



# **Manual de Utilização**

## **Série BluePlant**

MU224000 Rev. G

Dezembro de 2025

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 suas partes e peças 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 descrito 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.

## DIREITOS AUTORAIS

Nexto, Série Ponto, MasterTool, Grano, WebPLC e BluePlant são marcas registradas da Altus Sistemas de Automação S.A.

*Windows, Windows NT e Windows Vista* são marcas registradas da Microsoft Corporation.

# Sumário

<b>1. SUPORTE TÉCNICO BLUEPLANT .....</b>	<b>1</b>
Introdução ao BluePlant .....	1
Documentos Relacionados a este Manual .....	1
Inspeção Visual .....	2
Suporte Técnico .....	2
Mensagens de Advertência Utilizadas neste Manual .....	2
<b>2. SÉRIE BLUEPLANT .....</b>	<b>3</b>
Características do BluePlant .....	3
Modelos do BluePlant .....	3
Produtos Relacionados .....	4
SoftKey .....	4
HardKey .....	5
Acessórios .....	6
<b>3. DESCRIÇÃO TÉCNICA DO BLUEPLANT .....</b>	<b>7</b>
Características Gerais dos Modelos BluePlant .....	7
Características Gerais Comuns .....	8
Características Gerais .....	8
Software com Segurança Intrínseca .....	8
Mecanismo Gráfico Superior .....	9
Recursos de Manutenção, Teste e Diagnóstico Avançado .....	9
Servidores e Extensões .NET Incorporados .....	9
Características Inovadoras do Produto .....	9
Banco de Dados em Tempo Real (Tags) .....	9
Linguagens .NET e Scripts .....	11
Alarmes e Segurança .....	11
Tendência e Historiadores .....	13
Dispositivos e Redes .....	13
Conjunto de Dados .....	14
Relatórios .....	15
Telas do Cliente .....	15

---

Objetos do Runtime .....	16
Isolação de Módulo.....	16
Ferramentas do Runtime e Diagnóstico .....	16
Implantação e Teste de Projeto.....	17
Drivers de Comunicação .....	17
<b>4. BLUEPLANT.....</b>	<b>18</b>
<b>Requisitos Mínimos para Instalação e Operação.....</b>	<b>18</b>
<b>Instalação .....</b>	<b>19</b>
<b>Licenças e Hardkey.....</b>	<b>26</b>
<b>Iniciando com o BluePlant.....</b>	<b>27</b>
Gerenciamento de Projeto .....	27
<b>Primeiros Passos com o BluePlant.....</b>	<b>28</b>
Selecionando o Projeto Demo .....	29
Criando um Novo Projeto .....	29
Interface do Usuário.....	29
Criando Tags no Projeto.....	30
Associando tags a um protocolo comunicação.....	32
Incluindo objetos na tela principal .....	34
Executando um Projeto .....	37
<b>5. MENU PRINCIPAL DO BLUEPLANT .....</b>	<b>42</b>
<b>Edição da Aplicação.....</b>	<b>42</b>
<b>Diagramação da Aplicação .....</b>	<b>43</b>
<b>Execução da Aplicação .....</b>	<b>43</b>
<b>Informações da Aplicação .....</b>	<b>45</b>
<b>6. COMPONENTES DO BLUEPLANT .....</b>	<b>46</b>
<b>Menu Edit .....</b>	<b>46</b>
Editando Tags (Tags).....	46
Editando as Configurações de Segurança (Security) .....	58
Editando Dispositivos (Devices) .....	62
Editando Alarmes (Alarms).....	72
Editando Banco de Dados (Datasets).....	77
Editando Scripts (Scripts).....	96

---

---

Editando Displays.....	104
Editando Relatórios (Reports) .....	112
<b>Menu Draw .....</b>	<b>115</b>
Ferramentas de Seleção.....	117
Objetos Geométricos.....	118
Componentes de Tela.....	118
Entrada e Saída de Texto.....	119
Biblioteca de Símbolos .....	120
Controles Avançados .....	120
Alarme.....	126
Tendência (Legacy) .....	128
Tendência .....	134
Grade de Dados .....	136
Barra de Ferramentas Horizontal.....	138
Configuração Dinâmica .....	139
CodeBehind.....	147
Símbolos Locais .....	148
<b>Menu Run .....</b>	<b>152</b>
Build .....	152
Test .....	153
Publish .....	156
UseCount.....	157
Dictionaries .....	159
Extensions .....	163
<b>Menu Info.....</b>	<b>165</b>
Project .....	165
Track .....	169
Notes .....	172
<b>Ferramentas da Aplicação .....</b>	<b>173</b>
Tstartup .....	173
PropertyWatch.....	174
TraceWindow .....	175
ModuleInformation.....	178

---

---

<b>Objetos Runtime .....</b>	<b>178</b>
Namespace Tag .....	180
Namespace Security.....	188
Namespace Alarm.....	192
Namespace Device.....	197
Namespace Dataset.....	200
Namespace Script .....	210
Namespace Display.....	212
Namespace Report .....	212
Namespace Info .....	214
Namespace Server .....	220
Namespace Client .....	222
<b>Configurações Avançadas.....</b>	<b>227</b>
Linhas de Comando .....	227
Executando o BluePlant como um Serviço do Windows.....	231
TWebServer .....	232
Clientes Remotos.....	232
Instalando o Web Server no IIS .....	233
<b>7. CENÁRIOS DE SISTEMAS TÍPICOS.....</b>	<b>239</b>
<b>Configurações dos Sistemas .....</b>	<b>239</b>
Sistema Standalone .....	239
Sistema de Entradas e Saídas Distribuídas .....	239
Sistema Cliente e Servidor .....	240
Sistema de Servidor Redundante .....	241
Sistema de Controle .....	246
Sistema de Controle Distribuído e Distribuido Redundante .....	247
Sistema de Compartilhamento de Carga .....	248
<b>8. GLOSSÁRIO .....</b>	<b>249</b>

# 1. Suporte Técnico BluePlant

## Introdução ao BluePlant

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

O BluePlant atende requisitos de alto desempenho, recursos de conectividade ampliados, interface gráfica de usuário extremamente rica e poderosa e mecanismos avançados de aquisição de dados em tempo real. A seleção de drivers incorporados no BluePlant, a capacidade de engenharia distribuída, a redundância, o suporte OPC garantem uma experiência nova e única com o BluePlant. Criado 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, executando em computadores, Web, tablets e smartphones.



**Figura 1-1. Altus BluePlant**

## Documentos Relacionados a este Manual

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](http://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

## **Inspeção Visual**

Antes de proceder à instalação, é recomendável fazer uma inspeção visual cuidadosa dos equipamentos, verificando se não há danos causados pelo transporte. Verifique se todos os componentes de seu pedido estão em perfeito estado. Em caso de defeitos, informe a companhia transportadora e o representante ou distribuidor Altus mais próximo.

É importante registrar o número de série de cada equipamento recebido, bem como as revisões de software, caso existentes. Essas informações serão necessárias caso se necessite contatar o Suporte Técnico da Altus.

## **Suporte Técnico**

Para entrar em contato com o Suporte Técnico da Altus em São Leopoldo, RS, ligue para +55 51 3589-9500. Para conhecer os centros de Suporte Técnico da Altus existentes em outras localidades, consulte nosso site ([www.altus.com.br](http://www.altus.com.br)) ou envie um e-mail para [altus@altus.com.br](mailto:altus@altus.com.br).

Se o equipamento já estiver instalado, tenha em mãos as seguintes informações ao solicitar assistência:

- A configuração do sistema instalado (sistema operacional, banco de dados, arquitetura utilizada e atendimento dos requisitos de hardware - computador)
- O modelo de BluePlant utilizado e acessórios adicionais
- O conteúdo do projeto (módulos utilizados), obtido através do programador BluePlant

## **Mensagens de Advertência Utilizadas neste Manual**

Neste manual, as mensagens de advertência apresentarão os seguintes formatos e significados:

### **PERIGO:**

**Relatam causas potenciais, que se não observadas, levam a danos à integridade física e saúde, patrimônio, meio ambiente e perda da produção.**

### **CUIDADO:**

**Relatam detalhes de configuração, aplicação e instalação que devem ser seguidos para evitar condições que possam levar a falha do sistema e suas consequências relacionadas.**

### **ATENÇÃO:**

**Indicam detalhes importantes de configuração, aplicação ou instalação para obtenção da máxima performance operacional do sistema.**

## 2. Série BluePlant

### Características do BluePlant

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	
<b>BluePlant Enterprise</b>	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
<b>BluePlant Lite</b>	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
<b>BluePlant Express</b>	Projetado somente para avaliação, e não para instalação em campo, limitada em 75 pontos de comunicação e uma hora de execução do Runtime
<b>BluePlant Student</b>	Projetado para escolas e universidades. O tamanho da aplicação varia de 150 até 1500 pontos de comunicação e uma hora de execução do Runtime

**Tabela 2-1. 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.

### Modelos do BluePlant

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. Os modelos disponíveis seguem na Tabela 2-1 e na Tabela 2-2 estão as quantidades de pontos de comunicação para cada modelo de BluePlant.

BluePlant Enterprise (Pontos de comunicação)	BluePlant Lite (Pontos de comunicação)	BluePlant Express (Pontos de comunicação)	BluePlant Student (Pontos de comunicação)
-	-	75	-
150	150	-	-
300	300	-	-
500	500	-	-
1500	1500	-	1500
2500	-	-	-
5000	-	-	-
15000	-	-	-
25000	-	-	-
50000	-	-	-
100000	-	-	-
Ultimate	-	-	-

**Tabela 2-2. Quantidade de Pontos de Comunicação por Modelo de BluePlant**

## Produtos Relacionados

### SoftKey

A tabela a seguir indica os códigos que devem ser utilizados para adquirir o produto:

	BluePlant Enterprise		BluePlant Lite		BluePlant Express	BluePlant Student
Pontos de comunicação	Engineering Runtime	Runtime	Engineering Runtime	Runtime	Engineering Runtime	Engineering Runtime
75	-	-	-	-	BP6400	-
150	BP1403	BP1303	BP2403	BP2303	-	-
300	BP1405	BP1305	BP2405	BP2305	-	-
500	BP1407	BP1307	BP2407	BP2307	-	-
1.500	BP1409	BP1309	BP2409	BP2309	-	BP4400
2.500	BP1411	BP1311	-	-	-	-
5.000	BP1413	BP1313	-	-	-	-
15.000	BP1415	BP1315	-	-	-	-
Ultimate	BP1499	BP1399	-	-	-	-

**Tabela 2-3. Produtos Relacionados por Modelo de BluePlant de SoftKey**

### Notas:

**Ultimate:** Aplicações com mais de 15.000 pontos de comunicação devem utilizar essa licença.

**BluePlant Enterprise:** Esse modelo acompanha três clientes para o BluePlant, demais acessórios devem ser requisitados separadamente.

**BluePlant Lite:** Esse modelo acompanha três clientes para o BluePlant, não sendo possível adicionar mais acessórios.

**Engineering:** Utilizado para edição do projeto e necessário para integradores que não possuem SoftKey de Integrador BP5021.

**Runtime:** Ambiente independente onde pode ser executado sem a utilização de licença de Engineering porém sem a possibilidade de realizar alterações.

### Soluções para Integrador de Sistemas

A tabela abaixo apresenta os códigos de produtos que atendem as necessidades de integradores de sistemas.

Código	Descrição
BP5021	Solução para integradores de sistemas – BluePlant Lite/Enterprise SoftKey

**Tabela 2-4. Soluções para Integradores de Sistemas com SoftKey**

#### Nota:

**BP5021** Solução que permite desenvolver projetos para o BluePlant Lite e BluePlant Enterprise, sem limite de validade da licença. Permite a execução do Runtime por 72 horas.

### HardKey (Produto descontinuado)

O modelo de Hardkey não é mais comercializado pela Altus sendo sua venda realizada apenas em casos especiais sob consulta. A tabela a seguir indica os códigos de produtos comercializado pela Altus no modelo de licenciamento por Hardkey:

	BluePlant Enterprise		BluePlant Lite		BluePlant Express	BluePlant Student
Pontos de comunicação	Engineering Runtime	Runtime	Engineering Runtime	Runtime	Engineering Runtime	Engineering Runtime
75	-	-	-	-	BP6400	-
150	BP1203	BP1103	BP2203	BP2103	-	-
300	BP1205	BP1105	BP2205	BP2105	-	-
500	BP1207	BP1107	BP2207	BP2107	-	-
1.500	BP1209	BP1109	BP2209	BP2109	-	BP4400
2.500	BP1211	BP1111	-	-	-	-
5.000	BP1213	BP1113	-	-	-	-
15.000	BP1215	BP1115	-	-	-	-
25.000	BP1217	BP1117	-	-	-	-
50.000	BP1219	BP1119	-	-	-	-
100.000	BP1221	BP1121	-	-	-	-
Ultimate	BP1299	BP1199	-	-	-	-

**Tabela 2-5. Produtos Relacionados por Modelo de BluePlant de Hardkey**

#### Notas:

**Ultimate:** Aplicações com mais de 100.000 tags devem utilizar essa licença.

**BluePlant Enterprise:** Esse modelo acompanha três clientes para o BluePlant, demais acessórios devem ser requisitados separadamente.

**BluePlant Lite:** Esse modelo acompanha três clientes para o BluePlant.

#### Atenção:

- Cada endereço IP conta como um acesso de cliente. Exemplo: Se a aplicação foi desenvolvida para usar 2 monitores, e ficam no mesmo computador (mesmo endereço IP) será necessário apenas 1 licença de cliente.
- Para redundância, será necessário adquirir 2 licenças com modelos iguais/equivalentes e acessórios iguais.

### *Soluções para Integrador de Sistemas*

A tabela abaixo apresenta os códigos de produtos que atendem as necessidades de integradores de sistemas.

Código	Descrição
<b>BP5001</b>	Solução para integradores de sistemas com licenças temporárias – 1 ano (produto descontinuado)
<b>BP5003</b>	Solução para integradores de sistemas com licenças temporárias – 3 anos (produto descontinuado)
<b>BP5010</b>	Solução para integradores de sistemas – BluePlant Lite (produto descontinuado)
<b>BP5020</b>	Solução para integradores de sistemas – BluePlant Lite/Enterprise

**Tabela 2-6. Soluções para Integradores de Sistemas com HardKey**

#### **Notas:**

**BP5001, BP5003 e BP5010:** Produtos descontinuados.

**BP5020:** Solução que permite desenvolver projetos para o BluePlant Lite e BluePlant Enterprise, sem limite de validade da licença. Permite a execução do Runtime por 72 horas.

### **Acessórios**

A tabela abaixo contém os códigos dos acessórios que podem ser adquiridos. Estes acessórios somente estão disponíveis no modelo BluePlant Enterprise.

Código	Descrição
<b>BP9501</b>	Cliente web de visualização para Internet Explorer (produto descontinuado)
<b>BP9601</b>	Cliente BluePlant
<b>BP9699</b>	Usuário Engenharia
<b>BP9701</b>	Cliente web completo para Internet Explorer (produto descontinuado)
<b>BP9801</b>	Cliente iPad/iPhone

**Tabela 2-7. Acessórios para BluePlant Enterprise**

#### **Notas:**

**BP9501, BP9701 e BP9801:** Produtos descontinuados. A partir da versão de BluePlant 2018.1.1 os produtos BP9501 e BP9701 foram unificados na licença BP9601.

### 3. Descrição Técnica do BluePlant

Este capítulo apresenta as características técnicas dos modelos de BluePlant, abordando as partes integrantes do sistema, sua arquitetura e características gerais.

#### Características Gerais dos Modelos BluePlant

	BluePlant Lite	BluePlant Student	BluePlant Express	BluePlant Enterprise
<b>Limite de execução do Runtime</b>	Não	Sim	Sim	Não
<b>Servidor OPC DA</b>	Sim	Sim	Não	Sim
<b>Linguagem C#</b>	Não	Sim	Sim	Sim
<b>Execução multi-threading de scripts</b>	Não	Não	Não	Sim
<b>Matriz de tags (múltiplas dimensões)</b>	Não	Não	Não	Sim
<b>Tipos do usuário (múltiplos níveis)</b>	Não	Não	Não	Sim
<b>Extensão SDK e integração de toolkits</b>	Não	Não	Não	Sim
<b>Configuração da tabela de histórico</b>	Sim	Não	Não	Sim
<b>Clientes rich remotos concorrentes</b>	Não	Sim	Sim	Sim
<b>Clientes web remotos concorrentes (completo e/ou visualização)</b>	Sim, 3 clientes	Sim	Sim	Sim
<b>Redundância de nós de dispositivos</b>	Não	Sim	Sim	Sim
<b>Redundância de servidores</b>	Não	Não	Não	Sim
<b>Relatório através de objetos gráficos</b>	Não	Sim	Sim	Sim
<b>Condições estendidas de alarmes</b>	Não	Sim	Sim	Sim
<b>Controle de versão de projeto</b>	Não	Sim	Sim	Sim
<b>Controle de alterações por objetos</b>	Não	Sim	Sim	Sim
<b>Compressão automática do histórico</b>	Não	Não	Não	Sim
<b>Controle de acesso WPF</b>	Não	Sim	Sim	Sim
<b>Hot start</b>	Não	Não	Não	Sim
<b>Modo teste</b>	Não	Sim	Sim	Sim

#### Notas:

**Limite de execução do Runtime:** Execução do Runtime limitada em uma hora. O Runtime pode ser reiniciado.

**Linguagem C#:** É possível criar scripts utilizando a linguagem C#.

**Execução multi-threading de scripts:** Essa funcionalidade permite criar scripts e gerar diferentes threads para cada script criado. Quando essa característica está habilitada, a execução das threads é concorrente. Quando essa característica não está habilitada, a execução das threads é sequencial.

**Tipos do usuário (múltiplos níveis):** É permitido criar novos tipos de dados, sendo possível utilizar até quatro níveis encadeados de tipos de dados.

**Extensão SDK e integração de toolkits:** É possível criar bibliotecas proprietárias (dlls), com funções específicas e utilizar em projetos.

**Clientes rich remotos concorrentes:** BluePlant Enterprise deve ser instalado na máquina remota e as licenças devem estar presentes na máquina servidora. O número de clientes rich remotos concorrentes depende do número de licenças adquiridas.

**Clientes web remotos concorrentes (completo e/ou visualização):** O número de clientes web remotos concorrentes tanto de visualização quanto o completo depende do número de licenças adquiridas, exceto para o BluePlant Lite que permite somente um cliente remoto completo e nenhum cliente remoto de visualização.

**Redundância de servidores:** Para utilizar essa funcionalidade são necessários dois servidores e duas licenças softkeys. Cada servidor deve ter a sua própria softkey e então é possível configurar os servidores como um par redundante.

**Controle de alterações por objetos:** Essa característica permite controlar as alterações feitas em telas, tags, scripts, módulos e outras modificações em objetos no projeto.

**Hot start:** É possível modificar a aplicação e recarregá-la sem parar o sistema.

## Características Gerais Comuns

	BluePlant Lite, BluePlant Express, BluePlant Student e BluePlant Enterprise
Protocolos simultâneos	Todos os modelos do BluePlant possuem pelo menos quatro canais
Cliente OPC	Sim
Abertura de múltiplos projetos	Sim
Linguagem VisualBasic .NET	Sim
Scripts para avaliações de expressões matemáticas.	Sim
Scripts para criar classes .NET e tarefas	Sim
Integração com base de dados SQL	Sim
Historiador e Logger	Sim
Alarme e Proteção	Sim
Editor Gráfico WPF	Sim
Ferramentas de engenharia e depuração	Sim
Editor de relatório	Sim
Controle de alterações por tabelas	Sim
Localization	Sim

### Notas:

**Protocolos simultâneos:** Execução simultânea de drivers de comunicação durante a execução do Runtime.

**Controle de alterações por tabelas:** O controle de alterações por tabelas informa o que foi modificado, removido ou inserido, mas não informa onde as alterações foram feitas.

**Localization:** Essa característica traduz textos de telas e alarmes no Runtime.

## Características Gerais

### 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.

### **Características Inovadoras do Produto**

#### **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.

Banco de dados em tempo real (Tags)	
<b>Supporte estendido a tipos de tags</b>	Digital, Analógico Inteiro, Analógico Double, Analógico Decimal, Texto, Temporizador, Contador, Date/Time (variáveis de data e hora)
<b>Tag do tipo tabela de dados incorporado</b>	Acesso aos resultados da consulta do conjunto de dados no objeto de tabela de dados .NET padrão
<b>Tags de referência</b>	Utilização de tags de referência para chavear o link do Tag na execução do Runtime
<b>Tag do tipo matriz</b>	Definição de tags do tipo matriz (uma a três dimensões, dependendo da versão do produto)
<b>Tipos e estruturas definidas pelo usuário</b>	Definição de tipos próprios para o banco de dados em tempo real
<b>Propriedades do Tag</b>	Amplo conjunto de propriedades de tag acessíveis na configuração do Runtime

**Tabela 3-1. Banco de Dados em Tempo Real (Tags)**

### 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.

Linguagens .NET e Scripts	
<b>Criação de funções VB.NET e procedimentos</b>	Acessar objetos no BluePlant diretamente a partir do código
<b>Execução de scripts em eventos e programação</b>	Fácil conexão com tags e eventos do processo usando scripts
<b>Supporte a bibliotecas de classe</b>	Criação de classes próprias acessíveis de outros scripts e telas
<b>Editor .NET integrado com Intellisense</b>	Auxiliar na seleção de nomes da tag e propriedades de objetos
<b>Supporte a exceções e mensagens de rastreamento</b>	Todo o .NET Framework, componentes externos e serviços são facilmente integrados

**Tabela 3-2. Linguagens .NET e Scripts**

### 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.

Alarmes e Segurança	
<b>Múltiplas condições de alarme</b>	Hi, HiHi, Lo, LoLo, taxa de alteração e desvio
<b>Alta resolução</b>	Intervalo da estampa de tempo em milissegundo (quando disponível), usando o tempo remoto de E/S, não a hora do computador
<b>Objetos de visualização integrados</b>	Objeto gráfico online e histórico, quando executado localmente ou na Web
<b>Grupo de alarme e item de objetos</b>	Acessar as propriedades dos alarmes diretamente, por exemplo, "total de alarmes ativos", sem a necessidade de criar tags na aplicação

**Tabela 3-3. Alarmes e Segurança****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.

Tendência e Historiadores	
<b>Conexão com banco de dados ADO</b>	Informações do historiador podem ser salvas em qualquer base de dados externa com suporte ADO.NET
<b>Banco de dados SQL integrado</b>	Quando não definidos como um banco de dados externo, o log acontece no banco de dados SQL integrado
<b>Alta resolução</b>	Intervalo da estampa de tempo em milissegundos (quando disponível), usando o tempo de E/S remoto, não a hora do computador
<b>Disparador por Tag ou grupo</b>	Permitir o salvamento de um registro de acordo com a alteração do Tag ou baseado em eventos de processo
<b>Banda morta do historiador por Tag</b>	Permitir a definição de uma variação mínima de Tag para disparo do registro
<b>Configuração do intervalo de tempo mínimo</b>	Permitir a definição de um intervalo mínimo para gravação, permitindo a criação de bancos de dados mais compactos
<b>Tabelas de banco de dados com múltiplos tags</b>	Permitir a criação de um grupo de tags e armazena-os na mesma tabela de dados para acelerar a gravação e carregamento
<b>Objeto de visualização de tendência integrado</b>	Objeto gráfico de histórico e online executado localmente ou na Web

**Tabela 3-4. Tendência e Historiadores****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-serials. 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.

Dispositivos e Redes	
<b>Importação de configurações de pontos de dados</b>	Copiar e colar do Microsoft Excel, importar bancos de dados de servidores CSV ou OPC
<b>Comunicação executada em processo isolado</b>	Proteção total para o ambiente do Runtime e desempenho avançado em processadores multi-core
<b>Fácil comunicação com vários canais</b>	Criar automaticamente múltiplas tarefas nos protocolos multi-serial ou TCP/IP
<b>Nomeação abstrata para nós e estações</b>	Fornecer maneira fácil de renomear e manter o endereço IP e a configuração de rede de E/S
<b>Criação dinâmica de blocos otimizados</b>	Seleção simples de pontos de leitura e escrita, baseada no protocolo, e criação de blocos otimizados
<b>Configuração de pontos segue a sintaxe do dispositivo</b>	Endereçar pontos do dispositivo, esse endereçamento é usado nas ferramentas de programação do CP
<b>Propriedades de canais e nós</b>	Acessar as propriedades diretamente, por exemplo, status do nó e tags da aplicação não são necessários
<b>Customização de eventos de escrita</b>	Configuração fácil para comandos e eventos, escrita de todos os eventos ou apenas no caso de alteração de valor

**Tabela 3-5. Dispositivos e Redes**

### 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.

Conjunto de dados	
<b>Acessar textos, CSV e arquivos XML</b>	Definir a ligação em tempo real com tags e conteúdo de arquivo
<b>Definição de múltiplas fontes de banco de dados</b>	Gerenciar facilmente múltiplas conexões de banco de dados
<b>Mapeamento de tags com tabelas de dados</b>	Utilitário de configuração de alto nível para gerenciar as tabelas de banco de dados usadas no projeto
<b>Definição de consultas e mapeamentos</b>	Gerenciar várias consultas disparadas por eventos de processo e condições de filtro usando pontos de dados em tempo real
<b>Poderoso objeto de visualização de grade de dados</b>	Abrangente e poderoso objeto de grade de dados para criar interfaces de usuário tanto locais quanto na Web
<b>Propriedades de tabela e consultas</b>	Acessar propriedades (por exemplo, contagem de linhas) diretamente, sem a necessidade de criação de tags na aplicação

**Tabela 3-6. Conjunto 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.

Relatórios	
<b>Editor integrado</b>	Editor de texto fácil de usar, permitindo a inclusão de tabelas, imagens, hiperlinks e formatação de texto
<b>Supporte a texto, HTML e XPS</b>	Salvar relatórios em vários formatos, como XPS, por exemplo, permitindo fácil implantação em ambientes distribuídos
<b>Copiar e colar</b>	Editar no Microsoft Word ou em editor HTML ou RTF, basta copiar e colar o conteúdo do BluePlant
<b>Tags em tempo real facilmente inseridas</b>	Com apenas um clique, o usuário pode adicionar valores de dados em tempo real aos relatórios

**Tabela 3-7. Relatórios****Nota:**

**Geração de relatório:** Não é possível fazer relatórios baseados em resultados dinâmicos a partir do banco de dados, ou seja, a pesquisa deve sempre retornar um número exato de linhas.

**Nome do relatório:** Limitado a 63 caracteres e é bloqueado o uso de caracteres especiais (como ., \$ %, etc).

**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 accessem diretamente todos os objetos do Runtime criados no projeto.

Isso significa que tags temporários não são necessários 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 consequente 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.

Ferramentas de Diagnóstico e do Runtime	
<b>Modo teste</b>	Executar projetos com proteção, como, somente leitura em dispositivos externos ou arquivos temporários no historiador
<b>Informações do módulo</b>	Ferramentas avançadas para perfil de desempenho e diagnósticos de sistemas internos
<b>Ferramenta de localização</b>	Criar interface de usuário do operador em várias linguagens e, dinamicamente, alterna entre elas no Runtime
<b>Janela de rastreamento</b>	Ao criar uma aplicação esta ferramenta fornece monitoração de Tag e mensagens de diagnóstico do sistema
<b>Monitoração de propriedade</b>	Verificar e simular valores de Tag e propriedades, iniciar e parar módulos funcionais

**Tabela 3-8. Ferramentas do Runtime e de Diagnóstico**

## 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.

Ferramentas de Implantação e Teste de Projeto	
<b>Abertura de vários projetos</b>	Abrir vários projetos simultaneamente no computador
<b>Engenharia remota</b>	Acessar e editar remotamente a configuração do projeto
<b>Execução como serviço do Windows</b>	Executar o Runtime no servidor, instalado como um serviço do Windows
<b>Proteção de chaveamento de aplicações</b>	Proteger contra chaveamentos de aplicações não autorizadas em interfaces do operador usando as teclas CTRL+ALT+DEL do Windows ou outras
<b>Atalhos de inicialização</b>	Utilizar atalhos simples de inicialização e parâmetros para customização de inicialização
<b>Projeto de arquivo único e recursos embarcados</b>	A configuração de todo o projeto é salva em um único arquivo protegido, incluindo todas as imagens e bitmaps usados em telas e relatórios

**Tabela 3-9. Ferramentas de Implantação e Teste de Projeto**

## Drivers de Comunicação

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

Drivers de Comunicação	
<b>MODBUS</b>	RTU-TCP RTU-TCP Escravo
<b>OPC</b>	DA

**Tabela 3-10. Drivers de Comunicação**

## 4. BluePlant

### Requisitos Mínimos para Instalação e Operação

Os modelos BluePlant Student, BluePlant Express e BluePlant Lite apresentam quatro canais disponíveis e o modelo BluePlant Enterprise conta com até 16 canais. As tabelas a seguir mostram os requisitos mínimos para instalação e operação do BluePlant usando diferentes quantidades de canais.

BluePlant Express, BluePlant Student, BluePlant Lite e BluePlant Enterprise (até 4 canais)	
<b>Plataforma</b>	Windows 7 Service Pack 1 (x86 e x64), Windows 8.1 (x86 e x64), Windows 10 (x86 e x64), Windows Server 2008 R2 SP1 (x64) ou Windows Server 2012 R2 (x64)
<b>Processador</b>	Processador a 1 GHz ou superior
<b>Espaço em disco</b>	2,5 GB de espaço em disco disponível (mínimo)
<b>Memória RAM</b>	2 Gbytes (mínimo), 4 Gbytes (recomendado)
<b>Resolução</b>	1024 x 768 (mínimo), 1280 x 1024 (recomendado)
<b>Idioma</b>	Qualquer idioma

**Tabela 4-1. Configuração para até 4 Canais**

BluePlant Enterprise (até 8 canais)	
<b>Plataforma</b>	Windows 7 Service Pack 1 (x64), Windows 8.1 (x64), Windows 10 (x64), Windows Server 2008 R2 SP1 (x64) ou Windows Server 2012 R2 (x64)
<b>Processador</b>	Intel Core i5 (mínimo)
<b>Espaço em disco</b>	2,5 GB de espaço em disco disponível (mínimo)
<b>Memória RAM</b>	4 Gbytes (mínimo), 6 Gbytes (recomendado)
<b>Resolução</b>	1024 x 768 (mínimo), 1280 x 1024 (recomendado)
<b>Idioma</b>	Qualquer idioma

**Tabela 4-2. Configuração para até 8 Canais**

BluePlant Enterprise (até 16 canais)	
<b>Plataforma</b>	Windows 7 Service Pack 1 (x64), Windows 8.1 (x64), Windows 10 (x64), Windows Server 2008 R2 SP1 (x64) ou Windows Server 2012 R2 (x64)
<b>Processador</b>	Intel Core i7 (mínimo)
<b>Espaço em disco</b>	2,5 GB de espaço em disco disponível (mínimo)
<b>Memória RAM</b>	6 Gbytes (mínimo), 8 Gbytes (recomendado)
<b>Resolução</b>	1024 x 768 (mínimo), 1280 x 1024 (recomendado)
<b>Idioma</b>	Qualquer idioma

**Tabela 4-3. Configuração para até 16 Canais**

BluePlant Enterprise (Redundância de Servidores)	
<b>Plataforma</b>	Windows 7 Service Pack 1 (x64), Windows 8.1 (x64), Windows 10 (x64), Windows Server 2008 R2 SP1 (x64) ou Windows Server 2012 R2 (x64)
<b>Processador</b>	Intel Core i7 (mínimo)
<b>Espaço em disco</b>	2,5 GB de espaço em disco disponível (mínimo)
<b>Memória RAM</b>	16 Gbytes
<b>Resolução</b>	1024 x 768 (mínimo), 1280 x 1024 (recomendado)
<b>Idioma</b>	Qualquer idioma

**Tabela 4-4. Configuração para Redundância de Servidores**

**Notas:**

**Plataforma:** Sistemas operacionais que não possuem identificação de tipo de processador compatível, considerar somente 64 bits.

**Plataforma:** É necessária a instalação do Microsoft .NET Framework 4.6.

**ATENÇÃO:**

A quantidade de canais e o desempenho da aquisição de dados são os principais motivos para selecionar os requisitos de operação. Se forem necessários mais de 16 canais ou se existirem outras demandas específicas do projeto, recomenda-se fortemente contatar o Suporte Técnico da Altus através do site [www.altus.com.br](http://www.altus.com.br) ou e-mail [altus@altus.com.br](mailto:altus@altus.com.br).

## Instalação

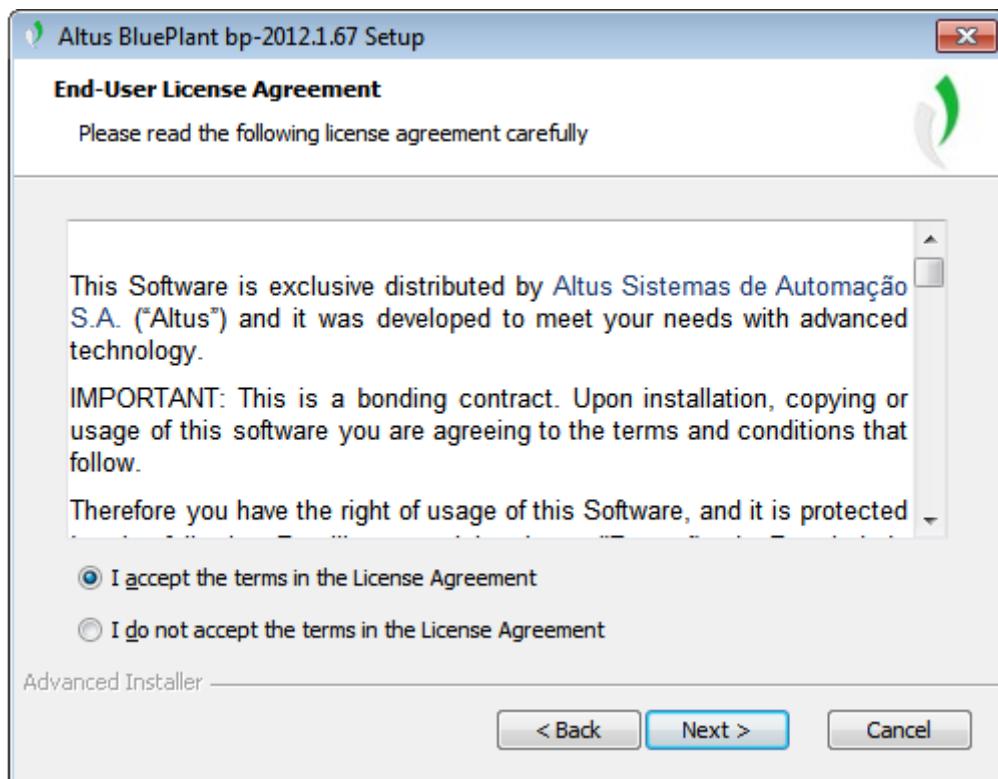
Para executar a instalação do software BluePlant é necessário realizar o download do arquivo de instalação a partir do site [www.altus.com.br](http://www.altus.com.br). Em seguida, devem ser fechados todos os programas em execução no PC, clicar duas vezes no arquivo de instalação.

A primeira tela do instalador será a tela de entrada do instalador, clique em *Next* para continuar a instalação do BluePlant.



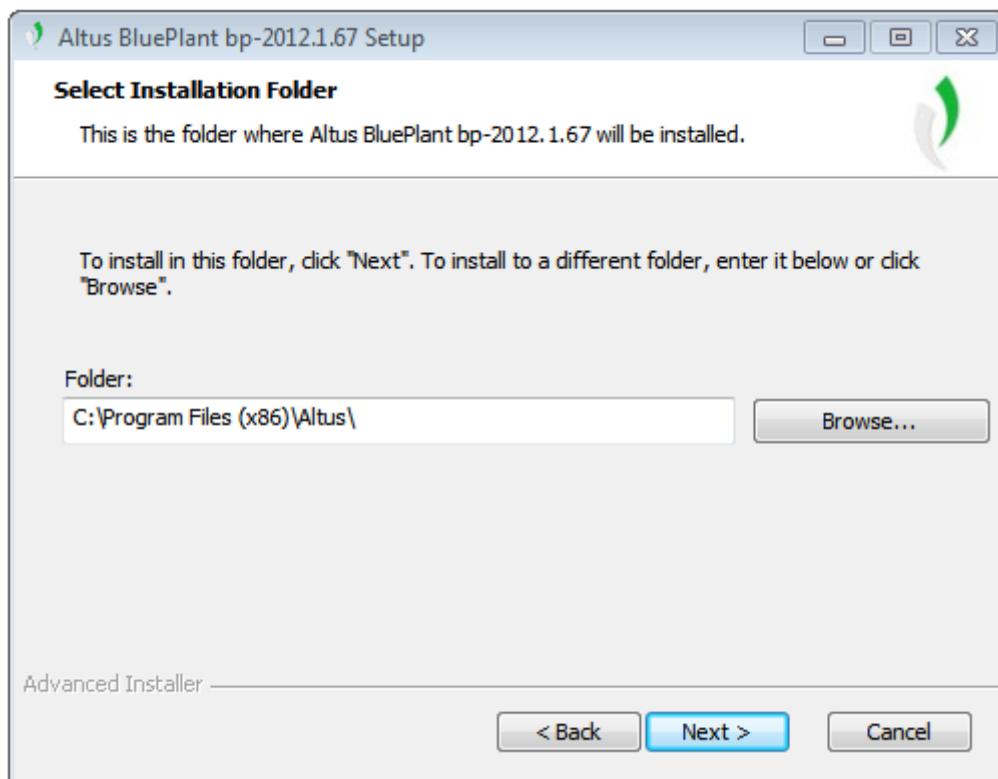
**Figura 4-1. Tela de Entrada**

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 *Next*.



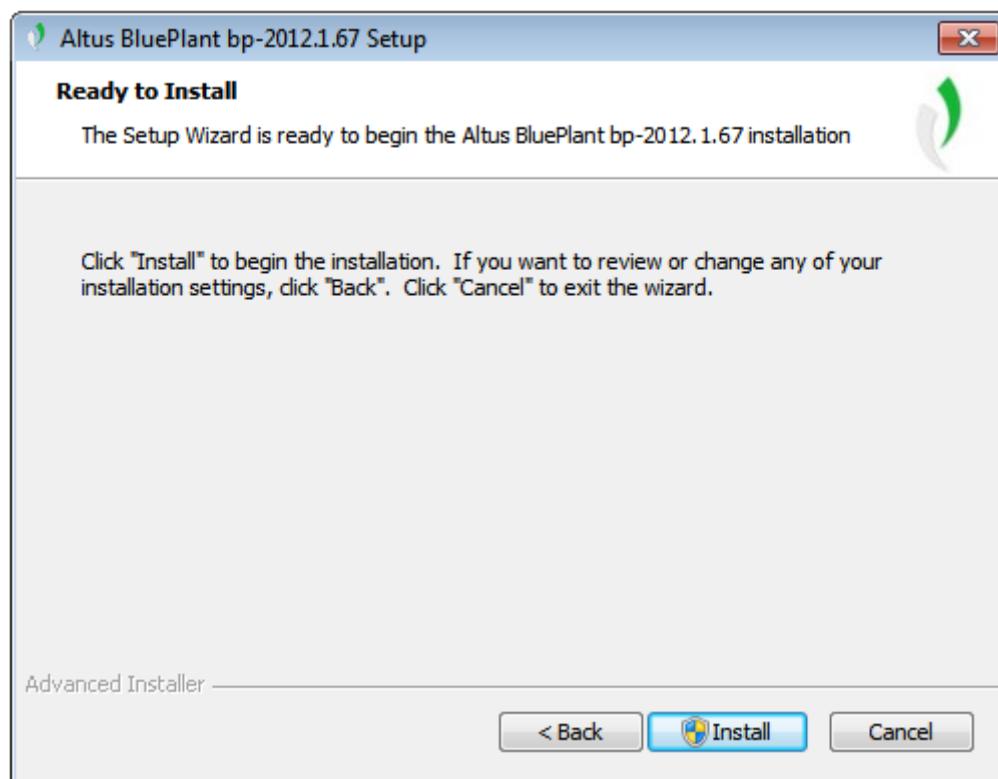
**Figura 4-2. Tela do Contrato de Licença do BluePlant**

Aparecerá a tela de seleção da pasta de instalação, clicar no botão *Next* para prosseguir.



**Figura 4-3. Tela de Seleção da Pasta de Instalação**

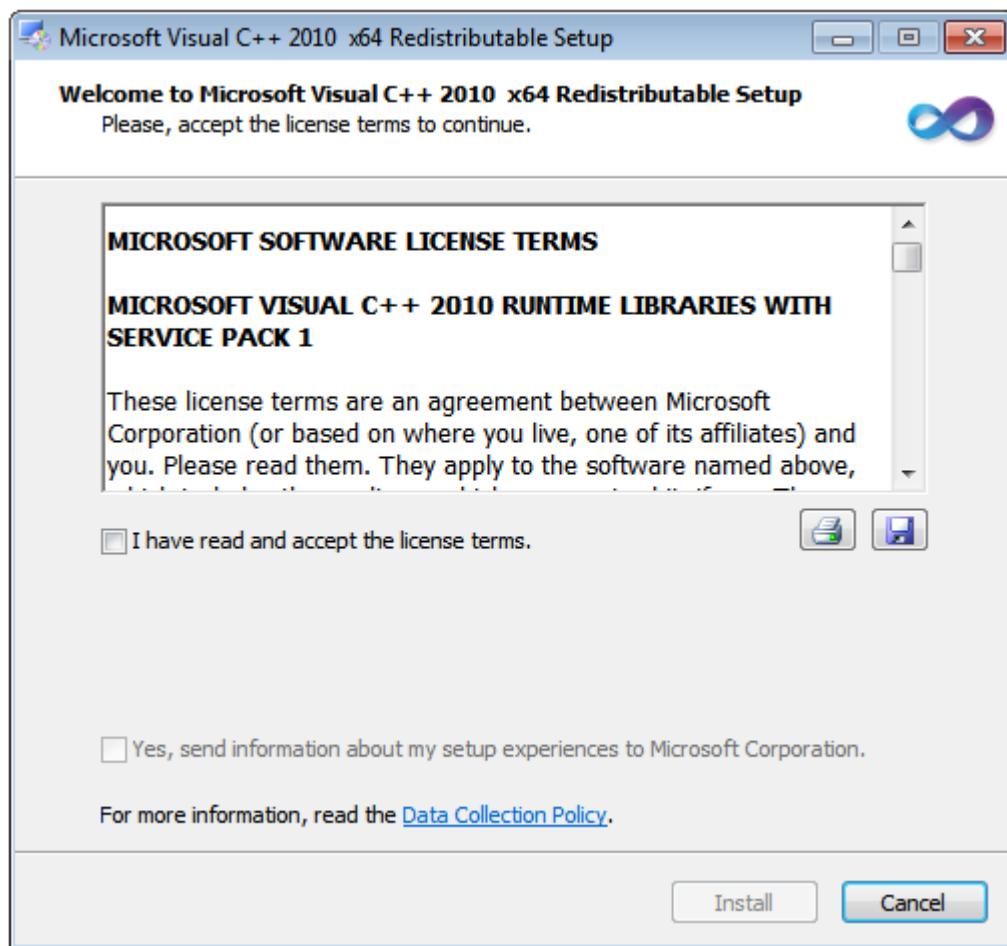
A próxima tela a aparecer habilita o botão de instalação.



**Figura 4-4. Tela de Inicialização da Instalação**

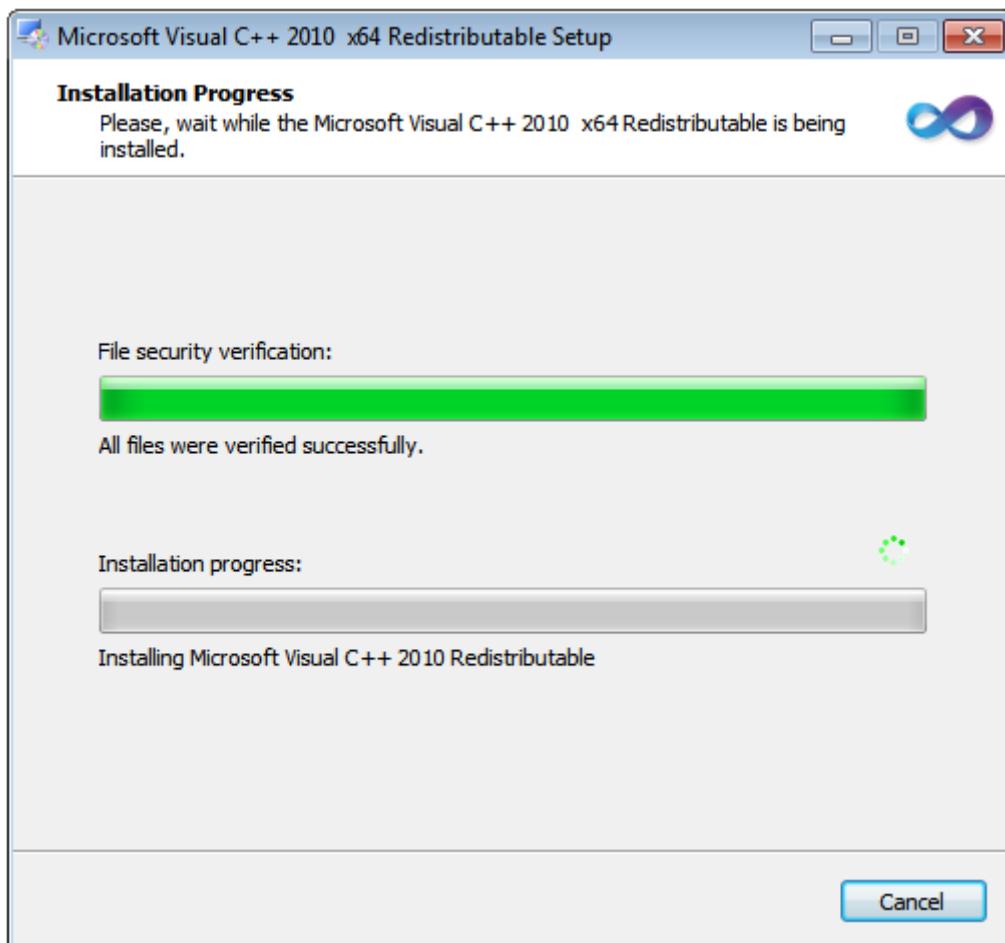
Normalmente na primeira instalação é necessário instalar os pré-requisitos do software, que são compostos basicamente pelo Microsoft Visual C++ e Advosol OPC Core Components.

Assim, da-se início a instalação do Microsoft Visual C++, cuja primeira tela apresentada é a dos termos de licença.



**Figura 4-5. Tela dos Termos de licença do Microsoft Visual C++**

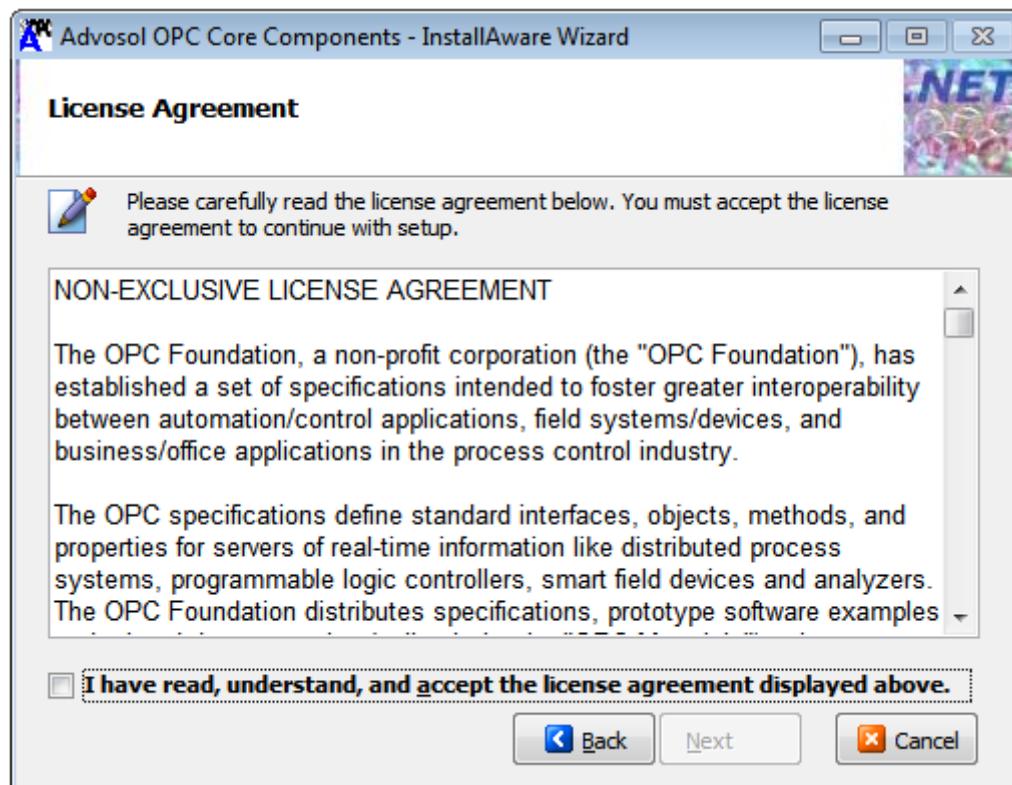
Ao apertar o botão Install, inicia a instalação do software.



**Figura 4-6. Tela de Instalação do Microsoft Visual C++**

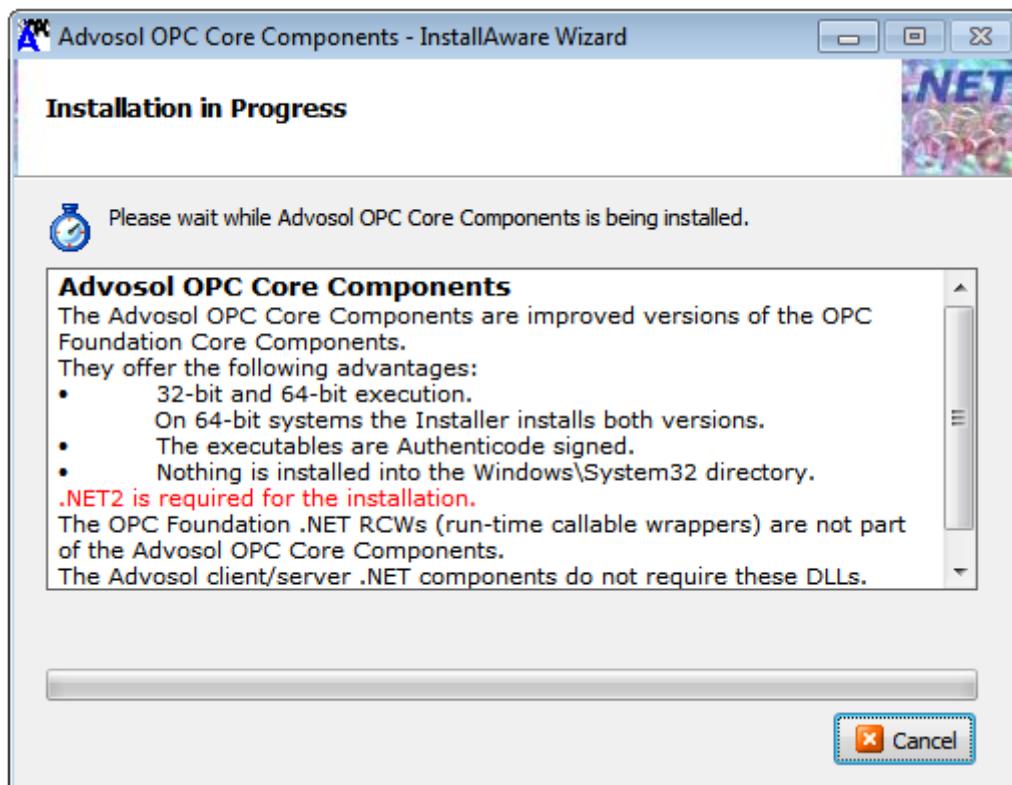
Durante a instalação, aparecerá outra tela de licença, a qual deve ser lida atentamente. Este acordo é do software Advosol OPC Core Components. Este software oferece um suporte completo na plataforma .NET para todos os principais padrões OPC. OPC é um conjunto de especificações normatizadas. Tais padrões 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.

Caso os termos sejam aceitos, a caixa de verificação deve ser marcada para continuar a instalação. A próxima tela da instalação irá aparecer, para prosseguir, clicar no botão Next.



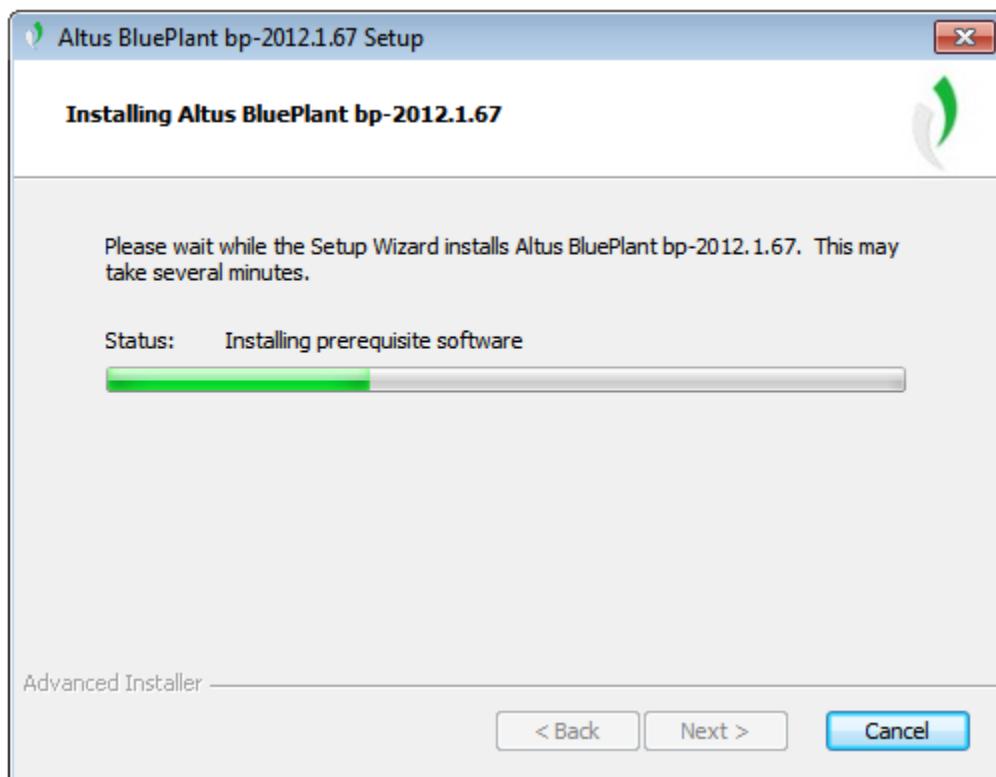
**Figura 4-7. Tela do Contrato de Licença do Advosol**

A tela de instalação do Advosol OPC Core Components deve aparecer. Aguardar enquanto os arquivos necessários são instalados no computador. Isso pode levar alguns minutos dependendo da configuração do computador.



**Figura 4-8. Tela de Instalação do Advosol**

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 4-9. Tela de Instalação do BluePlant**

Após a conclusão da instalação, a tela a seguir será exibida. Clicar no botão Finish para concluir o procedimento de instalação.



**Figura 4-10. 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

### Produto Descontinuado

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 Tabela 2-1, 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” conforme Figura 4-14.

#### 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](mailto:altus@altus.com.br).

## Iniciando com o BluePlant

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

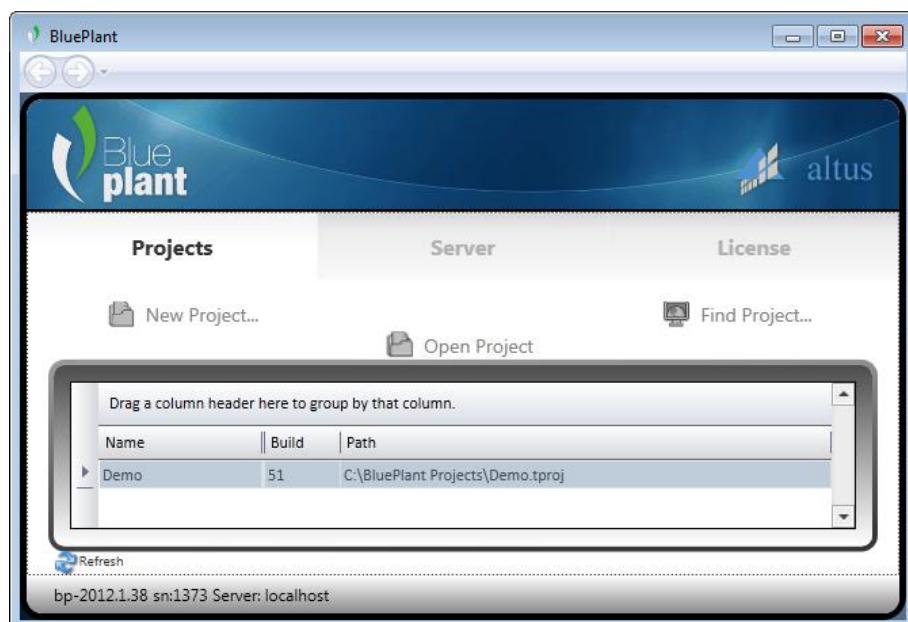


**Figura 4-11. Ícone do BluePlant**

## 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*

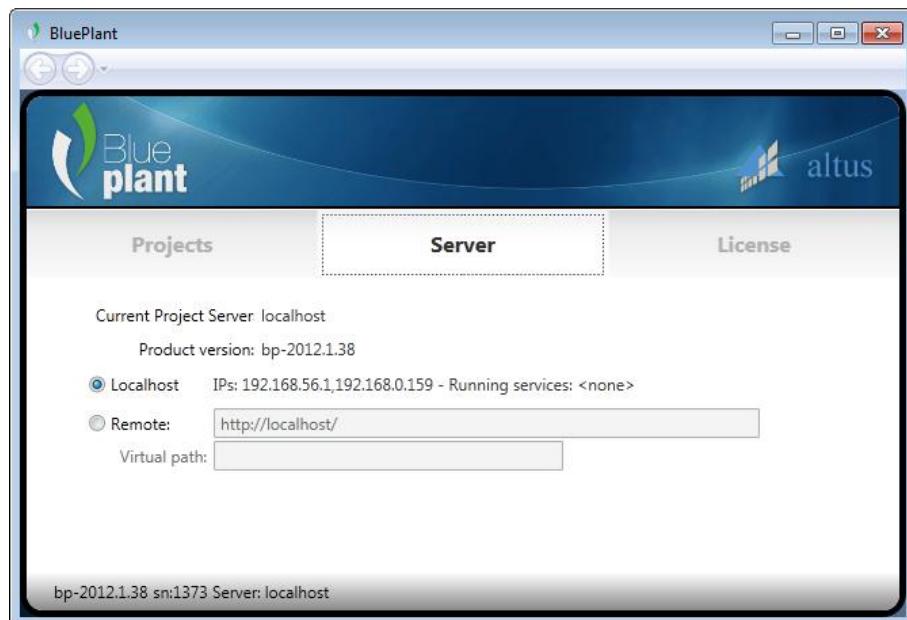


**Figura 4-12. Tela de Gerenciamento de Projeto**

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.

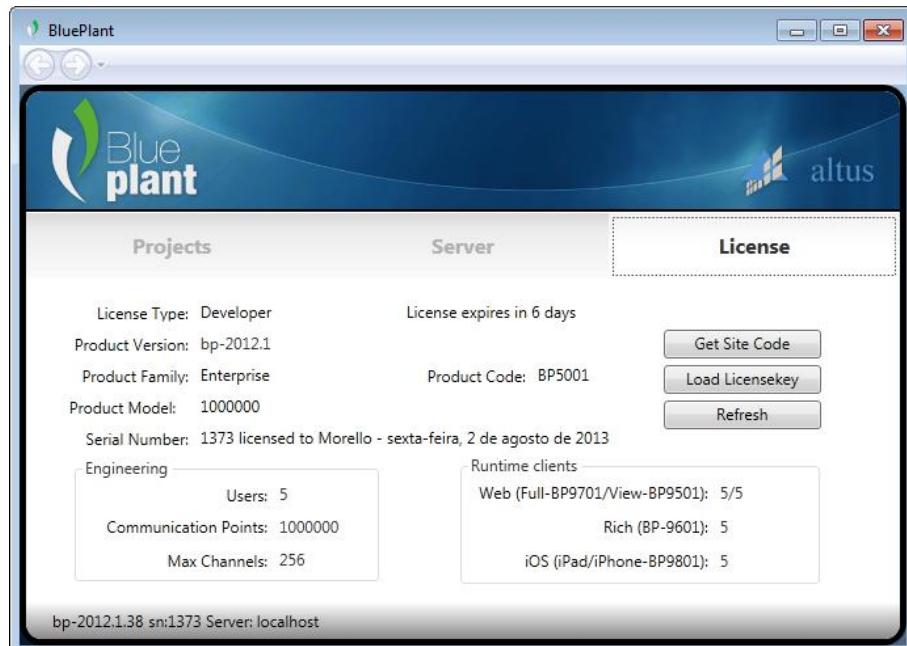
Na guia “*Projects*”, estão os ícones para criar um projeto *New Project*, para abrir um projeto selecionado *Open Project...* e para localizar outros projetos *Find Project...* que não estão localizados no diretório padrão do BluePlant.

