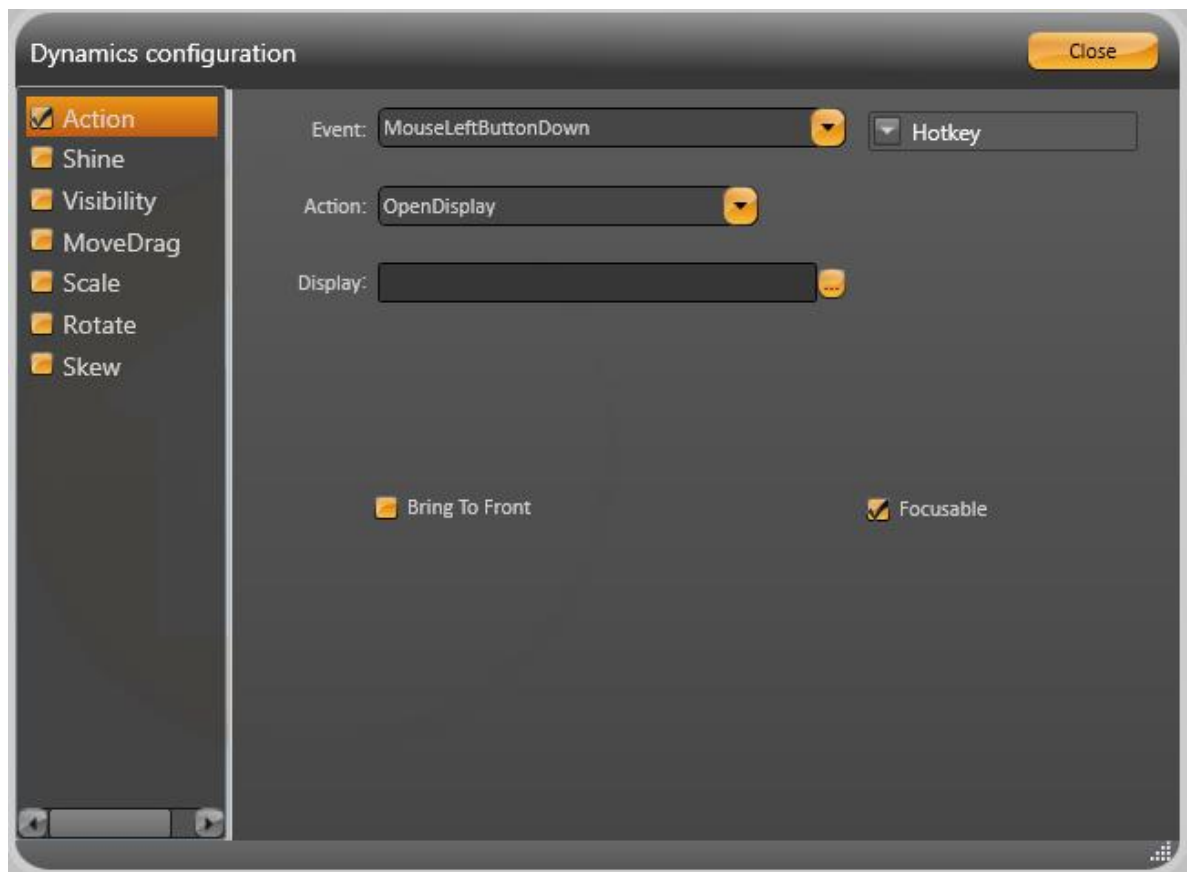


Figura 4-6. Configuração de Page Selector

XPS Viewer

Este objeto trata-se de um visualizador de documentos no formato XPS. Ao dar um clique sobre o visualizador é aberta a opção de configuração *XpsViewer* no menu *Draw*. O caminho completo do documento deve ir no campo *Document*.

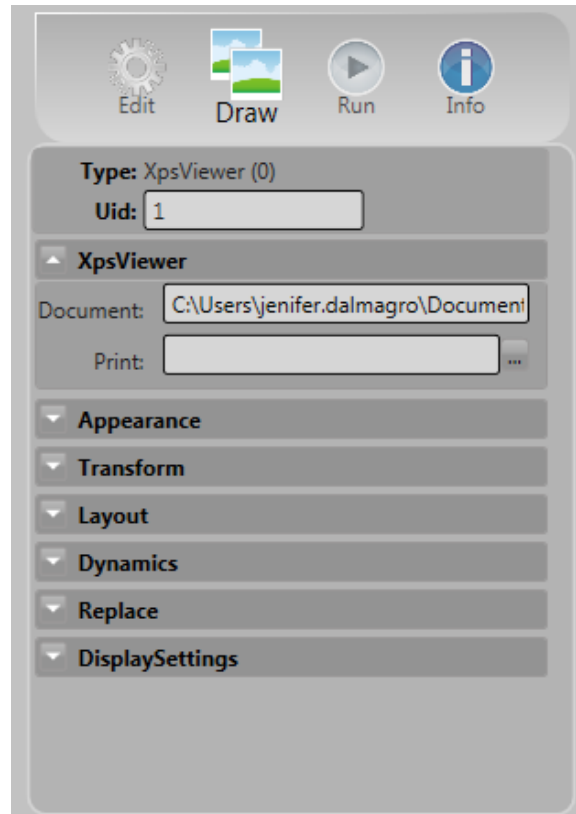


Figura 4-7. Configuração do XPS Viewer

Circular Panel

Para configurar um Painel Circular basta dar um clique duplo sobre o objeto, então aparecerá a tela mostrada na figura a seguir.



Figura 4-8. Configuração de Painel Circular

Control Name: Define um nome para o controle, desta forma ele poderá ser acessado no script CodeBehind.

Columns: Para cada valor que você deseja incluir, selecione uma seção e configure as definições, à direita da lista, que estão descritas a seguir. O painel exibe uma seção para cada coluna.

Image: Escolha em Resources uma imagem para exibir.

Preview: Somente leitura. Exibição de imagem selecionada.

Pie Chart

Para configurar um gráfico de pizza basta dar um clique duplo sobre o objeto, então aparecerá a tela mostrada na figura a seguir.



Figura 4-9. Configuração de Gráfico de Pizza

Control Name: Define um nome para o controle, então ele pode ser acessado no script.

ChartType: Selecione o tipo de gráfico.

Data Items: Para cada valor que você deseja incluir, selecione uma seção e configure as definições, ao lado direito da lista, que estão descritas a seguir. O gráfico mostra uma seção para cada item de dados.

FieldName: Digite um nome para o item de dados.

LinkedValue: Digite um nome de Tag como fonte de dados para o item de dados.

Brush: Clique para selecionar a cor para o item de dados.

Bar Chart

Para configurar um gráfico de barras basta dar um clique duplo sobre o objeto, então aparecerá a tela mostrada na figura a seguir.

Figura 4-10. Configuração de Gráfico de Barras

Control Name: Defina um nome para o controle, para que ele possa ser acessado no script CodeBehind. Ver CodeBehind.

Type: Selecione o tipo de gráfico.

Data Source: Entre o nome do Dataset Table ou query para usar para o gráfico.

Grid Lines: Clique para selecionar a cor das linhas de grade.

Window: Clique para selecionar a cor para o fundo da janela com o gráfico.

Labels: Clique para selecionar a cor para os rótulos.

Show Horizontal Labels 45°: Selecione para mostrar os rótulos do eixo horizontal em angulo de 45°.

Show value over bar: Selecione para exibir o valor da barra acima da barra.

Data Items: Para cada coluna, existente na fonte de dados, que você deseja incluir, marque uma barra e configure as definições ao lado direito da lista, que estão descritas a seguir. O gráfico exibe uma barra para cada linha de definição.

FieldValue: Nome da coluna no banco de dados.

Min.: Digite uma Tag para definir o valor mínimo.

Max.: Digite uma Tag para definir o valor máximo

Brush: Clique para selecionar a cor para o item de dados.

Componente WPF Control

Executando um duplo clique de mouse sobre o componente aparecerá uma janela para a escolha do objeto a ser criado, dentre os controles WPF cadastrados no projeto, ou para instalar um novo controle WPF.

Alarmes e Tendências



Alarmes

O objeto *Alarm* é utilizado para controlar advertências.



⚠ Cria uma Janela de Alarme.

Este item permite criar e posicionar a janela de alarme que tem a aparência mostrada na figura a seguir.

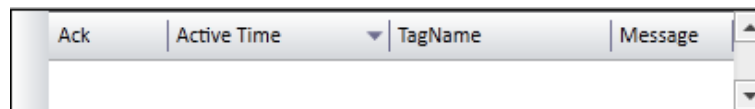


Figura 4-11. Janela de Alarmes



Figura 4-12. Configuração da Janela de Alarme

Control Name: Nome qualquer dado ao controle. É utilizado via CodeBehind para relacionar o objeto gráfico a uma classe .Net. (TAlarmWindow)

MaxLines: Número máximo de linhas que será apresentado no objeto. O filtro é feito no servidor, portanto, os dados nem chegam a ser enviados do servidor para o cliente com a finalidade de ganhar desempenho.

List: Tipos de alarmes que serão apresentados no objeto.

- *OnlineAlarms:* Somente alarmes online
- *AlarmsHistory:* Somente alarmes históricos
- *Events:* Somente eventos
- *AlarmsHistory+Events:* alarmes históricos mais os eventos

History Interval: DateTimeOffset representando o início do período de histórico.

History Interval (To): DateTimeOffset representando o fim do período de histórico.

Ack by Page: Configuração de um objeto que quando recebe um trigger reconhece todos os alarmes visíveis no objeto de alarme.

Show Column Titles: Habilita mostrar os títulos das colunas.

Show group control: Habilita a área onde pode arrastar os nomes das colunas para fazer agrupamentos.

Show group control (Label): Texto que aparecerá dentro da área do *Group control*.

AllowSort: Habilita a funcionalidade de sort no lado direito de cada coluna.

Allow Column Reorder: Habilita a funcionalidade de arrastar a coluna para o lugar de outra coluna trocando assim a ordem entre elas.

Display Value Column as String: Habilita substituir na coluna *Value* o valor pela sua string correspondente, configurado no dicionário.

Filter: Este campo é um "where" de um sql statement, portanto, a sintaxe válida é qualquer coisa válida em um where sql, levando em conta os nomes das colunas existente no objeto. Exemplo:

```
[TagName] = 'Tag.Tag1' AND [Group] = 'Alarm.Group.Critical'
```

Refresh: Configuração de um objeto que quando recebe um trigger faz um refresh do objeto de alarme.

Display Millisecond: Habilita a visualização dos milissegundos na formatação das datas.

Ack Selected Line: Configura o hotkey ou combinação de hotkeys para reconhecer a linha selecionada no objeto de alarme.

Lista Columns: Através das setas para cima e para baixo, podemos mudar a ordem na qual as colunas serão visualizadas no objeto de alarme.

Para cada coluna podemos configurar:

Visible: Habilita se a coluna estará visível ou não.

Allow Filter: Habilita a opção de Filtro na coluna.

Show in column chooser: Permite selecionar e arrastar a coluna.

Title: Configura o título que será visualizado no cabeçalho da coluna.

Width: Configura a largura da coluna.

Sort: Configura o tipo de sort que será aplicado automaticamente na coluna.

Tendência (Legada)

O objeto Tendência (Legada) mostra o registro do historiador de uma Tag.



Cria uma janela de tendência.

Atenção:

Este controle foi substituído pelo apresentado no próximo item. Não recomendamos sua utilização em novos projetos.

Posicione a janela de tendência e com um duplo clique abra a janela de configuração conforme mostrado a seguir.

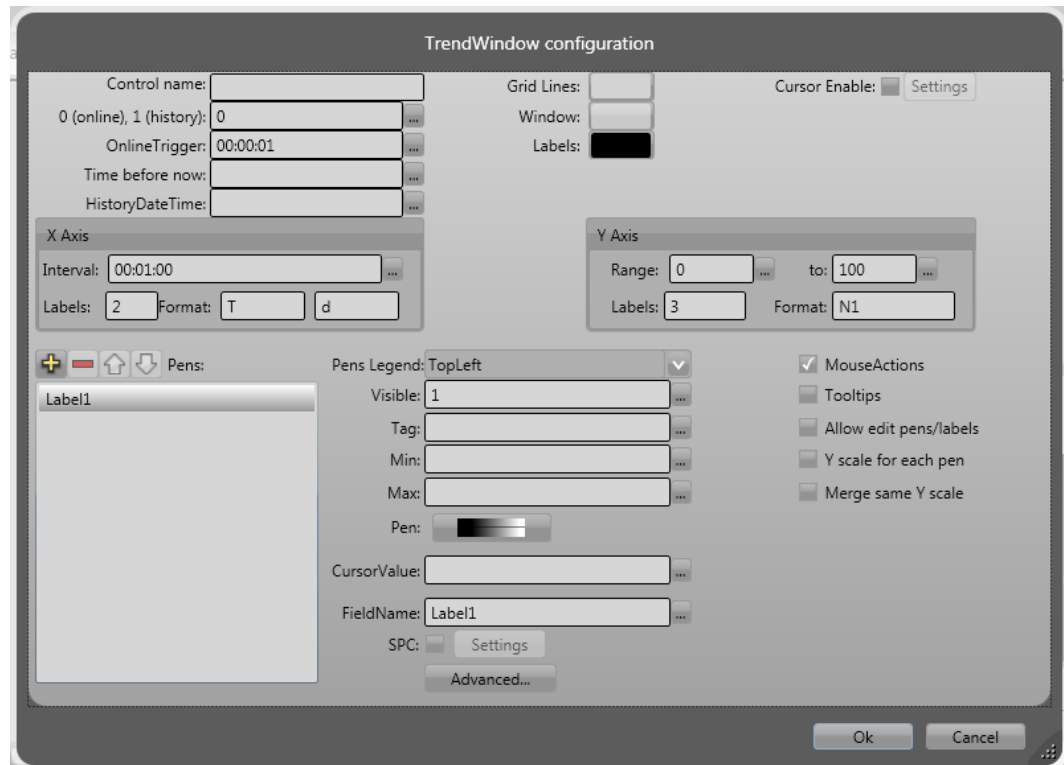


Figura 4-13. Configuração da Janela de Tendência

ControlName: Define um nome para o controle, desta forma pode ser acessado no script código associado. Consulte: Acesso ao objeto Tela no script associado.

0(online) 1(history): 0 - o controle mostra os dados online. 1 - o controle mostra os dados do historiador. Exemplo:

Tag.onLineHist

OnlineTrigger: Taxa de atualização do controle de tendência online. Exemplo:

00:00:01 ou {Tag.trendTimeSpan}

Time Before now: Tempo mostrado na janela de tendência antes da última medida. Exemplo:

00:00:01 ou {Tag.trendTimeSpan}

HistoryDateTime: O ponto inicial para o controle de tendência do historiador. Exemplo:

{Tag.initialTrendHistory}

Grid Lines: Define a cor das linhas de grade na tela de fundo do controle de tendência.

Window: Define a cor da tela de fundo do controle de tendência.

Labels: Define a cor dos rótulos do controle de tendência.

Offline Marker: Quando o modulo de Historian está parado no Trend mostra as penas com preenchimento.

Cursor Enable: Habilita (selecionado) ou desabilita (não selecionado) o cursor vertical.

Settings: Abre a janela Cursor Settings, apresentada na figura a seguir, cuja descrição dos campos de configuração seguem.

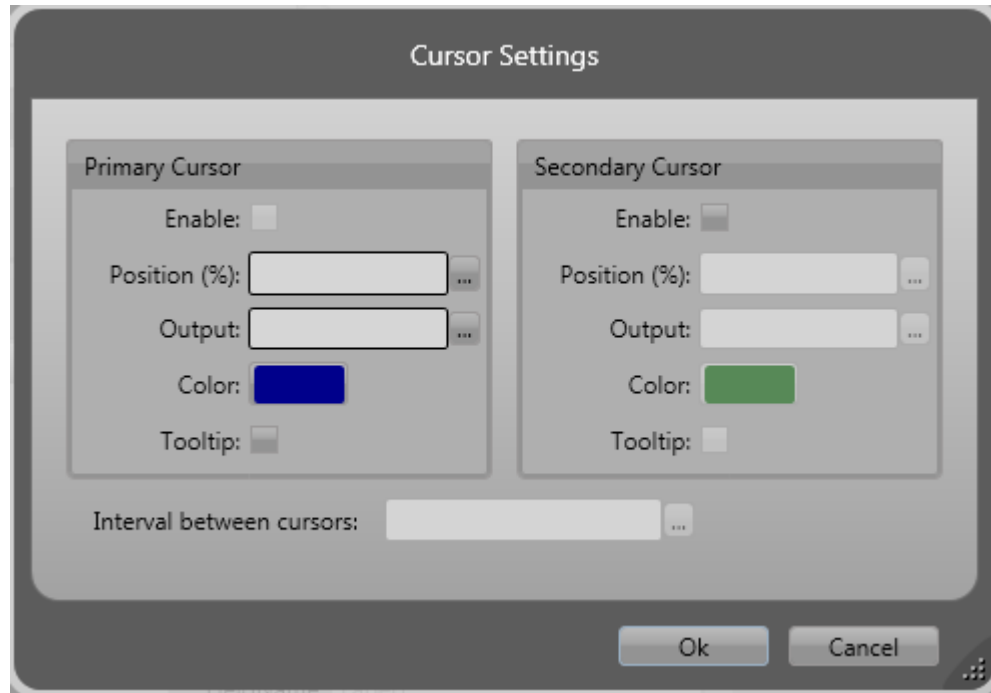


Figura 4-14. Cursor Settings

Cursor Settings Enable (exclusivo secundário): Quando marcada habilita outro cursor vertical.

Cursor Settings Position (%): Indica a posição do cursor, onde 0 significa posição inicial e 100 a posição final. Exemplo:

`Tag.cursorPos`

Cursor Settings Output: Indica o valor do eixo X para a posição atual do cursor. Exemplo:

`Tag.cursorOut`

Cursor Settings Color: Define a cor do cursor.

Cursor Settings Tooltip: Mostra a propriedade do cursor (hora e data).

Cursor Settings Interval Between Cursors: Variável somente de leitura que indica o intervalo entre os cursores, com uma resolução máxima de 1 ms.

Movimentação dos Cursores: A posição dos cursores dentro da janela de controle de tendência é alterada através do uso os botões do mouse. O botão esquerdo controla o cursor primário, já o cursor secundário é movimentado pelo botão direito.

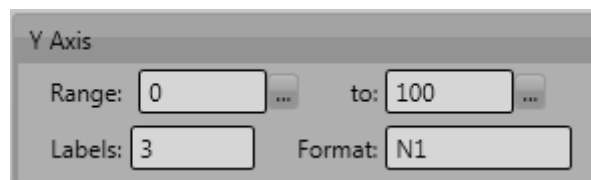


Figura 4-15. Eixo Y

Range: Define os valores mínimo e máximo para o eixo Y.

Labels: Define a quantidade de linhas de grade horizontais.

Format: Formato dos valores no eixo Y. Para formatos numéricos válidos, consulte o item referente ao formato das Tags. Exemplo: N1 (número com 1 casa decimal).

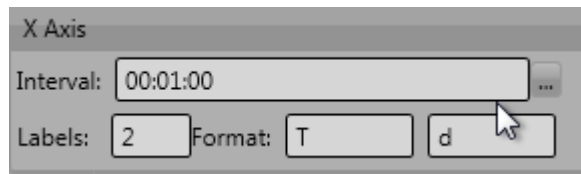


Figura 4-16. Eixo X

Interval: Define o TimeSpan do eixo X.

Labels: Define a quantidade de linhas de grade horizontais.

Format: O formato do eixo X é definido por dois campos: formato da primeira e da segunda linha. Isto é especialmente útil para representar marcas de rótulo que requerem dois níveis de informação. Para formatos de data e tempo válidos consulte o item referente ao formato das Tags. Exemplo: T (Tempo) para o formato da primeira linha, d (data abreviada) para o formato da segunda linha.

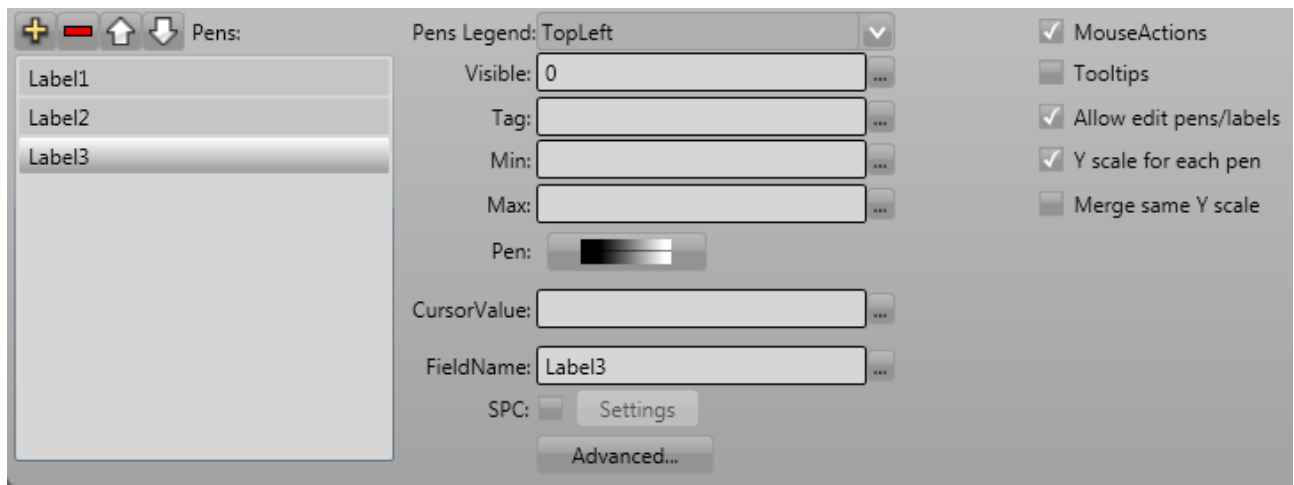


Figura 4-17. Configuração da Caneta

Mouse Actions: Habilita (selecionado) desabilita (não selecionado) a interação do mouse na janela de tendência.

Tooltips: Com essa opção habilitada, colocando o mouse em cima de um Marker mostra informações relacionada ao mesmo.

Allow edit pens/labels: Habilita (selecionado) desabilita (não selecionado) a edição da legenda/escala em Y em tempo real na janela, dando um duplo clique sobre o label da legenda ou nos limites da escala em Y na janela de tendência.

Y scale for each pen: Habilita (selecionado) desabilita (não selecionado) a escala Y individual para para caneta na janela de tendência.

Merge same Y scale: Habilita (selecionado) desabilita (não selecionado) a mescla da escala Y das canetas caso estas sejam iguais na janela de tendência.

Lista Pens: Através das setas, podemos criar novas linhas e mudar a ordem na qual estas serão visualizadas na legenda do gráfico de tendência.

Os campos abaixo são configurações individuais, para cada linha de tendência.

Pens Legend: Define a posição da legenda das canetas no controle de tendência.

Visible: Mostra (1) ou oculta (0) a caneta selecionada. Exemplo: 1 ou {showPen1}

Tag: Define a Tag que fornecerá o valor para a caneta.

Min: Referência de escala linear para o valor da Tag, de acordo com a faixa do eixo Y.

Max: Referência de escala linear para o valor da Tag, de acordo com a faixa do eixo Y.

Pen: Ao clicar neste botão é aberta a janela da figura a seguir onde é possível definir o estilo, cor e espessura da linha da caneta.

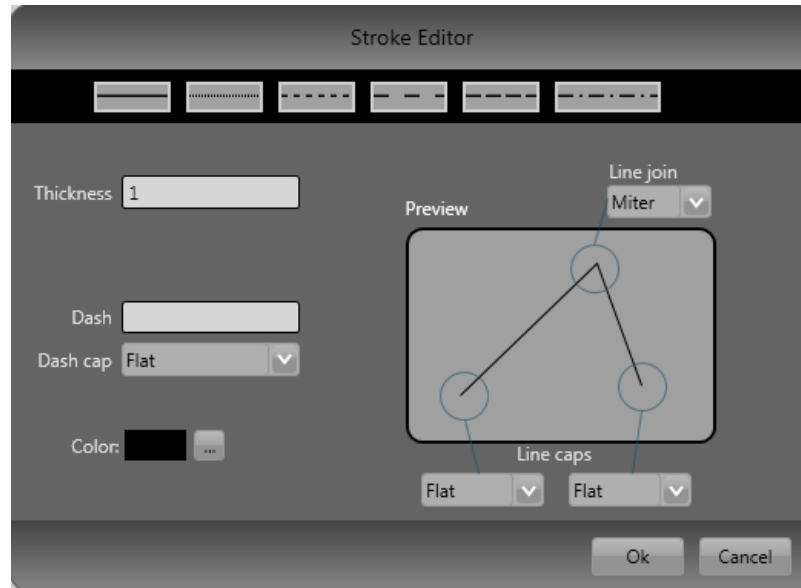


Figura 4-18. Configuração de Estilo da Linha

CursorValue: Define a Tag que receberá o valor real do eixo Y, de acordo com a posição do cursor. Exemplo:

`Tag.pen1CursorValue.`

FieldName: Define o nome da caneta.

SPC: Habilita (selecionado) ou desabilita (não selecionado) algumas funções matemáticas para a caneta selecionada, por exemplo mínimo, máximo, média... Clicando em settings é possível selecionar estas funções.

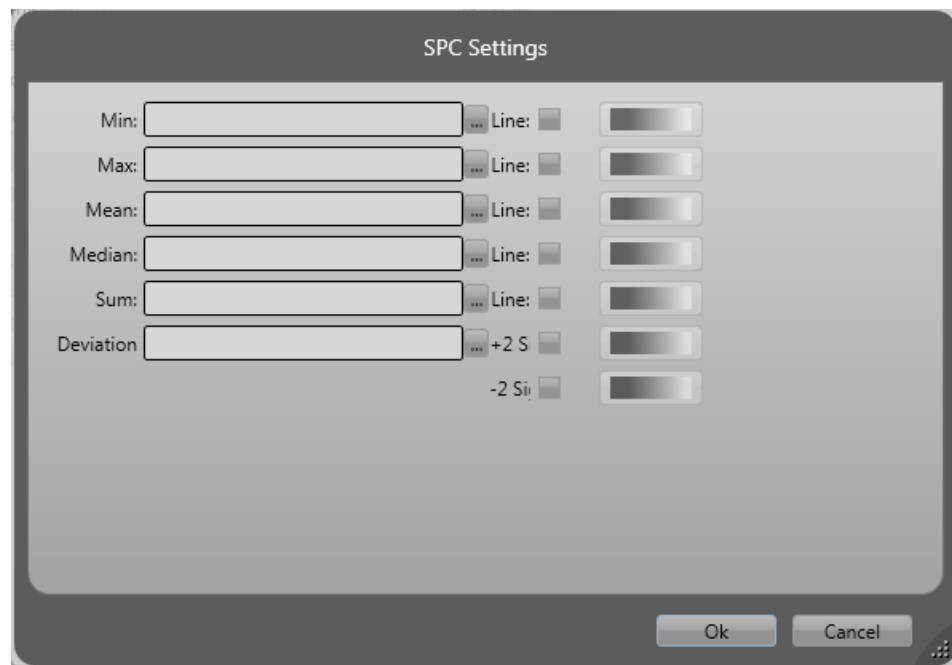


Figura 4-19. SPC Settings

Advanced: Abre a janela apresentada na figura a seguir.

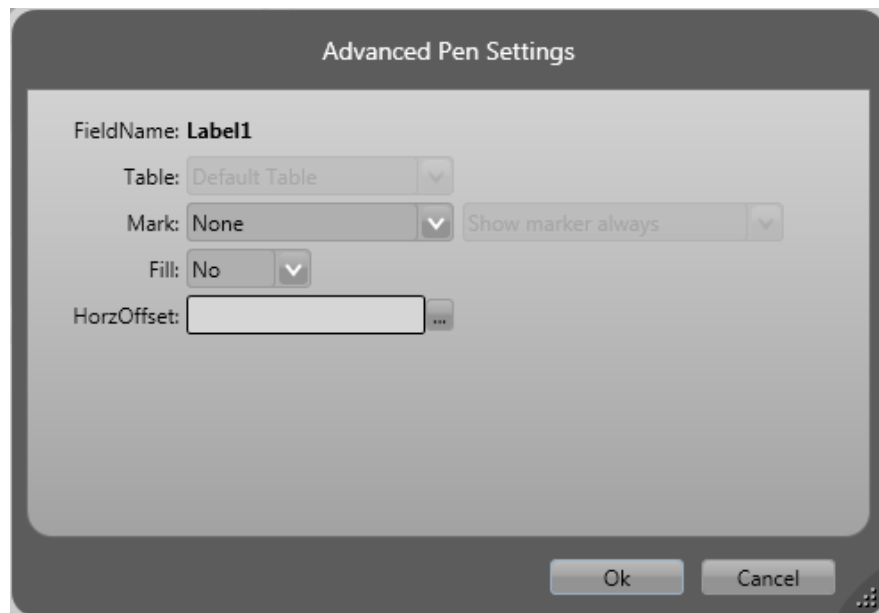


Figura 4-20. Opções Avançadas

Advanced Pen Settings FieldName: Apresenta o nome da linha cujas configurações podem ser alteradas nesta janela.

Advanced Pen Settings Table: Permite selecionar valores para serem plotados pela Pena de uma tabela diferente da default.

Advanced Pen Settings Mark: Define a marca para cada ponto na linha de tendência, pode ser selecionada a opção *Show marker always*, onde a linha sempre apresentará a marca escolhida ou ainda a opção *Show marker only when Tag quality is not good*, onde as marcas só serão adicionadas à linha caso a qualidade do ponto não seja boa.

Advanced Pen Settings Fill: Caso seja selecionada a opção Yes, preenche a área do gráfico abaixo da linha.

Advanced Pen Settings HorzOffset: Coloca uma Tag com o valor em TimeSpan para mostra somente naquela pena os dados passados (histórico).

O controle de tendência permite que várias canetas sejam exibidas junto. Quando as suas canetas não estão na mesma faixa use as dicas abaixo para ajustar os dados no mesmo gráfico e assim obter uma melhor visualização. Se alguma caneta apresentar uma faixa inferior, 0 a 1, por exemplo, pode-se configurar a propriedade máxima da caneta para 1. Assim, quando o valor do real for 1, o valor 100 será exibido no gráfico (escala 100/1). Se alguma caneta apresentar uma faixa superior, 0 a 1000, por exemplo, pode-se configurar a propriedade máxima da caneta para 1000. Assim, quando o valor do real for 1000, o valor 100 será exibido no gráfico (escala 1/10).

Tendência

O objeto Tendência mostra o registro do historiador de uma Tag.



Cria uma janela de tendência.

Posicione a janela de tendência e com um duplo clique abra a janela de configuração. Campos associados:

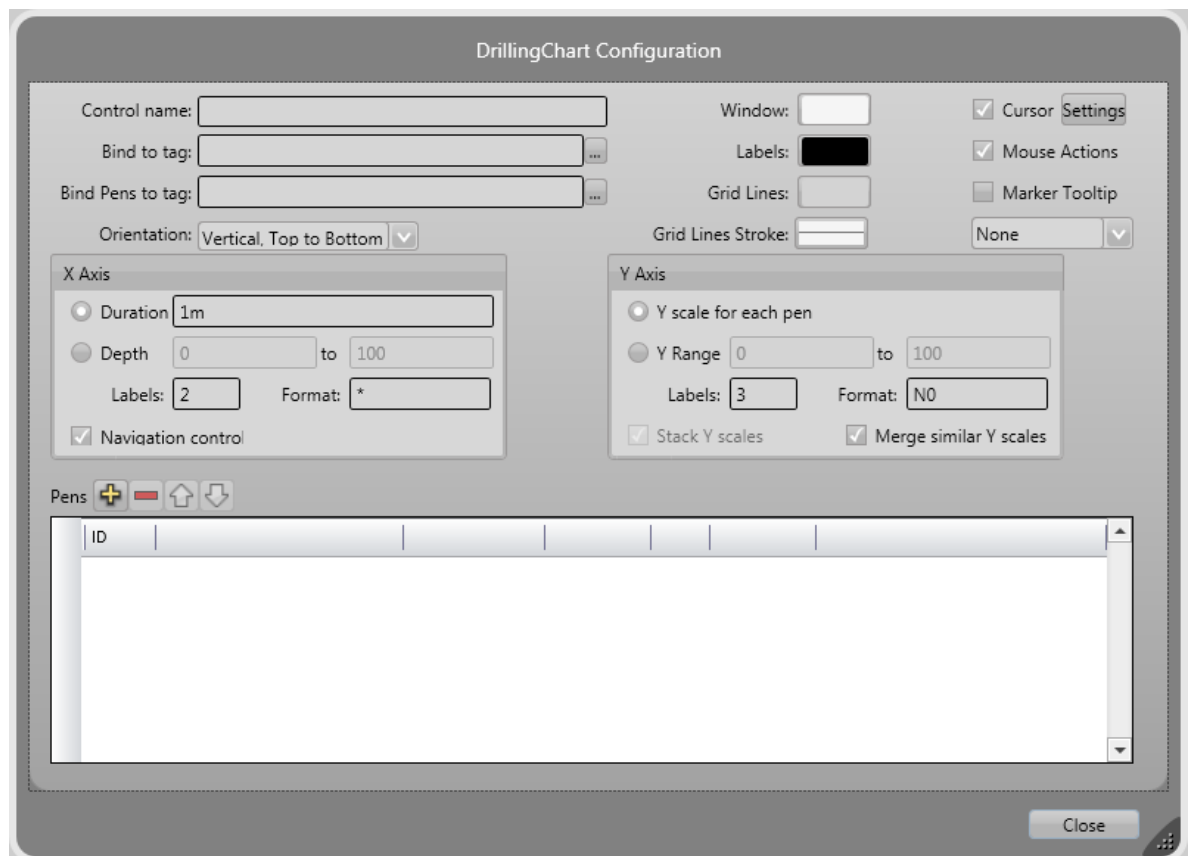


Figura 4-21. Configuração da Janela de Tendência

ControlName: Define um nome para o controle, desta forma pode ser acessado no script código associado. Acesso ao objeto Tela no script associado.

Bind to Tag: Escolha no botão uma propriedade de um objeto para que a configuração deste objeto tendência seja salva nesta propriedade.

Bind Pens to Tag: Escolha no botão uma propriedade de um objeto na qual os valores das penas serão salvos.

Orientation: Pode-se escolher a orientação do gráfico: *Horizontal*; *Vertical de cima para baixo* ou *Vertical de baixo para cima*.

Window: Define a cor da tela de fundo do controle de tendência.

Labels: Define a cor dos rótulos do controle de tendência

Grid Lines: Define a cor das linhas de grade na tela de fundo do controle de tendência.

Grid Lines Stroke: Define o formato das linhas de grade, cheia, tracejada, etc. e sua espessura.

Cursor Settings: Pressione o botão *Settings* para definir até dois cursores par o gráfico. Um primário e um secundário. Para cada um deles pode ser definido: se Habilitado ou não, a cor e forma da linha, uma legenda (Tooltip), e se devem ou não ser mostrados junto com o cursor os valores atuais das grandezas em x e y.

Mouse Actions: Habilita (selecionado) desabilita (não selecionado) a interação do mouse na janela de tendência.

Marker Tooltips: Com essa opção habilitada, colocando o mouse em cima de um Marker mostra informações relacionada ao mesmo.


Y Axis:

- **Y scale for each pen:** Marque esta opção se desejar atribuir escalas própria para cada pena. Neste caso o *Range* será definido na tabela de penas abaixo.
- **Y Range:** Define os valores mínimo e máximo para o eixo Y.
- **Labels:** Define a quantidade de linhas de grade horizontais.
- **Format:** Formato dos valores no eixo Y. Para formatos numéricos válidos, consulte o item referente ao formato das Tags. Exemplo: N1 (número com 1 casa decimal).
- **Stack Y scales:** Marque esta opção para que as escalas apareçam verticalmente uma sobre a outra. Se não marcar as escalas apareceram lado a lado.
- **Merge similar Y scales:** Marque para que se as escalas forem similares somente mostrar uma delas.

X Axis:

- **Duration:** Define o intervalo de tempo do eixo-X. Escolha esta opção se o eixo X é de tempo.
- **Depth:** Escolha esta opção se o eixo X não é de tempo. Neste caso defina os valores mínimo e máximo para o eixo X.
- **Labels:** Define a quantidade de linhas de grade horizontais.
- **Format:** O formato do eixo X é definido por dois campos: formato da primeira e da segunda linha. Isto é especialmente útil para representar marcas de rótulo que requerem dois níveis de informação. Para formatos de data e tempo válidos consulte o item referente ao formato das Tags. Exemplo: *t* (Tempo) para o formato da primeira linha, *d* (data abreviada) para o formato da segunda linha.
- **Navigation controls:** Marque esta opção se deseja que apareça, abaixo do eixo X ícones para o controle de navegação do gráfico: Aumentar (Zoom in), diminuir (Zoom out), deslocar para a esquerda, deslocar para a direita, resetar valores e ocupar toda a

tela. 

- **Pens:**  Esses botões permitem adicionar nova pena, excluir uma pena e ainda alterar a ordem das linhas com definição de penas. Para cada linha os dados a seguir devem ser definidos.
- **TagName:** Define a Tag que irá fornecer o valor para a pena.
- **PenSettings:** Clicando na célula aparece janela para definições decor e formato da linha, se preenchimento abaixo da linha, e símbolos sobre os pontos.
- **MinValue:** Mínimo valor da escala linear para o valor da Tag, de acordo com o intervalo do eixo Y.
- **MaxValue:** Máximo valor da escala linear para o valor da Tag, de acordo com o intervalo do eixo X.
- **Pen Label:** Rótulo com o título para a pena.
- **Auto:** Marcando esta célula, o range será automático, de acordo com os valores apresentados.
- **SPC:** Habilita (selecionado) ou desabilita (não selecionado) algumas funções matemáticas para a pena selecionada, por exemplo mínimo, máximo, média...

O controle de tendência permite que várias penas sejam exibidas junto. Quando as suas penas não estão na mesma faixa use as dicas abaixo para ajustar os dados no mesmo gráfico e assim obter uma melhor visualização.

Se alguma pena apresentar uma faixa inferior, 0 a 1, por exemplo, pode-se configurar a propriedade máxima da pena para 1. Assim, quando o valor do real for 1, o valor 100 será exibido no gráfico (escala 100/1).

Se alguma pena apresentar uma faixa superior, 0 a 1000, por exemplo, pode-se configurar a propriedade máxima da pena para 1000. Assim, quando o valor do real for 1000, o valor 100 será exibido no gráfico (escala 1/10).



Estudo Dirigido 4-1: recursos Historian e TrendChart

Com o auxílio do instrutor configure os recursos *Historian* e *TrendChart*, além de criar filtros para geração de relatórios.



DICA: caso você queira saber mais sobre os recursos *Historian* e *TrendChart* no BluePlant, assista o Webinar a seguir. Nele é explicada a configuração dos recursos *Historian* e *TrendChart*, além da criação de filtros para geração de relatórios.



Configuração Avançada



Grade de Dados



Cria uma janela DataGrid.

Posicione a janela *DataGrid* e dê um duplo clique para abrir a janela de configuração.

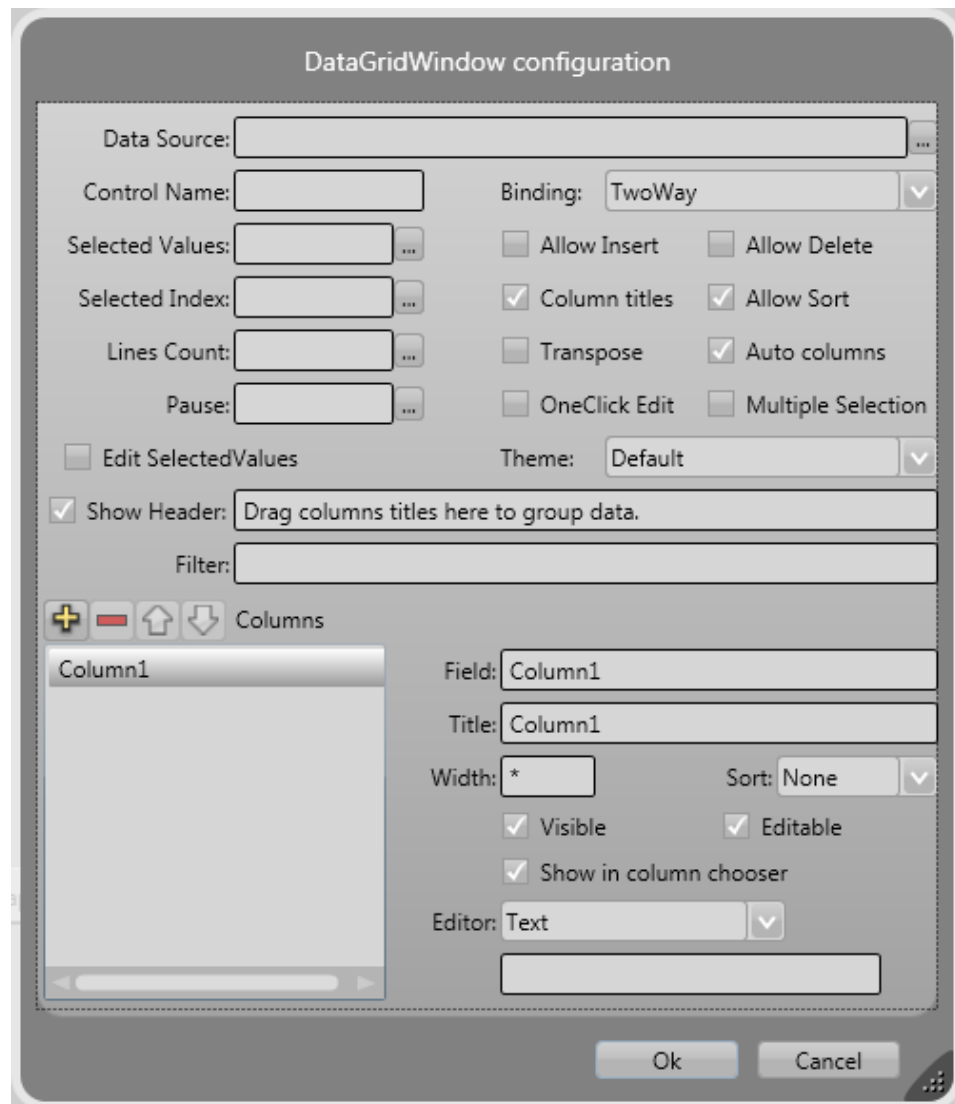


Figura 4-22. Configuração da Janela do DataGrid

Data Source: Objeto Dataset no qual quer se mostrar o conteúdo no DataGrid. Pode ser tanto Dataset.Table quanto Dataset.Query.

Control Name: Nome qualquer dado ao controle. É utilizado via CodeBehind para relacionar o objeto gráfico a uma classe .Net. (TDataGridWindow)

Selected Values: Tag ou Tag array do tipo Text, onde o conteúdo de cada coluna será colocado dentro de cada posição do array.

Selected Index: Objeto que recebe qual a linha que está selecionada. A numeração de linhas começa em 0.

Lines Count: Objeto que recebe a quantidade de linhas existente no DataGridView.

Theme: Seleção do tema visual que será utilizado.

Binding mode: Seleciona a forma do fluxo de dados:

- **OneTime:** O objeto grid é populado somente uma única vez.
- **OneWay:** Os dados somente saem do banco de dados para popular o GRID.
- **TwoWay:** Os dados saem do banco para popular o GRID, e quando modificado no GRID são atualizados também no banco de dados.
- **OneWayToSource:** Os dados são somente atualizados do GRID para o banco de dados.

Allow Insert: Habilita inserção de novas linhas no objeto DataGridView.

Column titles: Habilita mostrar o título das colunas.

Transpose: Habilita a transposição do objeto.

OneClick Edit: Habilita a edição em um clique.

Allow Delete: Habilita deletar a informação.

Allow Sort: Habilita poder mudar a ordem de visualização.

Auto columns: Habilita as colunas a serem criadas automaticamente em função das colunas existentes no datatable do Dataset.Table ou Dataset.Query.

Multiple Selection: Habilita múltiplas seleções.

Edit Selected values: Habilita se o conteúdo da coluna poderá ser modificado.

Filter: Este campo é um "where" de um sql statement, portanto, a sintaxe válida é qualquer coisa válida em um where sql, levando em conta os nomes das colunas existentes no objeto. Exemplo:


```
[Col1] = 'ABC' AND [Col2] = 'DEF'
```

Lista Columns: Através das setas para cima e para baixo, podemos mudar a ordem na qual as colunas serão visualizadas no objeto, e com os botões + e - podemos adicionar ou remover colunas.

Para cada coluna podemos configurar:

- **Field:** Nome da coluna no banco. Se esta informação estiver errada não será possível associar a coluna ao datatable do banco.
- **Title:** Configura o título que será visualizado no cabeçalho da coluna.
- **Width:** Configura a largura da coluna.
- **Sort:** Configura o tipo de sort que será aplicado automaticamente na coluna.
- **Visible:** Habilita se a coluna estará visível ou não.
- **Editatble:** Habilita se o conteúdo da coluna poderá ser modificado.
- **Show in column chooser:** Permite selecionar e arrastar a coluna.
- **Editor:** Seleciona o tipo de dados da coluna.

Barra de Ferramentas Horizontal

Representação Gráfica dos elementos	Descrição
	Open Grid Settings: Configuração e ajuste da grade



	Zoom: Ajuste de aproximação ou afastamento da tela
	Group: Realiza o agrupamento dos objetos selecionados
	Ungroup. Realiza o desagrupamento dos objetos selecionados
	Union: Realiza a união de objetos geométricos gerando um único objeto
	Intersect: Realiza a intersecção de objetos geométricos gerando um único objeto
	Exclude: Realiza a exclusão da forma geométrica frontal dentre os objetos selecionados gerando um novo objeto
	Exclusive-Or: Realiza a exclusão da intersecção das formas geométricas selecionadas gerando um novo objeto
	Align Left: Realiza os alinhamentos dos objetos à esquerda do último objeto selecionado
	Align Horizontal Center: Centraliza os objetos horizontalmente sendo referência o último objeto selecionado
	Align Right: Realiza os alinhamentos dos objetos à direita do último objeto selecionado
	Align Top: Realiza os alinhamentos dos objetos na parte superior do último objeto selecionado
	Align Vertical Center: Centraliza os objetos verticalmente sendo referência o último objeto selecionado
	Align Bottom: Realiza os alinhamentos dos objetos na parte inferior do último objeto selecionado
	Move to The Front: Movimenta para frente os objetos selecionados
	Move to The Back: Movimenta para trás os objetos selecionados
	Resize Width: Realiza o ajuste da largura dos objetos selecionados tendo como base o último objeto selecionado
	Resize Height: Realiza o ajuste da altura dos objetos selecionados tendo como base o último objeto selecionado
	Space Evenly Horizontally: Realiza o espaçamento horizontal igual entre os objetos selecionados
	Space Evenly Vertically: Realiza o espaçamento vertical igual entre os objetos selecionados
	Flip Horizontally: Realiza a inversão horizontal dos objetos selecionados
	Flip Vertically: Realiza a inversão vertical dos objetos selecionados
	Lock Element: Bloqueia todos os objetos selecionados
	Unlock Element: Desbloqueia o objeto selecionado através da Ferramenta de Seleção Direta
	Unlock All Elements Realiza o desbloqueio de todos os elementos da tela

Tabela 4-1. Elementos da Barra de Ferramentas Horizontal

Configuração Dinâmica

A opção *Dynamic* permite acessar a janela de Configuração Dinâmica. Selecione a dinâmica ou não a fim de habilitá-la ou desabilitá-la respectivamente. A figura a seguir ilustra essa configuração.



Figura 4-23. Configuração Dinâmica

Os itens que compõem a configuração dinâmica como mostrado na figura anterior são descritos na sequência:

Action

Executa ações disparadas pela interface do usuário. As configurações associadas são:

Event: escolha um dos eventos do Mouse. Mais de um evento pode ser selecionado para cada ação. Exemplo: uma ação para *MouseLeftButtonDown* (ao pressionar o botão esquerdo do mouse) e outra para *MouseLeftButtonUp* (ao soltar o botão esquerdo do mouse)

Action: escolha uma ação para o evento determinado.

- Nenhum: sem ação
- SetValue: define um valor para o objeto
 - Object: objeto que receberá o valor
 - Value: valor que será passado ao objeto
- ToggleValue: alterna o valor do objeto. Se o valor atual do objeto é zero, o valor será 1. Se o valor atual do objeto é diferente de zero, o valor será 0
- Object: define o objeto que será alternado
- OpenDisplay: abre uma tela
- Display: indica o nome da tela que será aberta
 - CloseDisplay: fecha uma tela
 - OpenLayout: abre um Layout.
- Layout: o nome do Layout que será aberto
- RunScript: executa um script localizado no código associado à tela. Escreva o novo nome do método e clique no botão *New* ou selecione um dos métodos existentes na *ComboBox*
- RunExpressions: executa a expressão determinada
- Expression: digite a expressão. Ex: `Tag.a + 1`, ou `Tag.a + Tag.b`, ou `Math.Cos(Tag.angle) * Math.PI`
- Result (opcional): digite a Tag ou a propriedade que receberá o valor da expressão

Hotkey: teclas de atalho rápido. Exemplos:

1. Executar expressões, tais como somar dois valores e passar o resultado para outra Tag.
Sintaxe:

```
Tag.quantity1 + Tag.quantity2
```


Resultado:

`Tag.totalQuantity`

2. Incrementar uma Tag:

`TagCounter + 1`

Resultado:

`TagCounter`

3. Incrementar uma Tag de 0 a 10:

`If(TagCounter < 10 , TagCounter + 1 , 0)`

Resultado:

`TagCounter`

Shine

Altera a aparência do objeto dinamicamente. As configurações associadas são:

- **IsMouseOver:** entre com uma Tag que receberá o *OverValue* ou o *NotOverValue*.
- **OverValue:** valor *IsMouseOver* quando o mouse está sobre o objeto.
- **NotOverValue:** valor *IsMouseOver* quando o mouse NÃO está sobre o objeto.
- **ShowUid:** mostra uma janela com o valor contido em *Tooltip*.
- **Tooltip:** valor a ser exibido quando o mouse estiver sobre o objeto na janela *ShowUid*.
- **Mouse over appearance:** a aparência do objeto quando o mouse está sobre ele.
 - *Opacity:* opacidade do objeto (0 = transparente, 1 = opaco).
 - *Scale:* tamanho do objeto (0.5 = metade, 1 = mesmo tamanho, 1.5 = uma vez e meia, 2 = tamanho duplo).
 - *OuterGlow:* define a cor do efeito *OuterGlow* (borda degrade que destaca o objeto), a *checkbox* habilita ou desabilita o mesmo.
 - *TextColor:* define a cor do texto, a *checkbox* habilita ou desabilita o mesmo.
- **Mouse is not over appearance:** a aparência do objeto quando o mouse não está sobre ele.
 - *Opacity:* opacidade do objeto (0 = transparente, 1 = opaco).
 - *Scale:* tamanho do objeto (0.5 = metade, 1 = mesmo tamanho, 1.5 = uma vez e meia, 2 = duas vezes).
- **IsSelected appearance:** a aparência do objeto quando está selecionado
 - *IsSelected:* define se o objeto está selecionado ou não.
 - *Opacity:* opacidade do objeto (0 = transparente, 1 = opaco).
 - *Scale:* tamanho do objeto (0.5 = metade, 1 = mesmo tamanho, 1.5 = uma vez e meia, 2 = duas vezes).




Scale Reference: dinâmica da escala.



Centro: escala com referência no centro do objeto.



Esquerda: escala com referência à esquerda do objeto.

-  Acima: escala com referência na parte superior do objeto.
-  Direita: escala com referência à direita do objeto.
-  Abaixo: escala com referência na parte inferior do objeto.

TextIO

Dinâmica de Entrada e Saída de Texto. É composta pelas configurações ilustradas a seguir:

- **Blinding Mode:** as configurações estão indicadas a seguir.
 - *TwoWay*: faz a leitura e escrita do valor do objeto selecionado em *ObjectName*
 - *InputOnly*: somente faz a escrita de valores do objeto selecionado em *ObjectName*
 - *OutputOnly*: somente faz a amostragem de valores do objeto selecionado em *ObjectName*
- **ObjectName:** indica o texto que será mostrado no objeto. Se o texto é um valor de Tag ou uma propriedade, ele deve estar entre chaves, por exemplo: {Tag.analogInt1}.
- **DesignModeCaption:** valor mostrado no modo design.
 - *ShowObjectNames*: o conteúdo do campo *ObjectName* exatamente como é.
 - *ShowPlaceHolders*: os caracteres ### são mostrados, a quantidade de caracteres é definida pelo campo *MaxLength*.
- **Input Range:** define a faixa numérica para o valor inserido.
- **MaxLength:** define a quantidade máxima de caracteres.
- **Format:** define o formato de exibição, caso não seja inserido formato o mesmo de ajusta para o tamanho e formato do dado recebido. Em caso de definição do formato, atentar para que este, dependendo do formato definido, realiza arredondamentos nos valores como, por exemplo, Tag.analogInt1= 20.006. Se definido formato com 0.00, o valor exibido será 20.01. Por outro lado, se o texto é um valor de Tag ou uma propriedade, ele deve estar entre chaves como, por exemplo: {Tag.analogInt1}.

HyperLink

Abre um hyperlink.

- **HyperLink type:** escolha uma das opções http, ftp, file, mailto, telnet.
- **Url:** configure a url que será aberta.
- **Hotkey:** teclas de atalho rápido.

Securitiy

Define as permissões do objeto em tempo de execução.

- **Disable:** entre com uma Tag, propriedade ou expressão retornando um valor. Considere o valor resultante:
 - 0 - objeto será habilitado
 - >0 - objeto será desabilitado
- **Verify permissions:** quando selecionado, apenas os grupos de permissões escolhidos, podem acessar o objeto.

- **Confirm message:** quando selecionado, mostra o diálogo configurado antes de tomar alguma ação.

FillColor

Altera a cor de preenchimento do objeto dinamicamente.

- **Expression:** representa o valor usado para a dinâmica FillColor.
- **Bad quality:** caso a qualidade do objeto no campo *Expression* assuma o valor zero (ruim) o preenchimento assumirá a cor e a frequência de piscada configuradas.
- **Undefined quality:** caso a qualidade do objeto no campo *Expression* assuma um valor indefinido (diferente de 0 e de 192) o preenchimento assumirá a cor e a frequência de piscada configuradas.
- **ChangeColor:** altera a cor de preenchimento e tem as configurações indicadas a seguir.
 - *UsingLimits:* a cor resultante será dada quando o valor for igual ou maior que um dos limites
 - *AbsoluteValue:* a cor será o valor da expressão, neste caso o valor deve ser uma cor válida. Exemplos: "White" ou "#FFFFFF".

Exemplo de uso de limites:

Se 1 = Vermelho e 10 = Azul então, quando o valor for 0, o objeto terá sua própria cor (dinâmica de preenchimento de cor não terá ação). Quando o valor for de 1 a 9, o objeto terá a cor vermelha e se o valor for maior que 10, o objeto terá a cor azul.

LineColor

Altera a cor da linha do objeto dinamicamente.

- **Expression:** representa o valor usado para a dinâmica *LineColor*.
- **Bad quality:** caso a qualidade do objeto no campo *Expression* assuma o valor zero (ruim) a linha assumirá a cor e a frequência de piscada configuradas.
- **Undefined quality:** caso a qualidade do objeto no campo *Expression* assuma um valor indefinido (diferente de 0 e de 192) a linha assumirá a cor e a frequência de piscada configuradas.
- **ChangeColor:** altera a cor da linha e segue o padrão de configuração da propriedade *FillColor*.

TextColor

Altera a cor do texto do objeto dinamicamente.

- **Expression:** representa o valor usado para a dinâmica de *TextColor*.
- **Bad quality:** caso a qualidade do objeto no campo *Expression* assuma o valor zero (ruim) o texto assumirá a cor e a frequência de piscada configuradas.
- **Undefined quality:** caso a qualidade do objeto no campo *Expression* assuma um valor indefinido (diferente de 0 e de 192) o texto assumirá a cor e a frequência de piscada configuradas.
- **ChangeColor:** altera a cor do texto e segue o padrão de configuração da propriedade *FillColor*.

Bargraph

Dinâmica de Gráfico de Barras.

- **Expression:** representa o valor usado para a dinâmica de gráfico de barras.
- **Value Range:** os valores mínimos e máximos que corresponderão ao percentual de preenchimento mínimo e máximo.
- **Fill (%):** o percentual de preenchimento do gráfico de barras mínimo e máximo.
- **Bar Color:** cor do gráfico de barras.
- **Orientation:** orientação do gráfico de barras.







Representação Gráfica dos elementos	Descrição
	Movimentação das barras do gráfico da parte inferior para superior
	Movimentação das barras do gráfico do centro para as extremidades horizontais
	Movimentação das barras do gráfico da parte superior para inferior
	Movimentação das barras do gráfico da esquerda para direita
	Movimentação das barras do gráfico do centro para as extremidades verticais
	Movimentação das barras do gráfico da direita para a esquerda

Tabela 4-2. Gráfico de Barras

Visibility

Altera a visibilidade e opacidade do objeto dinamicamente.

- **Visible:** entre com uma Tag, propriedade ou expressão retornando um valor. Considere o valor resultante: 0 - objeto será visível e >0 - objeto será oculto.
- **Opacity:** utiliza as configurações indicadas a seguir.
 - *Object Value:* o valor usado para definir a opacidade.
 - *Range:* os valores mínimos e máximos que corresponderão à opacidade mínima e máxima.
 - *Opacity:* a opacidade mínima e máxima (0 - invisível, 0.5 - um pouco transparente, 1 - opaco).
- **Hide when security is disabled:** se selecionado, oculta o item enquanto o usuário logado não tiver a permissão.

MoveDrag

Move o objeto dinamicamente.

- **Blinding Mode:** utiliza as configurações indicadas a seguir.
 - *TwoWay:* movimento de entrada e saída.
 - *InputOnly:* movimento somente de entrada, o objeto não se move quando seu valor é alterado.
 - *OutputOnly:* movimento somente de saída, o objeto não se move com a interação do usuário
- **Horizontal Move:** utiliza as configurações indicadas a seguir.
 - *Expression:* representa o valor usado para o movimento horizontal.
 - *Range:* os valores mínimos e máximos que corresponderão à posição horizontal mínima e máxima.
 - *Position:* posição horizontal mínima e máxima.
- **Vertical Move:** utiliza as configurações indicadas a seguir.
 - *Expression:* representa o valor usado para o movimento vertical.
 - *Range:* os valores mínimos e máximos que corresponderão à posição vertical mínima e máxima.

- *Position*: posição vertical mínima e máxima.
- **Use previous object for position reference**: quando habilitado utilizado os limites do objeto com ID anterior ao ID do mesmo com os limites da movimentação.

Scale

Altera o tamanho do objeto dinamicamente.

- **Width Scale**: utiliza as configurações indicadas a seguir.
 - *Expression*: representa o valor usado para a escala de largura
 - *Range*: os valores mínimos e máximos que corresponderão ao percentual de escala de largura mínima e máxima
 - *Scale (%)*: o percentual de escala de largura mínima e máxima
- **Height Scale**: utiliza as configurações indicadas a seguir.
 - *Expression*: representa o valor usado para a escala de altura
 - *Range*: os valores mínimos e máximos que corresponderão ao percentual de escala de altura mínima e máxima
 - *Scale (%)*: o percentual de escala de altura mínima e máxima

- **Scale Reference**:



Centro: escala com referência no centro do objeto.



Esquerda: escala com referência à esquerda do objeto.



Acima: escala com referência na parte superior do objeto.



Direita: escala com referência à direita do objeto.



Abaixo: escala com referência na parte inferior do objeto.

Rotate

Gira o objeto dinamicamente.

- **Expression**: representa o valor usado para a rotação.
- **Value Range**: os valores mínimos e máximos que corresponderão ao ângulo mínimo e máximo. Ex: 0 a 100.
- **Angle**: ângulo de rotação mínimo e máximo. Ex: 0 a 360°.
- **RPM**: quando selecionado utiliza o valor no campo *Expression* para indicar quantas rotações de 360° serão realizadas em 1 minuto.
- **Center Reference**:



Centro: rotação com referência no centro do objeto.




Esquerda: rotação com referência à esquerda do objeto.



Acima: rotação com referência na parte superior do objeto



Direita: rotação com referência à direita do objeto.


 Abaixo: rotação com referência na parte inferior do objeto.


- **Clockwise:** sentido de rotação do objeto, se selecionado é adotado o sentido horário, caso contrário, anti-horário.
- **Offset X ou Offset Y:** valor que será deslocado o centro da rotação em relação ao ponto de referência no objeto.
- **ON/OFF:** quando com o valor 1 habilita a rotação do objeto, quando em 0 desabilita.

Skew


Inclina o objeto dinamicamente.

- **X-axis Skew:** configurações.
 - *Object Value:* representa o valor usado para a inclinação do eixo X.
 - *Range:* os valores mínimos e máximos que corresponderão ao ângulo mínimo e máximo de inclinação do eixo X. Exemplo: 0 a 100.
 - *Skew (°):* os valores mínimos e máximos de inclinação do eixo X. Exemplo: 0 a 180°.
- **Y-axis Skew:** configurações.
 - *Object Value:* representa o valor usado para a inclinação do eixo Y.
 - *Range:* os valores mínimos e máximos que corresponderão ao ângulo mínimo e máximo de inclinação do eixo Y. Exemplo: 0 a 100.
 - *Skew (°):* os valores mínimos e máximos de inclinação do eixo Y. Exemplo: 0 a 180°.
- **Skew Reference:**

 Centro: inclinação com referência no centro do objeto.

 Esquerda: inclinação com referência à esquerda do objeto.

 Acima: inclinação com referência na parte superior do objeto.

 Direita: inclinação com referência à direita do objeto.

 Abaixo: inclinação com referência na parte inferior do objeto.

TextOutput

Dinâmica de Saída de Texto.

- **Expression:** indica o texto que será mostrado no objeto. Se o texto é um valor de Tag ou propriedade ele deve estar entre chaves. Exemplo: {Tag.analogInt1}.
- **Localizable:** indica se o texto deve ser traduzido ao mudar o dicionário.
- **DesignModeCaption:** o valor mostrado no modo design com as configurações indicadas a seguir.
 - *ShowObjectName:* o conteúdo do campo *Expression* é mostrado exatamente como está.
 - *ShowPlaceHolders:* os caracteres ### são mostrados, a quantidade de caracteres é definida pelo campo *MaxLength*.

- **MaxLength:** define a máxima quantidade de caracteres.
- **Format:** define o formato do dado.

CodeBehind

Use *DrawCodeBehind* para definir um conjunto de funções vinculadas às telas conforme ilustrado na figura a seguir.

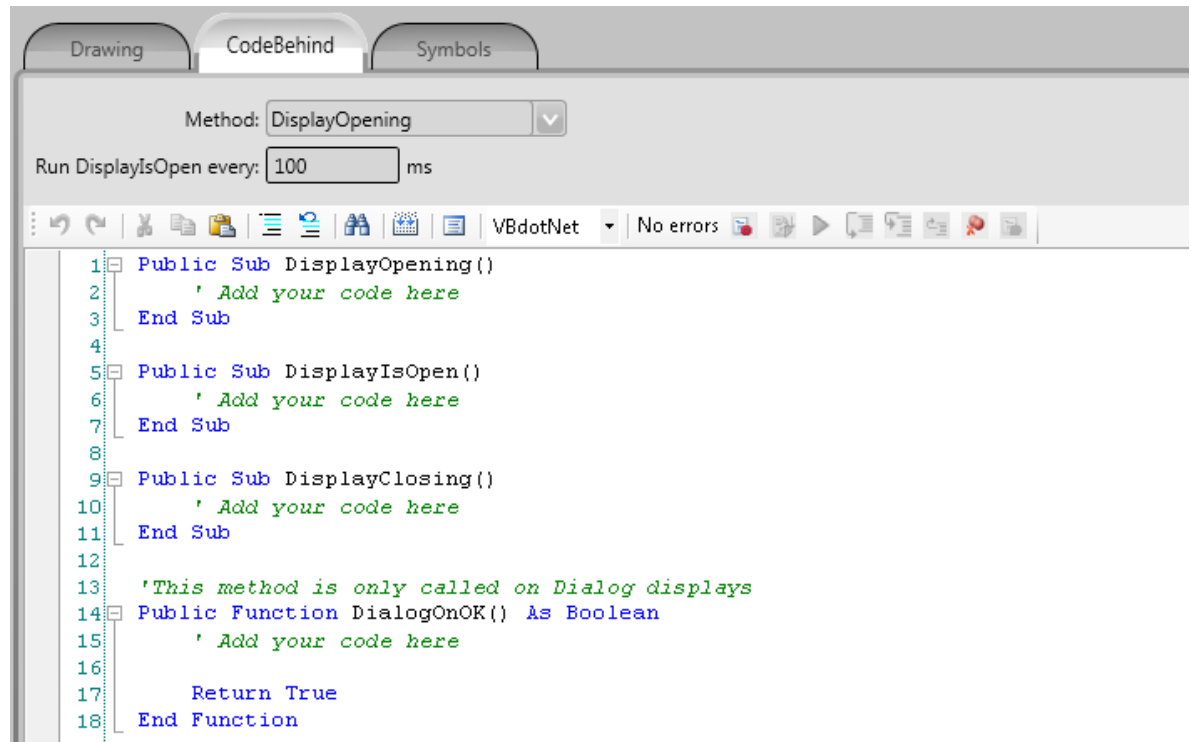


Figura 4-24. CodeBehind

Estas funções podem ser executadas na abertura ou fechamento, ou quando a tela é aberta, dependendo do código configurado. Pode-se usar o código associado para definir os métodos de uso dos comandos de entrada e de mouse a serem executados em uma tela específica.

Define-se no campo *Run DisplayIsOpened every* o intervalo que irá executar o CodeBehind.

Para telas tipo DIÁLOGO use o método *DialogOnOk* incorporado, o qual é chamado quando o botão OK incorporado é pressionado. Se no método retornar TRUE, o diálogo é fechado. Se retornar FALSE continuará aberto. Este método é normalmente usado para garantir a validação dos dados no diálogo (o usuário é solicitado a corrigir itens incorretos antes do fechamento do diálogo). Opções:

- *DisplayOpening()*: executado quando a tela está abrindo.
- *DisplayIsOpen()*: chamado em um intervalo regular enquanto a tela é aberta.
- *DisplayClosing()*: executado quando a tela está fechando.
- *DialogOnOK()*: chamado quando o botão OK é pressionado em uma tela de Diálogo.

Retorno "1" permite o fechamento do diálogo. Retorno "0" evita que o diálogo seja fechado. Pode-se acrescentar variáveis .NET e métodos nesta página. Em função das telas cliente serem projetadas para execução em ambientes distribuídos e Web recomenda-se evitar o uso de funções que não permitam execução de confiança parcial ou que se refiram a caminhos de arquivo físico.

Símbolos Locais

Símbolos são objetos definidos pelo usuário contendo desenhos e propriedades dinâmicas. A figura a seguir mostra um símbolo e o menu de contexto associado.

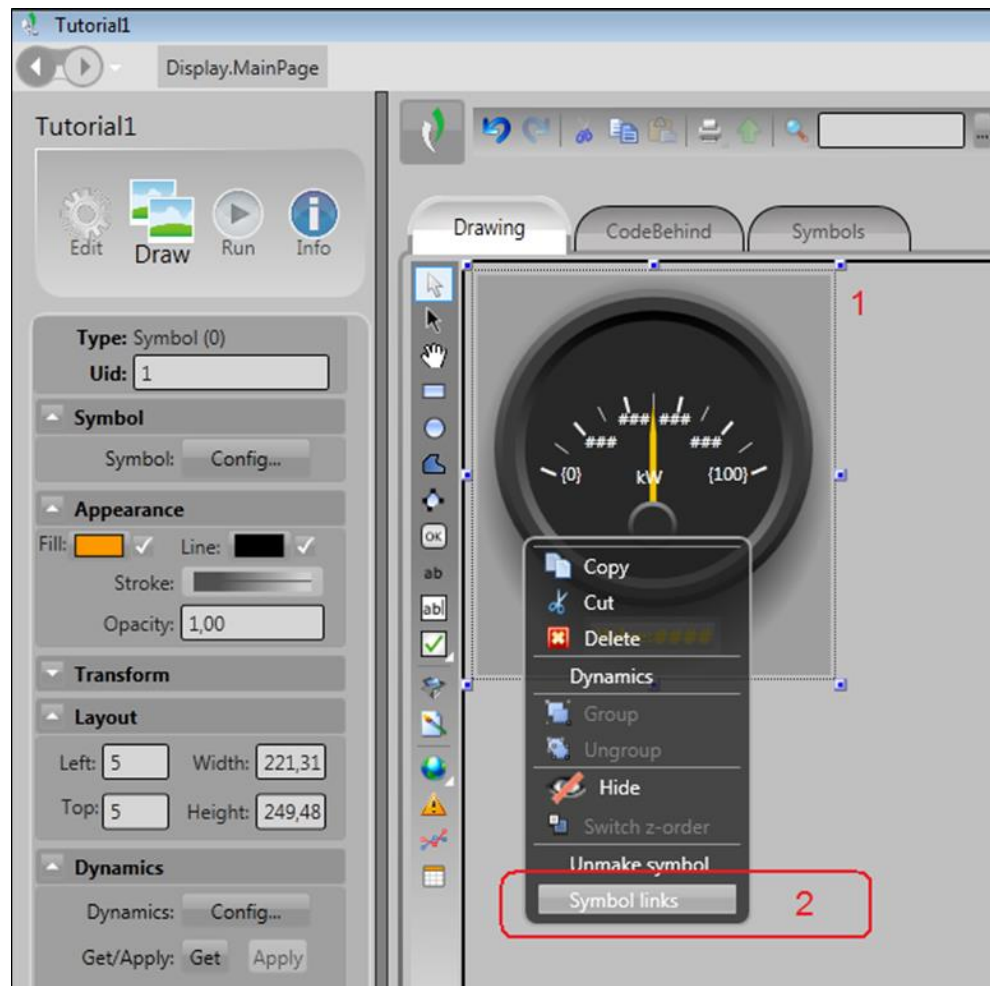


Figura 4-25. Menu Contexto do Símbolo

Para simplificar o uso de símbolos em outras telas e aplicações pode-se definir rótulos em propriedades dinâmicas onde uma *TagName* é esperado. Use a sintaxe:

#LabelName: ou #LabelName:DefaultValue.

Quando usar símbolos em telas ou relatórios selecione novos nomes de Tags na janela de configuração a serem aplicados nos campos de Rótulo definidos. Um conjunto de Símbolos que é visível a todos os projetos é incorporado ao BluePlant framework. Estes símbolos são definidos no arquivo *SymbolLibrary.tproj* localizado na pasta de instalação binária do Produto. O arquivo da biblioteca de símbolo global pode ser editado como um arquivo de projeto. Pode-se também criar e salvar seus próprios símbolos.

Criando Símbolos

As imagens das figuras a seguir apresentam os passos necessários para criar um símbolo. Os procedimentos também estão descritos na sequência.

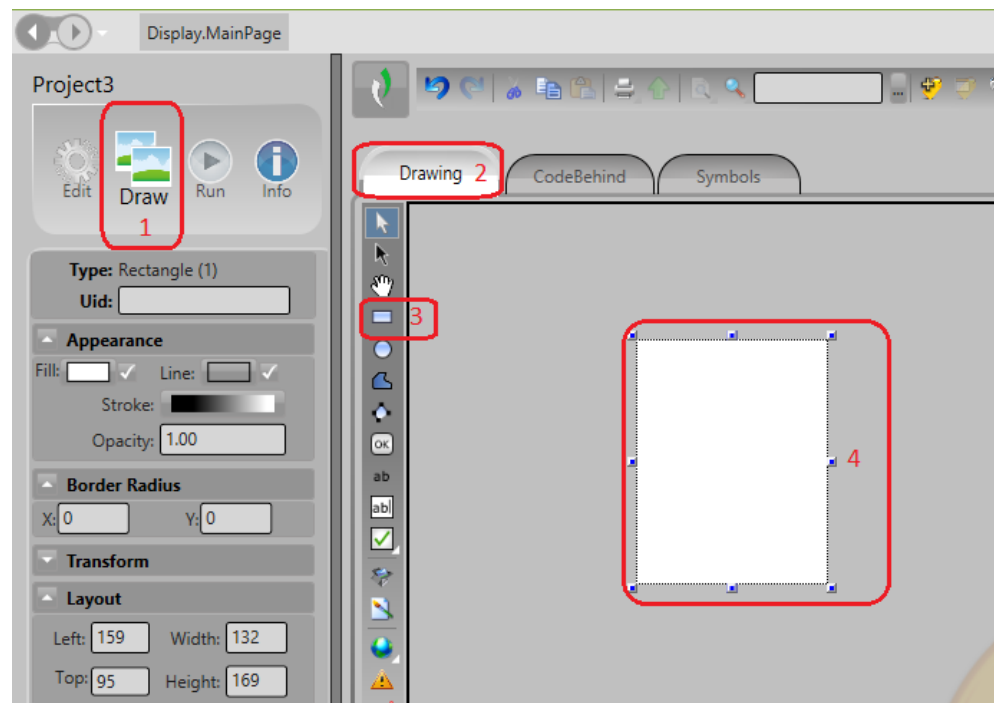


Figura 4-26. Criando um Símbolo

1. Selecionar o menu *Draw*;
2. Clicar na guia *Drawing*;
3. Selecionar o objeto retângulo;
4. Desenhar um retângulo na tela e clicar duas vezes no objeto criado. A tela abaixo deverá aparecer.

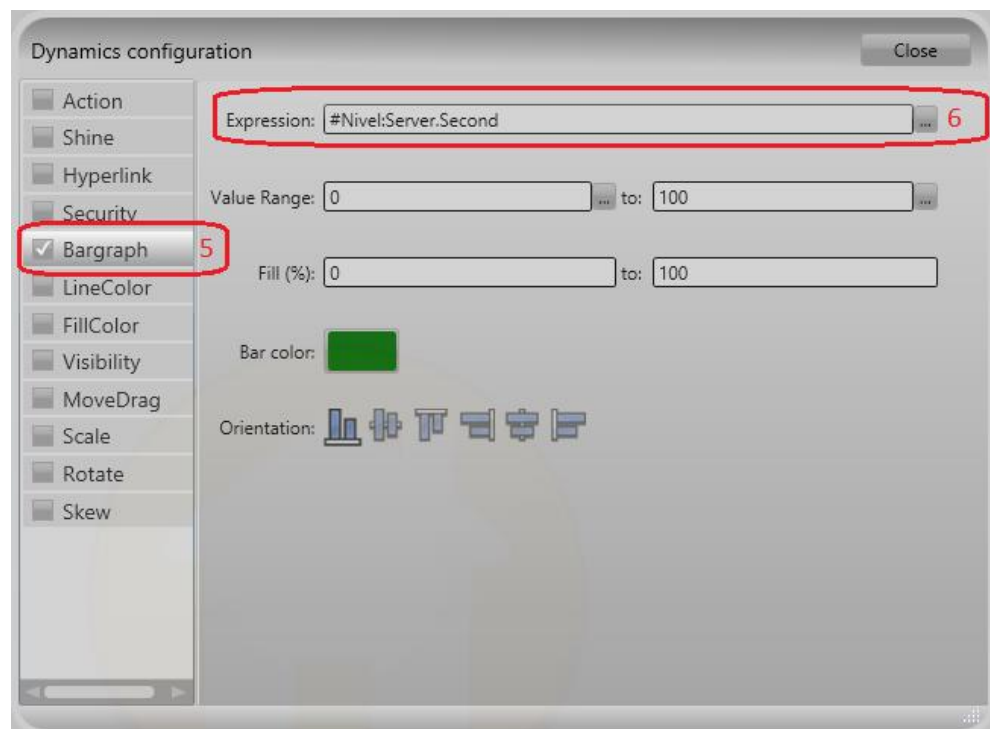


Figura 4-27. Configuração Dinâmica

5. Selecionar a opção *Bargraph*;
6. Digitar no campo Expression a sintaxe abaixo para expor a propriedade Nível:

```
#Nivel:Server.Second.
```

As propriedades de um símbolo são indicadas por meio do uso de mnemônicos conforme mostrado a seguir.

```
#<nome>:<valor> ou #<nome>:(<expression>)
```

7. Fechar a tela de *Dynamics Configuration*;
8. Selecionar todos os objetos que compõem o símbolo, neste caso o retângulo;
9. Clicar com botão direito do mouse para abertura do menu de contexto;
10. Selecionar a opção *Make new Symbol*;
11. Preencher as informações: Nome do símbolo, categoria, descrição e clicar em *YES*;
12. Este símbolo estará disponível para ser inserido em qualquer ponto do projeto através do *Symbol Library*.

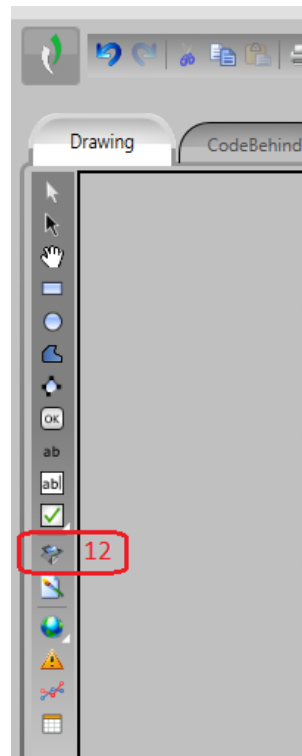


Figura 4-28. Abrir Symbol Library

13. Ir no ícone *Symbol Library* como mostrado na figura anterior e inserir o símbolo criado na tela;
14. Ir na tela de configuração do símbolo dando um duplo click. Aparecerá uma propriedade *Nivel* com o valor *Server.Second* que pode ser alterado para a Tag desejada, podendo cada símbolo ter o seu valor de nível independente.

Modificando um Símbolo

Os passos a seguir, descrevem o que deve ser feito para que um símbolo seja modificado.

1. Selecionar o símbolo que se deseja modificar;
2. Clicar com o botão direito do mouse para abertura do menu de contexto;

3. Selecionar a opção *Edit <nome do símbolo>*;
4. Fazer todas as alterações desejadas no símbolo;
5. Selecionar todos os objetos que compõem o símbolo;
6. Clicar com botão direito do mouse para abertura do menu de contexto;
7. Selecionar a opção *Update <nome do símbolo>*.

Todos os símbolos com este nome da tela corrente serão alterados, se desejar que todos os símbolos com este nome do projeto sejam alterados será necessário fazer um build salvando as telas



Estudo Dirigido 4-2: desenho da aplicação

1. No projeto do Processo de Mistura e Secagem acrescente os objetos que comporão a tela de visão geral, a qual permite ao operador uma visão global do processo de operação da planta. Nessa tela devem ser apresentados os dados mais significativos à operação e objetos que representam o processo. Os dados devem resumir os parâmetros a serem controlados ou monitorados.
2. Acrescente uma tela de malha à aplicação. Selecione uma tela de plano de fundo. Acrescente uma Tendência e crie 4 curvas. Acrescente botões para que seja possível entrar e sair do modo Histórico. Acrescente um botão que mostre a janela de legenda da tendência. Acrescente uma Base de Dados e uma conexão CSV, abra o arquivo em Excel ou Notepad e verifique seu conteúdo.
3. Acrescente uma nova tela à aplicação. Selecione um plano de fundo. Acrescente alguns alarmes ao servidor de alarme. Conecte uma tela de Informações em pelo menos um dos alarmes. Acrescente um Visualizador de Alarme à tela, assim como objetos que tornem possível disparar o alarme.
4. Acrescente alguns usuários/grupos de segurança diferentes à aplicação. Acrescente uma nova tela. Selecione uma tela de plano de fundo. Conecte alguns objetos a diferentes grupos de segurança. Oculte alguns objetos se o usuário não estiver conectado. Acrescente botões para realizar login, logout e exportação/importação de usuários.
5. Acrescente uma nova tela à aplicação e selecione um plano de fundo. Acrescente um indicador analógico à tela. Acrescente um botão para que, quando o usuário clicar nele, um valor seja escrito na Tag conectada ao indicador analógico. Acrescente uma Caixa de Mensagem com pergunta de Confirmação. Acrescente um objeto de Caixa de Texto e um botão à tela. Mostre uma Caixa de Mensagem com o texto a partir da Caixa de Texto. Salve o texto em um arquivo no disco.
6. Acrescente uma Receita à aplicação. Acrescente alguns itens de receita. Acrescente uma nova tela à aplicação e selecione uma tela de plano de fundo. Acrescente botões para carregar, salvar e excluir as receitas.

📌 **DICA:** consulte o Instrutor para orientações adicionais.

5. Implementando a Aplicação



Este capítulo detalha os passos necessários para executar uma aplicação editada e desenhada (tópicos dos capítulos anteriores). Ele inclui ainda considerações sobre informações da aplicação. Além disso, são explorados recursos da aplicação como ferramentas e objetos em tempo de execução e configurações avançadas do BluePlant.

Execução da Aplicação



O ambiente de execução (*Run*) fornece acesso a todos os recursos de execução do Projeto.

Build

No menu *Build* é possível compilar todos os *displays* e *scripts* para uma verificação final, após o término da implementação do projeto. O comando *Build* gera uma verificação completa das lógicas de aplicação, porém não se faz necessário executar o comando *Build* durante a execução do projeto, pois o BluePlant é capaz de verificar automaticamente todas as modificações, durante a edição do projeto, de forma transparente para o usuário.

Messages

Na guia *Messages* é possível dar o comando de *Build*, através do botão de mesmo nome e verificar o resultado da compilação de cada *script* e *display* do projeto.

The screenshot shows the BluePlant interface with the Messages panel open. The Run button in the left sidebar is highlighted with a red box and a '1'. The Build button is also highlighted with a red box and a '2'. The Messages panel shows a table of build results for various modules, with one error highlighted in red.

Module	Object ID	BuildStatus	Output	BuildErrors	Error Location
Script	Class.ServerMain	✓	Debug	none	
Script	Class.ClientMain	✓	Debug	none	
Display	MainPage	✗	Debug	1 error	Line 3
Display	SelectPage	✓	Debug	none	
Display	LogOn	✓	Debug	none	
Display	Header	✓	Debug	none	
Display	About	✓	Debug	none	
Display	HTML5_Logon	✓	Debug	none	
Display	HTML5_MainPage	✓	Debug	none	

Figura 5-1. Guia Messages

Os itens que compõem o menu *Messages* estão numerados na figura anterior e são descritos na sequência.

1. Selecionar a opção *Run*;
2. Selecionar a opção *Build*;
3. Selecionar a opção *Messages* para exibição dos dados relativos ao comando de *Build*.

Todas as colunas apresentadas na figura anterior são somente leitura, com a intenção de sinalizar se há algum problema detectado. Caso o *Build* encontre um erro, a coluna *BuildStatus* apresentará um X vermelho, para corrigi-lo, basta clicar duas vezes sobre a linha que apresenta o erro, e esta ação abrirá o objeto que está com problema.

History

Na guia *History* é possível verificar um resumo de informações sobre os últimos comandos de *Build* executados no projeto. Através do botão *Pack* é possível configurar a exclusão dos registros mais antigos, de acordo com a numeração apresentada na coluna *Build*, que pode ser vista na figura a seguir.

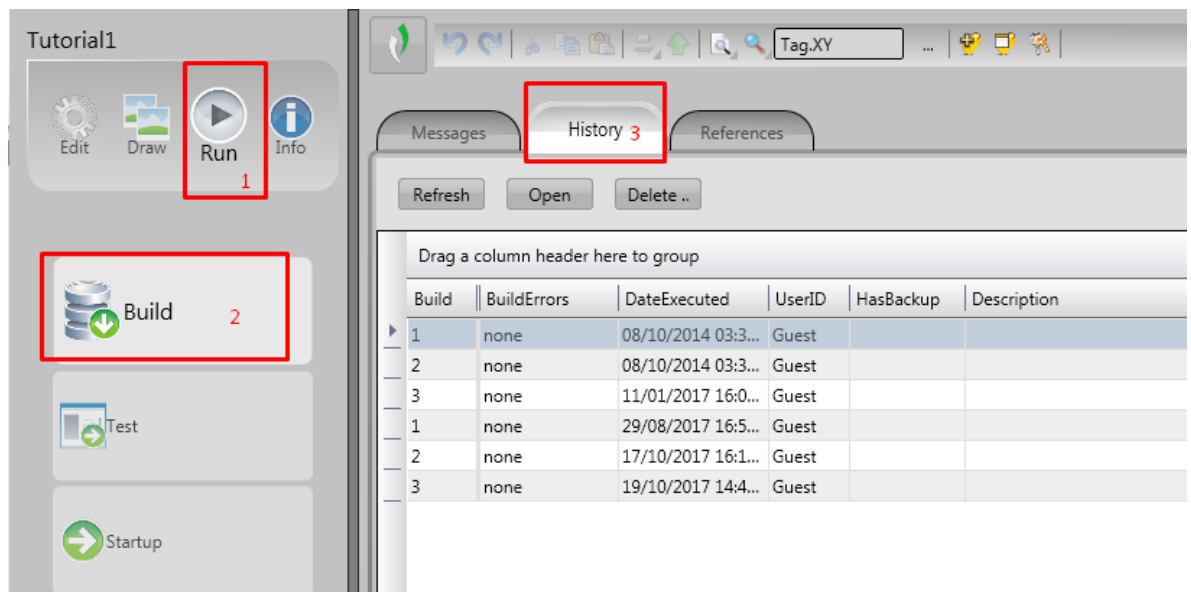


Figura 5-2. Guia History

Os itens que compõem a guia *History* estão numerados na figura anterior e são descritos na sequência:

1. Selecionar a opção *Run*;
2. Selecionar a opção *Build*;
3. Selecionar a opção *History* para exibição dos dados relativos ao histórico dos comandos de *Build*.

References

A guia *References* deve ser utilizada para o registro de referências a arquivos de código externos ou do próprio usuário.

Test

No menu *Test* é possível executar o projeto em modo de teste, o que permite uma proteção de ter o projeto executando de modo semelhante ao *Startup*, porém podendo selecionar que módulos serão executados, tendo as funções de Alarme e Historiador sendo executados em bancos de dados temporários, de forma a não interferir nos bancos de dados definidos para a execução do projeto. Também é possível ativar ferramentas de diagnóstico e todos os Dispositivos são executados no modo de somente leitura.