A guia *Server* tem a aparência mostrada na Figura 4-13 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 4-13. 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 softkey como pode ser visto na Figura 4-14.



**Figura 4-14. Guia Licenças**

## Primeiros Passos com o BluePlant

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 manual.

## Selecionando o Projeto Demo

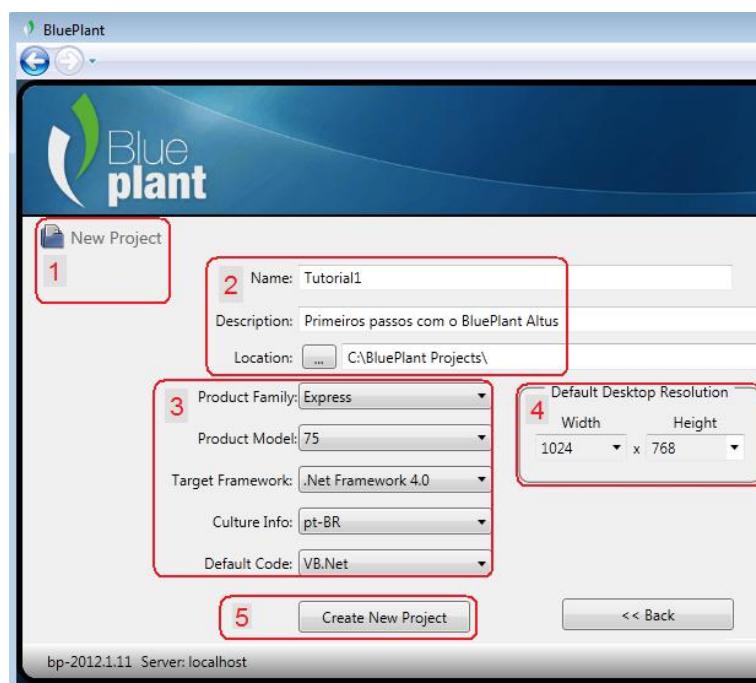
A partir da guia “Projects...” no editor de projetos, pode ser selecionado o projeto Demo.tproj, o qual contempla uma visão geral das funcionalidades e recursos do BluePlant.

Para abrir o projeto, basta selecionar o projeto presente na lista e clicar em “Open Project”.

## 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...* que pode ser visto na Figura 4-12.

Na janela seguinte, apresentada na Figura 4-15, estão indicados os passos para criar um novo projeto.



**Figura 4-15. Criando um Novo Projeto**

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.

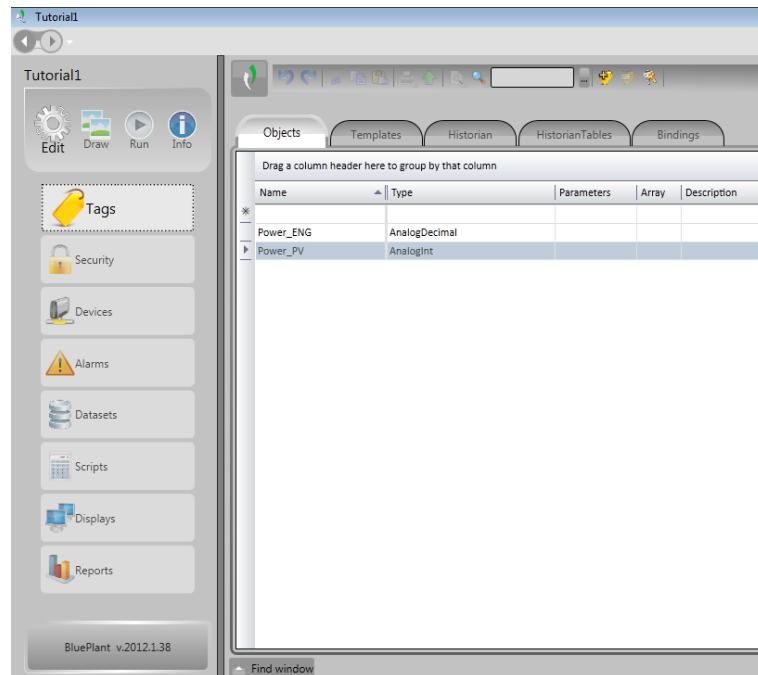
Notas:

- Campo *Nome* tem limite de no máximo 64 caracteres
- Campo *Descrição* tem limite de no máximo 256 caracteres
- 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 4-16 ilustra a interface do usuário.



**Figura 4-16. Interface do Usuário**

**Nota:**

**Colunas Comuns:** Durante a edição de projeto no BluePlant o usuário se deparará com diversas opções comuns nas tabelas de configuração do software. Segue a definição destas colunas:

- **ID:** ID do registro de banco de dados atribuído a este objeto na tabela. Os números são atribuídos iniciando de 1 para o primeiro objecto criado e continuam crescendo sequencialmente conforme necessário.
- **BuiltIn:** Indica se faz parte da configuração default do projeto. Linhas marcadas com BuiltIn não podem ser deletadas.
- **DateModified:** Data em que o objeto foi modificado pela ultima vez
- **DateCreated:** Data em que o objeto foi criado inicialmente.
- **Comment:** Espaço para adicionar um coment
- **Description:** Espaço para digitar uma descrição do objeto.

**Refresh:** Quando a coluna está em edição e algum campo dela é alterado pelo software, para não atrapalhar a edição do usuário, este campo não é modificado instantaneamente, aparecendo o símbolo de refresh (↻ Refresh) no canto superior direito do ambiente em edição. Ao finalizar a edição, o usuário deve pressionar o símbolo para ter sua tela atualizada.

### Criando Tags no Projeto



No ambiente de edição é possível editar objetos como, por exemplo, tags. A Figura 4-17 mostra o caminho para a edição de tags. Neste exemplo, foram criados dois tags analógicos: 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.

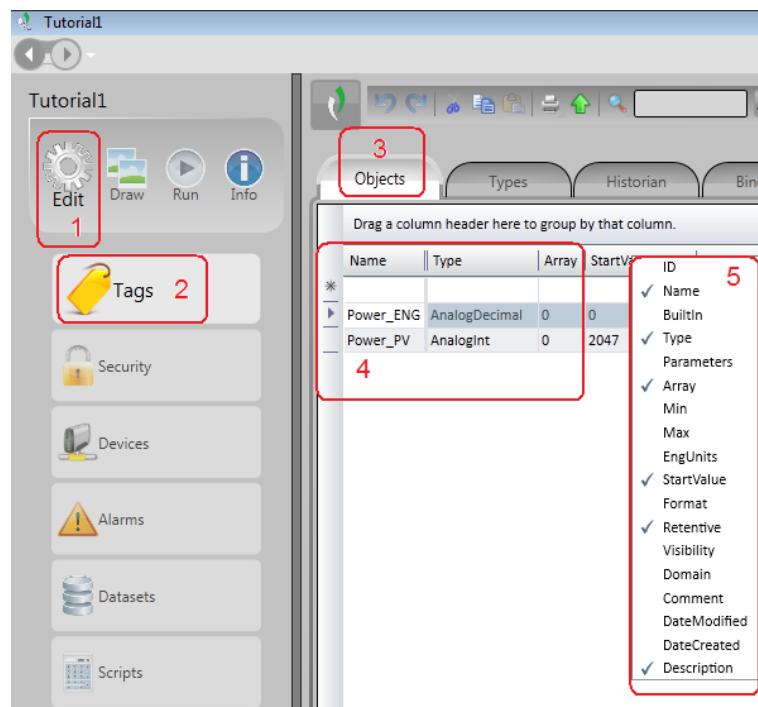


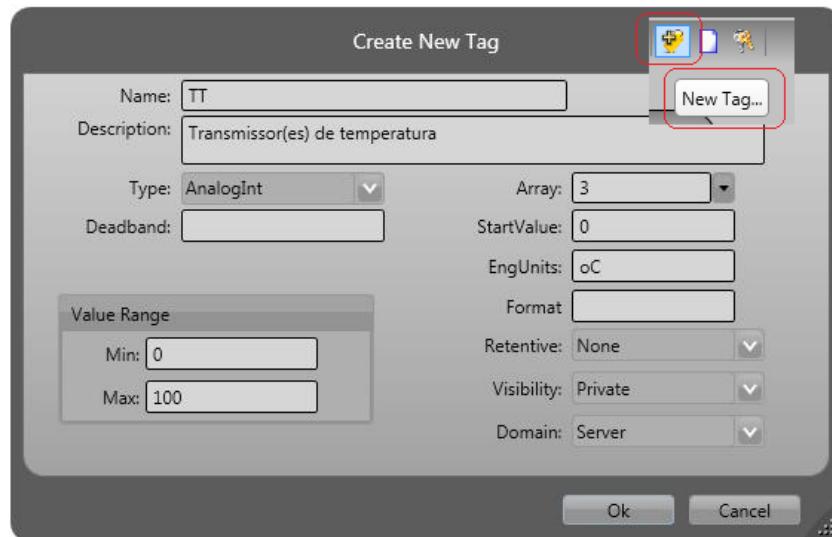
Figura 4-17. Editando Tags do Projeto

Os passos para a criação desses 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 os tags do usuário
4. Digitar os dados relativos ao novo 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 dos 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 um novo tag acessando a barra de ferramentas no topo do ambiente de edição. Basta clicar no ícone e configurar os parâmetros do tag.

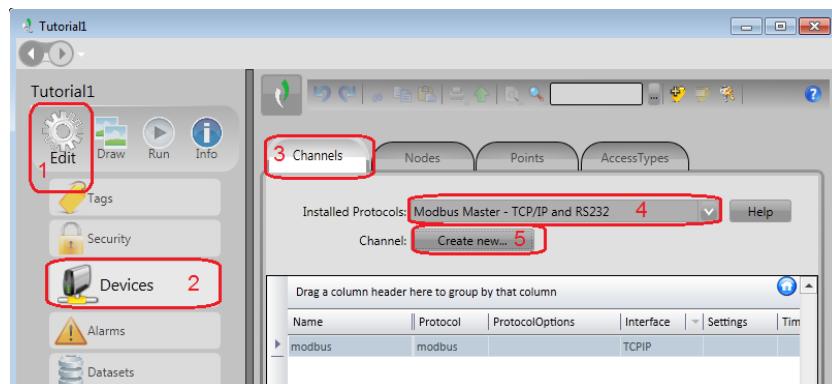
A Figura 4-18 mostra o processo de criação de um tag vetor de uma dimensão, com três posições do tipo analógico inteiro a partir do Menu Editar. Um tag vetor é um conjunto de tags com o mesmo nome, cuja identificação se dá através de índices. Neste exemplo, o acesso em outros locais da aplicação a esse tag, se dá via sintaxe  $TT[0]$ ,  $TT[1]$ ,  $TT[2]$  e  $TT[3]$ .



**Figura 4-18. Processo de Criação de Tags**

#### Associando tags a um protocolo comunicação

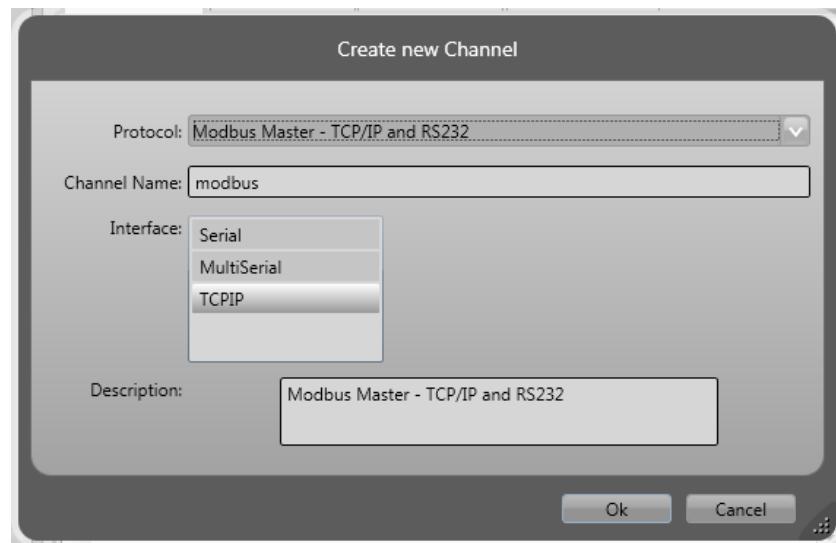
Para realizar a associação das tags criadas a um protocolo de comunicação, os passos descritos a seguir devem ser executados.



**Figura 4-19. Processo de Criação de Canal de Comunicação**

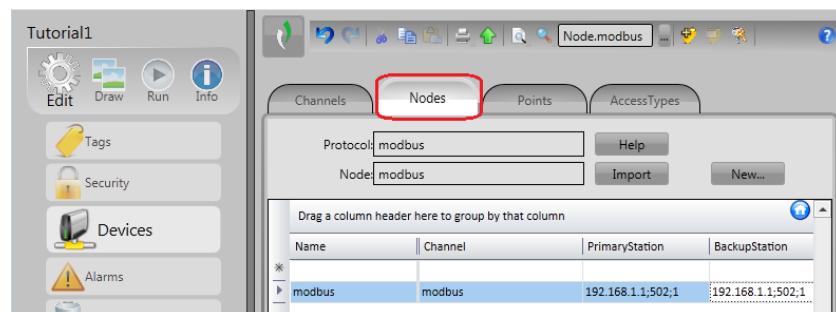
1. Acessar o menu *Edit*
2. Selecionar a opção *Devices*
3. Acessar a guia *Channels* na qual deve aparecer o campo com os protocolos disponíveis
4. Selecionar no campo *Installed Protocols* o protocolo de comunicação a ser utilizado
5. Para confirmar a criação do canal, clicar no botão *Create new...*

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 4-20.



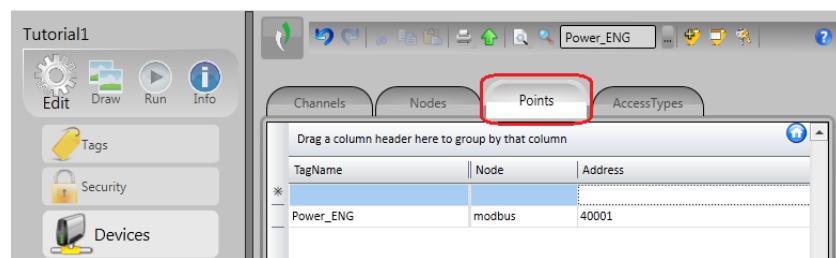
**Figura 4-20. Opções de Protocolos para Criação de um Canal**

Na guia *Nodes* é o local onde são criados os nós de acordo com os canais existentes. Para mais detalhes de como criar e configurar nós, pode ser consultado no capítulo **Editando Dispositivos**.



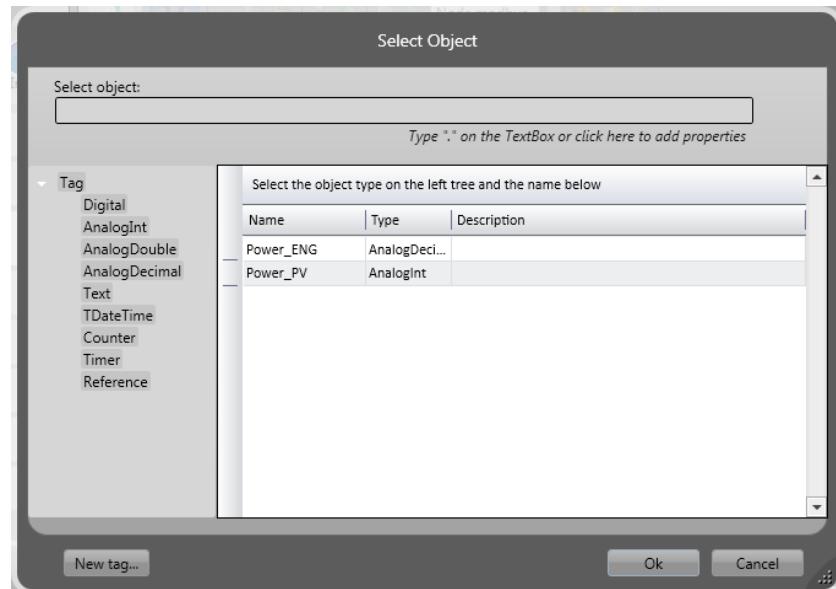
**Figura 4-21. Inclusão e Configuração de Nós**

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



**Figura 4-22. Associação de Tags com Nós e Endereços de Protocolos**

Ao selecionar o campo *TagName* da guia *Point*, deve abrir uma janela como apresentado na Figura 4-23, onde estão todos os tags criados e podem ser selecionados para associar ao nó criado.



**Figura 4-23. Seleção do Tag para Associar a um Nô**

### 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 4-24 conforme a numeração correspondente.

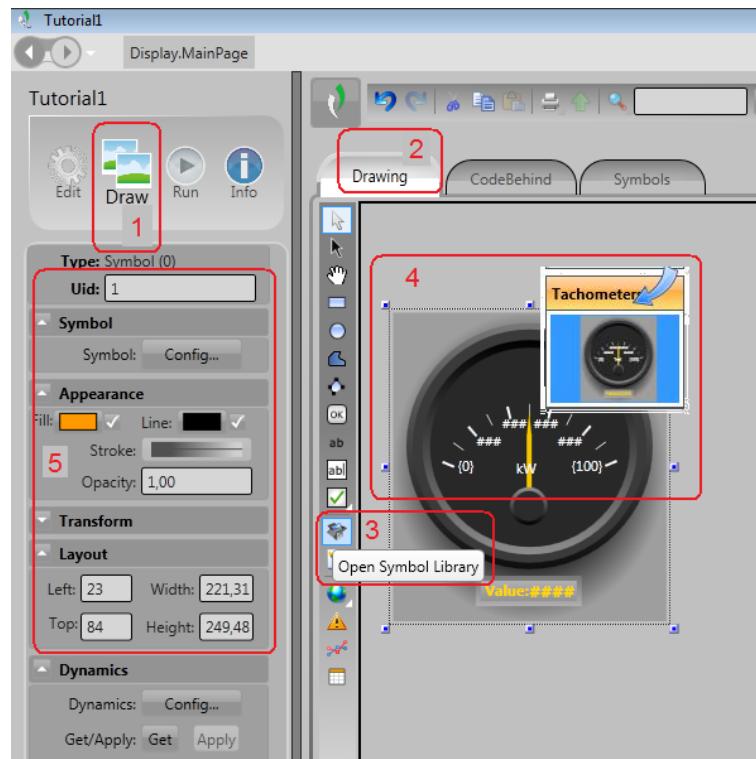


Figura 4-24. Inclusão de um Símbolo do Tipo Tacômetro

Para associar um 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 4-25 mostra essas etapas de associação de tag.

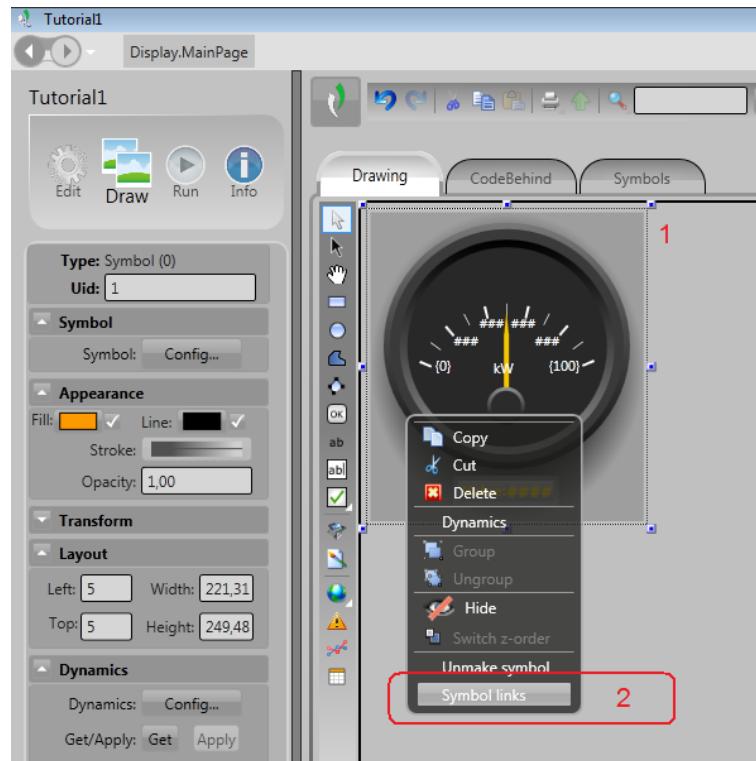


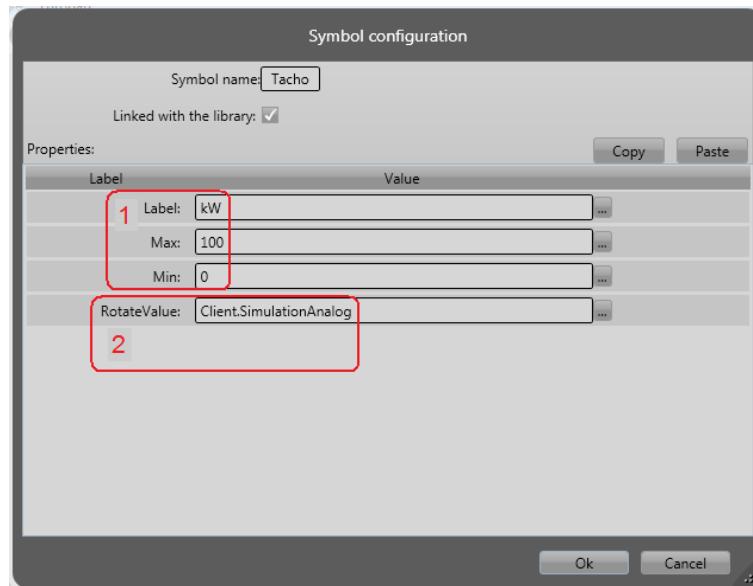
Figura 4-25. Associação de Tag a um Símbolo

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

1. Configurar as propriedades do símbolo (rótulo e valores máximo/mínimo)
2. Associar um tag ao símbolo no campo *RotateValue*

Neste caso foi utilizado o tag cliente *SimulationAnalog* para simular o movimento do ponteiro do medidor.

A Figura 4-26 mostra essa configuração.



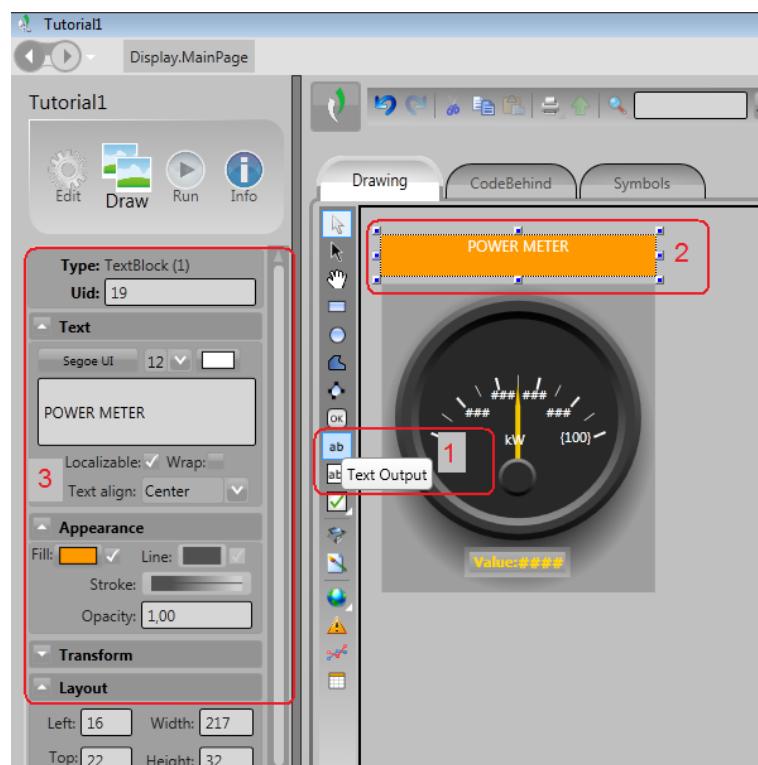
**Figura 4-26. Configuração do Tag Associado ao Símbolo**

#### *Inclusão de Texto*

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

1. Selecionar a opção de *Saída de Texto*
2. Posicionar o texto no local desejado
3. Configurar as propriedades do texto

A Figura 4-27 ilustra esta sequência.



**Figura 4-27. 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* (Inicialização) e *Publish* (Publicar).

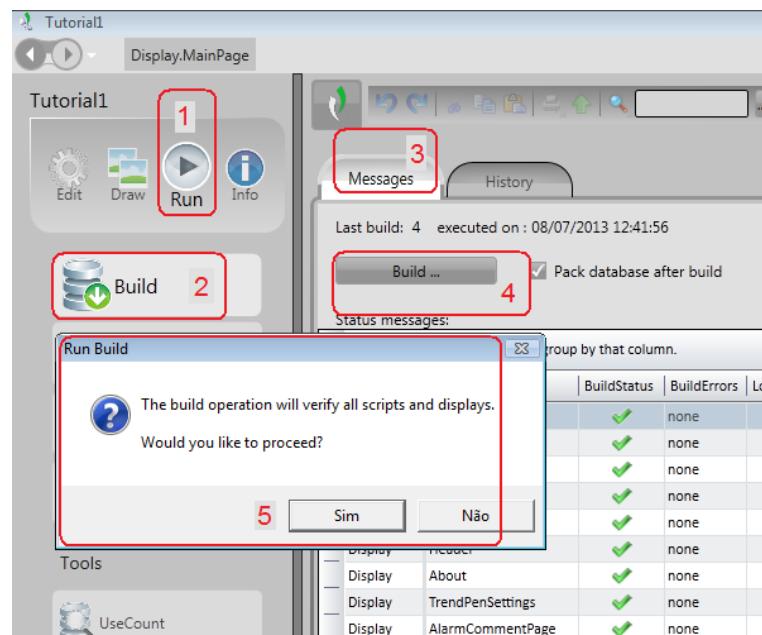
Além disso, nesse menu estão incluídas as seguintes ferramentas: *UseCount* (Contagem de uso), *Localization* (Localização) e *Extensions* (Extensões), as quais serão exploradas no decorrer desse manual.

### 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:

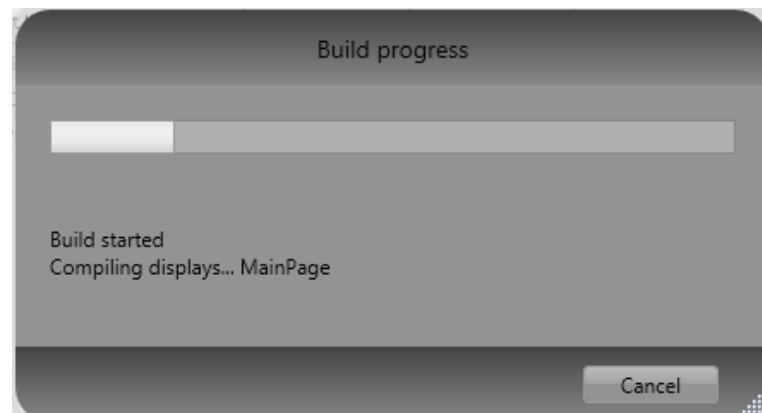
1. Selecionar o menu *Run*
2. Selecionar a opção *Build*
3. Acessar a aba *Messages*
4. Compilar o projeto
5. Confirmar a compilação

A Figura 4-28 ilustra o processo de compilação do projeto.



**Figura 4-28. Compilação do Projeto**

A Figura 4-29 mostra a janela de progresso da compilação.



**Figura 4-29. Progresso da Compilação**

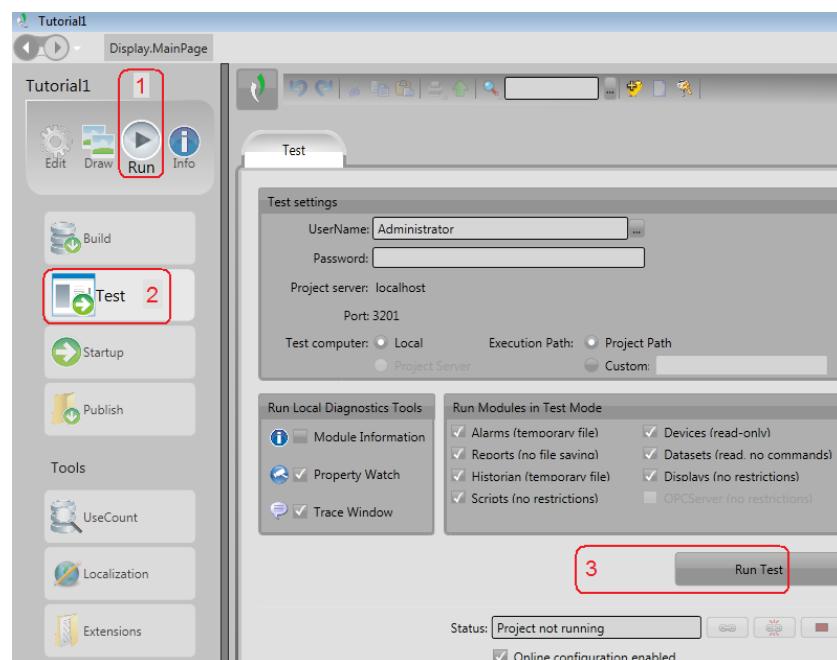
A Figura 4-30 ilustra o resultado da compilação e os erros, se existirem.

Status messages:			
Drag a column header here to group by that column.			
Module	Object ID	BuildStatus	BuildErrors
Script	Class.ServerMain	✓	none
Script	Class.ClientMain	✓	none
Display	MainPage	✓	none
Display	SelectPage	✓	none
Display	LogOn	✓	none

**Figura 4-30. Resultado da Compilação**

## Teste do Projeto

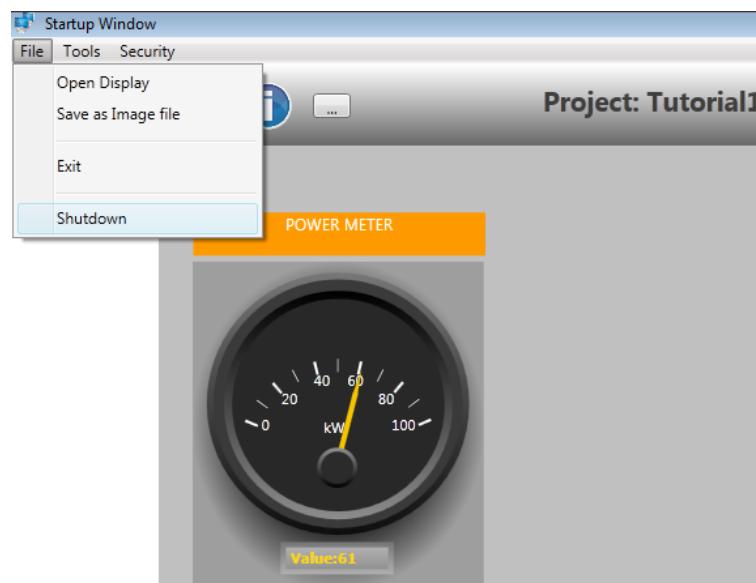
A Figura 4-31 mostra os passos descritos para testar o projeto.



**Figura 4-31. 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 diagnósticos e os módulos a serem testados
3. Iniciar o teste com as configurações definidas, confirmando através do botão *Run Test*

A Figura 4-32 mostra o resultado da execução do teste do projeto.

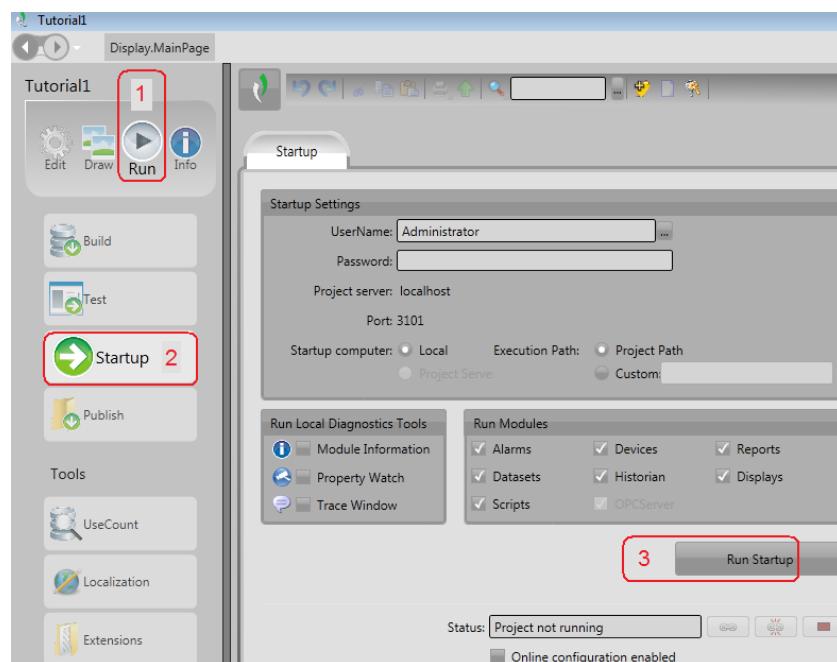


**Figura 4-32. Execução do Teste do Projeto**

Para finalizar o teste do projeto o usuário pode abrir o menu *File* e selecionar a opção *Shutdown* conforme representado na Figura 4-32.

## Inicialização do Projeto

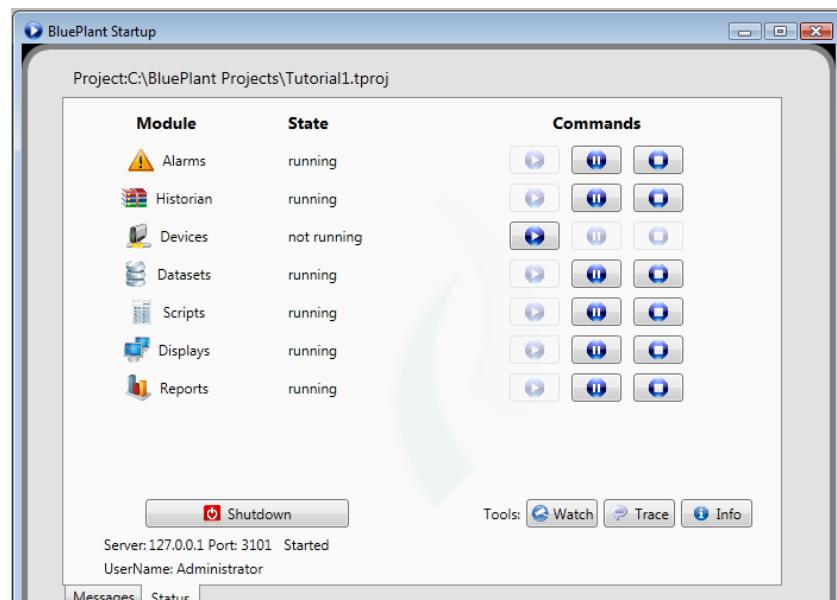
A Figura 4-33 mostra a sequência para inicialização do projeto.



**Figura 4-33. 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*

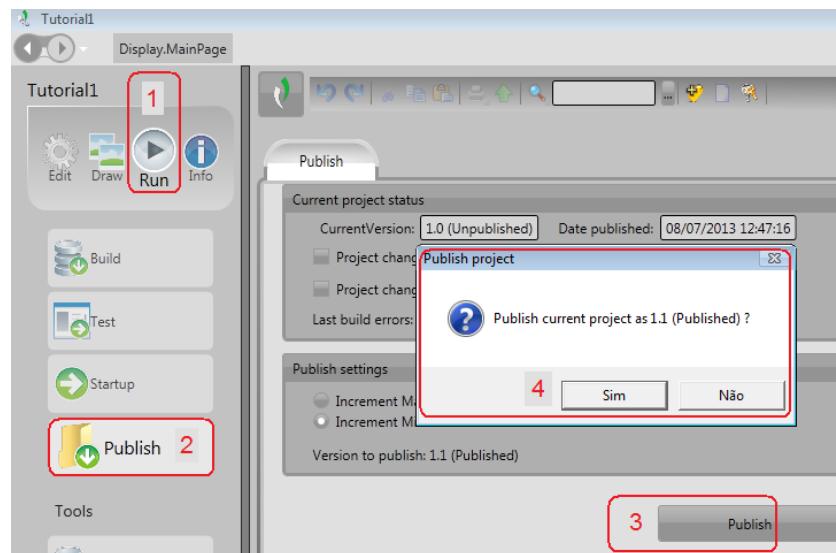
Juntamente com a visualização da tela principal do projeto será aberta uma janela mostrando o status e as mensagens de inicialização conforme pode ser observado na Figura 4-34.



**Figura 4-34. Status da Inicialização do Projeto**

### Publicação do Projeto

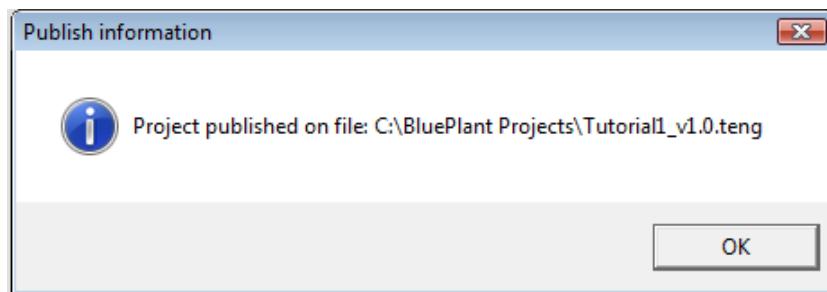
Quando o projeto está pronto para ser executado no campo, deve ser utilizado o recurso de publicação para configurar as opções de redundância (se aplicável) e para criar uma cópia do projeto na modalidade somente leitura com versão controlada, ou seja, uma cópia do projeto para a execução em campo. A extensão do arquivo gerado por ocasião da publicação é “.teng”. A Figura 4-35 mostra a sequência para publicação do projeto.



**Figura 4-35. 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 4-36 será exibida indicando o caminho de publicação do projeto.



**Figura 4-36. Caminho da Publicação do Projeto**

## 5. Menu Principal do BluePlant

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 supervisório compreende usualmente duas etapas:

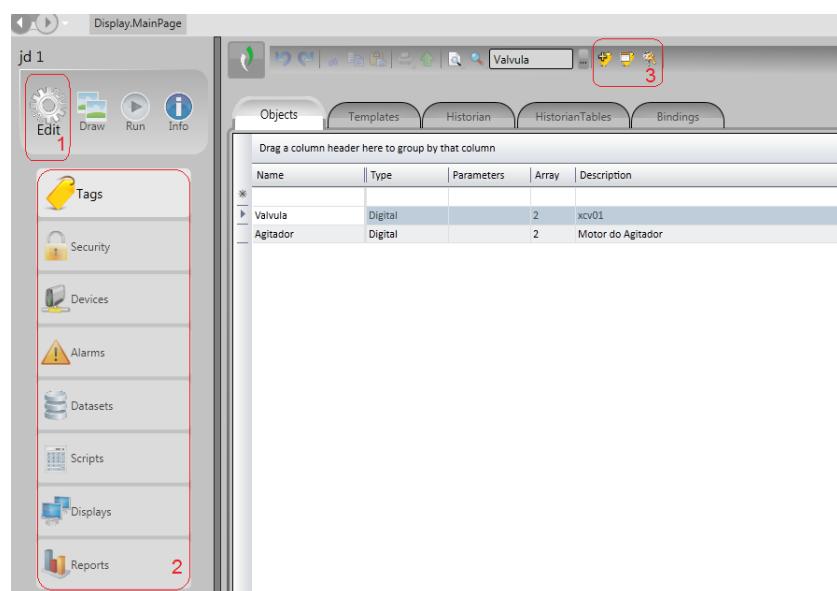
- Definir cada variável de processo na base de dados
- Definir sinóticos, gráficos e relatórios

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.

Neste capítulo são descritos os quatro 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. Esses quatro menus são descritos nas seções Edição da Aplicação, Diagramação da Aplicação, Execução da Aplicação e Informações da Aplicação respectivamente.

### Edição da Aplicação

O menu *Edit* permite acessar as ferramentas para a edição de um projeto no BluePlant. A Figura 5-1 ilustra a seleção desse menu.



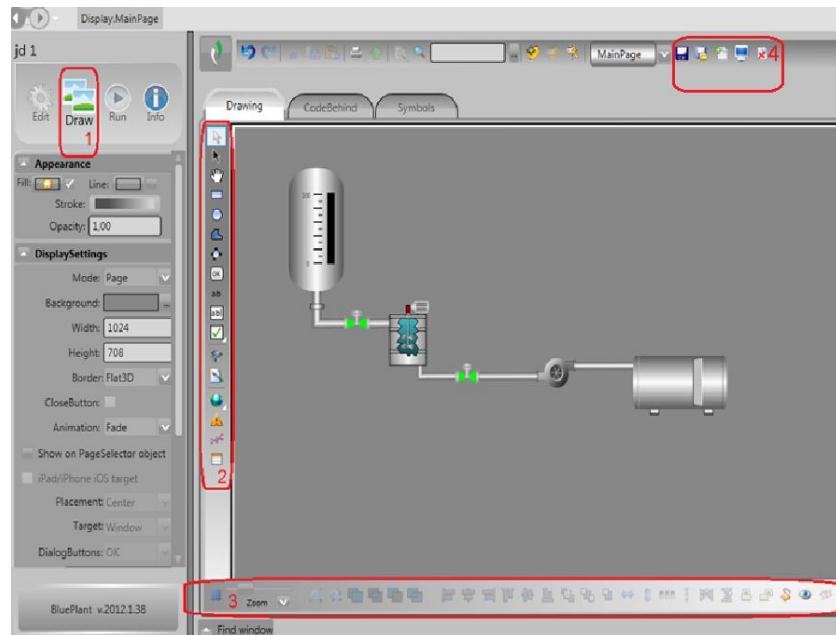
**Figura 5-1. Menu Edit**

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

1. Barra de seleção do menu de edição do projeto
2. Barra de ferramentas do menu de edição, contendo os recursos para a configuração dos seguintes elementos: tags, usuários e suas políticas de segurança, dispositivos e seus protocolos de comunicação, alarmes, banco de dados, linguagem Script, displays e relatórios
3. Atalhos na barra de ferramentas superior, incluindo as definições para criação de um novo tag, as propriedades das tags e as configurações de usuários bem como seu acesso aos elementos da aplicação

## Diagramação da Aplicação

No menu *Draw* é possível criar as telas e símbolos que compõem a aplicação. A Figura 5-2 ilustra a inclusão de símbolos na tela principal (MainPage) com o auxílio das barras de ferramenta associadas.



**Figura 5-2. Menu Draw**

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

1. Barra de seleção do menu de edição do projeto
2. Barra de ferramentas vertical usada para incluir, manipular e configurar os elementos que vão compor as telas da aplicação
3. Barra de ferramentas horizontal localizada abaixo da guia da tela, a qual contém comandos para agrupar, combinar, alinhar, bloquear o(s) componente(s) selecionado(s), entre outros
4. Barra de ferramentas TopToolBar localizada na parte superior da tela principal do menu Draw, a qual possui ícones com as seguintes funções: criar novo(s) display(s), apagar display(s), salvar display(s) e visualizar o(s) display(s) já existente(s) no projeto

## Execução da Aplicação

O menu *Run* permite o acesso a todas as funcionalidades de execução do projeto. Os itens de menu disponíveis neste ambiente são: Buid, Test, Startup, Publish e o módulo Tools. A Figura 5-3 ilustra esse menu.

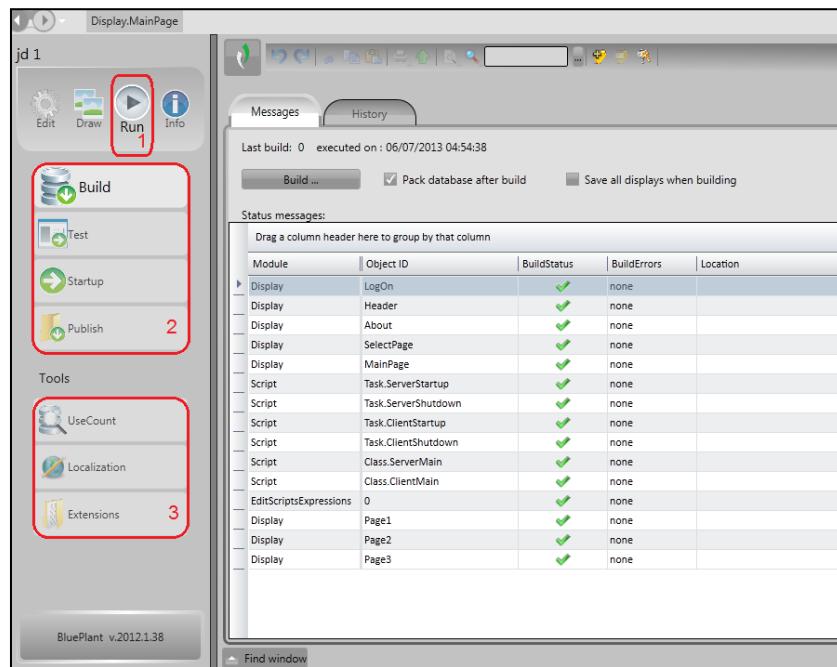


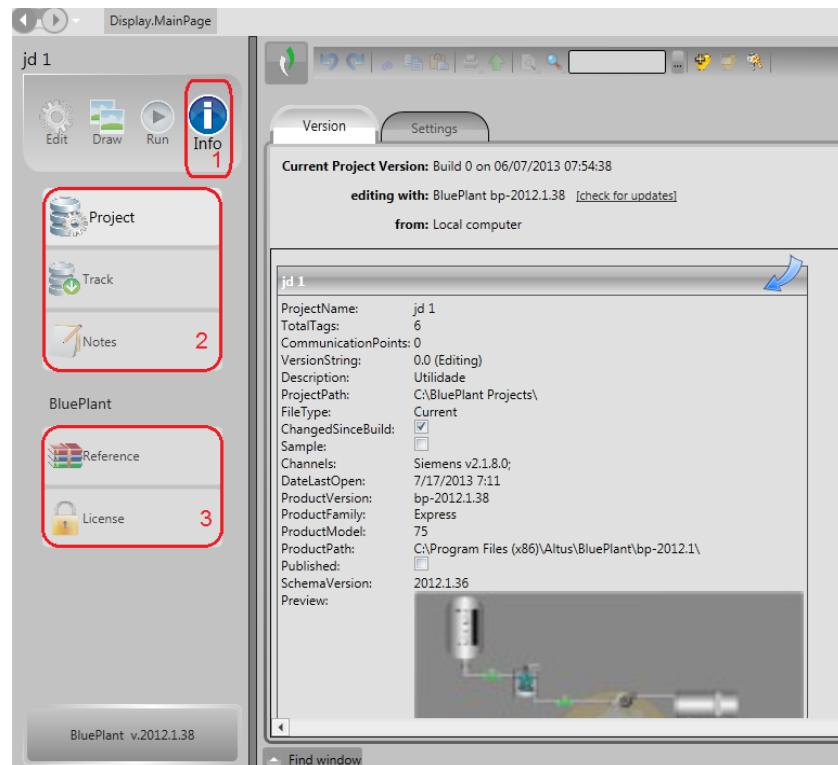
Figura 5-3. Menu Run

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

1. Barra de seleção do menu de execução do projeto
2. Funcionalidades da execução do projeto
  - Comando *Build*, o qual prepara um projeto para execução e o salva no Histórico de Compilação do Projeto. Ele está dividido em Compilação de Mensagens e Compilação de Histórico. A Compilação de Mensagens exibe as informações de status, erros, localização e módulo. A Compilação de Histórico exibe as informações de compilação, erros, data da execução e usuários
  - Comando *Test*, contemplando as configurações associadas ao teste do projeto
  - Comando *Startup* para inicializar as configurações do projeto
  - Comando *Publish* que permite acessar as configurações da publicação do projeto
3. Módulo Tools, incluindo as funcionalidades contagem de tags e objetos, via comandos *UseCount* e *CrossReference*, configurações de tradução no Runtime (Localization) e ferramentas de execução e Add-ons (Extentions)

## Informações da Aplicação

O ambiente *Info* permite o acesso às informações do projeto. A Figura 5-4 ilustra esse menu.



**Figura 5-4. Menu Info**

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

1. Barra de seleção do menu de informações do projeto
2. Informações da aplicação, incluindo detalhamento das configurações do projeto (Project), rastreamento das alterações efetuadas (Track) e notas do usuário (Notes)
3. Informações do aplicativo onde é possível acessar a ajuda do software e obter informações sobre a licença de uso da ferramenta

## 6. Componentes do BluePlant

Este capítulo detalha 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
- Aplicativos BluePlant: telas em outros sistemas operacionais e serviço no Windows

### 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 6-1. Funcionalidades do Menu de Edit**

Nas próximas seções serão detalhadas essas funcionalidades.

#### Editando Tags (Tags)

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

#### Edição de Objetos Tag (Objects)

Use os tags (e suas propriedades pré-definidas) incluídos 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*.



### Nota:

*Tag* no contexto de uma configuração de projeto refere-se a uma variável de processo. A Figura 6-1 ilustra a edição de objetos Tags.

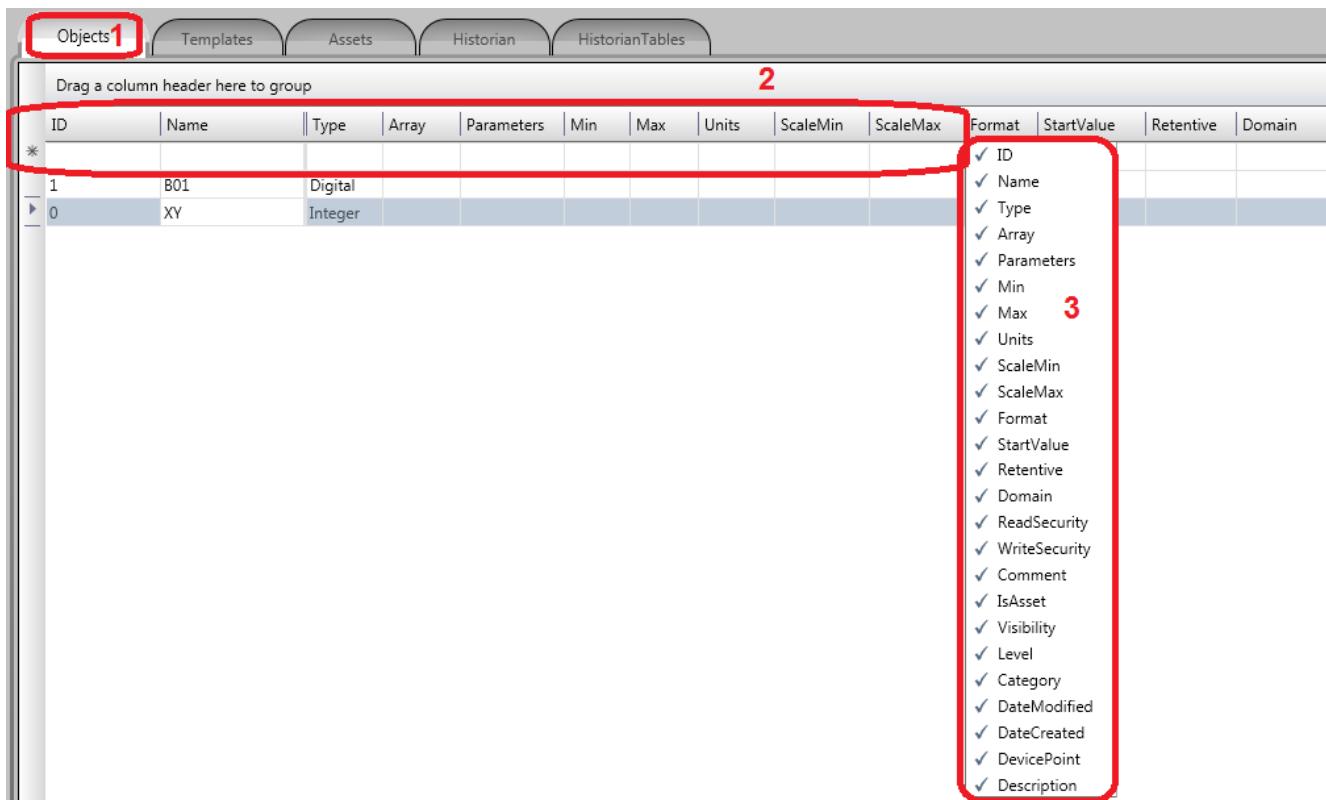


Figura 6-1. Edição de objetos do Tag

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 ao novo Tag da aplicação na linha marcada com um asterisco (\*);
  3. Selecionar as opções de configuração dos 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 6-1.

Name \_\_\_\_\_

Define o nome do 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 do 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 6-2. Tipos de Disponíveis para Tags

**Notas:**

**Integer:** Variável do tipo inteiro de 32 bits cujo intervalo de valores é -2147483648 até 2147483647.

**Long:** Variável do tipo inteiro de 64 bits cujo intervalo de valores é -9223372036854775808 até 9223372036854775807.

**Double:** Variável do tipo float com os seguintes intervalos de valores:

- Valores negativos: -1.79769313486231570E+308 até -4.94065645841246544E-324
- Valores positivos: 4.94065645841246544E-324 até 1.79769313486231570E+308

**Decimal:** Variável do tipo decimal com os seguintes intervalos de valores:

- Sem ponto decimal: 0 até +/-79228162514264337593543950335
- Com ponto decimal: 0 até +/-7.9228162514264337593543950335 com 28 casas para a direita do ponto decimal

O tipo Decimal foi criado em .NET para permitir o cálculo com maior precisão do que o tipo Double, o que se faz necessário quando se lida com números altos e nenhum arredondamento é permitido, como cálculos envolvendo dinheiro ou medidas de engenharia. Porém as operações matemáticas em um Decimal podem ser mais lentas, do que um Double.

**Text:** Variável do tipo string que aceita de 0 a 2 bilhões de caracteres Unicode.

**Timer:** É um valor inteiro de 32 bits que é gerado automaticamente pelo sistema para implementar vários tipos de temporizador que podem ser definidos na coluna Parameters:

- SquareWave: uma onda quadrada é gerada alterando o valor do tag entre 0 e o seu valor atual, o período dos pulsos é definido no parâmetro *Interval*.
- Pulse: o valor do tag permanece em zero, porém no fim de cada período, definido no parâmetro *Interval*, é gerado um pulso com o valor definido em *StartValue*.
- DelayOff: durante o Runtime, se você definir o tag com algum valor diferente do *StartValue*, o Tag vai segurar este valor pelo período de tempo especificado no parâmetro *Interval*. Passado o período do *Interval* o Tag volta ao seu *StartValue*.
- Comparer: O tag assume o valor configurado em *StartValue* após passar o tempo configurado no parâmetro *Interval*, só voltando a zero após a meia noite (00:00:00).

Caso o parâmetro *StartValue* não esteja definido o valor adotado é 1.

**Reference:** Um objeto de referência deve ser inicializado para apontar para outro objeto. Isso normalmente é feito usando a seguinte sintaxe no corpo do script:

```
@Tag.Reference1.Link = @Tag.TagName.GetName();      (VB)
@Tag.Reference1.Link = @Tag.TagName.GetName();      (C#)
```

O tipo do tag que o Tag de Referência aponta é definido na coluna Parameters. Todos os Tags de Referência tem um atributo adicional em Runtime que é o *link*, que contém, em tempo de execução o Tag para o qual a Referência está apontando.

## Array

Quando não definido (em branco) o 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 um Tag Array de tamanho "5", a matriz é criada a partir do Tag[0] ao 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.

### Nota:

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 do tag de acordo com o tipo: *banda morta* para tags analógicos e *ajustes* para tags do tipo *contador* e *temporizador*. Quando criando *Tag de referência* (ou PONTEIROS) define o tag de referência de destino na coluna *Tipo de referência*.

## Min

Configura o valor mínimo admissível para o 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 o 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. Para formatos numéricos válidos consulte o tópico Formatos Numéricos Padronizados. Exemplo: N1 (número com uma casa decimal). Para formatos de data e hora válidos consulte o tópico Formatos de Data e Hora Padronizados. Exemplo: 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 6-3. 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 6-4. Exemplo de Formatos DateTime**

Exemplo: padrão de tempo no formato longo

Formato = HH:mm:ss (hora;minuto;segundo)

**Start Value**

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

**Retentive**

Especifica se as Propriedades e o Valor do 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 do Tag não são salvos (não retentivo)
- *ValueOnly*: somente o valor do Tag é salvo quando modificado e pode ser usado como valor de inicialização na próxima execução
- *Properties*: todas as propriedades do Tag são salvadas, incluindo o valor, quando este é modificado para serem usadas na próxima inicialização
- *PropertiesOnly*: as propriedades do Tag são salvadas, mas o valor não

Ao aplicar propriedades retentivas, as modificações são salvadas 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.

**Nota:**

Durante a edição é necessário pressionar a tecla ENTER na célula da tabela para a confirmação das configurações efetuadas.

**Domain**

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

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

**Nota:**

Tags do tipo cliente não devem ser usados 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 dos 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 do Tag.

**WriteSecurity**

Define as permissões de escrita do Tag.

**Comment**

Define opcionalmente qualquer comentário sobre este Tag.

**Visibility**

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

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

**Level**

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

**Category**

Define uma ou mais categorias para este atributo. Consultar seção Categories.

**DevicePoint**

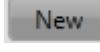
Informa o mapeamento desta tag em um driver de comunicação.

**Description**

Define opcionalmente qualquer descrição sobre este Tag.

***Edição de Tipos Definidos pelo Usuário (Templates)***

Um tag de tipo existente pode ser localizado no campo *User Custom Type*.

Clique  para apagar um tag de tipo estrutura existente e  para criar um novo tag de tipo estrutura.

Um tag de tipo do usuário pode ser usado da mesma forma que os tags incorporados.

**Nota:**

Nas versões *Express* e *Lite* os 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 um tag denominado *loop* do tipo PID, sendo que a sintaxe para acesso ao seu valor é:

```
loop.setpoint e loop.PV
```

A Figura 6-2 ilustra a edição de tipos definidos pelo usuário.

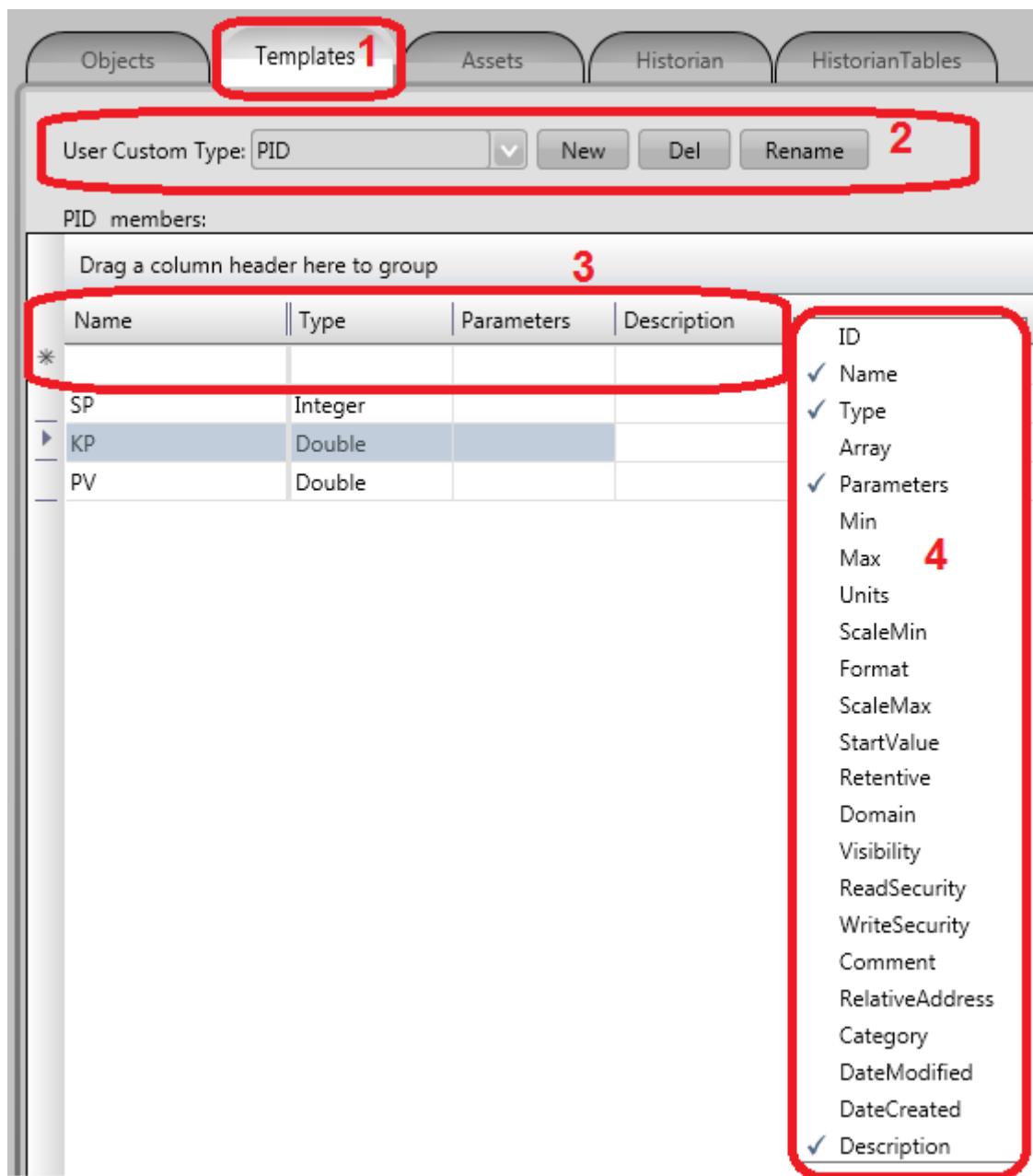


Figura 6-2. Edição de Templates Tag

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 6-2, possuem as mesmas características dos campos da aba *Objects*, já explicados na seção **Edição de Objetos Tag**.