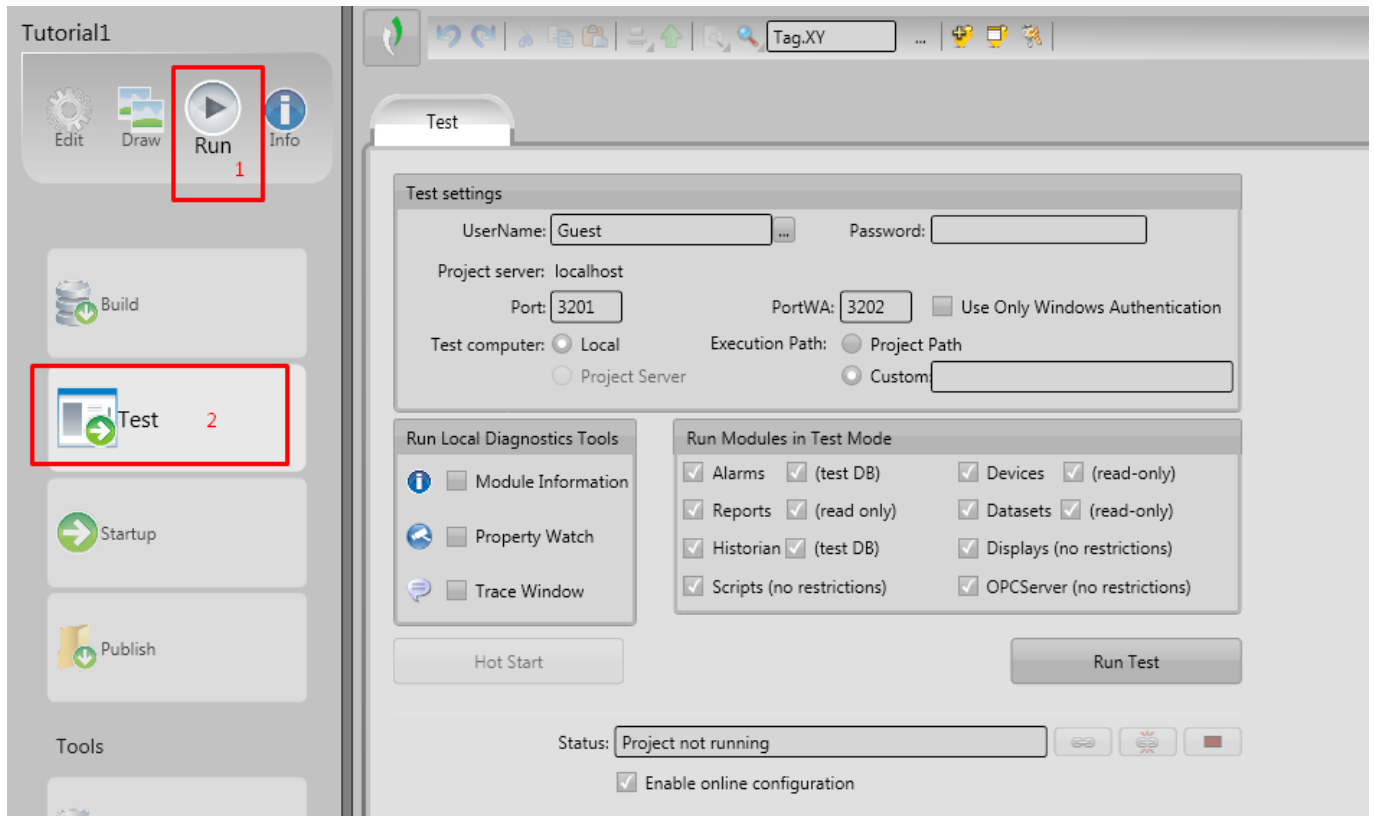


Figura 5-3. Menu Test

Os itens que compõem o menu *Test* estão numerados na figura anterior e são descritos na sequência.

1. Selecionar a opção *Run*;
2. Selecionar a opção *Test* para exibição das configurações disponíveis para a execução do projeto em modo teste.

Test Settings: apresenta configurações gerais do projeto para a execução deste em modo teste.

- *UserName* e *Password*: respectivamente o usuário, que executará o teste do projeto e sua senha.
- *Project Server* e *Port*: informa o servidor e a porta de comunicação que estão configurados para este projeto.
- *Test computer*: caso o servidor usado para o projeto não seja o computador local, é possível escolher entre executar o projeto em modo teste localmente ou no servidor.
- *Execution Path*: escolha da pasta onde serão salvas as informações de execução do projeto.

Run Local Diagnostics Tools: habilita ou desabilita a execução das ferramentas da aplicação (*ModuleInformation*, *PropertyWatch* e *TraceWindow*).

Run Modules in Test Mode: permite habilitar ou desabilitar a execução dos módulos listados de acordo com as restrições apresentadas ao seu lado.

Run Test: inicia a execução do projeto em modo teste.

Status: apresenta informações sobre a execução do projeto. Os botões ao lado da área de exibição de STATUS permitem alterar o estado da aplicação, desconectando, conectando e parando a aplicação.

Online configuration enabled: quando habilitada permite que modificações feitas no projeto durante a execução sejam adicionadas a este sem que seja necessário parar a sua execução.

Startup

No menu *Startup* é possível executar o projeto tal qual ele rodará na produção, executando as funções de Historiador e Alarme nos bancos de dados definidos no projeto, e não mais em arquivos temporários.

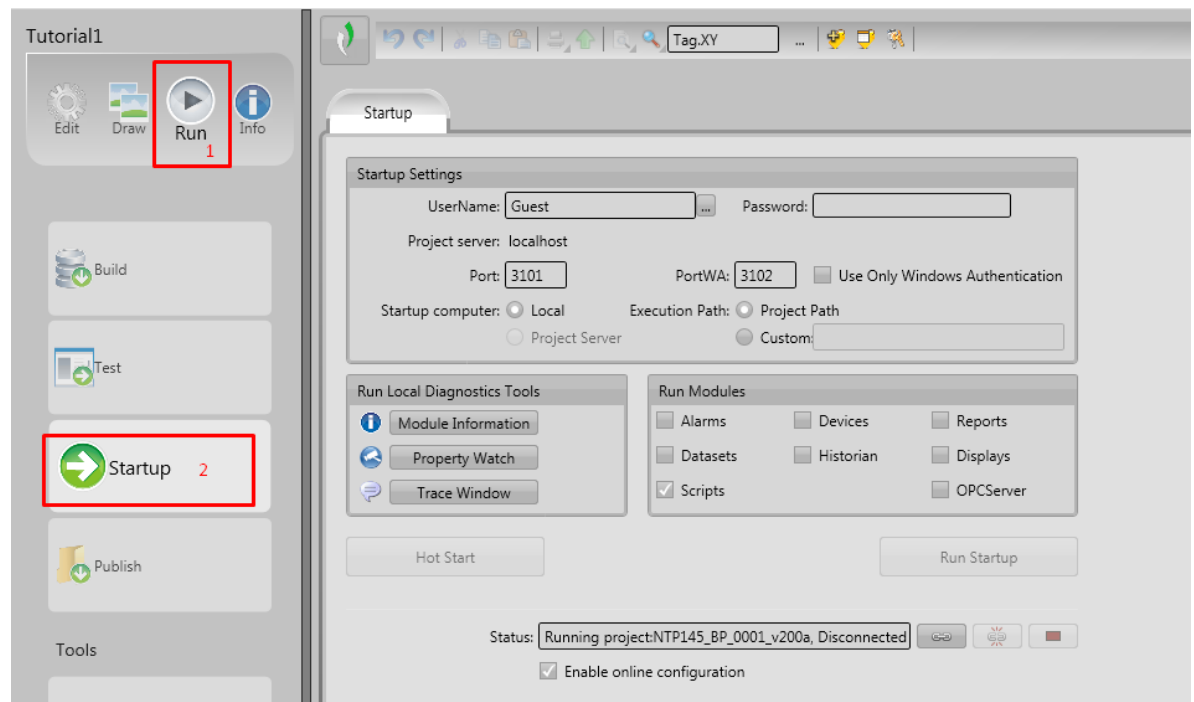


Figura 5-4. Menu Startup

Os itens que compõem o menu *Startup* estão numerados na figura anterior e são descritos na sequência.

1. Selecionar a opção *Run*;
2. Selecionar a opção *Startup* para exibição das configurações disponíveis para a execução do projeto.

Startup Settings: apresenta configurações gerais do projeto para a execução deste.

- *UserName* e *Password*: respectivamente o usuário, que executará o teste do projeto e sua senha.
- *Project Server* e *Port*: informa o servidor e a porta de comunicação que estão configurados para este projeto.

- *Startup Computer*: caso o servidor usado para o projeto não seja o computador local, é possível escolher entre executar o projeto localmente ou no servidor.
- *Execution Path*: escolha da pasta onde serão salvas as informações de execução do projeto.

Run Local Diagnostics Tools: habilita ou desabilita a execução das ferramentas da aplicação (*ModuleInformation*, *PropertyWatch* e *TraceWindow*).

Run Modules: permite habilitar ou desabilitar a execução dos módulos listados.

Run Startup: inicia a execução do projeto.

Status: apresenta informações sobre a execução do projeto. Os botões ao lado da área de exibição de STATUS permitem alterar o estado da aplicação, desconectando, conectando e parando a aplicação.

Online Configuration Enabled: quando habilitada permite que modificações feitas no projeto durante a execução sejam adicionadas a este sem que seja necessário parar a sua execução.

Só é possível executar corretamente uma instância do *Runtime*. Por esta razão quando é executado o comando *Run Startup* em um projeto aberto, o comando permanece desabilitado até que todos os módulos do *Runtime* sejam finalizados. Caso outro projeto seja aberto, o comando *Run Startup* não irá executar corretamente se uma instância de *Runtime* já estiver em execução a partir de um outro projeto ou por linha de comando.

Publish

No menu *Publish* é possível criar uma versão do projeto somente leitura, adequada à utilização no campo.

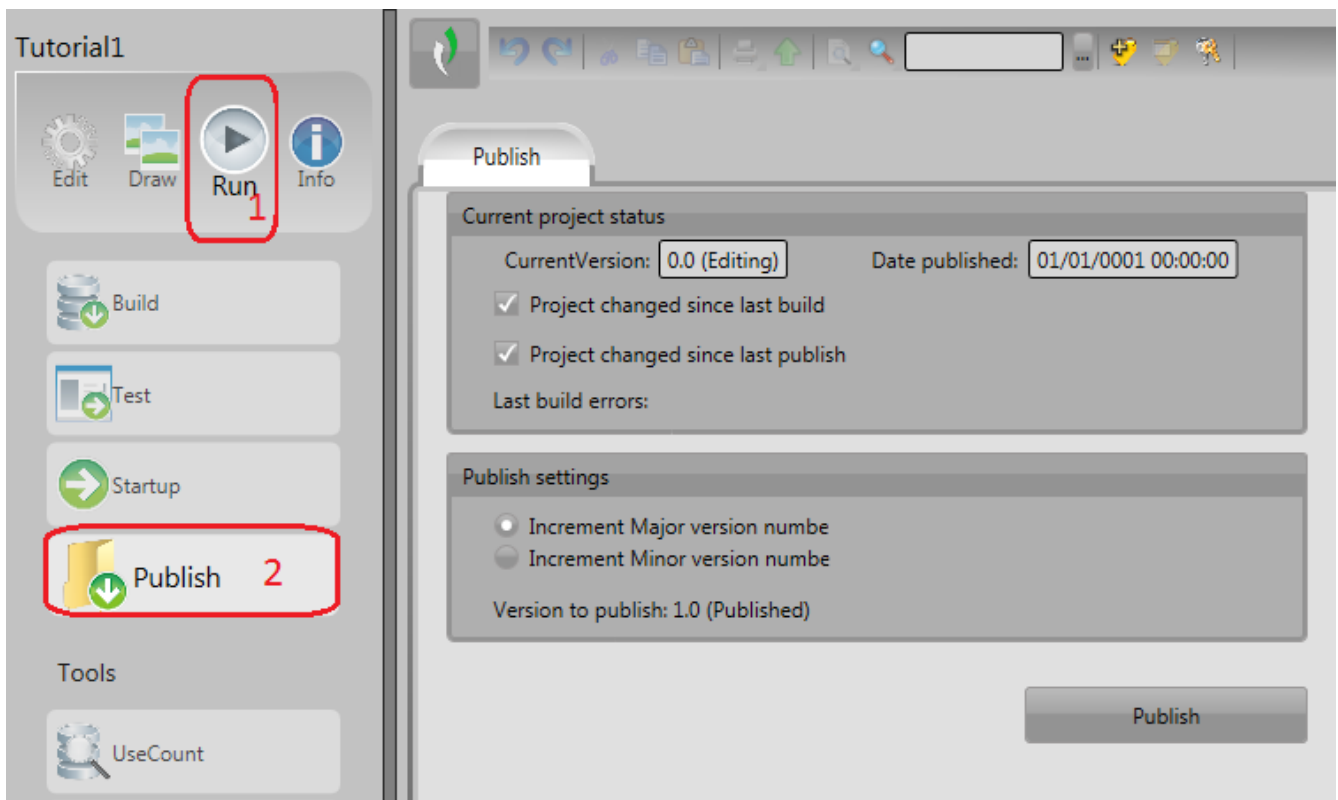


Figura 5-5. Menu Publish

Os itens que compõem o menu *Publish* estão numerados na figura anterior e são descritos na sequência.

1. Selecionar a opção *Run*
2. Selecionar a opção *Publish* para exibição das configurações disponíveis para a publicação do projeto.

Current project status: apresenta as seguintes informações sobre o projeto:

- *CurrentVersion:* versão atual que se encontra o projeto.
- *Date published:* data da última publicação.
- *Project changed since last build:* se selecionado, mostra que houve alterações no projeto desde o último comando de *Build* dado.
- *Project changed since last publish:* se selecionado, mostra que houve alterações no projeto desde a última publicação.

Publish settings: Seleciona se a próxima versão deve ter o primeiro (*Major version number*) ou o segundo (*Minor version number*) número incrementado.

Ao clicar no botão *Publish* o projeto é publicado de acordo com as opções configuradas nesta tela e é criado um arquivo com a extensão “.teng”, cuja execução é de somente leitura, com a versão de publicação configurada.

UseCount

O menu *UseCount* fornece o cálculo do número de vezes que as Tags são usadas dentro do projeto. Fornece, também, as referências cruzadas do projeto inteiro.

UseCount

Na guia *UseCount* são mostrados os objetos do projeto e o número de vezes que estes são usados.

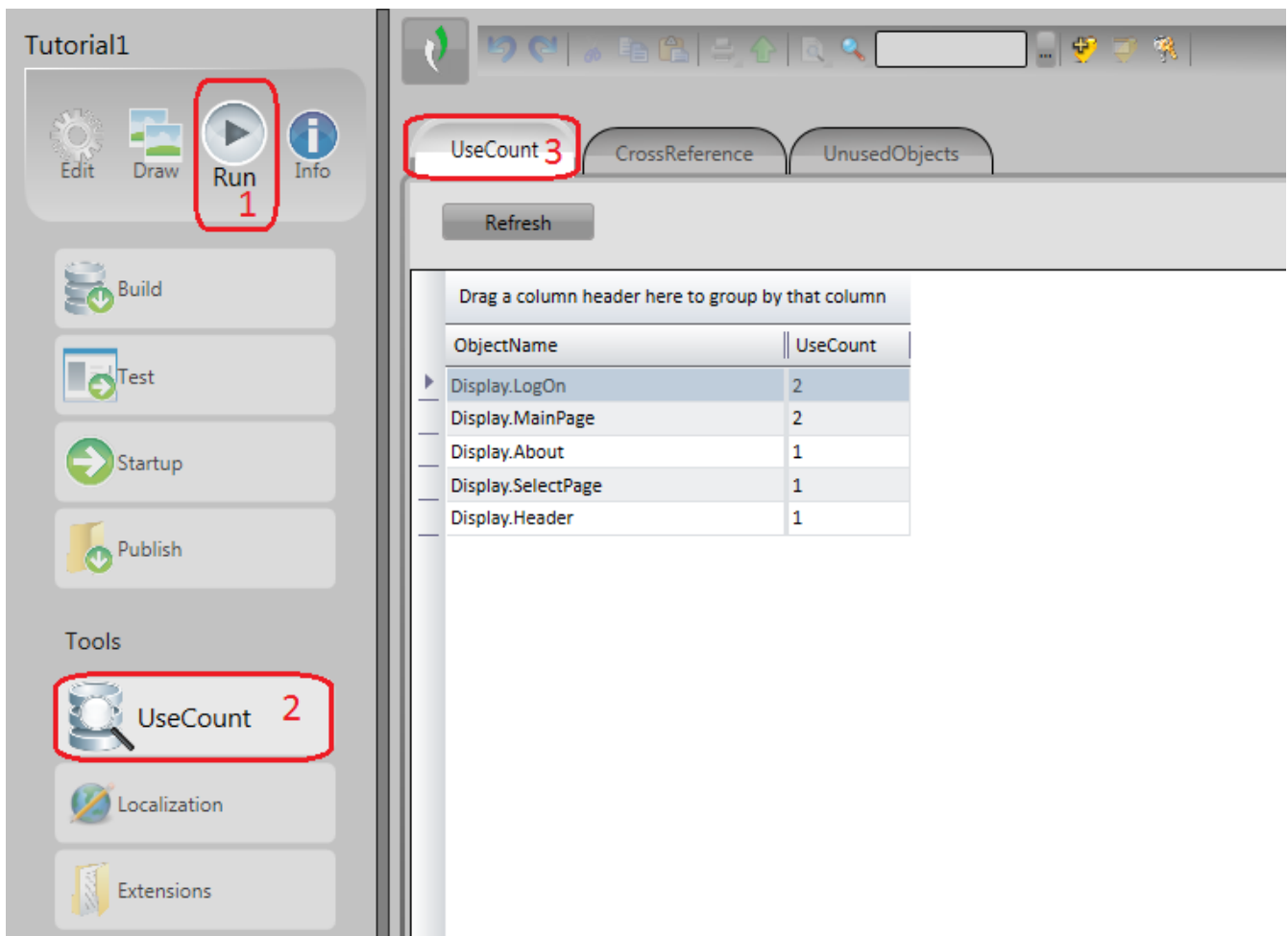


Figura 5-6. Menu UseCount

Os itens que compõem o menu *UseCount* estão numerados na figura anterior e são descritos na sequência.

1. Selecionar a opção *Run*;
2. Selecionar a opção *UseCount*;
3. Selecionar a opção *UseCount* para exibição das informações sobre o uso dos objetos no projeto.

CrossReference

Na guia *CrossReference* são apresentados os objetos do projeto e onde estes são usados.

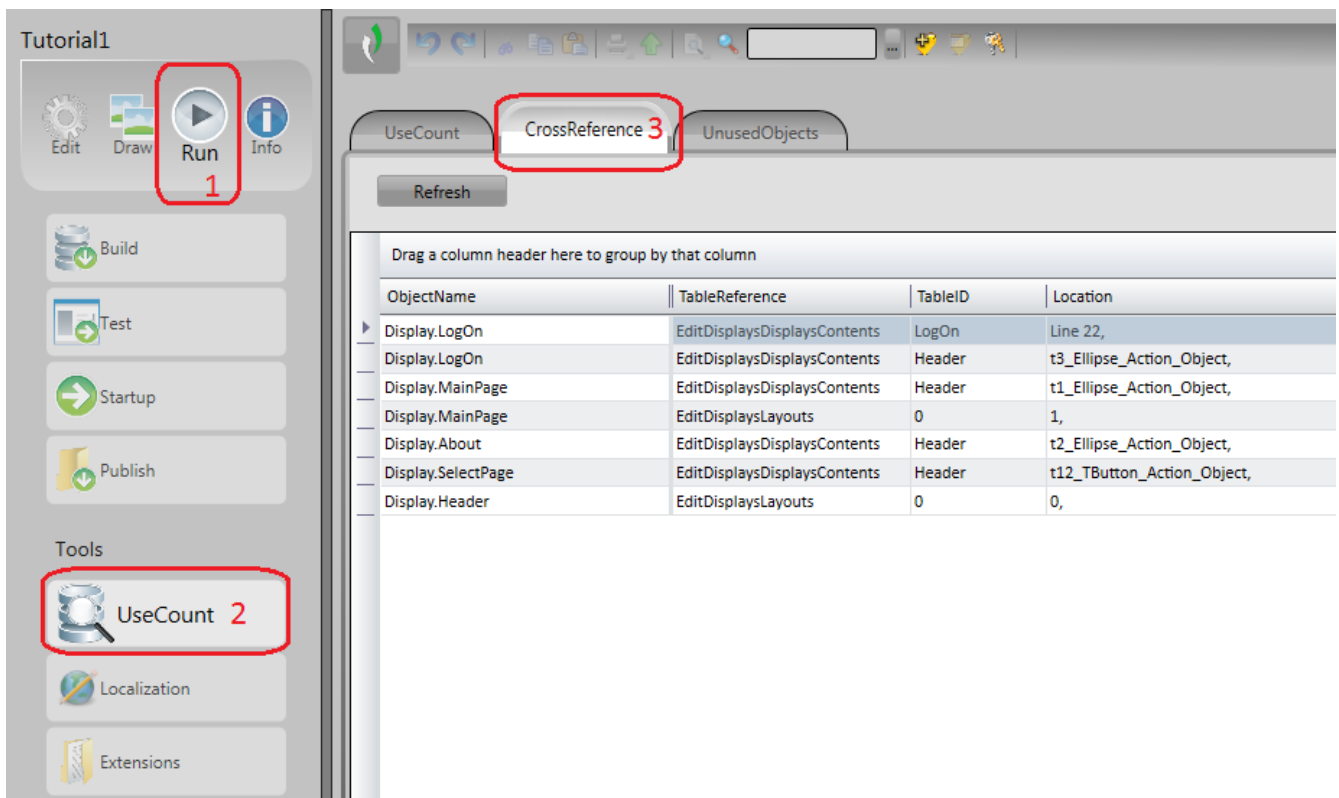


Figura 5-7. Menu CrossReference

Os itens que compõem o menu *CrossReference* estão numerados na figura anterior e são descritos na sequência.

1. Selecionar a opção *Run*;
2. Selecionar a opção *UseCount*;
3. Selecionar a opção *CrossReference* para exibição das informações sobre o uso dos objetos no projeto.

UnusedObjects

Na guia *UnusedObjects* são apresentados os objetos do projeto que não estão sendo usados

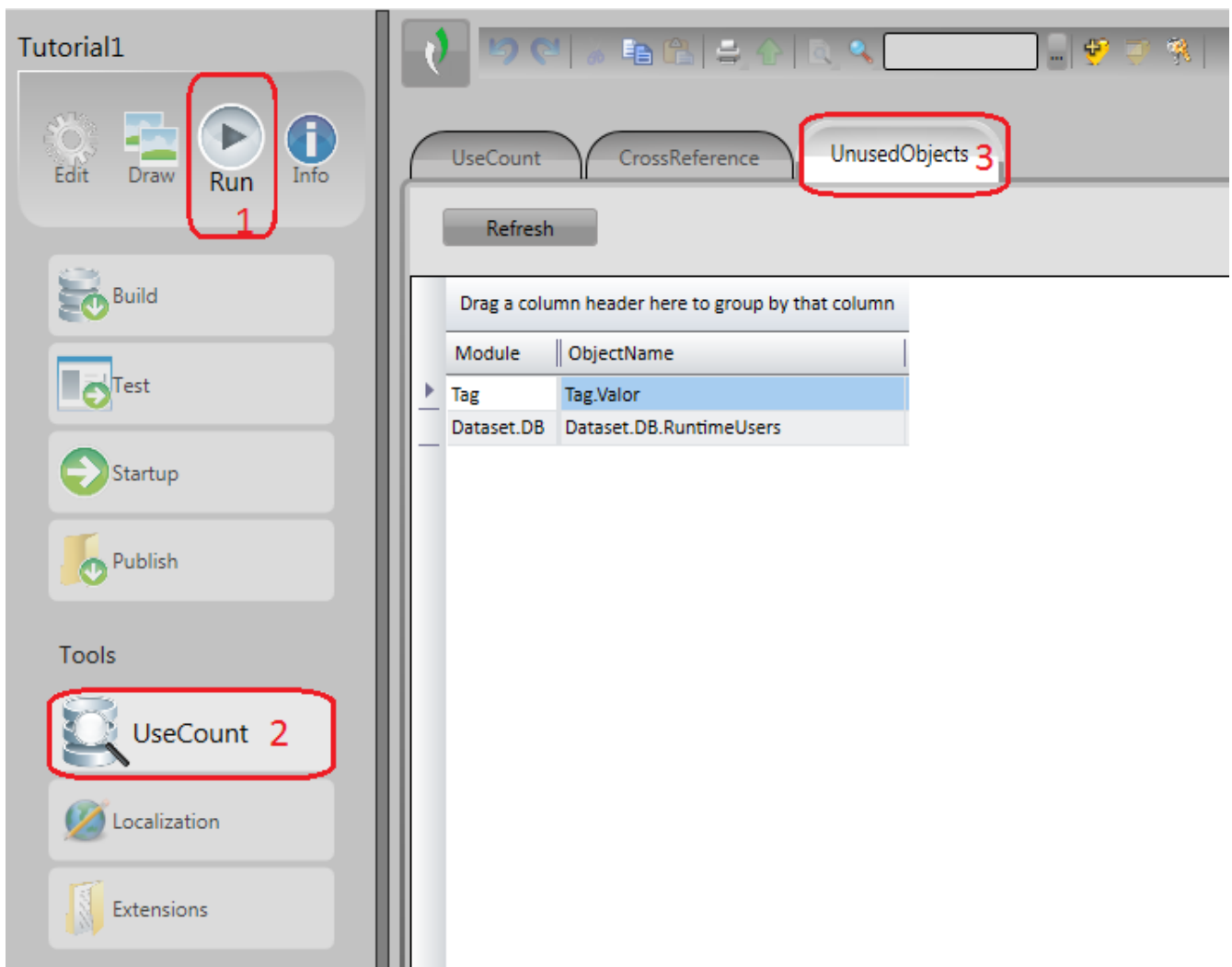


Figura 5-8. Menu UnusedObjects

Os itens que compõem o menu *UnusedObjects* estão numerados na figura anterior e são descritos na sequência.

1. Selecionar a opção *Run*;
2. Selecionar a opção *UseCount*;
3. Selecionar a opção *UnusedObjects* para exibição das informações sobre o uso dos objetos no projeto.

Dictionaries

Localization

O menu *Localization* possui a possibilidade de configurar mais de um idioma para ser usado em mensagens de alarme e nos textos de display.

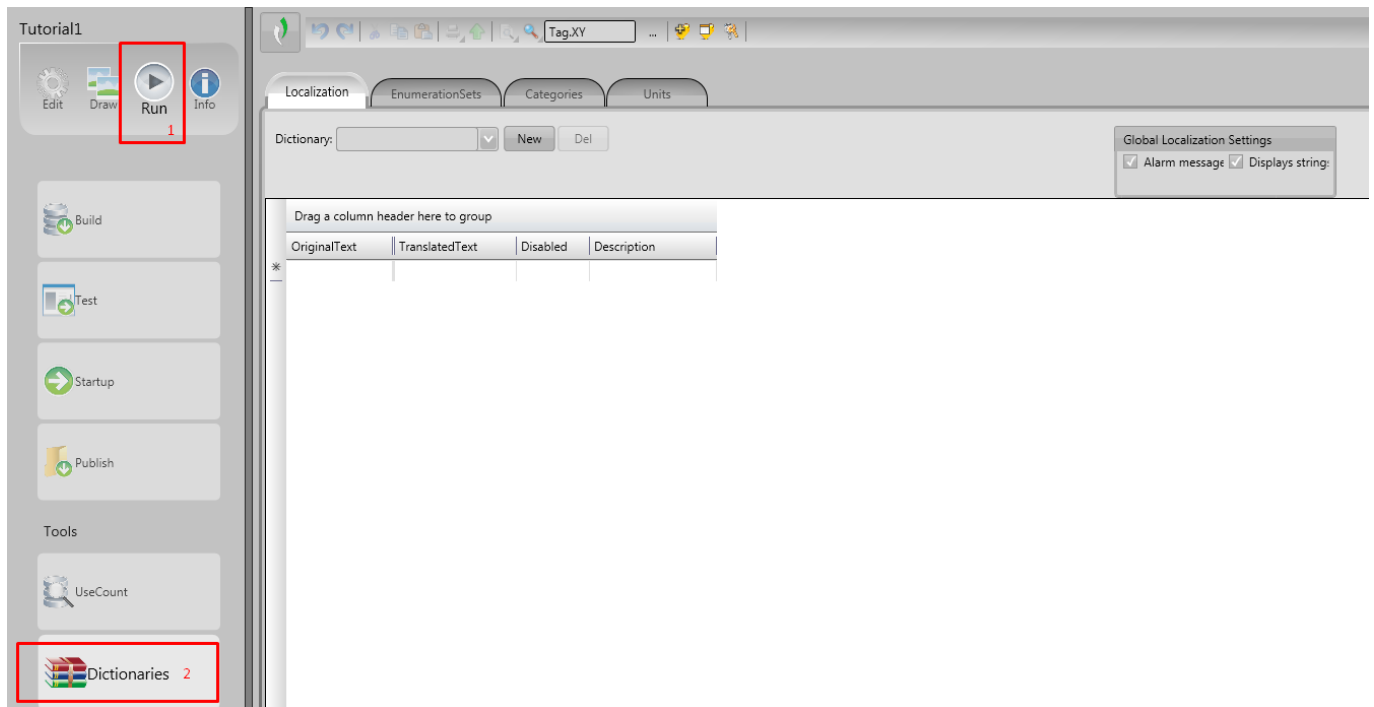


Figura 5-9. Menu Localization

Os itens que compõem o menu *Localization* estão numerados na figura anterior e são descritos na sequência.

1. Selecionar a opção *Run*;
2. Selecionar a opção *Localization* para exibição das informações.

EnumerationsSets

Você também pode usar os dicionários para traduzir os valores das Tags em texto. Por exemplo, um dispositivo pode ter um valor de 0 (desligado) ou 1 (ligado). Ao invés de exibir "0" ou "1", você pode exibir as palavras "Ligado" ou "Desligado".

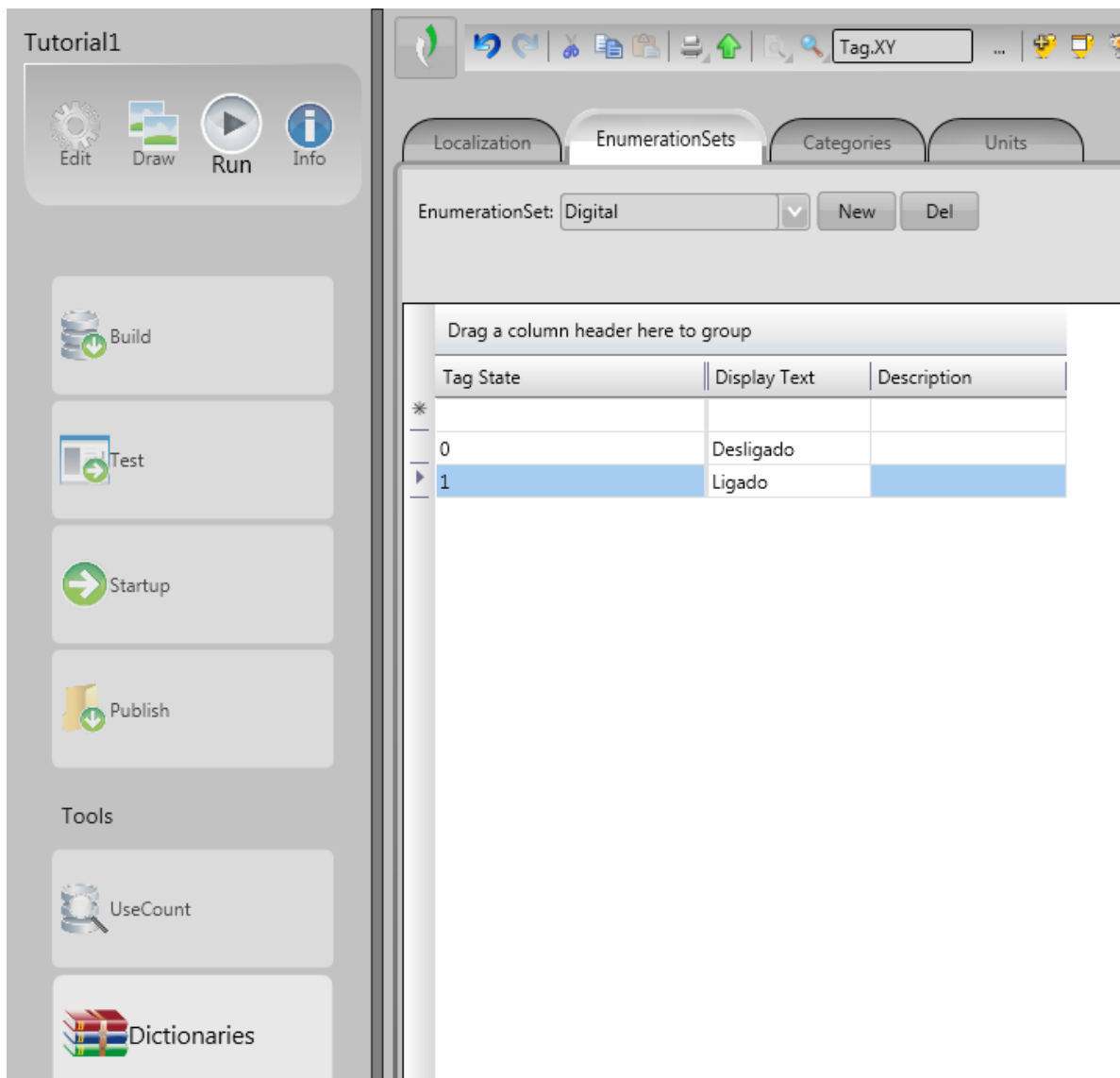


Figura 5-10. Menu EnumerationSets

Para configurar conjuntos de enumeração:

1. Vá para Run > Dictionaries> Enumeration Sets.
2. Clique New. Aparecerá a janela Criar Novo Dicionário.
3. Digite um nome para o dicionário.
4. Clique em OK.
5. Na coluna Valor, digite o valor numérico, e na coluna Texto traduzido, digite o texto a ser exibido.
6. Para as Tags que usam os valores numéricos com o Conjunto de Enumeração, configurar a coluna Parâmetros para usar este conjunto.

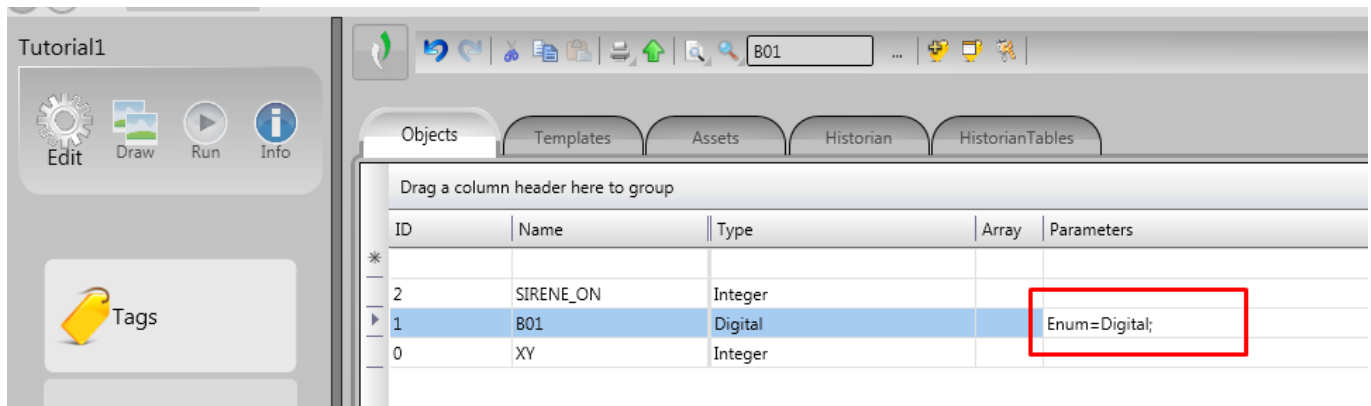


Figura 5-11. Configuração dos parâmetros

7. Ao configurar um componente de uma tela para usar estas Tags deve-se definir o TextOutput para um dos seguintes atributos:

- Com *TagName* ou *Tagname.Value* para mostrar o valor numérico da Tag.
- Com *Tagname.ValueAsString* ou *Tagname.DisplayValue* para mostrar o texto do conjunto de enumeração.

Categories

Um Tag pode pertencer a uma categoria pré-definida de objetos. Por exemplo: Tags de transformadores, Tags de alimentadores e Tags de proteção podem ser categorias. As categorias devem ser pré-definidas no Dicionário de Categorias.

Você pode criar Categorias para permitir a classificação vários tipos de objetos da Engenharia. A maioria das tabelas de configuração, como Tags, Nós de Dispositivos, Grupos de Alarme, têm a coluna Categoria, onde você pode aplicar uma ou mais categorias para esse objeto.

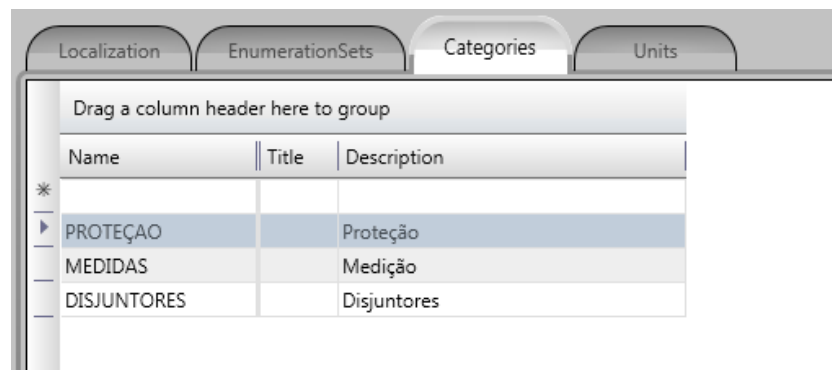


Figura 5-12. Menu Categories

Uma mesma Tag ou objeto pode pertencer a várias categorias.

Por exemplo: suponha que foram definidas duas categorias: TAGS_DE_TRAFOS e TAGS_DE_MEDIDAS. Poderíamos querer pesquisar, dentro de um código de script, as Tags que são de pontos de medição de tensão nos transformadores. No cadastramento destes pontos especificaríamos no campo *Categories* destas Tags estas duas categorias.

Uma vez a que a categoria foi criada, passará a estar disponível na coluna específica da categoria na aba *Object*, para a criação de Tags. Para cada Tag escolha todas em que ela se enquadra.

As categorias de objeto também podem ser acessadas durante o tempo de execução para a filtragem e outras aplicações personalizadas, utilizando a propriedade *Categoria* dos objetos, como critérios para a escolha de objetos. Uma lista de todas as categorias definidas no projeto está disponível em tempo de execução na propriedade *Server.Categories*.

Extensions

O menu *Extensions* possui a possibilidade de importar dados de configurações para OPC Servers, tabelas de configuração, ou para um projeto inteiro.

Import

A guia *Import* permite realizar importações de componentes do projeto. Nesta guia são encontrados assistentes para importações a partir de projetos do BluePlant, do OPC Server e de arquivos CSV.

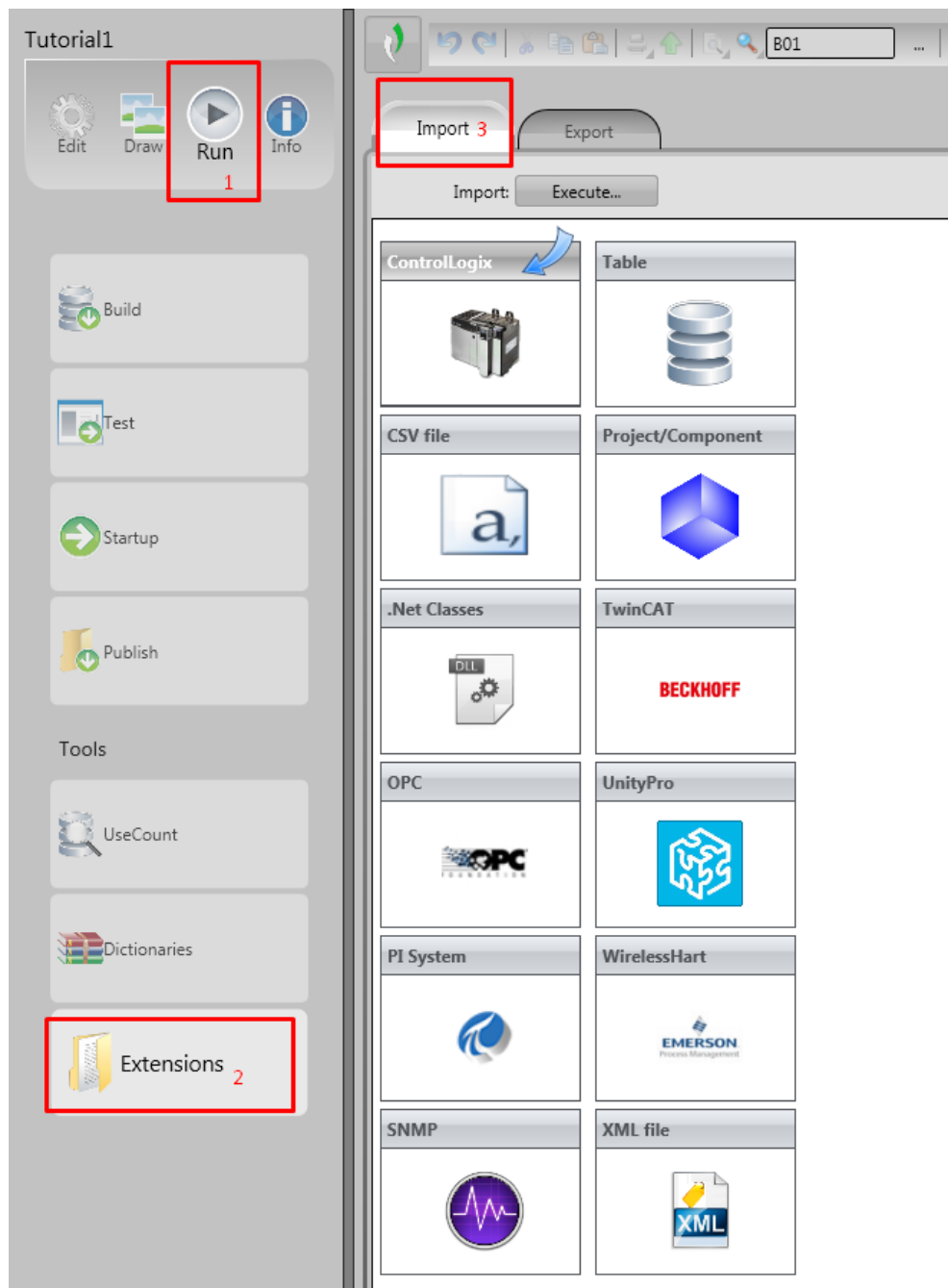


Figura 5-13. Menu Import

Os itens que compõem o menu *Import* estão numerados na figura anterior e são descritos na sequência.

1. Selecionar a opção *Run*;
2. Selecionar a opção *Extensions*;
3. Selecionar a opção *Import* para visualizar as opções de importação.

Export

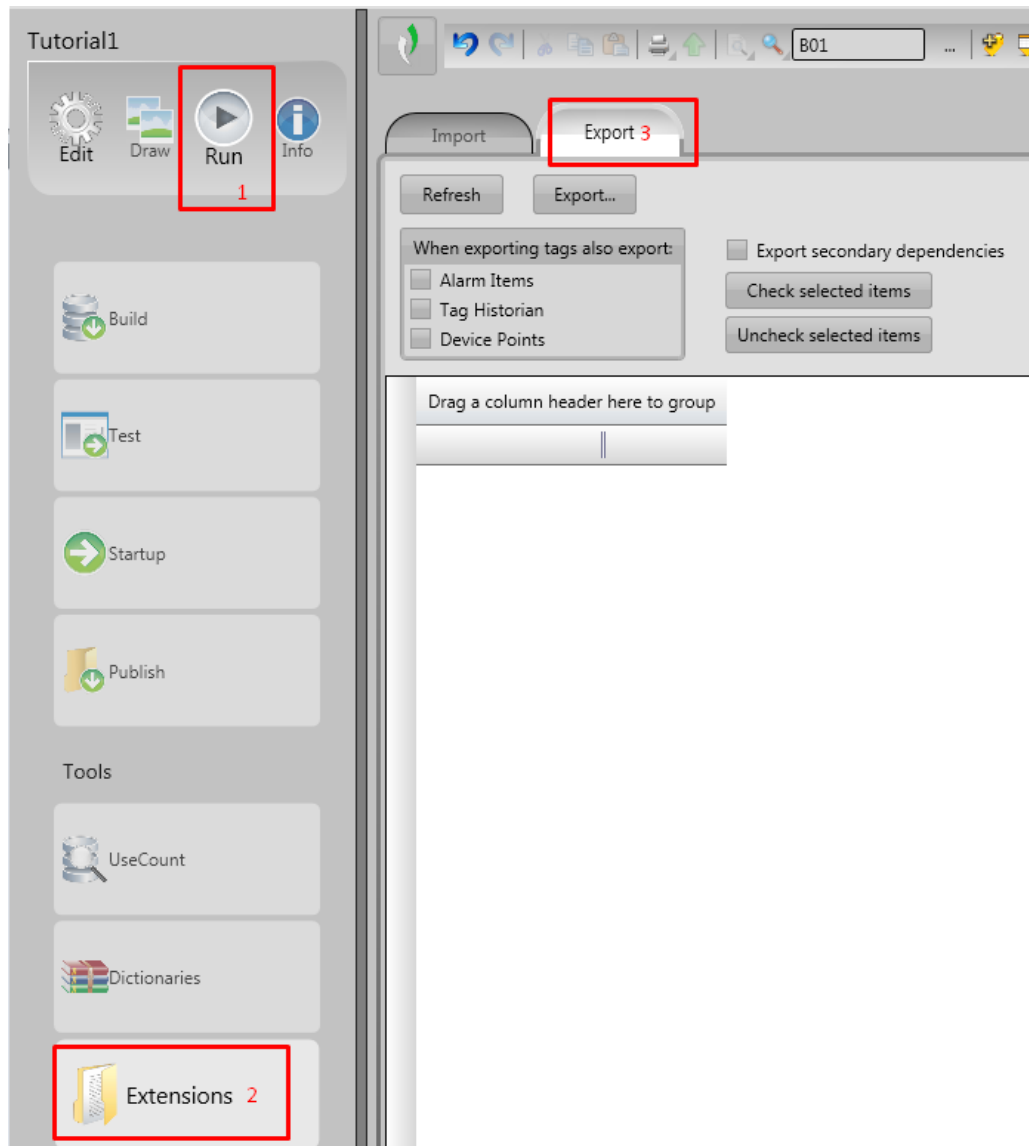


Figura 5-14. Menu Export

Os itens que compõem o menu *Export* estão numerados na figura anterior e são descritos na sequência.

1. Selecionar a opção *Run*;
2. Selecionar a opção *Extensions*;
3. Selecionar a opção *Export* para visualizar as opções de exportação.

Informações da Aplicação



Project

Submenu de informações do projeto atual.

Version

A guia *Version*, ilustrada na figura a seguir, contém informações sobre o projeto atual. Alguns campos de informação também estão disponíveis durante o *Runtime* usando *Namespace Info*.

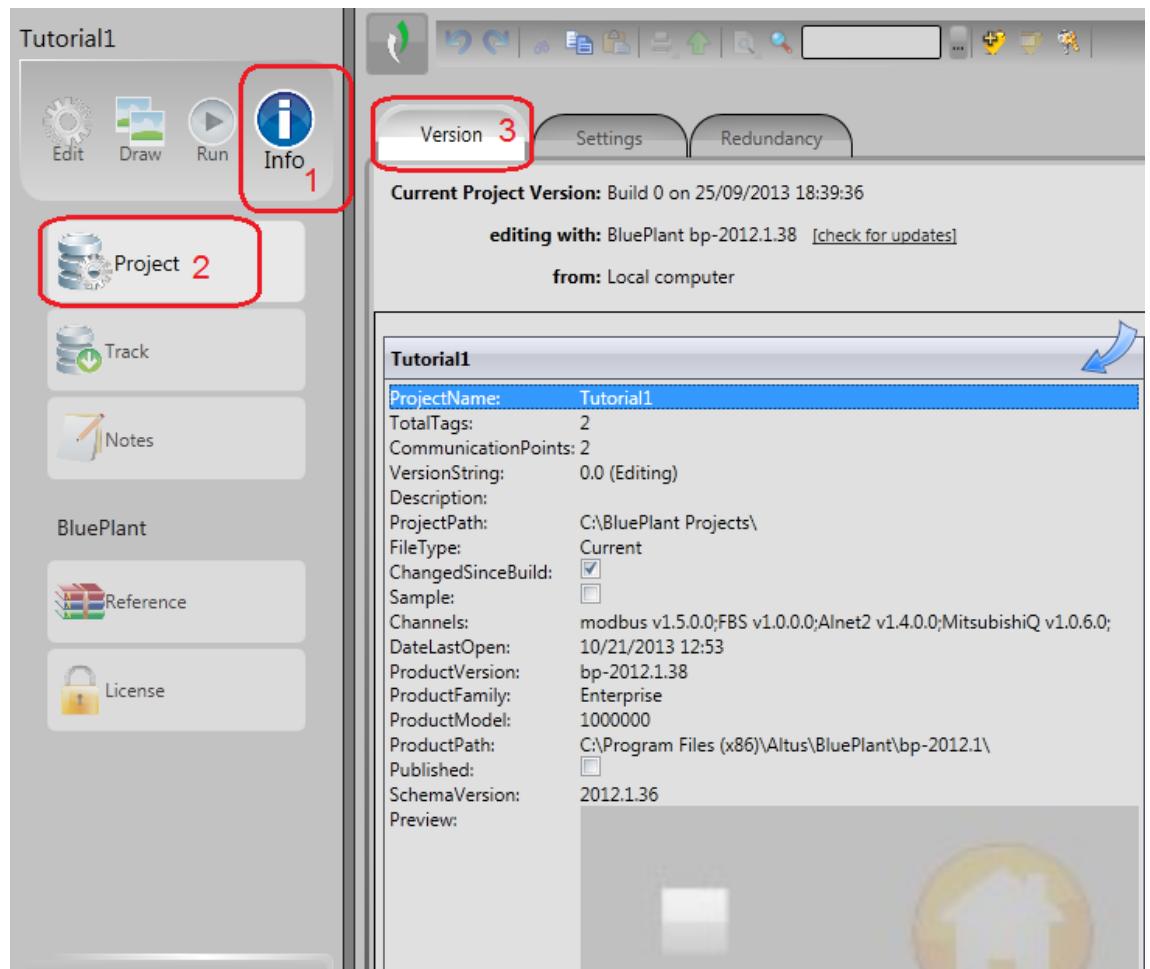


Figura 5-15. Guia Version

Os itens que compõem o menu *Version* estão numerados na figura anterior e são descritos na sequência.

1. Selecionar a opção *Info*;
2. Selecionar a opção *Project*;
3. Selecionar a opção *Version* para exibição dos dados relativos ao projeto.

Settings

Na sequência são explicados os itens da guia *Settings* conforme ilustrado na figura a seguir. Nesta guia são definidas as configurações globais para o projeto. Alguns campos de informação também estão disponíveis durante o *Runtime* usando *Namespace Info*.

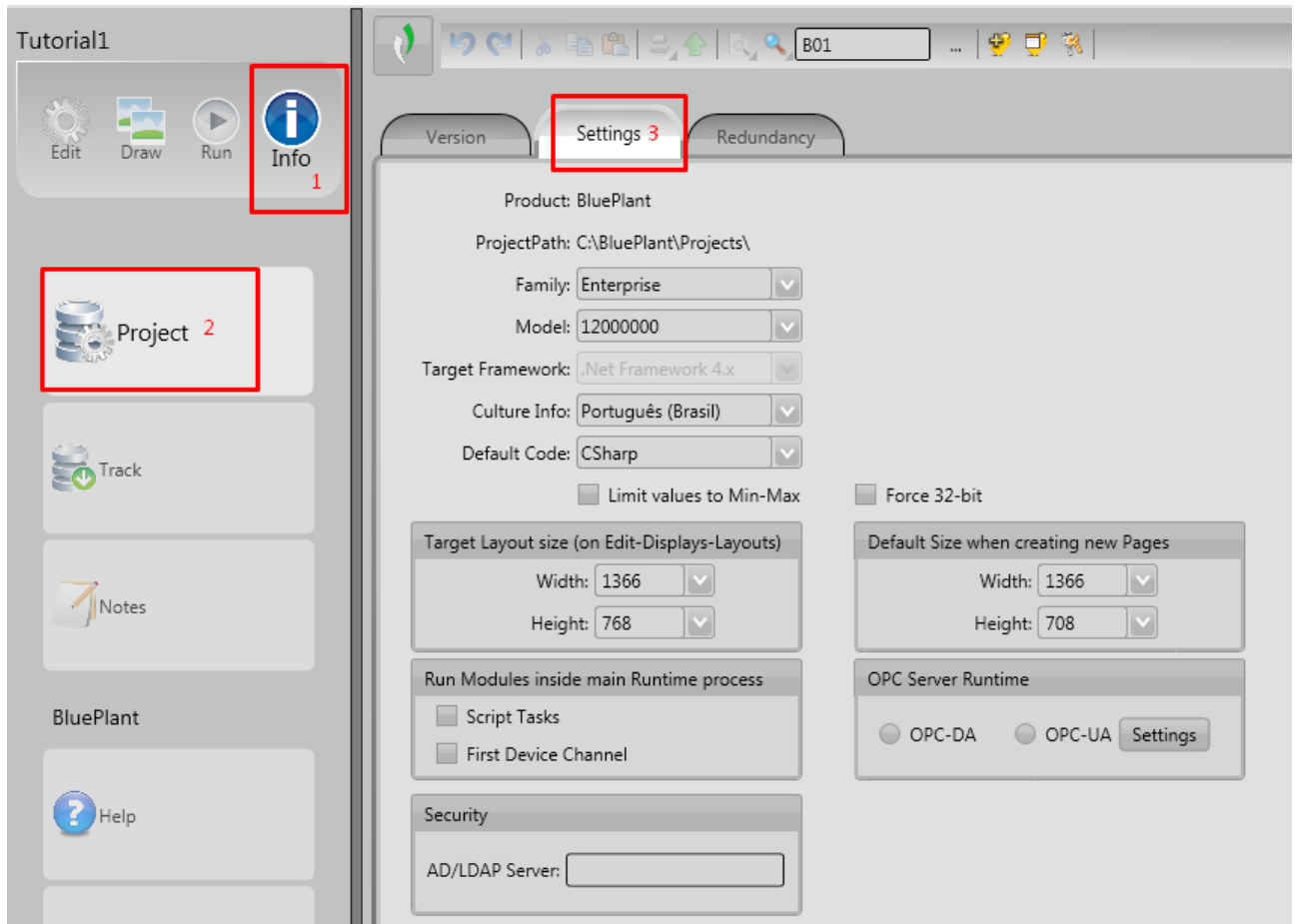


Figura 5-16. Guia Settings

Os itens que compõem a guia *Settings* estão numerados na figura anterior e são descritos na sequência.

1. Selecionar a opção *Info*;
2. Selecionar a opção *Project*;
3. Selecionar a opção *Settings*.

Project Path: Local onde o projeto aberto está salvo no computador.

Family: Seleciona o Modelo de BluePlant correspondente.

Model: Seleciona o modelo de *Runtime*.

Culture Info: Seleciona o idioma para o projeto em tempo de execução. Esta configuração controla a formatação de datas, números e outras informações conforme o idioma selecionado.

Target Framework: Seleciona o framework, plataforma no qual o BluePlant executa.

Default Code: Seleciona a linguagem do Script. As seguintes Linguagens podem ser selecionadas: VB.Net e CSharp.

Limit values to Min-Max: Se esta caixa estiver marcada o sistema não permitirá colocar valores fora do intervalo Min-Max de cada Tag.

Force 32-bit: Se esta caixa estiver marcada o sistema irá executar em ambiente 32 bits.

Target Layout size (on Edit-Displays-Layouts): Define a resolução padrão para novos layouts a serem criados no projeto. Isso não afeta as páginas existentes.

Figura 5-17. Ajuste do Tamanho Padrão para novos Layouts

Default size when creating new pages: Define a resolução padrão para novas telas a serem criados no projeto. Isso não afeta as páginas existentes.

Figura 5-18. Ajuste do Tamanho Padrão para novas Páginas

Run modules inside realtime process: Se esta caixa for marcada, executa módulos dentro do módulo *TServer: Script Tasks* (tarefa de Script) e *First Device Channel* (executável do primeiro canal do dispositivo).

Figura 5-19. Ajuste de módulos dentro do TServer

OPC Server Runtime: Define o tipo de OPC a ser executado (OPC-DA ou OPC-UA).

Figura 5-20. Ajuste do protocolo OPC Server

Security: Define configurações de segurança usando servidor AD/LDAP.

Figura 5-21. Ajuste do Security

Redundancy

Na sequência são explicados os itens da guia *Redundancy* conforme ilustrado na figura a seguir.

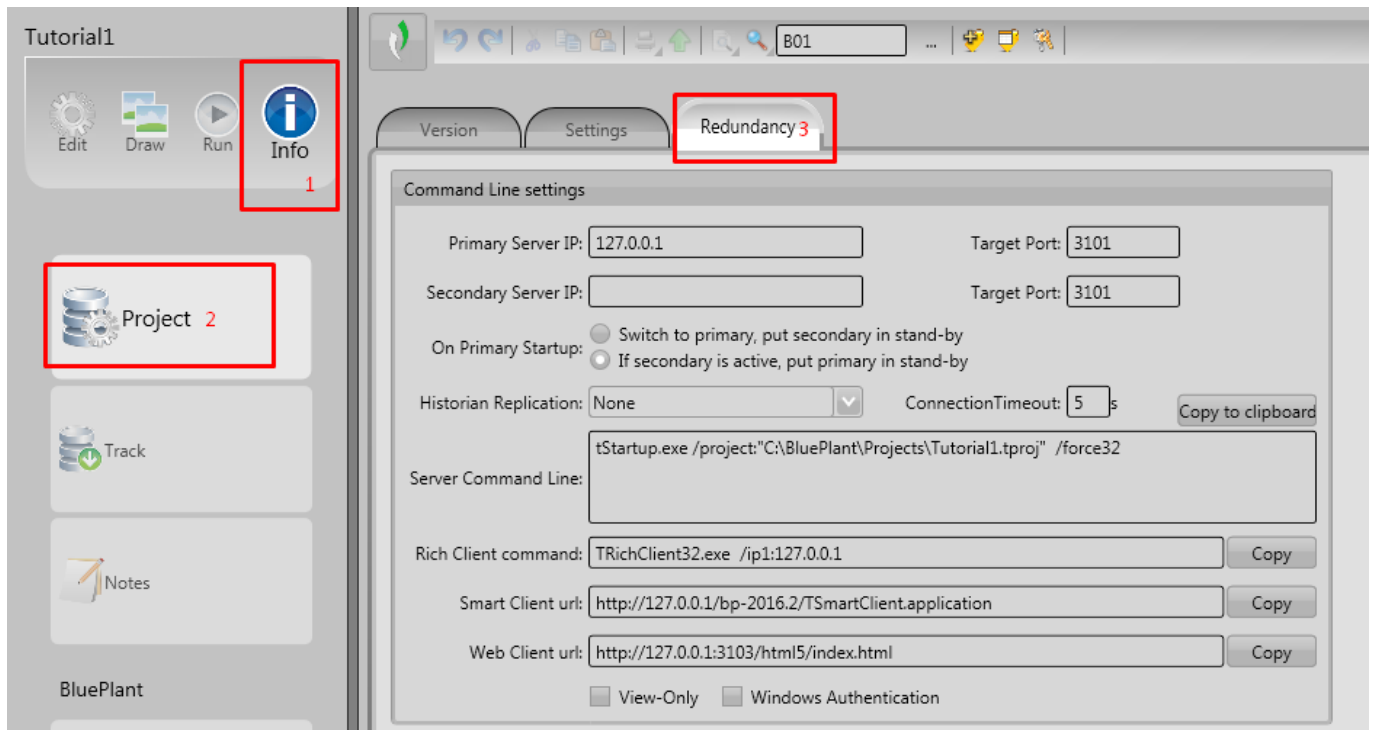


Figura 5-22. Guia Redundancy

Primary Server IP: Campo para configuração do endereço IP do servidor denominado primário.

Secondary Server IP: Campo para configuração do endereço IP do servidor denominado secundário.

Target Port: Número da porta que o servidor irá utilizar para comunicar com os outros módulos e servidores.

On Primary Startup: Define o comportamento do servidor primário quando este for iniciado. As opções estão indicadas a seguir.

- *Switch to primary, put secondary in standy-by:* alterna para o servidor primário colocando o secundário em espera.
- *If secondary is active, put primary in stand-by:* se o servidor secundário está ativo coloca o primário em espera.

Historian Replication: Define quais históricos irão fazer replicação de dados automáticos. As opções são: *None* (nenhum), *Alarm Historian* (histórico de alarmes), *Tag Historian* (histórico de Tags) e *Alarm and Tag Historian* (histórico de alarmes e Tags).

Connection Timeout: Tempo especificado que o servidor reserva para aguardar quando não há comunicação entre os servidores redundantes e assume como ativo.

Server Command Line: Linha de comando utilizada para iniciar os servidores.

Rich Client Command: Linha de comando utilizada para iniciar os clientes .NET.

Smart Client Command: Linha de comando utilizada para iniciar os clientes smart.

Web Cliente url: Linha de comando utilizada para iniciar os clientes HTML5.

Configuração da redundância de servidores: a figura a seguir exemplifica a configuração de redundância de servidores. Para realizar esta operação é necessário consultar a **NAP157 - Redundância de Servidores SCADA com BluePlant**.

The screenshot shows the 'Redundancy' configuration window. At the top, there are three tabs: 'Version', 'Settings', and 'Redundancy'. The 'Redundancy' tab is selected. Below the tabs, the window is titled 'Command Line settings'. It contains several configuration fields:

- Primary Server IP:** 192.168.16.87
- Secondary Server IP:** 192.168.16.86
- Target Port:** 3101 (for both primary and secondary)
- On Primary Startup:** Two radio buttons. The first is 'Switch to primary, put secondary in stand-by' (unselected). The second is 'If secondary is active, put primary in stand-by' (selected).
- Historian Replication:** A dropdown menu set to 'Alarm and Tag Historian'.
- ConnectionTimeout:** 5 s
- Copy to clipboard:** A button next to the ConnectionTimeout field.
- Server Command Line:** A text area containing the command: `tStartup.exe /project:"C:\BluePlant\Projects\Tutorial1.tproj" /redundancy /ip1:192.168.16.87 /ip2:192.168.16.86 /connectiontimeout:5`
- Rich Client command:** A text field containing: `TRichClient.exe /ip1:192.168.16.87 /ip2:192.168.16.86 /connectiontimeout:5` with a 'Copy' button.
- Smart Client url:** A text field containing: `http://192.168.16.87/bp-2016.2/TSmartClient.application?ip2=192.168.16.86&connectiontimeout=5` with a 'Copy' button.
- Web Client url:** A text field containing: `http://192.168.16.87:3103/html5/index.html` with a 'Copy' button.
- View-Only:** An unchecked checkbox.
- Windows Authentication:** An unchecked checkbox.

Figura 5-23. Exemplo de Configuração de Redundância de Servidores

Track

Tables

Esta guia do menu *Info*, opção *Track*, rastreia o status das tabelas de configuração. A figura a seguir mostra essa seleção.

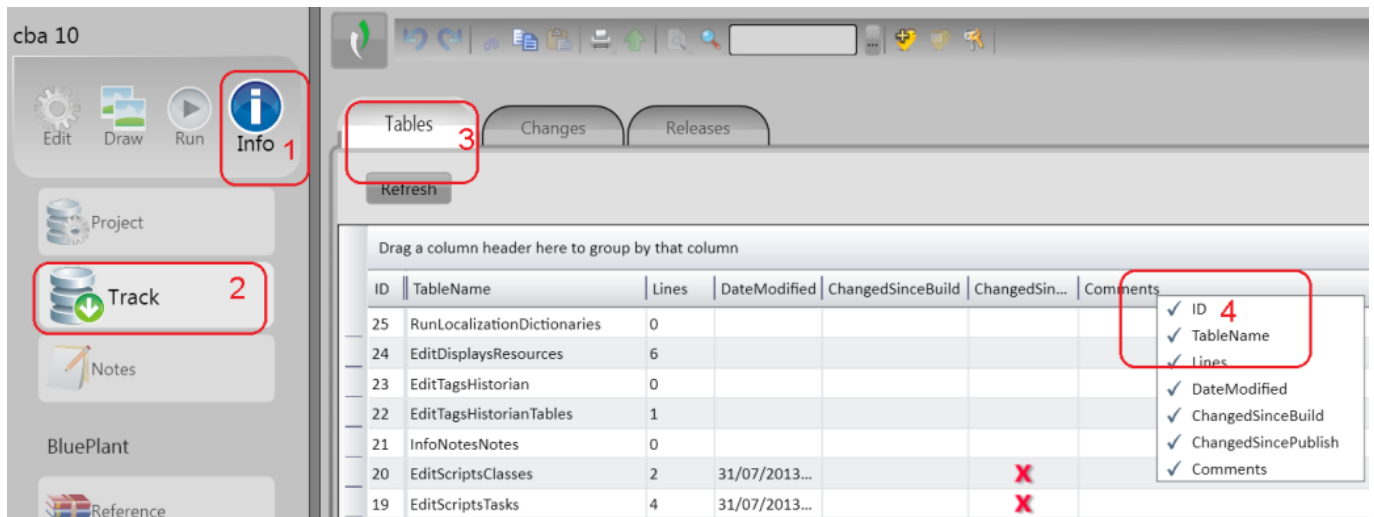


Figura 5-24. Guia Tables do Menu Info, Opção Track

Os itens que compõem o menu *Info > Track > Tables* estão numerados na figura anterior e são descritos na sequência.

1. Selecionar a opção *Info*;
2. Selecionar a opção *Track*;
3. Selecionar a opção *Tables*;
4. Clicar com o botão direito do mouse em qualquer um dos cabeçalhos de coluna e selecionar as opções a serem exibidas.

Os itens da guia *Tables* são explicados a seguir e podem ser visualizados na figura que segue.

TableName: Nome da Tabela do projeto. Atributos: *ReadOnly* (somente leitura).

Lines: Número atual de linhas na tabela de configuração. Atributos: *ReadOnly* (somente leitura).

DateModified: Data da última alteração. Atributos: *ReadOnly*.

ChangedSinceBuild: Alterações desde a última execução do comando de compilação. Atributos: *ReadOnly* (somente leitura).

ChangedSincePublish: Alterações desde a última execução do comando de publicação. Atributos: *ReadOnly* (somente leitura).

Comments: Comentários definidos pelo usuário. Atributos: editável.

Changes

Esta guia do menu *Info* opção *Track*, rastreia as modificações do projeto. A figura a seguir mostra essa seleção. O campo *Tracking* define quando as mudanças serão salvas.

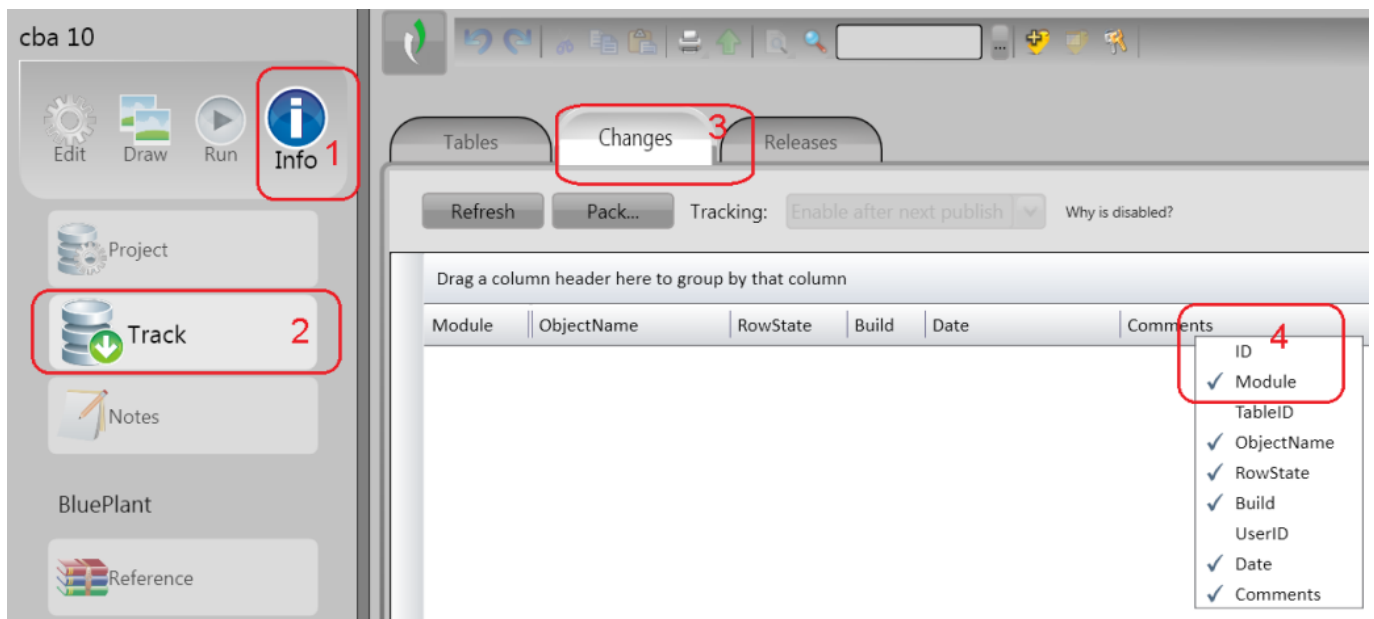


Figura 5-25. Guia Changes do Menu Info, Opção Track

Os itens que compõem o menu *Info > Track > Changes* estão numerados na figura anterior e são descritos a seguir.

1. Selecionar a opção *Info*;
2. Selecionar a opção *Track*;
3. Selecionar a opção *Changes*;
4. Clicar com o botão direito do mouse em qualquer um dos cabeçalhos de coluna e selecionar as opções a serem exibidas.

Module: Nome do módulo onde o objeto está definido. Atributos: *ReadOnly* (somente leitura).

ObjectName: Nome do objeto ou índice de linhas. Atributos: *ReadOnly* (somente leitura).

RowState: Operação executada no objeto. Atributos: *ReadOnly* (somente leitura).

Build: Número de compilação do projeto quando a operação foi executada. Atributos: *ReadOnly* (somente leitura).

Date: Data de modificação da execução. Atributos: *ReadOnly* (somente leitura).

Comments: Comentários definidos pelo usuário. Atributos: editável.

TableId: Indica onde está o objeto dentro do módulo. Atributos: *ReadOnly* (somente leitura).

Releases

Esta guia do menu *Info* opção *Track* rastreia as versões publicadas do projeto. A figura a seguir mostra essa seleção.

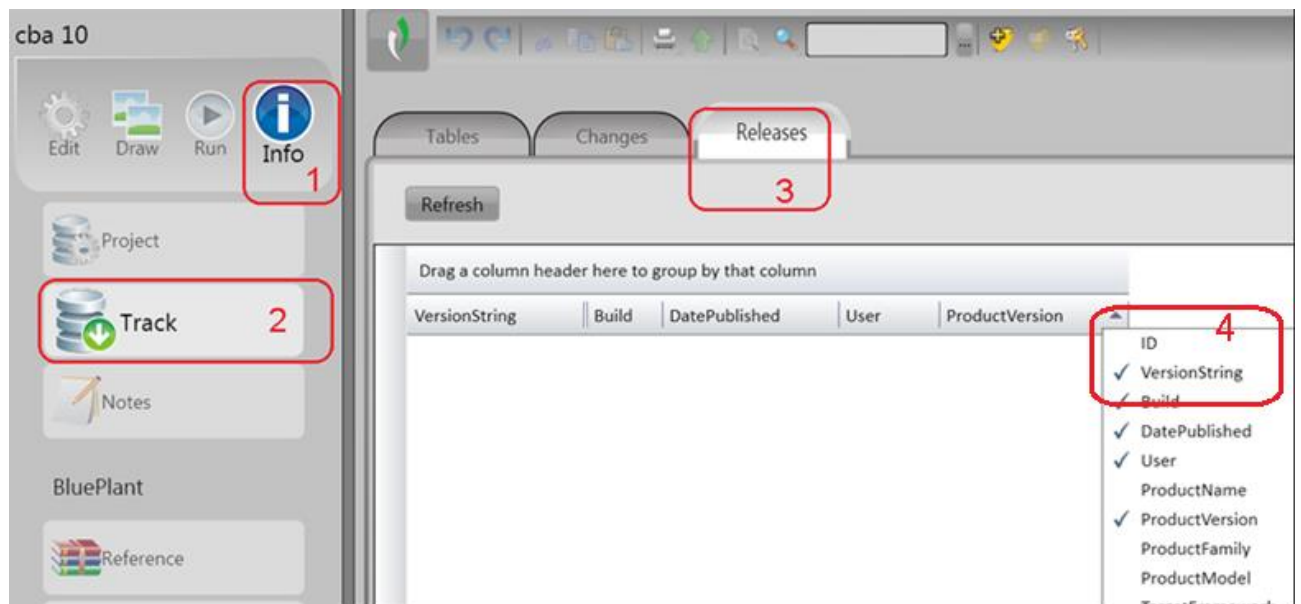


Figura 5-26. Guia Releases do Menu Info, Opção Track

Os itens que compõem o menu *Info* > *Track* > *Releases* estão numerados na figura anterior e são descritos a seguir.

1. Selecionar a opção *Info*;
2. Selecionar a opção *Track*;
3. Selecionar a opção *Releases*;
4. Clicar com o botão direito do mouse em qualquer um dos cabeçalhos de coluna para selecionar as opções a serem exibidas.

Os itens da guia *Releases* estão explicados a seguir e podem ser visualizados na figura que segue.

Versionstring: Versão publicada. Atributos: *ReadOnly* (somente leitura).

Build: Número da compilação quando a versão foi publicada. Atributos: *ReadOnly* (somente leitura).

DatePublished: Data de publicação da versão para este projeto. Atributos: *ReadOnly* (somente leitura).

User: Nome do usuário que publicou esta versão. Atributos: *ReadOnly* (somente leitura).

Description: Descrição da versão Publicada. Atributos: *ReadWrite* (escrita e leitura).

Product Name: Nome do produto no qual foi gerada a versão. No será sempre BluePlant. Atributos: *ReadOnly* (somente leitura).

ProductVersion: Versão do BluePlant na qual a versão foi publicada. *ReadWrite* (escrita e leitura).

ProductFamily: Família de Produto na qual a versão foi publicada (escrita e leitura).

ProductModel: Geralmente apresenta valor *Unlimited* (ilimitado) para indicar licença sem prazo de validade. *ReadWrite* (escrita e leitura).

TargetFramework: Indica a versão do .Net Framework usado para publicação da versão. *ReadWrite* (escrita e leitura).

Notes

Permite a criação de notas e mensagens tipo *PostIt* visualizáveis por todos os membros da equipe de design do projeto. As notas são visíveis na área de trabalho durante a edição do projeto.

Esta interface fornece uma tabela do tipo *quadro de mensagens* para a criação dos lembretes críticos (notas) que são acessadas por todo o pessoal envolvido no desenvolvimento do projeto.

Quando o usuário clica em uma mensagem fechada ela permanece fechada, porém não é excluída. Para excluir uma mensagem selecione a nota na tabela, dê um duplo clique e selecione a opção relacionada à exclusão da linha selecionada. Um clique em *IsOpen* na linha de mensagem faz com que ela seja aberta.

Para implementar mensagens para os operadores de projeto durante a execução do mesmo use a função *OpenPopupNote()* disponível nas opções de objetos *Runtime* do cliente.

Esta guia do menu *Info* opção *Notes* edita notas do usuário sobre a configuração do projeto. A figura a seguir mostra essa seleção.

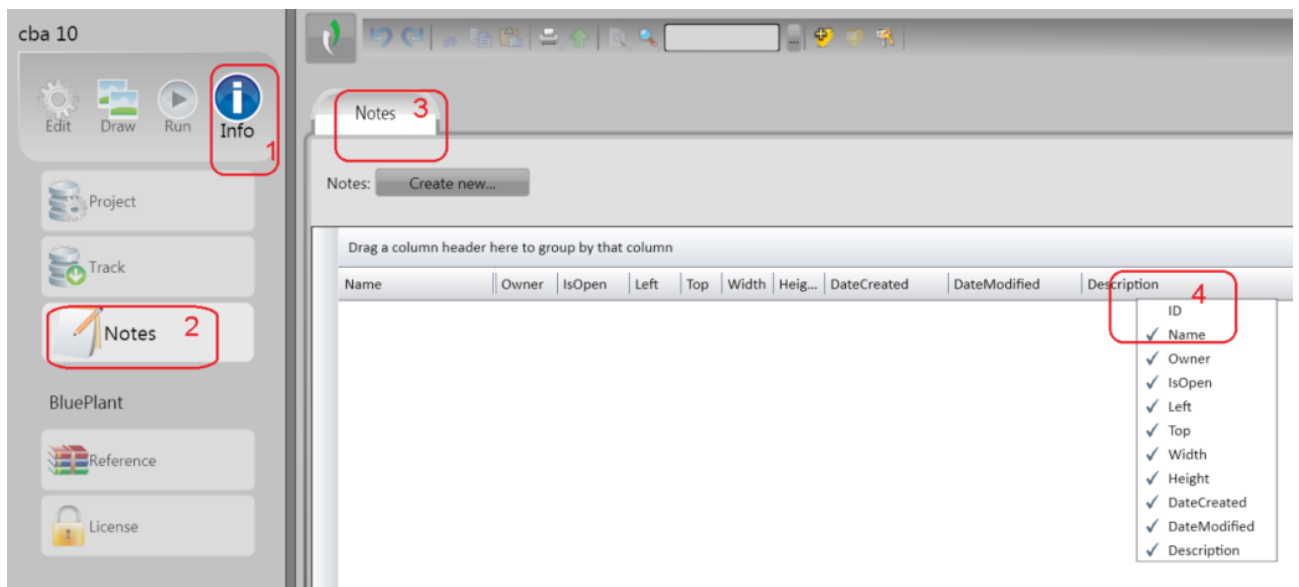


Figura 5-27. Guia Notes do Menu Info Opção Notes

Os itens que compõem o menu *Info > Notes > Notes* estão numerados na figura anterior e são descritos a seguir.

1. Selecionar a opção *Info*;
2. Selecionar a opção *Notes*;
3. Selecionar a guia *Notes*;
4. Clicar com o botão direito do mouse em qualquer um dos cabeçalhos de coluna e selecionar as opções a serem exibidas.

Clique em *Create New* e preencha conforme as descrições abaixo:



Note Name: Título que será escrito no PostIt.


Description: Mensagem que será visualizada no PostIt.


Em seguida na aba *Notes* existem os seguintes itens:


Name: Título que será escrito no PostIt. (somente leitura)


Owner: Usuário que criou o PostIt. (somente leitura)

IsOpen: Situação do PostIt,  está aberto para visualização e  está fechado para visualização. (leitura e escrita)

Left: Posição horizontal do PostIt na tela, quando o mesmo foi fechado pela tecla  do PostIt ou criado com referência de posição no lado esquerdo do monitor. (Somente leitura)

Top: Posição vertical do PostIt na tela, quando o mesmo foi fechado pela tecla  do PostIt ou criado com referência de posição no lado superior do monitor. (Somente leitura)

Width: Largura do PostIt, quando o mesmo foi fechado pela tecla  do PostIt ou criado. (Somente leitura)

Height: Altura do PostIt, quando o mesmo foi fechado pela tecla  do PostIt ou criado. (Somente leitura)

DataCreated: Data e hora em que o PostIt foi criado. (Somente leitura)

DataModified: Data e hora em que o PostIt foi modificado. (Somente leitura)

Description: Mensagem que será visualizada no PostIt. (leitura e escrita)

Ferramentas da Aplicação



Esta seção explora as ferramentas disponíveis no BluePlant. São elas: *TStartup*, *PropertyWatch*, *TraceWindow* e *ModuleInformation*.

Tstartup

Ao executar o projeto usando o gerenciador ou interfaces de boas-vindas, a janela TStartup aparece conforme ilustrado na figura a seguir. Ela contempla informações sobre Status dos Módulos (rodando, parado ou pausado). Pode-se chamar a janela de Startup usando a linha de comando. Os comandos associados são:

- Botão *Play*: inicia a execução do módulo;
- Botão *Stop*: interrompe a execução de um módulo, fecha as conexões e libera os recursos;
- Botão *Pause*: pausa a execução do módulo, geralmente usado pelo recurso de redundância de servidores;
- Botão *Shutdown*: para o servidor BluePlant (TServer) e todos os módulos;
- Botão *Watch*: inicia o *PropertyWatch*;
- Botão *Trace*: inicia o *TraceWindow*;
- Botão *Info*: inicia o *ModuleInformation*.

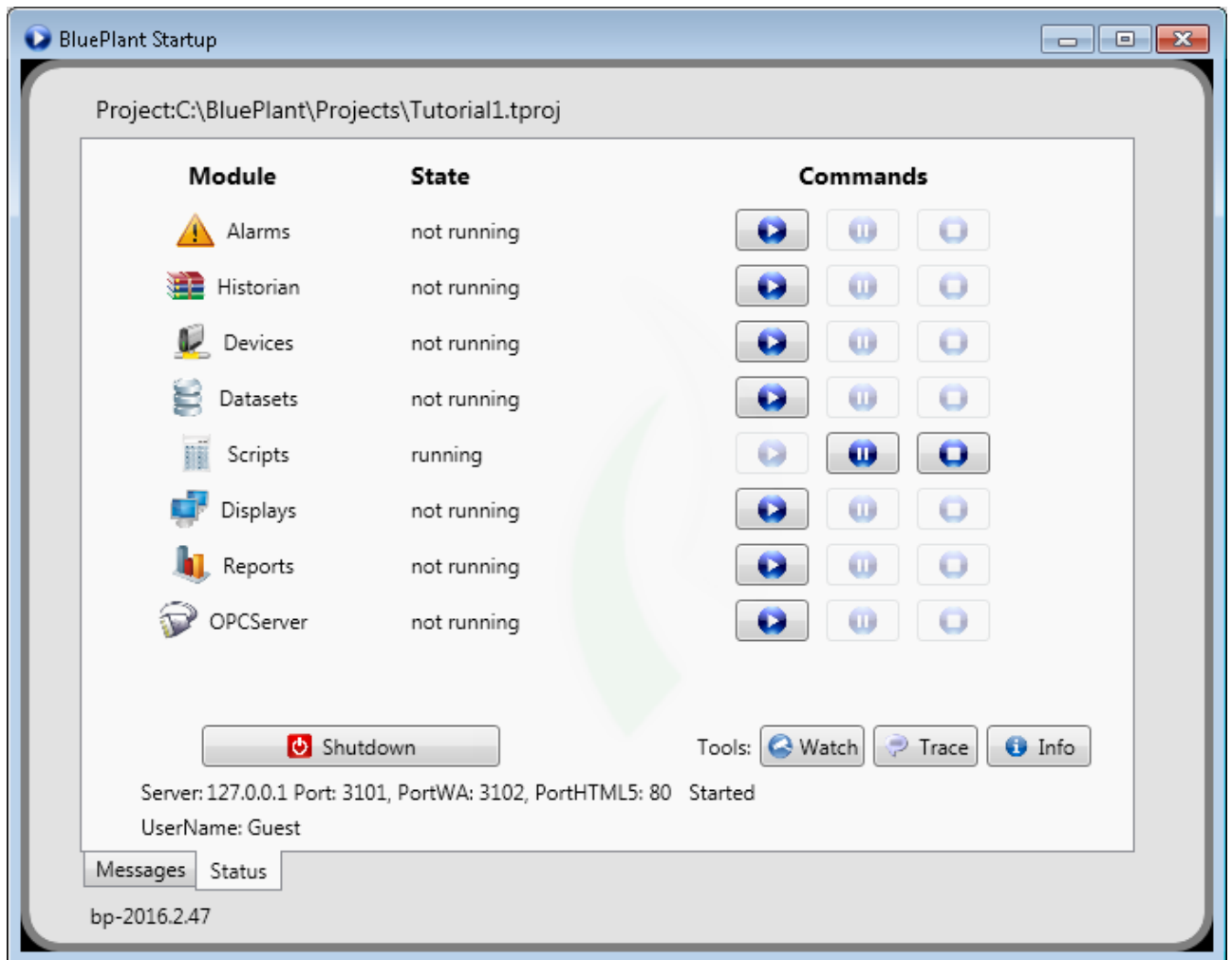


Figura 5-28. Janela de Startup

Estudo Dirigido 5-1: atalho do TStartup e diferentes tipos de clientes



Com o auxílio do instrutor crie um atalho do projeto para *Tstartup*, inicialize um projeto junto com o Windows e inicie diferentes tipos de Clientes com atalhos.

• **DICA:** caso você queira saber mais sobre atalhos do *Tstartup* e diferentes tipos de clientes, assista o Webinar a seguir. Nele você vai aprender a criar o atalho do projeto para *Tstartup*, inicializar um projeto junto com o Windows e a iniciar diferentes tipos de clientes com atalhos.



PropertyWatch

O *PropertyWatch* pode ser usado para acessar o servidor de domínio de Tags e propriedades e mudar seus valores conforme ilustrado na figura a seguir. Pode-se chamar o *PropertyWach* usando a linha de comando.

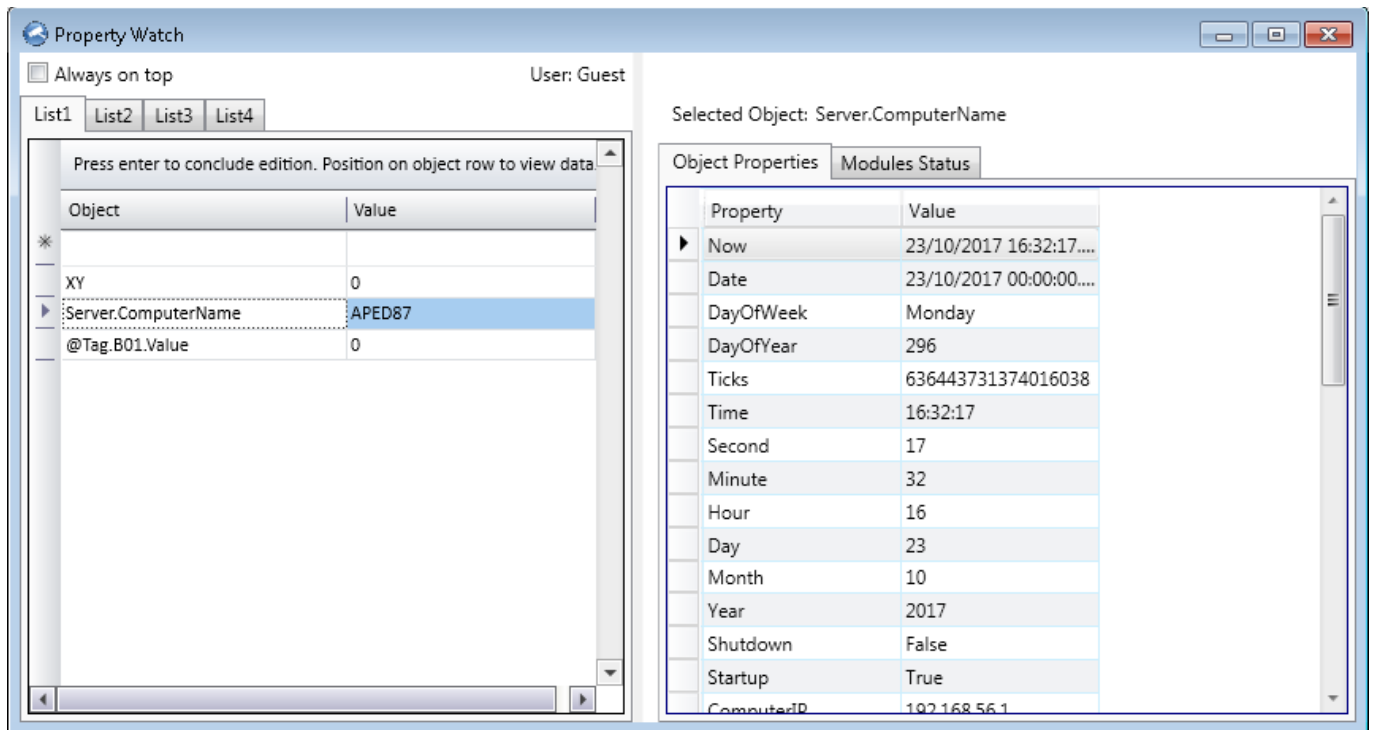


Figura 5-29. PropertyWatch

TraceWindow

A *TraceWindow* é usada para capturar as mensagens dos módulos de *Runtime* do BluePlant conforme ilustrado na figura a seguir. Estas mensagens são úteis para compreender o comportamento em *Runtime*.