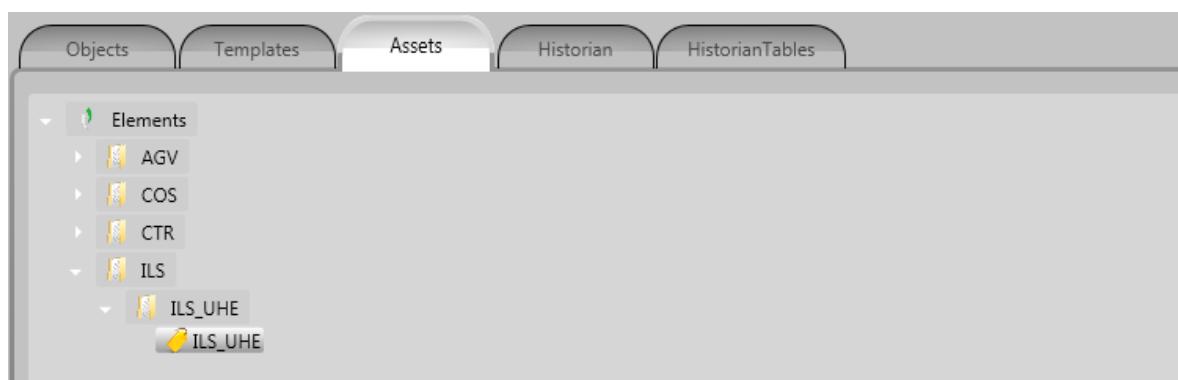
### Edição de Níveis (Assets)

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 relacionados 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 coisas com, UHEs, Subestações, Setores de tensão.

Define-se na aba Níveis(Assets) do espaço de trabalho dos 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*.
  - o 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 O objeto torna-se um filho do nível selecionado.
7. Continue adicionando níveis filho ou irmão e inserção de ativos, conforme a necessidade.
  - o 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 novos ou já existentes, selecione o novo nível na coluna *Level*.

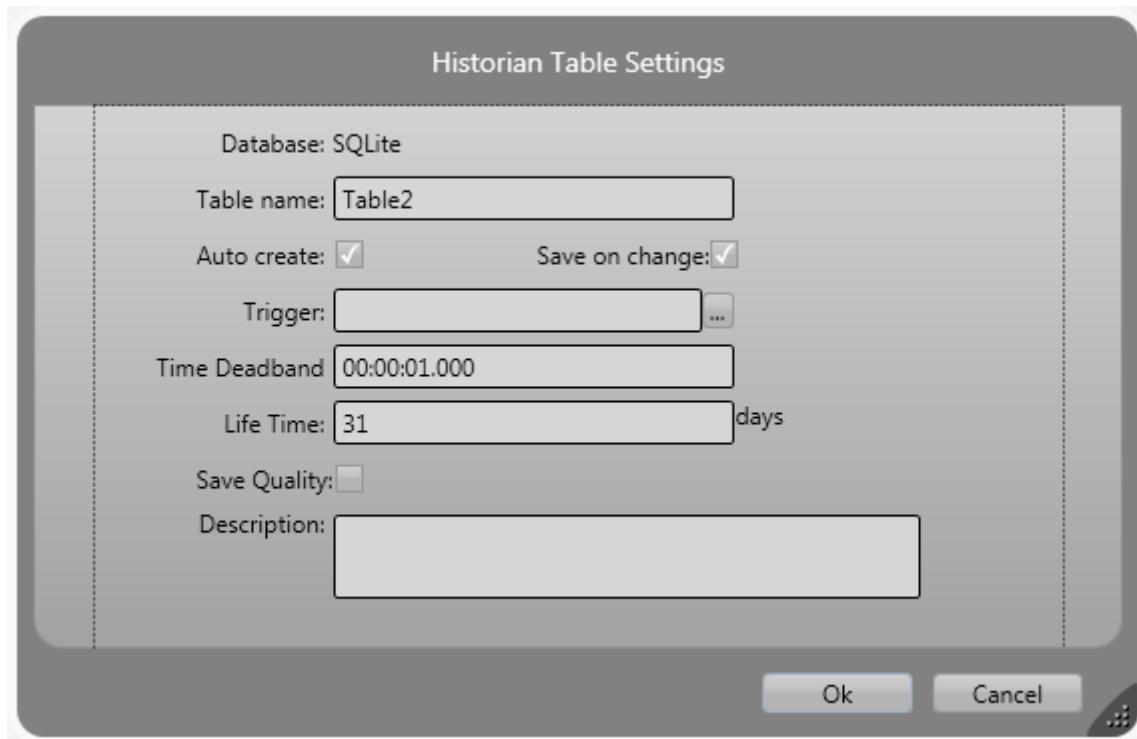


**Figura 6-3. Edição de Nível (Assets)**

### Edição de Historiador (Historian)

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

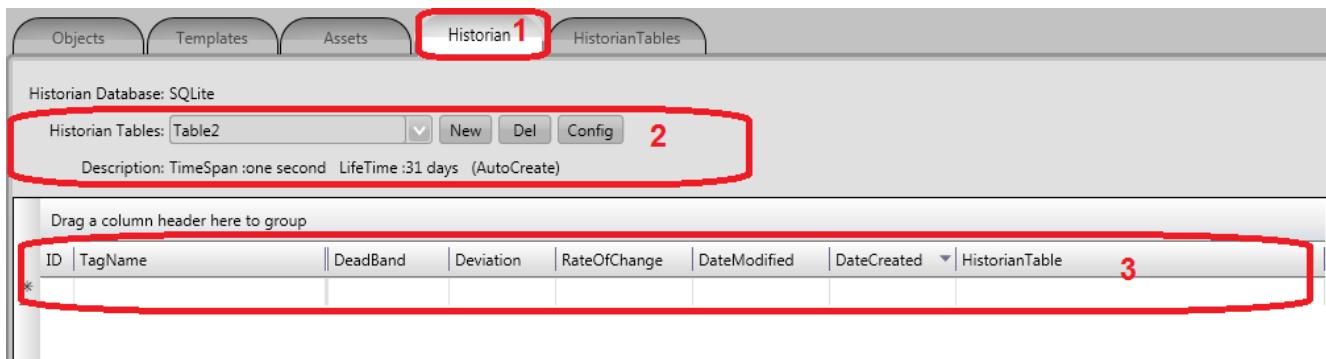
Clique **Del** para apagar uma Tabela de Historiador existente ou clique **New** para configurar uma nova Tabela de Historiador. A Figura 6-4 mostra as configurações da Tabela do Historiador.



**Figura 6-4. Configurações da Tabela do Historiador**

O banco de dados onde os tags são salvos é definido em `EditDatasetDBs` no DB chamado `TagHistorian`.

A Figura 6 4 ilustra as configurações associadas ao historiador de tags.



**Figura 6-5. 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 mostradas na Figura 6-4
3. Digitar os dados relativos ao 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 6-5.

**TagName**

Indica o nome do 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 o 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 o 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 o 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 deste Tag nesta tabela.

**DateCreated**

Registro da data da inserção deste Tag nesta tabela.

**Historian Table**

Define a tabela do banco de dados onde os Tags do Historiador são salvos.

**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 na Figura 6-4, 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 um tag antes de gerar um novo registro para o mesmo 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 dos tags associados 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 as Configurações de Segurança (Security)

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

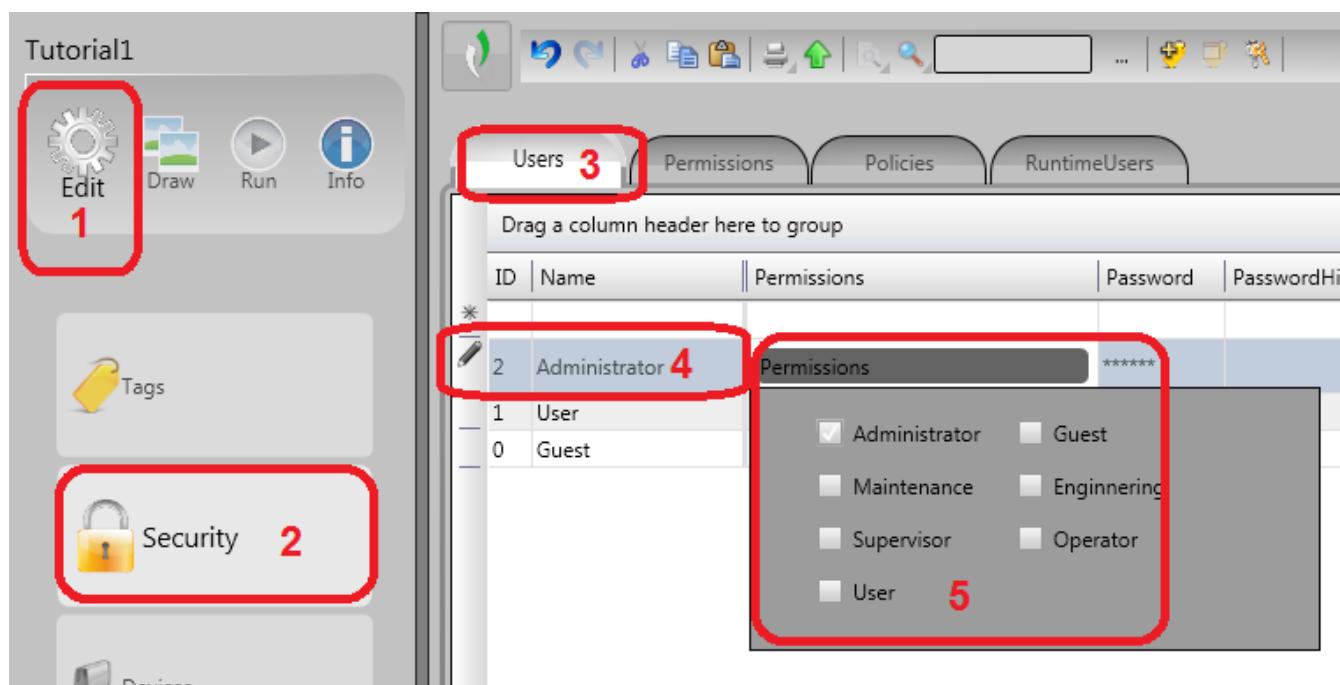
#### Edição de Usuários (Users)

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 6-6 mostra a edição dos perfis dos usuários no âmbito da segurança.



**Figura 6-6. Edição de Perfis de Usuários**

Os passos para edição/inclusão de perfis de usuários estão numerados na Figura 6-6 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**

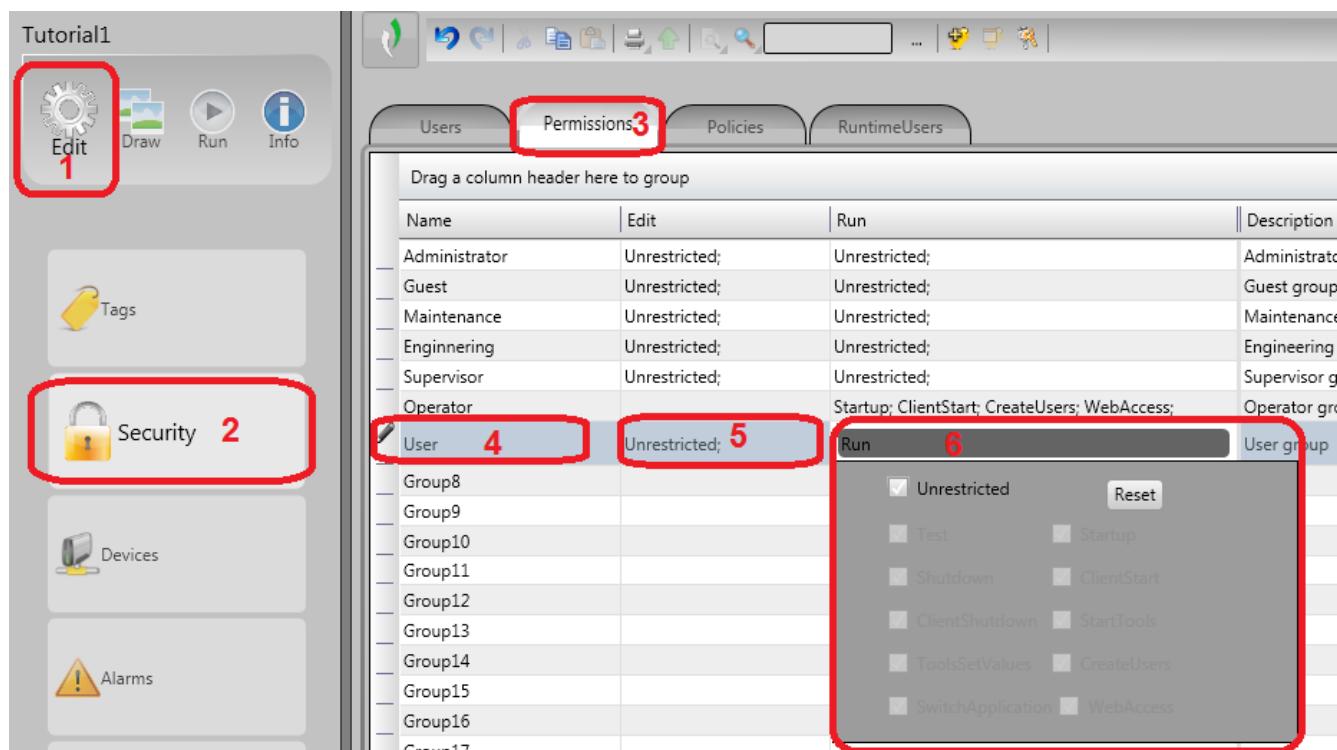
É possível configurar uma senha e uma dica da senha para cada usuário.

Os campos de configuração desta aba são:

- Name: nome do usuário
- Permissions: seleção do grupo de permissões para este usuário, para mais informações ver **Edição de Permissões**
- Password: definição da senha para o usuário, para mais informações sobre configuração de senhas ver **Edição de Políticas**
- PasswordHint: onde se insere a ajuda de senha
- Policy: seleção da opção de política de uso para este usuário, para mais informações sobre configuração de políticas ver **Edição de Políticas**
- 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 6-7 ilustra essa seleção.



**Figura 6-7. Edição de Permissões de Segurança**

Os passos para editar as permissões existentes estão ilustrados na Figura 6-7 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:**

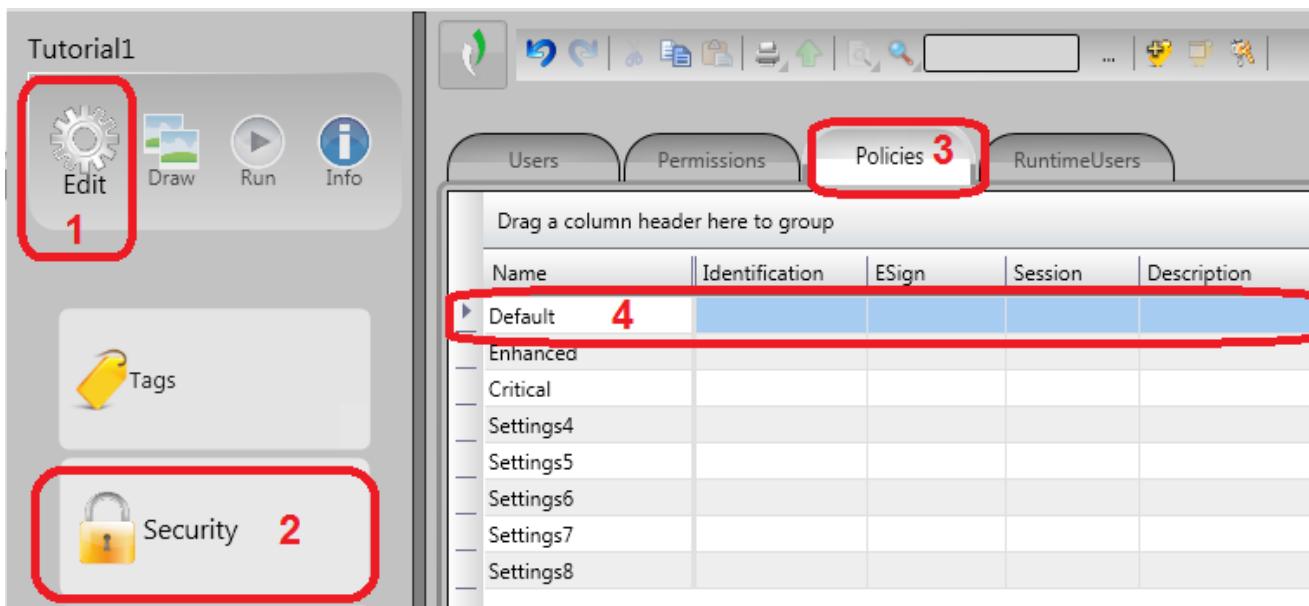
A configuração das permissões de edição, serve para que os usuários configurados com esta permissão possam realizar determinados tipos de modificações no projeto. As configurações das permissões do Runtime servem para que os usuários configurados com esta permissão possam realizar determinadas ações 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)
  - 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)
  - ToolsSetValue (valores de ajuste das ferramentas de inicialização)
  - CreateUsers (criação de usuários)
  - SwitchApplication (alternar aplicação)
  - 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 6-8 ilustra essa seleção.



**Figura 6-8. Edição de Políticas de Segurança**

Os passos para editar as políticas de segurança existentes estão ilustrados na Figura 6-8 e descritos a seguir:

1. Selecionar o menu *Edit*
2. Acessar o item *Security*
3. Clicar na guia *Policies* para acessar as políticas de permissões do sistema

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.

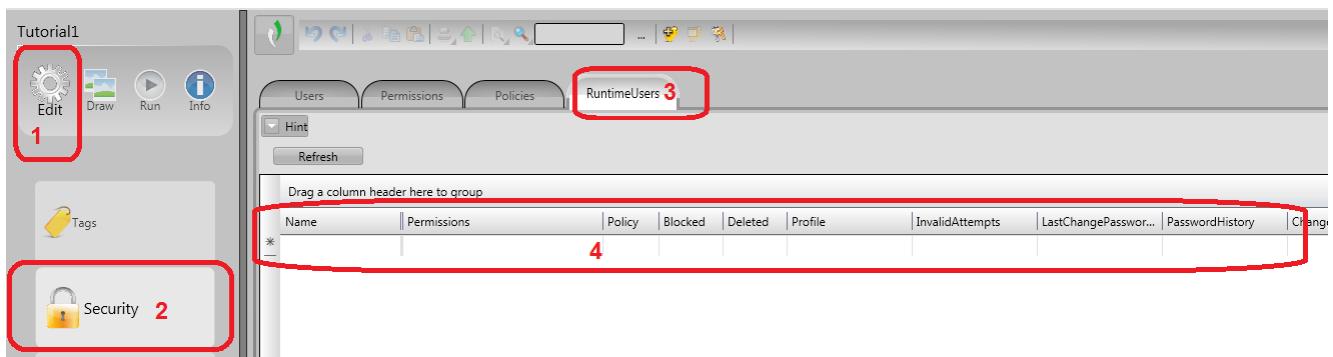


Figura 6-9. RuntimeUsers

Os passos para visualizar os usuários criados em execução são:

1. Selecionar o menu *Edit*
2. Acessar o item *Security*
3. Clicar na guia *RuntimeUsers* e clicar no botão *Refresh*
4. Será exibido todos os usuários bem como suas configurações.

### Editando Dispositivos (Devices)

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 6-10 ilustra essa seleção.

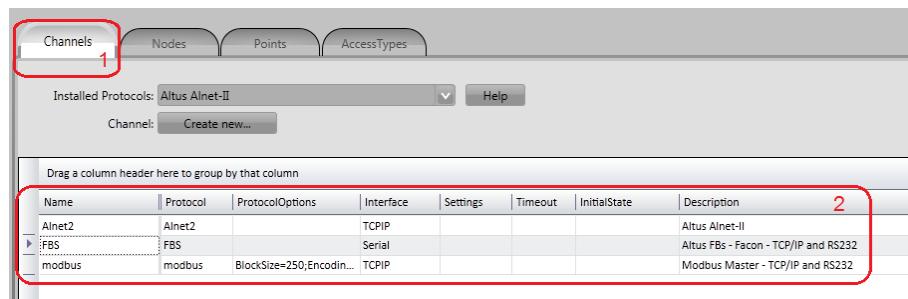


Figura 6-10. Edição de Canais de Comunicação

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 6-10.

#### 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:

- 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

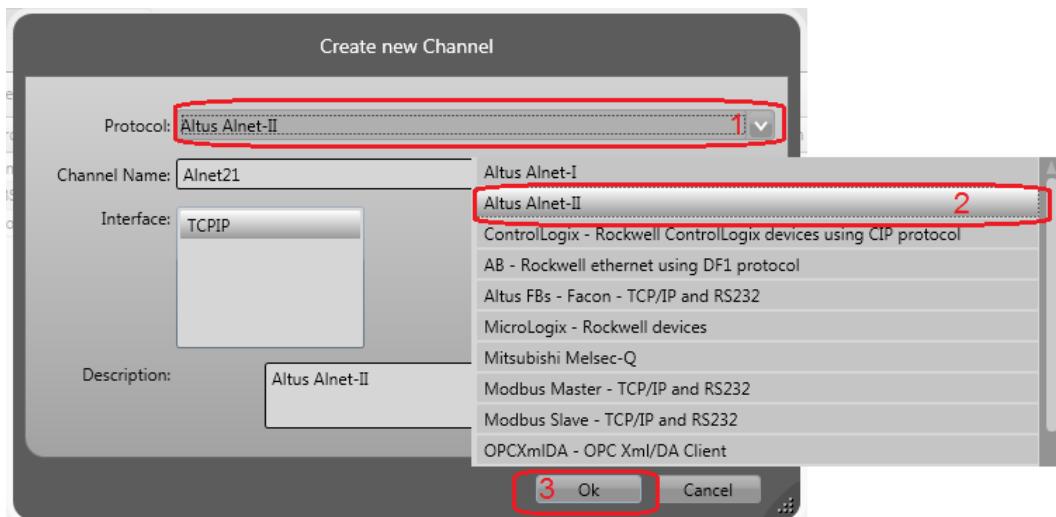
Define o endereço IP primário e backup dos servidores nos quais serão executados o canal.

#### 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 6-9 ilustra essa seleção.



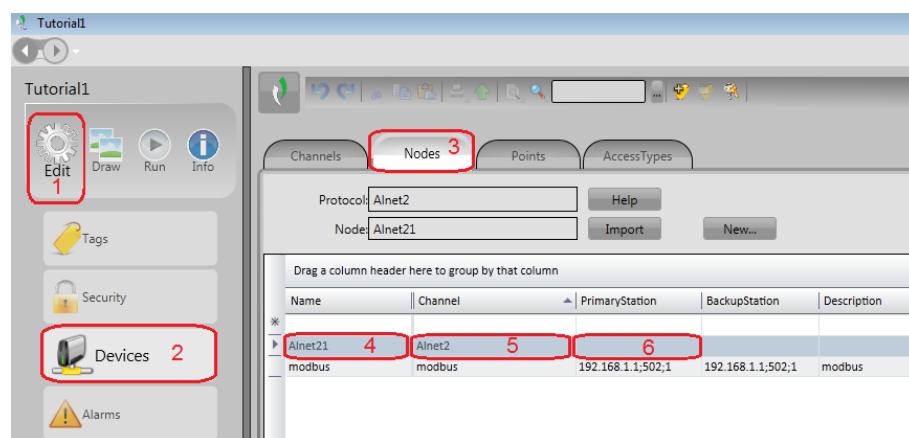
**Figura 6-11. Edição de um Novo Canal de Comunicação**

Na sequência será explicado o passo a passo para a criação de um novo canal de comunicação conforme exemplo ilustrado na Figura 6-11.

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 6-12 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 6-12. Edição do Nô para o Canal de Comunicação**

Os passos para editar ou incluir os nós existentes estão ilustrado na Figura 6-12 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.

Name	Channel	PrimaryStation	BackupStation
Node1	Alnet	1;0	1;0

DestNode: 1  
DestSubNet: 0  
OriginNode: 0  
OriginSubNet: 0

**Figura 6-13. 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 ultima sincronização executada pelo protocolo, válido para OPC.

#### SyncStation

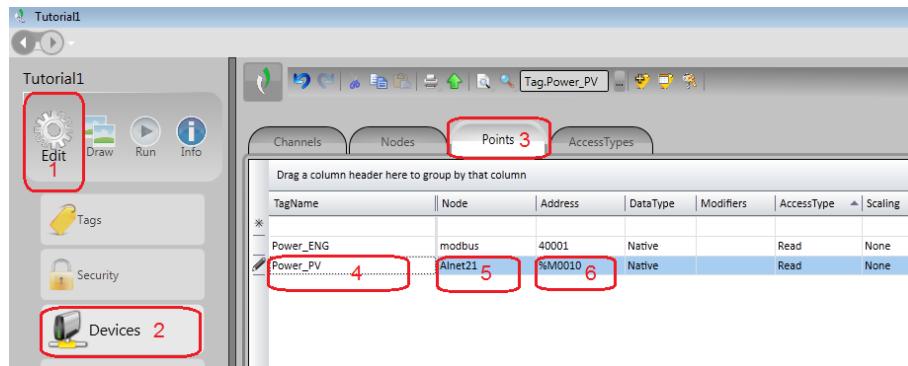
Apresenta as configurações da estação que executou a sincronização de dados com este nó, válido para OPC.

#### 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 nos tags.



**Figura 6-14. Edição de Pontos**

Os passos para editar ou incluir os pontos de comunicação estão ilustrado na Figura 6-14 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

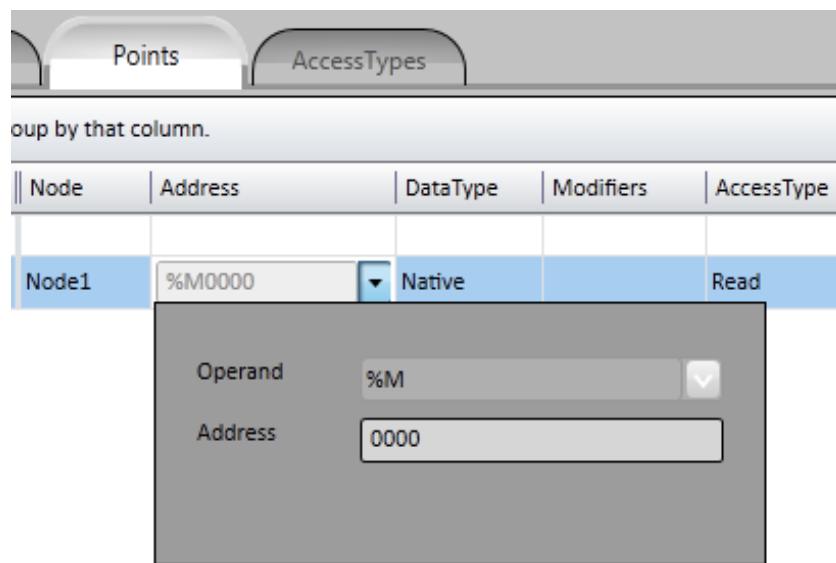
Define o TagName a ser lido ou escrito no Dispositivo identificado.

#### 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 6-15 ilustra um exemplo de parametrização do endereço (operando) no caso do protocolo ALNET.



**Figura 6-15. 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)
- UInt (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)

**Nota:**

**Compatibilidade de variáveis com Mastertool IEC XE:** A tabela que segue apresenta a compatibilidade entre os tipos de tags do BluePlant e as variáveis do Mastertool IEC XE, no caso do uso de protocolos de comunicação com a opção *Native* para os tipos de pontos.

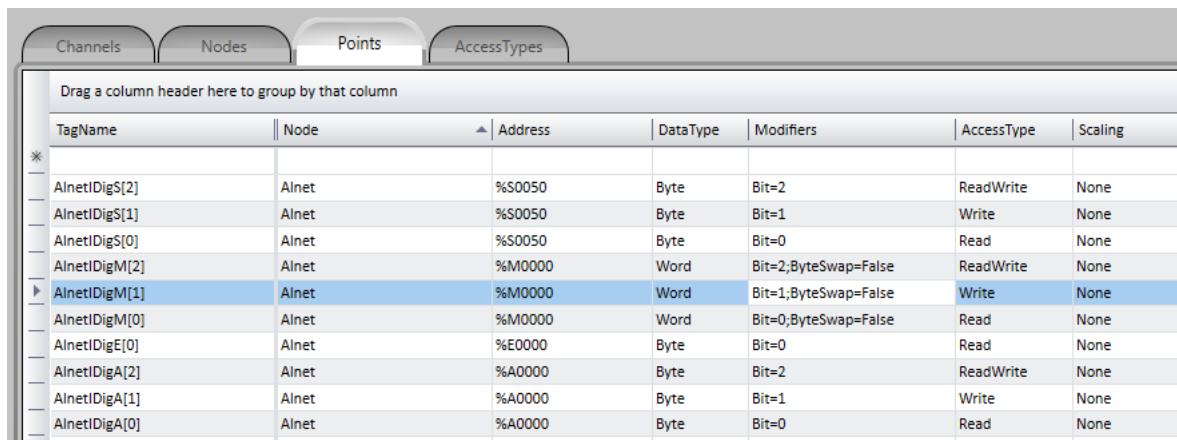
Variáveis do Mastertool IEC XE	Tipo de Tags do BluePlant	Tipo de DataType do BluePlant	Modifiers
<b>BOOL</b>	Digital	Native	
<b>BIT</b>	Digital	Native	
<b>BYTE</b>	Integer	Byte	
<b>SINT</b>	Integer	Native	
<b>USINT</b>	Integer	Native	
<b>WORD</b>	Integer	Native	
<b>INT</b>	Integer	Short	
<b>UINT</b>	Integer	Native	
<b>DWORD</b>	Double	Dword	WordSwap=True;
<b>DINT</b>	Integer	Dword	WordSwap=True
<b>UDINT</b>	Double	Dword	WordSwap=True;
<b>LWORD</b>	Long	ULong	WordSwap=True; DWordSwap=True
<b>LINT</b>	Long	Long	WordSwap=True; DWordSwap=True
<b>ULINT</b>	Long	Long	WordSwap=True; DWordSwap=True
<b>REAL</b>	Double	Single	WordSwap=True
<b>LREAL</b>	Double	Real	WordSwap=True; DWordSwap=True
<b>STRING</b>	Text	Native	
<b>WSTRING</b>	Text	Native	
<b>TIME</b>	Timer	Native	
<b>LTIME</b>	Double	Native	
<b>TIME_OF_DAY</b>	Double	Native	
<b>DATE</b>	DateTime	Native	
<b>DATE_AND_TIME</b>	DateTime	Native	

**Tabela 6-5. Compatibilidade entre os Tags do BluePlant e as Variáveis do Mastertool IEC XE**

#### 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.



TagName	Node	Address	DataType	Modifiers	AccessType	Scaling
AlnetDigS[2]	Alnet	%S0050	Byte	Bit=2	ReadWrite	None
AlnetDigS[1]	Alnet	%S0050	Byte	Bit=1	Write	None
AlnetDigS[0]	Alnet	%S0050	Byte	Bit=0	Read	None
AlnetDigM[2]	Alnet	%M0000	Word	Bit=2,ByteSwap=False	ReadWrite	None
AlnetDigM[1]	Alnet	%M0000	Word	Bit=1,ByteSwap=False	Write	None
AlnetDigM[0]	Alnet	%M0000	Word	Bit=0,ByteSwap=False	Read	None
AlnetDigE[0]	Alnet	%E0000	Byte	Bit=0	Read	None
AlnetDigA[2]	Alnet	%A0000	Byte	Bit=2	ReadWrite	None
AlnetDigA[1]	Alnet	%A0000	Byte	Bit=1	Write	None
AlnetDigA[0]	Alnet	%A0000	Byte	Bit=0	Read	None

**Figura 6-16. 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 do string (Somente valido para tags de formato Text).

**AccessType**

Define o comportamento de leitura e escrita para cada ponto. Para mais informações verificar **Tipo de Acesso**.

**Label**

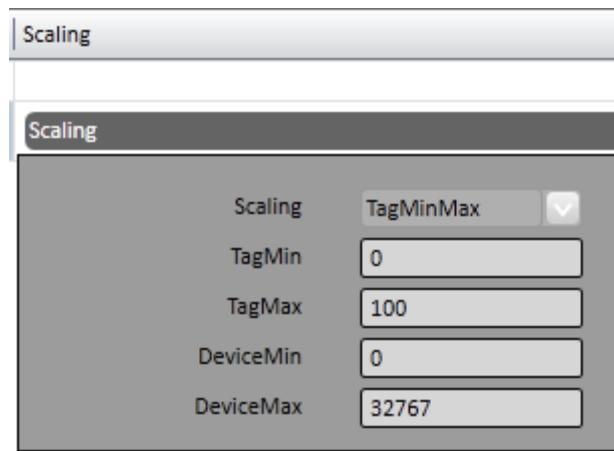
Adição de um rótulo para o ponto de comunicação.

**Scaling**

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

- None (nenhuma conversão)
- TagMinMax (escala baseada no valor mínimo e máximo do tag)
- Liner (escala linear)
- Equation (escala via equação linear)

A Figura 6-17 ilustra a parametrização de conversão de escala baseada na opção TagMinMax.

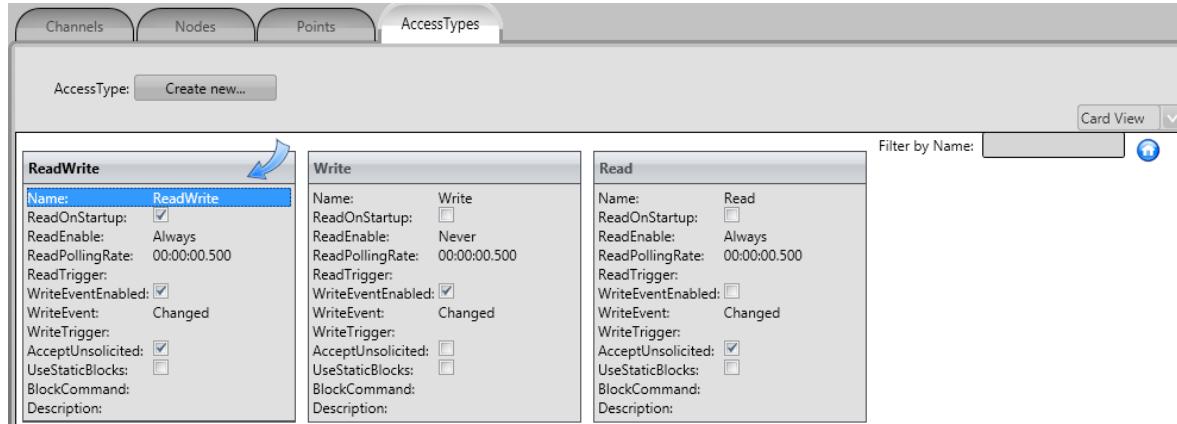


**Figura 6-17. Conversão de Escala**

#### *Tipo de Acesso (AccessType)*

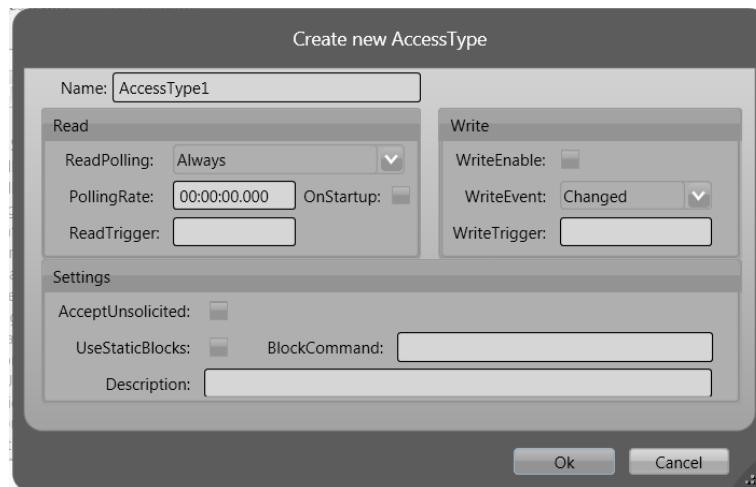
Define as características do tipo de acesso comuns aos pontos do dispositivo. Podem ser configurados os tipos de acesso que controlam a leitura, escrita e outras configurações quando o aplicativo acessa dados do PLC ou IED, sempre de acordo com as possibilidades existentes no protocolo em questão. Como pode ser visto na Figura 6-18, cujo modo de visualização selecionado é *CardView*, há três tipos de acesso padrão:

- **ReadWrite** (leitura e escrita)
- **Write** (escrita)
- **Read** (somente leitura)



**Figura 6-18. Tipo de Acesso**

Clique em **Create new...** para criar um novo AccessType.



**Figura 6-19. AccessType**

Na sequência são explicados os principais itens do menu de tipo de acesso conforme ilustrado na Figura 6-19.

#### 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 o Tag vinculado é alterado
- ChangedUp: escreve quando o Tag vinculado sofre um aumento de valor
- ChangedDown: escreve quando o Tag vinculado 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 (Alarms)

Um Alarme pode ser configurado através da associação de um 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 do 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)

#### Grupos

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

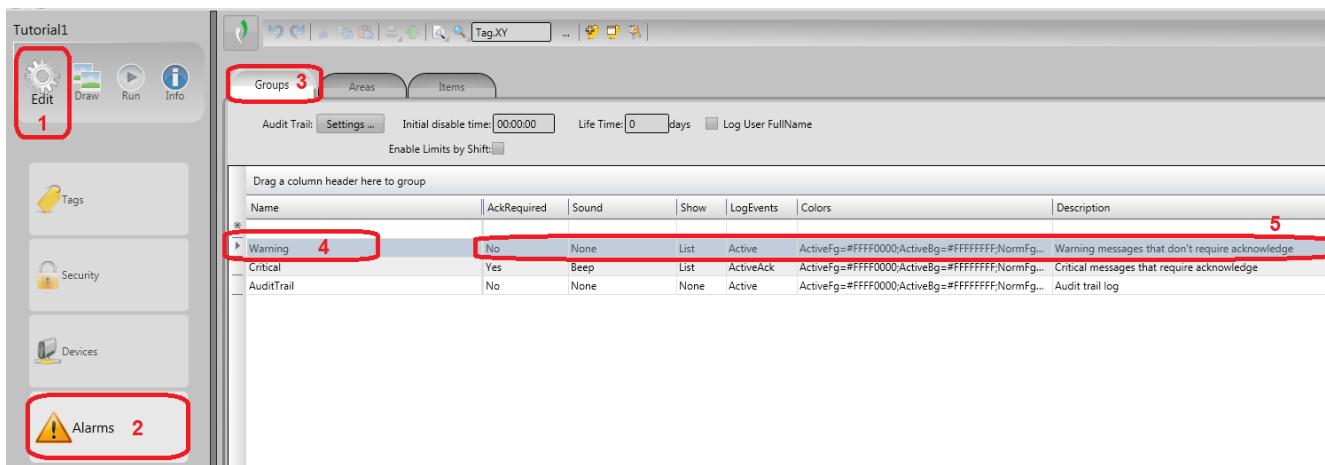


Figura 6-20. Grupos de Alarmes

Os passos para editar ou incluir os grupos de alarmes estão ilustrados na Figura 6-20 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 os tags são salvos 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.

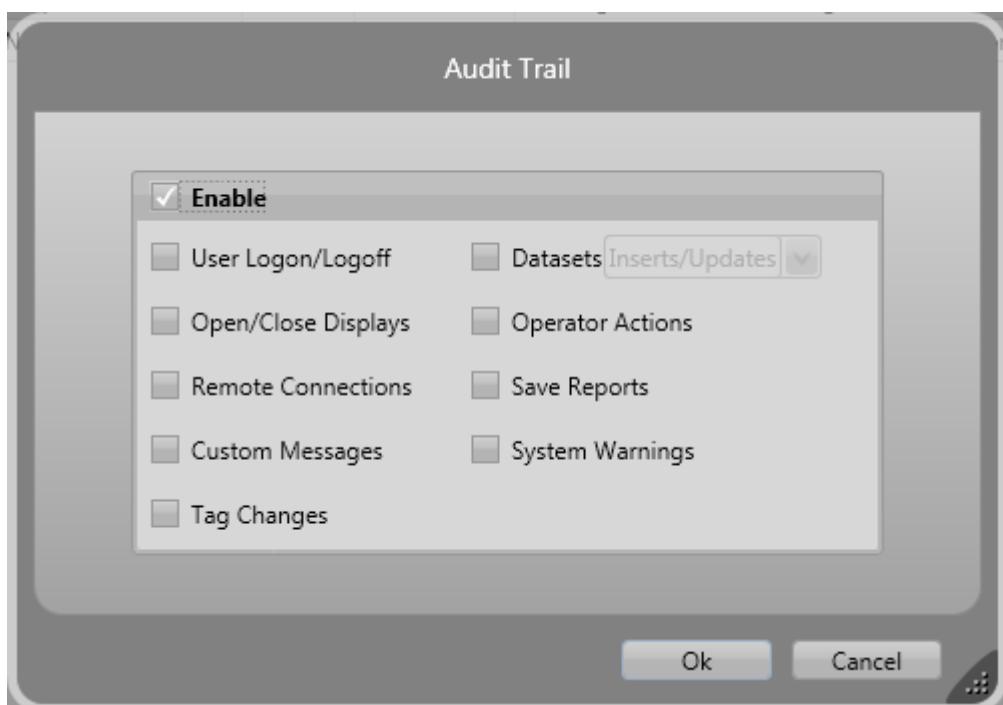
**NotificationMethod**

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 6-21. Audit Trail****Opções gerais**

*Initial Disable Time*: na parte superior da aba, digite quanto tempo o sistema deve esperar durante a inicialização antes de considerar que um estado de alarme deve ser um alarme.

- Durante a partida do sistema, os IEDs podem apresentar alarmes, por não ter sido inicializados. Esta configuração informa ao sistema para esperar pelo período de tempo configurado, para deixar que se complete a inicialização, antes de o sistema considerar alarmes.

*Life Time*: informa o tempo que o Historiador de alarmes deve manter os dados de alarme, em dias. Caso esteja com o valor zerado, nunca é limpado os dados do banco.

*Enable Limits by Shift*: é acionado em tempo de execução o mecanismo de escolha de conjuntos de limites por “turno”, para implementar por exemplo, limites diferenciados por períodos do dia. (carga leve, media, pesada).

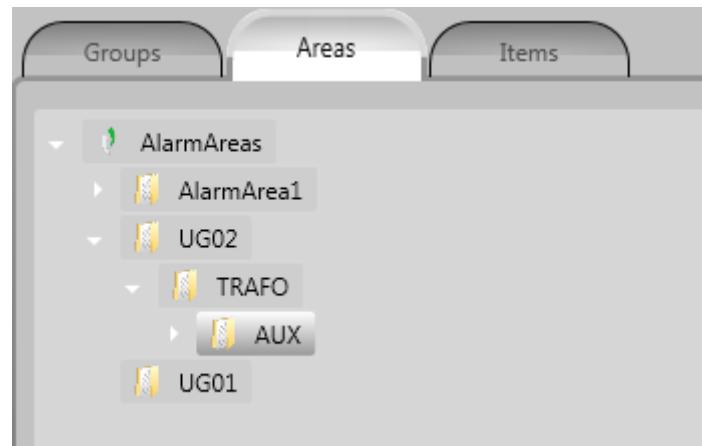
**Areas de Alarme (Areas)**

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:

1. Vá para *Edit>Alarms>Areas*.
2. 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 area.
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.

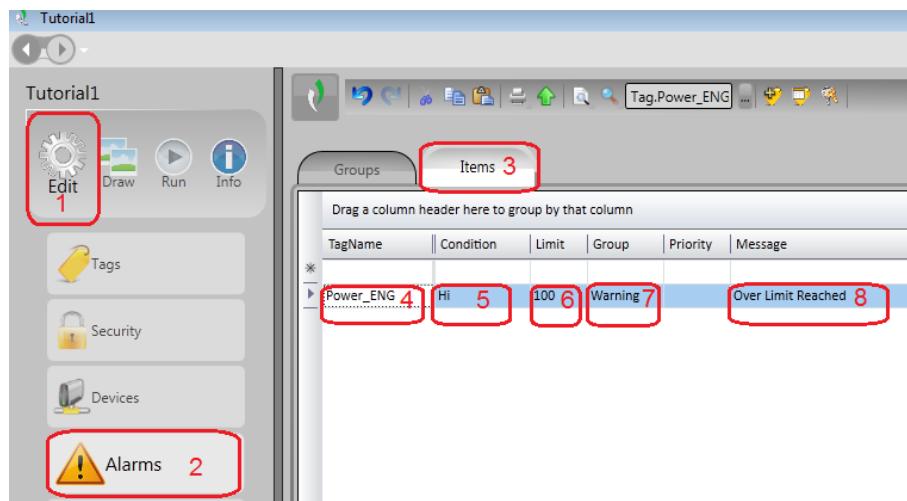
**Figura 6-22. Areas de Alarmes**

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 6-23. Itens de Alarmes**

Os passos para editar ou incluir os grupos de alarmes estão ilustrados na Figura 6-26 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 do 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 o *TagName* que será avaliado para gerar o Alarme.

#### Condition

Condição de avaliação para gerar alarmes. Opções...

- Hi: Tag  $\geq$  limite
- HiHi: Tag  $\geq$  limite (quando reconhecido automaticamente, reconhece o alarme Hi para o mesmo Tag)
- Lo: Tag  $\leq$  limite
- LoLo: Tag  $\leq$  limite (quando reconhecido automaticamente, reconhece o alarme Lo para o mesmo Tag)
- RateOfChange: Taxa de alteração do 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 >= limite
- LessThan: Tag < limite
- LessEqual: Tag <= limite
- Changed: valor do Tag foi alterado
- ChangedUp: valor do Tag aumentou
- ChangedDown: valor do Tag diminuiu
- NotEqual: valor do 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 do Tag desejado na mensagem usando a notação de vinculação XAML entre chaves, conforme ilustrado no exemplo a seguir:

```
ValordoTag = {NomedoTag.Value}
```

#### Editando Banco de Dados (Datasets)

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.

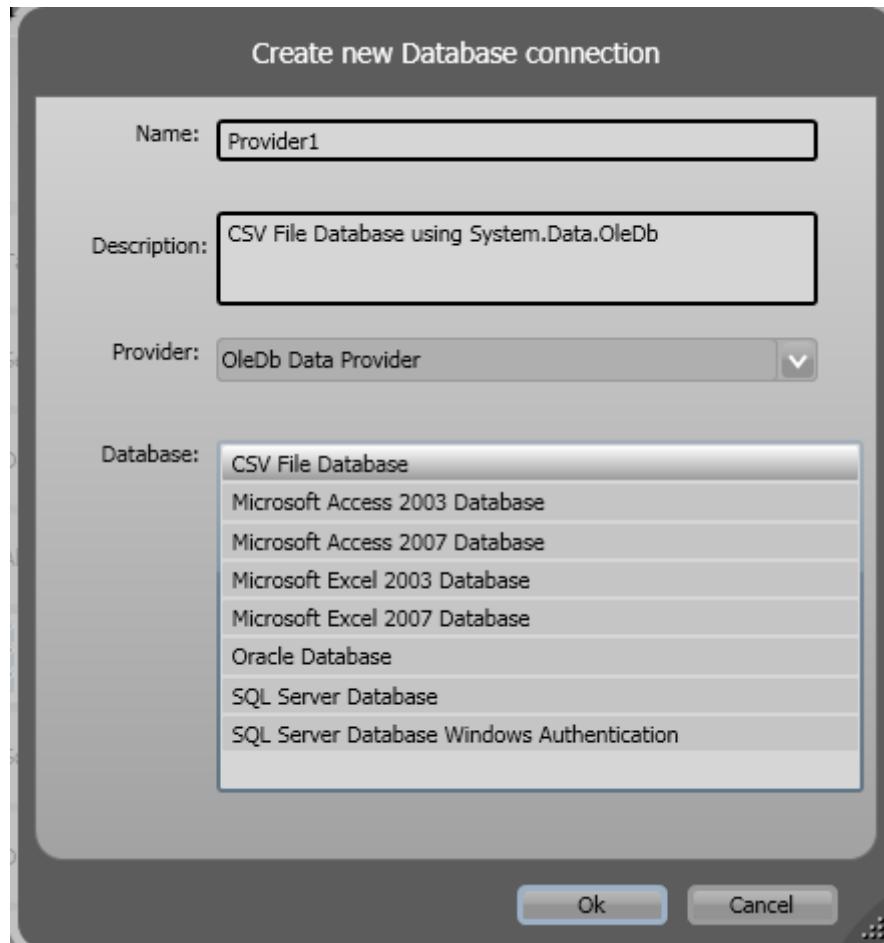
#### ATENÇÃO:

Consulte a seção de **Documentos Relacionados a este Manual** para obter informações adicionais quanto ao uso de banco de dados com o BluePlant.

**DBs**

Através da guia DBs menu *Edit > Datasets > BDs* é possível acessar a configuração do banco de dados. A partir do botão **Create new...** estão disponíveis as opções para criação de uma nova conexão, estas opções são apresentadas a partir de uma tela (Figura 6-24).

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 6-24. 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
- 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
- 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 **Test** 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.

#### Conexão com Bases de Dados

Nesta seção serão apresentadas as diferentes possibilidades de bases de dados a serem configuradas no BluePlant.

## Conexão com Excel

Conecta o Excel usando um driver ODBC, ODBC DSN ou OleDB. Para a conexão via ODBC deve-se proceder conforme segue:

1. Selecione e nomeie uma faixa de linhas e colunas na planilha. Isto permitirá ao BluePlant ler as informações como uma tabela
2. Escolha um dos seguintes processos de nomeação de acordo com a versão do Microsoft Excel

No Microsoft Office 2007: clique com o botão direito na seleção e escolha *Nomear uma Faixa*. A Figura 6-25 ilustra essa opção.

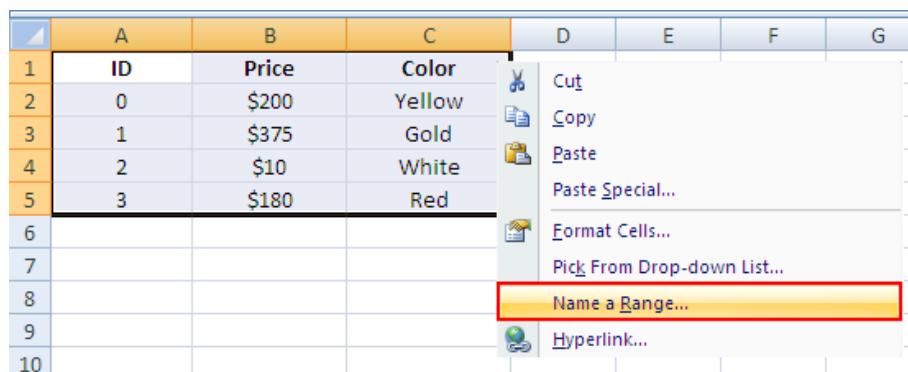


Figura 6-25. Conexão com Excel 2007

No Microsoft Office 2003: no Microsoft Excel vá para *Inserir > Nome > Definir...*. A Figura 6-26 ilustra essa opção.

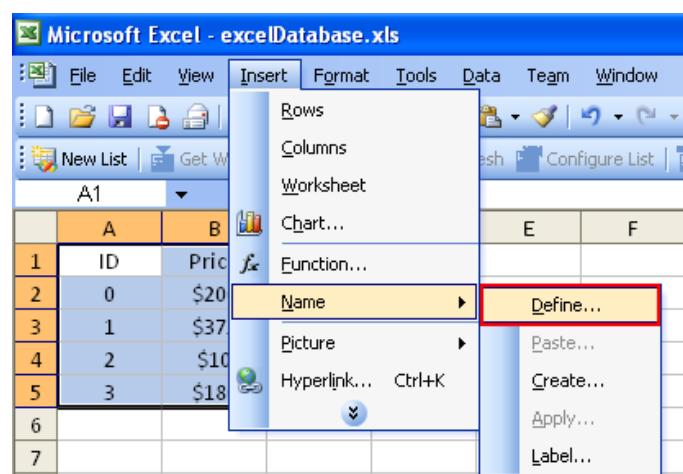
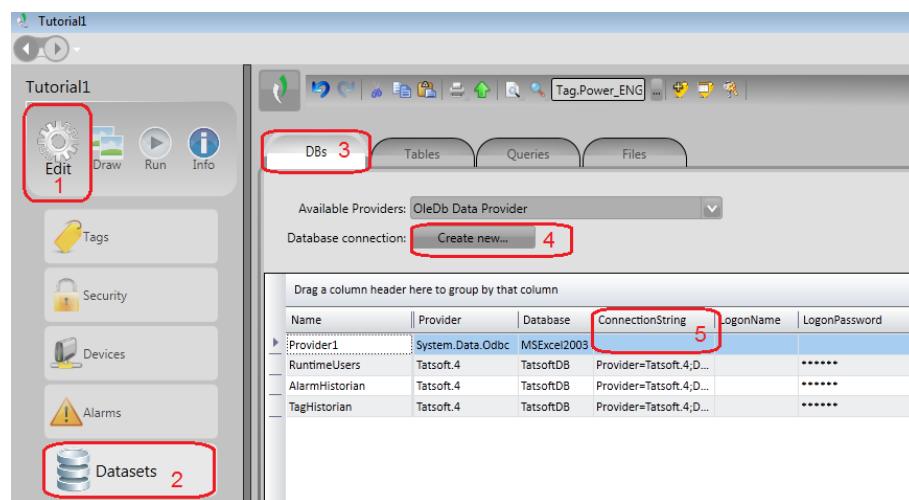


Figura 6-26. Conexão com Excel 2003

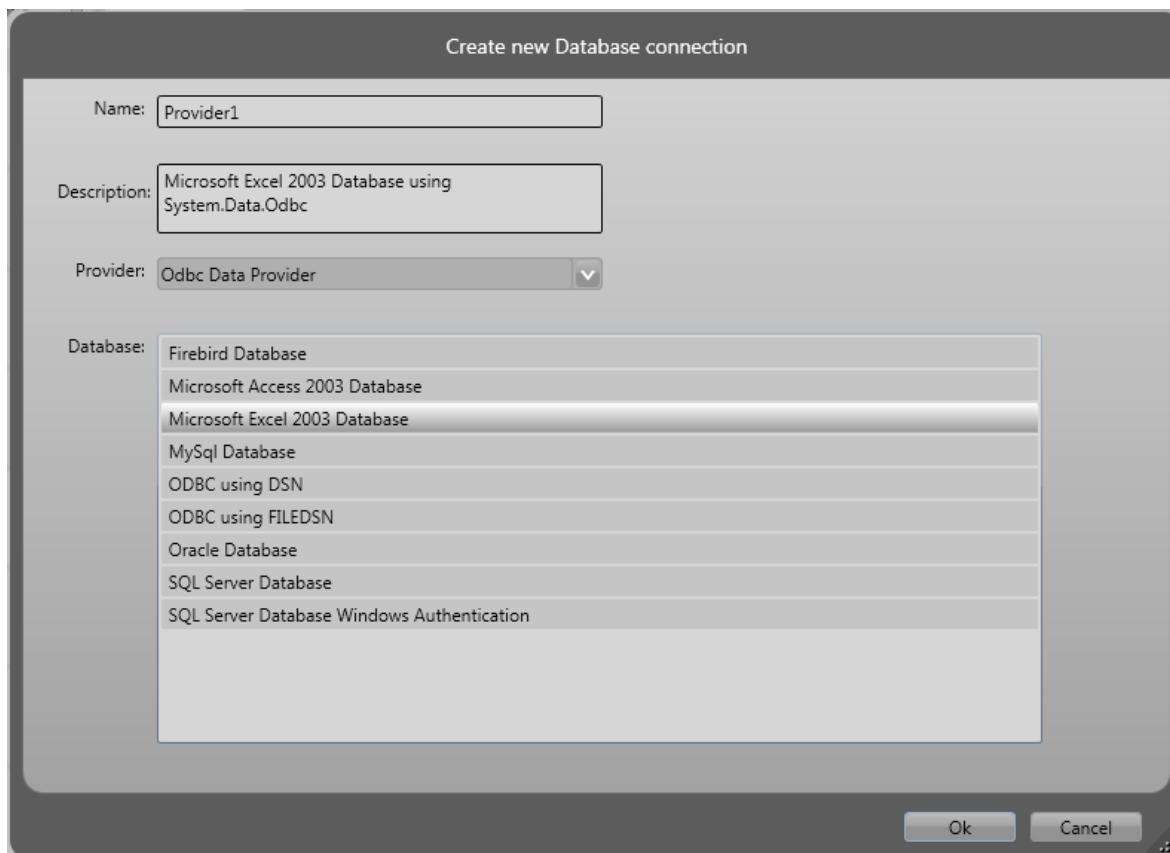
Dê um nome à seleção (por exemplo, *itemsTable*). O arquivo Excel está pronto para uso.



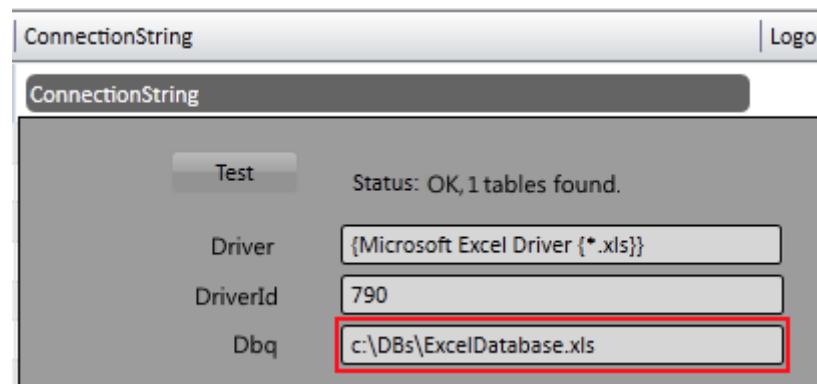
**Figura 6-27. Configuração da Conexão com Provider Excel 2003**

Os passos para criar uma nova conexão com banco de dados estão ilustrados na Figura 6-27 e descritos a seguir:

1. Selecionar o menu *Edit*
2. Acessar o item *Datasets*
3. Clicar na guia *DBs* para acesso as conexões criadas com bancos de dados
4. Botão para criar uma nova conexão com Excel 2003, conforme pode ser visto na Figura 6-28
5. Na coluna *ConnectionString* é onde se configura a conexão e teste com o banco de dados. Na Figura 6-29 pode ser visualizada a configuração e teste



**Figura 6-28. Criar uma Nova Conexão com Banco de Dados**



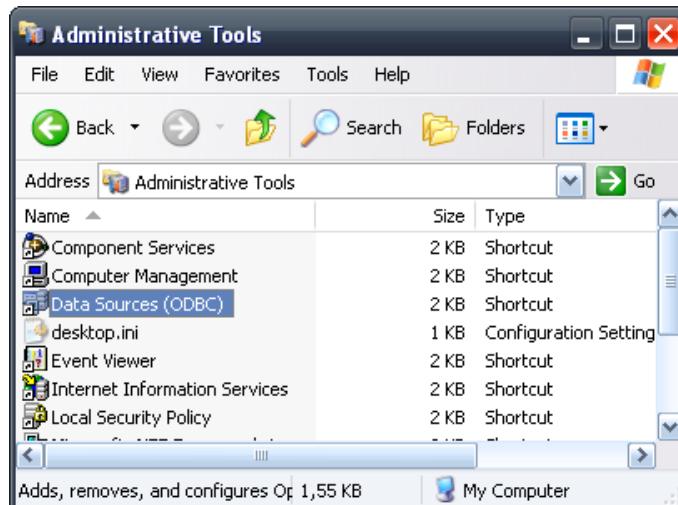
**Figura 6-29. Teste da Conexão com Banco de Dados**

**ATENÇÃO:**

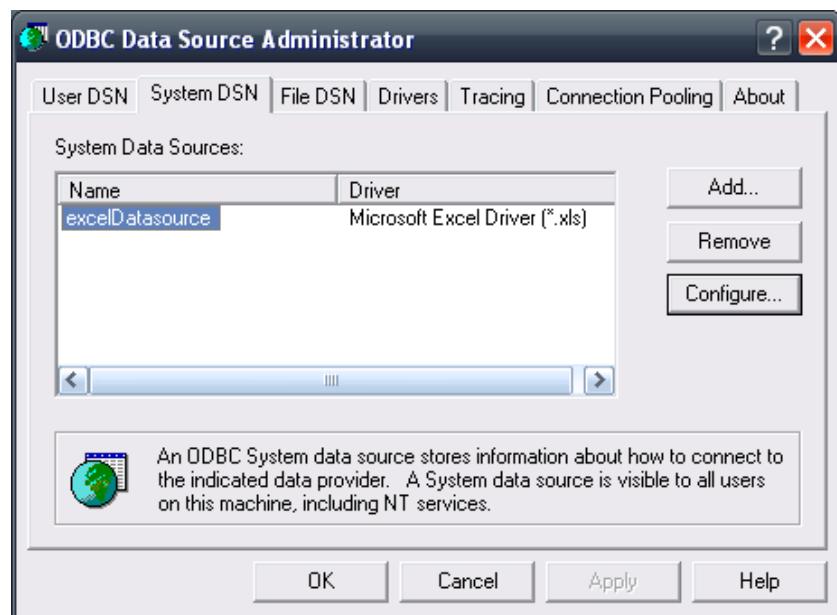
Para o correto funcionamento da conexão em sistemas operacionais de 64 bits o Microsoft Excel também deve ser 64 bits. Caso o sistema operacional seja 32 bits o Microsoft Excel deverá ser 32 bits.