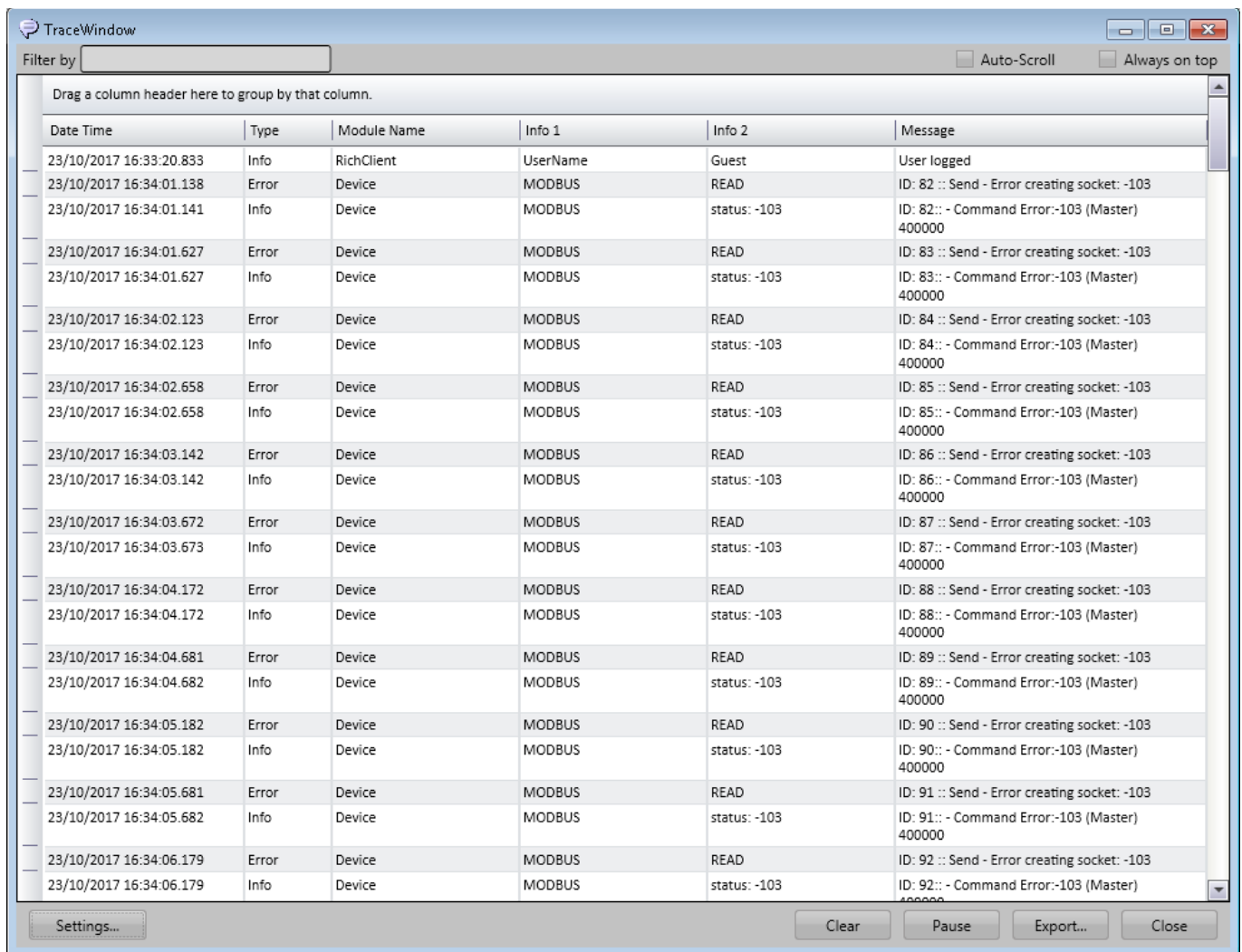


Figura 5-30. TraceWindow

As opções de ajuste mostradas na figura a seguir são:

- *Modules*: determina de quais módulos a *TraceWindow* recebe mensagens;
- *Type*: determina os tipos de mensagem que a *TraceWindow* recebe.

Pode-se adicionar uma Tag ou propriedade no campo *ObjectName*, de maneira que, ao clicar no botão *Add*, uma notificação será enviada para a *TraceWindow* sempre que o objeto mudar o seu valor. Pode-se chamar a *TraceWindow* usando a linha de comando e parâmetros específicos. Para enviar mensagens para a *TraceWindow* do projeto, deve-se chamar a função *Info.Trace(string str)*.

As saídas são:

- *Grid*: indica o número máximo de arquivos que aparecem na grade;
- *File*: um nome de arquivo pode ser definido para armazenamento das mensagens.

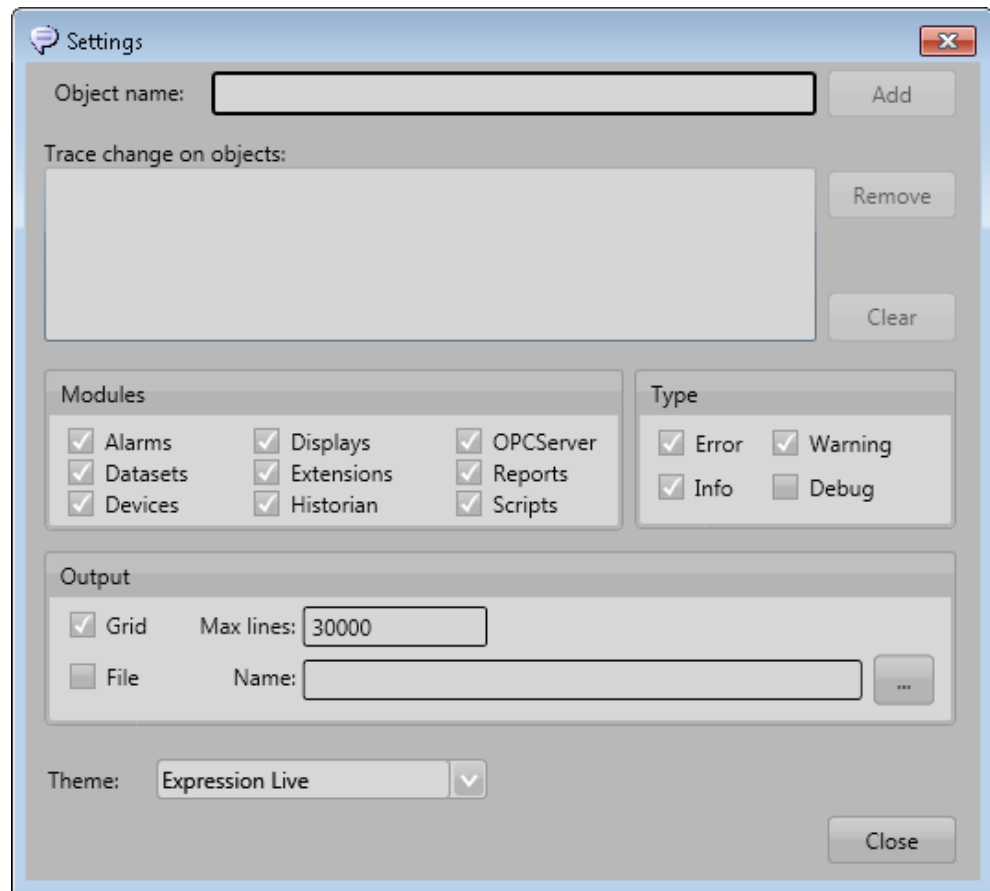


Figura 5-31. Configurações

TraceWindow Message

Seleciona os tipos de mensagem da janela de rastreamento. Os seguintes tipos de mensagem da janela de rastreamento estão disponíveis: Erro, Informação, Advertência e Depuração.

ModuleInformation

ModuleInformation contém informações avançadas sobre a execução dos módulos conforme ilustrado na figura a seguir. Pode-se chamar o *ModuleInformation* usando a linha de comando e parâmetros específicos.

The screenshot shows a software window titled "Module Information". On the left is a tree view with the following structure:

- TCPServer [SERVER]
 - Report [127.0.0.1]
 - Dataset [127.0.0.1]
 - Alarm [SERVER]
 - Device Module [127.0.0.1]
 - modbus
 - Information (selected)
 - Advanced
 - Nodes
 - Read Groups Infor
 - Write Groups Infor
 - Threads
 - Sorted Items
 - Discarded Items
 - Historian [SERVER]
 - Historian.Table.Table1
 - Historian.Table.Table2
 - ScriptTaskServer [127.0.0.1]
 - Tasks
 - Expressions
 - ScriptTaskClient [127.0.0.1]
 - Expressions
 - Tasks

On the right, a table displays the following data:

Item	Value
Executing Thread Amount	0
Configured Thread Amount	1
Pending Event Amount	0
Configured Channel Amount	1
Created Channel Amount	1
Started Channel Amount	1
Read Group Amount	1
Write Group Amount	0
Added Items Amount	1
Discarded Items Amount	0
Success Communication Amount	0
Failed Communication Amount	331
Last Error Code	-103
Last Error TimeStamp	23/10/2017 16:36:07...
Average Cycle Time	506
Average Execution Time	16
Last Execution Time	16
Node Redundancy	True
TimeBetweenHandlerExecute	00:00:00.0120007
TimeHandlerExecution	00:00:00

At the bottom of the window, there are three buttons: "Settings...", "Pause", and "Close".

Figura 5-32. Informações do Módulo

Objetos em Tempo de Execução



Esta funcionalidade permite visualizar todos os elementos abertos, incluindo Telas, Scripts e Relatórios. Os objetos do *Runtime* estão em grupos que contém informações específicas de suas funcionalidades, sendo chamados de *Namespace*. Segue a lista dos *Namespaces* disponíveis.

- Namespace Tag
- Namespace Security
- Namespace Alarm
- Namespace Device
- Namespace Dataset
- Namespace Script
- Namespace Display
- Namespace Report
- Namespace Info
- Namespace Server
- Namespace Client

Durante a configuração do projeto, digite diretamente no campo. O *Intellisense* guiará o usuário em relação aos *Namespaces* válidos para aquele campo assim como os membros disponíveis. No editor de código (*Interfaces*> *ScriptCodeEditor* e *DrawCodeBehind*) e na dinâmica *TextOutput* é necessário utilizar o prefixo "@" para evitar conflito com os *Namespaces.NET*. Exemplo: Nos diálogos e grades como da figura a seguir use:

```
Tag.Analog1  
Alarm.Group.Warning.TotalActive  
Device.Node.Node1.Status
```

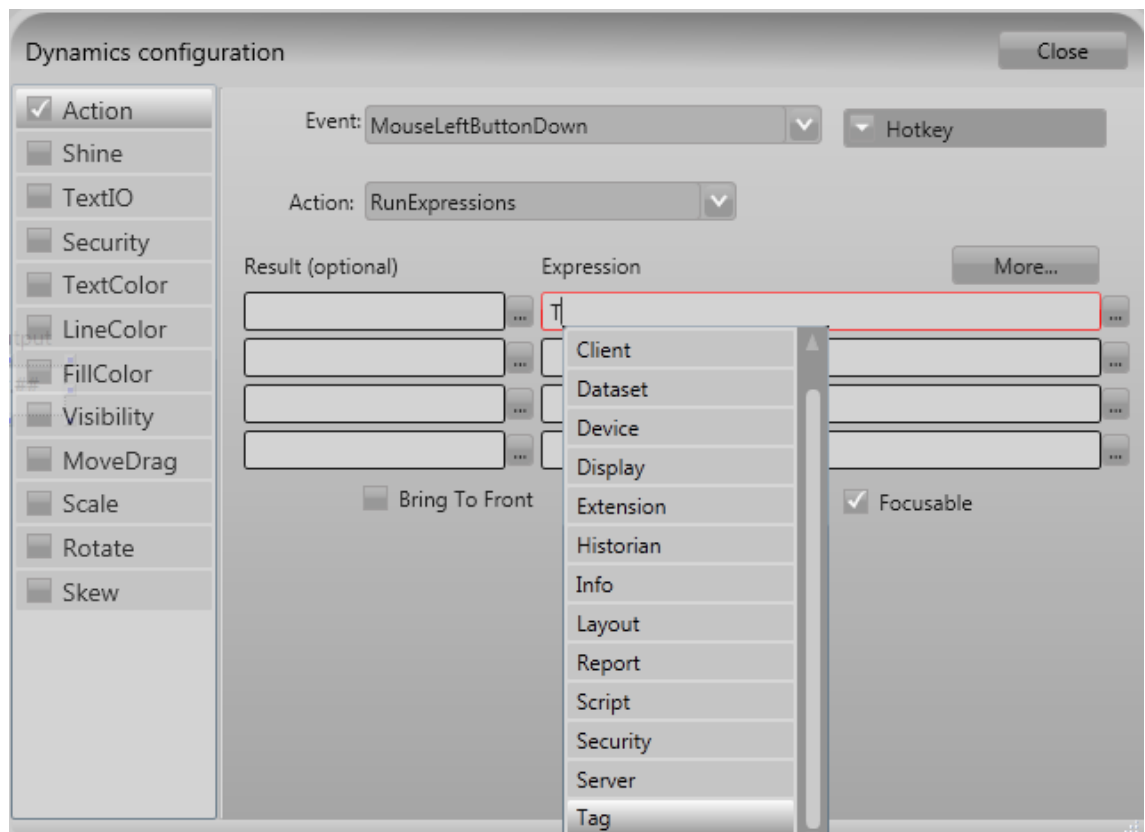


Figura 5-33. Namespaces em Diálogos

No código script da figura a seguir use:

```
@Tag.Analog1
@Alarm.Group.Warning.TotalActive
@Device.Node.Node1.Status
```

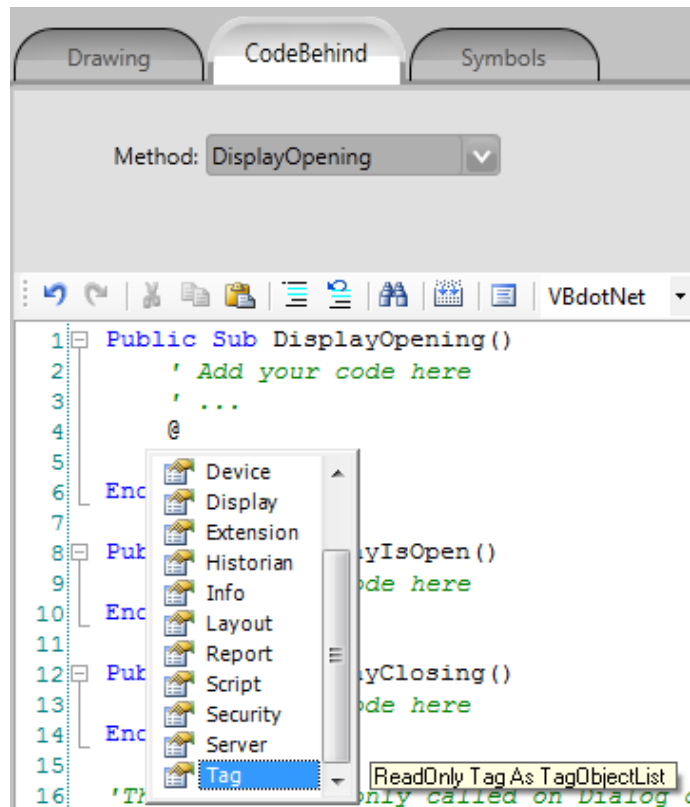


Figura 5-34. Namespaces em Scripts

Alguns campos nas grades e diálogos são permitidos somente para um tipo de objeto (um *namespace*), por exemplo, *Tag* ou *Display*. Quando o usuário digita um *Namespace* neste campo o *Intellisense* o levará até os objetos disponíveis.

Namespace Tag

Todas as variáveis de tempo real do projeto, ou *Tags de Projeto*, como são normalmente chamadas nos contextos de automação de processos, estão disponíveis neste *Namespace*. As Tags são criadas na interface *EditTagsObjects*. Use a sintaxe *Tag.Namespace* para referir-se à Tag criada e *Tag.namespace.Min* para acessar a propriedade *Mínimo*. Para cada Tag é definido um tipo para o seu valor.

A sintaxe *Tag.<TagName>* está disponível para todos os tipos de Tag. Para saber o tipo de uma Tag em particular é possível acessar propriedades adicionais e métodos usando a sintaxe: *Tag.<TagName>.<PropertyName>*. O elemento *TagObj* mostra propriedades e métodos disponíveis para todos os tipos de Tag. Por outro lado, *Analog* mostra as propriedades comuns a todas Tags Analógicas (refere-se ao tipo de Tag para propriedades específicas). São permitidos os tipos criados pelo usuário em combinação com os tipos incorporados.

ClassTagObj

Classes de base para todos os objetos Tag.

ForceValue: força a propriedade de valor do objeto ao valor dado como parâmetro. Parâmetros: valor do objeto. Exemplo:

```
@Tag.AnalogInt1.ForceValue(10)      (VB)
@Tag.AnalogInt1.ForceValue(10);    (C#)
```

Tostring: retorna uma string que representa o objeto atual. Exemplo:

```
Dim s as string                      (VB)
@Tag.AnalogInt1 = 33
```

```
s = @Tag.AnalogInt1.ToString();
string s;                                (C#)
@Tag.AnalogInt1 = 33;
s = @Tag.AnalogInt1.ToString();
```

Neste exemplo, s será avaliado como "33".

AlarmDisable: fornece ou define o estado de alarme desabilitado da Tag. Se 0 (zero), a propriedade *AlarmDisable* não fica ativa. No caso de valores diferentes de 0 (zero), a propriedade *AlarmDisable* fica ativa. Quando a propriedade *AlarmDisable* não está ativa, todos os alarmes associados à Tag atual não são tratados. Exemplo:

```
@Tag.AnalogInt1.AlarmDisable = 1          (VB)
@Tag.AnalogInt1.AlarmDisable = 1;        (C#)
```

Neste exemplo, os alarmes associados ao objeto *AnalogInt1* não serão tratados.

AlarmState: fornece ou define o estado de alarme da Tag. Se 0 (zero): fora de estado de alarme. No caso de valores diferentes de 0 (zero): em estado de alarme. Exemplo:

```
Dim alarmState as Integer                 (VB)
alarmState = @Tag.AnalogInt1.AlarmState
int alarmState;                           (C#)
alarmState = @Tag.AnalogInt1.AlarmState;
```

GetName: fornece o nome da Tag. Exemplo:

```
@Tag.ReferenceTagName.Link = @Tag.TagName.GetName() (VB)
@Tag.ReferenceTagName.Link = @Tag.TagName.GetName(); (C#)
```

Changed: fornece ou define o estado alterado do valor da Tag. True: valor alterado. False: valor inalterado.

Domain: fornece ou define a propriedade de Domínio da Tag. Se 0 (zero): servidor. Se 1: cliente.

Units: fornece ou define as unidades de engenharia usadas para quantificar a Tag.

Format: fornece ou define o formato do valor da Tag para fins de exibição.

Historian: fornece o objeto *HistoryItem* onde a Tag atual está configurada como *TagName*. Se Tags duplicados forem permitidos nas tabelas de Historiadores, então o Historiador apontará para o último objeto *HistoryItem* onde a Tag atual está configurada como *TagName*. Exemplo:

```
@Tag.AnalogInt1.Historian.Deadband = 3      (VB)
@Tag.AnalogInt1.Historian.Deadband = 3;    (C#)
```

Locked: fornece ou define o estado bloqueado da Tag. Quando uma Tag está bloqueada, o valor usado para o processamento origina-se da propriedade *LockValue* e não da propriedade de valor. Se 0 (zero): desbloqueado. No caso de valores diferentes de 0 (zero): bloqueado. Exemplo:

```
@Tag.AnalogDouble1.Locked = 1              (VB)
@Tag.AnalogDouble1.Locked = 1;            (C#)
```

Quality: fornece ou define o estado de qualidade da Tag. Valores permitidos:

Valor	Qualidade
0	Ruim
192	Bom

Tabela 5-1. Valores Padrões

Quando a qualidade é definida via OPC, ela está sujeita à especificação do servidor conectado. As qualidades padrões do protocolo OPC estão indicadas na tabela a seguir.

Valor	Qualidade	
0	Qualidade Ruim	Não Especificada
1	Valor de Limite de Campo	Baixo
2		Alto
3		Constante
4	Qualidade Ruim	Erro de Configuração

8		Não Conectado
12		Falha no Dispositivo
16		Falha no Sensor
20		Último Valor Conhecido
24		Falha de Comunicação
28		Fora de Serviço
64	Qualidade Incerta	Não Especificado
68		Último Valor Utilizável
80		Sensor sem Precisão
84		Unidades Excedidas
88		Subnormal
192	Qualidade Boa	Bom
216		Sobreposição Local
255	Sem Qualidade	Sem Valor

Tabela 5-2. Valores de Qualidade OPC

Exemplo:

```
@Tag.AnalogInt1.Quality = 192      (VB)
@Tag.AnalogInt1.Quality = 192;    (C#)
```

Retentive: fornece ou define a propriedade Retentiva da Tag, a qual especifica se a propriedade de Valor da Tag deve ser salva quando a aplicação for encerrada. O valor salvo é então usado como o valor de inicialização na execução da próxima aplicação. Se 0 (zero): não retentivo. No caso de valores diferentes de 0 (zero): retentivo. Exemplo:

```
@Tag.AnalogInt1.Retentive = 1      (VB)
@Tag.AnalogInt1.Retentive = 1;    (C#)
```

TimeStamp: fornece ou define a propriedade *TimeStamp* da Tag. Exemplo:

```
Dim dt As DateTimeOffset           (VB)
dt = @Tag.AnalogInt1.Timestamp
DateTimeOffset dt;                (C#)
dt = @Tag.AnalogInt1.Timestamp;
```

ValueType: fornece a propriedade *ValueType* da Tag. Exemplo:

```
Dim doubleType As Integer         (VB)
doubleType = @Tag.AnalogInt1.ValueType
int doubleType;                   (C#)
doubleType = @Tag.AnalogInt1.ValueType;
```

Visibility: fornece ou define o estado de visibilidade da Tag. Se 0 (zero): privado, 1: protegido e 2: público. Exemplo:

```
Dim visibilityState as Integer    (VB)
visibilityState = @Tag.AnalogInt1.Visibility
int visibilityState;              (C#)
visibilityState = @Tag.AnalogInt1.Visibility;
```

Class Digital

Propriedades do *Runtime* para Classe Digital. Valores possíveis: 0 = false e 1 = true. Equivalente no script:

- C#: int32
- VB: int
- .NET: int

Toggle: alterna a propriedade de valor da Tag entre 0 (zero) e 1. Se o valor atual é 0 (zero), então o novo valor é 1. Se o valor atual é 1, então o novo valor é 0 (zero). Exemplo:

```
Dim newValue as Byte              (VB)
```

```
newValue = @Tag.Digital1.ToggleValue()  
byte newValue; (C#)  
newValue = @Tag.Digital1.ToggleValue();
```

LockValue: fornece ou define a propriedade *LockedValue* da Tag. Quando uma Tag está bloqueada, o valor usado para o processamento origina-se da propriedade LockValue e não da propriedade Valor. Exemplo:

```
@Tag.Digital1.LockValue = 1 (VB)  
@Tag.Digital1.LockValue = 1; (C#)
```

State: fornece a propriedade de Estado da Tag digital. Valor = 0 corresponde ao estado = False (VB) ou estado = false (C#). Valor = 1 corresponde ao estado = True (VB) ou estado= true (C#).

Value: fornece ou define a propriedade de Valor da Tag digital. Valores válidos: 0 (zero) ou 1. Exemplo:

```
@Tag.Digital1.Value = 1 (VB)  
@Tag.Digital1.Value = 1; (C#)
```


Class Analog

Propriedades do *Runtime* para a classe analógica.

Membro	Descrição
Bit0	Fornece ou define o bit 0 do valor da Tag.
Bit1	Fornece ou define o bit 1 do valor da Tag.
Bit2	Fornece ou define o bit 2 do valor da Tag.
Bit3	Fornece ou define o bit 3 do valor da Tag.
Bit4	Fornece ou define o bit 4 do valor da Tag.
Bit5	Fornece ou define o bit 5 do valor da Tag.
Bit6	Fornece ou define o bit 6 do valor da Tag.
Bit7	Fornece ou define o bit 7 do valor da Tag.
Bit8	Fornece ou define o bit 8 do valor da Tag.
Bit9	Fornece ou define o bit 9 do valor da Tag.
Bit10	Fornece ou define o bit 10 do valor da Tag.
Bit11	Fornece ou define o bit 11 do valor da Tag.
Bit12	Fornece ou define o bit 12 do valor da Tag.
Bit13	Fornece ou define o bit 13 do valor da Tag.
Bit14	Fornece ou define o bit 14 do valor da Tag.
Bit15	Fornece ou define o bit 15 do valor da Tag.
Bit16	Fornece ou define o bit 16 do valor da Tag.
Bit17	Fornece ou define o bit 17 do valor da Tag.
Bit18	Fornece ou define o bit 18 do valor da Tag.
Bit19	Fornece ou define o bit 19 do valor da Tag.
Bit20	Fornece ou define o bit 20 do valor da Tag.
Bit21	Fornece ou define o bit 21 do valor da Tag.
Bit22	Fornece ou define o bit 22 do valor da Tag.
Bit23	Fornece ou define o bit 23 do valor da Tag.
Bit24	Fornece ou define o bit 24 do valor da Tag.
Bit25	Fornece ou define o bit 25 do valor da Tag.
Bit26	Fornece ou define o bit 26 do valor da Tag.
Bit27	Fornece ou define o bit 27 do valor da Tag.
Bit28	Fornece ou define o bit 28 do valor da Tag.
Bit29	Fornece ou define o bit 29 do valor da Tag.
Bit30	Fornece ou define o bit 30 do valor da Tag.
Bit31	Fornece ou define o bit 31 do valor da Tag.
Hi	Fornece ou define o limite high da Tag.
HiHi	Fornece ou define o limite high-high da Tag.
Lo	Fornece ou define o limite low da Tag.
LoLo	Fornece ou define o limite low-low da Tag.

Tabela 5-3. Propriedades do *Runtime* para Analog Class

Class Analog<T>

Propriedades de *Runtime* para Classe *Analog <T>*.

Deadband: fornece ou define a banda morta da Tag analógica. Exemplo:

```
@Tag.AnalogDouble1.Deadband = 5           (VB)
@Tag.AnalogDouble1.Deadband = 5;         (C#)
```

LockValue: fornece ou define o valor de bloqueio da Tag analógico. Exemplo:

```
@Tag.AnalogDouble1.LockValue = 50        (VB)
@Tag.AnalogDouble1.LockValue = 50;       (C#)
```

Analog.Min e *Analog.Max*: fornece ou define o valor mínimo ou máximo da Tag analógica.

Exemplo:

```
@Tag.AnalogDouble1.Min = 100          (VB)
@Tag.AnalogDouble1.Min = 100;        (C#)
@Tag.AnalogDouble1.Max = 100          (VB)
@Tag.AnalogDouble1.Max = 100;        (C#)
```

StartValue: fornece ou define o valor inicial da Tag analógica. Exemplo:

```
@Tag.AnalogDouble1.StartValue = 50    (VB)
@Tag.AnalogDouble1.StartValue = 50;    (C#)
```

State: fornece ou define o estado da Tag analógica. Se o valor é igual a 0 (zero), o estado é FALSE. Se o valor é diferente de 0 (zero), o estado é TRUE. Exemplo:

```
Dim state as Boolean                    (VB)
state = @Tag.AnalogInt1.State
bool state;                             (C#)
state = @Tag.AnalogInt1.State;
@Tag.AnalogInt1.Value = 55              (VB) ou
@Tag.AnalogInt1 = 55
@Tag.AnalogInt1.Value = 55;             (C#) ou
@Tag.AnalogInt1 = 55;
```

Class AnalogInt

Propriedades do *Runtime* para a classe AnalogInt.

Class AnalogDecimal

Propriedades do *Runtime* para a classe Decimal Analógica.

Class AnalogDoble

Propriedades do *Runtime* para a classe AnalogDouble.

Class Text

Propriedades do *Runtime* para classe texto.

LockValue: fornece ou define o valor de bloqueio da Tag de texto. Exemplo:

```
@Tag.Text1.LockValue = "Welcome"      (VB)
@Tag.Text1.LockValue = "Welcome";      (C#)
```

Value: fornece ou define o valor da Tag de texto. Exemplo:

```
@Tag.Text1.Value = "My text"           (VB) ou
@Tag.Text1 = "My text"
@Tag.Text1.Value = "My text";           (C#) ou
@Tag.Text1 = "My text";
```

Class TDateTime

Propriedades do *Runtime* para Classe TDateTime.

LockValue: fornece ou define o valor de bloqueio da Tag de TDateTime. Exemplo:

```
@Tag.Text1.LockValue = "Welcome"      (VB)
@Tag.Text1.LockValue = "Welcome";      (C#)
```

Value: fornece ou define o valor da Tag de TDateTime. Exemplo:

```
@Tag.Text1.Value = "My text"           (VB) ou
@Tag.Text1 = "My text"
```

```
@Tag.Text1.Value = "My text";           (C#) ou
@Tag.Text1 = "My text";
```

Class Counter

Define propriedades do *Runtime* para Classe *Counter*.

Event: fornece o evento da Tag *Counter*. Valores possíveis: "Change", "ChangeUp" e "ChangeDown". Exemplo:

```
Dim counter1Event as string             (VB)
counter1Event = @Tag.Counter1.Event
string counter1Event;                  (C#)
counter1Event = @Tag.Counter1.Event;
```

Model: fornece o modelo da Tag *Counter*. Valores possíveis: "Up" e "Down". Exemplo:

```
Dim counter1Model as string            (VB)
counter1Model = @Tag.Counter1.Model
string counter1Model;                 (C#)
counter1Model = @Tag.Counter1.Model;
```

Trigger: fornece ou define o disparo da Tag *Counter*. Exemplo:

```
@Tag.Counter1.Trigger = "Tag.Digital1" (VB)
@Tag.Counter1.Trigger = "Tag.Digital1"; (C#)
```

Class Timer

Propriedades do *Runtime* para Classe *Timer*.

Interval: fornece ou define o intervalo da Tag *Timer*. Trata-se de uma string que representa o intervalo de tempo exibido no formato "hh:mm:ss.mmm". Exemplo:

```
@Tag.Timer1.Interval = "0:0:10"       (VB)
@Tag.Timer1.Interval = "0:0:10";      (C#)
```

Model: fornece o modelo da Tag *Timer*. Valores possíveis: "SquareWave", "Pulse" e "Comparer". Exemplo:

```
Dim timerModel as string               (VB)
timerModel = @Tag.Timer1.Model
string timerModel;                    (C#)
timerModel = @Tag.Timer1.Model;
```

Class Reference

Propriedades de *Runtime* para a Classe *Reference*.

Link: fornece ou define o link da Tag *Reference*. Exemplo:

```
@Tag.Reference1.Link = @Tag.TagName.GetName() (VB)
@Tag.Reference1.Link = @Tag.TagName.GetName(); (C#)
```

Class TDataTable

Propriedades do *Runtime* para Classe *TDataTable*.

Initialize: define uma nova referência ao objeto Tabela. Este método é usado somente internamente.

Table: fornece uma cópia do objeto *DataTable*. Exemplo:

```
Dim dt as New TDataTable(parent, id)     (VB)
Dim table As DataTable
table = dt.Table
TDataTable dt = new TDataTable(parent, id); (C#)
DataTable table;
```

```
table = dt.Table;
```

OverwriteOnUpdate: fornece ou define o operando *OverwriteOnUpdate*. Esta propriedade é usada somente internamente.

Update: atualiza o objeto Tabela. Este método é usado somente internamente. Parâmetros: DataTable, table.

Classe UserType

Propriedades do *Runtime* para Classe UserType.



Pesquisa na Documentação: outros objetos em tempo de execução

Consulte, na documentação do produto, as classes associadas aos *Namespaces* restantes (já analisamos com detalhes as classes pertencentes às Tags na seção anterior) e identifique seus parâmetros analisando exemplos de parametrização. A tabela a seguir pode servir como referência para a pesquisa.

Fonte: Manual de Utilização Série BluePlant – MU224000!

Namespace	Classe	Parâmetros e Exemplos
Security	<i>ModuleSecurity</i>	...
	<i>SecurityPermission</i>	
	<i>DateCreated</i>	
	<i>SecurityUser</i>	
Alarm	<i>AlarmGroup</i>	
	<i>AlarmItem</i>	
	<i>ModuleAlarm</i>	
Device	<i>DeviceAccessType</i>	
	<i>DeviceChannel</i>	
	<i>DeviceNode</i>	
	<i>ModuleDevice</i>	
Dataset	<i>DatasetDB</i>	
	<i>DatasetFile</i>	
	<i>DatasetQuery</i>	
	<i>DatasetTable</i>	
	<i>ModuleDataset</i>	
Script	<i>ModuleScript</i>	
	<i>ScriptTask</i>	
Display	<i>Dispay</i>	
	<i>Layout</i>	
Report	<i>ModuleReport</i>	
	<i>ReportItem</i>	
Info	<i>ModuleInfo</i>	
	<i>InfoProjectVersion</i>	
	<i>InfoProjectSettings</i>	
	<i>InfoModuleList</i>	
	<i>InfoLicense</i>	
	<i>InfoModule</i>	
Server	<i>ServerStation</i>	
Client	<i>ClientStation</i>	

Recursos Avançados



Esta seção contém informação adicional sobre os aplicativos do BluePlant incluindo linha de comando, execução do BluePlant como um Serviço do Windows, clientes remotos e instalação do Web Server no IIS.

Linhas de Comando

As informações sobre as ferramentas BluePlant e executáveis que podem ser chamados usando linhas de comando e parâmetros específicos estão descritos a seguir. Ao executar os comandos sem os parâmetros opcionais serão assumidos os valores padrões.

TStartup

Inicializa o projeto BluePlant, os módulos que irão iniciar são os configurados no projeto, na aba *Run*, opção *Executar*. Deve ser executado o arquivo TStartup.exe da pasta de instalação do BluePlant. Para mais informações a respeito da configuração de projetos redundantes, consultar a NAP157 - Redundância de Servidores SCADA com BluePlant.

Parâmetros:

/project: caminho do projeto (com extensão) e nome entre aspas duplas

/username: (opcional) , nome do usuário que será usado para inicializar o servidor. Caso não seja especificado será usado o usuário hóspede

/redundancy: (opcional) indica que a redundância do servidor está sendo usada (requer ip1 e ip2)

/ip1: (sem redundância, opcional) endereço IP do Servidor Primário

/port1: (opcional) porta TCP do Servidor Primário

/ip2: (sem redundância, opcional) endereço IP do Servidor Secundário

/port2: (opcional) porta TCP do Servidor Secundário

Exemplos:

1. "C:\Program Files\Altus\BluePlant\bp-9.1\TStartup.exe" /project:"C:\BluePlant Projects\Project1.tproj"
2. "C:\Program Files\Altus\BluePlant\bp-9.1\TStartup.exe" /project:"C:\BluePlantProjects\Project1.tproj" /username:Administrator /port1:3101
3. "C:\Program Files\Altus\BluePlant\bp-9.1\TStartup.exe" /project:"C:\BluePlant Projects\Project1.tproj" /username:Administrator /redundancy /ip1:192.168.1.1 /port1:3101 /ip2:192.168.1.2 /port2:3101

TVisualizer

Inicializa o Módulo BluePlant Visualizer, ou seja, inicia somente as telas configuradas no BluePlant. Deve ser executado o arquivo TVisualizer.exe da pasta de instalação do BluePlant. Este comando só funcionará se houver um projeto em execução no local indicado no parâmetro ip1.

Parâmetros:

/username: (opcional) nome do usuário que será usado para inicializar o servidor. Caso não seja especificado será usado o usuário hóspede.

/redundancy: (opcional) indica que a redundância do servidor está sendo usada (requer ip1 e ip2)

/ip1: (sem redundância, opcional) endereço IP do Servidor Primário, valor padrão é: localhost, ou 127.0.0.1

/port1: (opcional) porta TCP do Servidor Primário, valor padrão é: 3101

/ip2: (opcional) endereço IP do Servidor Secundário

/port2: (opcional) porta TCP do Servidor Secundário

Exemplos:

1. " C:\Program Files\Altus\BluePlant\bp-9.1\TVisualizer.exe"
2. " C:\Program Files\Altus\BluePlant\bp-9.1\TVisualizer.exe" /username:Administrator /ip1:192.168.1.1 /port1:3101
3. " C:\Program Files\Altus\BluePlant\bp-9.1\TVisualizer.exe" /username:Administrator /redundancy /ip1:192.168.1.1 /port1:3101 /ip2:192.168.1.2 /port2:3101

TraceWindow

Inicializa a ferramenta BluePlant TraceWindow. Deve ser executado o arquivo TraceWindow.exe da pasta de instalação do BluePlant. Este comando só funcionará se houver um projeto em execução no local indicado no parâmetro ip1.

Parâmetros:

/username: (opcional) nome do usuário que será usado para inicializar o TraceWindow. Caso não seja especificado será usado o usuário hóspede.

/redundancy: (opcional) indica que a redundância do servidor está sendo usada (requer ip1 e ip2)

/ip1: (sem redundância, opcional) endereço do Servidor Primário, valor padrão é: localhost, ou 127.0.0.1

/port1: (opcional) porta TCP do Servidor Primário, valor padrão é: 3101

/ip2: (opcional) endereço IP do Servidor Secundário

/port2: (opcional) porta TCP do Servidor Secundário

/connectiontimeout: (opcional) tempo em segundos para perda de comunicação

/AutoRunDiagnostics: (opcional) tempo em minutos para salvar um log do projeto para depuração da aplicação pelo suporte da Altus. Recomendável utilizar valor "30". Antes de iniciar o projeto limpe todos os arquivos que estão na pasta C:\ProgramData\BluePlant. Após ocorrer um problema, compacte a pasta e envie para o suporte da Altus descrevendo o problema ocorrido.

Exemplos:

1. " C:\Program Files\Altus\BluePlant\bp-9.1\TraceWindow.exe"
2. " C:\Program Files\Altus\BluePlant\bp-9.1\TraceWindow.exe" /username:Administrator /ip1:192.168.1.1 /port1:3101
3. " C:\Program Files\Altus\BluePlant\bp-9.1\TraceWindow.exe" /username:Administrator /redundancy /ip1:192.168.1.1 /port1:3101 /ip2:192.168.1.2 /port2:3101

PropertyWatch

Inicializa a ferramenta de diagnóstico PropertyWatch. Deve ser executado o arquivo PropertyWatch.exe da pasta de instalação do BluePlant. Este comando só funcionará se houver um projeto em execução no local indicado no parâmetro ip1.

Parâmetros:

/username: (opcional) nome do usuário que será usado para inicializar o PropertyWatch. Caso não seja especificado será usado o usuário hóspede.

/redundancy: (opcional) indica que a redundância do servidor está sendo usada (requer ip1 e ip2)

/ip1: (sem redundância, opcional) endereço do Servidor Primário, valor padrão é: localhost, ou 127.0.0.1

/port1: (opcional) porta TCP do Servidor Primário, valor padrão é: 3101

/ip2: (opcional) IP endereço IP do Servidor Secundário

/port2: (opcional) porta TCP do Servidor Secundário

Exemplos:

1. " C:\Program Files\Altus\BluePlant\bp-9.1\PropertyWatch.exe"
2. " C:\Program Files\Altus\BluePlant\bp-9.1\PropertyWatch.exe" /username:Administrator /ip1:192.168.1.1 /port1:3101
3. " C:\Program Files\Altus\BluePlant\bp-9.1\PropertyWatch.exe" /username:Administrator /redundancy /ip1:192.168.1.1 /port1:3101 /ip2:192.168.1.2 /port2:3101

ModuleInformation

Inicializa a ferramenta de diagnóstico Módulo Informação. Deve ser executado o arquivo ModuleInformation.exe da pasta de instalação do BluePlant. Este comando só funcionará se houver um projeto em execução no local indicado no parâmetro ip1.

Parâmetros:

/username: (opcional) nome do usuário que será usado para inicializar o Módulo Informação. Caso não seja especificado será usado o usuário hóspede.

/redundancy: (opcional) indica que a redundância do servidor está sendo usada (requer ip1 e ip2)

/ip1: (sem redundância, opcional) endereço do Servidor Primário, valor padrão é: localhost, ou 127.0.0.1

/port1: (opcional) porta TCP do Servidor Primário, valor padrão é: 3101

/ip2: (opcional) IP endereço do Servidor Secundário

/port2: (opcional) porta TCP do Servidor Secundário

Exemplos:

1. " C:\Program Files\Altus\BluePlant\bp-9.1\ModuleInformation.exe"
2. " C:\Program Files\Altus\BluePlant\bp-9.1\ModuleInformation.exe" /username:Administrator /ip1:192.168.1.1 /port1:3101
3. " C:\Program Files\Altus\BluePlant\bp-9.1\ModuleInformation.exe" /username:Administrator /redundancy /ip1:192.168.1.1 /port1:3101 /ip2:192.168.1.2 /port2:3101

DisableTaskSwitchProtection

Executando o arquivo *DisableTaskSwitchProtection.bat* é instalado um driver que bloqueia o uso das teclas CTRL+ALT+DEL enquanto o *TVisualizer* está em execução. Pode ser necessário reiniciar o computador para isso. Esta ação deve ser executada no computador do cliente somente uma vez.

EnableTaskSwitchProtection

Executando o arquivo *EnableTaskSwitchProtection.bat* é desinstalado o driver que bloqueia o uso das teclas CTRL+ALT+DEL enquanto o *TVisualizer* está em execução. Pode ser necessário reiniciar o computador para isso. Esta ação deve ser executada no computador do cliente somente uma vez.

RegServer

Registra o servidor BluePlant OPC.

UnRegServer

Cancela o registro do servidor BluePlant OPC.

Executando o BluePlant como um Serviço do Windows

Este procedimento descreve como instalar aplicações BluePlant do servidor *Runtime* para execução como um serviço do Windows.

Por favor, observe que, para permitir a Engenharia Distribuída e para servir páginas para clientes Web, o usuário também precisa ter um servidor Web habilitado. Consulte o tópico Instalando o Web Server no IIS para informações sobre esse procedimento. A fim de executar o aplicativo como um serviço do Windows digite: <.NET Framework Install Path>\installutil <Install Path>\<BluePlant Version>\TStartupAsService.exe.

No prompt do DOS (deve-se Executar como Administrador), execute o seguinte comando:

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>installutil "C:\Program Files\Altus\bp-2012.1\TStartupAsService.exe"
```

Em seguida, o usuário precisa incluir a linha de comando para o serviço, que é a mesma do programa *tStartup.exe*: /project:<projectNameAndPath>.

Não está disponível ainda um utilitário para configurar o registro. O usuário precisa fazer isso manualmente. Deve-se definir no Registro do Windows e configurar os parâmetros. Abra o Editor de Registro (regedit) e vá para a seguinte chave:

```
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TStartup\ImagePath"
```

```
"C:\Program Files\Altus\bp-2012.1\TStartupAsService.exe" "/project:C:\BluePlant Projects\Project1.tproj"
```

Note que nos exemplos acima, deve-se alterar o caminho de instalação do aplicativo para a instalação no seu computador.

Finalmente, nas janelas *Services* (Ferramentas Administrativas), deve-se configurar o *TStartup Service*.

Pode-se configurá-lo como *Automatic*, de forma que o projeto selecionado abra quando o computador é inicializado.

TWebServer

O *TWebServer* é um *WebServer* embutido na instalação do BluePlant. Ele pode ser utilizado como solução em sistemas que não possuam um *WebServer* tal como o IIS. É recomendado que o *WebServer* utilizado seja o IIS. Caso o IIS não seja utilizado é importante se tomar cuidado com o uso do BlueWave e do BluePlant num mesmo servidor ou computador. Não é possível utilizar os dois sistemas ao mesmo tempo. Caso se deseje utilizar o BluePlant em um sistema

em que já exista uma instalação do BlueWave é necessário ter alguns cuidados antes da utilização de cada um. Existe uma instalação do TWebServer juntamente com o BlueWave e outra com o BluePlant. Para rodar o BluePlant corretamente deve ser parado o *TWebServer* do BlueWave e inicializado de forma manual o *TWebServer* do BluePlant.

Clientes Remotos

O único pré-requisito para executar o aplicativo no cliente remoto é a instalação do .Net Framework. O aplicativo não precisa ser instalado nos computadores clientes. Uma vez que o servidor BluePlant tenha o WebServer (IIS ou TWebServer) em execução, pode-se digitar o seguinte endereço do navegador Internet Explorer:

<http://<server ip or name>/<BluePlant Version>/tvisualizerremote.application> para o cliente Rich e <http://<server ip or name>/<BluePlant Version>/tvisualizerweb.xbap> para o cliente Web.

Dependendo da versão do Internet Explorer a seguinte configuração deve ser feita: desativar o modo protegido na Zona Internet ou Intranet conforme ilustrado na figura a seguir.

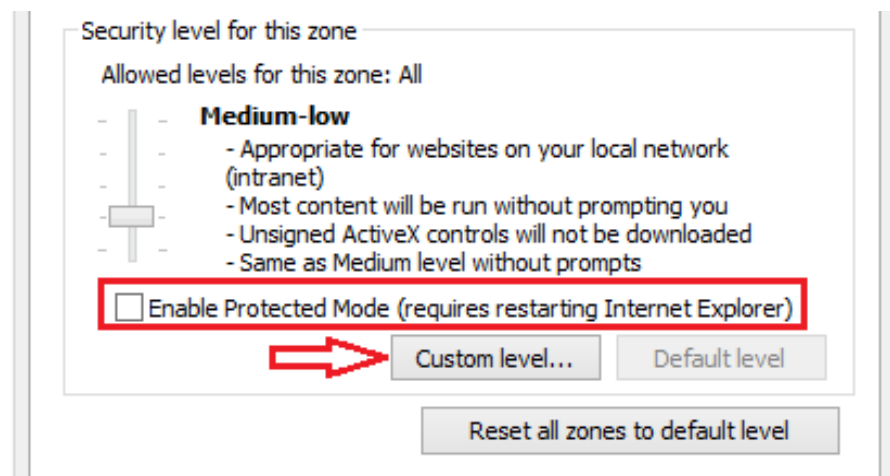


Figura 5-35. Desativar o Modo Protegido

Habilitar aplicações do navegador XAML conforme mostrado na figura a seguir.

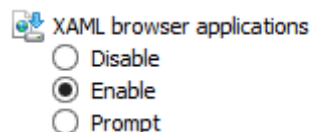


Figura 5-36. Habilitar Aplicações do Navegador XAML

Exemplos:

1. Cliente Remoto Rich:

<http://127.0.0.1/bp-2012.1/TVisualizerRemote.application>

2. Cliente Web:

<http://127.0.0.1/bp-2012.1/TVisualizerWeb.xbap>

Instalando o Web Server no IIS

Ao instalar o aplicativo, se o IIS for localizado, o usuário não será capaz de instalar o nosso servidor Web embutido (*TWebServer.exe*).

O programa irá rodar sem o servidor Web, acessando remotamente configurações do projeto e servindo páginas Web em *Runtime*. Todas as outras ferramentas de execução ou engenharia do BluePlant serão executadas sem a necessidade de instalar o TWebServer ou configurar o IIS.

Se o usuário desejar habilitar o acesso de engenharia remoto e os clientes Web usando o IIS, ele precisa instalar alguns serviços no IIS. Este manual explica como instalar serviços BluePlant no IIS. Esta descrição refere-se ao IIS 7.x, mas outras versões não apresentam muitas diferenças.

Outro cenário em que o IIS é necessário é quando o usuário quer executar o RUNTIME como um serviço do Windows, e ainda ser capaz de fornecer páginas da Web e engenharia remota.

Pode-se instalar três serviços dentro do IIS, o primeiro deles é o *TProjectServer* que permite acessar remotamente suas configurações de projeto. Outro serviço é o *TVisualizerWeb/TVisualizerRemote* que permite o acesso remoto a interfaces em *Runtime*. Finalmente, o *iDataPanel* é o serviço para fornecer dados para dispositivos iOS. Use as seguintes instruções passo-a-passo para configurar o IIS

Procedimento de Instalação

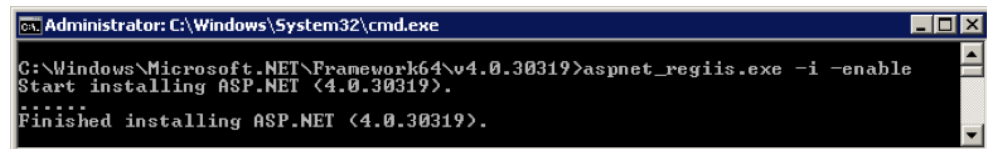
Este procedimento pode ser executado no Windows 7, x64, com o IIS 7.5. Com versões mais antigas do sistema operacional e do IIS, o procedimento pode ser ligeiramente diferente, como a definição de um *Virtual Directory* em vez de *Add Application*.

1. Verifique se o IIS está instalado e funcionando, por exemplo, tentando abrir páginas HTML.
2. Habilite o ASP.Net e manipuladores “.Svc” para o Serviço WCF.

Instalação no IIS 7.x

Se o IIS foi instalado após a instalação do .NET Framework é necessário executar o procedimento descrito em <http://msdn.microsoft.com/en-us/library/ms752252.aspx> que consiste em executar os seguintes programas usando o prompt de comando como administrador (conforme ilustrado nas figuras a seguir):

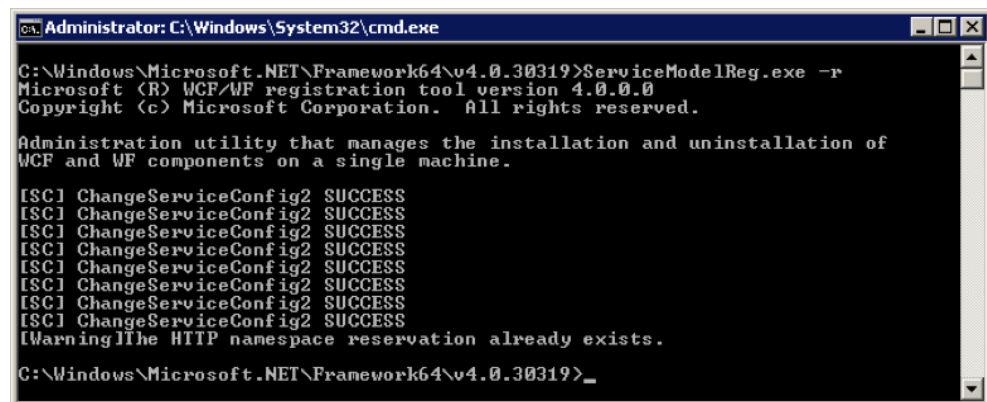
"aspnet_regiis -i -enable" (do diretório de instalação do .NET Framework*)



```
Administrator: C:\Windows\System32\cmd.exe
C:\Windows\Microsoft.NET\Framework64\v4.0.30319>aspnet_regiis.exe -i -enable
Start installing ASP.NET (4.0.30319).
*****
Finished installing ASP.NET (4.0.30319).
```

Figura 5-37. Procedimento de Instalação para o IIS 7.x

"ServiceModelReg.exe" -r (do diretório de instalação do .NET Framework*)



```
Administrator: C:\Windows\System32\cmd.exe
C:\Windows\Microsoft.NET\Framework64\v4.0.30319>ServiceModelReg.exe -r
Microsoft (R) WCF/WF registration tool version 4.0.0.0
Copyright (c) Microsoft Corporation. All rights reserved.

Administration utility that manages the installation and uninstallation of
WCF and WF components on a single machine.

[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[SC] ChangeServiceConfig2 SUCCESS
[Warning]The HTTP namespace reservation already exists.
C:\Windows\Microsoft.NET\Framework64\v4.0.30319>_
```

Figura 5-38. ServiceModelReg.exe

O diretório de instalação do .NET Framework é:
C:\Windows\Microsoft.NET\Framework\v4.0.30319,
ou
C:\Windows\Microsoft.NET\Framework64\v4.0.30319 (para Windows x64).

Instalação no IIS 8.0

No caso do IIS 8.0 o procedimento está descrito a seguir.

1. Vá para *Program and Features*, escolha *Turn Windows features on or off*.



Figura 5-39. Chaveamento de Funcionalidades do Windows

2. Habilite a opção *Internet Information Services > World Wide Web Services > Application Development Features > ASP.NET 4.5*.

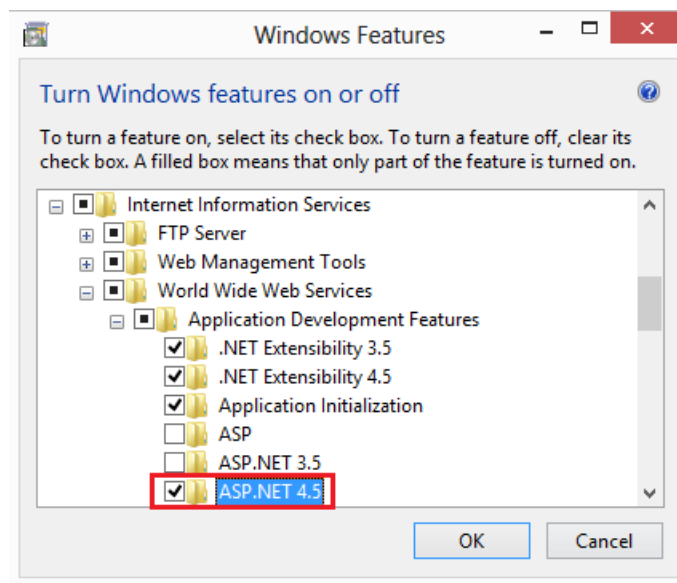


Figura 5-40. Funcionalidades do Windows

3. Habilite a opção *HTTP Activation* (no .Net Framework 4.5 Advanced Services).

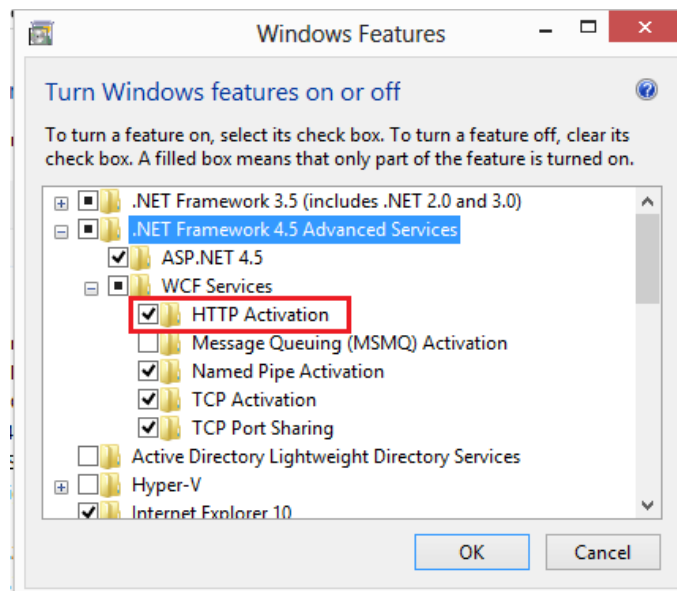


Figura 5-41. Ativação HTTP

- Verifique se a extensão ".svc" mapeada para a pasta está mapeada para *aspnet_isapi.dll* (no site *msdn.microsoft*, link anterior, é explicado como verificar isso para várias versões IIS).

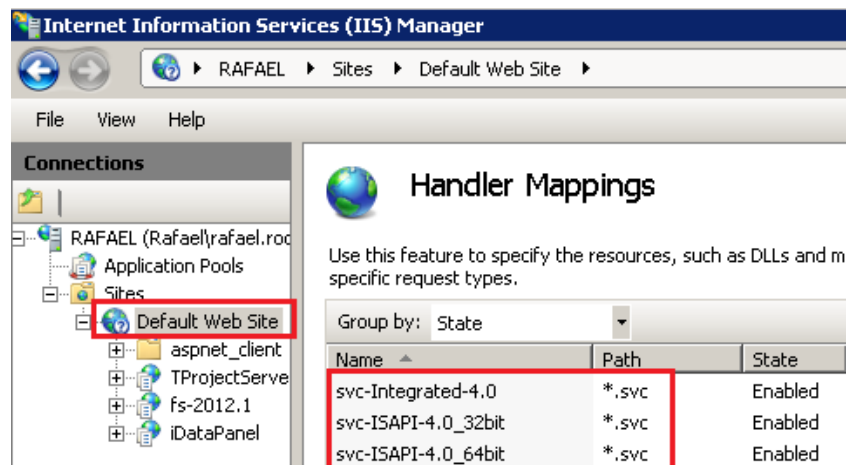


Figura 5-42. Mapeamento da Extensão

- Na pasta de instalação do BluePlant, execute o utilitário: *InstallTWebServer.exe /uninstall*, o qual removerá a instalação atual do *TWebServer* conforme ilustrado na figura a seguir.

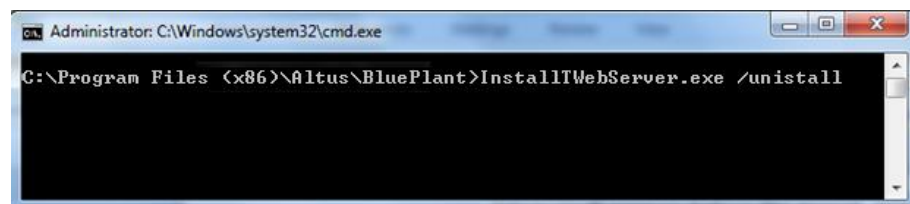


Figura 5-43. Utilitário InstallTWebServer

- Na sequência verifique se o *TWebServer* está definido para iniciar automaticamente. Abra o Editor do Registro do Windows (*regedit.exe*) e vá para a seguinte chave:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run.

7. Se for identificada uma referência a *TWebServer*, apague-a.
8. No IIS 7.x, abra *Sites/Default Web Site/Add Application* e configure a informação para os serviços conforme ilustrado na figura a seguir.

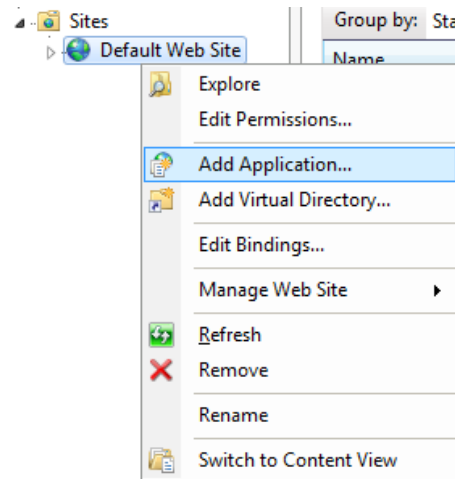


Figura 5-44. Informação para serviços

Os ajustes para o *TProjectServer* apresentadas na figura a seguir são:

- *Alias*: *TProjectServer*,
- *Physical Path* (1): verifique e corrija para a pasta de instalação BluePlant;
- *Application Pool*: configure para qualquer conjunto baseado no .NET 4.0.

Exemplo: ASP.NET v4.0.

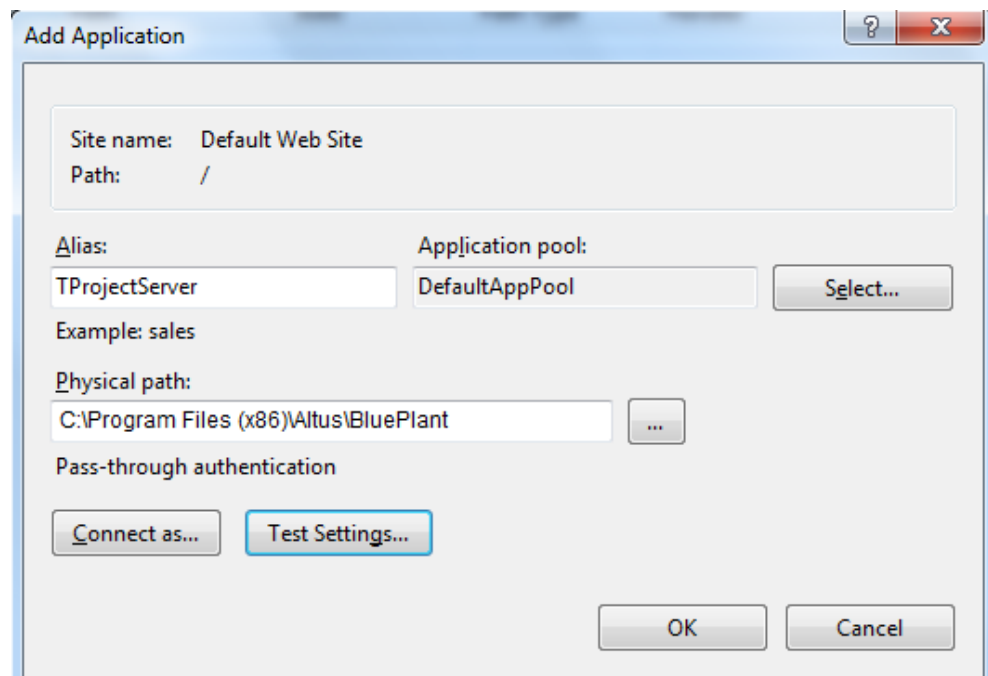


Figura 5-45. ProjectServer

O acesso como *Everyone* (pelo menos para leitura) deve ser configurado. No IE, use a URL <http://localhost/tprojectserver/service.svc> para verificar se o serviço foi corretamente instalado. Ela mostrará uma página com informações sobre o serviço. Os ajustes para o *TVisualizerWeb/TVisualizerRemote* são:

- *Alias*: versão do BluePlant, por exemplo: bp-9.1.6;
- *Physical Path*: pasta de instalação do BluePlant para a versão específica;
- *Application Pool*: configure para qualquer conjunto baseado no .NET 4.0.

Exemplo: ASP.NET v4.0

Security: A opção Security deve estar habilitada para *Everyone*, pelo menos para leitura. No IE, use a URL `http://localhost/bp-9.1.6/service.svc` para verificar se o serviço foi corretamente instalado. Ela deve mostrar uma página com informações sobre o serviço.

Os ajustes para o Serviço *iDataPanel* apresentados na figura a seguir são:

- *Alias*: *iDataPanel*;
- *Physical Path* (1): pasta de instalação do BluePlant para a versão específica;
- *Application Pool*: configure para qualquer conjunto baseado no .NET 4.0.

Exemplo: ASP.NET v4.0.

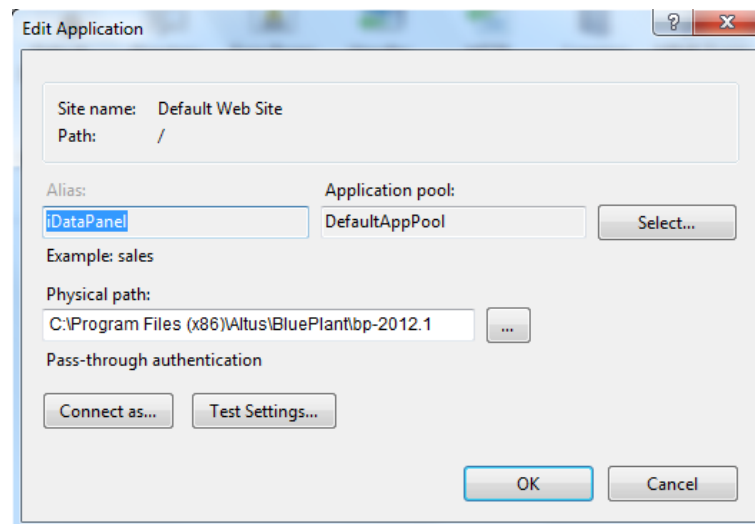


Figura 5-46. Serviço *iDataPanel*

Security: A opção Security deve estar habilitada para *Everyone*, pelo menos para leitura. No IE, use a URL `http://localhost/iDataPanel/iDataPanelService.svc` para verificar se o serviço foi corretamente instalado. Ela mostrará uma página com informações sobre o serviço.

Os ajustes para o *iDataPanelImages* (Diretório Virtual) ilustrados nas figuras a seguir são:

- *Alias*: *iDataPanelImages*;
- *Physical Path*: C:\BluePlant Projects\iDataPanelImages.

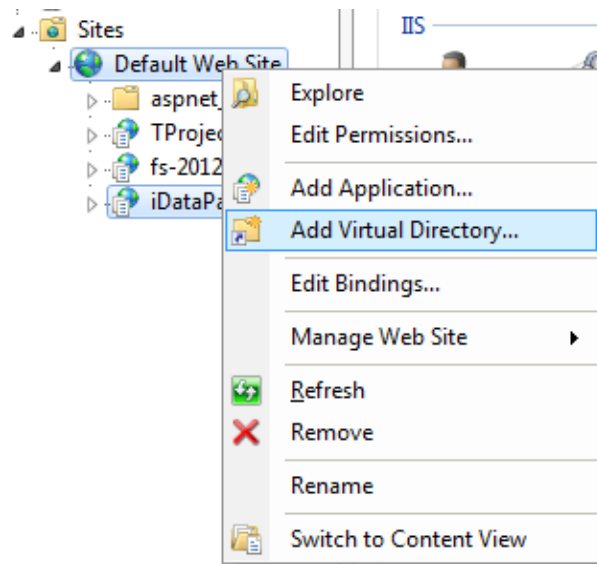


Figura 5-47. Diretório Virtual

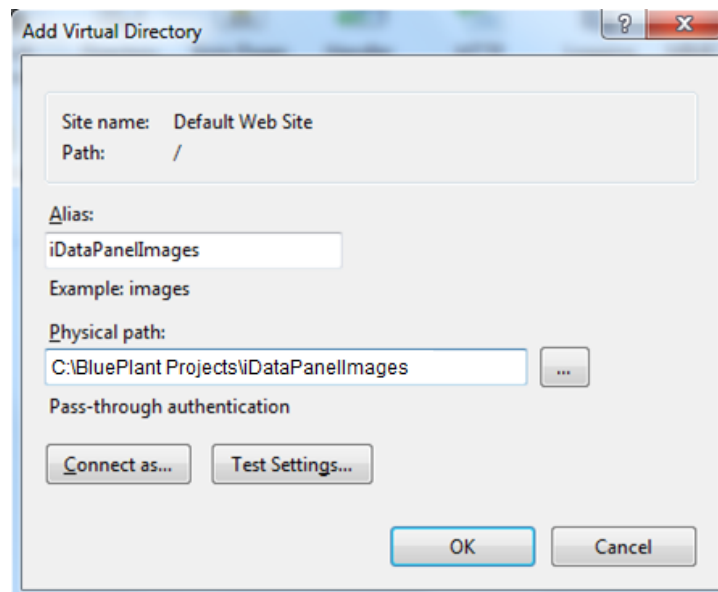


Figura 5-48. Acrescentar Diretório Virtual

Security: A opção Security deve estar habilitada para *Everyone*, pelo menos para leitura. *Anonymous Authentication* deve estar habilitada para os serviços conforme pode ser visto na figura a seguir.

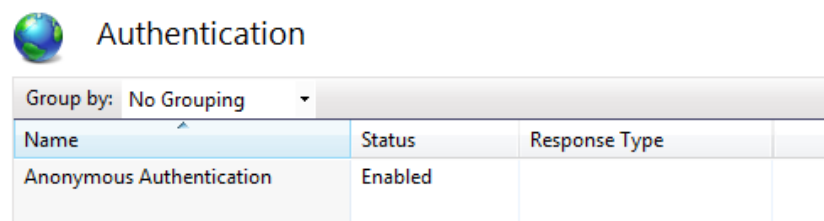


Figura 5-49. Autenticação

O IIS deve ser reiniciado após a configuração dos serviços.

Estudo Dirigido 5-1: formatação de dados para comunicação



Com o auxílio do instrutor configure a comunicação entre o BluePlant e o CLP Nexto Xpress em diferentes formatos, como Ponto Flutuante, usando *Formats*, *TextIO*, *Scaling* e outros a sua escolha.

📌 **DICA:** caso você queira saber mais sobre formatação de dados para comunicação, assista o Webinar a seguir. Nele você vai aprender a configurar a comunicação entre o BluePlant e o CLP Nexto Xpress em diferentes formatos, como Ponto Flutuante, usando *Formats*, *TextIO*, *Scaling*, entre outros.



Estudo Dirigido 5-2: implementação da aplicação Mistura e Secagem

Configure apropriadamente o teste e a execução da aplicação Mistura e Secagem. No projeto do Processo de Mistura e Secagem traduza a aplicação para dois diferentes idiomas a sua escolha com o Google Tradutor. Acrescente três memórias e defina ações para controlar o idioma no tempo de execução.

📌 **DICA:** Consulte o instrutor para orientações adicionais.

ANOTAÇÕES

ANOTAÇÕES

6. Aplicações, Avaliação e Encerramento

Cenários de Sistemas Típicos



Este capítulo descreve os cenários típicos de utilização do BluePlant para atender demandas que contemplam as áreas de produção, utilidades e manufatura.

Além disso, o projeto completo da aplicação pode incluir uma combinação desses cenários com o objetivo de atender as demandas de customização do cliente.

Independentemente do cenário considerado deve-se ter em mente que o BluePlant se constitui em um pacote único de forma que o servidor é sempre um BluePlant. No entanto, todos os módulos do BluePlant como Alarme, Historiador, Dispositivo, Banco de Dados etc, podem estar em computadores diferentes sendo gerenciados pelo Servidor BluePlant gerando assim um sistema distribuído.

Com base nessas premissas podemos ter as configurações descritas a seguir.

Sistema Standalone

Esse sistema caracteriza-se por uma instalação do BluePlant executando o servidor e o cliente SCADA no mesmo computador.



Figura 6-1. Sistema Stand Alone

Sistema de Entradas e Saídas Distribuídas

Esse sistema caracteriza-se por uma máquina BluePlant servidora e módulos de dispositivos rodando em computadores dedicados à comunicação com o processo. Nesse caso o cliente SCADA pode estar no mesmo computador do servidor ou em outro computador. A figura a seguir ilustra essa configuração.

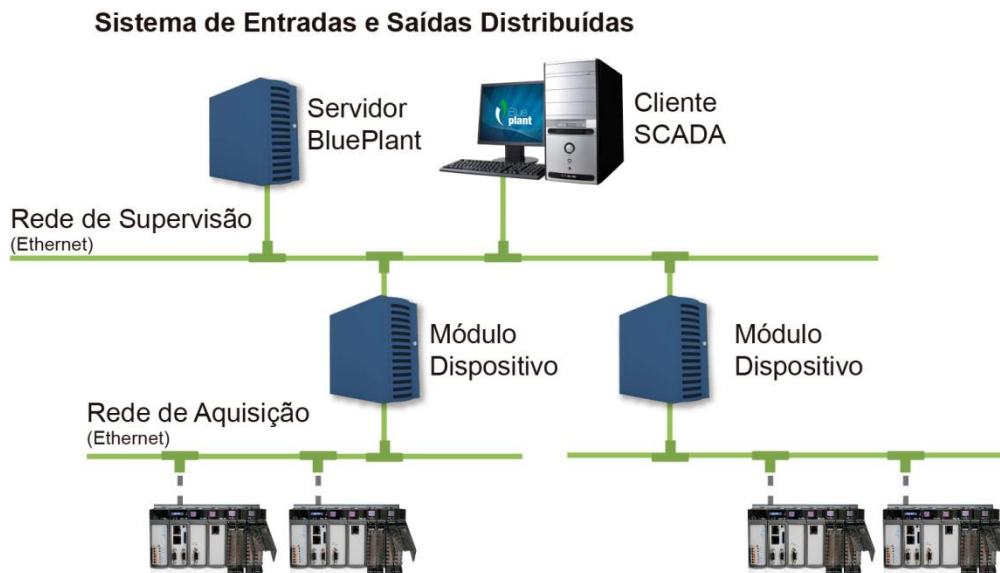


Figura 6-2. Sistema de Entradas e Saídas Distribuídas

Este modelo também é útil em plantas que contenham dispositivos com porta serial ou comunicações com capacidades limitadas. Colocando servidores de E/S no chão de fábrica, como interface com esses dispositivos, é possível otimizar as comunicações em redes lentas ou de baixa largura de banda melhorando o desempenho global.

Apesar da distribuição geográfica dos servidores de E/S em várias plantas, este tipo de arquitetura pode ser configurada como um sistema de cluster único, desde que este seja capaz de suportar vários servidores de E/S.

Sistema Cliente e Servidor

Esse sistema contempla um servidor BluePlant onde os módulos Alarme, Historiador, Banco de Dados estão sendo executados e os Clientes SCADA em outros computadores da rede LAN. A figura a seguir mostra esse sistema.

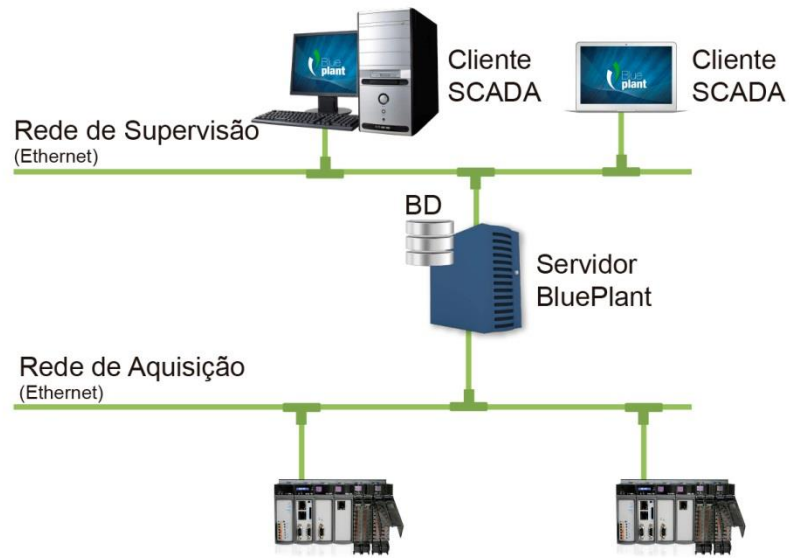
Sistema Cliente Servidor

Figura 6-3. Sistema Cliente e Servidor

A arquitetura cliente-servidor permite que os clientes possam ser distribuídos em vários computadores em uma LAN, criando um sistema que oferece benefícios de flexibilidade e desempenho. Este tipo de arquitetura também pode ser configurada como um sistema de cluster único.

Sistema de Servidor Redundante

Neste caso temos dois computadores distintos executando os servidores BluePlant, onde a redundância é feita automaticamente pelo próprio supervisor. Desta forma é necessário apenas determinar os endereços IP das estações primária e secundária. É possível termos as seguintes configurações de redundância:

O banco de dados de Alarme e/ou Historiador sendo executado em uma terceira máquina dedicada a históricos, conforme a figura a seguir.

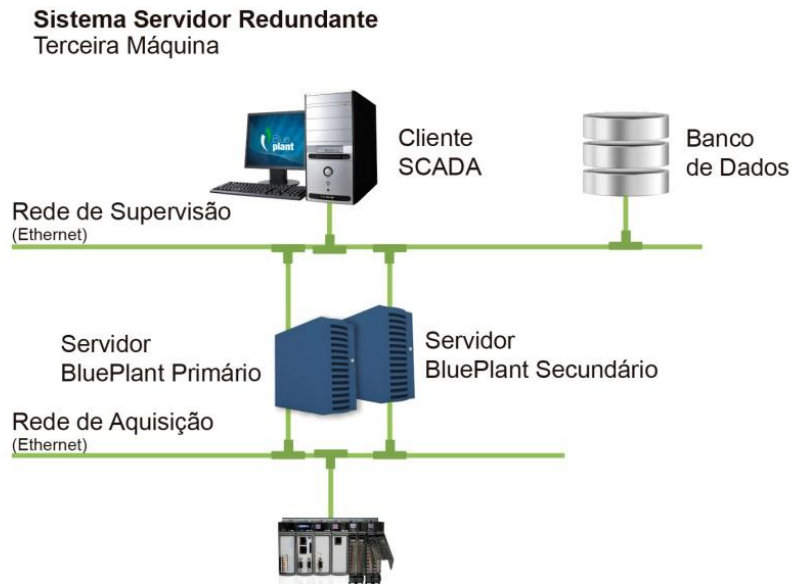


Figura 6-4. Sistema Servidor Redundante Terceira Máquina

Bancos de Dados Locais nos servidores primário e secundário são utilizados para armazenar os históricos dos módulos de Alarme e/ou Historiador, realizando o sincronismo de dados entre eles de forma automática conforme mostrado na figura a seguir.

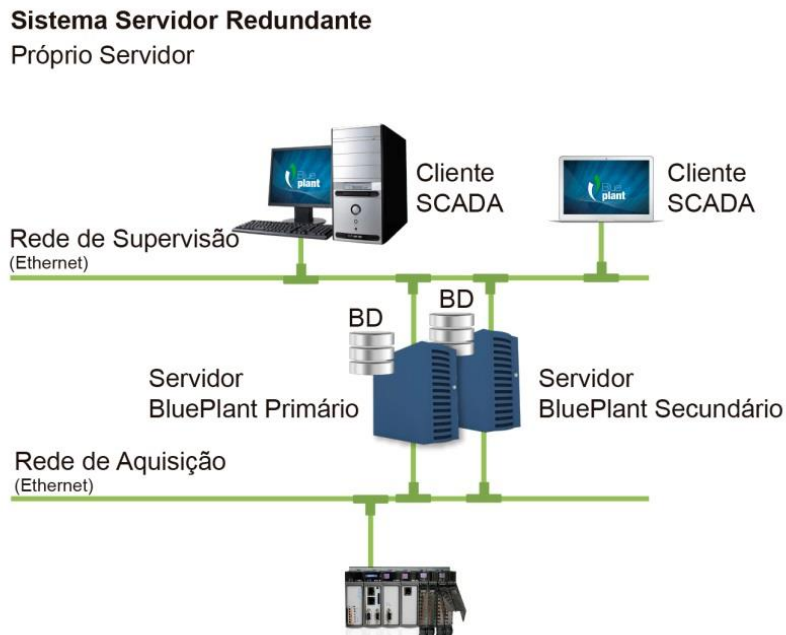


Figura 6-5. Sistema Servidor Redundante Banco de Dados Local

A possibilidade de definir os servidores primário e secundário dentro de um projeto permite a integração do hardware de redundância na infraestrutura do sistema. Isso ajuda a evitar situações em que ocorrerá um erro em um servidor no sistema global, tornando-o inoperante. Sistemas deste tipo são interessantes para garantir operação ininterrupta e confiabilidade na coleta de dados.

Outra configuração possível é a redundância nos dispositivos de comunicação, onde o(s) servidor(es) podem se comunicar com o dispositivo primário ou com um dispositivo reserva. Essa

redundância é habilitada configurando os endereços IPs dos dispositivos. O BluePlant fica responsável por gerenciar a troca entre o dispositivo primário e reserva quando houver algum falha na comunicação entre dispositivo e supervisor.

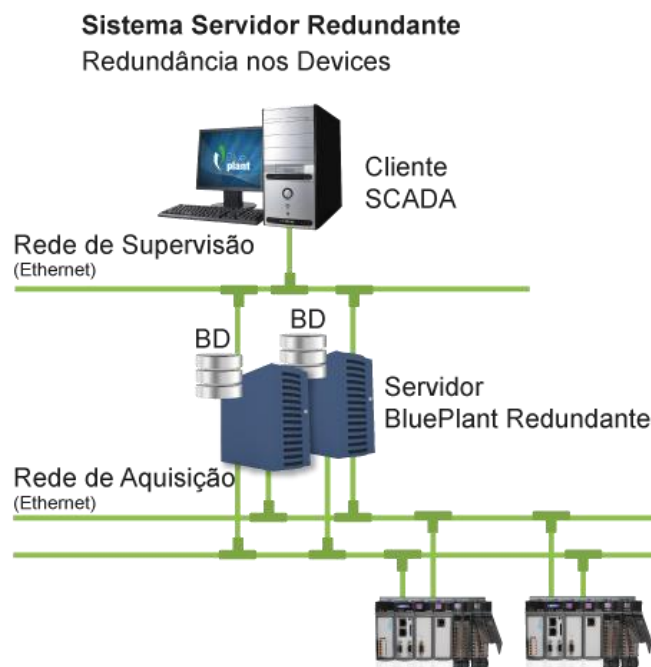


Figura 6-6. Sistema Servidor Redundante com Redundância nos Dispositivos

Mecanismo de funcionamento do Sistema Hot-Stand By

O Sistema Hot-Stand By é um sistema onde temos um Servidor Ativo e um outro Servidor Reserva pronto para assumir as comunicações e funcionalidades do Servidor Ativo em caso de chaveamento. Este chaveamento pode ocorrer por vários motivos:

- Desconexão de cabos do Servidor Ativo;
- Desligamento ou reinicialização do Servidor Ativo;
- Comando do usuário para chaveamento de Servidor Ativo;
- Falhas internas que indicam a indisponibilidade do Servidor Ativo no Sistema.

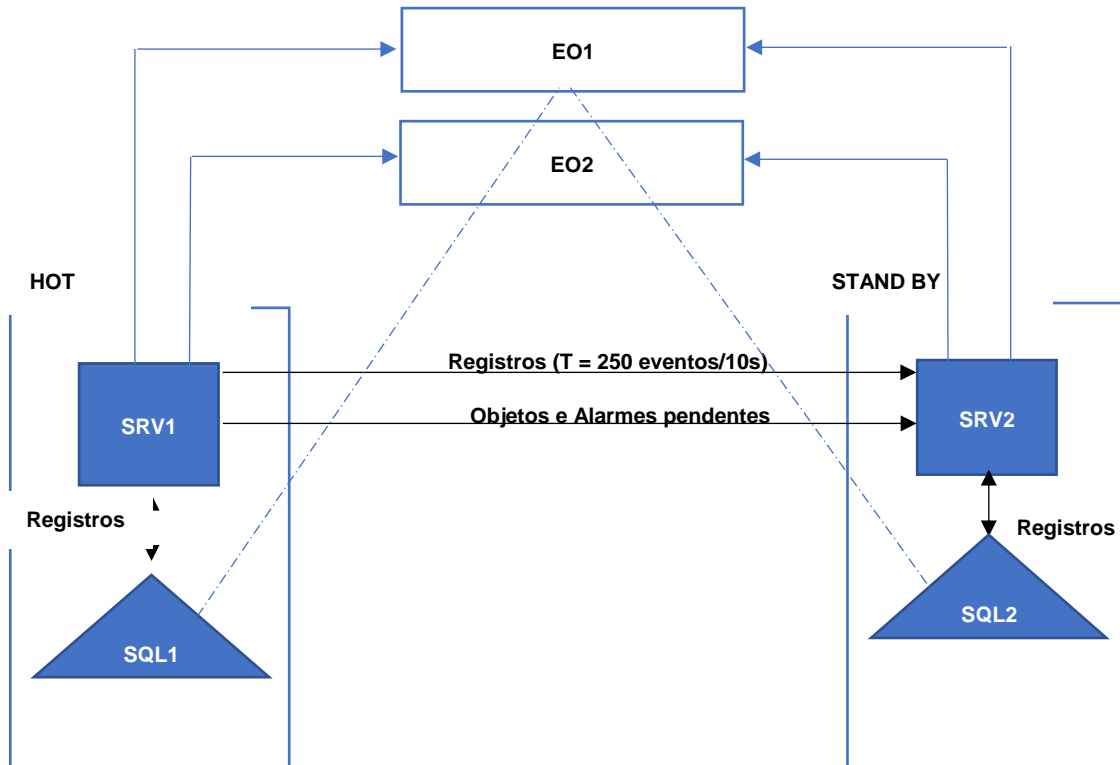


Figura 6-7. Mecanismo de funcionamento do Servidor Redundante

Onde:

- **EO1** = ESTAÇÃO DE OPERAÇÃO 1;
- **EO2** = ESTAÇÃO DE OPERAÇÃO 2;
- **SRV1** = SERVIDOR 1 (BluePlant);
- **SRV2** = SERVIDOR 2 (BluePlant);
- **SQL1** = BANCO DE DADOS 1;
- **SQL2** = BANCO DE DADOS 2;
- **Linha tracejada** = Banco cruzado.

Nesta situação não existem problemas e os Servidores estão trabalhando normalmente na arquitetura. Neste processo basicamente ocorre o seguinte:

- Todas as remotas estão comunicando com os drivers do Servidor Ativo;
- Os drivers do Servidor Stand By estão prontos para serem ativados, porém estão pausados e não atuam no sistema (é possível ver os sistemas em Pause na tela do Startup do Servidor Stand by);
- Cada servidor comunica apenas com o seu próprio banco de dados. Não existe comunicação direta entre os bancos de dados ou entre um servidor e o banco de dados do outro servidor. O Servidor Ativo sincroniza informações do seu banco de dados com o outro banco de dados somente através do Servidor Stand By;

- A cada 10 segundos, 250 informações novas que estão registradas no Banco de Dados 1 são sincronizadas com o Banco de Dados 2 conforme o processo descrito acima;
- Objetos pendentes são estados de objetos da aplicação que são atualizados entre os servidores (propriedades de Tags, dados de símbolos etc.);
- As estações de operação acessam a aplicação e trazem informações de ambos os servidores Hot e Stand By, “concatenando” (merge) informações dos dois bancos de dados, e, portanto, mostrando todas as informações de maneira íntegra (Banco Cruzado). Além disso, existe uma camada da aplicação que cuida para que eventos duplicados não sejam registrados no banco de dados. Este mecanismo é uma defesa do sistema para a seguinte situação de operação degradada.

Operação degradada

Ao retirar apenas um cabo de rede de um dos servidores, deve-se implementar um script no sistema para avaliar a situação como uma perda de disponibilidade e indicar uma pop up e um alarme para que o operador possa escolher chavear o sistema para que o Servidor ativo seja o servidor com os dois cabos conectados. Se o novo servidor ativo perder um cabo de rede, a pop up também deverá aparecer dando a opção de trocar o servidor ativo. De qualquer forma, o sistema ainda continua funcionando normalmente e o ideal é que a conexão seja recuperada o quanto antes para evitar a indisponibilidade iminente de um dos servidores. A perda de um ou os dois cabos de rede devem gerar alarmes no sistema.

Quando ocorre uma perda do servidor ativo (indisponibilidade do servidor no sistema), o stand By assume. Este aspecto é crítico pela seguinte razão: o chaveamento pode ter ocorrido quando existiam eventos no Banco de Dados 1 a serem sincronizados com o Banco de Dados 2, ou seja, entre o período de 10 segundos conforme mencionado anteriormente. Observação: Diminuir este tempo não garante mais disponibilidade ao sistema, pois uma interrupção por falha dupla (desconexão dos dois cabos de rede, perda de alimentação do servidor, erro de hardware etc.) pode ocorrer mesmo que este período seja pequeno. Além disso, esta razão não representa perda de eventos, pois os eventos foram registrados. A partir disso, é necessário recuperar o servidor na rede para que o funcionamento do Banco Cruzado mostre todos os eventos nas estações de operação.

Os desdobramentos dessa razão são:

- Os registros dos bancos de dados ficam desbalanceados, ou seja, o que existe em um banco de dados é diferente do que existe no outro. Por isso que as estações de operação trazem informações de ambos os bancos de dados (Banco Cruzado), ou seja, para o operador esta questão se torna transparente, desde que o sistema seja recuperado conforme será descrito na próxima seção.
- Se uma avalanche de eventos ocorrer, e, sem que o tempo de sincronização seja esperado, ocorrer então, uma troca de servidor ativo causada pela indisponibilidade de um servidor, o desequilíbrio entre os bancos de dados será suficiente para que os registros não sejam visualizados integralmente pelas estações de operação. O cálculo do tempo estipulado para poder ser realizado este procedimento está descrito na sequência.

$$\text{taxa média} \left[\frac{\text{eventos}}{\text{segundo}} \right] = \frac{\text{eventos} [\text{eventos}]}{\text{tempo} [\text{segundos}]}$$

$$\frac{250[\text{eventos}]}{10[\text{segundos}]} = \frac{35121 [\text{eventos}]}{\text{tempo} [\text{segundos}]}$$

$$\text{tempo} = \frac{35121 [\text{eventos}]}{25 [\text{eventos/segundo}]}$$

$$\text{tempo} \cong 23 \text{ minutos}$$

Recuperação do Sistema

O Sistema é dotado de alarmes que podem indicar ao operador a falta de um Servidor. Se nada de grave ocorreu ao servidor, por exemplo, apenas os cabos de rede desconectados, ou um desligamento acidental, ou falha de alimentação, então ele pode ser facilmente recuperado no sistema.

Quando este servidor retorna à rede, sendo o SRV1 ele ocupará a função de Servidor Ativo, caso não seja, ficará como o Servidor Stand By.

Sistema de Controle

Neste caso podemos ter vários servidores em plantas distintas (projetos diferentes) em uma arquitetura distribuída. Essa configuração permite que, em uma sala de controle, se possa acessar qualquer uma dessas plantas através de clientes específicos. É importante mencionar que os clientes das plantas não estarão integrados em uma só máquina, de forma que é necessário especificar qual planta deseja-se visualizar. A figura a seguir ilustra esse sistema.

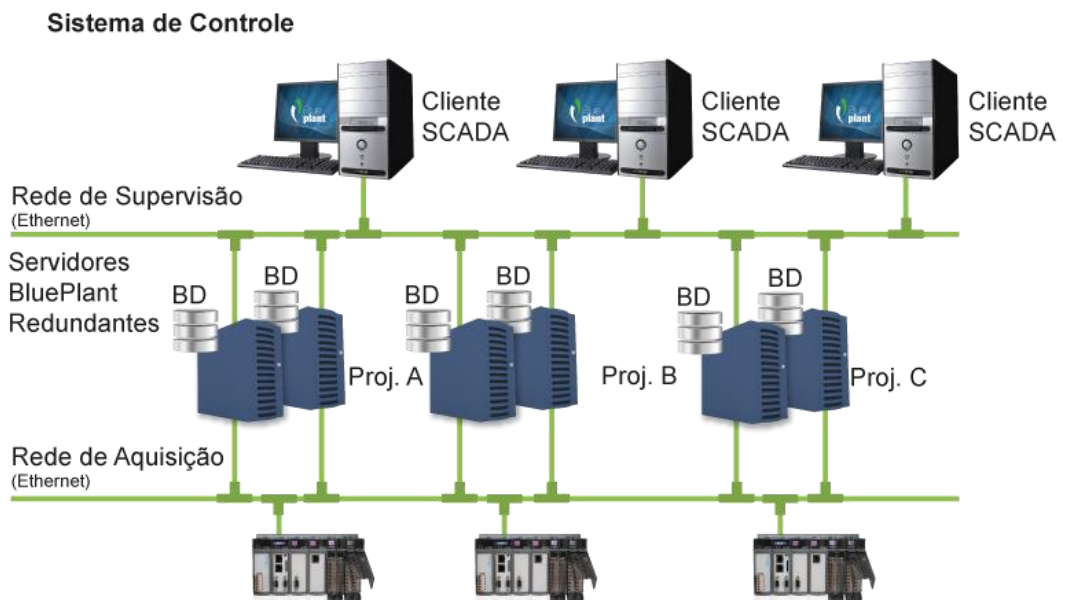


Figura 6-8. Sistema de Controle

Nesse cenário, o sistema é organizado em locais discretos controlados por operadores locais apoiados pelos servidores redundantes locais. Ao mesmo tempo, pode-se caracterizar um nível de gestão em uma sala de controle central que monitora simultaneamente todos os sites. Cada

site é representado no projeto como um cluster separado, agrupando seus servidores primário e standby.

Sistema de Controle Distribuído e Distribuído Redundante

Esse sistema contempla uma máquina BluePlant servidora e módulos Alarme, Historiador, Banco de Dados e Clientes SCADA em outros computadores da rede LAN, de maneira que cada módulo é executado em um computador distinto. A figura a seguir mostra esse sistema.

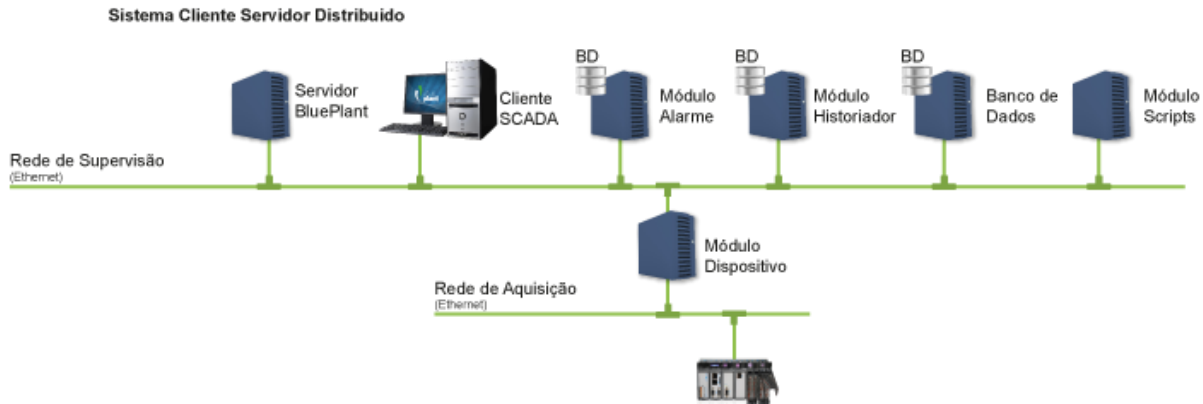


Figura 6-9. Sistema de Controle Distribuído

A arquitetura cliente-servidor permite que os módulos sejam distribuídos através de vários computadores em uma LAN, criando um sistema que oferece benefícios de flexibilidade e desempenho. Cada componente é identificado, no âmbito do projeto, por um endereço, permitindo que sua localização e requisitos de hardware possam ser considerados de forma independente. Este tipo de arquitetura também pode ser configurada como um sistema de cluster único.

O próximo cenário trata-se de um sistema redundante, mas com seu par em localização diferente. Esta configuração demanda uma rede com elevadas taxas de transmissão entre as localidades, visto que a quantidade de dados trocada entre os pares servidores é expressiva. A figura a seguir mostra essa configuração.

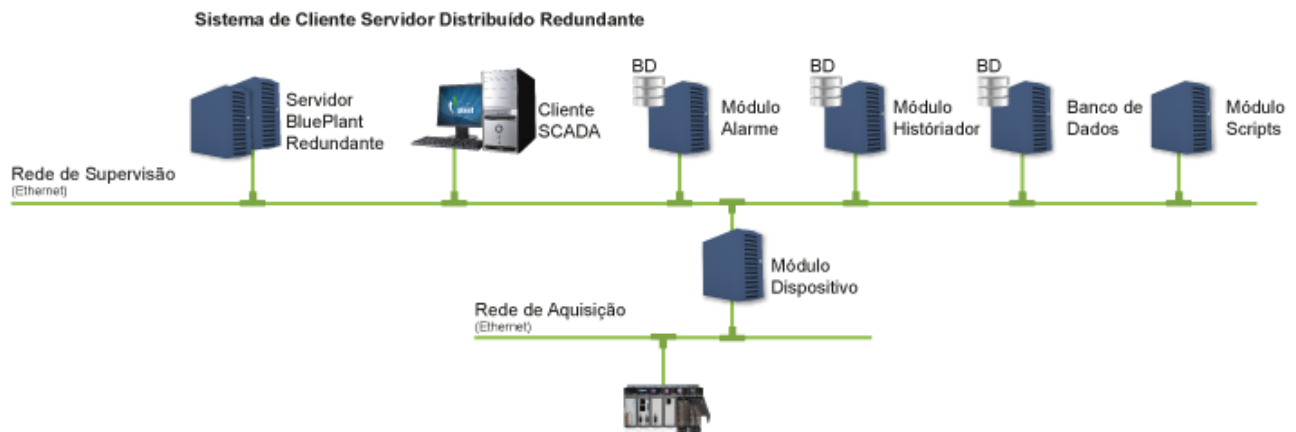


Figura 6-10. Sistema de Controle Distribuído Redundante

Neste cenário, o projeto contempla sites operados localmente cada um contendo seu próprio conjunto de servidores e clientes.

No caso de algum site tornar-se inoperante, o monitoramento ininterrupto é garantido distribuindo-se os servidores primário e secundário entre os diferentes sites, ou colocando-se os servidores como secundário numa localização central.

Um cluster é usado para definir o papel dos servidores em cada local, podendo ser visualizado em um projeto comum, sendo executado em cada cliente.

Sistema de Compartilhamento de Carga

É similar ao sistema cliente e servidor mencionado anteriormente, mas com redundância somente no módulo de dispositivo. Esse sistema possibilita uma melhor utilização da infraestrutura disponível, visto que permite o compartilhamento da carga de trabalho entre diferentes computadores e UCPs.

Esta abordagem pode ser usada para melhorar o desempenho da rede, os tempos de acesso de dados e a estabilidade geral do sistema.

Além disso, por meio de clusters é possível executar múltiplos servidores do mesmo tipo em um computador único.

Por outro lado, a distribuição dos servidores em dois clusters permite que eles operem como unidades redundantes entre si reduzindo o número de computadores exigidos.

Exercícios Dirigidos



Esta seção contempla exercícios dirigidos que ilustram a utilização da ferramenta em cenários específicos.

Consulte o instrutor para eventuais detalhamentos ou ajustes nos procedimentos indicados, pois podem ocorrer pequenas alterações dependendo da rota de aprendizagem adotada pelo instrutor em função do perfil da turma e de especificidades de cada edição do treinamento.


Exercício 1: Display MENU

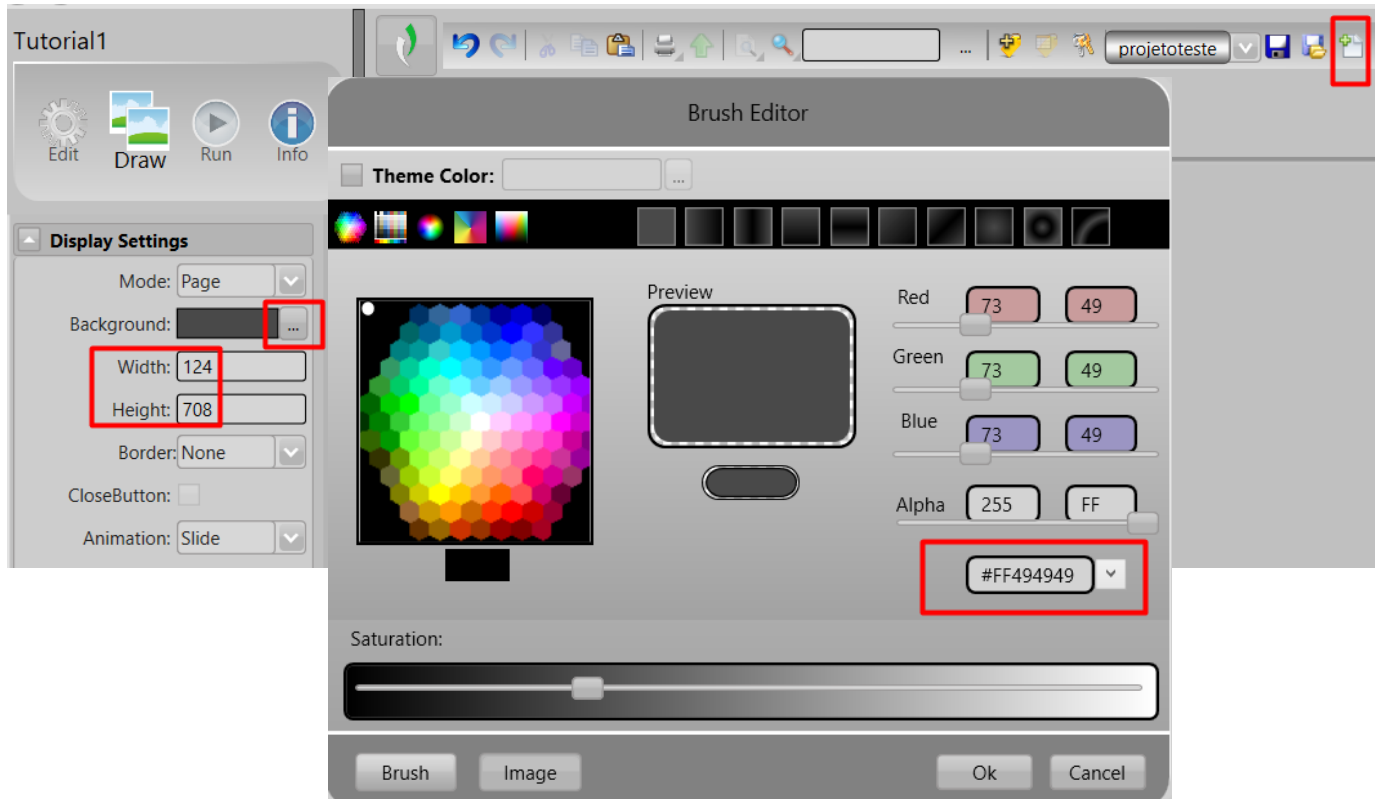
Objetivo


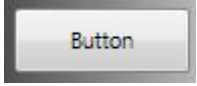
Criar um display do tipo menu com sete botões para alternar entre as telas da aplicação. Este menu deve ter a aparência indicada a seguir.

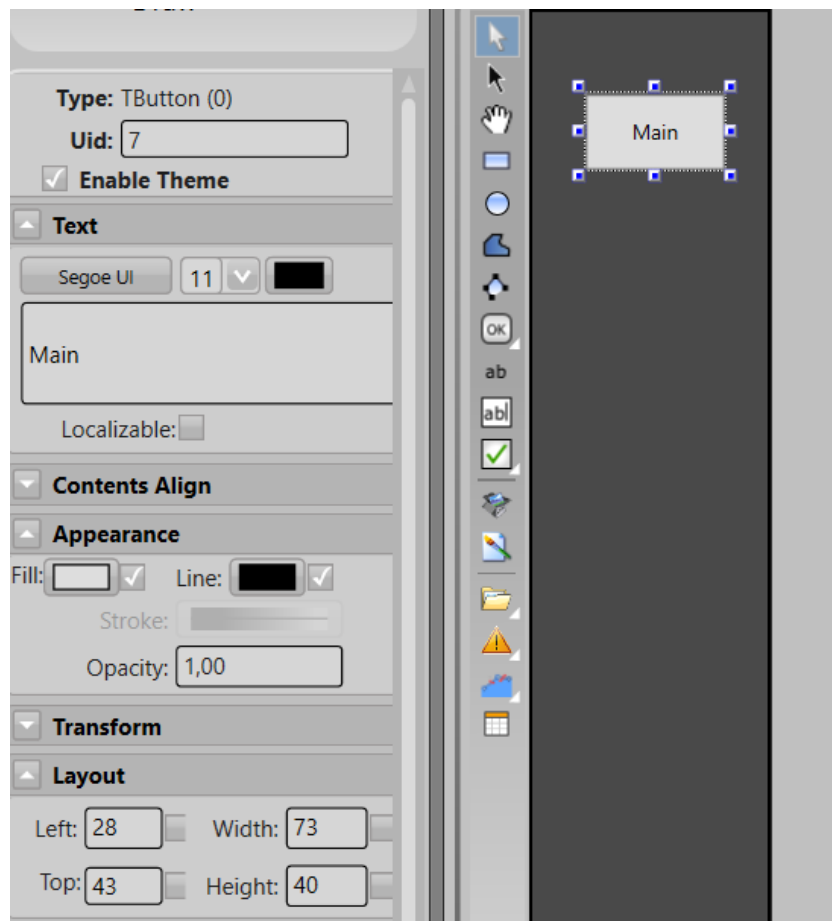


Procedimento

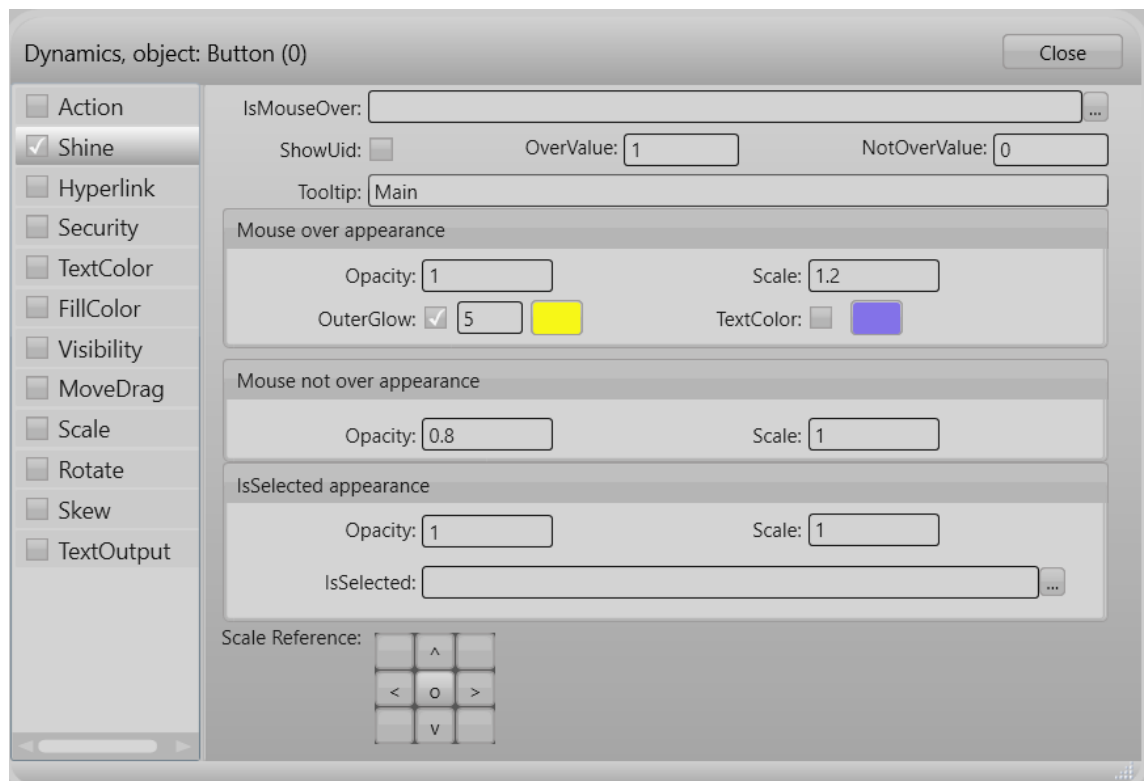
1. Clique no ambiente *Draw*.
2. Clique no ícone  na barra de ferramentas superior para criar um novo display.
3. Configure o novo display na opção *DisplaySettings* conforme mostrado a seguir.






4. Altere a cor de fundo.
5. Altere a largura para 124 e a altura para 708.
6. Clique no ícone  na barra de ferramentas *Draw* para criar um novo botão.
7. Desenhe um botão no display conforme mostrado... .
8. Clique no botão inserido e altere o texto na propriedade do Botão para "Main" no lado esquerdo. Veja a seguir.



9. Dê um duplo clique no botão *Main* para abrir a configuração da dinâmica do mesmo.
10. Configure a dinâmica do botão na caixa de diálogo mostrada a seguir.



11. Clique no ícone  na barra de ferramentas superior para visualizar uma prévia do display.
12. Feche a visualização prévia do display.
13. Faça seis cópias do botão Main e altere os textos dos botões para: *Device*, *Alarm Online*, *Alarm History*, *Trend*, *Dataset* e *Report*.
14. Selecione todos os sete botões via tecla Shift.
15. Clique no ícone  para alinhar à esquerda e no ícone  para uniformizar os espaços entre eles. Ambos os ícones estão localizados na barra de ferramentas inferior.

Exercício 2: Display MAIN


Objetivo

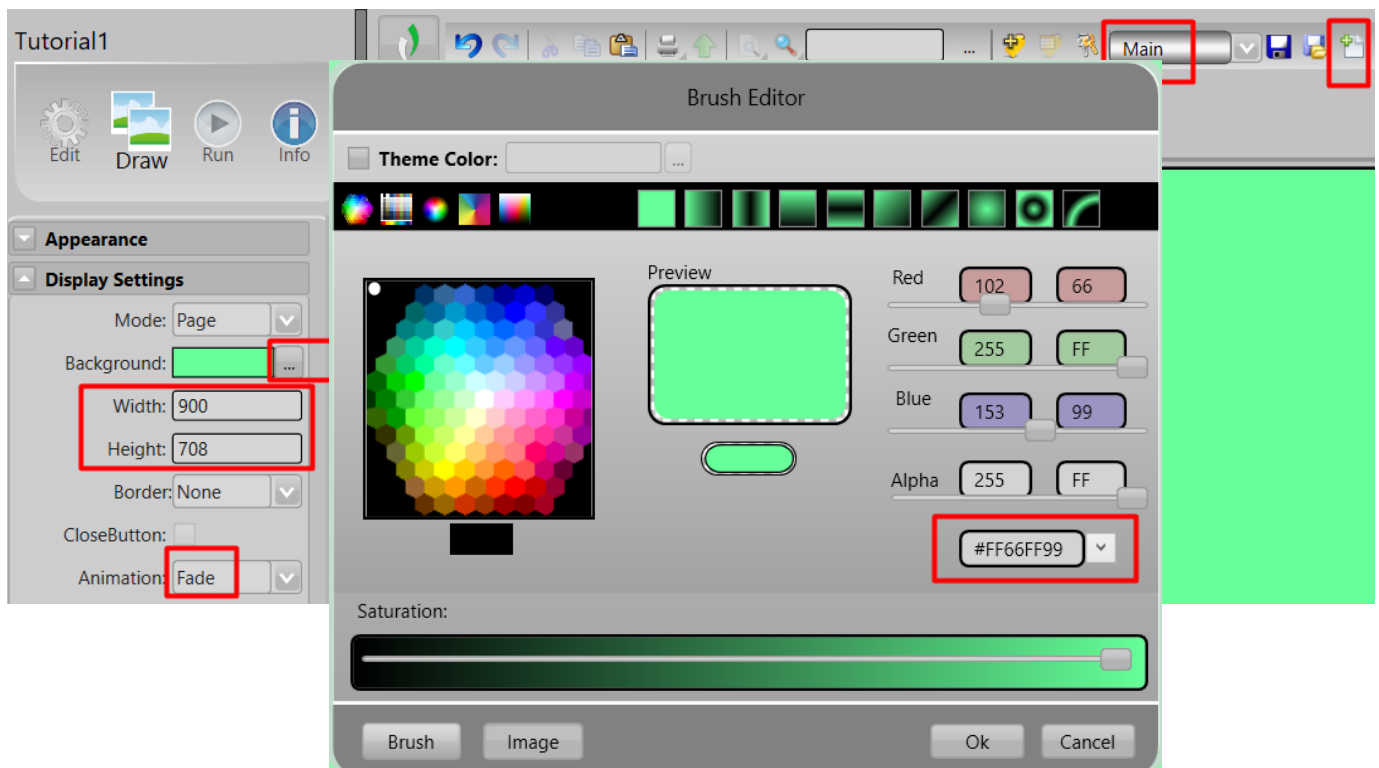
Criar o display Main. Ele simulará duas áreas onde a temperatura está aumentando e é necessário utilizar dois ventiladores para manter ou reduzir a temperatura.

Procedimento

1. Clique na opção *Edit* e, na sequência, na seção *Tags*.
2. Na guia *Objects* crie as Tags conforme mostrado a seguir.

Nome	Tipo	Parâmetros	Array
area	AnalogInt		
temperature	AnalogInt		2
fan2Enabled	Digital		2
fan1Enabled	Digital		2

3. Clique na opção *Draw* e, depois, no ícone  na barra de ferramentas superior para criar um novo display.
4. Configure o novo display em *DisplaySettings* conforme mostrado a seguir.

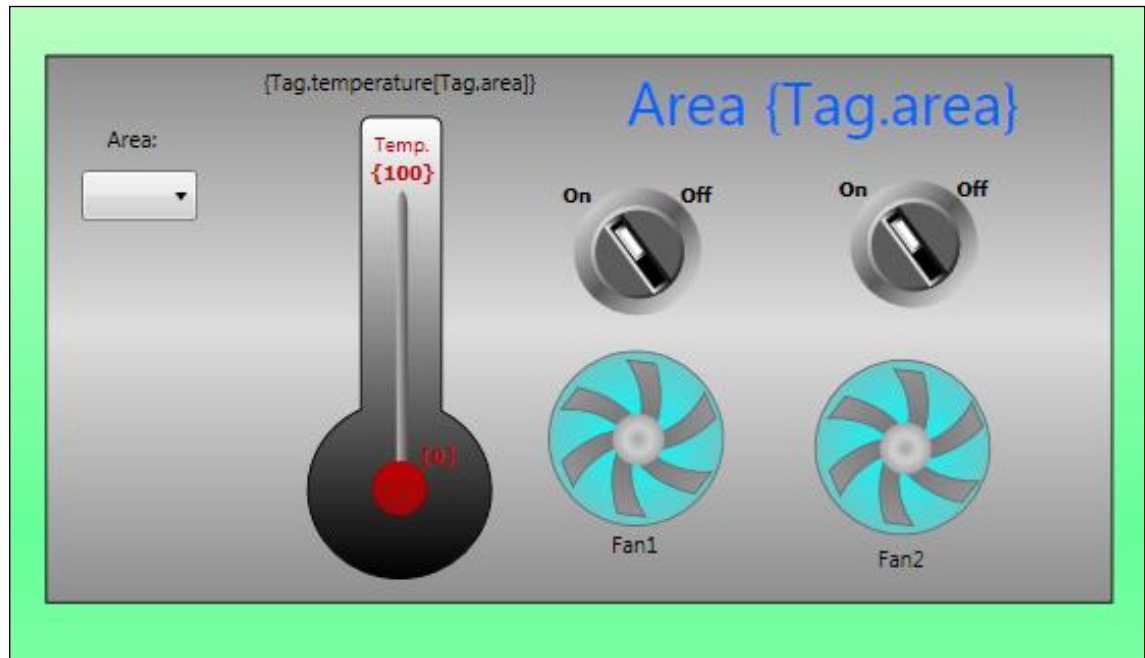


5. Clique com o botão direito do mouse no novo display e selecione *Insert Symbol*.
6. Selecione na categoria *ComboBox* a opção *Blower*.
7. Escolha o símbolo *RotateBlower2* e clique no botão *Ok*.

8. Insira o símbolo *RotateBlower2* no display.
9. Dê um clique com o botão direito do mouse no símbolo e selecione a opção *Unmake symbol*.
10. Crie um novo símbolo chamado *FanTraining* considerando a seguinte configuração dinâmica (Dynamics configuration): Saída de Texto (TextOutput) com a seguinte Expressão (Expression).

#Text:"Fan1"

11. Usando outros símbolos da biblioteca juntamente com o símbolo *FanTraining*, desenhe a tela principal conforme mostrado a seguir.



12. Configure a Caixa de Seleção (ComboBox) "Area" com os seguintes parâmetros: Tag.area e Fonte (Source): 1;2; (opção Text).
13. Configure (Symbol configuration) o símbolo *Gauge* (Indicador) com as seguintes propriedades: Label Temp, variação de 0 (min) a 100 (max) e indicação de nível dada por Tag.temperature[Tag.area].
14. Configure os novos símbolos *FanTraining* conforme indicado a seguir.
 - Fan 1: controle via Tag.fan1Enabled[Tag.area], variação de 0 (min) a 1000 (max), valor da rotação (RotateValue) via Client.Millisecond e Texto (Text) "Fan1".
 - Fan 2: controle via Tag.fan2Enabled[Tag.area], variação de 0 (min) a 1000 (max), valor da rotação (RotateValue) via Client.Millisecond e Texto (Text) "Fan2".
15. Configure os símbolos das chaves (SelectorSwitch) conforme segue.
 - Switch 1: controle via Tag.fan1Enabled[Tag.area], Label1 Off e Label2 On.
 - Switch 2: controle via Tag.fan2Enabled[Tag.area], Label1 Off e Label2 On.
16. Configure a aba *codebehind* e na função *DisplaysOpen* o código a seguir.