Por outro lado, para configurar uma base de dados do tipo ODBC DSN proceda conforme indicado a seguir:

No painel de controle do computador selecione *Administrative Tools* e dê um clique duplo em *Data Sources (ODBC)*, conforme Figura 6-30. No caso onde o sistema operacional for Windows 64 bits, o acesso é através do arquivo C:\Windows\SysWOW64\odbccad32.exe.

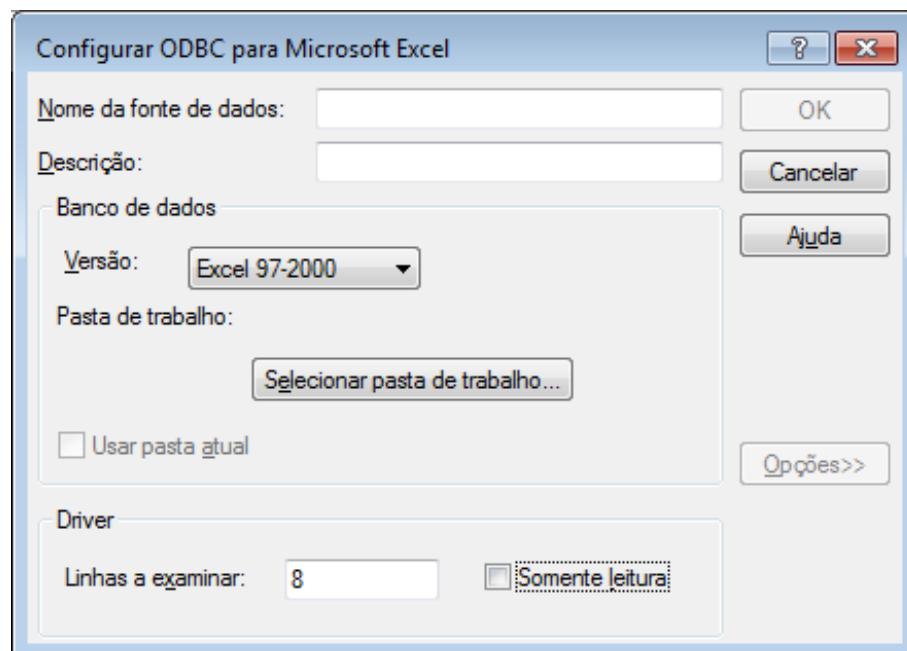


**Figura 6-30. ODBC com DSN**



**Figura 6-31. "ODBC Data Source Administrator"**

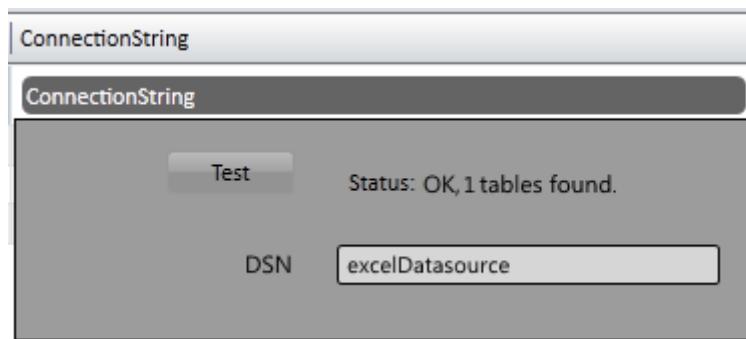
Na janela *ODBC Data Source Administrator* clique na opção de *Add...*. Será solicitado que seja selecionado um driver. Selecione o *Microsoft Excel Driver (\*.xls)* e seguinte tela deve aparecer conforme a Figura 6-32.



**Figura 6-32. Configuração da base de dados**

Clique em *Selecionar pasta de trabalho* e selecione o nome do arquivo Excel criado anteriormente. (por exemplo, "excelDatasource"). Para acesso de Escrita desmarque a checkbox *Somente leitura*.

No namespace do Datasets do BluePlant, escolher a guia *DBs* e criar um novo provider na opção correspondente. Nas opções do *OdBC Data Provider* escolha *ODBC using DSN* e clique em *Ok*. Na coluna *ConnectionString* da nova linha, entre com DSN no campo apropriado e clique no botão *Teste* para verificar a conexão com o Excel.

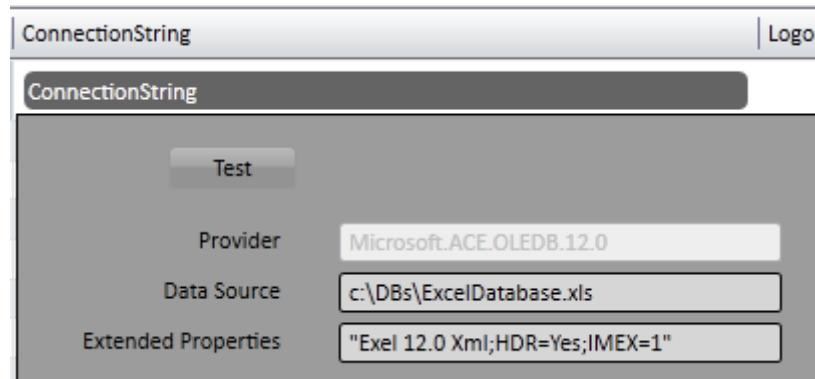


**Figura 6-33. Teste da Conexão à Base de Dados ODBC**

Para conectar o provider OLEDB com o Excel, pode ser feito através dos passos abaixo:

No namespace do Datasets do BluePlant, escolha a guia *DBs*. Selecione a opção *OleDb data provider* na caixa-combo e crie um novo provider na opção correspondente.

Escolha *Microsoft Excel Database* e clique em Ok. Clique na coluna *ConnectionString* do novo provider inserido, entre com o caminho e o nome do arquivo Excel (.xls) no campo *DataSource* conforme mostrado na Figura 6-34.



**Figura 6-34. Teste da Conexão à Base de Dados OLEDB**

#### Conexão com Banco de Dados Oracle

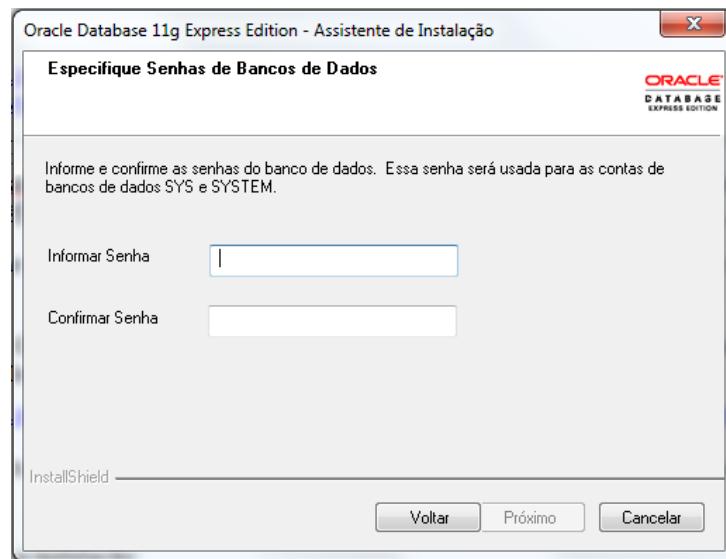
Este capítulo apresenta um exemplo de como criar uma base de dados Oracle.

#### Instalação e configuração do Banco de Dados Oracle

O Oracle utilizado no exemplo foi o Oracle Database Express Edition 11g Release 2, está disponível gratuitamente em:

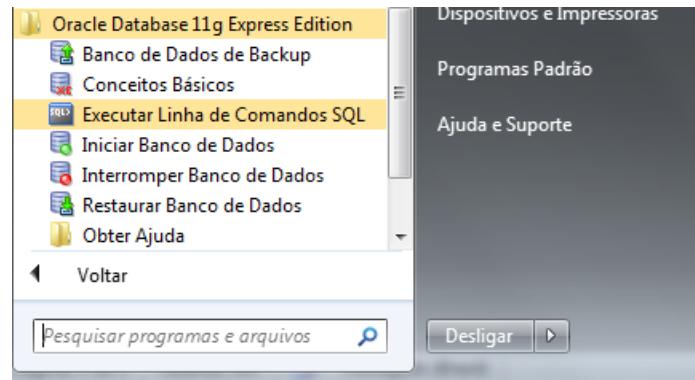
<http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>

Para instalar e configurar o Oracle Database Express Edition basta seguir a instalação padrão, até chegar na tela onde deve ser especificado a senha do banco de dados, esta senha deve ser salva pois será necessário para criação do Workspace mais tarde. Esta senha será usada para o usuário, SYS ou SYSTEM.



**Figura 6-35. Senha para usuário padrão Oracle**

Após a instalação finalizada a pasta oraclexe será criada no caminho especificado na instalação e um sub menu chamado *Oracle Database 11g Express Edition* no *Menu iniciar*, como mostrado na Figura 6-36.



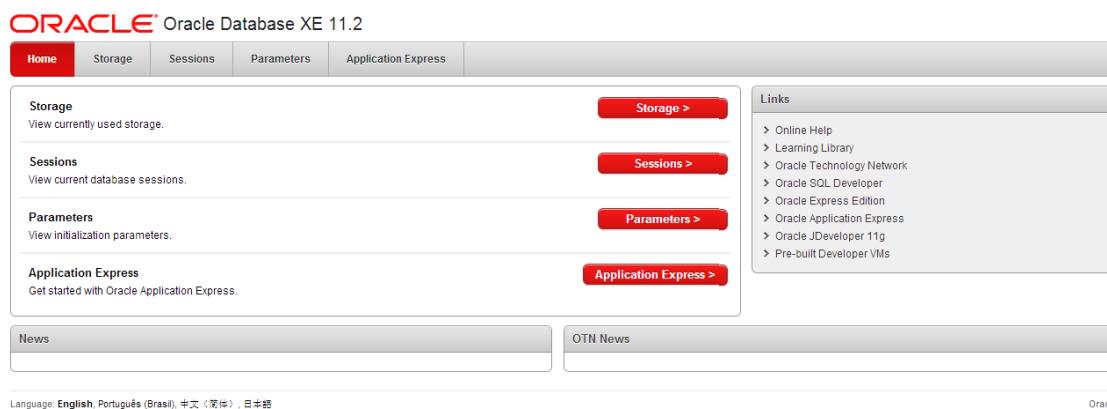
**Figura 6-36. Sub menu Oracle Database 11g Express Edition**

Deve ser executado o comando *Iniciar Banco de Dados* presente neste sub menu.

O banco de dados será iniciado, então deve-se ir para o caminho:

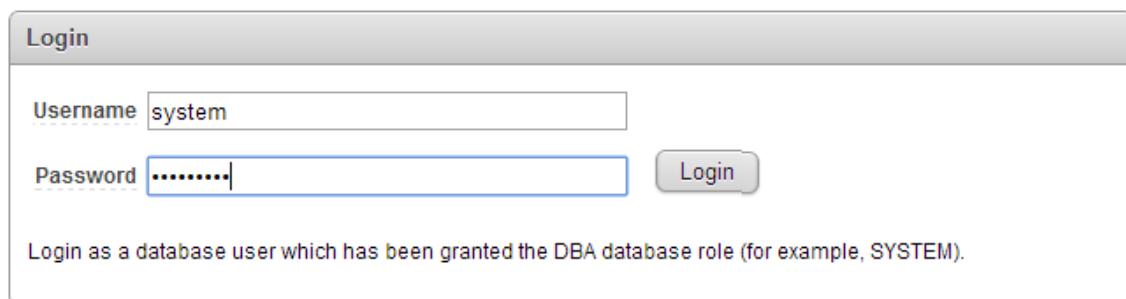
oraclexe\app\oracle\product\11.2.0\server\

E executar o arquivo *Get\_Started.url*. Após isso o configurador do seu banco de dados será aberto no navegador padrão instalado, como na Figura 6-37.



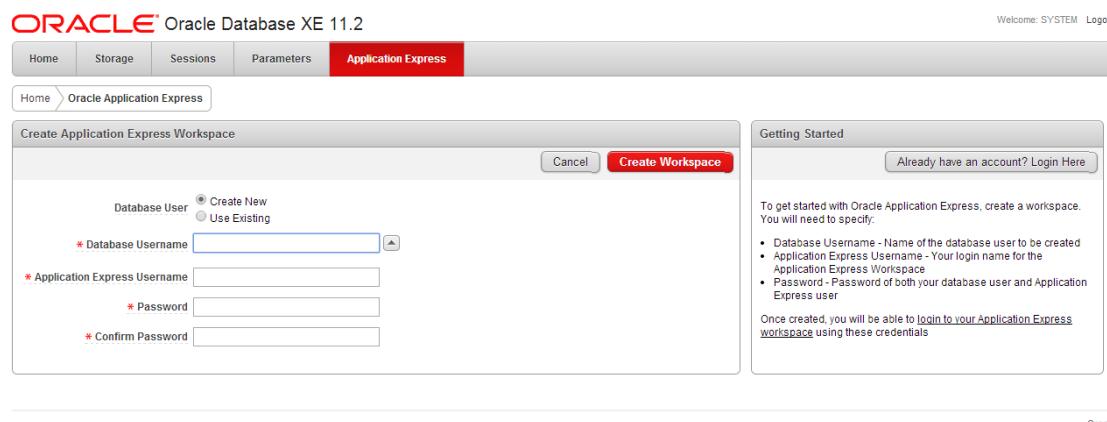
**Figura 6-37. Configurador Oracle por Browser**

Nesta página deve ser criado um Workspace, para isso, escolher a opção *Application Express*, será pedido para efetuar um login, como mostrado na Figura 6-38, para efetuar o Login deve ser utilizado o Usuário padrão SYSTEM ou SYS com a senha escolhida durante a instalação.



**Figura 6-38. Tela de login Oracle**

Após isso a tela mostrada na Figura 6-39 será aberta.



**Figura 6-39. Tela de criação de Workspace**

Marcar a opção *Create New* para criar uma nova aplicação, e preencher os dados, informando o nome desejado para o novo DataBase, um nome para a aplicação em *Application Express Username* e um Password para o novo Workspace.

**ATENÇÃO:**

Essas informações serão muito importantes para configurar o Banco de Dados com o BluePlant.

Após criado o Workspace deve-se fazer a instalação dos Providers.

#### Instalação e Configuração do Provider Oracle

O provider Oracle.DataAccess.Client utilizado no exemplo de conexão com banco de dados Oracle, está disponível para download gratuito no seguinte endereço:

<http://www.oracle.com/technetwork/database/windows/downloads/utilsoft-087491.html>

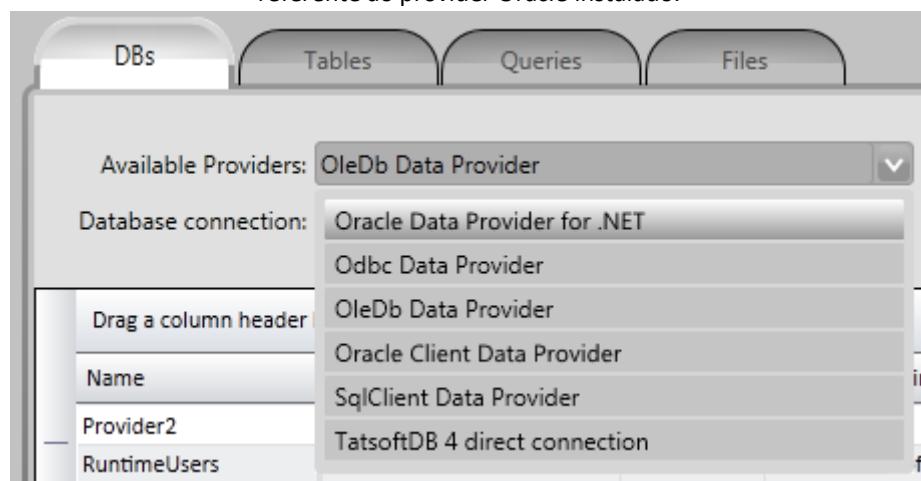
Deve ser feito o download do arquivo ODAC112021XCopy, no exemplo utilizamos a versão de 32 Bits:

Passos para a instalação:

1. Após o download, extrair o conteúdo do .zip;
2. Haverá um arquivo chamado install.bat;
3. Executar o Command Prompt (cmd.exe) com privilégios de Administrador;
4. Dentro da pasta do arquivo, executar o install.bat com a seguinte linha de comando:  
install.bat all C:\OracleClient true
5. Será criado o diretório C:\OracleClient;

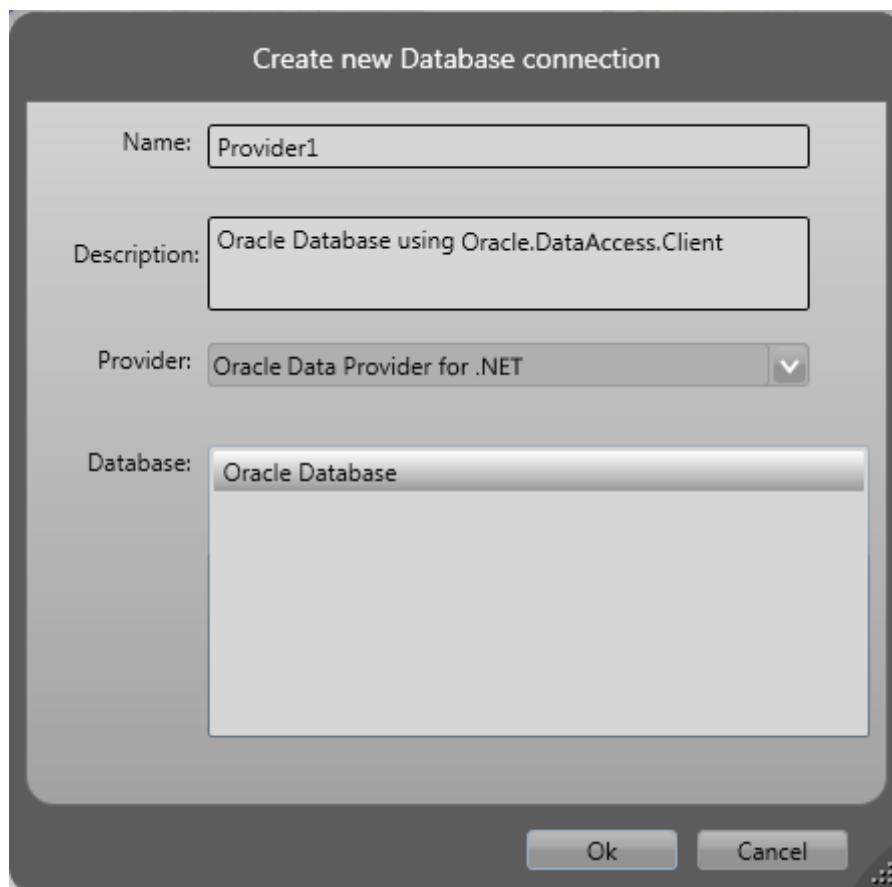
#### Utilização do Provider Oracle no BluePlant

Após os passos descritos acima, reiniciar o BluePlant. Na lista de Providers deve ter sido adicionado um novo provider referente ao provider Oracle instalado.



**Figura 6-40. Tela de DBs no BluePlant**

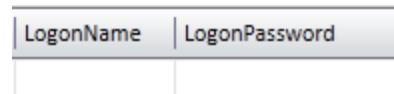
Após a confirmação da instalação do provider Oracle, clicar em **Create new...** e seleciona-lo na lista de providers instalados, conforme a Figura 6-41.



**Figura 6-41. Inserindo Provider Oracle**

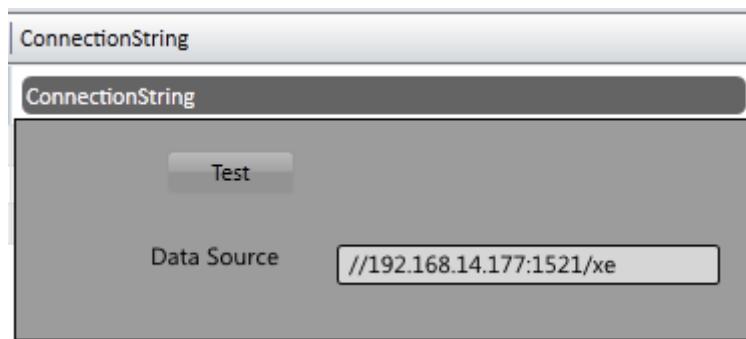
Após a criação da nova conexão, clicar na linha do banco de dados Oracle para inserir os valores s campos *Database Username* e *Password* do banco de dados nas respectivas colunas: *LogonName* e *LogonPassword*, estes devem ser previamente cadastradas no banco de dados, para assim garantir que a conexão seja estabelecida.

O LogonName é referente ao nome escolhido para o Workspace na tela de configurações web do Oracle Database 11g Express Edition, assim como a senha.



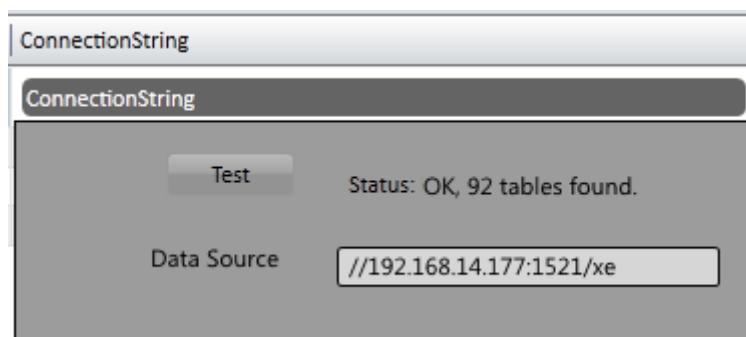
**Figura 6-42. Usuário e Senha para Conexão com Banco de Dados Oracle**

Agora acesse o campo da coluna *ConnectionString*, e insira as configurações da conexão, sendo o endereço IP onde o banco se encontra, a porta (padrão 1521) e o SID do banco (padrão XE) neste momento deve-se conseguir estabelecer a conexão. Conforme pode ser visto na Figura 6-43.



**Figura 6-43. Configuração do Endereço IP, Porta e SID do Banco de Dados Oracle**

Depois de configurado o endereço IP, porta e SID, para verificar se a conexão com o banco de dados Oracle está funcionando, basta clicar no botão *Test*, neste momento deve aparecer o número de tabelas inseridas no banco de dados informado, como mostrado na Figura 6-44.



**Figura 6-44. Teste de Conexão com Banco de Dados Oracle**

**ATENÇÃO:**

O provider utilizado no exemplo para conexão com banco de dados Oracle, requer clientes Oracle na versão 9.2 ou anteriores. O provider Oracle que acompanha o BluePlant é o System.Data.OracleClient, requer clientes Oracle na versão 8.1.7 ou superiores.

#### Conexão com Banco de Dados SQLServer

Este capítulo apresenta um exemplo de como criar uma base de dados SQL Server.

#### Instalação e configuração do Banco de Dados SQL Server

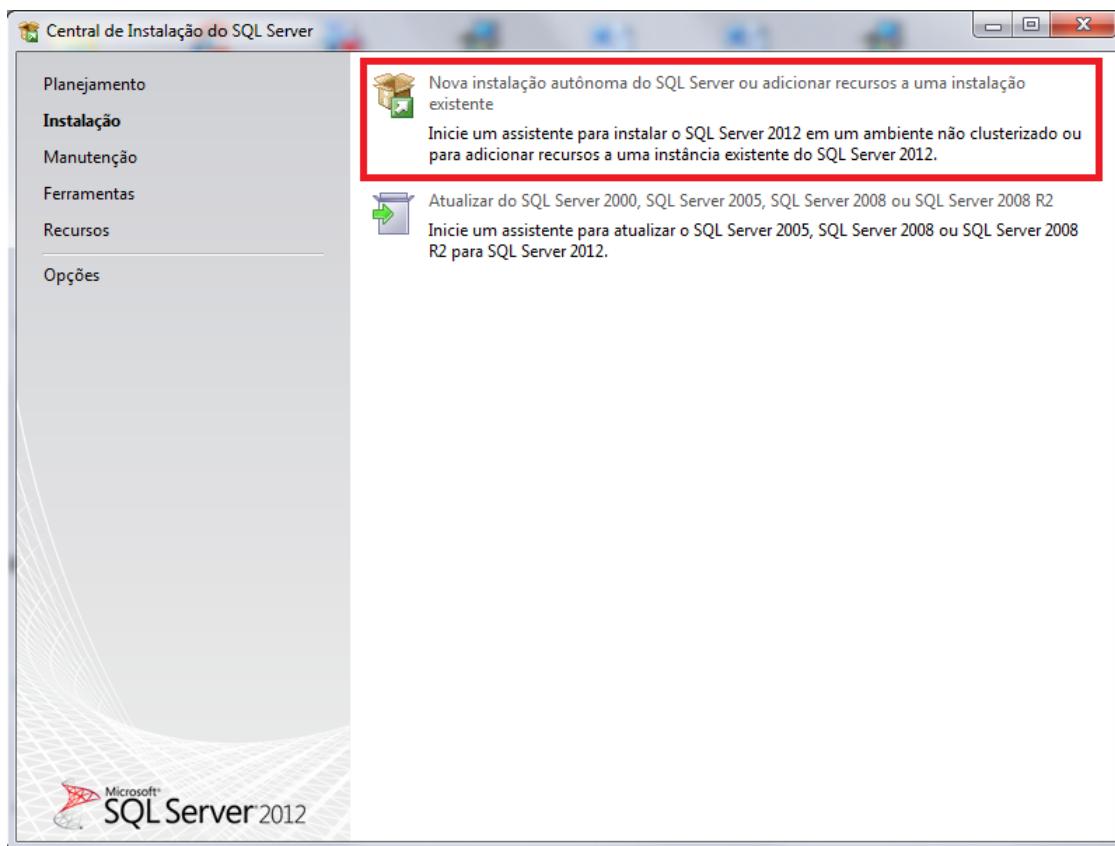
O SQL Server utilizado no exemplo foi o SQL Server 2012 Express, está disponível gratuitamente em:

<http://www.microsoft.com/pt-br/download/details.aspx?id=29062>

**ATENÇÃO:**

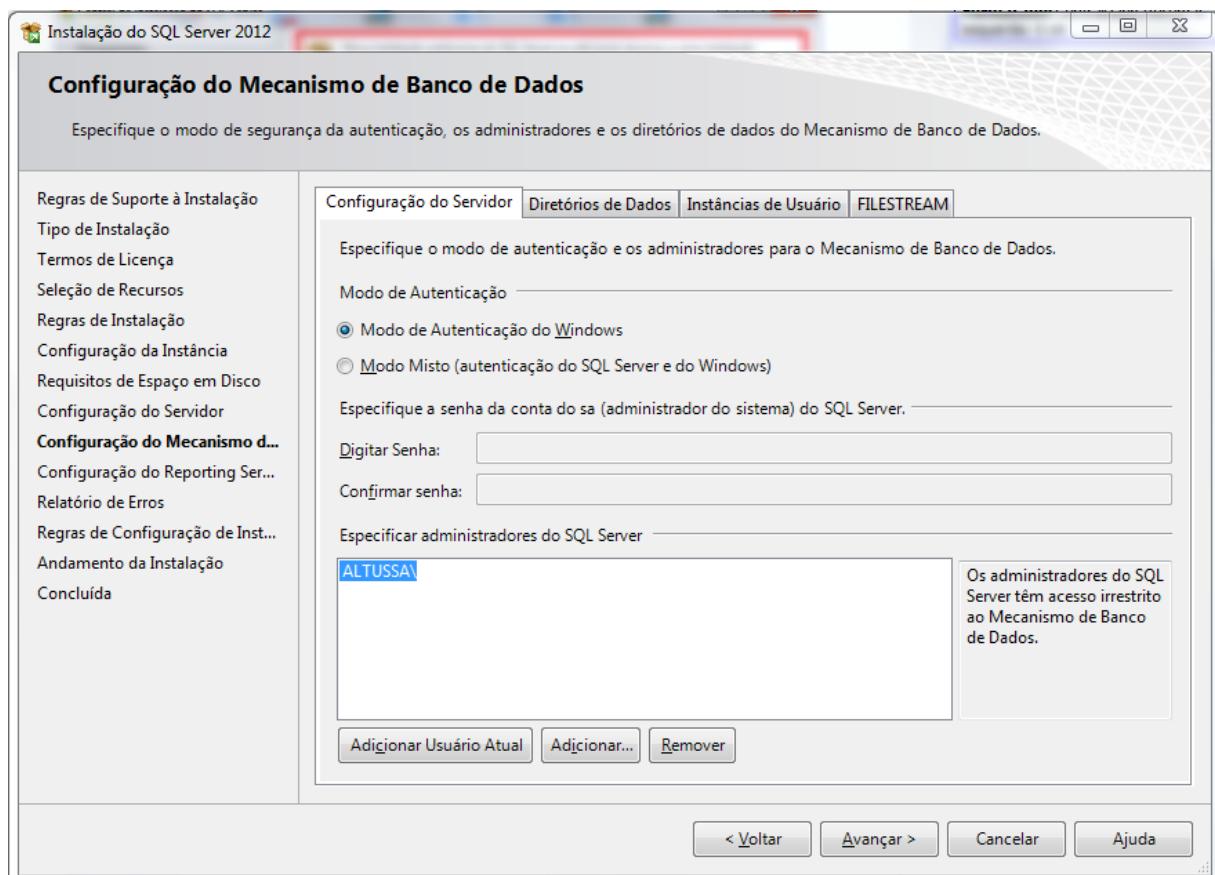
Antes de instalar e configurar o SQL Server 2012 Express deve ser instalados também os demais pacotes mostrados para download no link passado, após a instalação desses pacotes deve se seguir a instalação do SQL Server 2012 Express pelo SQLEXPRAADV\_xBites\_PTB.exe

Para instalar e configurar o SQL Server 2012 Express execute o .exe SQLEXPRAADV\_xBites\_PTB, a tela mostrada na Figura 6-45 irá aparecer, deve-se selecionar a primeira opção.



**Figura 6-45. Instalador SQL Server 2012 Express**

Após isso o instalador vai aparecer então apenas avance por ele deixando-o padrão até chegar na tela onde deve ser especificado o usuário e a senha do banco de dados, tela mostrada na Figura 6-45. Este usuário será usada para configurar o BluePlant com o SQL Server.



**Figura 6-46. Tela do Mecanismo de Banco de Dados**

Na tela de configuração do mecanismo de Banco de Dados recomenda-se que se deixe marcada a opção *Modo de Autenticação do Windows*, assim o SQL Server irá utilizar seu usuário do Windows como logon.

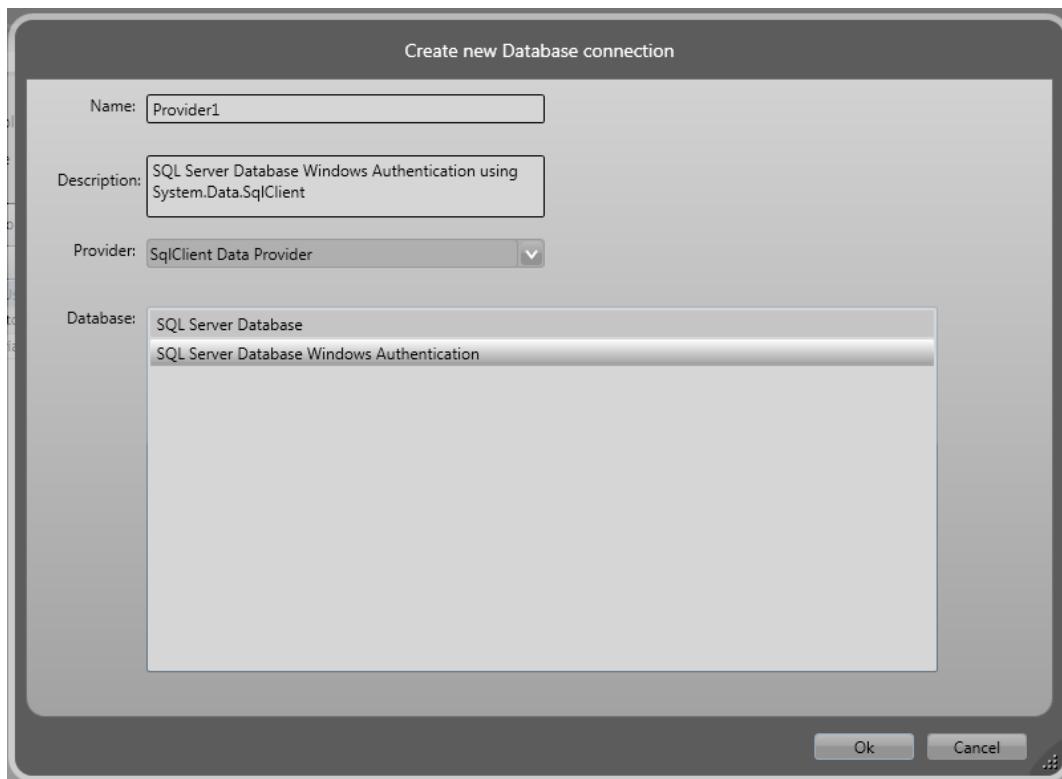
**ATENÇÃO:**

Com a instalação padrão o nome da sua instância será **MSSQLSERVER**.

#### Utilizando SQL Server no BluePlant

Após a instalação do SQL Server descrita acima, reinicie o BluePlant, e vá até a aba **Edit\Datasets** aba **DBs** e clique em **Create new...**, a tela para criar uma nova conexão vai abrir como mostrado na Figura 6-47.

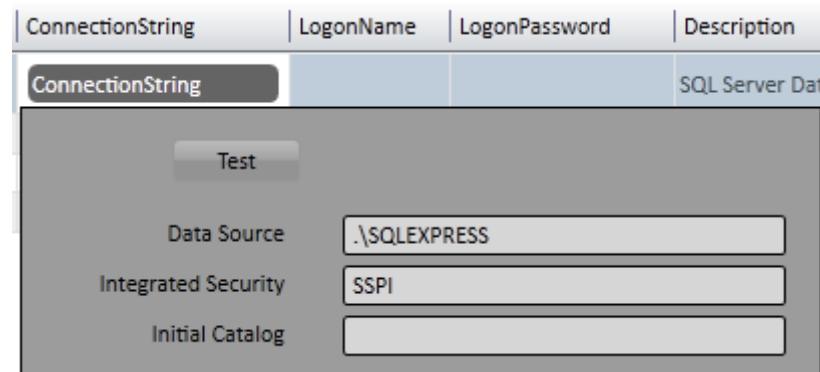
Selecione o provider do SQL Client Data Provider, na lista de providers instalados em DataBase selecione **SQL Server Database Windows Autentication** caso durante a instalação do SQL Server tenha sido escolhido autenticação do Windows, conforme a Figura 6-47.



**Figura 6-47. Tela de criação de nova Conexão configurada para SQL Server**

Escolher um nome e pode-se fazer uma descrição para a nova conexão, clique em Ok para finalizar, após isso, uma nova linha será inserida na tabelade conexões.

Após finalizada a criação deve-se configurar a string de conexão, para isso basta acessar o campo da coluna *ConnectionString*, a janela mostrada na Figura 6-48 será aberta.



**Figura 6-48. Tela de configuração da ConnectionString para o SQL Server**

Nela deve ser informado apenas o Data Source, sendo este o ip onde está o banco de dados ou nome do computador barra a sua instância criada durante a instalação, caso esta não tenha sido configurada, por padrão ela recebe o valor MSSQLSERVER.

Depois de configurado a ConnectionString, para verificar se a conexão com o banco de dados SQL Server está funcionado, clique no botão *Test*, neste momento deve aparecer o número de tabelas inseridas no banco de dados informado, como mostrado na Figura 6-49.

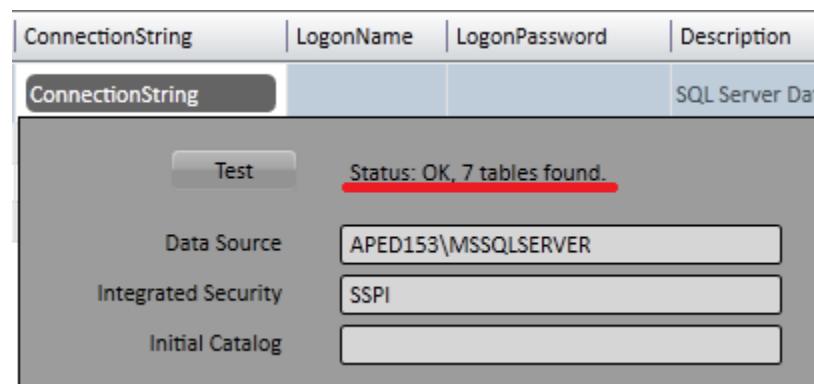


Figura 6-49. Tela de configuração da ConnectionString para o SQL Server com tabelas encontradas

#### Tables

A guia *Tables* permite acessar as tabelas de dados a partir das bases de dados configuradas (DBs). A Figura 6-50 ilustra essa seleção.

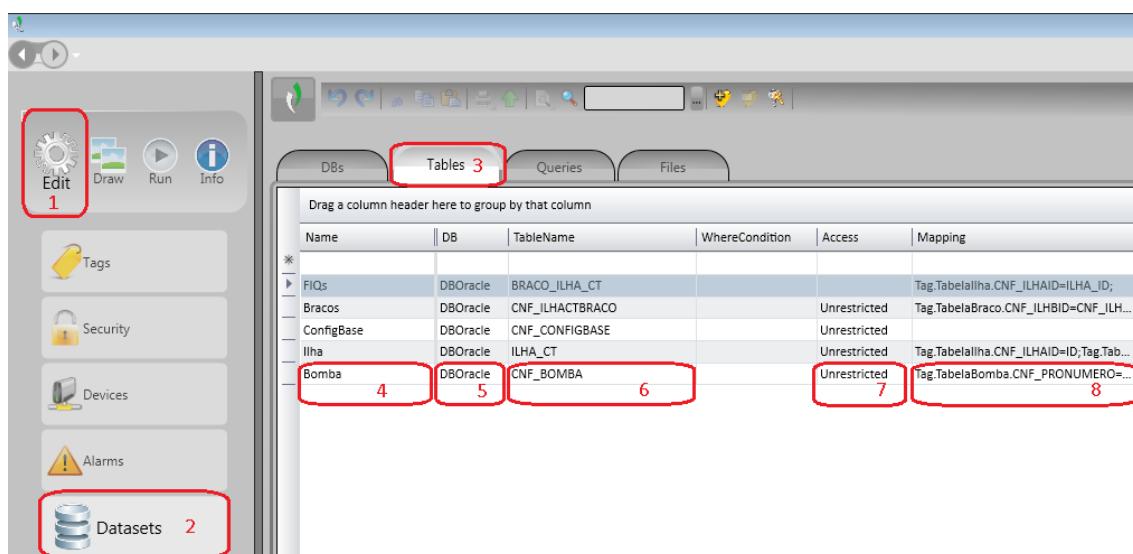


Figura 6-50. Edição de Tabelas de Dados

Os passos para editar ou incluir tabelas estão ilustrado na Figura 6-50 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 criado 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* os tags relacionados com as colunas da tabela do banco 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:

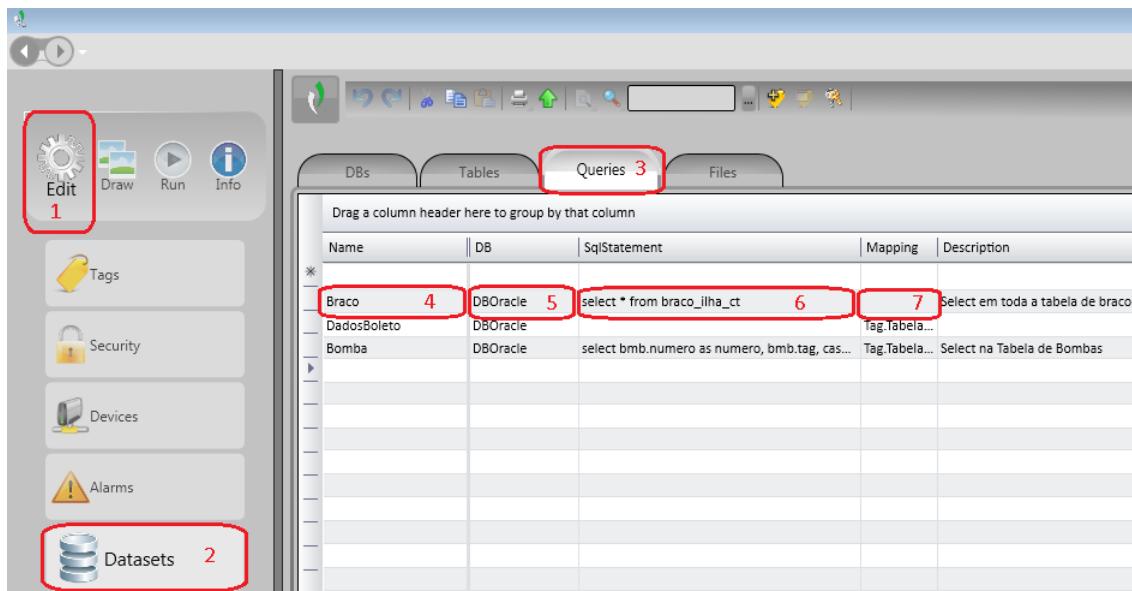
- Read (leitura)
- Insert (inserção)
- ReadWrite (leitura e escrita)
- Unrestricted (irrestrita)

**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 aos tags. Então, ao executar o comando de atualização, os conteúdos do Tag são escritos na linha.

**Queries**

Esta guia mapeia as bases de dados configuradas (DBs) conforme mostrado na Figura 6-51.



**Figura 6-51. Edição de Queries**

Os passos para editar ou incluir instruções SQL(queries) estão ilustrados na Figura 6-51 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* os 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

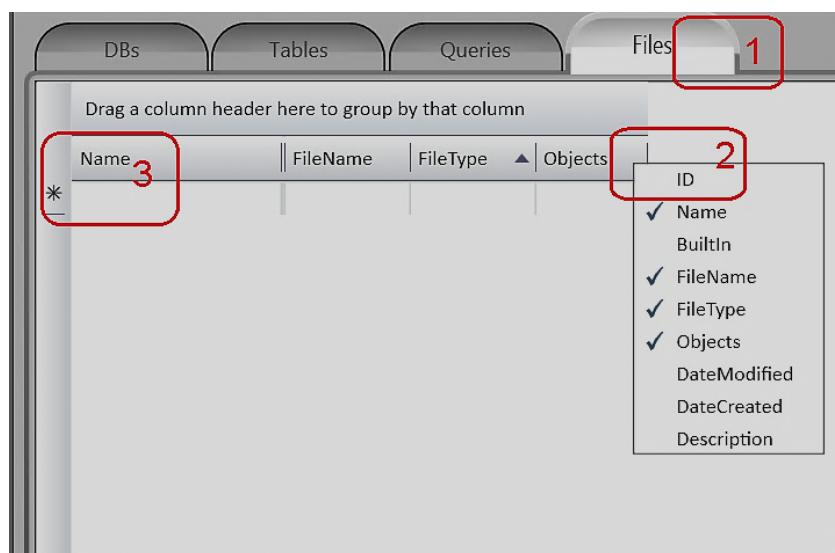
Instrução SQL utilizada para realizar consultas no banco de dados.

#### Mapping

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

#### Files

A guia Files define o(s) arquivo(s) para a troca de informações com os bancos de dados conforme ilustrado na Figura 6-52.



**Figura 6-52. 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 a ilustração da Figura 6-52.

**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 os conteúdos do arquivo com objetos Tag.

**Editando Scripts (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 6-53 enfatiza a aba *Tasks* para edição de tarefas de script. Existem quatro tarefas incorporadas:

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

Name	Code	Trigger	Period	Domain	BuildStatus	BuildErrors
*						
ClientShutdown	VBdotNet	Client.ShutDown		Client	✓	none
ClientStartup	VBdotNet	Client.Startup		Client	✓	none
ServerShutdown	VBdotNet	Server.ShutDo...		Server	✓	none
ServerStartup	VBdotNet	Server.Startup		Server	✓	none

**Figura 6-53. 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 a ilustração da Figura 6-53.

#### 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: Tarefa está pronta para ser executado desde o inicio.
- Disabled: 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

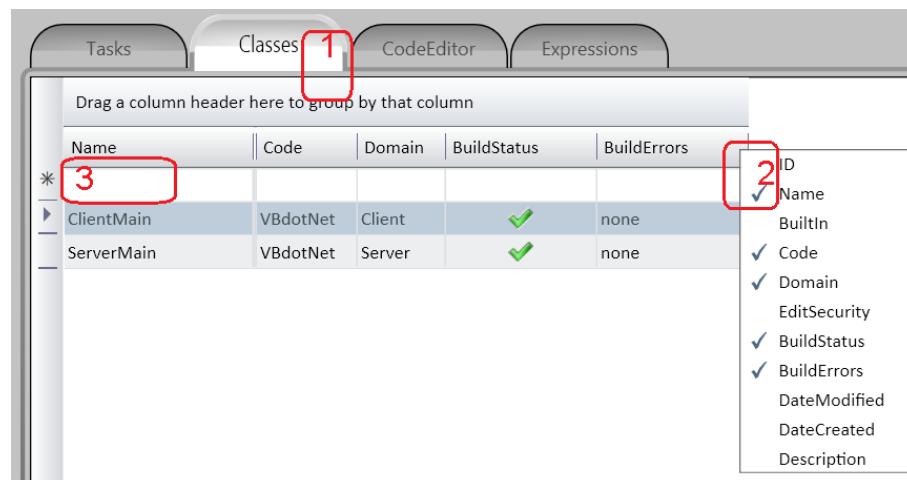
#### BuildErrors

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 6-54 mostra esse item do menu *Script*. Há duas *UserClasses* incorporadas:

- ServerMain: biblioteca de métodos disponível para todas as tarefas do servidor e clientes
- ClientMain: biblioteca de métodos disponível para todos clientes scripts, inclusive telas scripts



The screenshot shows a software interface with a toolbar at the top containing 'Tasks', 'Classes' (highlighted with a red box and number 1), 'CodeEditor', and 'Expressions'. Below the toolbar is a table titled 'Drag a column header here to group by that column'. The table has columns: 'Name', 'Code', 'Domain', 'BuildStatus', and 'BuildErrors'. There are two rows: 'ClientMain' (VBdotNet, Client, green checkmark, none) and 'ServerMain' (VBdotNet, Server, green checkmark, none). A context menu is open over the 'ClientMain' row, with a red box and number 2 highlighting it. The menu items are: ID, Name (checked), BuiltIn, Code (checked), Domain (checked), EditSecurity, BuildStatus (checked), BuildErrors (checked), DateModified, DateCreated, and Description. A red box and number 3 highlights the 'Name' column header in the table.

Name	Code	Domain	BuildStatus	BuildErrors
*				
ClientMain	VBdotNet	Client	✓	none
ServerMain	VBdotNet	Server	✓	none

**Figura 6-54. 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 a ilustração da Figura 6-54.

#### Name

Indica o nome da classe do usuário.

#### Code

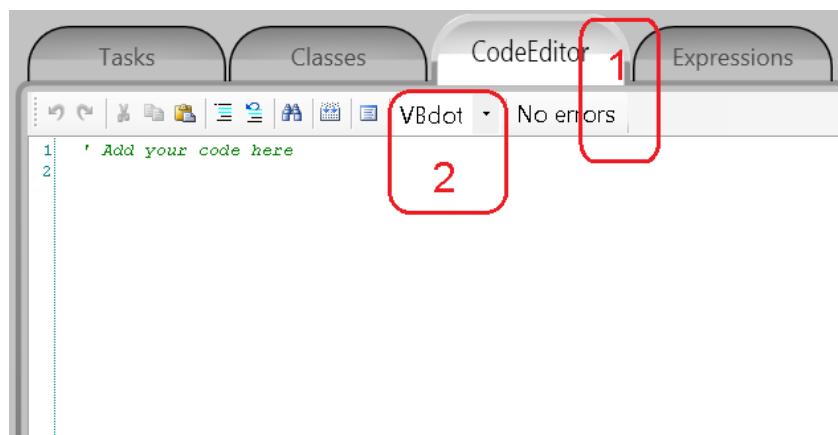
Indica a linguagem do Script. O usuário pode selecionar entre VBdotNet ou CSharp.

#### 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 6-55 (Item 1). As opções de linguagem de programação são: *VBdotNet* ou *CSharp*. A seleção da linguagem está ilustrada na Figura 6-55 (Item 2). Informações sobre a sintaxe das linguagens de programação podem ser obtidas no seguinte endereço <http://www.microsoft.com/net>.



**Figura 6-55. CodeEditor**

### 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 os 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 trêsparâmetros. 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") irá retornar uma string de acordo com o valor do tag.A.

Neste método dot NET, todos os três parâmetros são avaliados independentemente da condição. Por exemplo, se tem IF (tag.A = 1, script.class.client.Func1(),script.class.client.Func2())

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 esse cenário o Blueplant tem um método chamado TIF, que tem esse comportamento.

Então, usa-se a expressão:

TIF (tag.A = 1,script.class.client.Func1(),script.class.client.Func2())

Apenas a Func1 () ou a Func2 () vai ser executada, de acordo com o valor de Tag.A

O método TIF é definido na biblioteca de classes que é automaticamente exposta a expressões que estão na biblioteca de funções Toolkit.

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.

- 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:
  - Cliente: Expression será executada em cada sistema cliente. Estas são expressões que se aplicam localmente (no computador do usuário), por exemplo, geração de relatórios.
  - Servidor: Expression 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 um Tag ou objeto que será usado para disparar a execução da expressão. O disparo da execução ocorrerá quando o valor do tag ou objeto se alterar.
- DisableCondition: Digite ou selecione um 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. Exibetodos 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

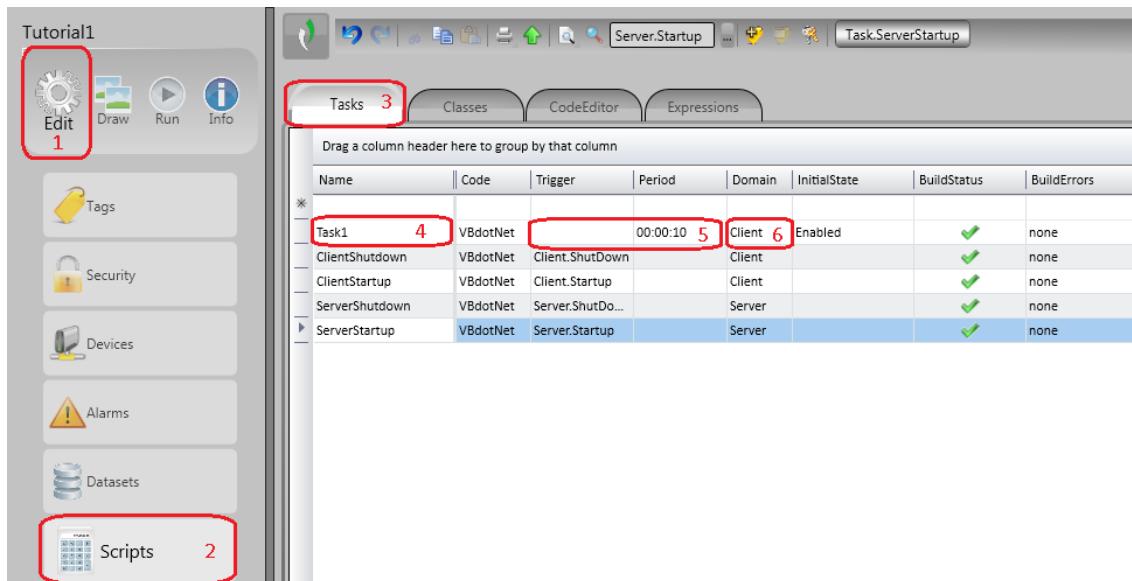


Figura 6-56. Tarefas de Scripts

Os passos para editar ou incluir uma tarefa de script estão ilustrados na Figura 6-56 e descritos a seguir:

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 6-57 a seguir apresenta os passos para a codificação da tarefa de script chamando o método de uma classe existente.

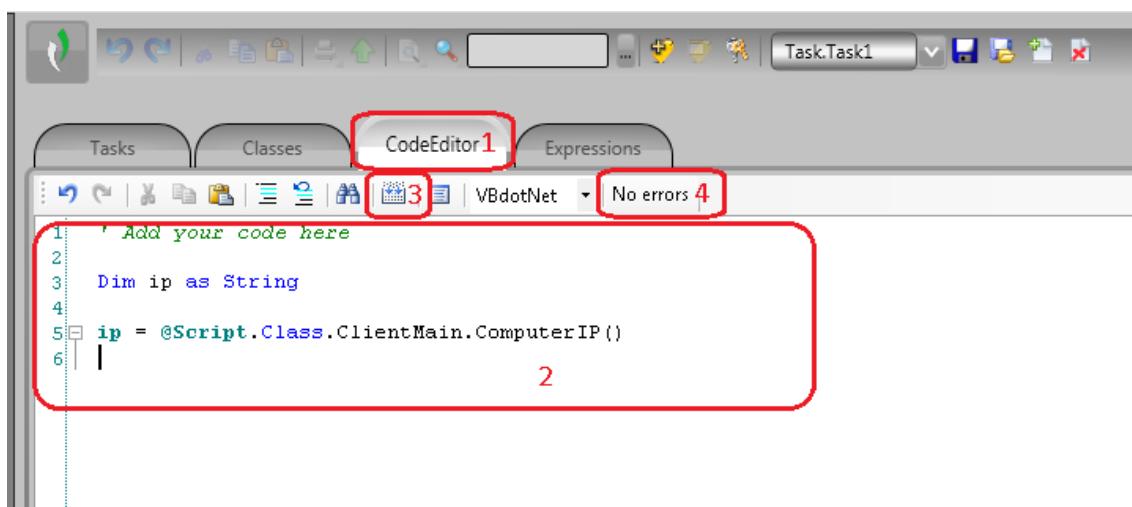


Figura 6-57. CodeEditor das Tarefas de Scripts

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

### Criando Informação de Depuração

O Blueplant tem um depurador dot NET 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 dot NET, o ambiente de Engenharia deve ser conectado ao ambiente do runtime e o depurador dot NET 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.
2. 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.
3. 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 6-58. Anexando depurador .NET

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

A Figura 6-65 mostra o *Editor de Código* 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.

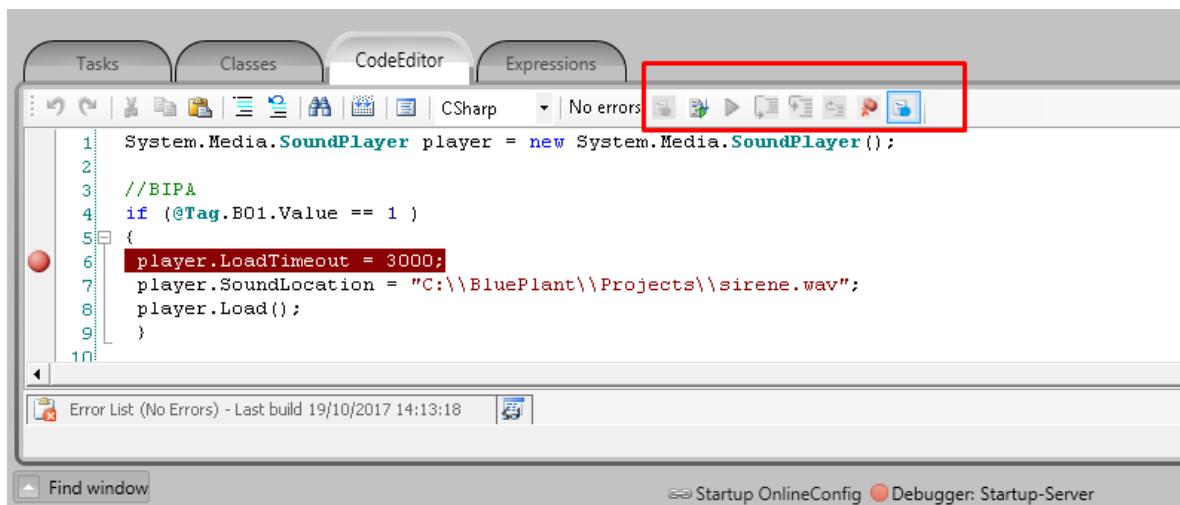


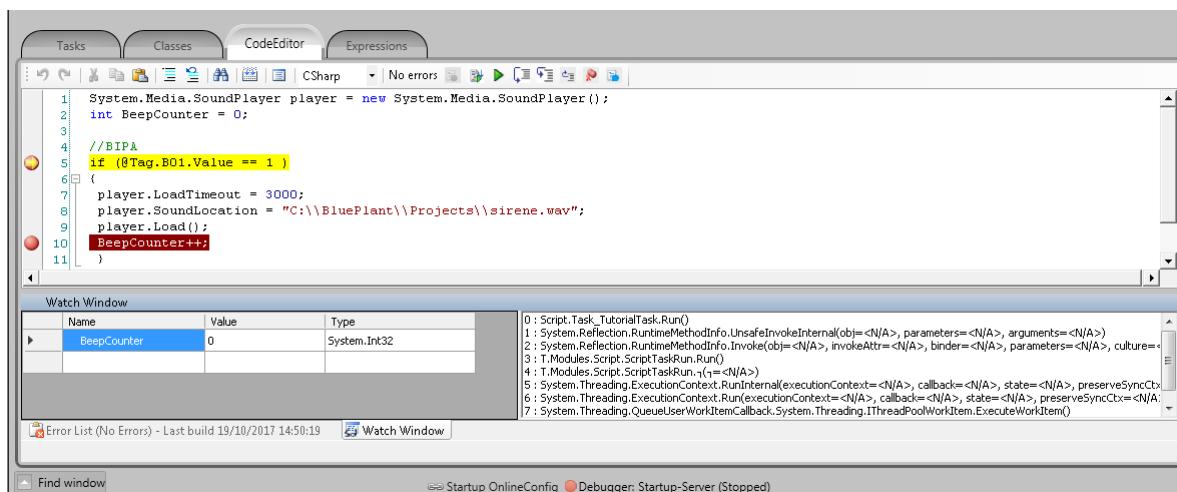
Figura 6-59. Editor de Código os Botões para Anexar e Desanexar depurador

#### 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 dot NET 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 6-60. Watch para variáveis dot NET**

Também pode-se adicionar variáveis locais do dot NET 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 distingua entre objetos de projeto, e variáveis locais do dot NET. Tenha em mente o Watch atualizado apenas quando a execução é interrompida. Se quiser ter os valores em tempo real para tags e objetos que pode-se abrir a ferramenta de diagnóstico *PropertyWatch*.