```

Public Sub DisplayIsOpen()
If @Tag.temperature(@Tag.area) < 100 Then
@Tag.temperature(@Tag.area) += 1
End If

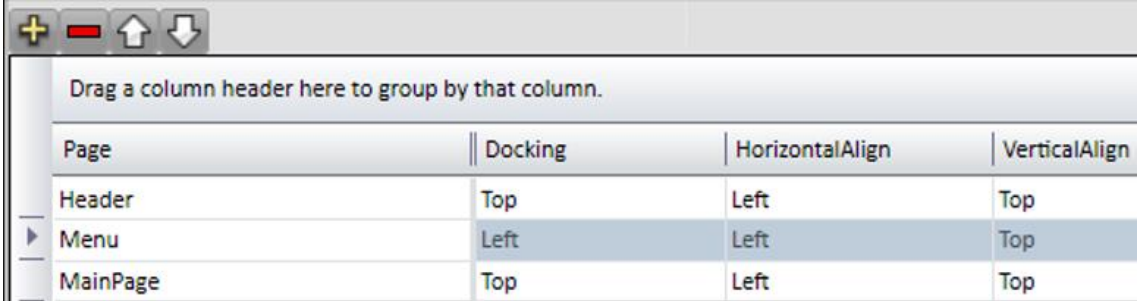
If @Tag.fan1Enabled(@Tag.area) = 1 Then
@Tag.temperature(@Tag.area) -= 1
End If

If @Tag.fan2Enabled(@Tag.area) = 1 Then
@Tag.temperature(@Tag.area) -= 1
End If

If @Tag.temperature(@Tag.area) <= 0 Then
@Tag.temperature(@Tag.area) = 0
End If
End Sub

```

17. Vá para o display "Menu".
18. Acrescente a ação dinâmica (*Dynamics Action*) ao botão Main e ajuste a ação para...
OpenDisplay = "MainPage".
19. Clique na opção Edit ... e na seção Display.
20. Clique na aba *Layouts*.
21. Configure o layout de *Startup* (Inicialização) conforme mostrado a seguir.



Page	Docking	HorizontalAlign	VerticalAlign
Header	Top	Left	Top
Menu	Left	Left	Top
MainPage	Top	Left	Top

Exercício 3: Dispositivos

Objetivo

Criar a um driver de comunicação usando o protocolo “Modbus Master – TCP and RS232” sobre uma interface TCP/IP.

Procedimento

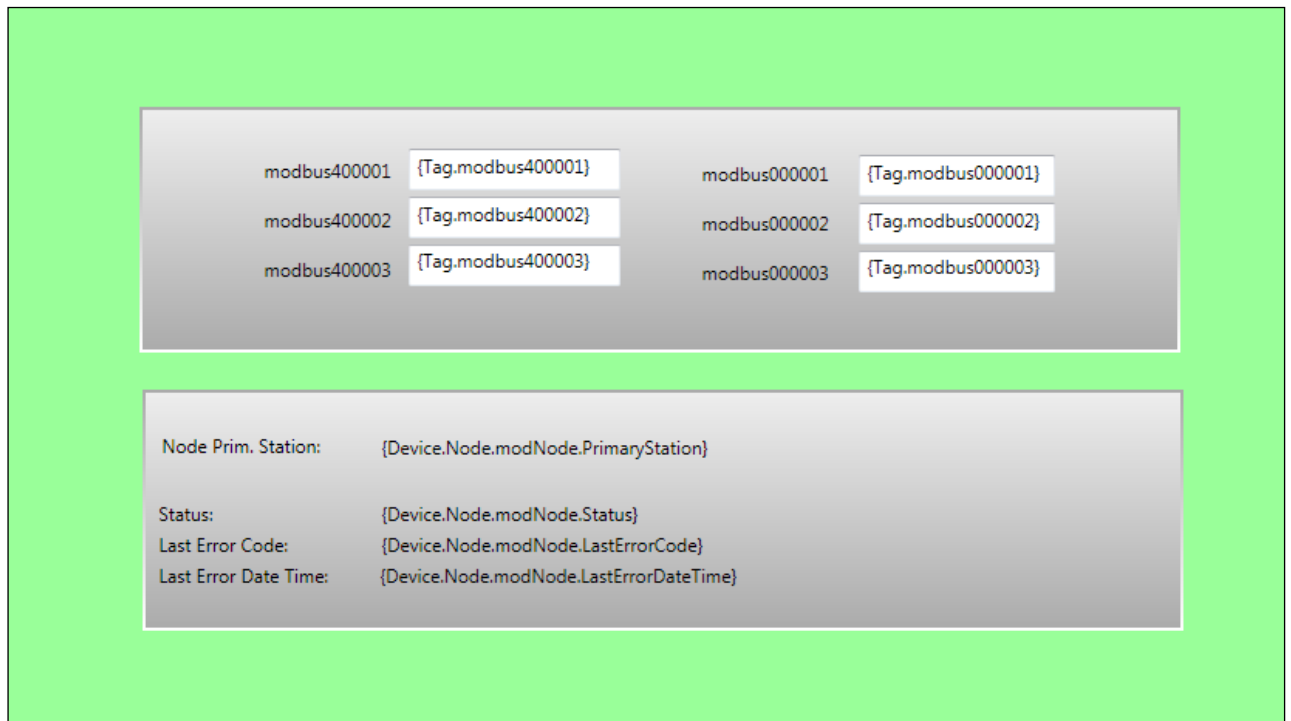
1. Clique na opção *Edit* e na seção *Tags*.
2. Crie as Tags mostradas a seguir.

Nome	Tipo
modbus000001	Digital
modbus000002	Digital
modbus000003	Digital
modbus400001	AnalogInt
modbus400002	AnalogInt
modbus400003	AnalogInt

3. Clique na opção *Edit* e na seção *Devices*.
4. Clique na aba *Channel*.
5. Clique no botão *Create New...*
6. Selecione o protocolo “Modbus Master – TCP and RS232”.
7. Selecione a interface TCP e clique no botão Ok.
8. Dê um duplo clique na célula *Protocol/Options* e altere a codificação para RTU TCP.
9. Clique na aba *Node*.
10. Digite “modNode” na célula do nome.
11. Selecione o canal Modbus na célula *Channel* e configure a *PrimaryStation*.
12. Pressione a tecla *Enter* para confirmar a configuração.
13. Clique na aba *Points* e configure-a conforme mostrado a seguir.

Nome da Tag	Nó	Endereço	Tipo de Dado	Tipo de Acesso
Tag.modbus000001	modNode	000001	Native	ReadWrite
Tag.modbus000002	modNode	000002	Native	ReadWrite
Tag.modbus000003	modNode	000003	Native	ReadWrite
Tag.modbus400001	modNode	400001	Native	ReadWrite
Tag.modbus400002	modNode	400002	Native	ReadWrite
Tag.modbus400003	modNode	400003	Native	ReadWrite

14. Clique na opção *Draw* e crie um novo display chamado “DevicePage”.
15. Ajuste a largura e altura para 900 x 708. O estilo do display deve parecer como segue.



16. Vá para o display "Menu".

17. Acrescente a ação dinâmica ao botão Device e ajuste a ação para *OpenDispay* = "DevicePage".

Exercício 4: Alarmes

Objetivo

Criar alarmes online para visualização em tempo real.

Procedimento

1. Crie uma nova Tag conforme mostrado a seguir.

Nome	Tipo
FilterAlarm	Text


2. Clique na opção *Edit* e na seção *Alarms*.
3. Crie dois novos grupos de alarme na aba *Groups*. Estes grupos devem ser configurados conforme mostrado a seguir.

Nome	Reconhecimento?	Som	Exibir	Registro de Eventos
WarningGroup	No	None	List	Active
AlarmGroup	Yes	Beep	List	All

4. Clique na aba *Items* e configure os itens conforme mostrado a seguir:

Nome da Tag	Condição	Limite	Grupo	Área	Mensagem
Tag.temperature[2]	Lo	20	Alarm.Group.WarningGroup	2	Area 2 - Temperature is Lo
Tag.temperature[2]	Hi	80	Alarm.Group.WarningGroup	2	Area 2 - Temperature is Hi
Tag.temperature[1]	Hi	80	Alarm.Group.WarningGroup	1	Area 1 - Temperature is Hi
Tag.temperature[1]	Lo	20	Alarm.Group.WarningGroup	1	Area 1 - Temperature is Lo
Tag.temperature[2]	HiHi	90	Alarm.Group.AlarmGroup	2	Area 2 - Temperature is HiHi
Tag.temperature[2]	LoLo	10	Alarm.Group.AlarmGroup	2	Area 2 - Temperature is LoLo
Tag.temperature[1]	LoLo	10	Alarm.Group.AlarmGroup	1	Area 1 - Temperature is LoLo
Tag.temperature[1]	HiHi	90	Alarm.Group.AlarmGroup	1	Area 1 - Temperature is HiHi

5. Clique na opção *Draw* e crie um novo display chamado "AlarmOnlinePage".

6. Clique no ícone  para desenhar um objeto do tipo Alarme no display e configure-o conforme mostrado a seguir.



The image shows a configuration dialog box titled "AlarmWindow configuration". It contains various settings for an alarm window, including control name, list selection, display options, and column configuration.

ControlName:

MaxLines:

List:

Merge Hi and HiHi lines

Show column titles

Show Header:

AllowSort Allow Column Reorder Display Value column as string

Filter: Display Milliseconds

Columns:

- AckStatus
- ActiveTime_Ticks
- TagName
- Group
- Value
- ID
- ItemName
- State
- AckRequired
- Condition

Visible Allow Filter

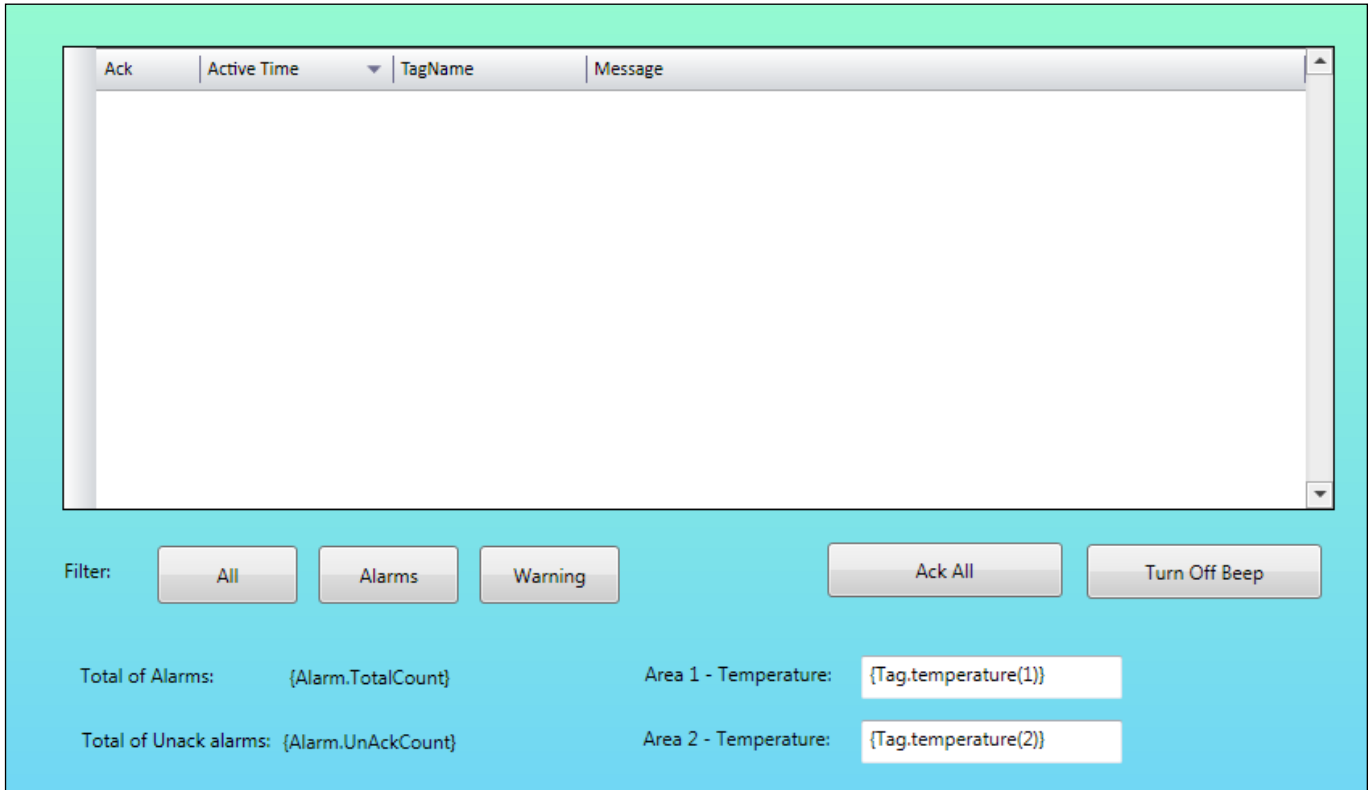
Show in column chooser

Title:

Width:

Sort:

7. O display da página de alarme online terá a aparência mostrada a seguir.



8. Configure (Dynamics configuration), na opção Ação (Action), os botões de filtro conforme indicado a seguir.
 - All: evento ao clicar no botão esquerdo do mouse (MoseLeftButtonDown) com ação de definir valor (SetValue) associada ao objeto Tag.FilterAlarm (valor: "").
 - Alarms: evento ao clicar no botão esquerdo do mouse (MoseLeftButtonDown) com ação de definir valor (SetValue) associada ao objeto Tag.FilterAlarm e valor "Group='Alarm.Group.AlarmGroup'".
 - Warning: evento ao clicar no botão esquerdo do mouse (MoseLeftButtonDown) com ação de definir valor (SetValue) associada ao objeto Tag.FilterAlarm e valor "Group='Alarm.Group.WarningGroup'".
9. Configure o botão de reconhecimento geral dos alarmes (*Ack All*) com ação dinâmica ajustando-o para "Alarm.AckAll".
10. Configure o botão Turn off Beep (Som ao desligar) com ação dinâmica ajustando-o para "Client.BeepOff".
11. Vá para o display "Menu".
12. Acrescente a ação dinâmica ao botão Alarm e ajuste a ação para *OpenDispay = "AlarmOnLinePage"*.

Exercício 5: Histórico de Alarmes


Objetivo

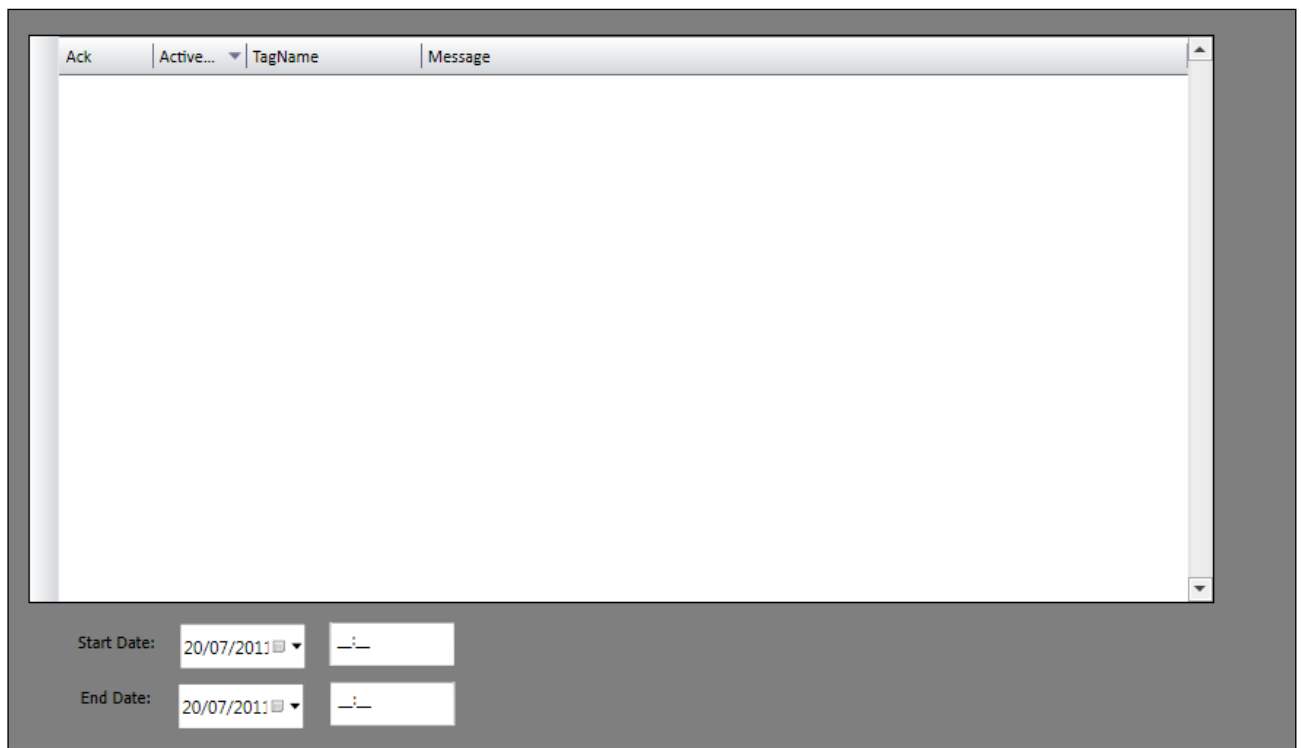
Criar um histórico de alarmes.

Procedimento

1. Crie duas novas Tags conforme mostrado a seguir:

Nome	Tipo
StartDate	DateTime
EndDate	DateTime

2. Clique na opção Draw e crie um novo display chamado "AlarmHistoryPage".
3. Dê um clique com o botão direito do mouse no ícone  e selecione a data de início e final no controle apropriado. Este display terá uma aparência como ilustrado a seguir.



4. Configure o objeto de alarme conforme mostrado a seguir:

5. Configure o seletor de data e o controle de *DateTimeTextBox* com a *StartDate* via *Tag.StartDate* e a *EndDate* via *Tag.EndDate*.
6. Vá para o display "Menu".
7. Acrescente a ação dinâmica ao botão Alarm History e ajuste a ação para *OpenDisplay = "AlarmHistoryPage"*.

Exercício 6: Tendência

Objetivo

Criar um gráfico de tendências para mostrar valores em tempo real.

Procedimento

1. Clique na opção *Edit* e na seção *Tags*.
2. Clique em *Types* e crie um novo tipo chamado "Trend".
3. Configure, no tipo *Trend*, os membros indicados a seguir.

Nome	Tipo	Parâmetros
ref	Reference	AnalogInt;
cursor	AnalogDouble	
pen	AnalogInt	

4. Crie as Tags indicadas a seguir.

Nome	Tipo	Parâmetros	Array
trend	Trend		2
cursorposition	AnalogDouble		
cursorDateTime	DateTime		
History	Digital		

5. Clique na opção *Edit* e na seção *Tags*.
6. Clique na aba *Historian*.
7. Crie uma nova tabela de histórico chamada "SaveOnChange" e configure-a conforme mostrado a seguir.

Historian Table Settings

Database: SQLite

Table name:

Auto create: Save on change:

Trigger: ...

Time Deadband:

Life Time: days

Save Quality: Normalized:

Value Columns Type: ▾

GetSamplesMethod: ...


Description:

Ok Cancel

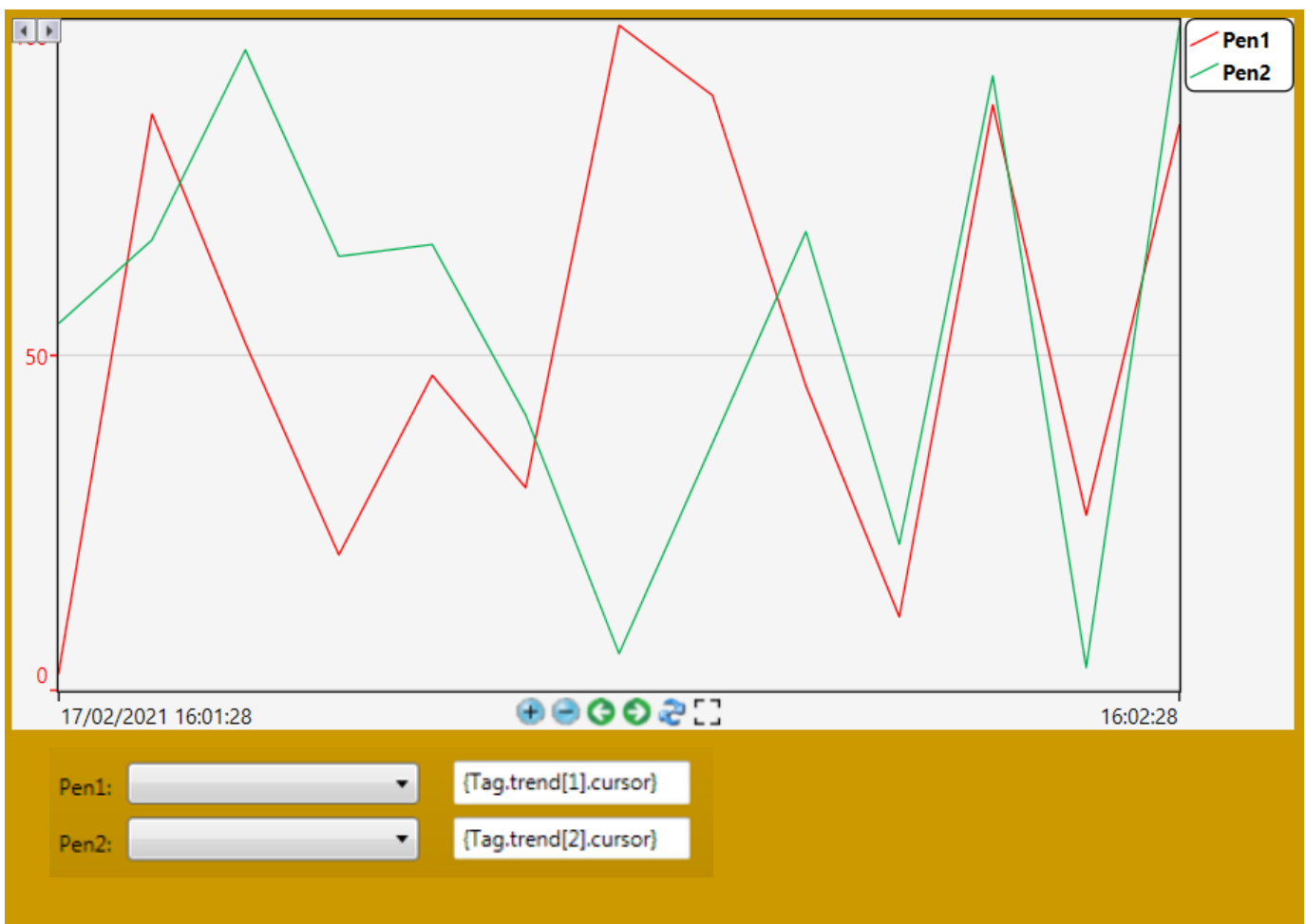
8. Configure o historiador conforme mostrado a seguir.

TagName	HistorianTable
Tag.temperature[1]	Historian.Table.SaveOnChange
Tag.temperature[2]	Historian.Table.SaveOnChange

9. Clique na opção *Draw* e crie um novo display chamado "TrendPage".

10. Clique no ícone  para desenhar um objeto de tendência no display (Control name: Trend1) e configure-o com duas penas (Pens) vinculadas às tags Trend[1].ref e Trend[1].cursor (cor vermelha) e Trend[2].ref e Trend[2].cursor (cor verde) respectivamente.

11. O display da página de tendência terá a aparência mostrada a seguir.



12. Configure os *Combo-box* conforme indicado a seguir.

- Pen1 (Trend[1].ref.Link): fontes (Source...) Temperature[1] e Temperature[2].
- Pen2 (Trend[2].ref.Link): fontes (Source...) Temperature[1] e Temperature[2].

13. Clique na aba *CodeBehind* e acrescente o seguinte código.

```
Private trend As TTrendWindow = Nothing

Public Sub DisplayOpening()

If trend Is Nothing Then
    trend = TryCast(Me.CurrentDisplay.GetControl("Trend1"), TTrendWindow)
End If

If trend IsNot Nothing Then
    trend.Home.Opacity = 0.4
End If
End Sub
```

14. Vá para o display "Menu".

15. Acrescente a ação dinâmica ao botão Trend e ajuste a ação para *OpenDisplay = "TrendPage"*.

Exercício 7: Banco de Dados

Objetivo

Criar uma *Database* (base de dados) e uma *DataGrid* (tabela de dados) para armazenamento e visualização, respectivamente, dos dados da aplicação.

Procedimento


1. Clique na opção *Edit* e na seção *Tags*.
2. Clique em *Types* e crie um novo tipo chamado "Employee".
3. No tipo criado configure seus membros conforme indicado a seguir.

Nome	Tipo
Admission	TDateTime
Name	Text
Identification	AnalogInt

4. Crie as Tags mostradas a seguir.

Nome	Tip
Employee	Employee

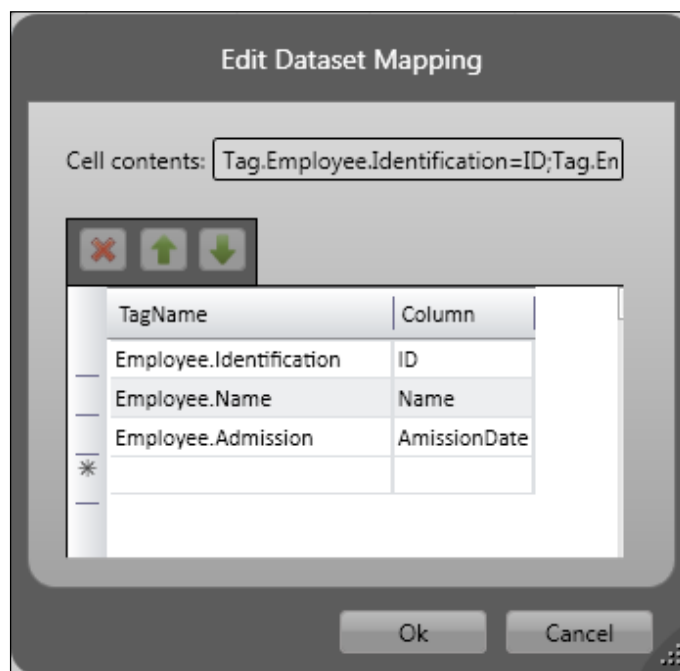
5. Usando o Windows Explorer vá para a <Pasta de instalação do BluePlant> \bp-9.1.6
6. Execute o arquivo *TatsoftDataBuilder.exe*
7. Clique no menu *File* (Arquivo) e em *New Database* (nova base de dados). Denomine-a "Training".
8. Dê um clique com o botão direito do mouse em *Tables* (Tabelas) e clique em *New Table* (Nova Tabela). Identifique-a como "Employees".
9. Configure três colunas conforme ilustrado a seguir.


Table Details		Column Details		
PK	Column Name	Column Type	Max Length	Allow Null
	ID	Int	4	<input type="checkbox"/>
	Name	NChar	50	<input checked="" type="checkbox"/>
	AdmissionDate	Date Time	8	<input checked="" type="checkbox"/>

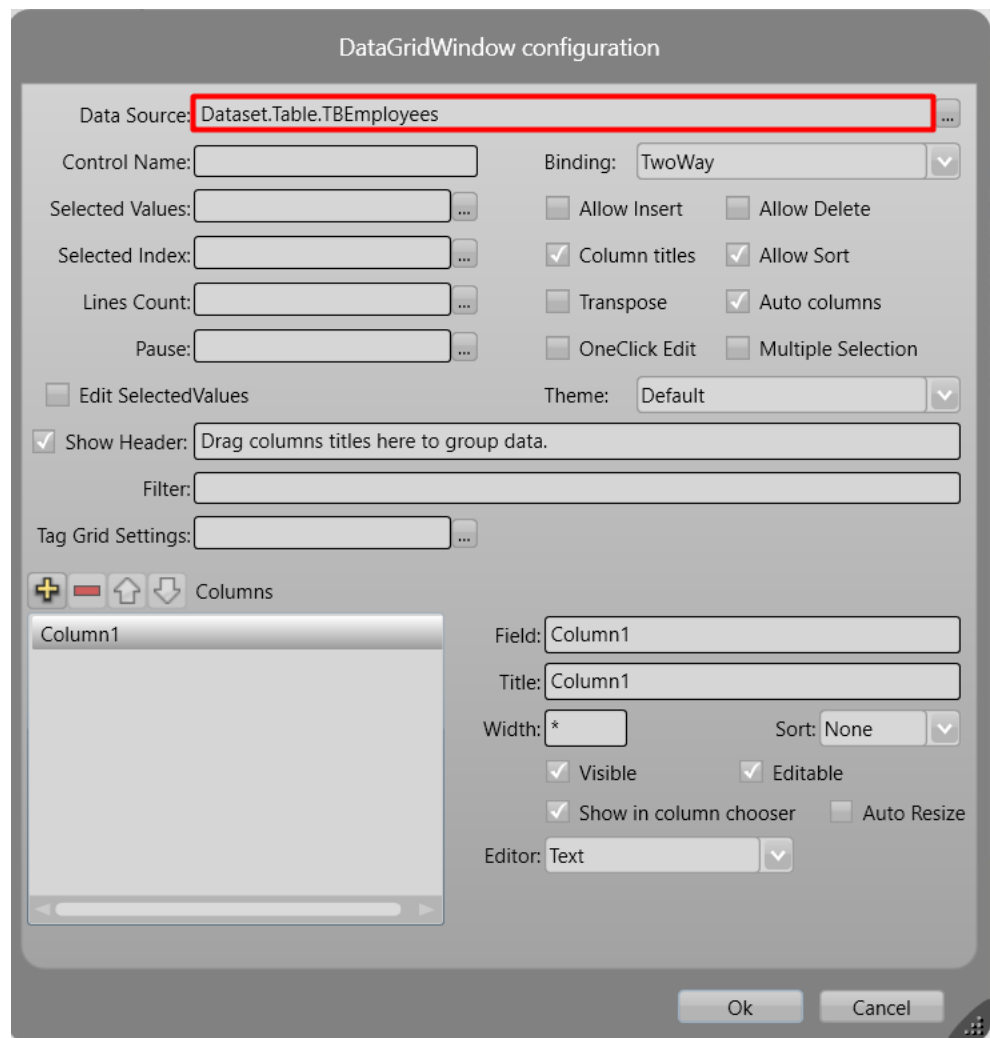
10. Salve-a.
11. Clique em *Data* na tabela *Employees* e insira alguns dados nas colunas.

ID	Name	AdmissionDate
1	John	22/02/2010 00:00:00
2	Mary	01/12/2010 00:00:00
3	Paul	03/05/2010 00:00:00
* <null>	<null>	<null>

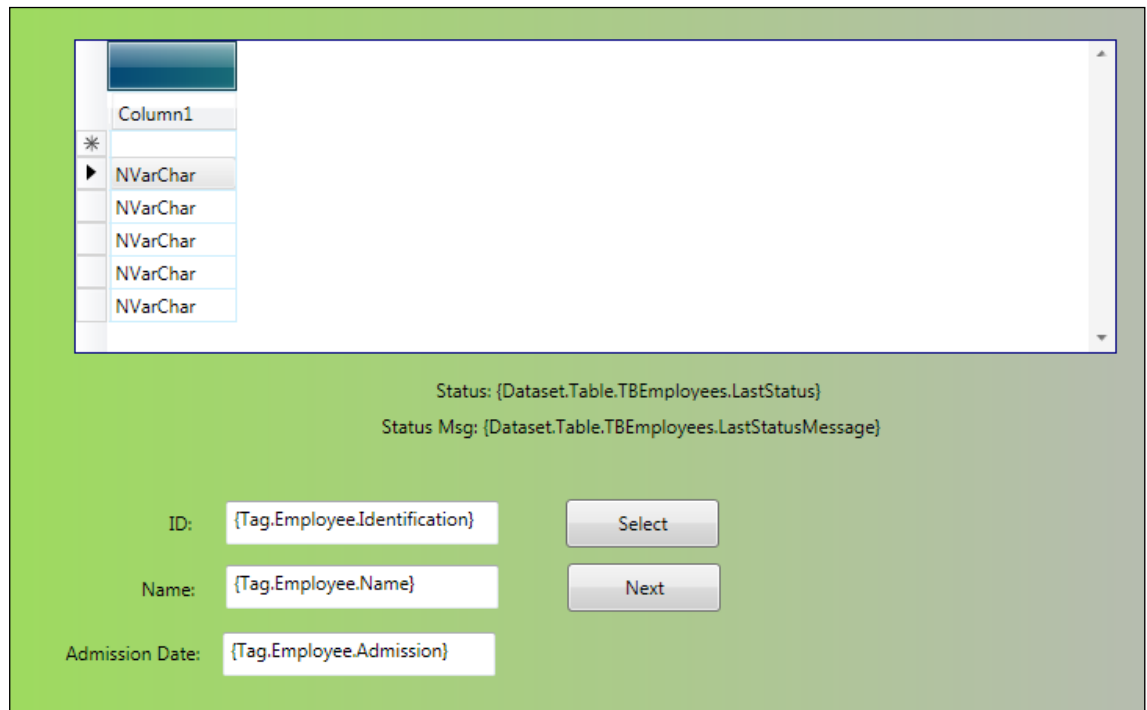
12. Retorne ao BluePlant, clique na opção *Edit* e seção *Datasets*.
13. Clique na aba *DBs* e em *Create new...*
14. Selecione a conexão direta *TatsoftDB 4* e clique no botão *Ok*.
15. Dê um duplo clique na célula *ConnectionString* e preencha o campo *DataSource* como o caminho completo do *Training.tdb4*.
16. Clique no botão *Test*.
17. Clique na aba *Tables*.
18. Configure o nome da tabela como "TBEmployees".
19. Selecione "Provider1" na célula do provedor.
20. Selecione "Employees" na célula *TableName*.
21. Configure a célula *Mapping* conforme indicado a seguir.



22. Clique na opção *Draw* e crie um novo display chamado "DatasetPage".
23. Clique no ícone  e desenhe uma tabela de dados (*DataGrid*) no display. Configure-a conforme indicado a seguir.



24. A página do conjunto de dados (*DatasetPage*) terá a aparência indicada a seguir.



25. Configure nos botões a ação dinâmica com o valor de *Toogle* (Alternar) com o comando *Dataset* respectivo.
 - *Dataset.Table.TBEmployees.Select*
 - *Dataset.Table.TBEmployees.Next*
26. Vá para o display "Menu".
27. Acrescente a ação dinâmica ao botão *Dataset* e ajuste a ação para *OpenDisplay = "DatasetPage"*.

Exercício 8: Relatório

Objetivo

Criar um relatório no formato XPS e exibi-lo no display.

Procedimento

1. Crie uma nova Tag conforme mostrado a seguir.

Nome	Tipo
ReportFile	Text

2. Clique na opção *Edit*, seção *Reports*.
3. Realize a configuração ilustrada a seguir.

Name	Padding	SaveFormat	SaveFileName
Report1	Compact	XPS	{ReportFile}.xps

4. Clique na aba *TextEditor* e configure-a conforme indicado a seguir.

The screenshot shows the software interface with the following elements:

- Ribbon:** Font (Tahoma, 18), Paragraph, Paste (Image, Hyperlink, Line, Table, Tag, Symbol), and Table (Insert row below, Insert row above, Insert left column, Insert column right, Delete row, Delete column, Delete table).
- Main Area:** A 3D cube icon and the text "Report Daily".
- Code Snippets:**


```

Date = {Client.Date}
Time = {Client.Time}

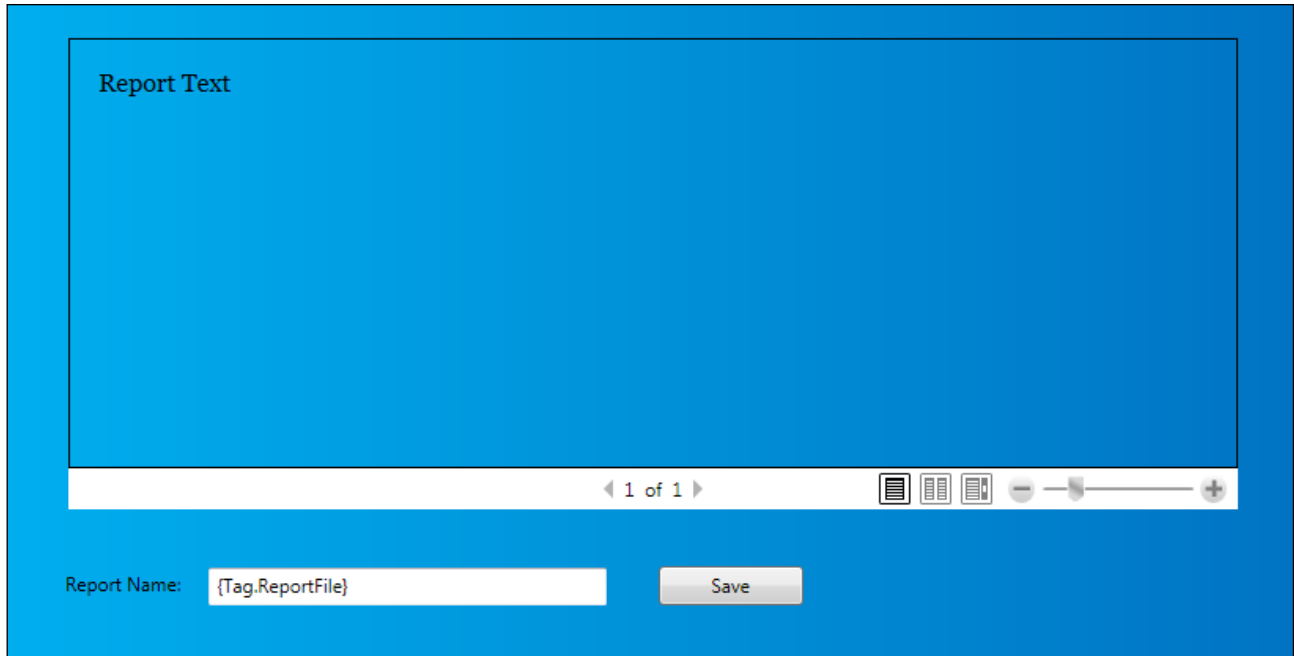
mosbus 400001 = {Tag.modbus400001}

mosbus 400002 = {Tag.modbus400002}

```

5. Clique na opção Draw e crie um novo display chamado "ReportPage".
6. Dê um clique com o botão direito do mouse no ícone  e selecione Report Viewer (Visualizador de Relatório). Desenhe-o no display.
7. Na janela ReportViewer, o relatório (Report) selecionado (...) deve ser Report1.

A página de relatório terá a aparência mostrada a seguir.




8. Configure o botão Save (Salvar) com: *Report.Report1.SaveCommand()*.
9. Vá para o display "Menu".
10. Acrescente a ação dinâmica ao botão Report e ajuste a ação para *OpenDisplay = "ReportPage"*.

Exercício 9: Compilar, Publicar e Rastrear

Objetivo

Compilar, executar, publicar e rastrear a aplicação.

Procedimento

1. Clique na opção *Run* (Executar), seção *Build* (Compilar).
2. Clique no botão *Build...* Aguarde o término da compilação.
3. Clique na seção *Publish* (Publicar).
4. Clique no botão *Publish*.
5. Verifique se o arquivo de projeto com a extensão “.teng” foi criado corretamente.
6. Clique na opção Info, seção *Track* (Rastrear).
7. Clique na aba *Changes* (Alterações).
8. Clique no ícone  na barra superior e selecione *Administrator with no password* (Administrador sem senha).
9. Clique na comboBox *Tracking* (Rastreamento) e escolha *Enabled* (Habilitado).
10. Crie uma nova Tag e retorne à aba *Changes* (na opção *Tracking*).
11. Verifique se o evento de criação de Tag foi registrado.

Exercício 10: Programação VB.NET e C#

Objetivo

Elaborar lógicas simples em VB.NET e C# como suporte ao desenvolvimento de aplicações no BluePlant. A lógica deve retornar o volume [m³] de material a partir da medição da massa [kg] via balança colocada na esteira transportadora. Um fator de correlação (densidade) deve ser especificado para converter de massa para volume.

📌*Dica: Densidade = Massa/Volume.

Procedimento

1- Resolução em VB.NET

```
Private Sub Correlacao_Click()
```

```
    Dim A ,B , C as Integer
```

```
    Massa = InputBox("Informe a massa de material: ", "Corr", "")
```

```
    Densidade = InputBox("Informe a densidade do material: ", "Corr", "")
```

```
    Volume = Massa / Densidade
```

```
    MsgBox Volume, vbInformation + vbOKOnly, "Corr"
```

```
End Sub
```

2- Resolução em C#

```
using System;
```

```
class Corr
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        int Massa;
```

```
        int Densidade;
```

```
        int Volume;
```

```
        Console.WriteLine("Informe a massa de material: ");
```

```
        Massa = int.Parse(Console.ReadLine());
```

```
        Console.WriteLine("Informe a densidade do material: ");
```

```
        Densidade = int.Parse(Console.ReadLine());
```

```
        Volume=Massa/Densidade;
```

```
        Console.WriteLine("O Volume eh {0}", Volume);
```

```
        Console.Read();
```

```
    }
```

```
}
```

3- Complemente os trechos de programa anteriores de tal forma que a lógica retorne uma indicação de que o valor atual ultrapassou o valor desejado do volume representado pela variável SETPOINT.

Situações de Aprendizagem Complementares



Esta seção inclui situações de aprendizagem que visam explorar funcionalidades complementares do BluePlant.

DICA: consulte o instrutor para orientações adicionais referentes à execução destas atividades.

1. Texto Dinâmico em Botão

Incluir na aplicação um botão que altere o nome apresentado quando a Tag associada ao mesmo mudar. Ele deve exibir LIGADO quando a Tag estiver em TRUE e DESLIGADO quando a Tag estiver em FALSE.

2. Comando e Monitoração via Pop-up

Incluir a funcionalidade de pop-up visando o comando e monitoração dos motores da aplicação. A ideia básica consiste em selecionar o motor desejado, por exemplo, MOTOR1 e visualizar, via janela de pop-up, seus atributos tais como, FUNCIONANDO, FALHA etc. Além disso, deve ser possível comandar o motor através dessa mesma janela (LIGAR e DESLIGAR).

3. Utilização de uma biblioteca WPF externa

Prever, na aplicação, o uso de uma biblioteca WPF externa - cuja fonte será indicada pelo Instrutor - visando demonstrar os recursos gráficos avançados da ferramenta em relação a objetos vetorizados (alta resolução).

ANOTAÇÕES

ANOTAÇÕES

Estudo de Caso Livre



Selecione uma aplicação da sua empresa, instituição ou de livre escolha para desenvolver o planejamento e o projeto do sistema de supervisão empregando o software BluePlant. Atente para as etapas indicadas abaixo na condução da atividade.

📌 **DICA:** consulte o instrutor para orientações adicionais referentes à execução desta atividade.

Edição da aplicação

- 1- Tags
- 2- Segurança e Alarmes
- 3- Dispositivos, Conjunto de dados e Scripts
- 4- Displays e Relatórios

Desenho da aplicação

- 5- Objetos estáticos
- 6- Objetos dinâmicos
- 7- Símbolos e Cores

Execução da Aplicação

- 8- Compilação, teste e inicialização
- 9- Funcionalidades em tempo de execução
- 10- Informações de projeto

Informações da Aplicação

- 11- Informações de projeto
- 12- Informações de versão
- 13- Objetos em tempo de execução

ANOTAÇÕES

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!!!

Apêndice

Guia de Referência Rápida Comparativa de Comandos VB.NET X C#

VB.NET	C#
Comentários	
<pre>' linha única Rem somente uma linha</pre>	<pre>// somente uma linha /* Multiplas linhas */ /// comentário XML em um linha /** comentário XML em múltiplas linhas */</pre>
Tipos de dados	
<p><i>Tipos por Valor</i></p> <p>Boolean Byte Char (exemplo: "A") Short, Integer, Long Single, Double Decimal Date</p> <p><i>Tipos por Referência</i></p> <p>Object String</p> <p>Dim x As Integer Console.WriteLine(x.GetType()) ' Imprime System.Int32 Console.WriteLine(TypeName(x)) ' Imprime Integer</p> <p><i>Conversão de Tipos</i></p> <p>Dim numDecimal As Single = 3.5 Dim numInt As Integer numInt = CType(numDecimal, Integer) ' Define para 4 numInt = CInt(numDecimal) ' mesmo resultado que com CType numInt = Int(numDecimal) ' Define para 3 (A função Int trunca o decimal)</p>	<p><i>Tipos por Valor</i></p> <p>bool byte, sbyte char (exemplo: 'A') short, ushort, int, uint, long, ulong float, double decimal DateTime</p> <p><i>Tipos por Referência</i></p> <p>object string</p> <p>int x; Console.WriteLine(x.GetType()); // Imprime System.Int32 Console.WriteLine(typeof(int)); // Imprime System.Int32</p> <p><i>Conversão de Tipos</i></p> <p>double numDecimal = 3.5; int numInt = (int) numDecimal; // define para 3 (trunca o decimal)</p>
Constantes	
Const MAX_STUDENTS As Integer = 25	const int MAX_STUDENTS = 25;
Enumerations	
<p>Enum Action Start [Stop] ' Stop é uma palavra reservada Rewind Forward End Enum</p> <p>Enum Status Flunk = 50 Pass = 70 Excel = 90 End Enum</p>	<pre>enum Action {Start, Stop, Rewind, Forward}; enum Status {Flunk = 50, Pass = 70, Excel = 90}; Action a = Action.Stop; if (a != Action.Start) Console.WriteLine(a + " e " + (int) a); // Imprime "Stop e 1" Console.WriteLine(Status.Pass); // Imprime Pass</pre>

<pre>Dim a As Action = Action.Stop If a <> Action.Start Then Console.WriteLine(a) ' Imprime 1 Console.WriteLine(Status.Pass) ' Imprime 70 Console.WriteLine(Status.Pass.ToString()) ' Imprime Pass</pre>	
Operadores	
<p><i>Comparaçã</i> = < > <= >= <></p> <p><i>Aritimético</i> + - * / Mod \ (divisão inteira) ^ (eleva a potência)</p> <p><i>Atribuição</i> = += -= *= /= \= ^= << >>= &=</p> <p><i>Bitwise</i> And AndAlso Or OrElse Not << >></p> <p><i>Logico</i> And AndAlso Or OrElse Not</p> <p>Note: AndAlso and OrElse sao avaliações lógica do tipo curto-circuito</p> <p><i>Concatenação de strings</i> &</p>	<p><i>Comparaçã</i> == < > <= >= !=</p> <p><i>Aritimético</i> + - * / % (mod) / (divisão inteira se ambos os operadores são ints) Math.Pow(x, y) (eleva a potência)</p> <p><i>Atribuição</i> = += -= *= /= %= &= = ^= << >>= ++ --</p> <p><i>Bitwise</i> & ^ ~ << >></p> <p><i>Logico</i> && !</p> <p>Note: && and sao avaliações lógica do tipo curto-circuito</p> <p><i>Concatenação de strings</i> +</p>
If	
<pre>saudacao= If(idade < 20, "Tudo bem ?", "Ola") ' Uma linha não requer "End If", nem "Else" If language = "VB.NET" Then langType = "texto" ' Usar : para por dois comandos na mesma linha If x <> 100 And y < 5 Then x *= 5 : y *= 2 ' Preferido If x <> 100 And y < 5 Then x *= 5 y *= 2 End If ' para quebrar linhas longas use _ (underscore) If teste1 < teste2 And teste3 > 100 Then _ outra linha 'If x > 5 Then x *= y Elseif x = 5 Then x += y Elseif x < 10 Then x -= y Else x /= y End If Select Case color ' Deve ser um tipo primitivo Case "pink", "red" r += 1 Case "blue" b += 1 Case "green" g += 1 Case Else other += 1 End Select</pre>	<pre>saudacao= idade < 20 ? "Tudo bem ?" : "Ola"; if (x != 100 && y < 5) { // múltiplos comandos precisam estar entre } x *= 5; y *= 2; } Não precisa de _ ou : desde que ; é usado para terminar cada comando if (x > 5) x *= y; else if (x == 5) x += y; else if (x < 10) x -= y; else x /= y; switch (color) { // precisa ser um inteiro ou string case "pink": case "red": r++; break; // break é obrigatorio case "blue": b++; break; case "green": g++; break; default: other++; break; }</pre>

Loops	
<p><i>Loops com Pré teste</i></p> <pre>While c < 10 c += 1 End While Do While c < 10 c += 1 Loop <i>loop com teste posterior</i> Do c += 1 Loop While c < 10 ' loopin em Array ou collection Dim nomes As String() = {"Janice", "Miriam", "Macoratti"} For Each s As String In nomes Console.WriteLine(s) Next</pre>	<p><i>Loops com pré teste</i></p> <pre>// não tem a palavra chave until while (i < 10) i++; for (i = 2; i <= 10; i += 2) Console.WriteLine(i); <i>loop com teste posterior</i> do i++; while (i < 10); ' loopin em Array ou collection string[] nomes = {"Janice", "Miriam", "Macoratti"}; foreach (string s in nomes) Console.WriteLine(s);</pre>
Arrays	
<pre>Dim nums() As Integer = {1, 2, 3} For i As Integer = 0 To nums.Length - 1 Console.WriteLine(nums(i)) Next ' 4 é o índice do último elemento, então ele trata 5 elementos Dim names(4) As String names(0) = "Macoratti" names(5) = "Miriam" ' Throws System.IndexOutOfRangeException ' Redimensiona um array mantendo os valores existentes ReDim Preserve nomes(6) Dim DuasD(rows-1, cols-1) As Single DuasD(2, 0) = 4.5 Dim jagged()() As Integer = { New Integer(4) {}, New Integer(1) {}, New Integer(2) {} } jagged(0)(4) = 5</pre>	<pre>int[] nums = {1, 2, 3}; for (int i = 0; i < nums.Length; i++) Console.WriteLine(nums[i]); // 5 é o tamanho do array string[] names = new string[5]; names[0] = "Macoratti"; names[5] = "Miriam"; // Throws System.IndexOutOfRangeException // C# não pode redimensionar um array dinamicamente. Copie em um novo array string[] nomes2 = new string[7]; Array.Copy(nomes, nomes2, nomes.Length); // ou names.CopyTo(nomes2, 0); float[,] DuasD = new float[rows, cols]; DuasD[2,0] = 4.5f; int[,] jagged = new int[3][] { new int[5], new int[2], new int[3] }; jagged[0][4] = 5;</pre>
Funções	
<pre>' Passa por valor (é o padrão) Sub TestFunc(ByVal x As Integer, ByRef y As Integer, ByRef z As Integer) x += 1 y += 1 z = 5 End Sub Dim a = 1, b = 1, c As Integer ' TestaFunc(a, b, c) Console.WriteLine("{0} {1} {2}", a, b, c) ' 1 2 5</pre>	<pre>// ' Passa por valor (é o padrão) void TestFunc(int x, ref int y, out int z) { x++; y++; z = 5; } int a = 1, b = 1, c; // c não precisa inicialização TestaFunc(a, ref b, out c); Console.WriteLine("{0} {1} {2}", a, b, c); // 1 2 5</pre>

<pre>' Aceita número de argumentos variáveis Function Sum(ByVal ParamArray nums As Integer()) As Integer Sum = 0 For Each i As Integer In nums Sum += i Next End Function Dim total As Integer = Sum(4, 3, 2, 1) ' retorna 10 Sub DizOla(ByVal nome As String, Optional ByVal prefix As String = "") Console.WriteLine(Saudacoes, " & prefix & " & nome) End Sub DizOla("Strangelove", "Dr.") DizOla("Macoratti")</pre>	<pre>' Aceita número de argumentos variáveis int Sum(params int[] nums) { int sum = 0; foreach (int i in nums) sum += i; return sum; } int total = Sum(4, 3, 2, 1); // retorna 10 void DizOla(string nome, string prefix) { Console.WriteLine("Saudacoes, " + prefix + " " + nome); } void DizOla(string nome) { DizOla(nome, ""); }</pre>
Tratamento de exceção	
<pre>' Tratamento de erro desatualizado On Error GoTo MyErrorHandler ... MyErrorHandler: Console.WriteLine(Err.Description) Dim ex As New Exception("Alguma coisa esta errada") Throw ex Try y = 0 x = 10 / y Catch ex As Exception When y = 0 ' Argumento e When é opcional Console.WriteLine(ex.Message) Finally Beep() End Try</pre>	<pre>Exception up = new Exception("Alguma coisa esta errada."); throw up; try { y = 0; x = 10 / y; } catch (Exception ex) { // Argumento é opcional Console.WriteLine(ex.Message); } finally { }</pre>
Namespaces	
<pre>Namespace Harding.Compsci.Graphics ... End Namespace ' or Namespace Harding Namespace Compsci Namespace Graphics ... End Namespace End Namespace End Namespace Imports Harding.Compsci.Graphics</pre>	<pre>namespace Harding.Compsci.Graphics { ... } // or namespace Harding { namespace Compsci { namespace Graphics { ... } } } using Harding.Compsci.Graphics;</pre>
Classes / Interfaces	
<pre>' Modificadores de acesso Public Private Friend Protected Protected Friend Shared ' herança Class FootballGame Inherits Competition</pre>	<pre>' Modificadores de acesso public private internal protected protected internal static ' herança class FootballGame : Competition { ... }</pre>

<pre> ... End Class ' Definição de interface Interface IAlarmClock ... End Interface // Extendendo uma interface Interface IAlarmClock Inherits IClock ... End Interface // Implementando uma interface Class WristWatch Implements IAlarmClock, ITimer ... End Class </pre>	<pre> ''' Definição de interface interface IAlarmClock { ... } // Extendendo uma interface interface IAlarmClock : IClock { ... } // Implementando uma interface class WristWatch : IAlarmClock, ITimer { ... } </pre>
Construtores	
<pre> Class SuperHeroii Private _nivelPoder As Integer Public Sub New () _nivelPoder = 0 End Sub Public Sub New (ByVal nivelPoder As Integer) Me._nivelPoder = nivelPoder End Sub Protected Overrides Sub Finalize () MyBase.Finalize() End Sub End Class </pre>	<pre> class SuperHeroi { private int _nivelPoder; public SuperHeroi() { _nivelPoder = 0; } public SuperHero8(int nivelPoder) { this._nivelPoder= nivelPoder; } ~SuperHeroi() { } } </pre>
Objetos	
<pre> Dimheroi As SuperHeroi = New SuperHeroi Withheroi .Nome = "HomedeFerro" .nivelPoder = 3 End With heroi.Defend("Miriam") heroi.Rest() ' Chamando um método Shared ' ou SuperHeroi.Rest() Dim heroi2 As SuperHeroi =heroi ' ambos se referem ao mesmo objeto hero2.Nome = "MulherGato" Console.WriteLine(heroi.Nome) ' Imprime MulherGato hero = Nothing ' Libera o objeto If heroi Is Nothing Then _ heroi = New SuperHeroi Dim obj As Object = New SuperHeroi If TypeOf obj Is SuperHeroi Then _ Console.WriteLine("è um objeto SuperHeroi .") </pre>	<pre> SuperHeroi heroi = new SuperHeroi(); hero.Nome = "HomedeFerro"; hero.nivelPoder = 3; hero.Defend("Miriam"); SuperHeroi.Rest(); // Chamando um método estático SuperHeroiheroi2 =heroi; // ambos se referem ao mesmo objeto hero2.Name = "MulherGato"; Console.WriteLine(hero.Nome); // Imprime MulherGato hero = null ; // libera o objeto if (heroi == null) heroi = new SuperHeroi(); Object obj = new SuperHeroi(); if (obj is SuperHeroi) Console.WriteLine("è um objeto SuperHeroi"); </pre>
Structs	
<pre> Structure Aluno Public nome As String Public gpa As Single </pre>	<pre> struct Aluno { public string nome; public float gpa; } </pre>

<pre>Public Sub New(ByVal nome As String, ByVal gpa As Single) Me.nome = nome Me.gpa = gpa End Sub End Structure Dim stu As Aluno = New Aluno("Jefferson", 3.5) Dim stu2 As Aluno = stu stu2.nome = "Jessica" Console.WriteLine(stu.nome) ' <i>Imprime Jefferson</i> Console.WriteLine(stu2.nome) ' <i>Imprime Jessica</i></pre>	<pre>public Aluno(string nome, float gpa) { this.nome = nome; this.gpa = gpa; } } Aluno stu = new Aluno("Jefferson", 3.5f); Aluno stu2 = stu; stu2.nome = "Jessica"; Console.WriteLine(stu.nome); // <i>Imprime Jefferson</i> Console.WriteLine(stu2.nome); // <i>Imprime Jessica</i></pre>
Propriedades	
<pre>Private _size As Integer Public Property Size() As Integer Get Return _size End Get Set (ByVal Value As Integer) If Value < 0 Then _size = 0 Else _size = Value End If End Set End Property foo.Size += 1</pre>	<pre>private int _size; public int Size { get { return _size; } set { if (value < 0) _size = 0; else _size = value; } } foo.Size++;</pre>
Console I/O	
<pre><i>Caracteres especiais</i> vbCrLf, vbCr, vbLf, vbNewLine vbNullString vbTab vbBack vbFormFeed vbVerticalTab "" Chr(65) ' <i>Retorna 'A'</i> Console.Write("Qual o seu nome ") Dim name As String = Console.ReadLine() Console.Write("Quantos anos voce tem? ") Dim idade As Integer = Val(Console.ReadLine()) Console.WriteLine("{0} tem {1} anos.", name, idade) ' ou Console.WriteLine(nome & " tem " & idade & " anos") Dim c As Integer c = Console.Read() ' <i>Le um único caractere</i> Console.WriteLine(c) ' <i>Imprime 65 se usuário entrar "A"</i></pre>	<pre><i>Sequencia de escape</i> \n, \r \t \\ \" Convert.ToChar(65) // <i>Retorna 'A' -</i> // ou (char) 65 Console.Write("Qual o seu nome "); string name = Console.ReadLine(); Console.Write("Quantos anos voce tem? "); int idade = Convert.ToInt32(Console.ReadLine()); Console.WriteLine("{0} tem {1} anos.", name, idade); // ou Console.WriteLine(nome + " tem " + idade + " anos."); int c = Console.Read(); // <i>Le um único caractere</i> Console.WriteLine(c); // <i>Imprime 65 se usuário entrar "A"</i></pre>
File I/O	
<pre>Imports System.IO Dim writer As StreamWriter = File.CreateText("c:\arquivo.txt") writer.WriteLine("Teste.") writer.Close() Dim reader As StreamReader = File.OpenText("c:\arquivo.txt") Dim line As String = reader.ReadLine() While Not line Is Nothing Console.WriteLine(line) line = reader.ReadLine()</pre>	<pre>using System.IO; StreamWriter writer = File.CreateText("c:\arquivo.txt"); writer.WriteLine("Teste."); writer.Close(); StreamReader reader = File.OpenText("c:\arquivo.txt"); string line = reader.ReadLine(); while (line != null) { Console.WriteLine(line); line = reader.ReadLine(); }</pre>

<pre>End While reader.Close() Dim str As String = "Texto data" Dim num As Integer = 123 Dim binWriter As New BinaryWriter (File.OpenWrite("c:\arquivo.dat")) binWriter.Write(str) binWriter.Write(num) binWriter.Close() Dim binReader As New BinaryReader (File.OpenRead("c:\arquivo.dat")) str = binReader.ReadString() num = binReader.ReadInt32() binReader.Close()</pre>	<pre>} reader.Close(); string str = "Texto data"; int num = 123; BinaryWriter binWriter = new BinaryWriter(File.OpenWrite("c:\arquivo.dat")); binWriter.Write(str); binWriter.Write(num); binWriter.Close(); BinaryReader binReader = new BinaryReader(File.OpenRead("c:\arquivo.dat")); str = binReader.ReadString(); num = binReader.ReadInt32(); binReader.Close();</pre>
---	--

Fonte: <http://www.macoratti.net/vbnxcshp.htm>

Glossário

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

Nó	Qualquer estação de uma rede com capacidade de comunicação utilizando um protocolo estabelecido.
Operandos	Elementos sobre os quais as instruções atuam. Podem representar constantes, variáveis ou um conjunto de variáveis.
PC	Sigla para programmable controller. É a abreviatura de controlador programável em inglês.
PDO	Do inglês, é uma mensagem do protocolo CAN contendo os dados operacionais.
Ping	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.
POU	, ou Unidade de Organização de Programa, é uma subdivisão do programa aplicativo que pode ser escrito em qualquer uma das linguagens disponíveis.
Programa aplicativo	É o programa carregado em um CP, que determina o funcionamento de uma máquina ou processo.
Protocolo	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.
RAM	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.
Rede de comunicação	Conjunto de equipamentos (nós) interconectados por canais de comunicação.
Reset	Comando para reinicializar o CP.
RUN	Comando para colocar o CP em modo de execução.
Set	Ação para atribuir o estado de nível lógico alto para uma variável booleana.
Software	Programas de computador, procedimentos e regras relacionadas à operação de um sistema de processamento de dados.
STOP	Comando para congelar o CP em seu estado atual.
Sub-rede	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.
Time-out	Tempo preestabelecido máximo para que uma comunicação seja completada. Se for excedido procedimentos de retentiva ou diagnóstico serão ativados.
Tooltip	Caixa de texto com uma ajuda ou local onde pode-se entrar com a ajuda.
UCP	Sigla para unidade central de processamento. Controla o fluxo de informações, interpreta e executa as instruções do programa e monitora os dispositivos do sistema.
UCP ativa	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.
Upload	Leitura do programa ou configuração do CP.
Visualização	Conjunto de telas do CP.
XML	Do inglês, Extensible Markup Language, é um padrão para gerar linguagens de marcação.
Zoom	No contexto da janela de função do teclado, é utilizado para a troca de telas.