### 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 6-61 ilustra esse ambiente no formato de tabela.

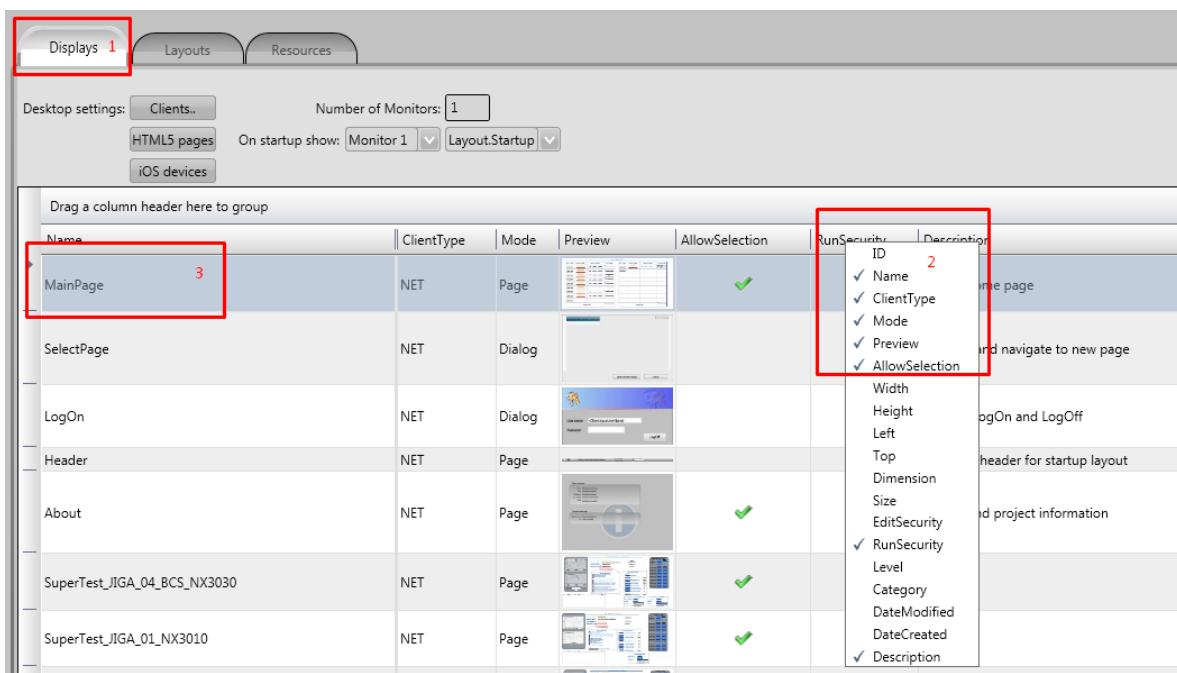


Figura 6-61. Edição de Displays

Os itens que compõem o menu de edição de telas estão numerados na Figura 6-67e são descritos na sequência:

1. Acesso 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:

- Página: este é o modo de exibição padrão. Quando uma Página é aberta automaticamente, a última página do layout atual é fechada
- Diálogo: 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 *PageSelecter* que é configurada na guia

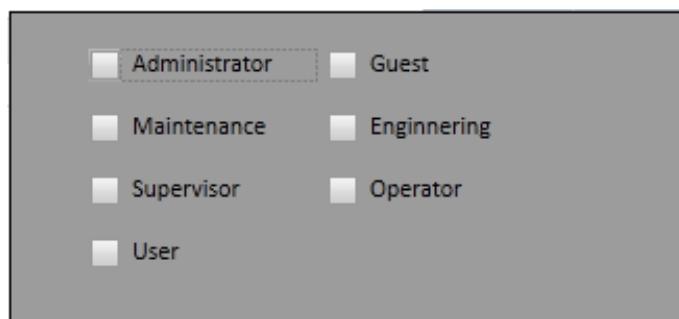
*Editar>Telas>Telas*. Para impedir essa seleção pelo operador padrão remova o marcador da página ou da opção *AllowSelection* na coluna dos objetos.

### EditSecurity

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

### RunSecurity

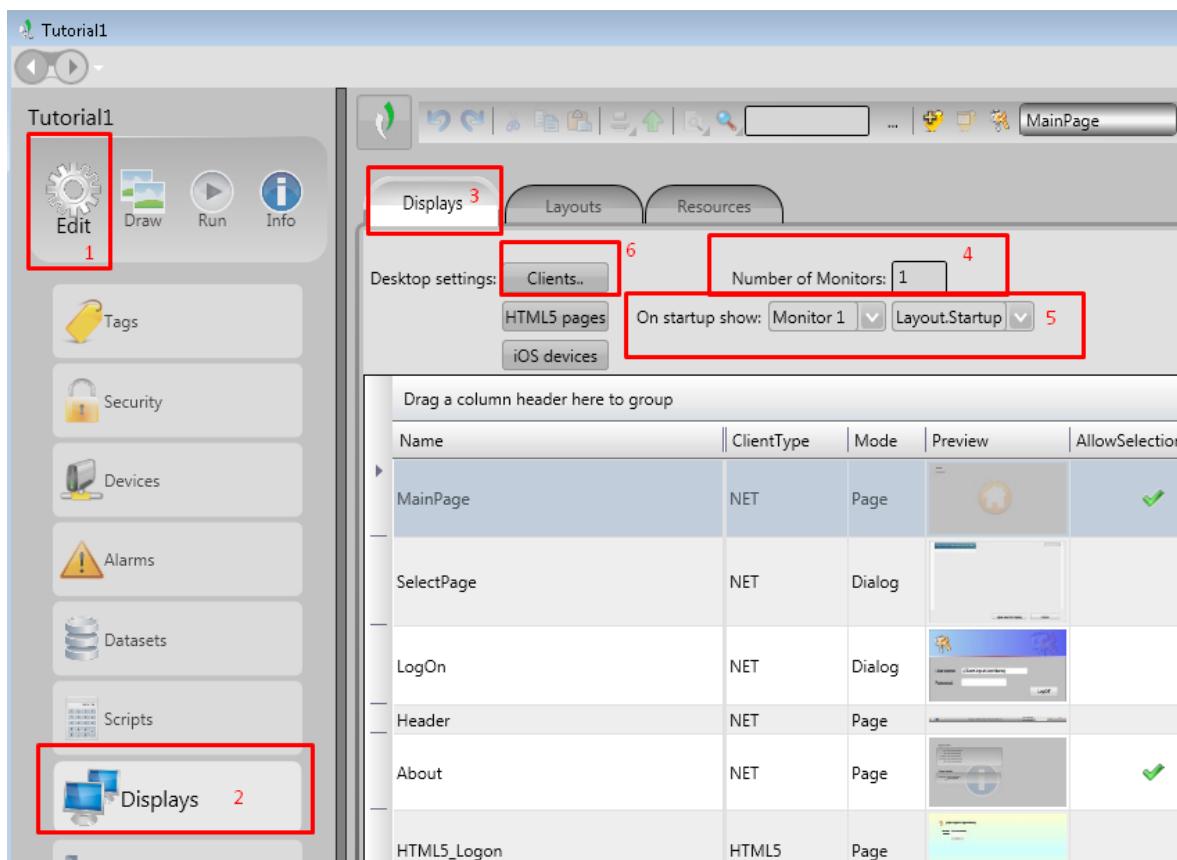
Grupos de permissões de segurança exigidos para abrir esta tela durante o Runtime. Clique no campo *RunSecurity* para abrir a janela *Run Security Selection*. Após concluir as seleções clique em qualquer lugar na tabela para fechar a janela de seleção. Suas seleções aparecerão no campo *RunSecurity*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



**Figura 6-62. Permissões no Runtime**

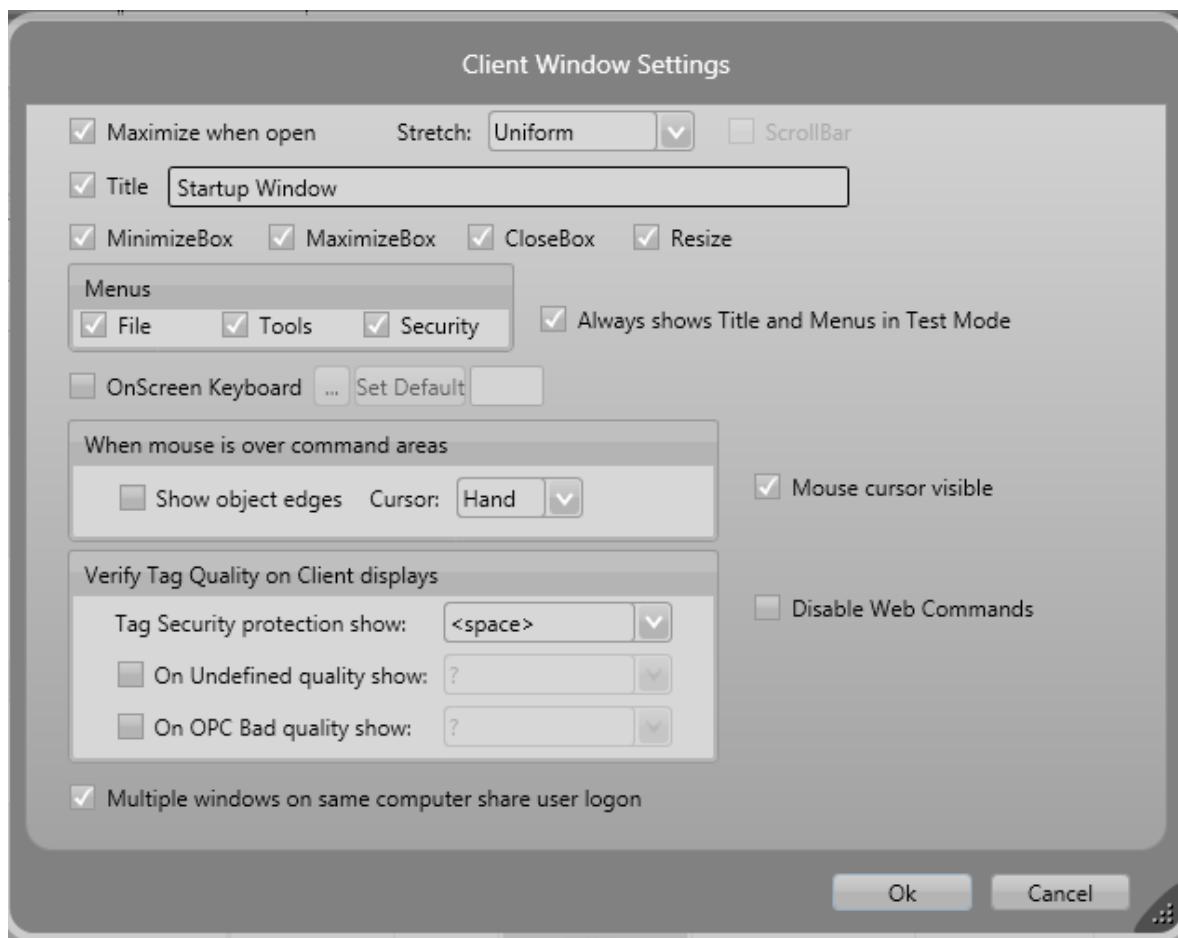
### Configurando Múltiplos Monitores

Os passos para configurar múltiplos monitores estão ilustrados na Figura 6-63 e descritos abaixo:



**Figura 6-63. Configurando Múltiplos Monitores**

1. Selecionar o menu *Edit*
2. Acessar o item *Displays*
3. Clicar na guia *Displays* para acesso as 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). Conforme Figura 6-64.



**Figura 6-64. 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:
  - **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.
  - **Uniforme:** 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**
  - **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
  - Cursor: Altera o tipo de cursor.
- **Verify Tag Quality on Client Displays:**
  - **On undefined quality show:** Seleccionar a opção e o caractere que deve ser mostrado quando a qualidade dos dados vindos do campo for indefinida.
  - **On OPC Bad quality show:** Seleccionar 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 6-65 ilustra a configuração dos layouts.

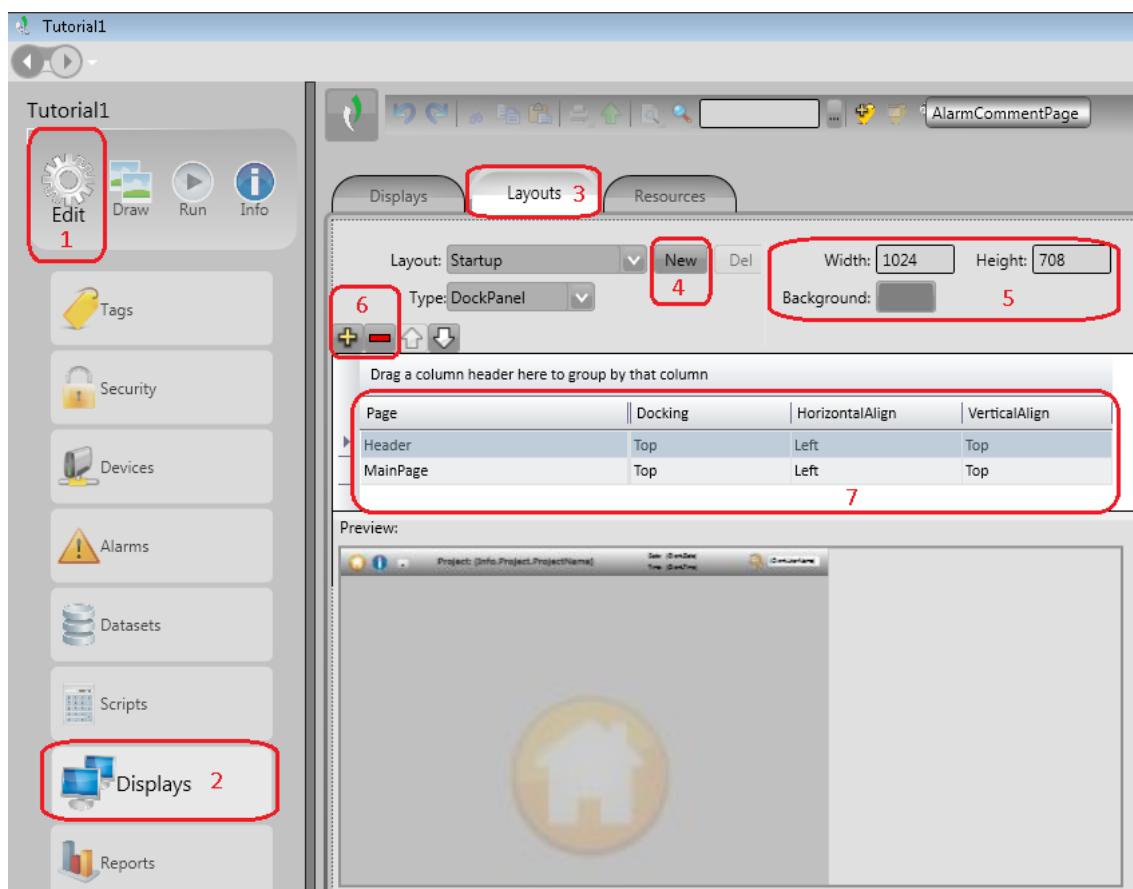


Figura 6-65. Edição de Layouts

Os passos para editar e incluir um novo layout estão ilustrados na Figura 6-71 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
- 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
- Direita

### VerticalAling

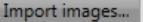
Define o alinhamento vertical. As opções possíveis são:

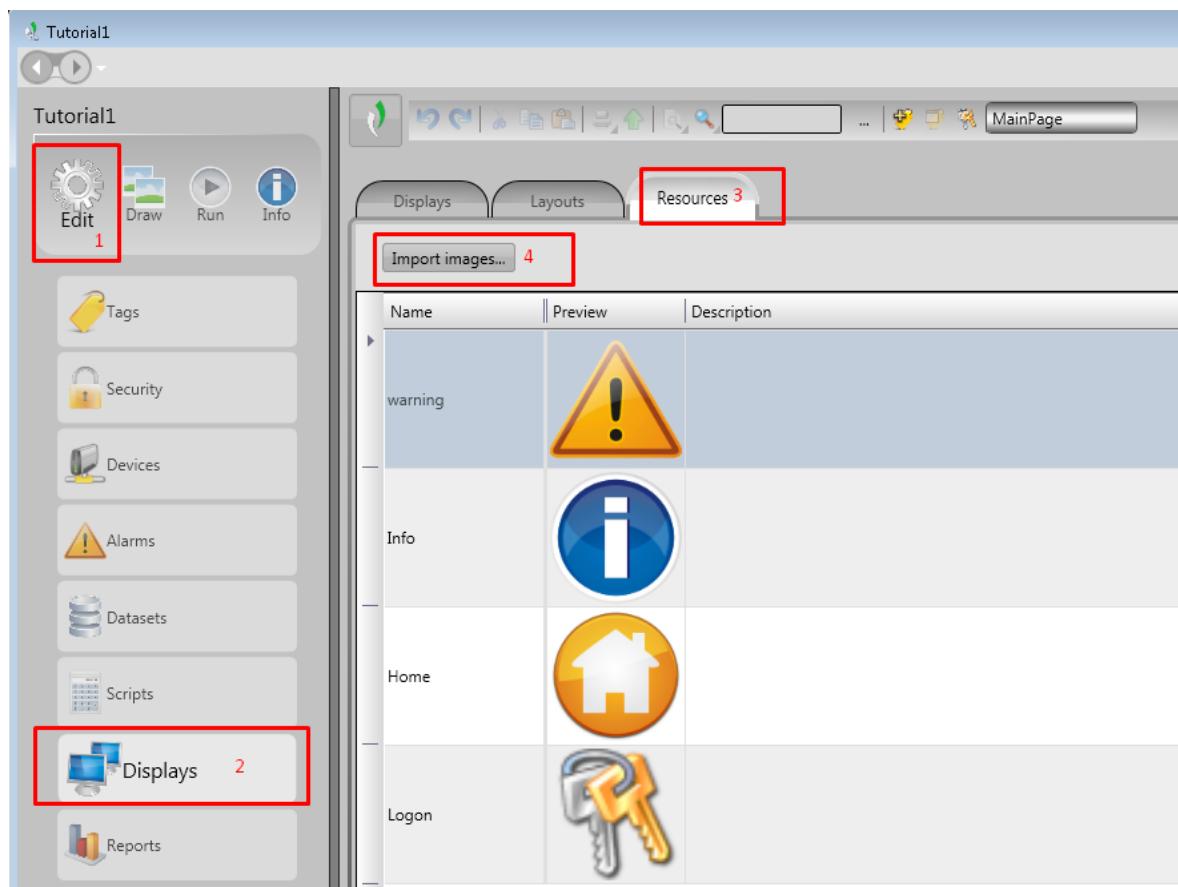
- Topo
- Centro
- Inferior

### 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  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 6-66 ilustra as três etapas do menu de edição de recursos.



**Figura 6-66. Edição de Recursos**

Os passos para importar novas imagens estão ilustrados na Figura 6-66 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 (Reports)

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 6-67.

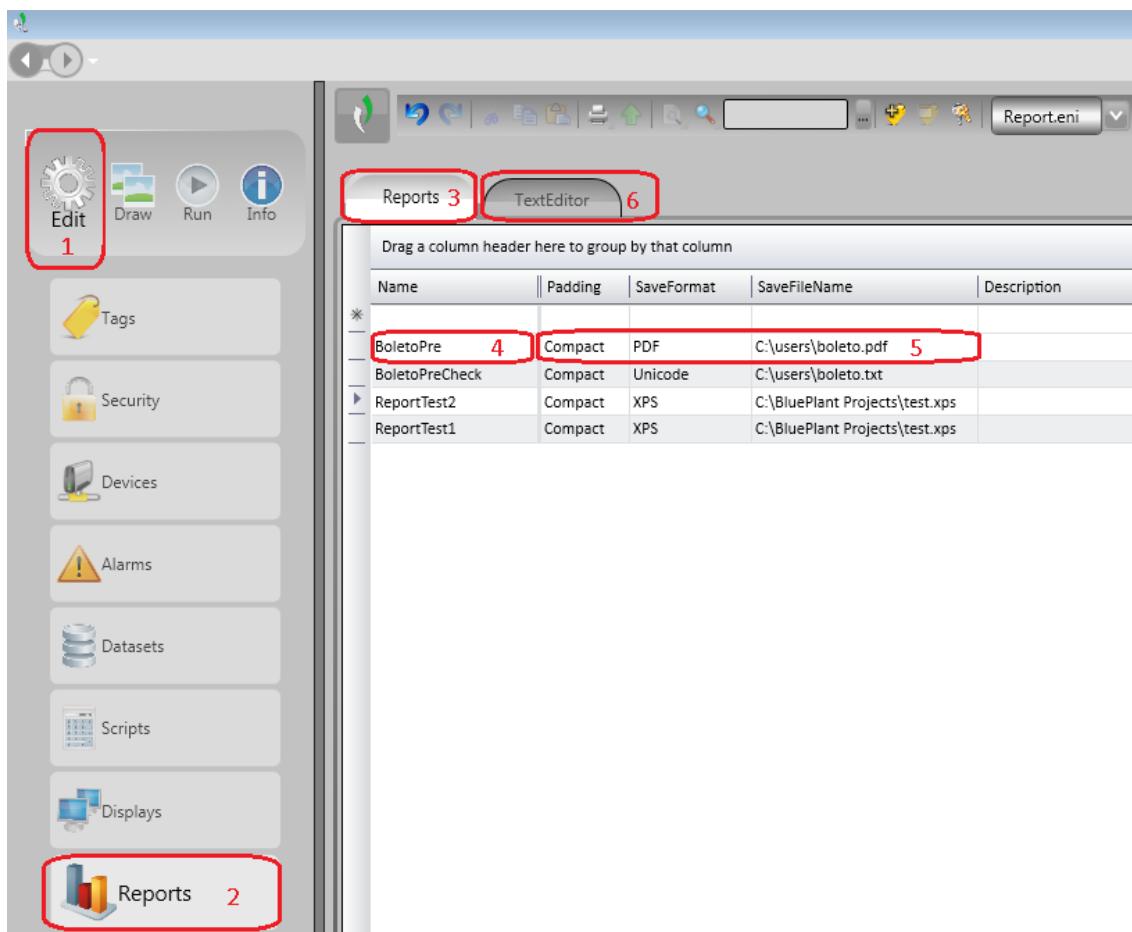
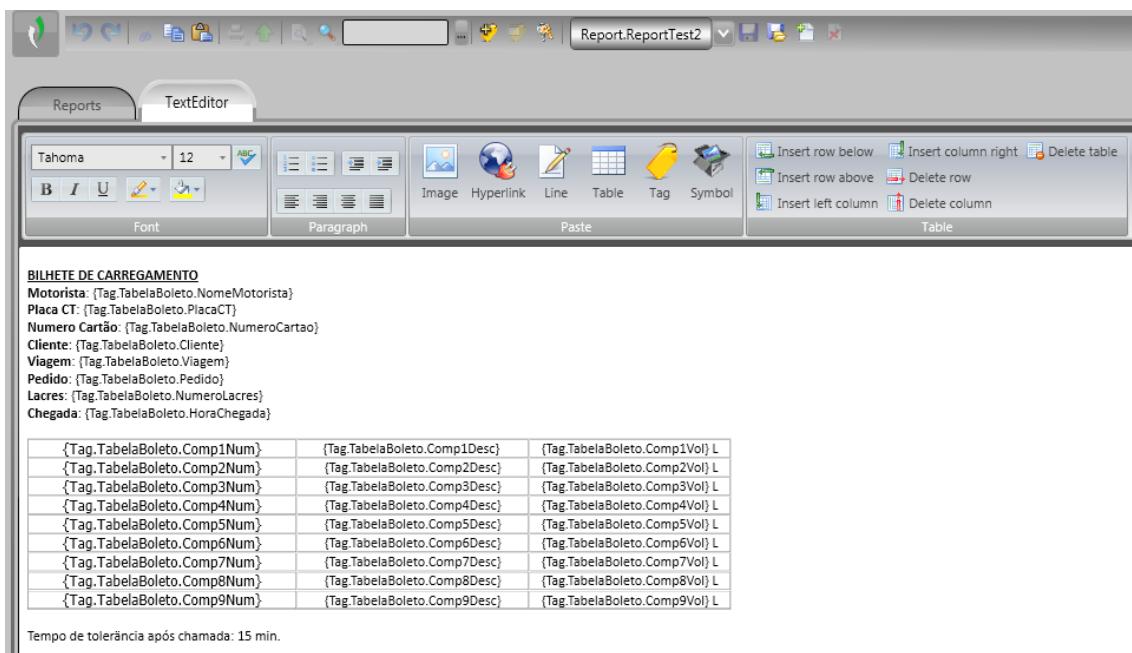


Figura 6-67. Configuração de Relatórios

Os passos para editar ou incluir um novo relatório estão ilustrados na Figura 6-67 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 6-68 mostra com maiores detalhes



**Figura 6-68. Edição de Relatório**

#### Name

Indica o nome do relatório a ser gerado.

#### Padding

Define os ajustes ao substituir um *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 dos tags avaliados quando da geração de um alarme, por exemplo. Para tanto, acrescente o nome do 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

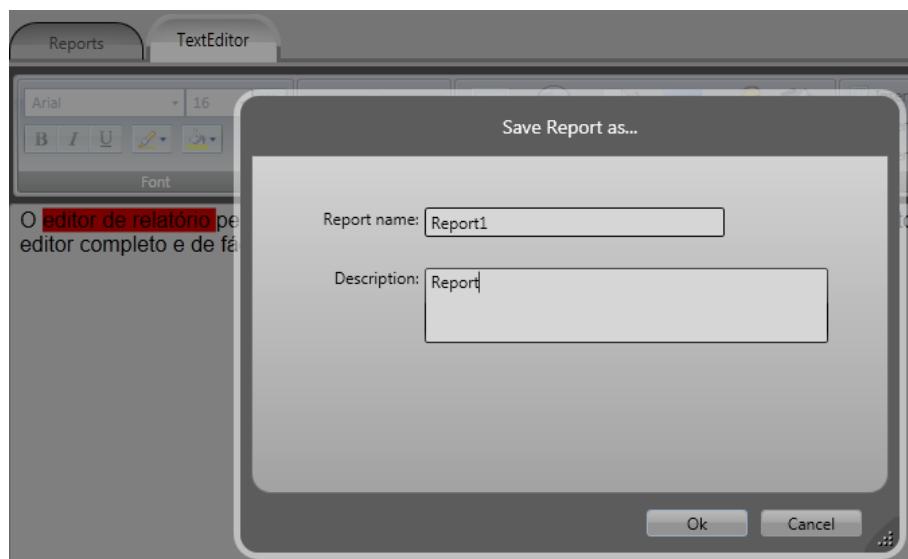
#### TextEditor

Esta guia contempla um editor de texto para criação de relatórios conforme ilustrado na Figura 6-69.



**Figura 6-69. Editor de Texto**

A Figura 6-70 mostra a edição e salvamento de um relatório com o auxílio do editor de texto embutido no software.



**Figura 6-70. Edição e Salvamento no Editor de Texto**

A Figura 6-71 mostra a consolidação do formulário editado na figura anterior com algumas configurações disponíveis no menu de edição de relatórios.

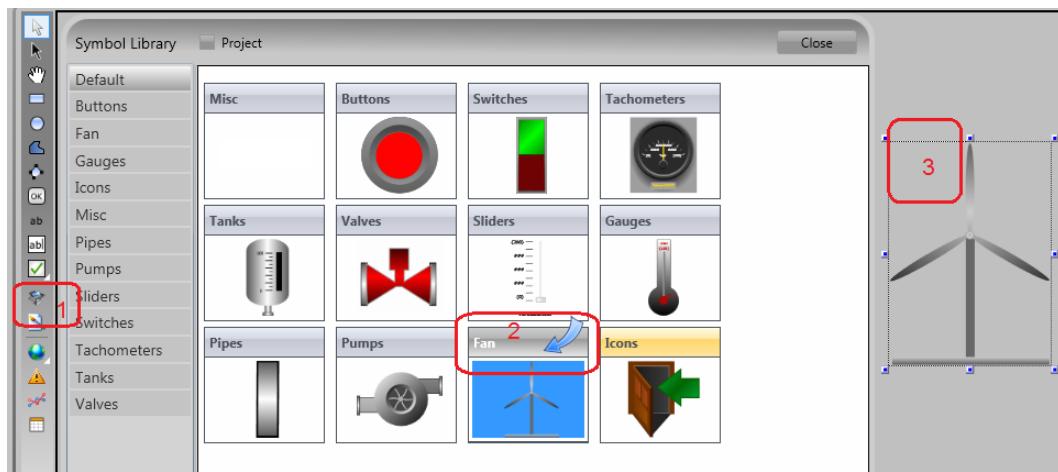
Drag a column header here to group by that column.				
Name	Padding	SaveFormat	SaveFileName	Description
Report1	PadLeft	PDF	Test_Report	Report

**Figura 6-71. Configuração de um Relatório**

## Menu Draw



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. 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 6-72 mostra a barra de ferramentas vertical.

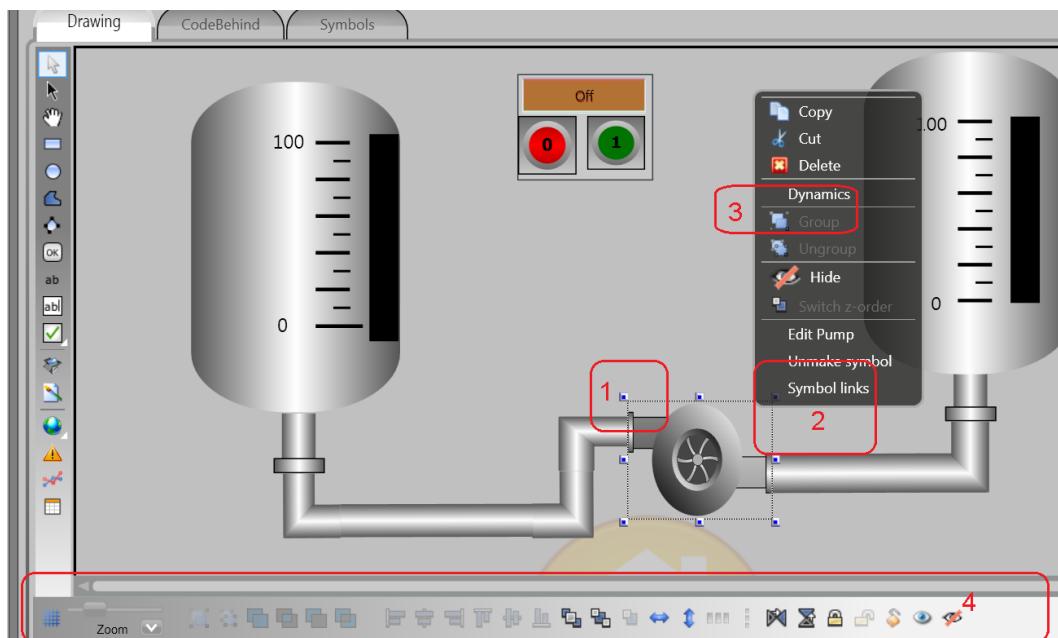


**Figura 6-72. Menu Draw e Barra de Ferramentas Vertical**

Os passos para a inclusão de um símbolo na tela estão numerados na Figura 6-78 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

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 6-73.

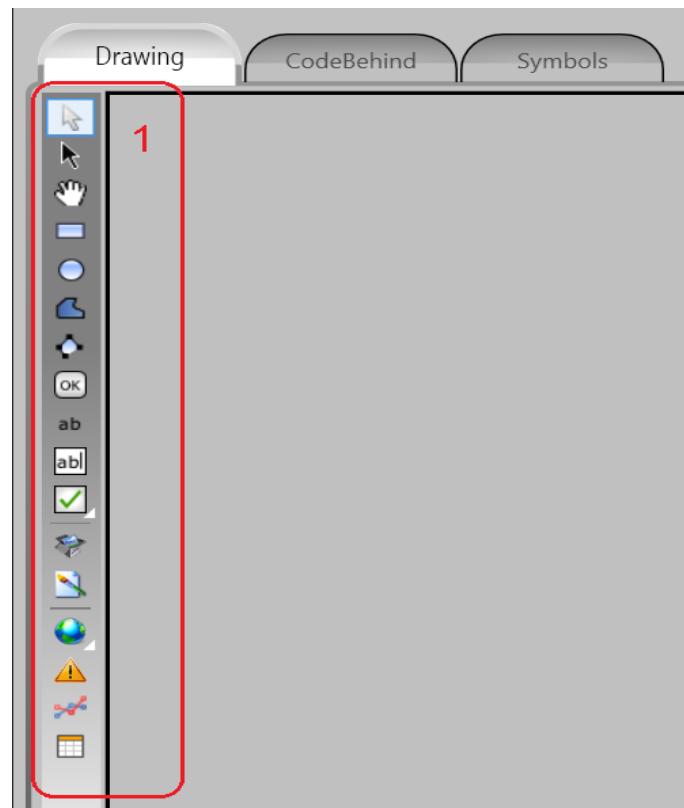


**Figura 6-73. Menu Draw e Barra de Ferramenta Horizontal**

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 6-74 estão detalhados na sequência.



**Figura 6-74. 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

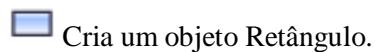
Use esta ferramenta para selecionar um objeto dentro de um grupo e modificar suas propriedades. Clique uma vez no objeto para selecioná-lo. É possível adicionar, remover e modificar os pontos em uma *Polyline* através desta ferramenta. Para mover o ponto, selecione-o com um clique e pressione o botão 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.



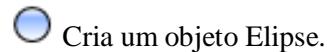
Ferramenta Mão

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.

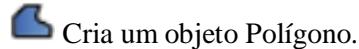
## Objetos Geométricos



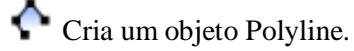
Cria um objeto Retângulo.



Cria um objeto Elipse.



Cria um objeto Polígono.



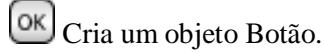
Cria um objeto Polyline.

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

Para mais informações de como adicionar, modificar e remover pontos após a criação de um *polígono* ou *polyline* está descrito na seção Ferramentas de Seleção.

## 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.

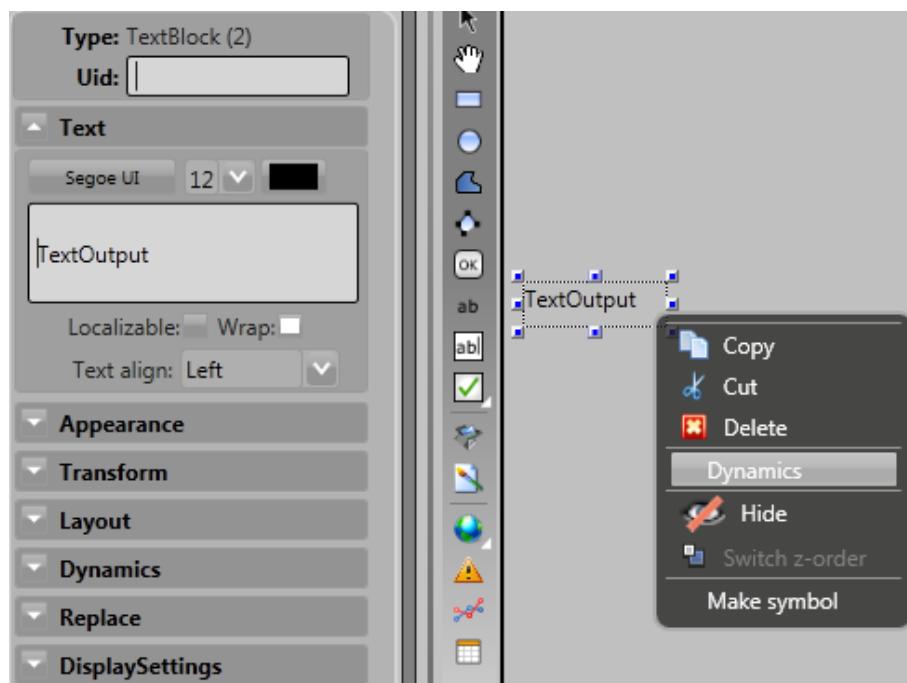
 Cria uma DateTimeTextBox.

 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 6-75.



**Figura 6-75. 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 6-76.

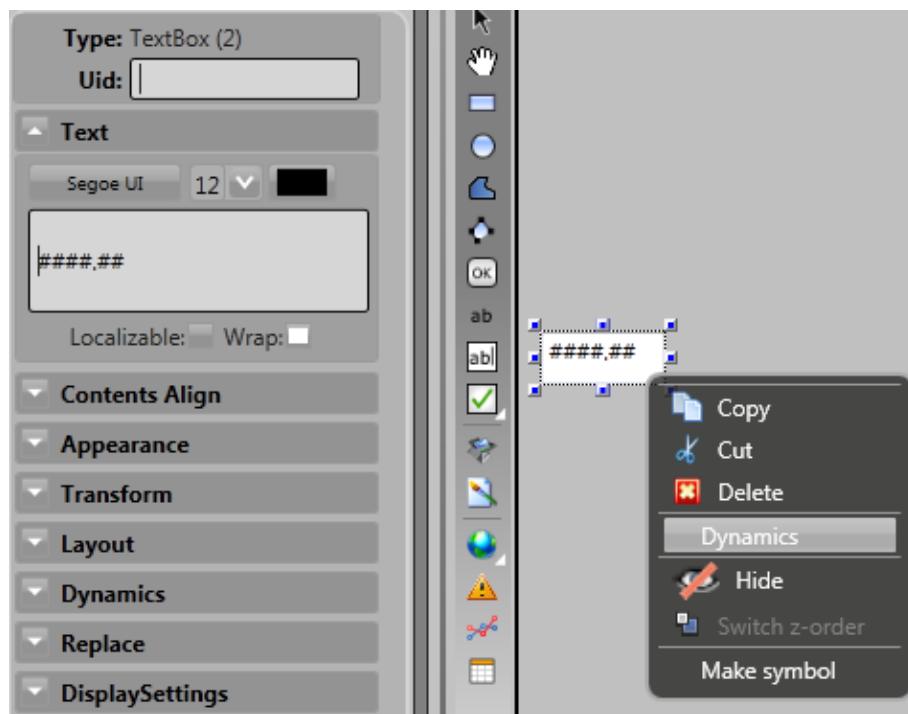


Figura 6-76. Objeto TextIO

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 o Tag, dê um clique duplo no objeto *TextIO* e, na janela de configuração 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.

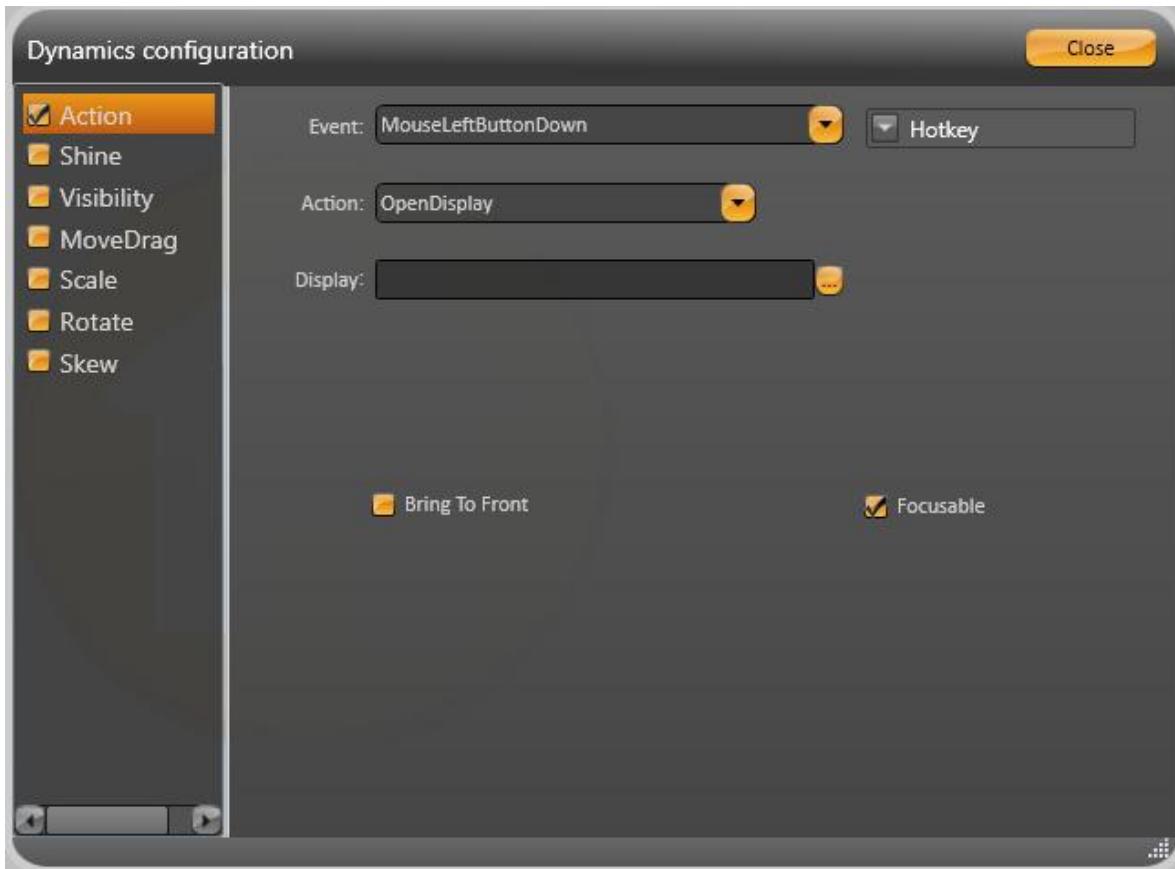


Cria um objeto ChildDisplay component. Este objeto não permite que a mesma tela seja aberta mais de uma vez.

-  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 6-77. 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*.

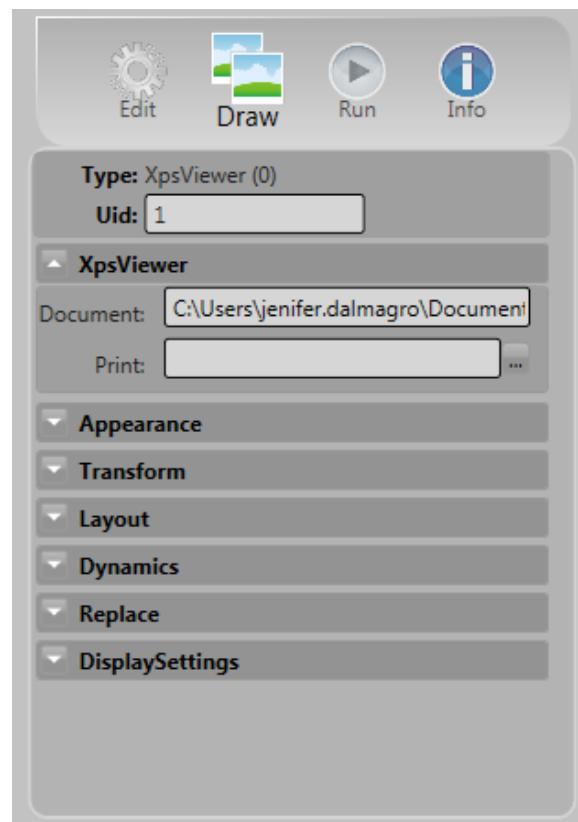


Figura 6-78. Configuração do XPS Viewer

*Circular Panel*

Para configurar um Painel Circular basta dar um clique duplo sobre o objeto, então aparecerá a tela mostrada na Figura 6-79.



**Figura 6-79. Configuração de Painel Circular**

**Control Name:** Define um nome para o controle, desta forma ele poderá ser acessado no script CodeBehind. Ver 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 6-80.



Figura 6-80. 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 6-81.



**Figura 6-81. 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:** Seleccione 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.

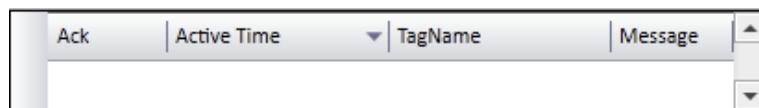
#### **Alarme**

Objeto alarme 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 6-82.



**Figura 6-82. Janela de Alarmes**



**Figura 6-83. 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. Ex:

```
[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 (Legacy)

O objeto Tendência mostra o registro do historiador de um tag.

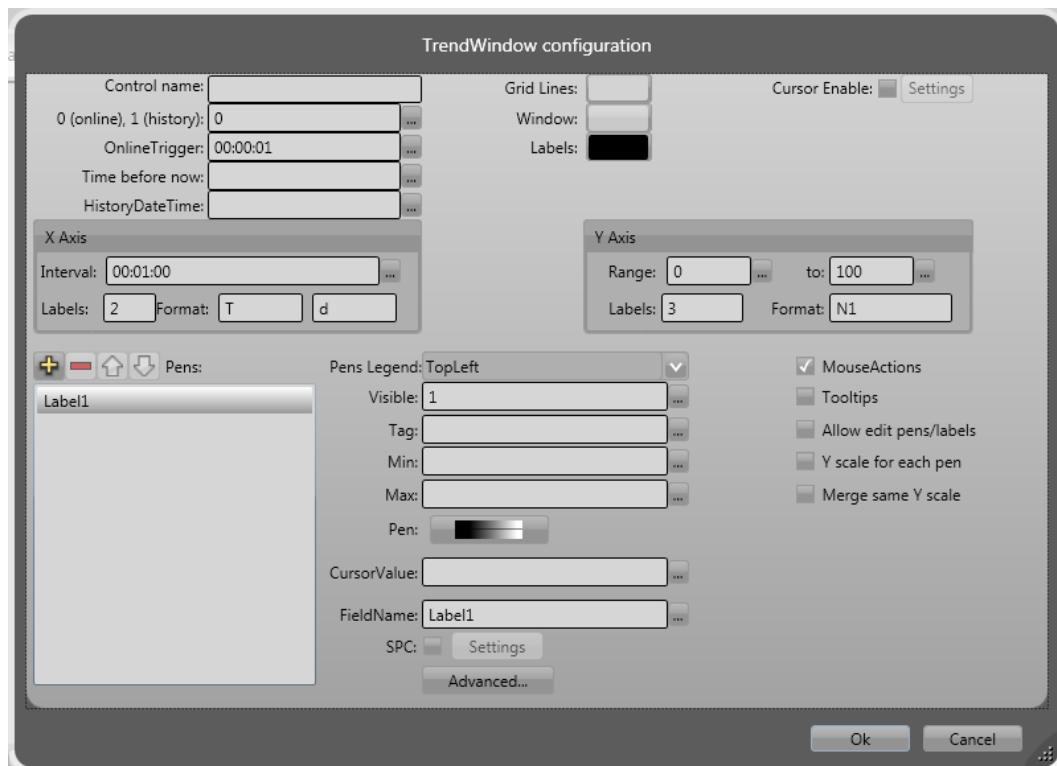


Cria uma janela de tendência.

**Atenção:**

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

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



**Figura 6-84. 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. Ex:

```
Tag.onLineHist
```

**OnlineTrigger:** Taxa de atualização do controle de tendência online. Ex:

```
00:00:01 ou {Tag.trendTimeSpan}
```

**Time Before now:** Tempo mostrado na janela de tendência antes da ultima medida. Ex:

```
00:00:01 ou {Tag.trendTimeSpan}
```

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

```
{Tag.initialTrendHistory}
```

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

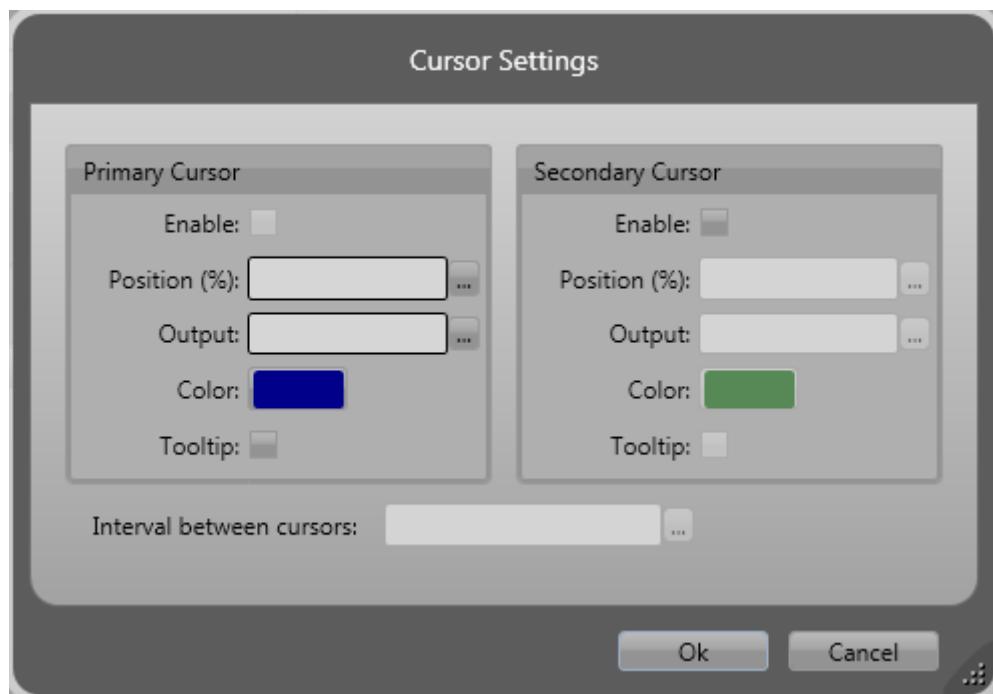
**Window:** Define a cor da tela de fundo do controle de tendência.

**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 6-85, cuja descrição dos campos de configuração seguem.



**Figura 6-85. 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. Ex:

Tag.cursorPos

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

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.

**Nota:**

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

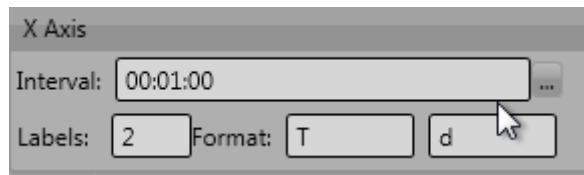


**Figura 6-86. Eixo Y**

**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 dos tags. Ex: N1 (número com 1 casa decimal).



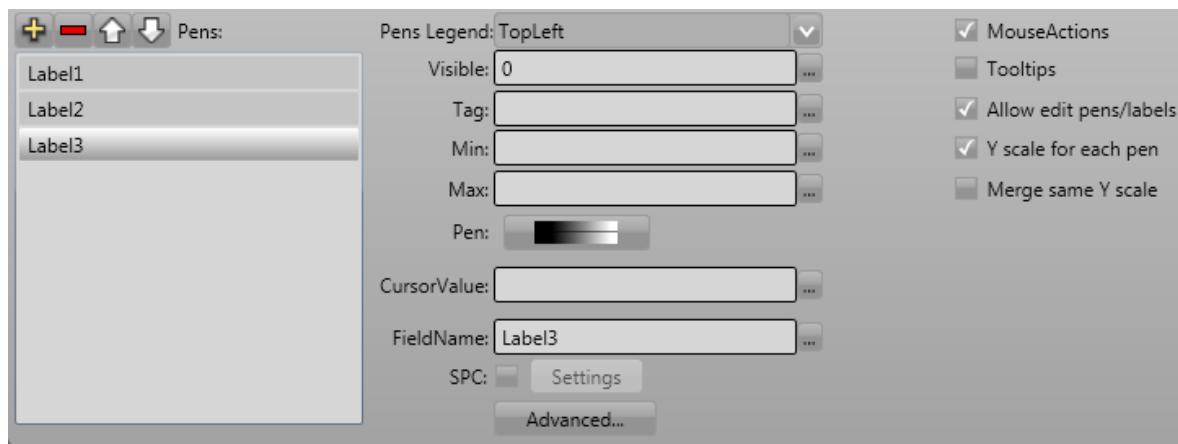
**Figura 6-87. Eixo X**

**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 dos tags. Ex:

T (Tempo) para o formato da primeira linha, d (data abreviada) para o formato da segunda linha.



**Figura 6-88. 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. Ex:

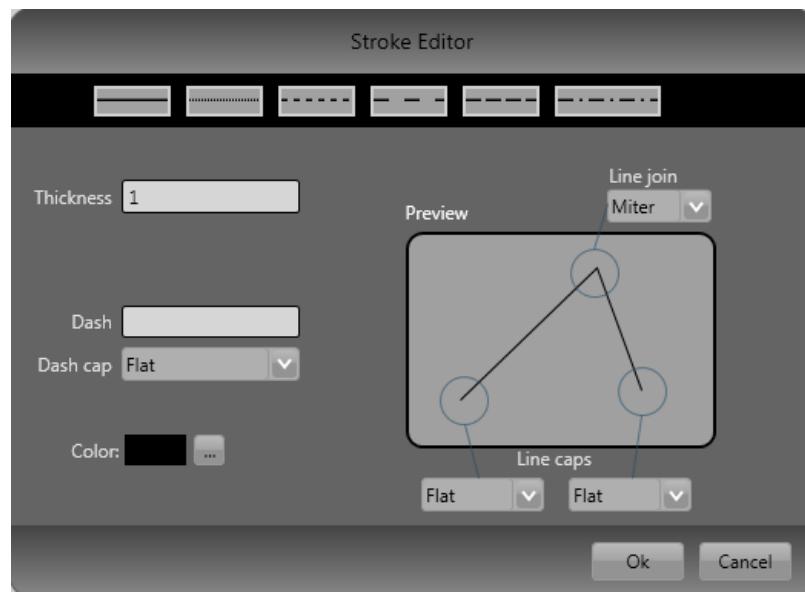
```
1 ou {showPen1}
```

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

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

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

**Pen:** Ao clicar neste botão é aberta a janela da Figura 6-89 onde é possível definir o estilo, cor e espessura da linha da caneta.



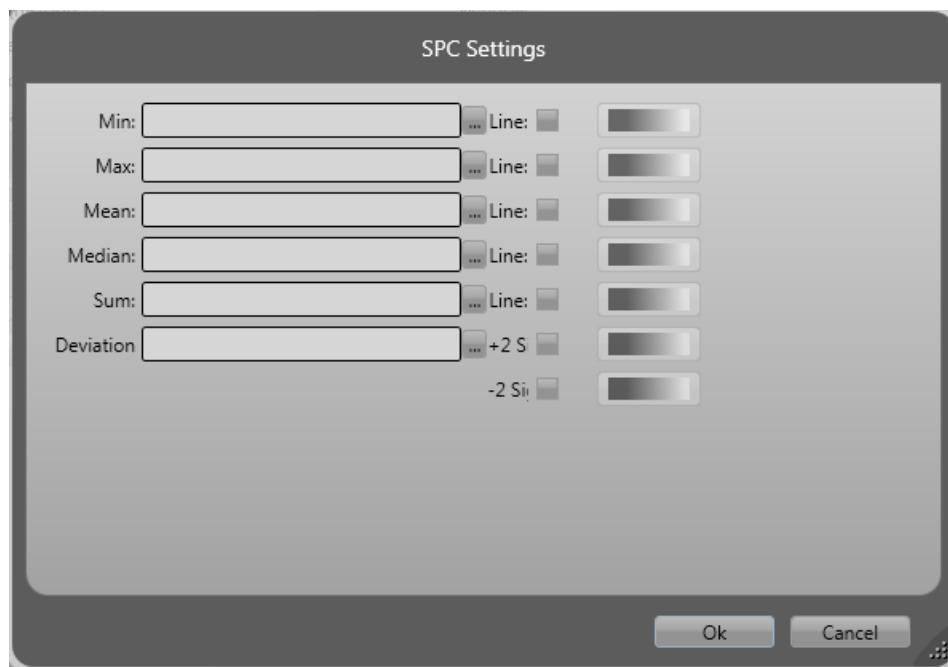
**Figura 6-89. Configuração de Estilo da Linha**

**CursorValue:** Define o tag que receberá o valor real do eixo Y, de acordo com a posição do cursor. Ex:

```
Tag.pen1CursorPosition.
```

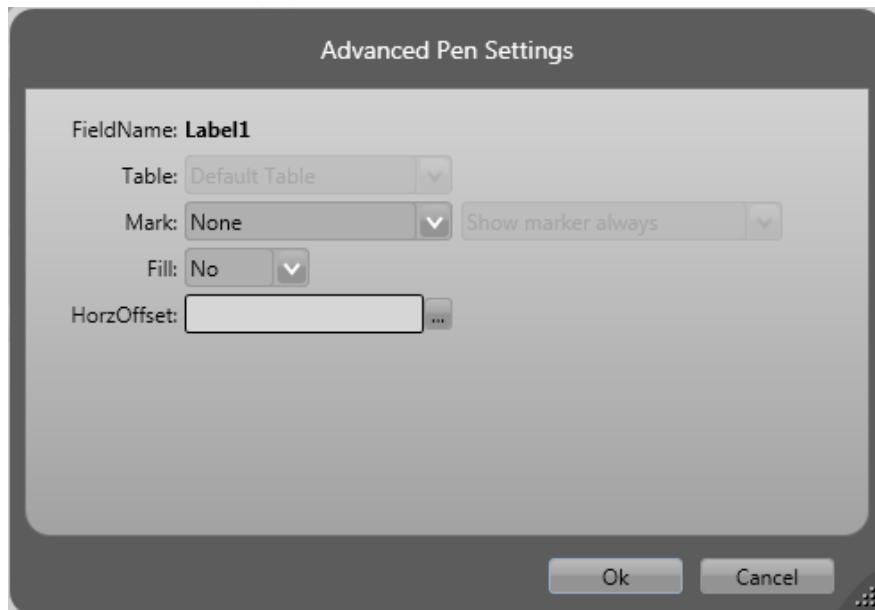
**FieldName:** Define o nome da caneta.

**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 6-90. SPC Settings**

**Advanced:** Abre a janela apresentada na Figura 6-91.



**Figura 6-91. Opções Avançadas**

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

**Advanced Pen Settings Table:** Pode escolher para pegar 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 um tag com o valor em TimeSpan para mostra somente naquela pena o dados passado (histórico).

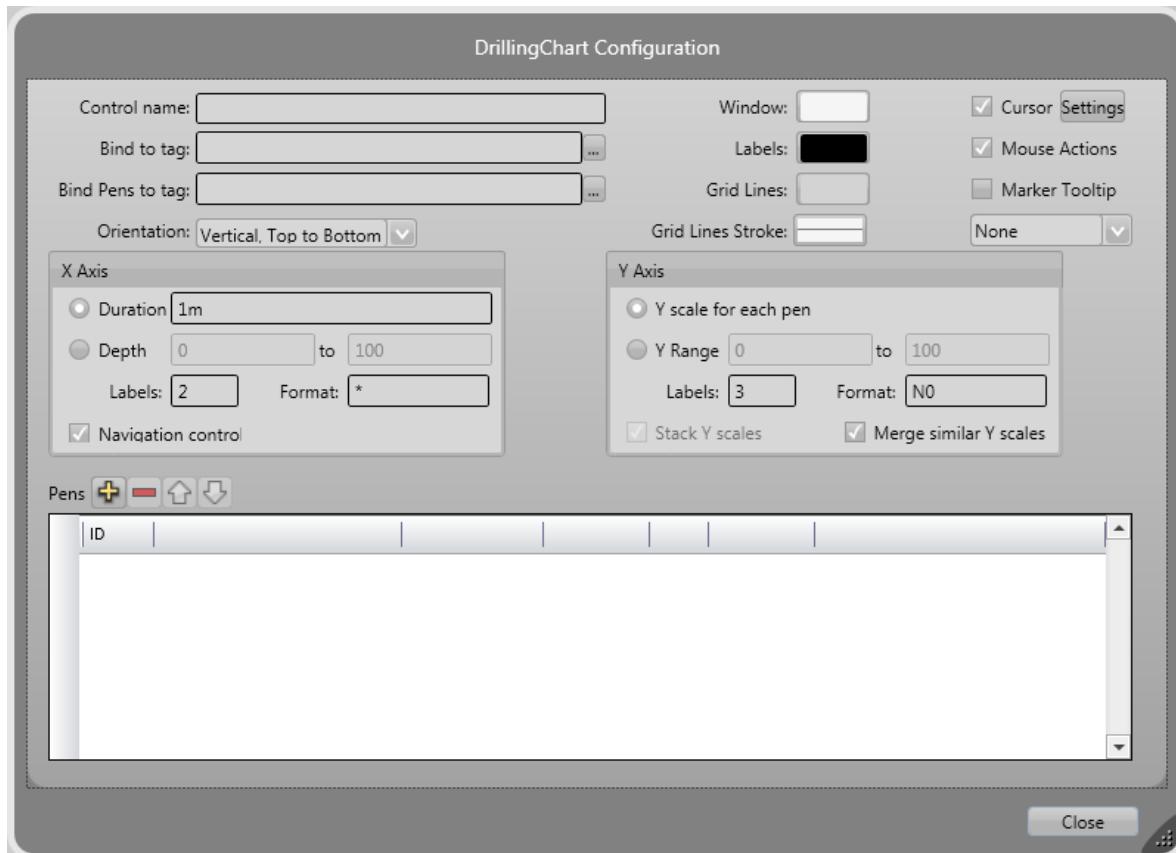
O controle de tendência permite que várias canetas sejam exibidas junto. Quando as suas canetas não estão na mesma faixa use as dicas abaixo para ajustar os dados no mesmo gráfico e assim obter uma melhor visualização. Se alguma caneta apresentar uma faixa inferior, 0 a 1, por exemplo, pode-se configurar 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 um tag.

 Cria uma janela de tendência.

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



**Figura 6-92. 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 dos tags. Ex: 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 dos tags. Ex: *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:** Os 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).

## Grade de Dados



Cria uma janela DataGrid.

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



**Figura 6-93. 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 esta selecionada. A numeração de linhas começa em 0.

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

**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 DataGrid.

**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. Ex:

```
[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:** Selecione 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 6-6. 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 6-94 ilustra essa configuração.



**Figura 6-94. Configuração Dinâmica**

Os itens que compõem a configuração dinâmica como mostrado na Figura 6-94 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

Ex: 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 o Tag ou a propriedade que receberá o valor da expressão

**Hotkey:** teclas de atalho rápido.

Exemplos:

1. Executar Expressões:

Somar dois valores e passar o resultado para outro Tag.

Expressão:

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

Resultado:

```
Tag.totalQuantity
```

2. Incrementar um Tag

Expressão:

```
TagCounter + 1
```

Resultado:

```
TagCounter
```

3. Incrementar um Tag (0 - 10)

Expressão:

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

Resultado:

```
TagCounter
```

Estes campos relacionados às expressões estão ilustrados na Figura 6-94.

### Shine

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

**IsMouseOver:** entre com um 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:** configurações.

- **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 o mesmo dependendo do formato realiza arredondamentos nos valores, por exemplo, Tag.analogInt1= 20,006, definido formato com 0.00, o valor exibido será 20,01.

**Nota:**

Se o texto é um valor de Tag ou uma propriedade, ele deve estar entre chaves, por exemplo:

```
{Tag.analogInt1}
```

## HyperLink

Abre um hyperlink.

**HyperLink type:** escolha uma das opções http, ftp, file, mailto, telnet.

**Url:** configure o url que será aberto.

**Hotkey:** teclas de atalho rápido.

#### *Securitiy*

Define as permissões do objeto em tempo de execução.

**Disable:** entre com um 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 com as seguintes configurações...

- **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. Ex: "White" ou "#FFFFFF"

Exemplo de uso de limites:

1 - Vermelho

10 - Azul

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.

Quando 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 com as seguintes configurações...

- **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. Ex: "White" ou "#FFFFFF"

Exemplo de uso de limites:

1 - Vermelho

10 - Azul

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.

Quando o valor for maior que 10, o objeto terá a cor azul.

#### *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 com as seguintes configurações...

- **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. Ex: "White" ou "#FFFFFF"

Exemplo de uso de limites:

1 - Vermelho

10 - Azul

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.

Quando o valor for maior que 10, o objeto terá a cor azul.

#### *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 6-7. Gráfico de Barras****Visibility**

Altera a visibilidade e opacidade do objeto dinamicamente.

**Visible:** entre com um tag, propriedade ou expressão retornando um valor. Considere o valor resultante:

0 - objeto será visível

>0 - objeto será oculto

**Opacity:** configurações.

- 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:** configurações.

- 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:** configurações.

- 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:** configurações.

- 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:** configurações.

- 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:** configurações.

- 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 Ex: 0 a 100
- Skew (°): os valores mínimos e máximos de inclinação do eixo X Ex: 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. Ex: 0 a 100
- Skew (°): o ângulo de inclinação do eixo Y mínimo e máximo. Ex: 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:

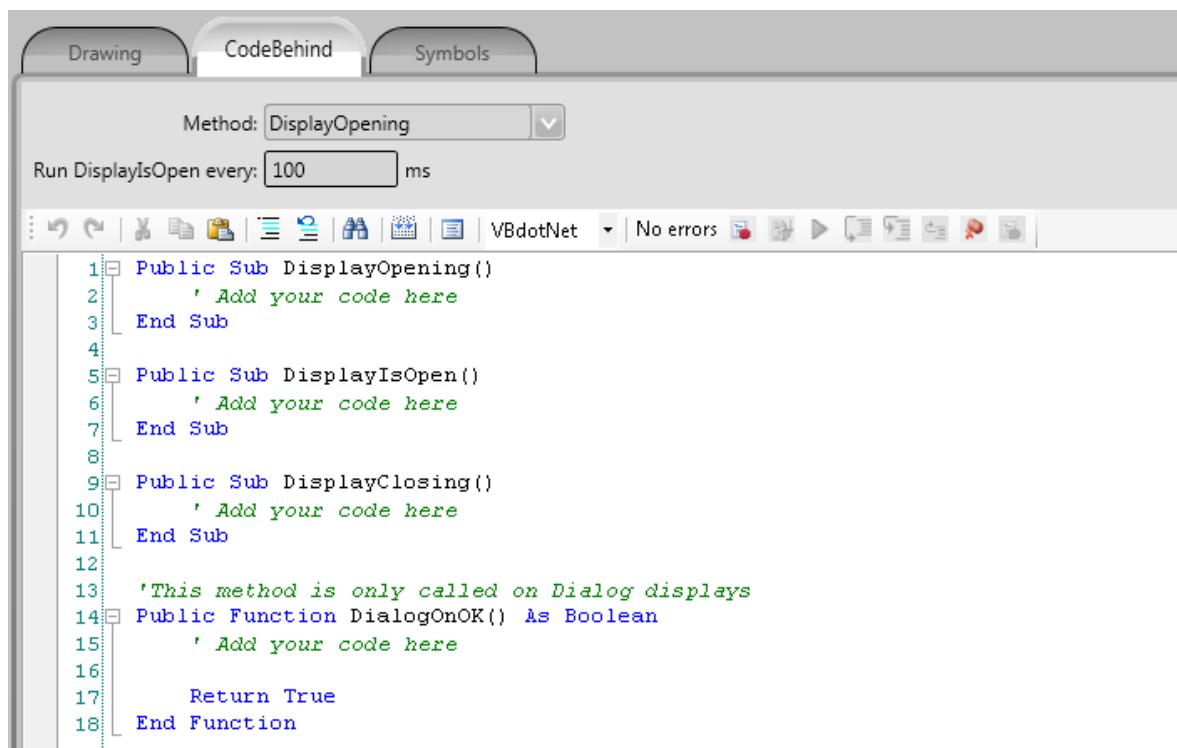
- ShowObjectNames: o conteúdo do campo *Expression* é mostrado exatamente como está
- ShowPlaceHolders: os caracteres ### são mostrados, a quantidade de caracteres é definida pelo campo *MaxLength*

**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 6-95.



```

1  Public Sub DisplayOpening()
2      ' Add your code here
3  End Sub
4
5  Public Sub DisplayIsOpen()
6      ' Add your code here
7  End Sub
8
9  Public Sub DisplayClosing()
10     ' Add your code here
11 End Sub
12
13 'This method is only called on Dialog displays
14 Public Function DialogOnOK() As Boolean
15     ' Add your code here
16
17     Return True
18 End Function

```

**Figura 6-95. 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

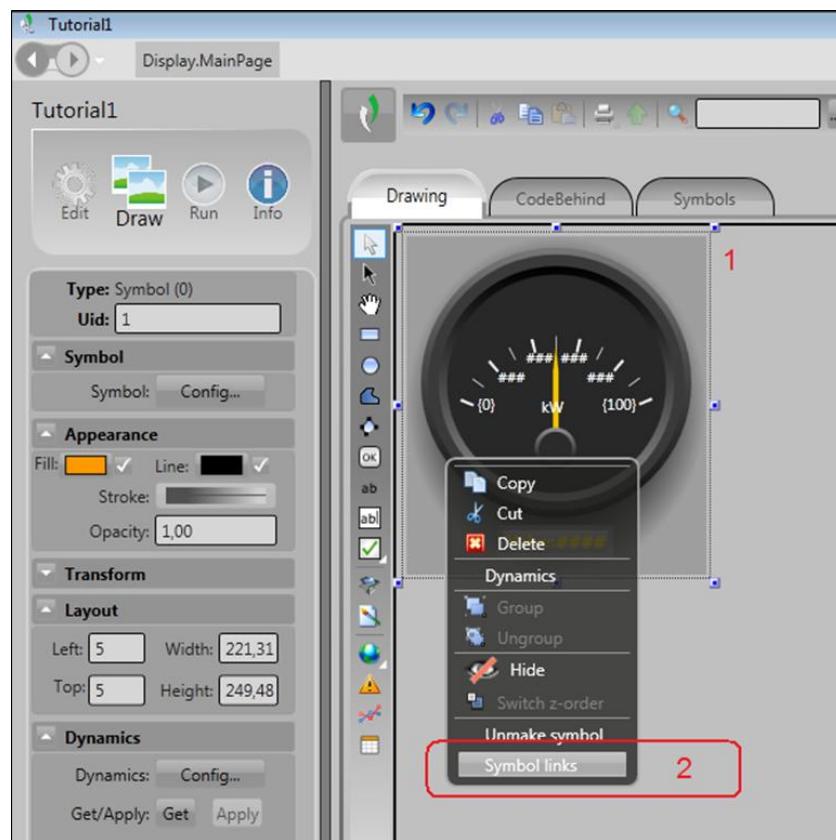
Retorno "1" permite o fechamento do diálogo. Retorno "0" evita que o diálogo seja fechado. Pode-se acrescentar variáveis .NET e métodos nesta página.

**Nota:**

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 6-96 mostra um símbolo e o menu de contexto associado.



**Figura 6-96. Menu Contexto do Símbolo**

Para simplificar o uso de símbolos em outras telas e aplicações pode-se definir rótulos em propriedades dinâmicas onde um *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 Símbolo

As imagens da Figura 6-97, Figura 6-98 e Figura 6-99 apresentam os passos necessários para criar um símbolo. Os procedimentos também estão descritos a seguir:

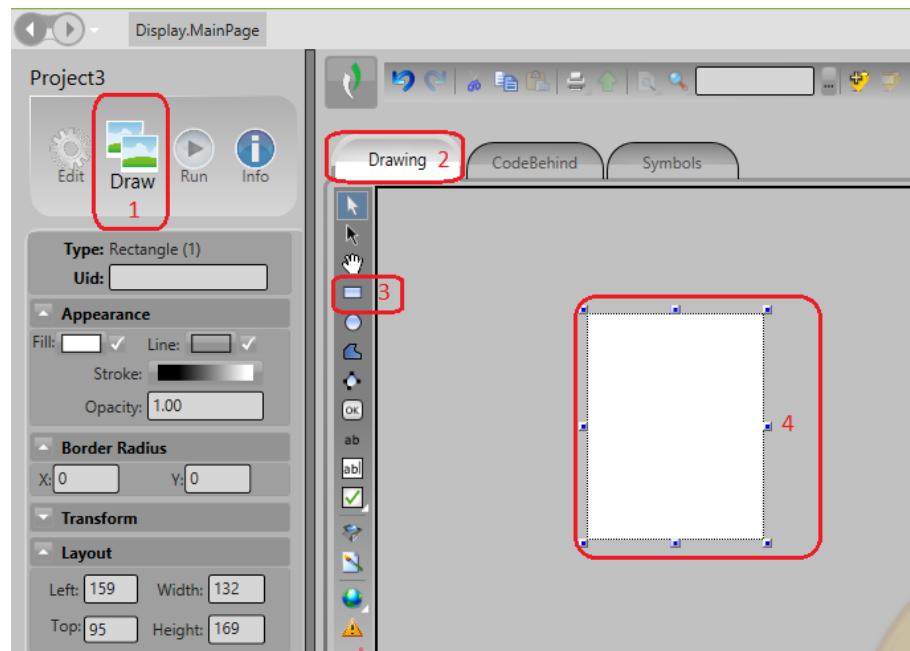


Figura 6-97. Criando um Símbolo

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

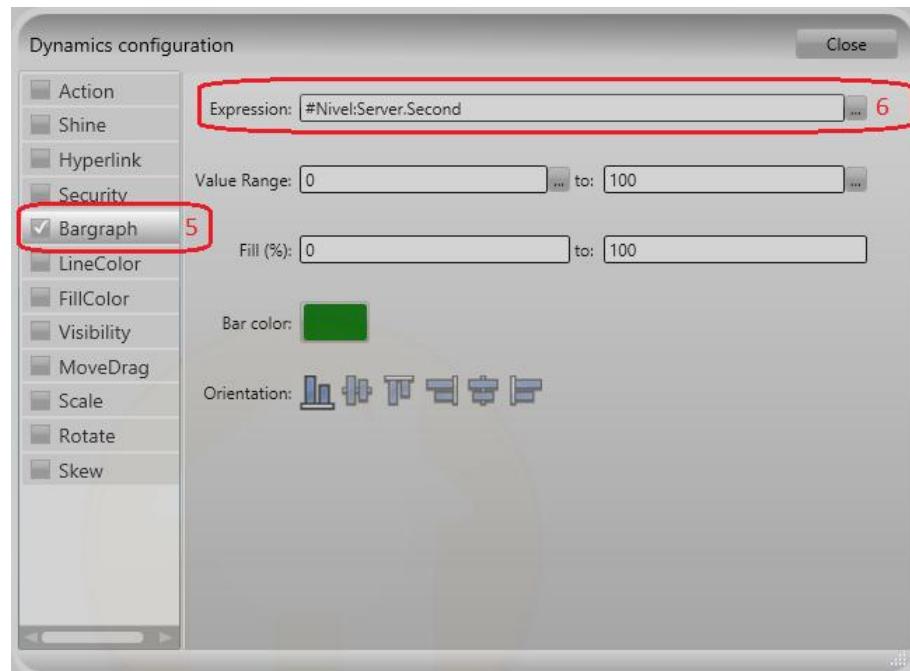


Figura 6-98. Configuração Dinâmica

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

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

**Nota:**

As propriedades de um símbolo são expostas através do uso de mnemônicos, que são:

```
#<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 6-99. Abrir Symbol Library**

13. Ir no ícone *Symbol Library* como mostrado na figura acima 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 o tag desejado, 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>*

8. 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

## Menu Run

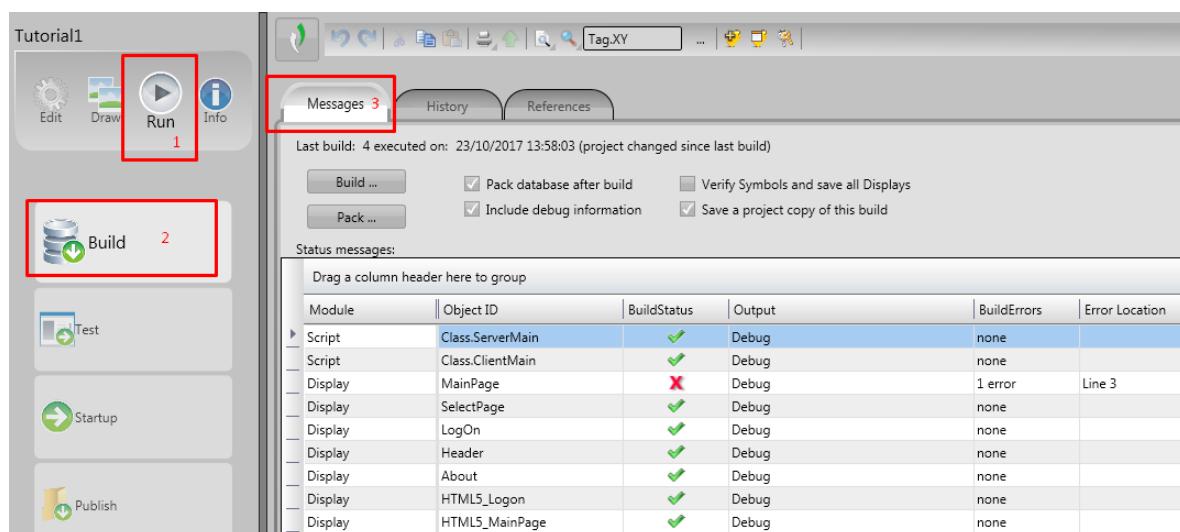
O ambiente Run fornece acesso a todos os recursos de execução do projeto

### 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.



**Figura 6-100. Guia Messages**

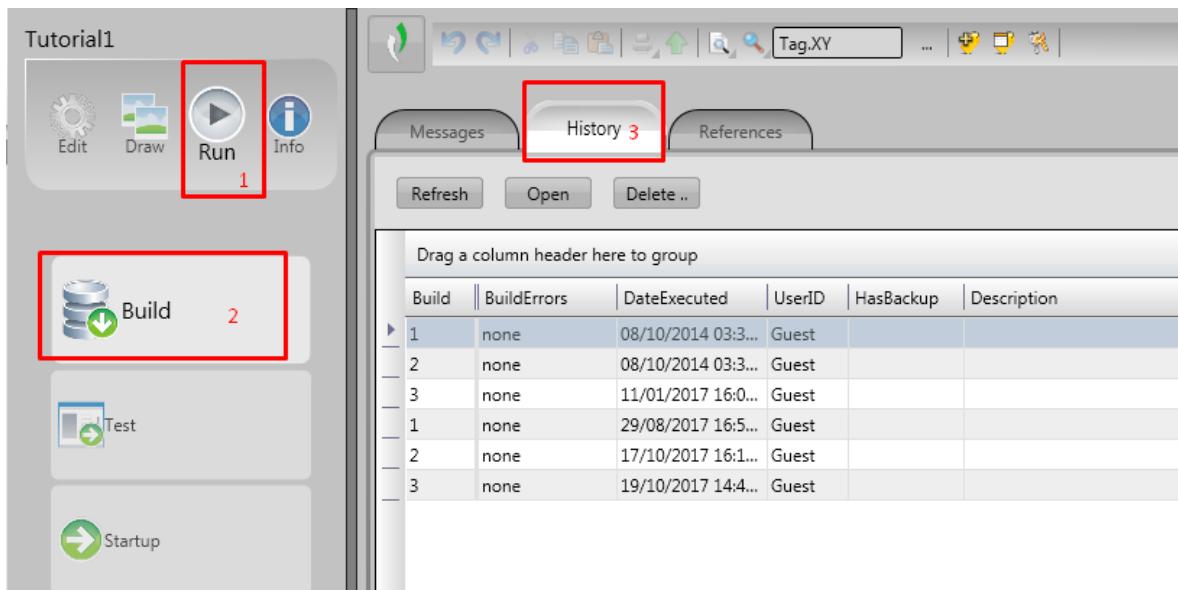
Os itens que compõem o menu *Messages* estão numerados na figura anterior e são descritos na sequência:

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 6-100 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 corrigí-lo, basta clicar duas vezes sobre a linha que apresenta o erro, pois esta ação abrirá o objeto com problemas.

## 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 6-101.



**Figura 6-101. Guia History**

Os itens que compõem a guia *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 *History* para exibição dos dados relativos ao histórico dos comandos de *Build*.

## References

A guia *References* deve ser utilizada para o registro de Referências à 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.

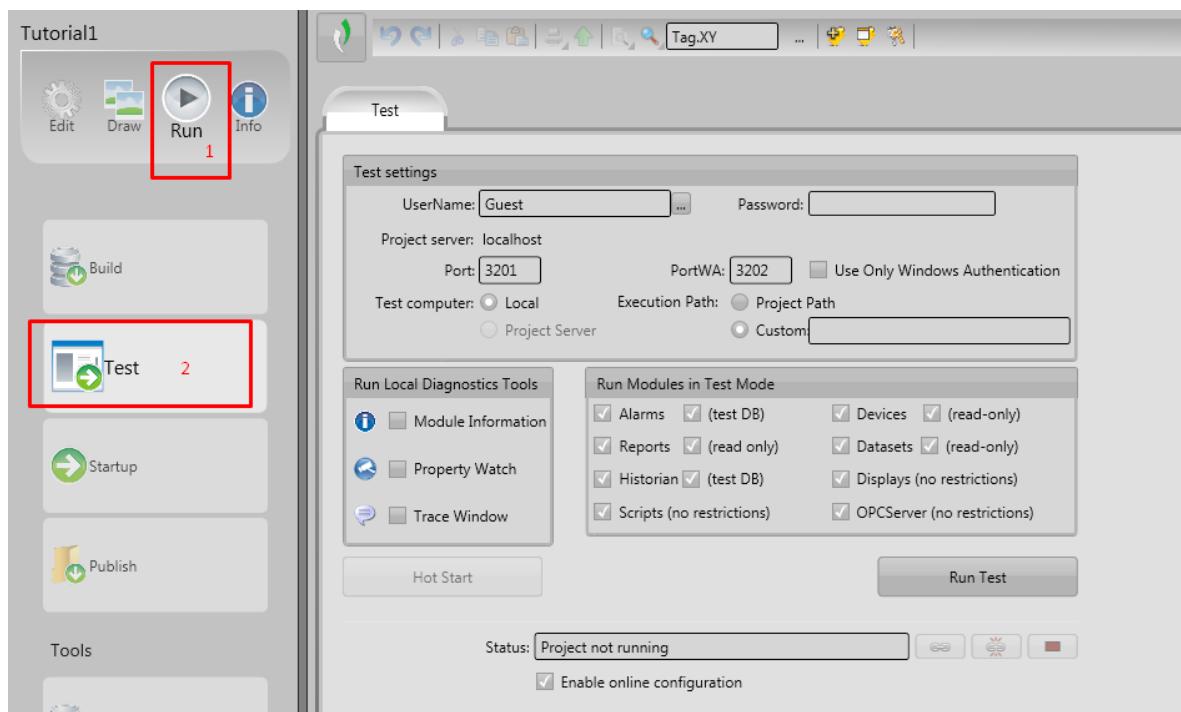
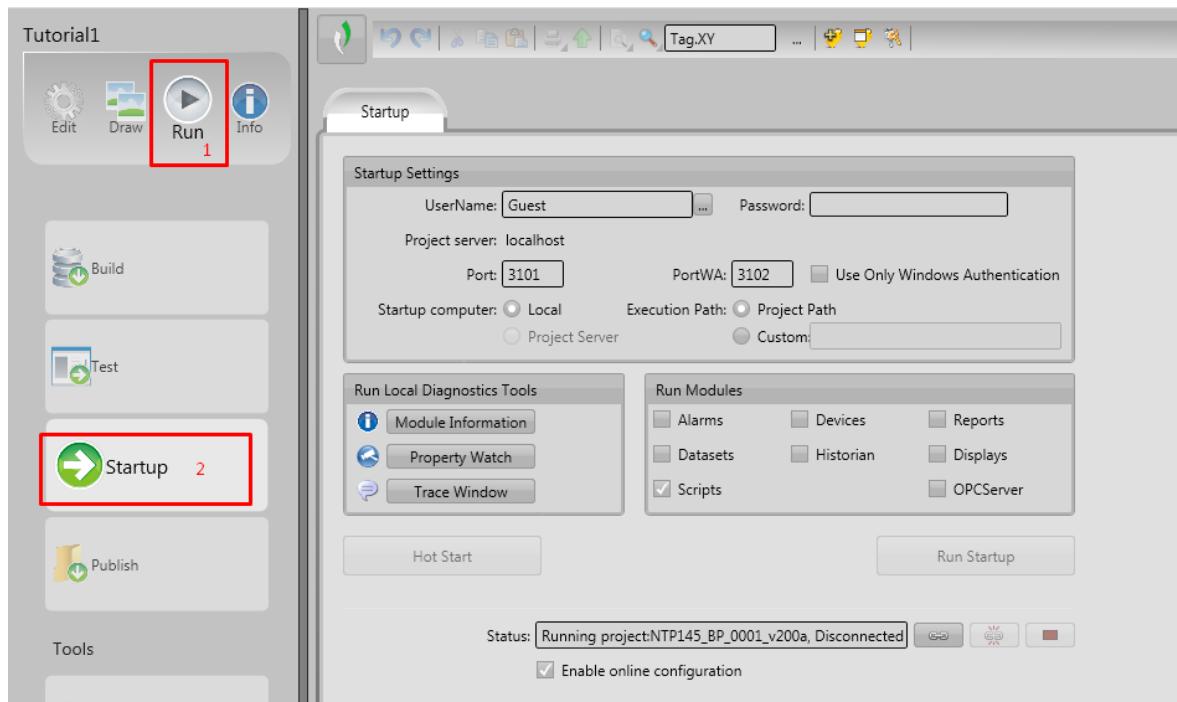


Figura 6-102. Menu Test

Os itens que compõem o menu *Test* estão numerados na Figura 6-112 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.
    - **Module Information:** se selecionado habilita a execução do **ModuleInformation**.
    - **Property Watch:** se selecionado habilita a execução do **PropertyWatch**.
    - **Trace Window:** se selecionado habilita a execução do **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.

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



**Figura 6-103. Menu Startup**

Os itens que compõem o menu *Startup* estão numerados na Figura 6-103 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.
    - **Module Information:** se selecionado habilita a execução do **ModuleInformation**.
    - **Property Watch:** se selecionado habilita a execução do **PropertyWatch**.
    - **Trace Window:** se selecionado habilita a execução do **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.

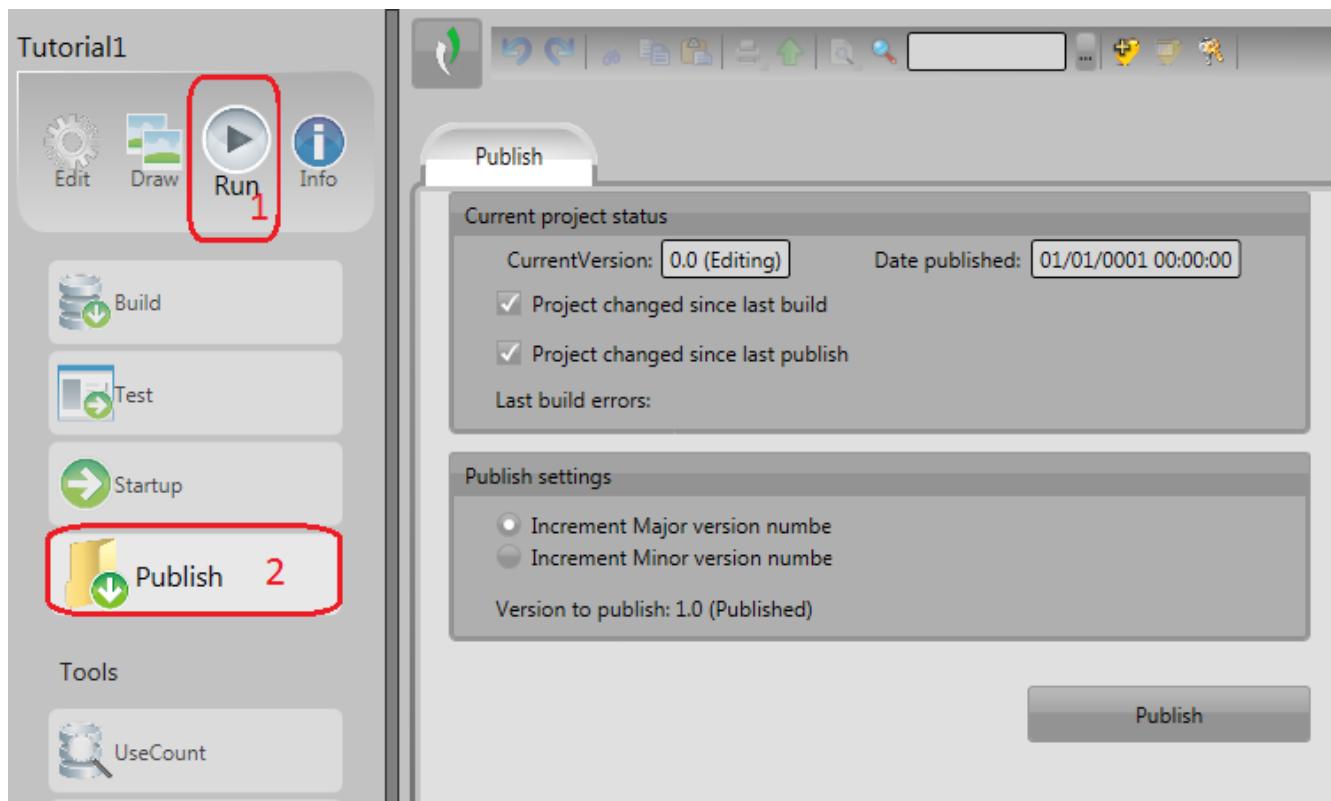


Figura 6-104. Menu Publish

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 apresentados os objetos do projeto e o número de vezes que estes são usados.

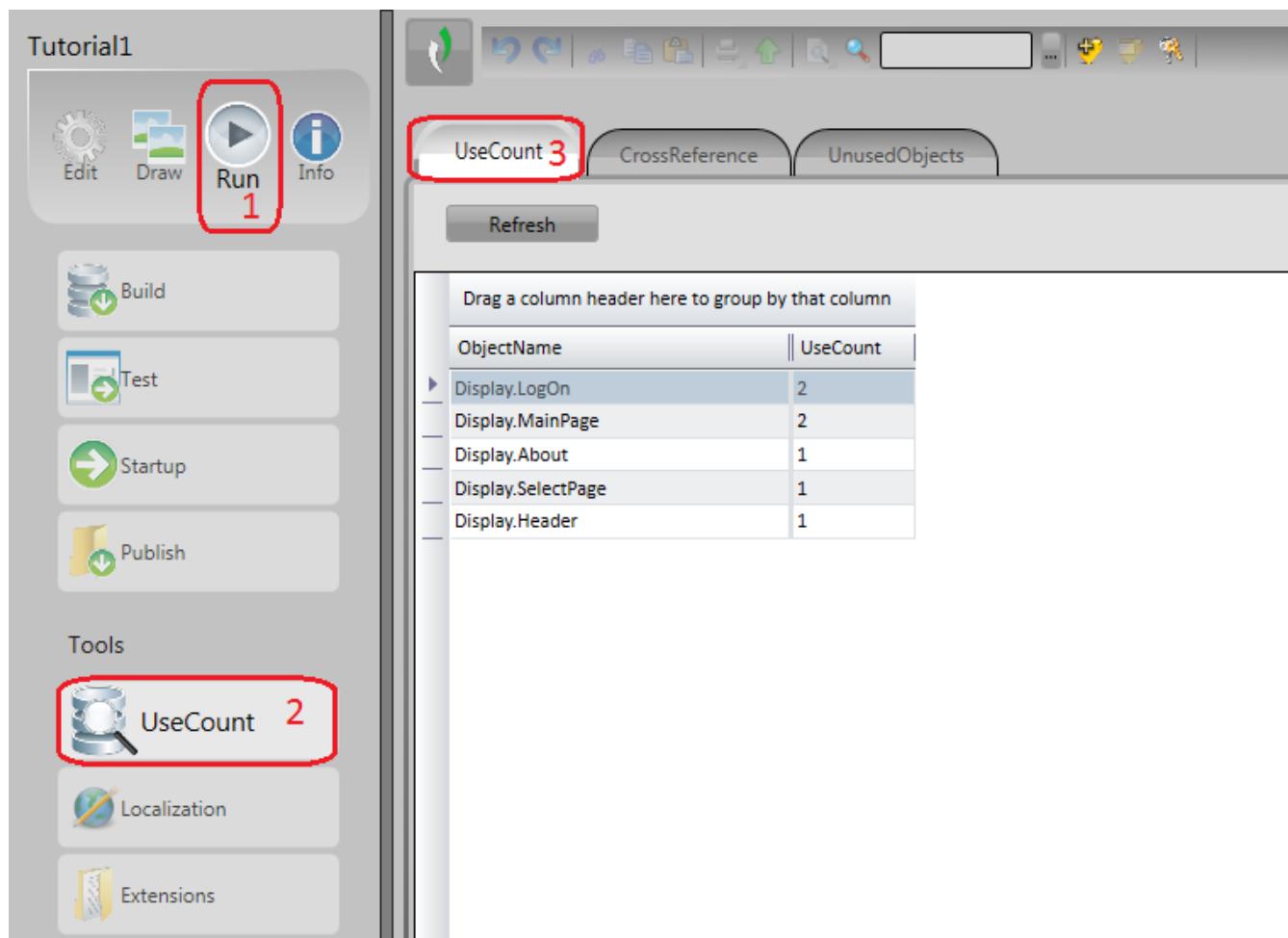


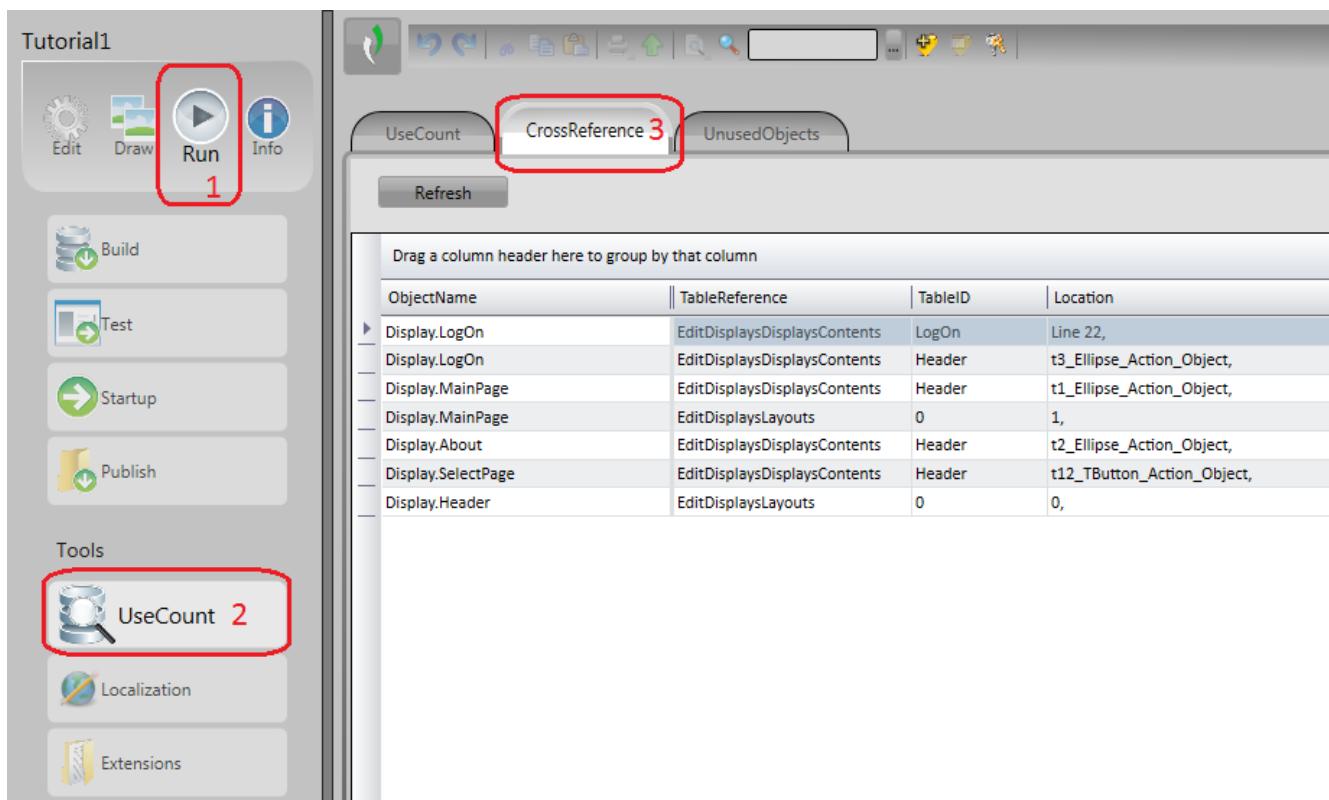
Figura 6-105. Menu UseCount

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

3. Selecionar a opção *Run*
4. Selecionar a opção *UseCount*
5. 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.



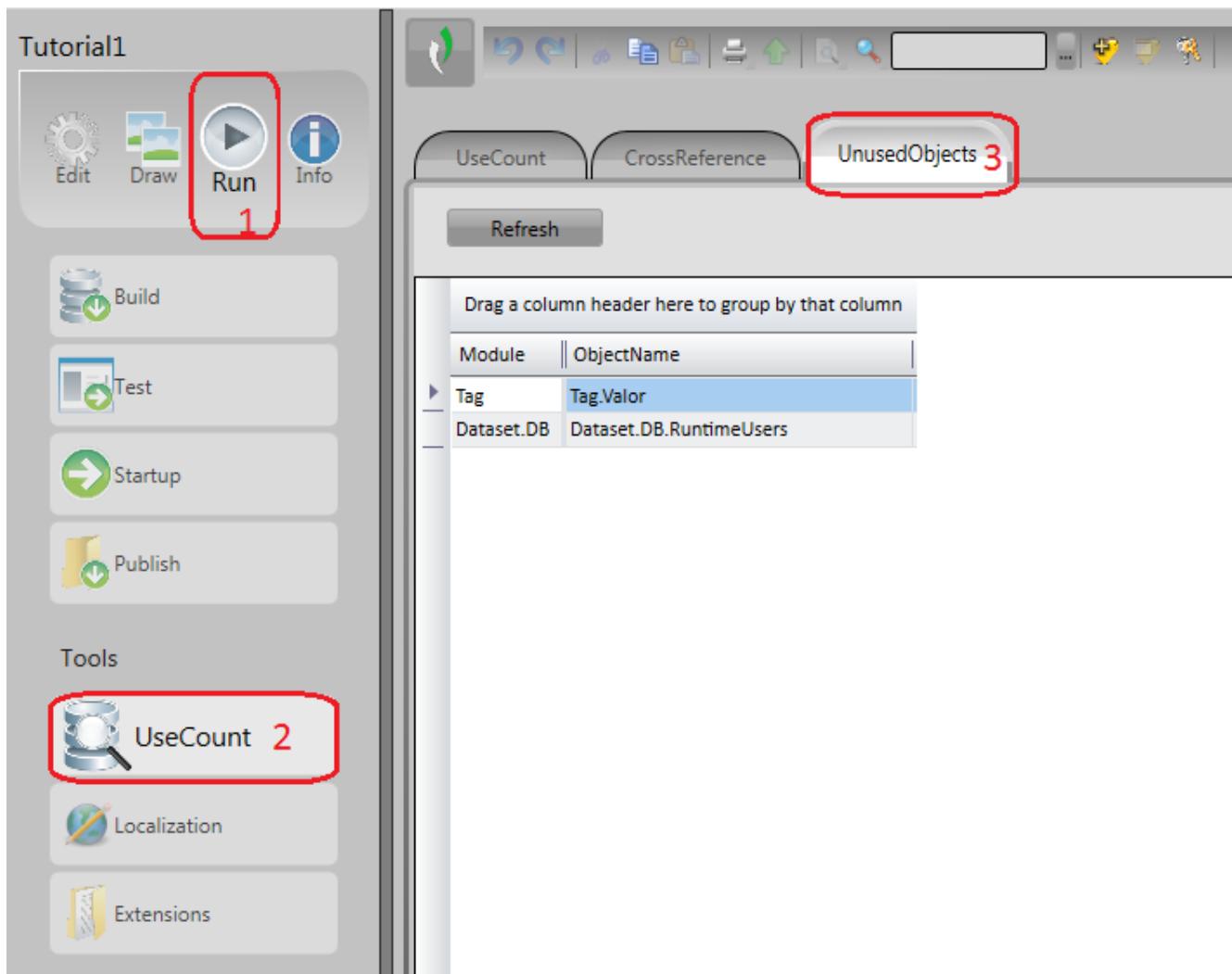
**Figura 6-106. Menu CrossReference**

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 6-107. Menu UnusedObjects**

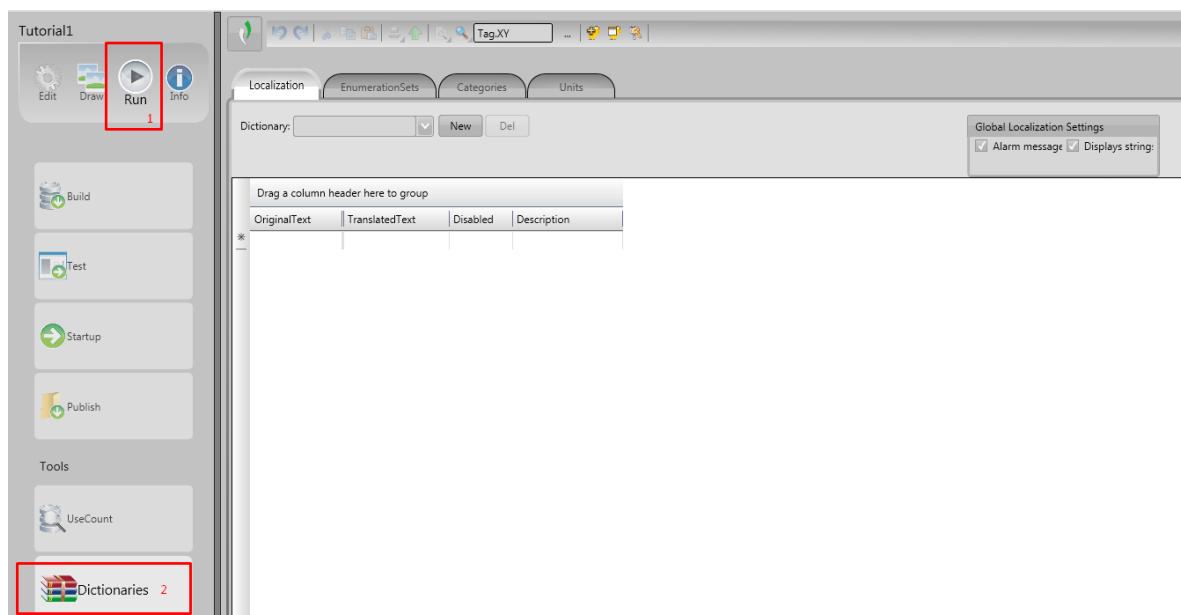
Os itens que compõem o menu *UnusedObjects* estão numerados na figura anterior e são descritos na sequência:

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 uma língua para ser usada em mensagens de alarme e nos textos de display.



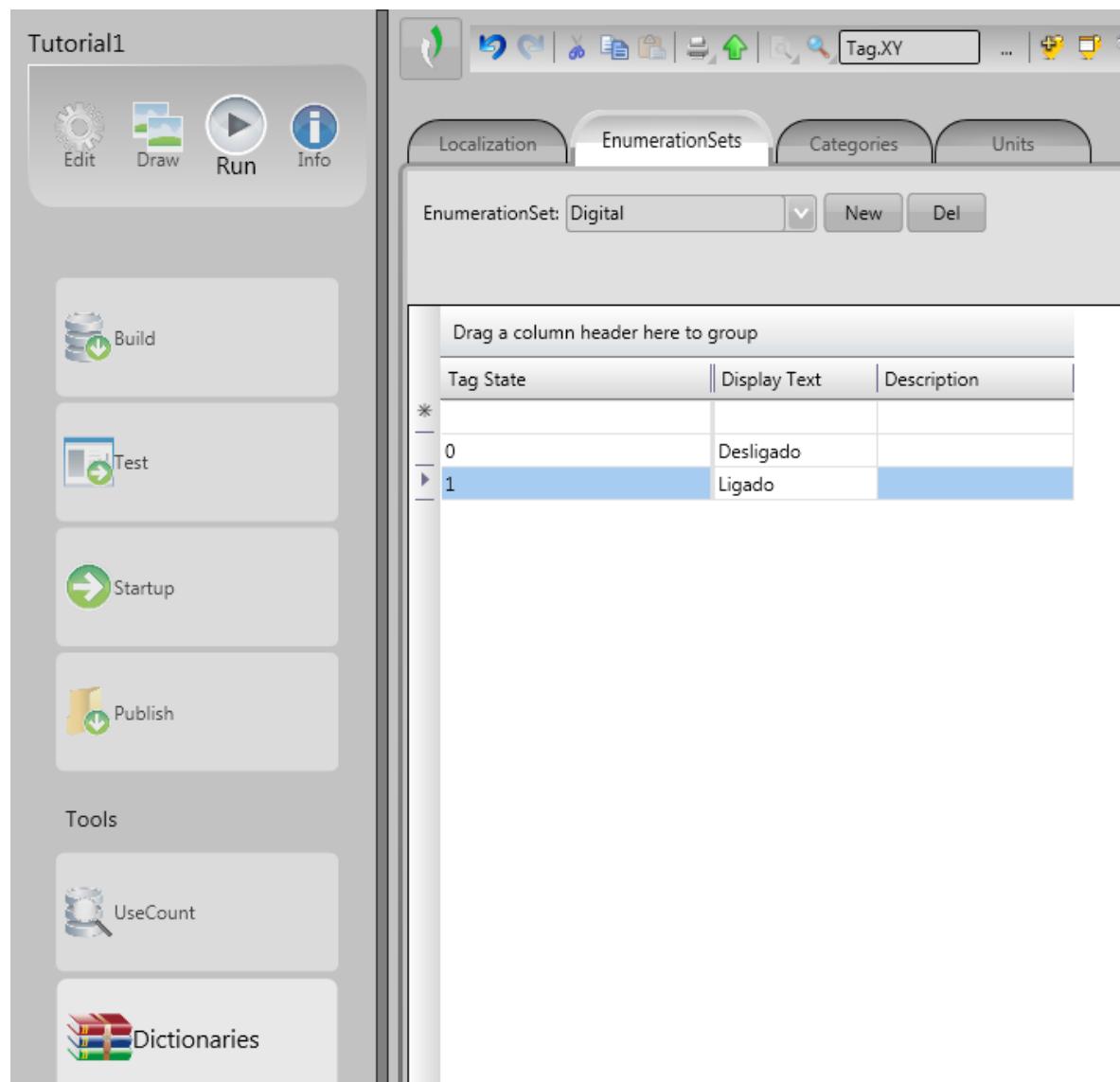
**Figura 6-108. Menu Localization**

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*

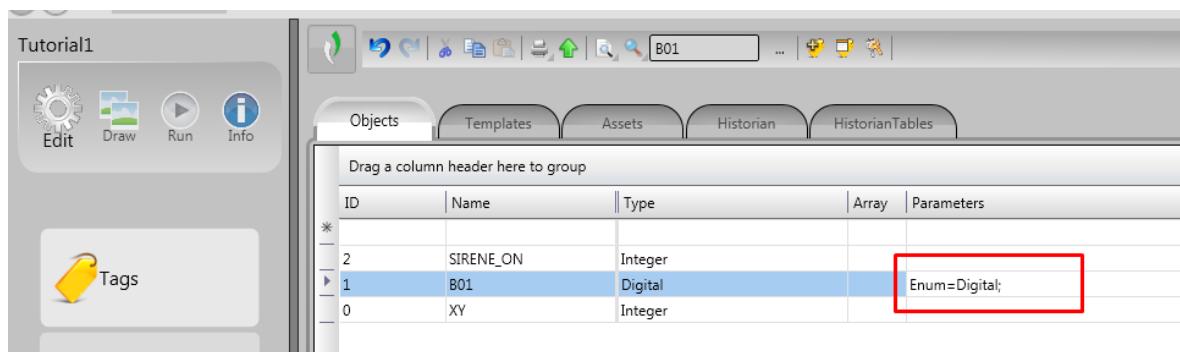
Você também pode usar os dicionários para traduzir os valores dos tags em texto. Por exemplo, um dispositivo pode ter um valor de 0 (desligado) ou 1(ligado). Ao invés de exibir "0" ou "1", você pode exibir as palavras "Ligado" ou "Desligado".



**Figura 6-109. 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 os Tags que usam os valores numéricos com o Conjunto de Enumeração, configurar a coluna Parâmetros para usar este conjunto.



**Figura 6-110. 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 traços, tags de alimentadores, 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.

Name	Title	Description
*	PROTECAO	Proteção
	MEDIDAS	Medição
	DISJUNTORES	Disjuntores

**Figura 6-111. Menu Categories**

Um mesmo 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, os tags que são de pontos de medição de tensão nos traços. No cadastramento destes pontos especificaríamos no campo *Categoria* destes tags estas duas categorias.

Uma vez a *Categoria* criada, passará a estar disponível na lista de existentes na coluna *Categoria* da aba *Objeto*, para a criação de Tags. Para cada tag escolha todas em que o mesmo se enquadra.

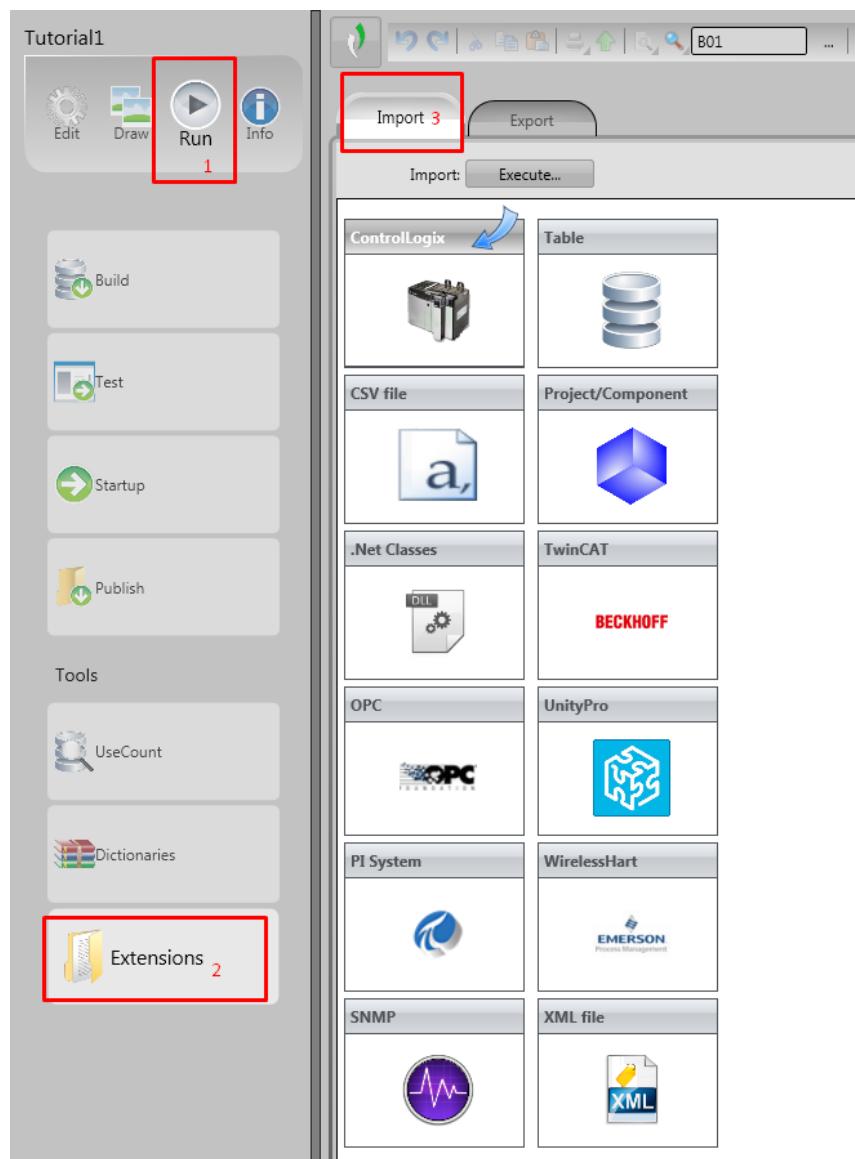
As Categorias de objeto também pode ser acessadas durante o tempo de execução para a filtragem e outras aplicações personalizadas, utilizando a propriedade Categoria dos objetos, como critérios para a escolha de objetos. Uma lista detodas as categorias definidas no projeto está disponível em tempo de 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.



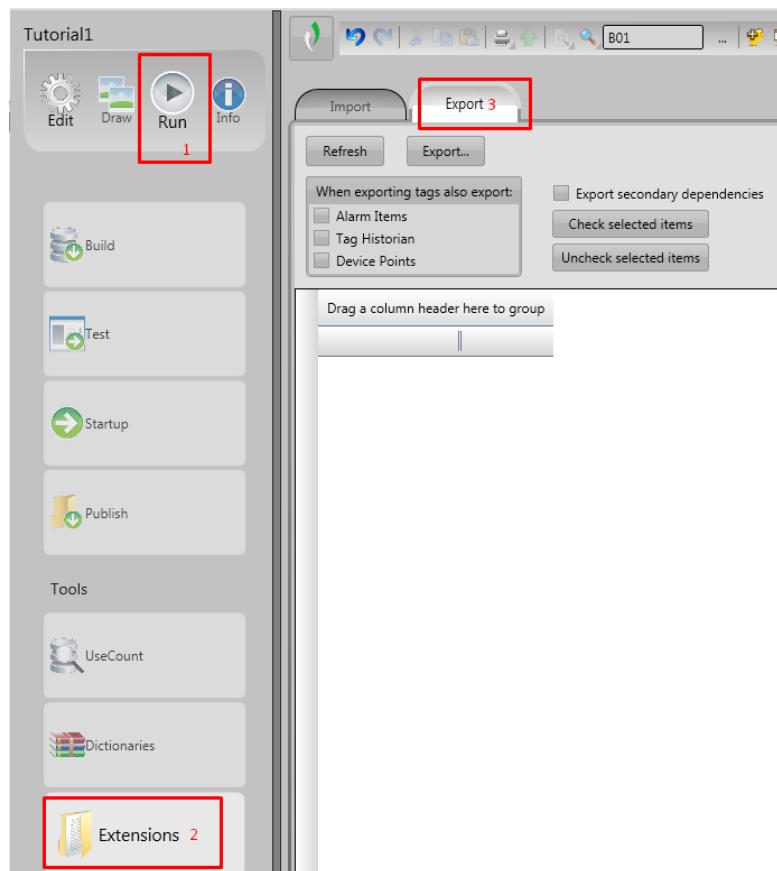
**Figura 6-112. Menu Import**

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 6-113. 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

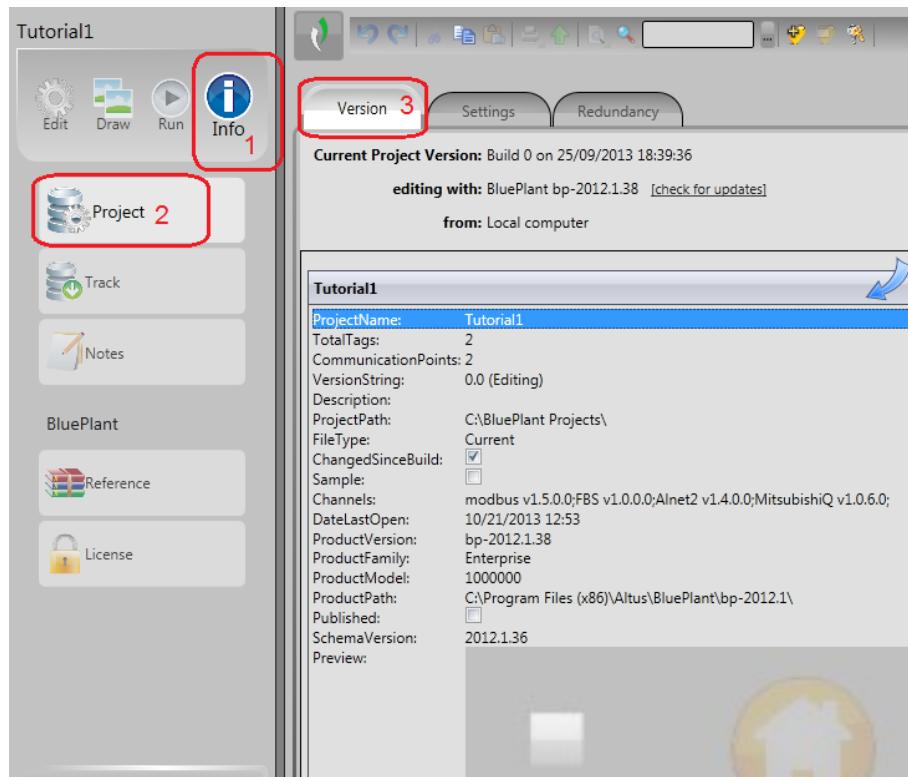
## Menu Info

### Project

Submenu de informações do projeto atual.

### Version

A guia *Version*, ilustrada na Figura 6-114, contém informações sobre o projeto atual. Alguns campos de informação também estão disponíveis durante o Runtime usando Namespace *Info*.



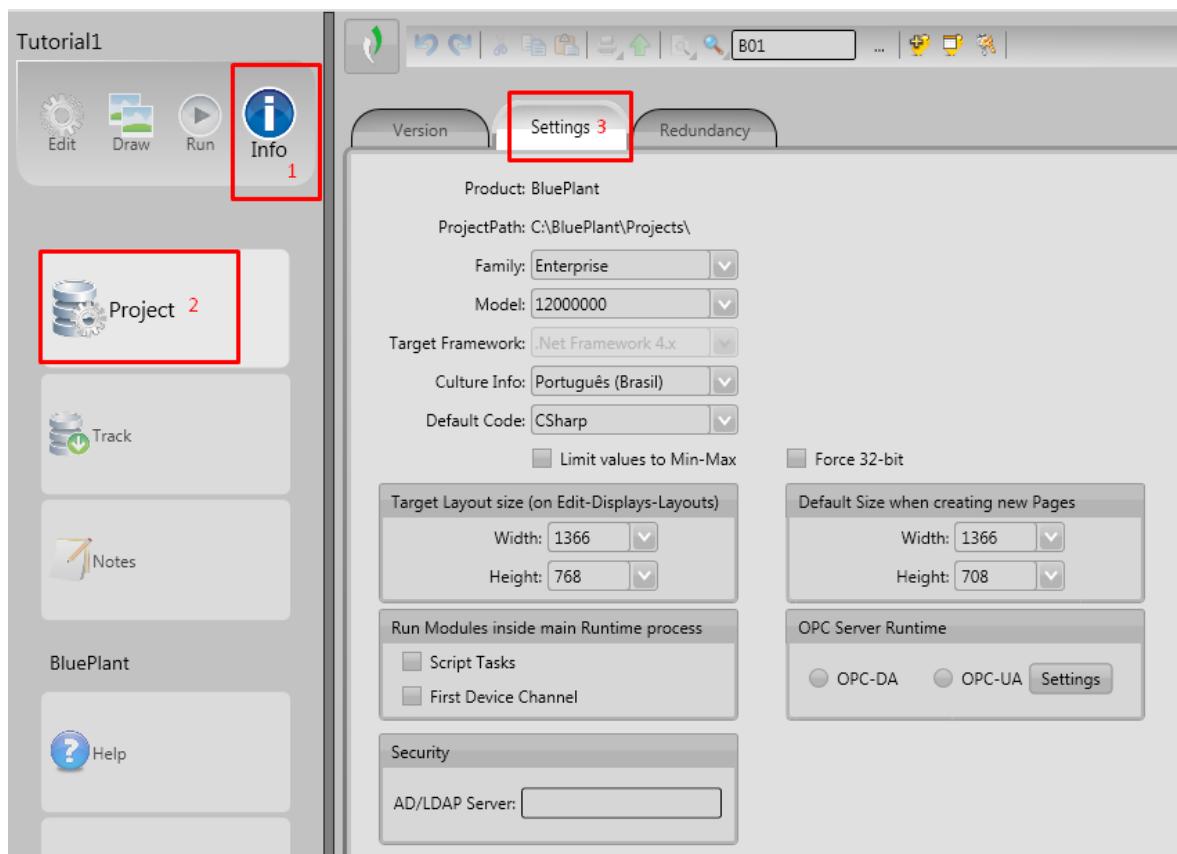
**Figura 6-114. Guia Version**

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 6-115. Nesta guia são definidas as configurações globais para o projeto. Alguns campos de informação também estão disponíveis durante o Runtime usando *Namespace Info*.



**Figura 6-115. 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. As opções de licença disponíveis podem ser verificadas em **Características do BluePlant**.

**Model:** Seleciona o modelo de Runtime. As opções de Runtime disponíveis podem ser verificadas em **Modelos do BluePlant**.

**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 o qual o BluePlant executa.

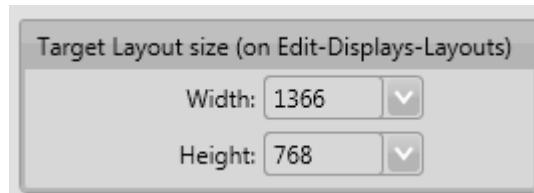
**Default Code:** Seleciona a linguagem do Script. As seguintes Linguagens podem ser selecionadas:

- VB.Net
- 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.

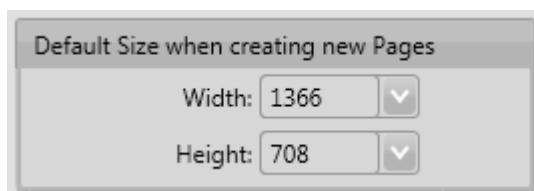
**Force 32-bit:** Se esta caixa estiver marcada o sistema irá executar em ambiente 32 bits.

**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 6-116. Ajuste do Tamanho Padrão para novos Layouts**

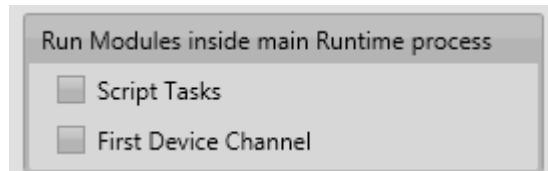
**Default size when creating new pages:** Define a resolução padrão para novas telas a serem criados no projeto. Isso não afeta as páginas existentes.



**Figura 6-117. Ajuste do Tamanho Padrão para novas Páginas**

**Run modules inside realtime process:** Se esta caixa for marcada, executa módulos dentro do módulo TServer:

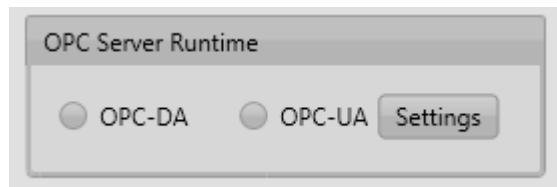
- Script Tasks: A tarefa de Script.
- First Device Channel: O executável do primeiro canal de device.



**Figura 6-118. Ajuste de módulos dentro do TServer**

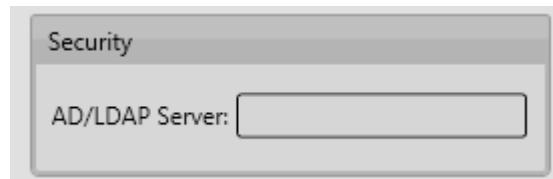
**OPC Server Runtime:** Define quel tipo de OPC irá executar:

- OPC-DA
- OPC-UA



**Figura 6-119. Ajuste do protocolo OPC Server**

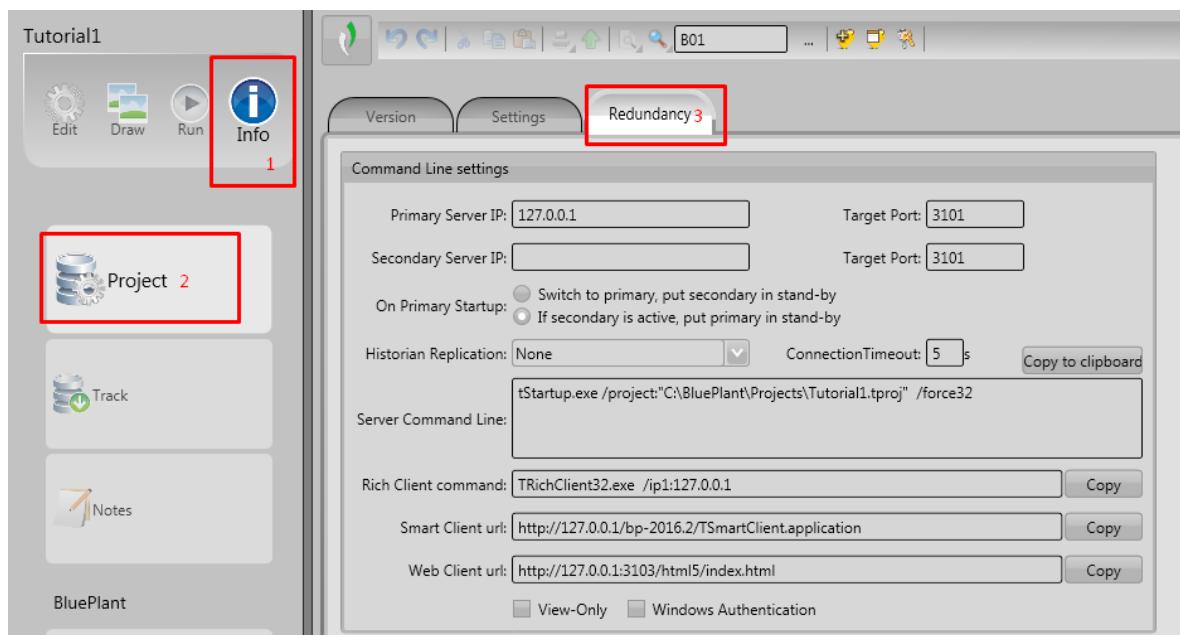
**Security:** Define configurações de segurança usando servidor AD/LDAP.



**Figura 6-120. Ajuste do Security**

#### Redundancy

Na sequência são explicados os itens da guia *Redundancy* conforme ilustrado na Figura 6-121.



**Figura 6-121. 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.

- Switch to primary, put secondary in standy-by
- If secondary is active, put primary in stand-by

**Historian Replication:** Define quais históricos irão fazer replicação de dados automáticos.

- None
- Alarm Historian
- Tag Historian
- Alarm and Tag Historian

**Connection Timeout:** Tempo especificado que o servidor reserva para aguardar quando não há comunicação entre os servidores redundantes e assume como ativo.

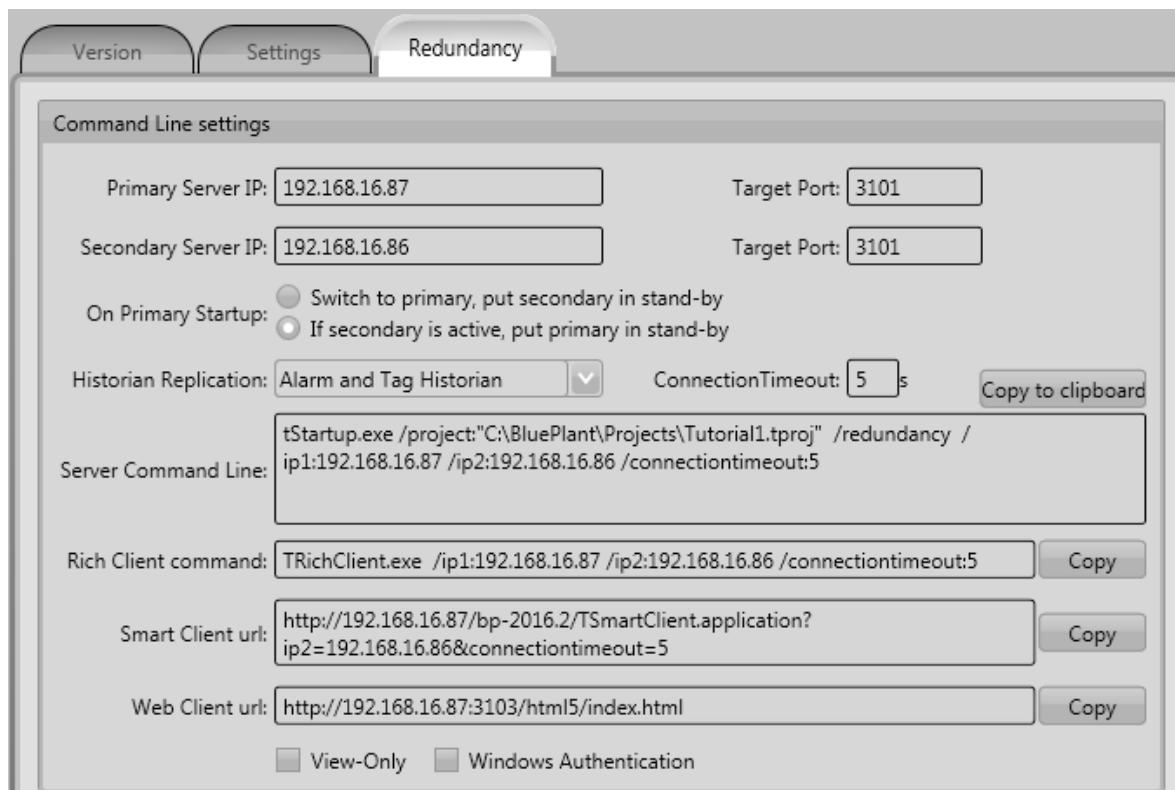
**Server Command Line:** Linha de comando utilizada para iniciar os servidores.

**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 Client url:** Linha de comando utilizada para iniciar os clientes HTML5.

**Configuração da redundância de servidores:** Segue a Figura 6-122 com um exemplo de configuração de redundância de servidores. Para realizar esta operação é necessário consultar a **NAP157 - Redundância de Servidores SCADA com BluePlant**.



**Figura 6-122. 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 6-123 mostra essa seleção.

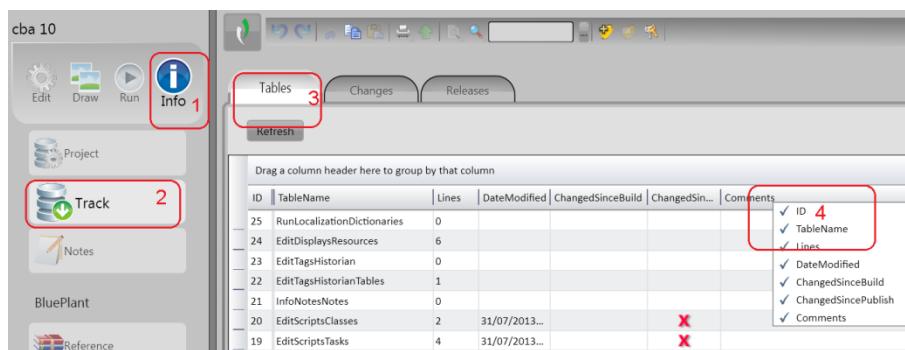


Figura 6-123. Guia Tables do Menu Info, Opção Track

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 6-123.

**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 6-124 mostra essa seleção. O campo *Tracking* define quando as mudanças serão salvas.

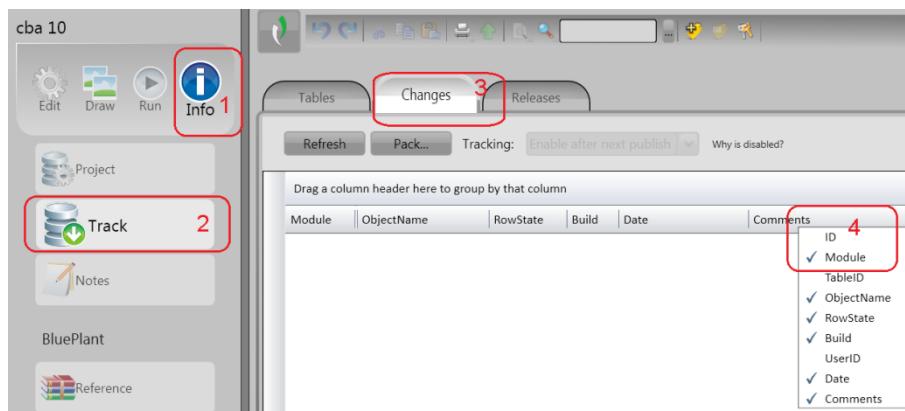


Figura 6-124. Guia Changes do Menu Info, Opção Track

Os itens que compõem o menu *Info> Track> Changes* estão numerados na figura anterior e são descritos abaixo:

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).

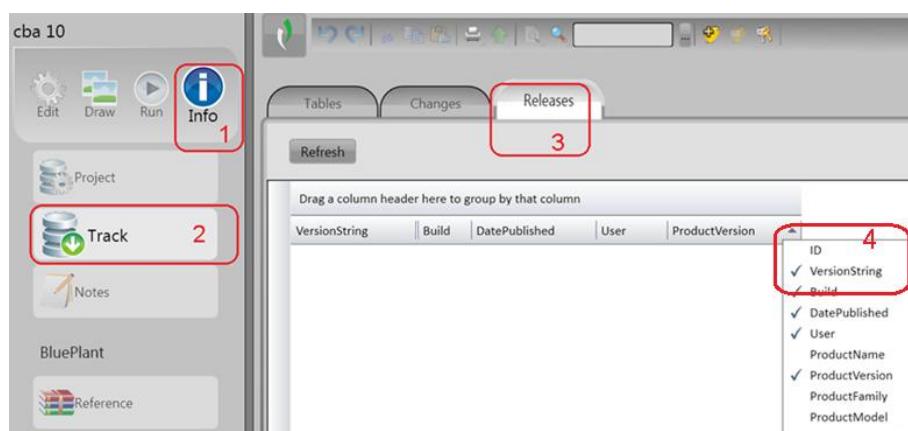
**Date:** Data de modificação da execução. Atributos: *ReadOnly* (somente leitura).

**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 6-125 mostra essa seleção.



**Figura 6-125. 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 abaixo e podem ser visualizados conforme Figura 6-125.

**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 para indicar licença sem prazo de expiração. 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 *Excluir 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 6-126 mostra essa seleção.

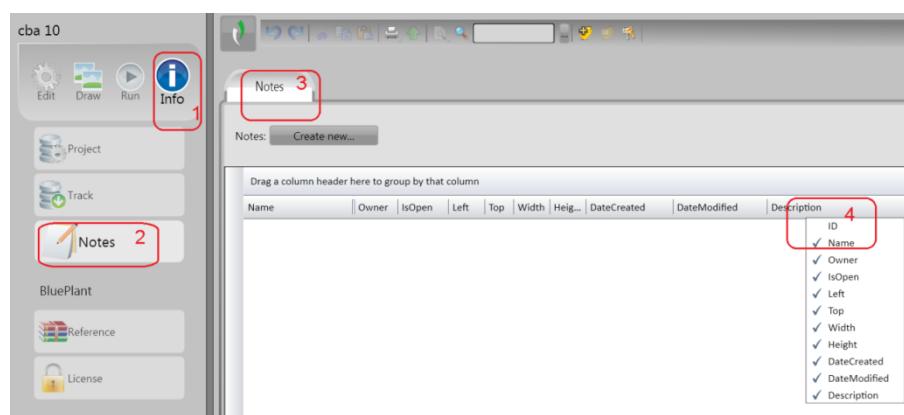


Figura 6-126. Guia Notes do Menu Info Opção Notes

Os itens que compõem o menu *Info>Notes>Notes* estão numerados na Figura 6-126 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,  esta aberto para visualização e  esta 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.

- TStartup
- PropertyWatch
- TraceWindow
- ModuleInformation

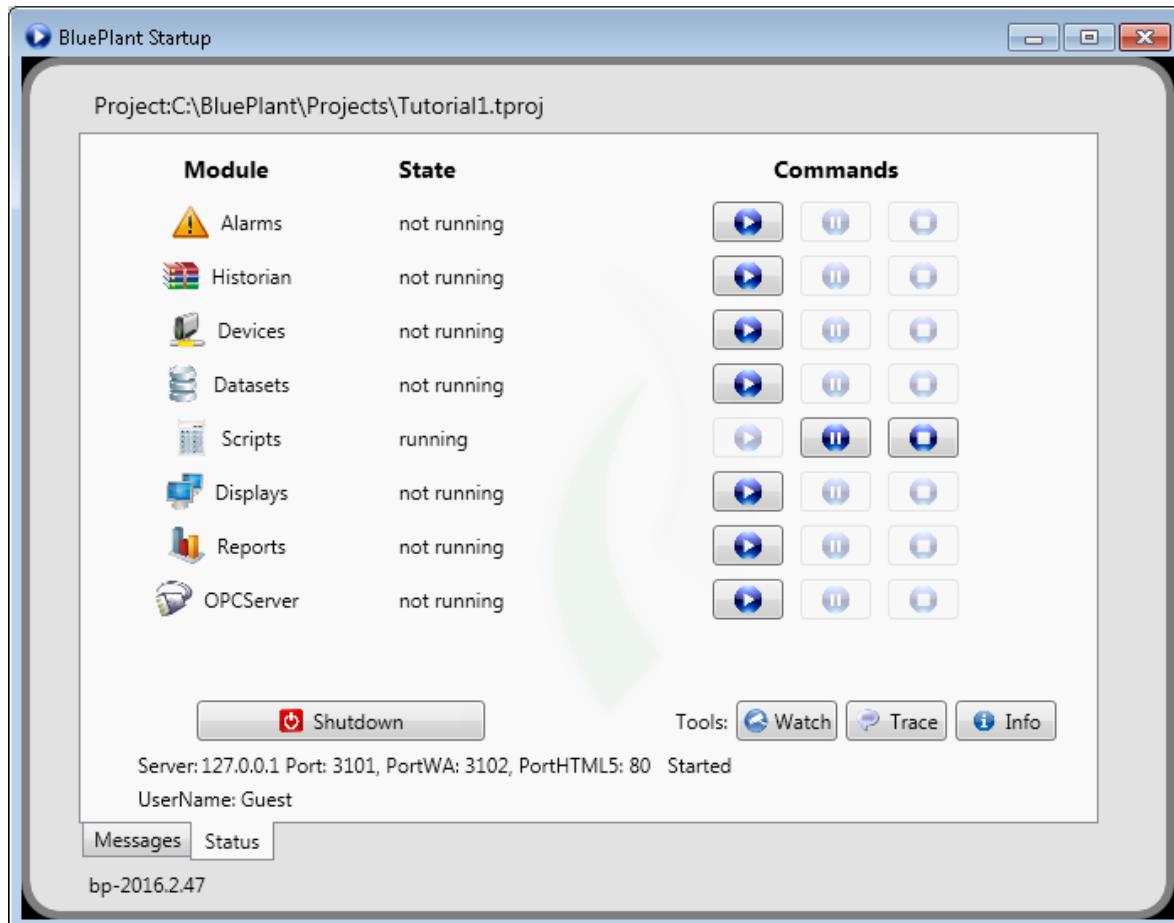
### TStartup

Ao executar o projeto usando o Gerenciador ou interfaces de boas-vindas, a janela TStartup aparece conforme ilustrado na Figura 6-127.

Conteúdo: informações sobre Status dos Módulos (rodando, parado ou pausado). 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*

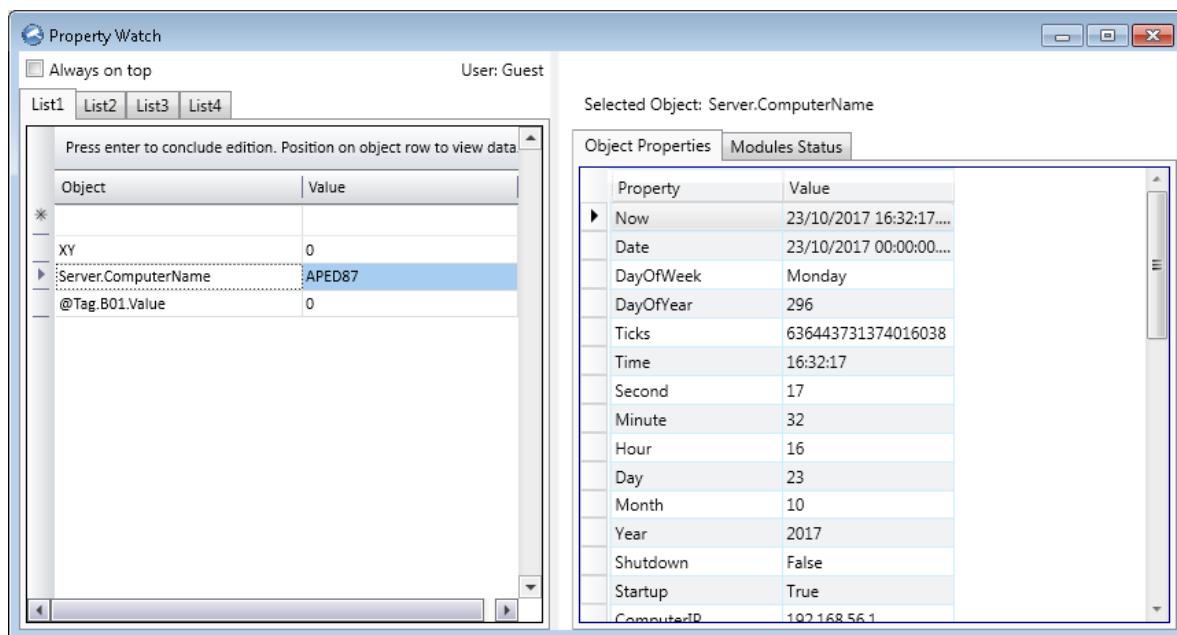


**Figura 6-127. Janela de Startup**

Pode-se chamar a janela de Startup usando a linha de comando. Os comandos disponíveis podem ser consultados na seção Linhas de Comando TStartup.

#### PropertyWatch

O *PropertyWatch* pode ser usado para acessar o servidor de Domínio de tags e Propriedades e mudar seus valores conforme ilustrado na Figura 6-128.



**Figura 6-128. PropertyWatch**

Pode-se chamar o *PropertyWach* usando a linha de comando. Os comandos disponíveis podem ser consultados na seção Linhas de Comando PropertyWatch.

#### TraceWindow

A *TraceWindow* é usada para capturar as mensagens dos módulos de Runtime do BluePlant conforme ilustrado na Figura 6-129. Estas mensagens são úteis para compreender o comportamento em Runtime.

Drag a column header here to group by that column.					
Date Time	Type	Module Name	Info 1	Info 2	Message
23/10/2017 16:33:20.833	Info	RichClient	UserName	Guest	User logged
23/10/2017 16:34:01.138	Error	Device	MODBUS	READ	ID: 82 :: Send - Error creating socket: -103
23/10/2017 16:34:01.141	Info	Device	MODBUS	status: -103	ID: 82:: - Command Error:-103 (Master) 400000
23/10/2017 16:34:01.627	Error	Device	MODBUS	READ	ID: 83 :: Send - Error creating socket: -103
23/10/2017 16:34:01.627	Info	Device	MODBUS	status: -103	ID: 83:: - Command Error:-103 (Master) 400000
23/10/2017 16:34:02.123	Error	Device	MODBUS	READ	ID: 84 :: Send - Error creating socket: -103
23/10/2017 16:34:02.123	Info	Device	MODBUS	status: -103	ID: 84:: - Command Error:-103 (Master) 400000
23/10/2017 16:34:02.658	Error	Device	MODBUS	READ	ID: 85 :: Send - Error creating socket: -103
23/10/2017 16:34:02.658	Info	Device	MODBUS	status: -103	ID: 85:: - Command Error:-103 (Master) 400000
23/10/2017 16:34:03.142	Error	Device	MODBUS	READ	ID: 86 :: Send - Error creating socket: -103
23/10/2017 16:34:03.142	Info	Device	MODBUS	status: -103	ID: 86:: - Command Error:-103 (Master) 400000
23/10/2017 16:34:03.672	Error	Device	MODBUS	READ	ID: 87 :: Send - Error creating socket: -103
23/10/2017 16:34:03.673	Info	Device	MODBUS	status: -103	ID: 87:: - Command Error:-103 (Master) 400000
23/10/2017 16:34:04.172	Error	Device	MODBUS	READ	ID: 88 :: Send - Error creating socket: -103
23/10/2017 16:34:04.172	Info	Device	MODBUS	status: -103	ID: 88:: - Command Error:-103 (Master) 400000
23/10/2017 16:34:04.681	Error	Device	MODBUS	READ	ID: 89 :: Send - Error creating socket: -103
23/10/2017 16:34:04.682	Info	Device	MODBUS	status: -103	ID: 89:: - Command Error:-103 (Master) 400000
23/10/2017 16:34:05.182	Error	Device	MODBUS	READ	ID: 90 :: Send - Error creating socket: -103
23/10/2017 16:34:05.182	Info	Device	MODBUS	status: -103	ID: 90:: - Command Error:-103 (Master) 400000
23/10/2017 16:34:05.681	Error	Device	MODBUS	READ	ID: 91 :: Send - Error creating socket: -103
23/10/2017 16:34:05.682	Info	Device	MODBUS	status: -103	ID: 91:: - Command Error:-103 (Master) 400000
23/10/2017 16:34:06.179	Error	Device	MODBUS	READ	ID: 92 :: Send - Error creating socket: -103
23/10/2017 16:34:06.179	Info	Device	MODBUS	status: -103	ID: 92:: - Command Error:-103 (Master) 400000

Figura 6-129. TraceWindow

### Ajustes:

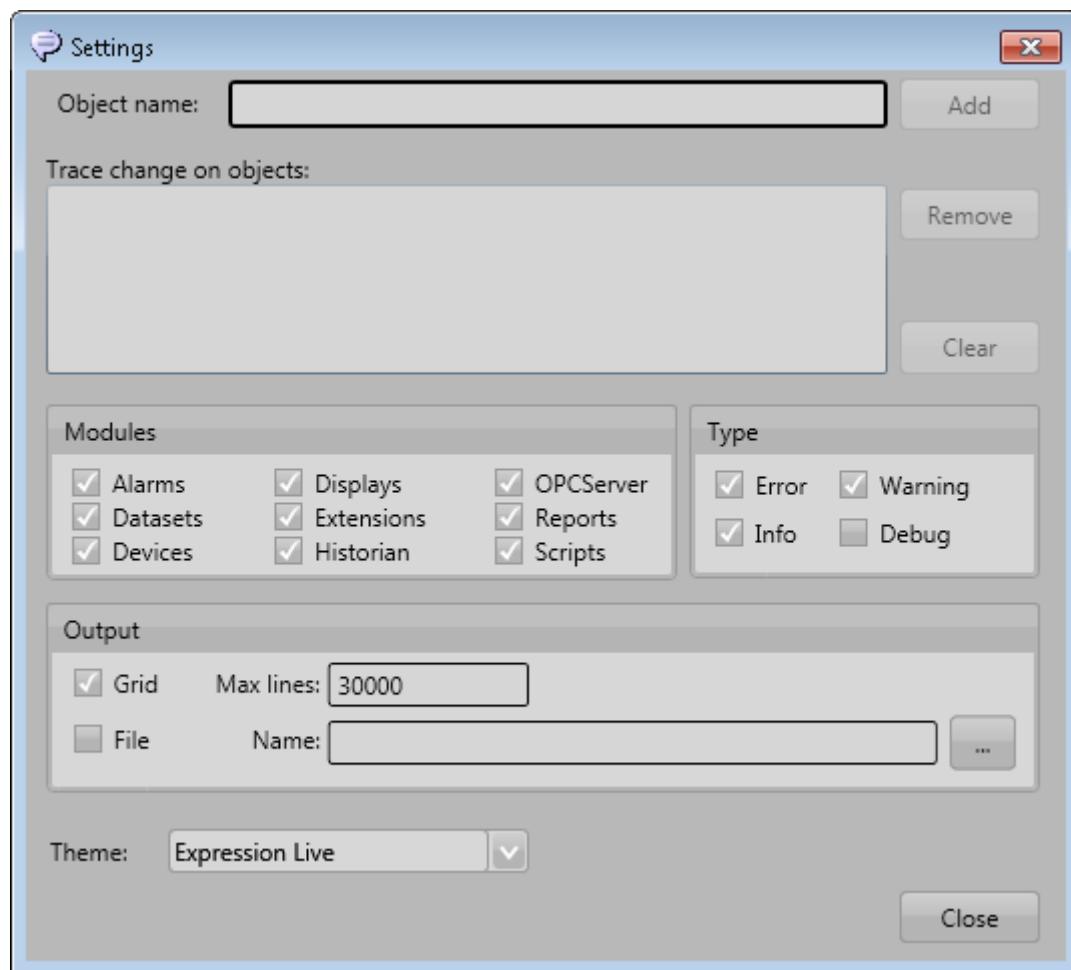
As opções de ajuste mostradas na Figura 6-130 são:

- 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.

### Saídas:

- Grid: indica o número máximo de arquivos que aparecem na grade
- File: um nome de arquivo pode ser definido para armazenamento das mensagens



**Figura 6-130. Configurações**

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)`.

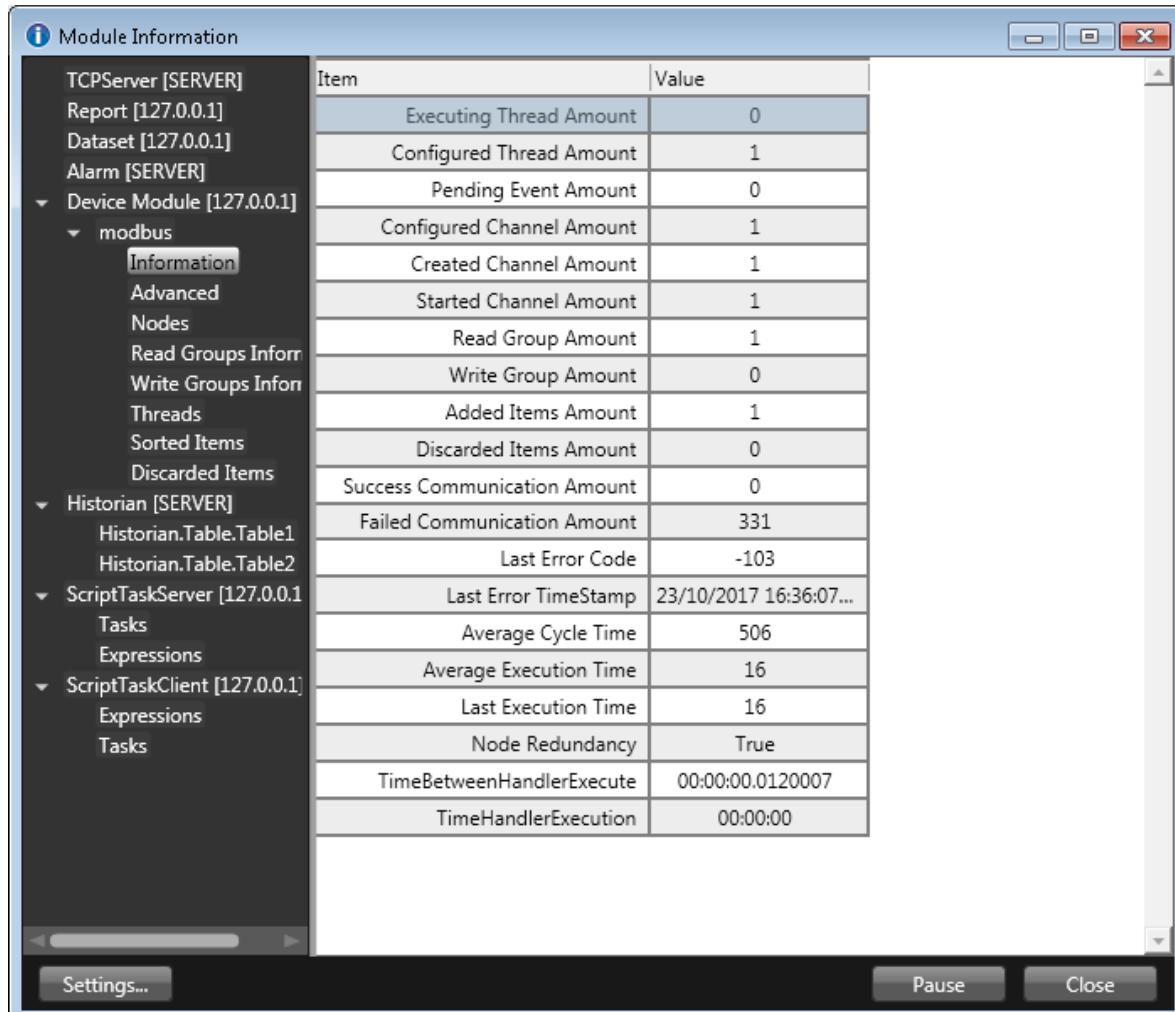
#### *TraceWindow Message*

Seleciona os tipos de mensagem da janela de rastreamento. Os seguintes tipos de mensagem da janela de rastreamento estão disponíveis:

- Erro
- Informação
- Advertência
- Depuração

## ModuleInformation

*ModuleInformation* contém informações avançadas sobre a execução dos módulos conforme ilustrado na Figura 6-131.



**Figura 6-131. Informações do Módulo**

Pode-se chamar o *ModuleInformation* usando a linha de comando e parâmetros específicos.

## Objetos Runtime

Esta funcionalidade permite visualizar todos os elementos abertos, incluindo Telas, Scripts e Relatórios. Os objetos do Runtime estão em grupos que contêm informações específicas de suas funcionalidades, sendo chamados de Namespaces. 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

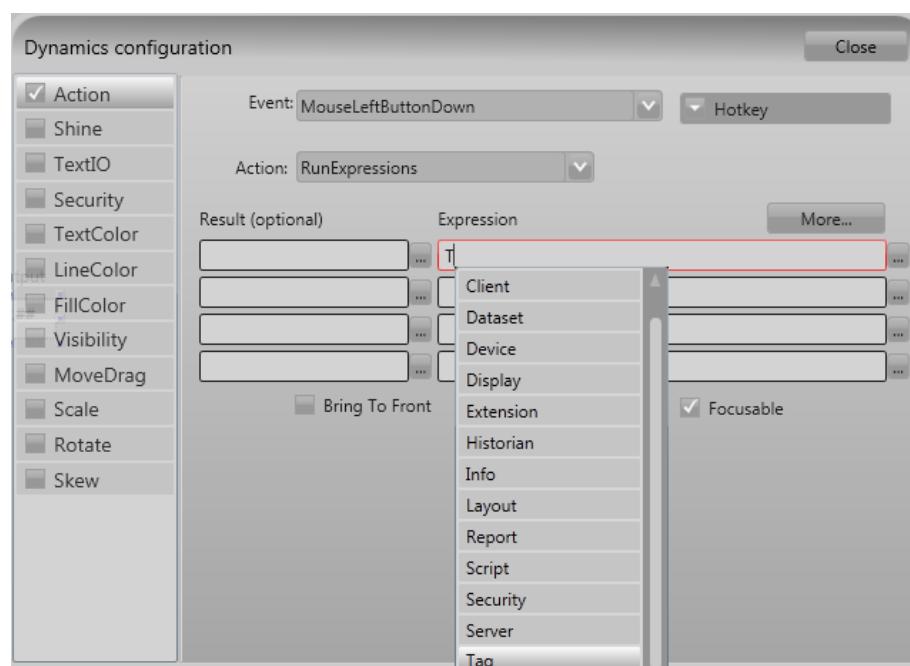
Durante a configuração do projeto, digite diretamente no campo. O *Intellisense* guiará o usuário em relação aos *namespaces* válidos para aquele campo assim como os membros disponíveis.

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 (Figura 6-132), use:

```
Tag.Analog1
Alarm.Group.Warning.TotalActive
Device.Node.Node1.Status
```



**Figura 6-132. Namespaces em Diálogos**

No código script (Figura 6-133), use:

```
@Tag.Analog1
@Alarm.Group.Warning.TotalActive
@Device.Node.Node1.Status
```

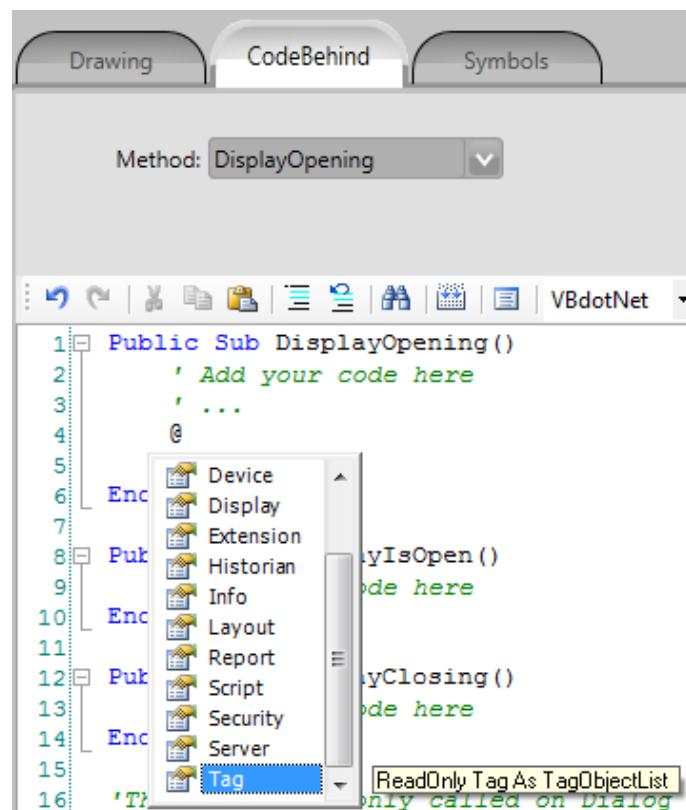


Figura 6-133. Namespaces em Scripts

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. Os tags são criados na interface *EditTagsObjects*. Use a sintaxe *Tag.Namespace* para referir-se ao tag criado e *Tag.namespace.Min* para acessar a Propriedade *Mínimo*. Para cada tag é definido um tipo para o seu valor. Os tipos incorporador estão disponíveis na Tabela 6-2.

A sintaxe *tag.<TagName>* está disponível para todos os tipos de tag. Para saber o tipo de um 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 todos tags Analógicos (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".

### Alarm Disable

Fornece ou define o estado de alarme desabilitado do 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 ao 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.

### Alarm State

Fornece ou define o estado de alarme do 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 do 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 do Tag. True: valor alterado. False: valor inalterado.

### Domain

Fornece ou define a propriedade de Domínio do Tag. Se 0 (zero): servidor. Se 1: cliente.

### Units

Fornece ou define as unidades de engenharia usadas para quantificar o Tag.

### Format

Fornece ou define o formato do valor do Tag para fins de exibição.

## Historian

Fornece o objeto HistoryItem onde o Tag atual está configurado como TagName. Se tags duplicados forem permitidos nas tabelas de Historiadores, então o Historiador apontará para o último objeto HistoryItem onde o Tag atual está configurado como TagName. Exemplo:

```
@Tag.AnalogInt1.Historian.Deadband = 3          (VB)
@Tag.AnalogInt1.Historian.Deadband = 3;           (C#)
```

## Locked

Fornece ou define o estado bloqueado do Tag. Quando um Tag está bloqueado, 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 do Tag. Valores permitidos:

Valor	Qualidade
0	Ruim
192	Bom

**Tabela 6-8. Valores Padrões**

Quando a qualidade é definida por OPC ela está sujeita à especificação do server conectado. As qualidades padrões do protocolo OPC seguem:

Valor	Qualidade	
0	Qualidade Ruim	Não Especificada
1		Baixo
2	Valor de Limite de Campo	Alto
3		Constante
4		Erro de Configuração
8		Não Conectado
12		Falha no Dispositivo
16	Qualidade Ruim	Falha no Sensor
20		Último Valor Conhecido
24		Falha de Comunicação
28		Fora de Serviço
64		Não Especificado
68		Último Valor Utilizável
80	Qualidade Incerta	Sensor sem Precisão
84		Unidades Excedidas
88		Subnormal
192	Qualidade Boa	Bom
216		Sobreposição Local
255	Sem Qualidade	Sem Valor

**Tabela 6-9. Valores de Qualidade OPC**

Exemplo:

```
@Tag.AnalogInt1.Quality = 192          (VB)
@Tag.AnalogInt1.Quality = 192;         (C#)
```

### Retentive

Fornece ou define a propriedade *Retentive* do Tag, a qual especifica se a propriedade de Valor do 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* do tag. Exemplo:

```
Dim dt As DateTimeOffset          (VB)
dt = @Tag.AnalogInt1.Timestamp
DateTimeOffset dt;                (C#)
dt = @Tag.AnalogInt1.Timestamp;
```

### ValueType

Fornece a propriedade *ValueType* do 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 do 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

Altera a propriedade de valor do 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* do Tag. Quando um Tag está bloqueado, 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 do 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 do 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 Analog Class.

Membro	Descrição
Bit0	Fornece ou define o bit 0 do valor do tag.
Bit1	Fornece ou define o bit 1 do valor do tag.
Bit2	Fornece ou define o bit 2 do valor do tag.
Bit3	Fornece ou define o bit 3 do valor do tag.
Bit4	Fornece ou define o bit 4 do valor do tag.
Bit5	Fornece ou define o bit 5 do valor do tag.
Bit6	Fornece ou define o bit 6 do valor do tag.
Bit7	Fornece ou define o bit 7 do valor do tag.
Bit8	Fornece ou define o bit 8 do valor do tag.
Bit9	Fornece ou define o bit 9 do valor do tag.
Bit10	Fornece ou define o bit 10 do valor do tag.
Bit11	Fornece ou define o bit 11 do valor do tag.
Bit12	Fornece ou define o bit 12 do valor do tag.
Bit13	Fornece ou define o bit 13 do valor do tag.
Bit14	Fornece ou define o bit 14 do valor do tag.
Bit15	Fornece ou define o bit 15 do valor do tag.
Bit16	Fornece ou define o bit 16 do valor do tag.
Bit17	Fornece ou define o bit 17 do valor do tag.
Bit18	Fornece ou define o bit 18 do valor do tag.
Bit19	Fornece ou define o bit 19 do valor do tag.
Bit20	Fornece ou define o bit 20 do valor do tag.
Bit21	Fornece ou define o bit 21 do valor do tag.
Bit22	Fornece ou define o bit 22 do valor do tag.
Bit23	Fornece ou define o bit 23 do valor do tag.
Bit24	Fornece ou define o bit 24 do valor do tag.
Bit25	Fornece ou define o bit 25 do valor do tag.
Bit26	Fornece ou define o bit 26 do valor do tag.
Bit27	Fornece ou define o bit 27 do valor do tag.
Bit28	Fornece ou define o bit 28 do valor do tag.
Bit29	Fornece ou define o bit 29 do valor do tag.
Bit30	Fornece ou define o bit 30 do valor do tag.
Bit31	Fornece ou define o bit 31 do valor do tag.
Hi	Fornece ou define o limite high do tag.
HiHi	Fornece ou define o limite high-high do tag.
Lo	Fornece ou define o limite low do tag.
LoLo	Fornece ou define o limite low-low do tag

**Tabela 6-10. Propriedades do Runtime para Analog Class**

*Class Analog<T>*

Propriedades de Runtime para Classe Analog <T>.

**Deadband**

Fornece ou define a banda morta do Tag analógico. Exemplo:

```
@Tag.AnalogDouble1.Deadband = 5          (VB)
@Tag.AnalogDouble1.Deadband = 5;          (C#)
```

### LockValue

Fornece ou define o valor de bloqueio do 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 do Tag analógico. 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 do tag analógico. Exemplo:

```
@Tag.AnalogDouble1.StartValue = 50          (VB)
@Tag.AnalogDouble1.StartValue = 50;          (C#)
```

### State

Fornece ou define o estado do Tag analógico. 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                (C#) ou
@Tag.AnalogInt1.Value = 55;         (C#) ou
@Tag.AnalogInt1 = 55;
```

### *Class AnalogInt*

Propriedades do Runtime para AnalogInt Class.

### *Class AnalogDecimal*

Propriedades do Runtime para Classe Decimal Analógica.

### *Class AnalogDoble*

Propriedades do Runtime para Classe AnalogDouble.

### *Class Text*

Propriedades do runtime para classe texto.

### LockValue

Fornece ou define o valor de bloqueio do tag de texto. Exemplo:

```
@Tag.Text1.LockValue = "Welcome"          (VB)
@Tag.Text1.LockValue = "Welcome";          (C#)
```

### Value

Fornece ou define o valor do 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 do Tag de TDateTime. Exemplo:

```
@Tag.Text1.LockValue = "Welcome"      (VB)
@Tag.Text1.LockValue = "Welcome";     (C#)
```

#### Value

Fornece ou define o valor do 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 do 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 do 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 do 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 do 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 do 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 do 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.

## Namespace Security

### Class ModuleSecurity

Propriedades do Runtime para objetos ModuleSecurity.

### GetPasswordHint

Retorna (exibe) a dica de senha para o nome do usuário selecionado. Parâmetro: string *userName*.  
Exemplo:

```
Dim pswHint As string = @Security.GetPasswordHint("User") (VB)
string pswHint = @Security.GetPasswordHint("User"); (C#)
```

#### AddRuntimeUser

Acrescenta um usuário à lista de usuários de Runtime. Parâmetros associados: string name; string permissionsStr; string password; string passwordHint; string policyStr; string profileEmail; string profilePhone e string profileCompleteName. Exemplo:

```
@Security.AddRuntimeUser("User", "User", "psw", "remember", "Default", "a@b.com", "67521855",
"Alfred Burns") (VB)
@Security.AddRuntimeUser("User", "User", "psw", "remember", "Default", "a@b.com", "67521855",
"Alfred Burns"); (C#)
```

#### GetListOfUserNames

Retorna a lista de nomes dos usuários de Runtime separados por \n (nova linha). Exemplo:

```
Dim userList as string          (VB)
userList = @Security.GetListOfUserNames()
string userList;               (C#)
userList = @Security.GetListOfUserNames();
```

### RemoveRuntimeUser

Remove o usuário de Runtime da lista de usuários. Parâmetro: string name. Exemplo:

```
@Security.RemoveRuntimeUser("User")      (VB)
@Security.RemoveRuntimeUser("User");      (C#)
```

### Permission

Fornece a lista de permissão. Refere-se aos objetos SecurityPermission. Exemplo:

```
Dim permissionList as SecurityPermissionList  (VB)
permissionList = @Security.Permission
SecurityPermissionList permissionList;          (C#)
permissionList = @Security.Permission;
```

### User

Fornece a lista de usuários. Fornece acesso aos objetos SecurityUser. Exemplo:

```
Dim userList as SecurityUserList  (VB)
userList = @Security.User
SecurityUserList userList;        (C#)
userList = @Security.User;
```

### Policy

Fornece a lista de Política. Exemplo:

```
Dim policyList as SecurityPolicyList  (VB)
policyList = @Security.Policy
SecurityPolicyList policyList;        (C#)
policyList = @Security.Policy;
```

### *Class SecurityPermission*

Propriedades do Runtime para objetos SecurityPermission.

#### *DateCreated*

Fornece a data e hora em que a SecurityPermission foi criada. Exemplo:

```
Dim permissionDate as DateTime          (VB)
permissionDate = @Security.PermissionAdministrator.DateCreated
DateTime permissionDate;               (C#)
permissionDate = @Security.PermissionAdministrator.DateCreated;
```

#### *DateModified*

Fornece a data e hora em que a SecurityPermission foi modificada. Exemplo:

```
Dim permissionDate as DateTime          (VB)
permissionDate = @Security.PermissionAdministrator.DateModified
DateTime permissionDate;               (C#)
permissionDate = @Security.PermissionAdministrator.DateModified;
```

#### *Description*

Fornece a descrição da SecurityPermission. Exemplo:

```
Dim permissionDescription as string      (VB)
permissionDescription = @Security.PermissionAdministrator.Description
int permissionDescription;              (C#)
permissionDescription = @Security.PermissionAdministrator.Description;
```

**Edit**

Fornece a permissão de Edição. Exemplo:

```
Dim editPermission as Integer      (VB)
editPermission = @Security.Permission.Administrator.Edit
int editPermission;               (C#)
editPermission = @Security.Permission.Administrator.Edit;
```

**Name**

Fornece o nome SecurityPermission. Exemplo:

```
Dim permissionName as string      (VB)
permissionName = @Security.Permission.Administrator.Name
string permissionName;           (C#)
permissionName = @Security.Permission.Administrator.Name;
```

**Run**

Fornece a permissão de execução. Exemplo:

```
Dim runPermission as Integer      (VB)
runPermission = @Security.Permission.Administrator.Run
int runPermission;               (C#)
runPermission = @Security.Permission.Administrator.Run;
```

**Class SecurityUser**

Propriedades do Runtime para objetos SecurityUser.

**DateCreated**

Fornece a data e hora em que o SecurityUser foi criado. Exemplo:

```
Dim userDate as DateTime      (VB)
userDate = @Security.User.Guest.DateCreated
DateTime userDate;           (C#)
userDate = @Security.User.Guest.DateCreated;
```

**DateModified**

Fornece a data e hora em que o SecurityUser foi modificado. Exemplo:

```
Dim userDate as DateTime      (VB)
userDate = @Security.User.Guest.DateModified
DateTime userDate;           (C#)
userDate = @Security.User.Guest.DateModified;
```

**PolicyName**

Fornece o nome das políticas do SecurityUser. Exemplo:

```
Dim userPolicyName as string      (VB)
userPolicyName = @Security.User.Guest.PolicyName
string userPolicyName;           (C#)
userPolicyName = @Security.User.Guest.PolicyName;
```

**SecurityUser.Blocked**

Fornece o estado bloqueado do SecurityUser. Exemplo:

```
Dim blockedState as Boolean      (VB)
blockedState = @Security.User.Guest.Blocked
bool blocked;                  (C#)
blocked = @Security.User.Guest.Blocked;
```

**SecurityUser.Deleted**

Fornece o estado excluído do SecurityUser. Exemplo:

```
Dim deletedState as Boolean      (VB)
deletedState = @Security.User.Guest.Deleted
bool deleted;                  (C#)
deleted = @Security.User.Guest.Deleted;
```

**SecurityUser.Name**

Fornece o nome do SecurityUser. Exemplo:

```
Dim userName as string      (VB)
userName = @Security.User.Guest.Name
string userName;            (C#)
userName = @Security.User.Guest.Name;
```

**SecurityUser.PasswordHint**

Fornece a dica de senha dos SecurityUsers. Exemplo:

```
Dim pswHint as string      (VB)
pswHint = @Security.User.Guest.PasswordHint
string pswHint;            (C#)
pswHint = @Security.User.Guest.PasswordHint;
```

**SecurityUser.Permissions**

Fornece as permissões do SecurityUser. Exemplo:

```
Dim userPermissions as Long      (VB)
userPermissions = @Security.User.Guest.Permissions
long userPermissions;          (C#)
userPermissions = @Security.User.Guest.Permissions;
```

**SecurityUser.PermissionsName**

Fornece o nome das permissões do SecurityUser. Exemplo:

```
Dim permissionsName as string      (VB)
permissionsName = @Security.User.Guest.Permissions
string permissionsName;          (C#)
permissionsName = @Security.User.Guest.Permissions;
```

**SecurityUser.Policy**

Fornece as políticas do SecurityUser. Exemplo:

```
Dim userPolicy as Long      (VB)
userPolicy = @Security.User.Guest.Policy
long userPolicy;            (C#)
userPolicy = @Security.User.Guest.Policy;
```

**SecurityUser.Profile**

Fornece o perfil do SecurityUser. Exemplo:

```
Dim userProfile as string      (VB)
userProfile = @Security.User.Guest.Profile
string userProfile;          (C#)
userProfile = @Security.User.Guest.Profile;
```

**Namespace Alarm**

Objetos de Runtime e métodos relacionados ao módulo de alarme.

*Class AlarmGroup*

Propriedades do Runtime para objetos de Grupo de Alarmes.

**AckAll**

Reconhece todos os itens de alarme que pertencem a este grupo. Alterna a propriedade definida para AckAll. Valores permitidos: 0: reconhecimento se o valor era 1 e 1: reconhecimento se o valor era 0. Exemplo:

```
@Alarm.AckAll = (@Alarm.AckAll==0) ? 1 : 0;
```

**AckRequired**

Fornece o Reconhecimento do operador exigido definido. Se 0: sem exigência de reconhecimento. Se 1: reconhecimento exigido. Exemplo:

```
int AckReq = @Alarm.Group.Critical.AckRequired;
se (AckReq == 1) @Alarm.AckAll = (@Alarm.AckAll==0) ? 1 : 0;
```

**Colors**

Representa as cores de primeiro plano e plano de fundo para cada estado de alarme. Esta propriedade representa o valor configurado na coluna Cores.

**Description**

Descrição do Grupo de Alarmes configurada em EditAlarmsGroups. Exemplo:

```
@Tag.string = @Alarm.Group.Critical.Description;
```

**Disable**

Habilita/Desabilita um Grupo de Alarmes. Se 0: habilita um Grupo de Alarmes. Se 1: desabilita um Grupo de Alarmes. Exemplo:

```
@Alarm.Group.Critical.Disable = 1;
```

**Id**

Fornece a ID de um Grupo de Alarmes. Exemplo:

```
@Tag.Int = @Alarm.Group.Critical.Id;
```

**LogEvents**

Fornece o tipo de arquivamento de Historiador em eventos de Alarme. Os valores permitidos são:

- 0 - Nenhum
- 1 - Ativo
- 2 - ActiveAck
- 3 - ActiveNorm
- 4 - All

Exemplo:

```
@Tag.Int = @Alarm.Group.Critical.LogEvents;
```

**PriorityItem**

Retorna o item de Alarme de maior prioridade de um Grupo de Alarmes. Exemplo:

```
@ALARM.PRIORITYITEM PITEM = @ALARM.GROUP.CRITICAL.PRIORITYITEM;
```

### Show

Fornece um valor configurado para exibir/ocultar um Alarme. Valores permitidos:

- 0 - Nenhum
- 1 - Lista

Exemplo:

```
@Tag.Int = @Alarm.Group.Critical.Show;
```

### Sound

Propriedade para descrever se o Som do Grupo de Alarmes está habilitado ou não. Valores permitidos:

- 0 - Nenhum
- 1 - Beep

Exemplo:

```
@Tag.Int = @Alarm.Group.Critical.Sound;
```

#### TotalCount

Fornece o número de Alarmes Ativos. Exemplo:

```
@Tag.Int = @Alarm.Group.Critical.TotalCount;
```

#### UnAckCount

Fornece o número de Alarmes Sem Reconhecimento. Exemplo:

```
@Tag.Int = @Alarm.Group.Critical.UnAckCount;
```

#### *Class AlarmItem*

Propriedades do Runtime para objetos de Itens de Alarme.

#### AckTime

Horário em que o Alarme está definido em *Reconhecimento*.

#### ActiveTime

Horário em que o Item de Alarme iniciou.

#### Alarm

Propriedade para verificar se o Item de Alarme está ativo.

#### ColorBG

Configurada a Cor do Plano de Fundo do Item de Alarme.

#### ColorFG

Configurada a Cor do Primeiro Plano do Item de Alarme.

#### Condition

Configura a condição de avaliação para gerar um Item de Alarme. Os valores permitidos estão descritos a seguir.

- Hi (alto)
- HiHi (alto-alto)
- Lo (baixo)
- LoLo (baixo-baixo)
- RateOfChange (taxa de variação)
- DeviationMinor (desvio para mais)
- DeviationMajor (desvio para menos)
- Equal (igual)
- NotEqual (diferente de)
- GreaterThan (maior que)
- GreaterEqual (maior ou igual)
- LessThan (menor que)

- LessEqual (menor ou igual)
- Changed (alterado)
- ChangedUp (alterado para cima)
- ChangedDown (alterado para baixo)

#### Deadband

Banda morta de um item de alarme definido. Configura o tempo de inatividade do item de alarme. O valor é escolhido pelo usuário.

#### Disable

Propriedade para desabilitar/habilitar um item de alarme. Valores permitidos:

- 0 - Habilitar
- 1 - Desabilitar

#### Group

Um Grupo ao qual pertence um Item de Alarme.

#### Id

ID do objeto (uso interno).

#### Limit

Fornece o valor definido para avaliar as condições do item de Alarme.

#### Message

Fornece a mensagem configurada para ser exibida quando ocorre o Alarme.

#### NormTime

Tempo que o Alarme leva para voltar ao "Normal".

#### Priority

Fornece uma Prioridade do Item de Alarme. O valor é escolhido pelo usuário.

#### Setpoint

Fornece o valor definido para avaliar as condições do Item de Alarme.

#### SetPointDeadband

Representa a banda morta para a propriedade SetPoint. Usada para as seguintes condições de alarme: DeviationMinor e DeviationMajor.

#### State

Fornece o estado de um item de alarme. Um item de alarme pode ser: Ativo, Reconhecimento, Normal e TagName. Neste último caso fornece o TagName definido que será avaliado para gerar um Item de Alarme.

#### UnAck

Propriedade Ler/Definir um item de alarme de reconhecimento. Valores permitidos:

- 0 - Não Reconhecer Item de Alarme
- 1 - Reconhecer Item de Alarme

***Class ModuleAlarm***

Métodos do Runtime e propriedades para o módulo de alarme.

**AckAll**

Dispara esta propriedade para reconhecer todos os Alarmes.

**BeepState**

Indica se o beep está configurado para ON (valor = 1) no computador cliente.

**Group**

Fornece acesso aos objetos do Grupo de Alarmes.

**InitializationMessage**

Lê/Define a Mensagem Inicial a ser exibida.

**Item**

Acesso aos objetos do Item de Alarme.

**LastErrorMessage**

Contém a mensagem de descrição do erro mais recente ocorrido no Módulo Alarme.

**LastStoredTimeStamp**

Contém o TimeStamp do erro mais recente ocorrido no Módulo Alarme.

**PriorityItem**

Referência ao Item de Alarme online de maior prioridade.

**QueryActive**

Objeto TDataTable com a lista atual de Alarmes ativos.

**TotalCount**

Contagem total de Alarmes ativos.

**UnAckCount**

Contagem total de Alarmes que exigem reconhecimento.

**Namespace Device**

Objetos Runtime e métodos relacionados ao Dispositivo.

***Class DeviceAccessType***

Propriedades do Runtime para objetos DeviceAccessType.

**AcceptUnsolicited**

Quando configurado como true, os pontos do dispositivo são autorizados a receber mensagens não solicitadas. O canal do dispositivo deve ter a propriedade AcceptUnsolicited configurada para TRUE para permitir mensagens não solicitadas.

**ReadOnStartup**

Quando configurado como true, o ponto do dispositivo será lido na inicialização do Módulo Dispositivo.

**ReadPooling**

Quando configurado como true, indica que o Pooling lido está habilitado para este ponto do dispositivo.

**ReadPoolingRate**

Indica a taxa de Pooling para o grupo de pontos.

**WriteEnable**

Quando configurado como true, o ponto do dispositivo será escrito para o dispositivo quando o valor deste se alterar.

*Class DeviceChannel*

Propriedades do Runtime para objetos DeviceChannel.

**Activity**

Indicação de atividade. O módulo dispositivo alterna este valor para cada operação completada neste Canal.

**LastErrorCode**

Último Código do Estado de Erro (mais recente) ocorrido neste canal. Valores permitidos:

Valor	Descrição
0	Success
-1	BuildCommandException
-2	ParseCommandUnsolicitedException
-3	ParseReplyException
-4	BuildReplyUnsolicitedException
-5	ChannelException
-6	NodeException
-100	Base Send Error
-101	Base SendAndWait Error
-102	TCP Create Error 1
-103	TCP Create Error 2
-104	TCP Create SocketError
-105	TCP Connect Callback Error
-106	TCP Receive Error
-107	UDP Create Error

-108	UDP Receive Error
-109	Serial Create Error
-110	Serial Receive Error
-111	TCP NotConnected
-112	Start message timeout
-113	Receiving bytes timeout
-114	End message timeout
-115	Connect timeout
-200	ProtocolError
-201	InvalidProtocol
-202	InvalidStation
-203	InvalidCommand
-204	InvalidMsgSequence
-205	InvalidCheckSum
-206	InvalidAddress
-207	InvalidModifiers

**Tabela 6-11. Códigos de Estado para Canal ou Nô****Nota:**

**Valor:** Valores positivos são códigos de erro de protocolos específicos.

**LastErrorCode**

TimeStamp do último erro (mais recente) ocorrido neste canal.

**Status**

Status atual para este canal. Valores permitidos podem ser vistos na Tabela 6-11.

**Class DeviceNode**

Propriedades do Runtime para objetos DeviceNode.

**Activity**

Indicação de atividade. O dispositivo alterna este valor para cada operação executada neste Nô.

**BackupStation**

Estação de backup atual para este Nô.

**IsBackup**

Indicação de estação de Backup ativa.

**IsPrimary**

Indicação de estação principal ativa.

**IsRedundancyEnabled**

Indicação de redundância de Nô.

**LastErrorCode**

Último estado de código de erro (mais recente) ocorrido neste Nó. Valores permitidos podem ser vistos na Tabela 6-11.

**LastErrorDateTime**

DateTime para o último (mais recente) erro neste Nó.

**PrimaryStation**

Estação principal atual para este Nó.

**Status**

Status atual para este Nó. Valores permitidos podem ser vistos na Tabela 6-11.

***Class ModuleDevice***

Métodos do Runtime e propriedades para o Módulo Dispositivo.

**AccessType**

Acesso aos objetos DeviceAccessType.

**Channel**

Acesso aos objetos DeviceChannel.

**Node**

Acesso aos objetos DeviceNode.

**Namespace Dataset**

Objetos de Runtime e métodos relacionados à base de dados.

***Class DatasetDB***

Propriedades do Runtime para objetos DatasetDB.

**ConnectionString**

String usada para conectar-se com o banco de dados.

**Database**

Nome do banco de dados do objeto DB.

**Id**

Identificação (ID) do objeto (Uso Interno).

**LogonName**

Nome de logon usada para conectar-se com o banco de dados.

**Provider**

Provider do banco de dados selecionado.

**Class DatasetFile**

Propriedades do Runtime para objetos DatasetFile.

**LoadCommand**

Carrega os valores dos tags configurados na propriedade Objetos a partir do arquivo indicado pela propriedade Nome do arquivo. O parâmetro associado é string statusMessage, ou seja, a mensagem com o status do comando de carga.

**SaveCommand**

Salva os valores dos tags configurados na propriedade Objetos a partir do arquivo indicado pela propriedade Nome do Arquivo. O parâmetro associado é string statusMessage ou seja, a mensagem com o status do comando de salvamento.

**Completed**

O valor desta propriedade é aumentado quando uma operação é concluída.

**Description**

Fornece a descrição do DatasetFile configurado.

**Disable**

Desabilita os comandos para o DatasetFile quando o valor é maior que zero. Valores permitidos:

- 0 - Habilita os comandos para o DatasetFile
- 1 - Desabilita os comandos para o DatasetFile

**FileName**

Caminho completo do arquivo que será criado ou carregado. Esta propriedade representa o valor configurado na coluna FileName.

**FileType**

Indica o formato do arquivo. Valores permitidos:

- ASCII = 0
- Unicode = 1
- XML = 2

Esta propriedade representa o valor configurado na coluna FileType.

**Id**

Identificação (ID) do objeto (Uso Interno).

**LastStatus**

Fornece o status da última (mais recente) operação assíncrona. Valores permitidos:

- 0 - Êxito
- ≠0 - Código de erro

**LastStatusMessage**

Fornece a mensagem de status da última (mais recente) operação assíncrona.

**Load**

Envia um comando Carregar assíncrono quando o valor é alterado. Exemplo:

```
if (@Dataset.File.File1Unicode.Load.Equals(0))
@Dataset.File.File1Unicode.Load = 1;
else
@Dataset.File.File1Unicode.Load = 0;
```

**LoadExecuted**

O valor desta propriedade é alterado quando o comando Carregar assíncrono se completa.

**Objects**

Contém os tags e índices a serem salvos ou carregados. Os valores permitidos são TagName, o qual representa o nome do Tag (o índice inicial também pode ser especificado) e Index, indicando o índice (se o Tag é um array, e o índice inicial é especificado). Exemplo:

TagName	Índice
Tag.doubleArray[0]	10
Tag.textArray[2]	5
Tag.SelectStatusMsg	

**Tabela 6-12. Exemplo da Opção Objects**

**Save**

Envia um comando Salvar quando o valor é alterado. Exemplo:

```
if (@Dataset.File.File1Unicode.Save.Equals(0))
@Dataset.File.File1Unicode.Save= 1;
else
@Dataset.File.File1Unicode.Save= 0;
```

**SaveExecuted**

O valor desta propriedade é alterado quando o comando de salvamento assíncrono se completa.

*Class DatasetQuery*

Propriedades do Runtime para objetos DatasetQuery.

**ExecuteCommand**

Executa um comando síncrono de acordo com o SqlStatement. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- 0 - Êxito
- ≠0 - Código do erro

**ExecuteCommandWithStatus**

Executa um comando síncrono de acordo com o SqlStatement; exibe uma mensagem de status. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- 0 - Êxito
- ≠0 - Código do erro

Parâmetros: out string statusMessage. A mensagem indica o status do comando Next.

## NextCommand

Executa um comando síncrono Next que aumenta o valor da propriedade CursorIndex. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- 0 - Êxito
- ≠0 - Código do erro

Os tags configurados na coluna de mapeamento receberão o valor da próxima linha.

## NextCommandWithStatus

Executa um comando síncrono Next que aumenta o valor da propriedade CursorIndex e exibe uma mensagem de status. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- 0 - Êxito
- ≠0 - Código do erro

O parâmetro associado é out string statusMessage, ou seja, a mensagem que indica o status do comando Next. Os tags configurados na coluna mapeamento receberão o valor da próxima linha.

## SelectCommand

Executa um comando síncrono Selecionar, de acordo com o SqlCommand. Os valores de retorno são: DataTable em caso de êxito; caso contrário, nulo. Os tags configurados na coluna mapeamento receberão o valor da primeira linha. Exemplo:

```
DataTable dataTable = @Dataset.Query.query1.SelectCommand();
if (dataTable != null && dataTable.Rows.Count > 0)
{
@Tag.firstItem = dataTable.Rows[0]["Item"].ToString();
}
```

## SelectCommandWithStatus

Executa um comando síncrono Selecionar, de acordo com o SqlCommand, e exibe informação de status. Os valores de retorno são: DataTable em caso de êxito; caso contrário, nulo. O primeiro parâmetro associado é: out int status. O status da operação está indicado a seguir:

- 0 - Êxito
- ≠0 - Código do erro

O parâmetro associado é: out string statusMessage, ou seja, a mensagem que indica o status do comando Select. Os tags configurados na coluna mapeamento receberão o valor da primeira linha.

Exemplo:

```
int status;
string statusMessage;
DataTable dataTable;
dataTable = @Dataset.Query.query1.SelectCommandWithStatus(out status, out statusMessage);
if (status == 0 && dataTable != null && dataTable.Rows.Count > 0)
{
@Tag.firstItem = dataTable.Rows[0]["Item"].ToString();
}
```

## AsyncContents

Contém a TDataTable resultante de um dos comandos assíncronos, tais como Select ou Update.

## Completed

O valor desta propriedade aumenta quando uma operação assíncrona se conclui.

**CursorIndex**

Definir a posição da linha atual no DatasetTable resultante.

**DB**

Fornece o DB configurado em EditDatasetsTables.

**Description**

Fornece a descrição do DatasetQuery.

**Disable**

Desabilita os comandos para o DatasetQuery quando o valor é maior que zero. Valores permitidos:

- 0 - Habilita os comandos para o DatasetQuery
- 1 - Desabilita os comandos para o DatasetQuery

**Execute**

O valor da propriedade Execute se altera quando a operação é concluída. Exemplo:

```
if (@Dataset.Query.Query1.Execute.Equals(0))
@Dataset.Query.Query1.Execute = 1;
else
@Dataset.Query.Query1.Execute = 0;
```

**ExecuteCompleted**

O valor da propriedade ExecuteCompleted se altera quando a operação é concluída.

**Id**

Identificação (ID) do Objeto (Uso Interno).

**LastStatus**

Fornece o status da última (mais recente) operação assíncrona. Valores permitidos:

- 0 - Êxito
- ≠0 - Código do erro

**LastStatusMessage**

Fornece a mensagem de status para a última (mais recente) operação assíncrona, onde uma string vazia indica êxito.

**Mapping**

Fornece o mapeamento das colunas do DataTable resultantes com os tags. Valores de retorno:

TagName	Coluna
int_Id	ID
txt_Name	Nome
txt_Description	Descrição

**Tabela 6-13. Mapeamento das Colunas do DataTable**

Resultado:

```
Tag.int_Id=ID;
Tag.txt_Name=Name;
Tag.txt_Description=Description;
```

## Next

Envia um comando Next assíncrono quando o valor é alterado. Exemplo:

```
if (@Dataset.Query.Query1.Next.Equals(0))
@Dataset.Query.Query1.Next = 1;
else
@Dataset.Query.Query1.Next = 0;
```

## NextExecuted

O valor desta propriedade se altera quando o comando assíncrono Next se completa.

## RowCount

Fornece o número total de linhas na DatasetTable resultante.

## Select

Envia um comando Select assíncrono quando o valor é alterado. Exemplo:

```
if (@Dataset.Query.Query1.Select.Equals(0))
@Dataset.Query.Query1.Select= 1;
else
@Dataset.Query.Query1.Select= 0;
```

## SelectExecuted

O valor desta propriedade se altera quando o comando Select assíncrono é concluída.

## SqlStatement

Define o comando SQL a ser executado.

## *Class DatasetTable*

Propriedades do Runtime para objetos DatasetTable.

## DeleteCommand

Exclui a linha atual da DatasetTable na posição indicada pela propriedade *CursorPosition*. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- 0 - Êxito
- ≠0 - Código do erro

## DeleteCommandWithStatus

Exclui a linha atual da DatasetTable na posição indicada pela propriedade *CursorPosition* e fornece uma mensagem de status. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- 0 - Êxito
- ≠0 - Código do erro

Parâmetros: out string statusMessage. A mensagem indica o status do comando de exclusão onde uma string vazia significa êxito.

### InsertCommand

Insere os valores dos tags configurados na coluna mapeamento na DatasetTable na posição indicada pela propriedade CursorIndex. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- 0 - Êxito
- ≠0 - Código do erro

### InsertCommandWithStatus

Insere os valores dos tags configurados na coluna mapeamento na DatasetTable na posição indicada pela propriedade CursorIndex e fornece uma mensagem de status. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- 0 - Êxito
- ≠0 - Código do erro

O parâmetro associado é: out string statusMessage, ou seja, a mensagem que indica o status do comando de inserção, onde uma string vazia significa êxito.

### NextCommand

Executa um comando Next síncrono o qual aumenta o valor da propriedade CursorIndex. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- 0 - Êxito
- ≠0 - Código do erro

Os tags configurados na coluna mapeamento receberão o valor da próxima linha.

### NextCommandWithStatus

Executa um comando Next síncrono o qual aumenta o valor da propriedade CursorIndex e fornece uma mensagem de status. Os valores de retorno em relação ao status da operação estão indicados a seguir:

- 0 - Êxito
- ≠0 - Código do erro

O parâmetro associado é: out string statusMessage, ou seja, a mensagem que indica o status do comando Next, onde uma string vazia significa êxito. Os tags configurados na coluna mapeamento receberão o valor da próxima linha.

### SelectCommand

Executa um comando Select síncrono na DatasetTable. Valor de retorno: DataTable em caso de êxito; caso contrário é nulo. Os tags configurados na coluna mapeamento receberão o valor da primeira linha. Exemplo:

```
DataTable dataTable = @Dataset.Table.table1.SelectCommand();
if (dataTable != null && dataTable.Rows.Count > 0)
{
    @Tag.firstItem = dataTable.Rows[0]["Item"].ToString();
}
```

### SelectCommandWithStatus

Executa um comando Select síncrono na DatasetTable e fornece uma mensagem de status. Valor de retorno: DataTable em caso de êxito; caso contrário é nulo.

Parâmetro 1: out int status. Status da operação de seleção:

- 0 - sucesso
- ≠0 - código de erro

Parâmetros 2: out string statusMessage. A mensagem indica o status do comando Selecionar, onde uma string vazia significa sucesso. Os tags configurados na coluna mapeamento receberão o valor da primeira linha. Exemplo:

```
int status;
string statusMessage;
DataTable dataTable;
dataTable = @Dataset.Table.table1.SelectCommandWithStatus(out status, out statusMessage);
if(status == 0 && dataTable != null && dataTable.Rows.Count > 0)
{
@Tag.firstItem = dataTable.Rows[0]["Item"].ToString();
}
```

#### UpdateCommand

Atualiza a linha atual da DatasetTable na posição indicada pela propriedade *CursorPosition* com os valores dos tags configurados na coluna mapeamento. Os valores de retorno são: DataTable em caso de êxito; caso contrário é nulo. O índice da linha atual é definido pela propriedade *CursorPosition*.

#### UpdateCommandWithStatus

Atualiza a linha atual da DatasetTable na posição indicada pela propriedade *CursorPosition* com os valores dos tags configurados na coluna mapeamento e fornece uma mensagem de status. Valor de retorno: DataTable em caso de êxito; caso contrário é nulo.

Parâmetro 1: out int status. Status da operação de atualização:

- 0 - êxito
- ≠0 = código de erro

Parâmetro 2: out string statusMessage. A mensagem indica o status do comando de atualização, onde uma string vazia significa êxito.

#### Access

Tipo de Acesso ao DatasetTable. Valores permitidos:

- 0 - Ler
- 1 - Inserir
- 2 - Ler/Escrever
- 3 - Irrestrito

#### AsyncContents

Contém a TDataTable resultante de um dos comandos assíncronos, tais como <i>Select ou Update.

#### Completed

O valor desta propriedade aumenta quando uma operação assíncrona é concluída.

#### CursorPosition

Define a posição da linha atual na DatasetTable.

**DB**

Fornece o DB configurado em EditDatasetsTables.

**Delete**

Envia um comando Delete assíncrono quando o valor se altera. Exemplo:

```
if (@Dataset.Table.table1.Delete.Equals(0))
@Dataset.Table.table1.Delete= 1;
else
@Dataset.Table.table1.Delete = 0;
```

**DeleteExecuted**

O valor desta propriedade se altera quando o comando Delete assíncrono é concluído.

**Description**

Fornece a Descrição do *DatasetTable*.

**Disable**

Desabilita os comandos para a *DatasetTable* quando o valor é maior que zero. Valores permitidos:

- 0 - Habilita os comandos para o *DatasetTable*
- > 0 - Desabilita os comandos para o *DatasetTable*

**Id**

Identificação (ID) do Objeto (Uso Interno).

**Insert**

Envia um comando Insert assíncrono quando o valor se altera. Exemplo:

```
if (@Dataset.Table.table1.Insert.Equals(0))
@Dataset.Table.table1.Insert= 1;
else
@Dataset.Table.table1.Insert = 0;
```

**InsertExecuted**

O valor desta propriedade é alterado quando o comando Insert assíncrono se completa.

**LastStatus**

Fornece o status da última (mais recente) operação assíncrona. Valores permitidos:

- 0 - êxito
- ≠0 - código de erro

**LastStatusMessage**

Fornece a mensagem de status da última (mais recente) operação assíncrona, onde uma string vazia significa êxito.

**Mapping**

Fornece o mapeamento das colunas da *DataTable* com os tags. Exemplo:

TagName	Coluna
int_Id	ID

txt_Nome	Nome
txt_Descrição	Descrição

Tabela 6-14. Mapping

### Resultado:

```
Tag.int_Id=ID;
Tag.txt_Name=Name;
Tag.txt_Description=Description;
```

### Next

Envia um comando Next assíncrono quando o valor é alterado. Exemplo:

```
if (@Dataset.Table.table1.Next.Equals(0))
@Dataset.Table.table1.Next= 1;
else
@Dataset.Table.table1.Next = 0;
```

### NextExecuted

O valor desta propriedade é alterado quando o comando assíncrono Next se completa.

### RowCount

Fornece o número total de linhas na DatasetTable.

### Select

Envia um comando Select assíncrono quando o valor é alterado. Exemplo:

```
if (@Dataset.Table.table1.Select.Equals(0))
@Dataset.Table.table1.Select = 1;
else
@Dataset.Table.table1.Select = 0;
```

### SelectExecuted

O valor da propriedade se altera quando o comando Select assíncrono é concluído.

### TableName

Fornece o nome da tabela de dados.

### Update

Envia um comando Update assíncrono quando o valor é alterado.. Exemplo:

```
if (@Dataset.Table.table1.Update.Equals(0))
@Dataset.Table.table1.Update = 1;
else
@Dataset.Table.table1.Update = 0;
```

### UpdateExecuted

O valor desta propriedade é alterado quando o comando Update assíncrono completa é concluído.

### WhereCondition

Define a condição Onde para a DatasetTable. A cláusula SQL WHERE é usada para selecionar dados condicionalmente. Exemplo:

Tipo	Preço(\$)	DateAdded
------	-----------	-----------

1	27	10/02/2008
1	120	07/10/2010
2	50	12/01/2009

**Tabela 6-15. SQL WHERE**

```
@Dataset.Table.table1.WhereCondition = "DateAdded > '01/01/2010'" ;
```

O comando retornará somente os itens onde DateAdded for maior que 01/01/2010

#### *Class ModuleDataset*

Métodos do Runtime e propriedades para o Módulo Dataset.

#### DB

Fornece acesso aos objetos DatasetDB.

#### File

Fornece acesso aos objetos DatasetFile.

#### Query

Fornece acesso aos objetos DatasetQuery.

#### Table

Fornece acesso aos objetos DatasetTable.

#### **Namespace Script**

Objetos Runtime e métodos relacionados à programação (Script).

#### *Class ModuleScript*

Métodos do Runtime e propriedades para o Módulo Script.

#### RunTasksSimultaneous

Habilita a execução simultânea de tarefas (Multi-Threading). Valores permitidos:

- 0 - Execução simultânea NÃO permitida
- 1 - Execução simultânea permitida

#### Task

Acesso aos objetos ScriptTask.

Sintaxe:

```
Script.Task.<TaskName>.
```

#### UserClass

Acesso aos objetos ScriptUserClass.

Sintaxe:

```
Script.UserClass.<UserClassName>.
```

**Class ScriptTask**

Classe de Tarefas Script.

**Description**

Fornece a Descrição da ScriptTask configurada em EditScriptsTasks.

**Disable**

Desabilita a execução da ScriptTask quando o valor é maior que zero. Valores permitidos:

- 0 - êxito
- ≠0 - código de erro

**Domain**

Indica o Domínio da ScriptTask. Valores permitidos:

- 0 - Servidor
- 1 - Cliente

No caso de Servidor o ScriptTask é executado no contexto do Servidor e não pode acessar os objetos Cliente, isto é, telas ou tags de domínio do cliente. Já no caso de Cliente o ScriptTask é executado em cada Cliente onde o ModuleScript esteja em execução.

**ErrorMessage**

Contém a mensagem com o último (mais recente) erro ocorrido na execução do script.

**Event**

Reservado para uso futuro.

**ExecutionsCount**

Fornece a quantidade de execuções desde que o módulo foi iniciado.

**LastCPUTime**

Fornece o tempo da UCP utilizado durante a última (mais recente) execução do script.

**LastExecution**

Contém o TimeSpan da última (mais recente) execução do script.

**Period**

Período de tempo exigido para executar uma Tarefa. Esta propriedade representa o valor configurado em EditScriptsTasks.

**Running**

Indica se o script atual está em execução. Valores permitidos:

- 0 - ScriptTask NÃO está em execução
- 1 - ScriptTask está em execução

**StopExecutionOnError**

Indica se a execução do script deve ser interrompida em caso de erro.

**Trigger**

Fornece o Tag ou Objeto que dispara a execução da Tarefa. Esta propriedade representa o valor configurado em EditScriptsTasks.

**Namespace Display**

Objetos Runtime e métodos relacionados às telas da aplicação.

*Class Display*

Propriedades do Runtime para objetos Tela. Sintaxe: Display.<DisplayName>

**Close**

Fecha a tela selecionada.

**Open**

Abre a tela selecionada.

**Description**

Fornece a descrição da tela configurada em EditDisplaysDisplays.

**Id**

Identificação (ID) do Objeto (Uso Interno).

**IsOpened**

Indica se a tela selecionada está aberta.

*Class Layout*

Propriedades do Runtime para objetos Layout. Sintaxe: Layout.<LayoutName>

**OpenCommand**

Abre o layout selecionado.

**Description**

Fornece a descrição do layout, configurado em EditDisplaysLayouts.

**Id**

ID do Objeto (Uso Interno).

**IsOpened**

Indica se o layout selecionado está aberto.

**Namespace Report**

Objetos de Runtime e métodos relacionados ao Módulo Relatório.

*Class ModuleReport*

Acesso aos objetos <ReportItem>.

**Class ReportItem**

Propriedades de Runtime para objetos Relatório.

**OpenCommand**

Abre o relatório selecionado. Valores de retorno:

- True - êxito
- False - erro

O ReportViewer pode ser usado para visualizar o conteúdo do relatório.

**SaveCommand**

Salva o relatório selecionado no caminho indicado pela propriedade SaveFileName. Valores de retorno:

- True - êxito
- False - erro

**Append**

Indica se o relatório será sobreescrito ou acrescentado nos comandos Salvar. Valores permitidos:

- 0 - Relatório sobreescrito nos comandos Salvar
- 1 - Relatório acrescentado nos comandos Salvar

**Completed**

O valor desta propriedade aumenta quando a operação Salvar ou Carregar é concluída.

**Description**

Fornece a descrição do ReportItem configurada em EditReportsReports.

**Disable**

Desabilita as operações ReportItem quando o valor é maior que zero. Valores de retorno:

- 0 - Habilita o ReportItem
- >0 - Desabilita o ReportItem

**Id**

Identificação (ID) do Objeto (Uso Interno).

**LastStatus**

Fornece ou define o status do último ReportItem processado. Valores permitidos:

- Success = 0
- InvalidMode = 1
- Disabled = 2
- NoObjectsConfigured = 3
- ReportException = 4
- InvalidContent = 5
- ModuleStopped = 6
- ModulePaused = 7

**LastStatusMessage**

Fornece ou define a mensagem de status do último ReportItem processado. Valores permitidos:

- Success
- InvalidMode
- Disabled
- NoObjectsConfigured
- ReportException
- InvalidContent
- ModuleStopped
- ModulePaused

**OpenExecuted**

Fornece ou define o estado OpenExecuted no relatório. Valores permitidos:

- 0 - Comando Abrir não executado
- 1 - Comando Abrir executado

**Padding**

Fornece o valor de ajuste no Relatório. Valores permitidos:

- Compact = 0
- PadRight = 1
- PadLeft = 2

**SaveExecuted**

Fornece ou define o estado SaveExecuted no relatório. Valores permitidos:

- 0 - Comando Salvar não executado
- 1 - Comando Salvar executado

**SaveFileName**

Fornece ou define o caminho completo usado ao salvar o Relatório.

**SaveFormat**

Fornece ou define o formato de salvamento no Relatório. Valores permitidos:

- XPS = 0
- Html = 1
- Unicode = 2
- ASCII = 3
- PDF = 4

**Namespace Info**

Objetos Runtime e métodos relacionados às informações do projeto.

**Class ModuleInfo****Trace**

Rastreia um objeto do sistema. As referências a este objeto são exibidas na janela de rastreamento.  
Parâmetro: string text. Exemplo:

@Info.Trace("Digital1")	(VB)
@Info.Trace("Digital1");	(C#)

**License**

Fornece o objeto InfoLicense que representa a licença atual.

**ExecutionPath**

Fornece o caminho de execução onde estão instalados os módulos executáveis do sistema.

**Module**

Fornece o objeto InfoModuleList que representa a lista dos módulos.

**OnlineConfig**

Fornece o estado de configuração online. Valores permitidos:

- TRUE - configuração online habilitada
- FALSE - configuração online desabilitada

**Project**

Fornece o objeto InfoProjectVersion que representa a informação da versão do projeto.

**ProjectSettings**

Fornece o objeto InfoProjectSettings que representa a informação das configurações do projeto.

**TestMode**

Fornece o estado do modo teste do projeto. Valores permitidos:

- TRUE - Projeto está no modo teste
- FALSE - Projeto não está no modo teste

**TestModeSync**

Fornece o estado de sincronização do modo teste. Valores permitidos:

- 0 - sincronização do modo teste está desabilitada
- ≠0 - sincronização do modo teste está habilitada

A sincronização do modo Teste permite ao Modo Teste trabalhar com os valores disponíveis dos tags atuais a partir da execução da inicialização.

**Nota:**

**TestModeSync:** Esta funcionalidade está disponível apenas na versão Enterprise.

### *Class InfoProjectVersion*

#### **CurrentBuild**

Fornece o número atual do compilador.

#### **DateCreated**

Fornece o objeto `DateTime` que representa a data em que a versão do projeto foi criada.

#### **DateModified**

Fornece o objeto `DateTime` que representa a data em que a versão do projeto foi modificada.

#### **Description**

Fornece a descrição da versão do projeto.

#### **ProductFamily**

Fornece a família do produto desta versão do projeto. Valores possíveis:

- Student
- Express
- Lite
- Enterprise

#### **ProductModel**

Fornece o modelo do produto da versão do projeto. Valores possíveis:

- 75
- 150
- 300
- 500
- 1500
- 2500
- 5000
- 15000
- 25000
- 50000
- 100000
- 1000000

#### **ProductName**

Fornece o nome do produto da versão do projeto. Valor possível: *BluePlant*.

#### **ProductVersion**

Fornece a versão do produto desta versão do projeto.

#### **ProjectName**

Fornece o nome do projeto desta versão do produto.

**ProjectPath**

Fornece o caminho desta versão do projeto.

**TargetFramework**

Fornece o framework destino desta versão do projeto.

**Versionstring**

Fornece a string desta versão do projeto. Exemplo: 1.0 (Editing).

*Class InfoProjectSettings***CultureInfo**

Fornece o idioma do projeto.

*Class InfoModuleList***Alarm**

Fornece o objeto InfoModule para o módulo Alarme.

**DataSet**

Fornece o objeto InfoModule para o módulo DataSet.

**Device**

Fornece o objeto InfoModule para o módulo Dispositivo.

**Display**

Fornece o objeto InfoModule para o módulo Tela.

**Historian**

Fornece o objeto InfoModule para o módulo Historiador.

**ModuleInformation**

Fornece o objeto InfoModule para o módulo ModuleInformation.

**OPCServer**

Fornece o objeto InfoModule para o módulo OPCServer.

**PropertyWatch**

Fornece o objeto InfoModule para o módulo PropertyWatch.

**Report**

Fornece o objeto InfoModule para o módulo Report.

**ReportServer**

Fornece o objeto InfoModule para o módulo ReportServer.

**Script**

Fornece o objeto *InfoModule* para o módulo *Script*.

**Security**

Fornece o objeto *InfoModule* para o módulo *Segurança*.

**Server**

Fornece o objeto *InfoModule* para o módulo *Servidor*.

**TraceWindow**

Fornece o objeto *InfoModule* para o módulo *TraceWindow*.

***Class InfoLicense*****AllowedRichClients**

Fornece o estado *AllowedRichDevices*. Valores permitidos:

- 0 - clientes Rich não permitidos
- ≠0 - clientes Rich permitidos

**AllowedRunInstances**

Fornece o estado *AllowedRunInstances*. Valores permitidos:

- 0 - instâncias de execução não permitidas
- ≠0 - instâncias de execução permitidas

**AllowedTagElements**

Fornece o estado *AllowedTagElements*. Valores permitidos:

- 0 - elementos do Tag não permitidos
- ≠0 - elementos do Tag permitidos

**AllowedWebClients**

Fornece o estado *AllowedWebClients*. Valores permitidos:

- 0 - clientes Web não permitidos
- ≠0 - clientes Web permitidos

**DateCreated**

Fornece a data e hora em que a licença foi criada.

**DateModified**

Fornece a data e hora em que a licença foi modificada.

**ExpirationDate**

Fornece a data e hora em que a licença expira.

**LicenseMedia**

Fornece informação sobre a mídia da licença. Valores possíveis:

- Hardkey (produto descontinuado)
- Softkey

**LicenseType**

Fornece informação sobre o tipo da licença. Valores possíveis:

- None
- Run
- Eng\_Run
- Developer

**ProductFamily**

Fornece a família do produto da licença. Valores possíveis:

- Student
- Express
- Lite
- Enterprise

**ProductModel**

Fornece um valor numérico que caracteriza o modelo do produto licenciado considerando as seguintes opções: 75, 150, 300, 500, 1.500, 2.500, 5.000, 15.000, 25000, 50.000, 100.000, 1.000.000.

**ProductVersion**

Fornece a versão de produto da licença.

**SerialNumber**

Fornece o número de série da licença.

**ServerConnected**

Fornece o endereço completo do servidor.

**AllowedEngineeringUsers**

Fornece o estado AllowedEngineeringUsers. Valores permitidos:

- 0 - usuários de engenharia não permitidos
- ≠0 - usuários de engenharia permitidos

**AllowedDevices**

Fornece o estado AllowedDevices. Valores permitidos:

- 0 - dispositivos não permitidos
- ≠0 - dispositivos permitidos

### *Class InfoModule*

#### **IsPaused**

Fornece ou define o estado IsPaused do módulo. Valores permitidos:

- TRUE - módulo está pausado
- FALSE - módulo não está pausado

#### **IsRunning**

Fornece ou define o estado IsRunning do módulo. Valores permitidos:

- TRUE - módulo em execução
- FALSE - módulo não está em execução

### **Namespace Server**

Objetos Runtime e métodos relacionados ao servidor.

#### *ServerStation*

A classe ServerStation contém informações sobre o computador no qual o Servidor do Runtime (programa TServer.exe) está sendo executado.

#### **LoadProjectVersion**

Carrega o projeto mostrado pelo caminho especificado. O caminho deve apontar para um projeto válido no servidor. Valores permitidos:

- TRUE - em caso de êxito
- FALSE - em caso de falha

Parâmetro associado: string projectPathName.

#### **SwitchToStandby**

Transfere a execução para o computador reserva se a redundância está habilitada. Valores permitidos:

- TRUE - em caso de êxito
- FALSE - em caso de falha

#### **HttpAddress**

Fornece o endereço http do servidor.

#### **IsPrimary**

Fornece o estado *IsPrimary*. Valores permitidos:

- TRUE - servidor é principal
- FALSE - servidor não é principal

#### **IsRedundancyEnabled**

Fornece o estado redundância de. Valores permitidos:

- TRUE - redundância habilitada
- FALSE - redundância desabilitada

**IsSecondary**

Fornece o estado *IsSecondary*. Valores permitidos:

- TRUE - servidor é secundário
- FALSE - servidor não é secundário

**IsStandByActive**

Fornece o estado *IsStandByActive*. Valores permitidos:

- TRUE - servidor reserva está ativo
- FALSE - servidor reserva está inativo

**IsSwitchToPrimaryEnabled**

Fornece o estado *IsSwitchToPrimaryEnabled*. Valores permitidos:

- TRUE - opção de troca para o principal habilitada
- FALSE - opção de troca para o principal desabilitada

**ServerStation.ComputerIP**

Fornece ou define o IP do computador do servidor.

**ServerStation.ComputerName**

Fornece ou define o nome do computador do servidor.

**ServerStation.Date**

Fornece a data do servidor.

**ServerStation.Day**

Fornece o dia do mês do servidor.

**ServerStation.DayOfWeek**

Fornece o dia da semana do servidor.

**ServerStation.DayOfYear**

Fornece o dia do ano do servidor. Valores permitidos: 1 a 366.

**ServerStation.Hour**

Fornece a hora do dia do servidor.

**ServerStation.Minute**

Fornece o componente minuto da data do servidor.

**ServerStation.Month**

Fornece o componente mês da data do servidor.

**ServerStation.Now**

Fornece a data local do servidor e offset de tempo.

**ServerStation.Second**

Fornece o componente segundo da data do servidor.

**ServerStation.ShutDown**

Fornece ou define o estado de desligamento do servidor. Valores permitidos:

- TRUE - servidor está sendo desligado
- FALSE - servidor não está sendo desligado

**ServerStation.Startup**

Fornece o estado de inicialização do servidor. Valores permitidos:

- TRUE - servidor foi inicializado
- FALSE - servidor não foi inicializado

**ServerStation.Ticks**

Fornece o número de marcações que representam a data e hora do servidor.

**ServerStation.Time**

Fornece a hora do dia do servidor.

**ServerStation.Year**

Fornece o componente ano da data do servidor.

**TimeMs**

Fornece a hora do dia do servidor (inclusive milissegundos).

**Namespace Client**

Objetos de Runtime e métodos relacionados ao Namespace do Cliente.

*Class ClientStation*

A classe ClientStation contém informações sobre o computador no qual o cliente está em execução (TVisualizer.Exe ou clientes Web).

**ChangeUserPassword**

Altera a senha do usuário referenciado. Valores permitidos:

- TRUE - em caso de sucesso
- FALSE - em caso de falha

Parâmetros associados: string username, string oldPassword e string newPassword.

**CloseDisplay**

Fecha a tela referenciada. Parâmetro associado: string displayName.

**GetPasswordHint**

Fornece a dica de senha para o usuário referenciado. Valor de Retorno: a dica da senha. Parâmetro associado: string userName.

**Locale**

Retorna o texto localizado. Valor de Retorno: texto localizado. Parâmetro associado: string text.

**LogOn**

Executa o logon do usuário especificado com a senha especificada. Parâmetros associados: string username e string password. Valores de retorno:

- OK = 0
- ServerNotAvailable = 1
- InvalidLogon = 2
- ServerNotConnected = 3
- UserBlocked = 4
- UserDeleted = 5
- CannotStartModule = 6
- InvalidUserName = 10
- InvalidPassword = 11
- PermissionsRestrictions = 12
- UnknownError = 99

**LogOnGuest**

Executa o logon do usuário como *Guest*.

**OpenDisplay**

Abre a tela na página do último layout. Parâmetro associado: string displayName

**OpenDisplayAtIndex**

Abre a tela na página do layout especificado pelo índice. Parâmetros: string displayName e int index.

**OpenLayout**

Abre o layout referenciado. O parâmetro associado é: string layoutName.

**OpenPopupNote**

Abre uma nota de popup com título e descrição especificados. Os parâmetros associados são: string title, string description, bool isReadOnly, double left e double top.

**OpenPreviousPage**

Abre a página exibida anteriormente. Valores permitidos:

- TRUE - em caso de sucesso
- FALSE - em caso de falha

**SaveLayoutAsImage**

Salva o layout como uma imagem. O caminho para o arquivo será definido no diálogo seguinte.

**SaveLayoutAsImageFile**

Salva o layout como arquivo de imagem. O parâmetro associado é: string fileName.

**SwitchToStandby**

Troca o servidor para o modo Standby. Valores permitidos:

- TRUE - em caso de sucesso
- FALSE - em caso de falha

**BeepOff**

Fornece ou define o estado de beep desligado do cliente. Valores permitidos:

- TRUE - beep desligado
- FALSE - beep ligado

**BlinkFast**

Fornece a propriedade de piscar rápido do cliente. Esta propriedade é uma variável digital que alterna entre 0 e 1, permanecendo 250 milissegundos em cada estado.

**BlinkSlow**

Fornece a propriedade de piscar devagar. Esta propriedade é uma variável digital que alterna entre 0 e 1, permanecendo 500 ms em cada estado.

**ComputerIP**

Fornece o IP do computador.

**ComputerName**

Fornece o nome do computador.

**CultureInfo**

Fornece ou define a seleção de idioma do cliente.

**CurrentPage**

Fornece o nome da página do cliente exibida atualmente.

**CurrentUser**

Fornece o usuário atual do cliente.

**Date**

Fornece a data do cliente.

**DateTime**

Fornece a data e hora do cliente.

**Day**

Fornece o dia do mês do cliente.

**DayOfWeek**

Fornece o dia da semana do cliente.

**DayOfYear**

Fornece o dia do ano do cliente.

**Dictionary**

Fornece ou define o dicionário do cliente.

**Hour**

Fornece o componente hora da data do cliente.

**InputPassword**

Fornece ou define a senha de entrada do cliente. Esta é uma variável auxiliar usada na janela de logon padrão do sistema.

**InputUserName**

Fornece ou define o nome do usuário de entrada do cliente. Esta é uma variável auxiliar usada na janela de logon padrão do sistema.

**IsWebBrowser**

Fornece o estado IsWebBrowser do cliente.

**LayoutName**

Fornece ou define o nome do layout do cliente.

**Millisecond**

Fornece o componente milissegundo da data do cliente.

**Minute**

Fornece o componente minuto da data do cliente.

**Month**

Fornece o componente mês da data do cliente.

**Now**

Fornece a data local do cliente e offset de tempo.

**OnScreenKeyboard**

Fornece ou define o estado de tela e teclado do cliente.

**PreviousPage**

Fornece o nome da página anterior do cliente.

**Second**

Fornece o componente segundo da data do cliente.

**SelectedPage**

Fornece ou define o nome da página selecionada do cliente.

---

**ServerHttpAddress**

Fornece o endereço http do cliente.

**Shutdown**

Fornece ou define o estado de desligamento do cliente. Valores permitidos:

- TRUE - cliente está desligado
- FALSE - cliente não está desligado

**SimulationAnalog**

Fornece uma variável de simulação [int] analógica, a qual varia de 0 a 100 (em incrementos de 1), retorna a 0 em um ciclo e então repete o mesmo padrão (onda dente de serra).

**SimulationDigital**

Fornece uma variável de simulação digital que alterna entre 0 e 1, permanecendo 3 segundos em cada estado.

**SimulationDouble**

Fornece uma variável de simulação [double] analógica, a qual varia de 0 a 100, retorna de 100 a 0, e então repete o mesmo padrão.

**Startup**

Fornece o estado de inicialização do cliente. Valores permitidos:

- TRUE - cliente foi inicializado
- FALSE - cliente não foi inicializado

**Ticks**

Fornece o número de marcações que representam a data e hora do cliente.

**Time**

Fornece o horário do dia do cliente.

**TimeMs**

Fornece o horário do dia do cliente incluindo milissegundos.

**Tomorrow**

Fornece o componente dia da data de amanhã do cliente.

**UserName**

Fornece o nome de usuário do cliente.

**UtcNow**

Fornece a data UTC e offset de horário do cliente.

**Year**

Fornece o componente ano da data do cliente.

**Yesterday**

Fornece o componente dia da data de ontem do cliente.

**CurrentPage**

Fornece o nome da pagina atual.

**DateTime**

Fornece a data e hora da página do cliente exibida atualmente.

**IsWebBrowser**

Fornece o estado IsWebBrowser do cliente. Valores permitidos:

- TRUE - o cliente está em execução com um navegador da Web
- FALSE - o cliente não está em execução com um navegador da Web

**OnScreenKeyboard**

Fornece ou define o estado de tela e teclado do cliente. Parâmetro associado: public bool OnScreenKeyboard { get; set; }. Os valores permitidos são:

- TRUE - funcionalidade do teclado e tela está ativa
- FALSE - funcionalidade do teclado e tela não está ativa

**PreviousPage**

Fornece o nome da página anterior do cliente.

**ServerHttpAddress**

Fornece o endereço http do cliente.

**SimulationAnalog**

Fornece uma variável de simulação [int] analógica, a qual varia de 0 a 100 (em incrementos de 1), retorna a 0 em um ciclo e então repete o mesmo padrão (onda dente de serra).

**SimulationDouble**

Fornece uma variável de simulação [double] analógica, a qual varia de 0 a 100, retorna de 100 a 0, e então repete o mesmo padrão.

## Configurações Avançadas

Esta seção contém informação adicional sobre os aplicativos do BluePlant incluindo:

- Linha de Comando
- Executando o BluePlant como um Serviço do Windows
- Clientes Remotos
- Instalando o 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.

**Nota:**

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.

##### **Nota:**

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:

```
"C:\Program Files\Altus\BluePlant\bp-9.1\TStartup.exe" /project:"C:\BluePlant Projects\Project1.tproj"
```

```
"C:\Program Files\Altus\BluePlant\bp-9.1\TStartup.exe"  
/project:"C:\BluePlantProjects\Project1.tproj" /username:Administrator /port1:3101
```

```
"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.

##### **Nota:**

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:

" C:\Program Files\Altus\BluePlant\bp-9.1\TVisualizer.exe"

" C:\Program Files\Altus\BluePlant\bp-9.1\TVisualizer.exe" /username:Administrator /ip1:192.168.1.1 /port1:3101

" 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.

**Nota:**

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, zipe a pasta e envie para o suporte da Altus descrevendo o problema ocorrido.

Exemplos:

" C:\Program Files\Altus\BluePlant\bp-9.1\TraceWindow.exe"

" C:\Program Files\Altus\BluePlant\bp-9.1\TraceWindow.exe" /username:Administrator /ip1:192.168.1.1 /port1:3101

" 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.

**Nota:**

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:

" C:\Program Files\Altus\BluePlant\bp-9.1\PropertyWatch.exe"

" C:\Program Files\Altus\BluePlant\bp-9.1\PropertyWatch.exe" /username:Administrator /ip1:192.168.1.1 /port1:3101

" 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.

##### **Nota:**

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 IP do Servidor Secundário

**/port2:** (opcional) porta TCP do Servidor Secundário

Exemplos:

" C:\Program Files\Altus\BluePlant\bp-9.1\ModuleInformation.exe"

" C:\Program Files\Altus\BluePlant\bp-9.1\ModuleInformation.exe" /username:Administrator /ip1:192.168.1.1 /port1:3101

" 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.

##### **Notas:**

- 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.

#### **Notas:**

- 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 também para servir páginas para clientes Web, o usuário também precisa ter um servidor Web habilitado. Consulte o tópico *Installing Web Server in IIS* para informações sobre esse procedimento.

A fim de executar o aplicativo como um serviço do Windows:

```
<.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:

Cliente Rich:

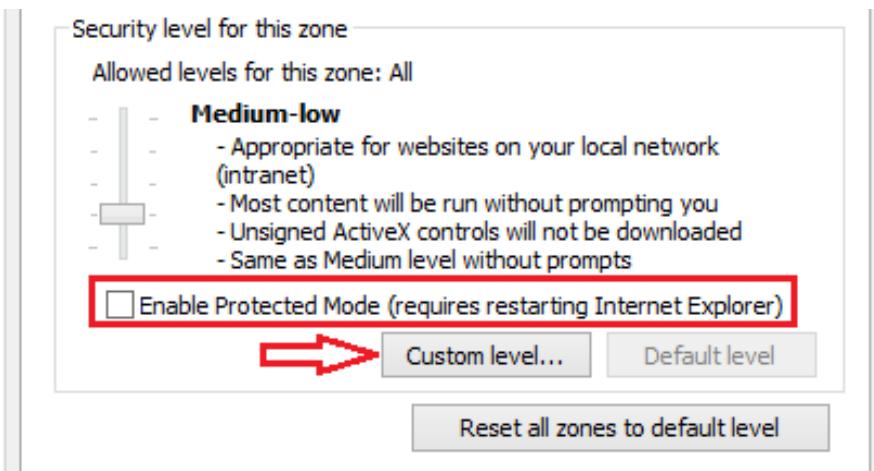
```
http://<server ip or name>/<BluePlant Version>/tvisualizerremote.application
```

Cliente Web:

```
http://<server ip or name>/<BluePlant Version>/tvisualizerweb.xbap
```

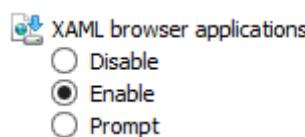
Dependendo da versão do Internet Explorer a seguinte configuração deve ser feita:

Desativar o modo protegido na Zona Internet ou Intranet conforme ilustrado na Figura 6-134.



**Figura 6-134. Desativar o Modo Protegido**

Habilitar aplicações do navegador XAML conforme mostrado na Figura 6-135.



**Figura 6-135. Habilitar Aplicações do Navegador XAML**

Exemplo:

Cliente Remoto Rich:

<http://127.0.0.1/bp-2012.1/TVisualizerRemote.application>

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

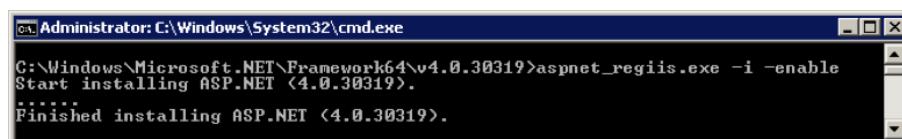
Este procedimento pode ser executado no Windows 7, x64, com o IIS 7.5. Com versões mais antigas do sistema operacional e do IIS, o procedimento pode ser ligeiramente diferente, como a definição de um *Virtual Directory* em vez de *Add Application*.

- Verifique se o IIS está instalado e funcionando, por exemplo, tentando abrir páginas HTML
- 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\*)



**Figura 6-136. 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
[Warning]The HTTP namespace reservation already exists.

C:\Windows\Microsoft.NET\Framework64\v4.0.30319>-

```

**Figura 6-137. 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*

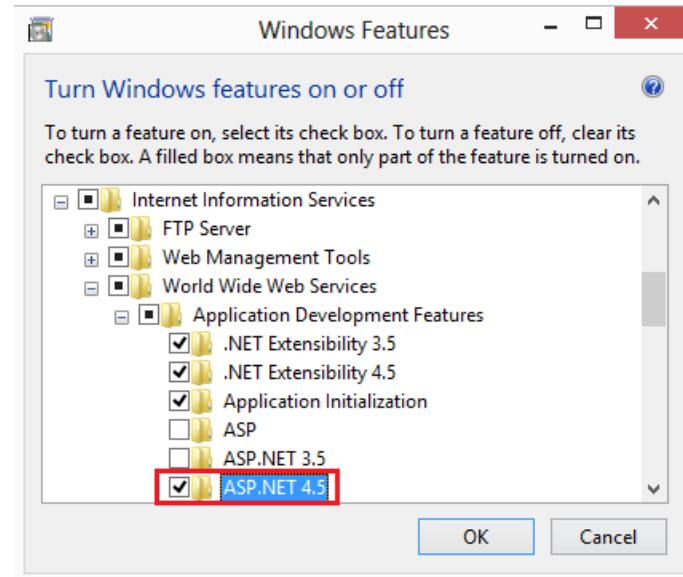
No caso do IIS 8.0 o procedimento está descrito a seguir.

Vá para *Program and Features*, escolha *Turn Windows features on or off*.



**Figura 6-138. Chaveamento de Funcionalidades do Windows**

Habilite a opção *Internet Information Services > World Wide Web Services > Application Development Features > ASP.NET 4.5*.



**Figura 6-139. Funcionalidades do Windows**

Habilite a opção *HTTP Activation* (no .Net Framework 4.5 Advanced Services).

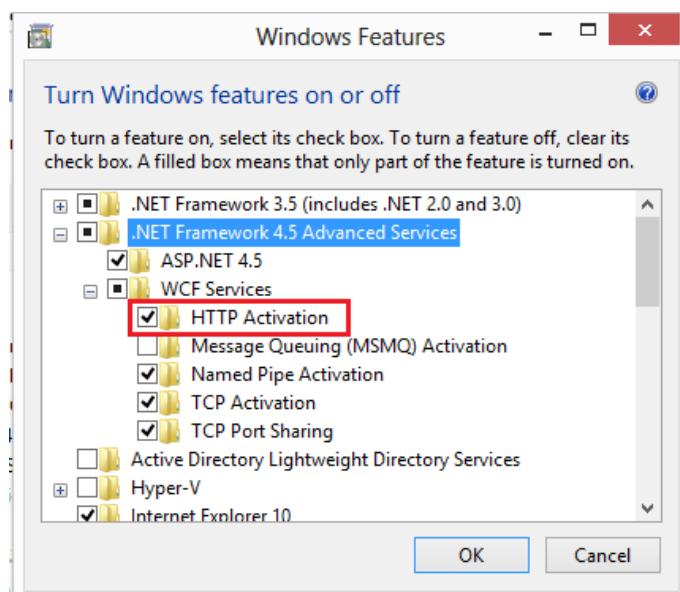


Figura 6-140. Ativação HTTP

Verifique se a extensão ".svc" mapeada para a pasta está mapeada para *aspnet\_isapi.dll* (no site [msdn.microsoft](http://msdn.microsoft.com), link anterior, é explicado como verificar isso para várias versões IIS).

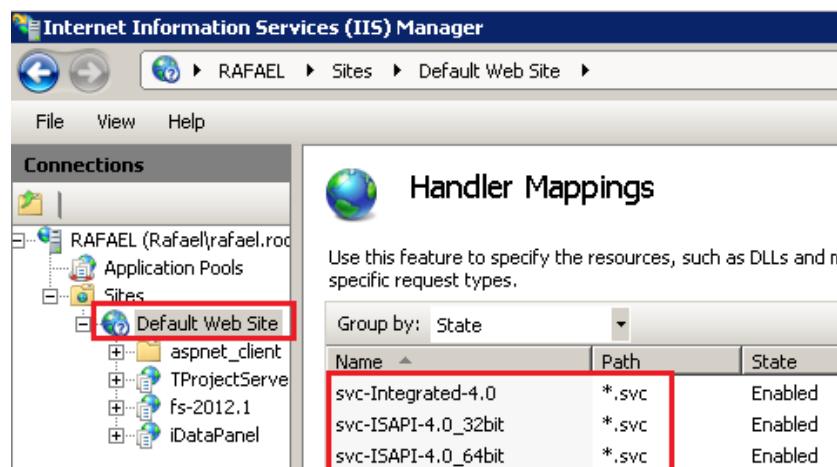


Figura 6-141. Mapeamento da Extensão

Na pasta de instalação do BluePlant, execute o utilitário: *InstallTWebServer.exe /uninstall*, o qual removerá a instalação atual do *TWebServer* conforme ilustrado na Figura 6-142.

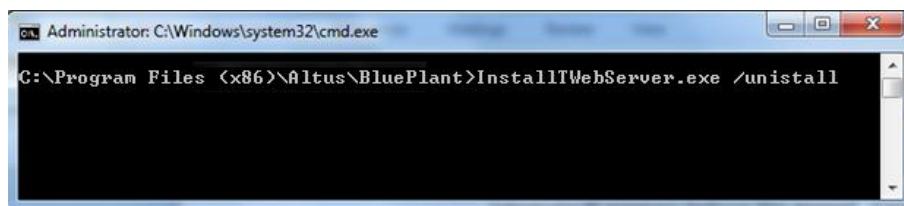


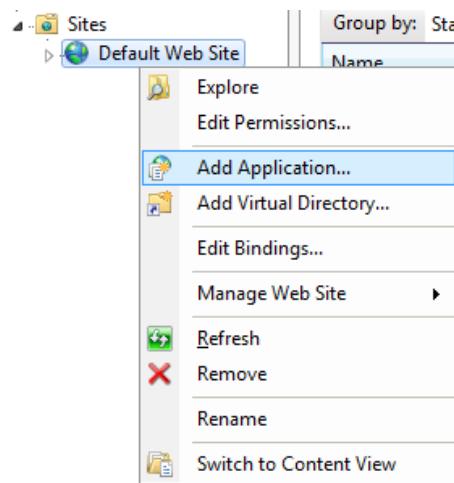
Figura 6-142. Utilitário InstallTWebServer

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.

Se for identificada uma referência a *TWebServer*, apague-a.

No IIS 7.x, abra *Sites/Default Web Site/Add Application* e configure a informação para os serviços conforme ilustrado na Figura 6-143.

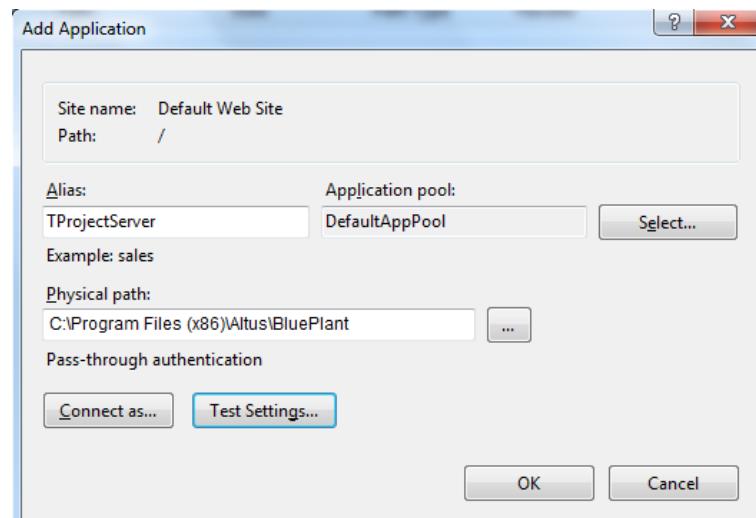


**Figura 6-143. Informação para serviços**

Os ajustes para o *TProjectServer* apresentadas na Figura 6-144 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 6-144. 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-2012.1
- 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

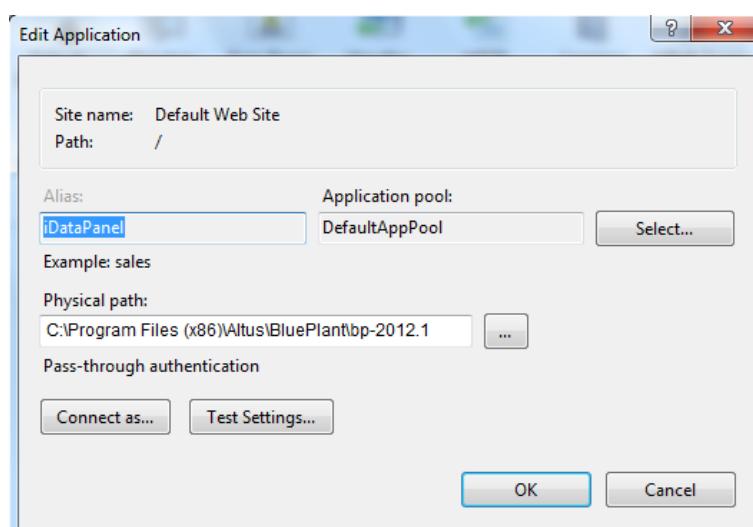
**Nota:**

**Security:** A opção Security deve estar habilitada para *Everyone*, pelo menos para leitura. No IE, use a URL <http://localhost/bp-2012.1/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 Figura 6-145 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 6-145. Serviço *iDataPanel***

**Nota:**

**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*

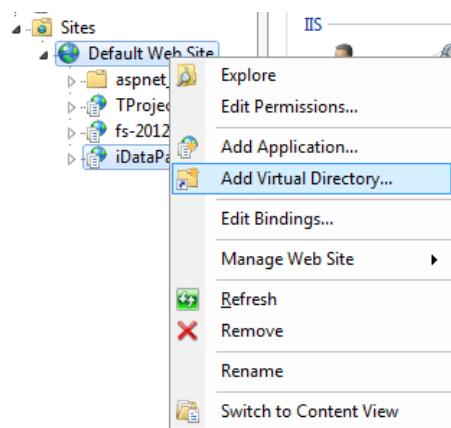


Figura 6-146. Diretório Virtual

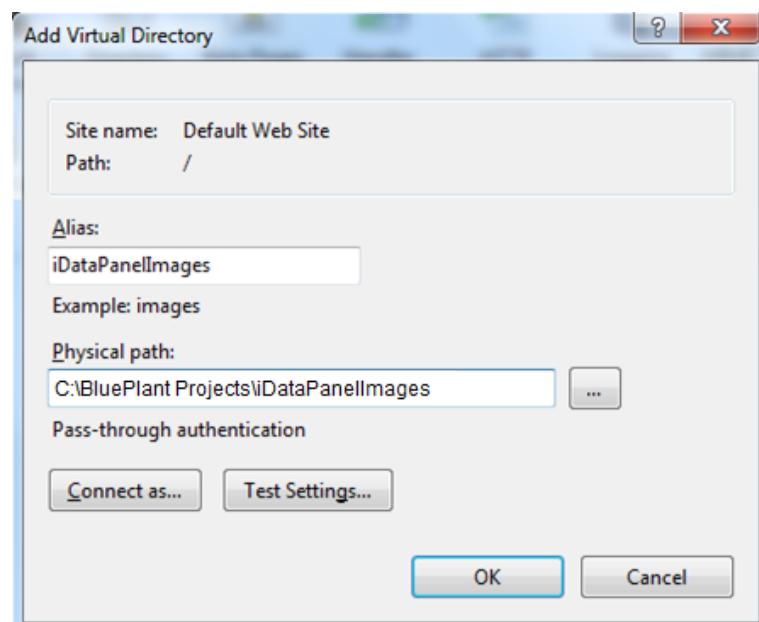


Figura 6-147. Acrescentar Diretório Virtual

#### Notas:

**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 6-148.



Authentication		
Group by: No Grouping		
Name	Status	Response Type
Anonymous Authentication	Enabled	

Figura 6-148. Autenticação

O IIS deve ser reiniciado após a configuração dos serviços.

## 7. 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.

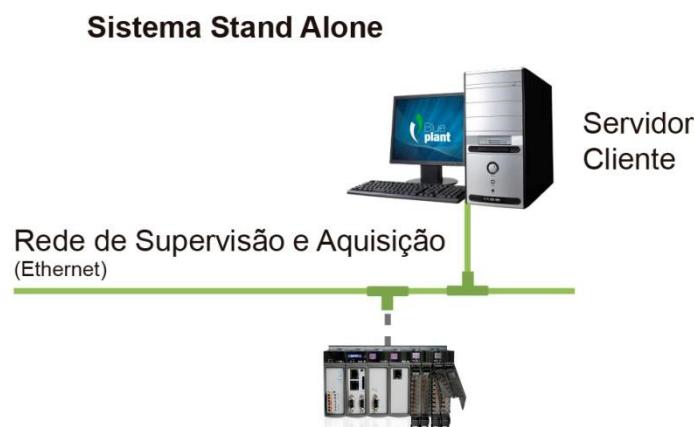
Independentemente do cenário considerado deve-se ter em mente que o BluePlant se constitui em um pacote único de forma que o servidor é sempre um BluePlant. No entanto, todos os módulos do BluePlant como Alarme, Historiador, Dispositivo, Banco de Dados etc, podem estar em computadores diferentes sendo gerenciados pelo Servidor BluePlant gerando assim um sistema distribuído.

Com base nessas premissas podemos ter as configurações descritas a seguir.

### Configurações dos Sistemas

#### Sistema Standalone

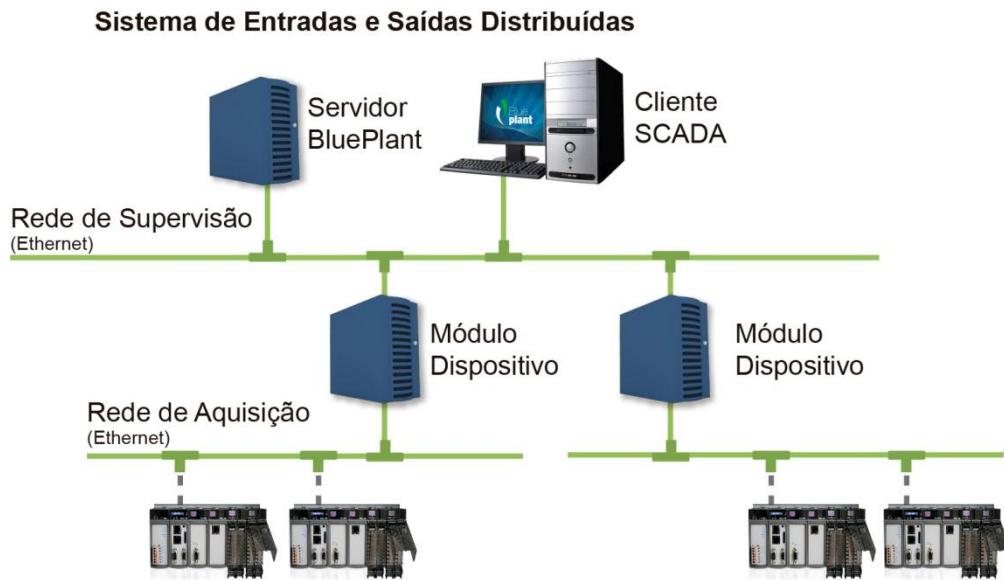
Esse sistema caracteriza-se por uma instalação do BluePlant executando o servidor e o cliente SCADA no mesmo computador.



**Figura 7-1. 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 7-2 ilustra essa configuração.



**Figura 7-2. Sistema de Entradas e Saídas Distribuídas**

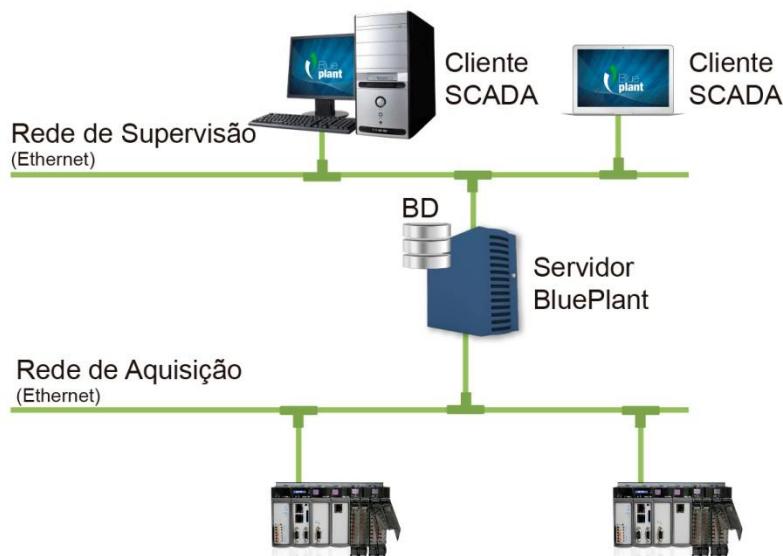
Este modelo também é útil em plantas que contenham dispositivos com porta serial ou comunicações com capacidades limitadas. Colocando servidores de E/S no chão de fábrica, como interface com esses dispositivos, é possível otimizar as comunicações em redes lentas ou de baixa largura de banda melhorando 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 7-3 mostra esse sistema.

### Sistema Cliente Servidor



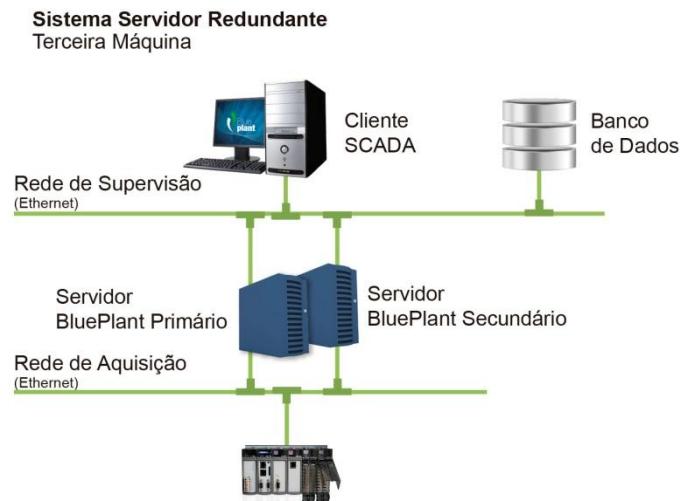
**Figura 7-3. Sistema Cliente e Servidor**

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.

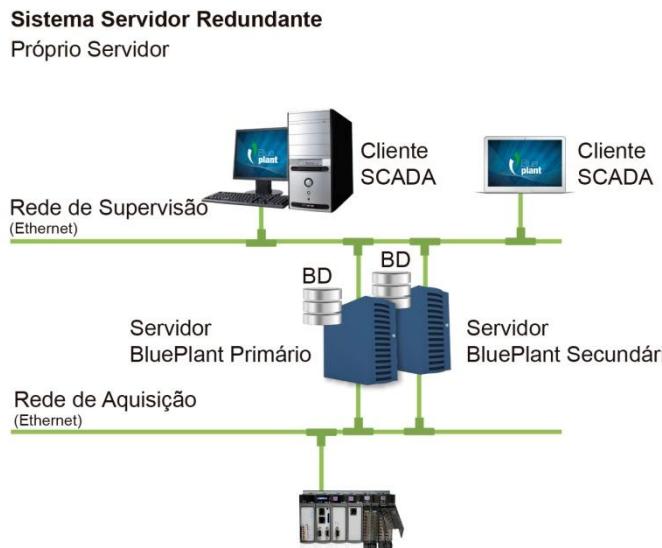
### Sistema de Servidor Redundante

Neste caso temos dois computadores distintos executando os servidores BluePlant, onde a redundância é feita automaticamente pelo próprio supervisório. 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 7-4.
- Banco de Dados Local 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 Figura 7-5.



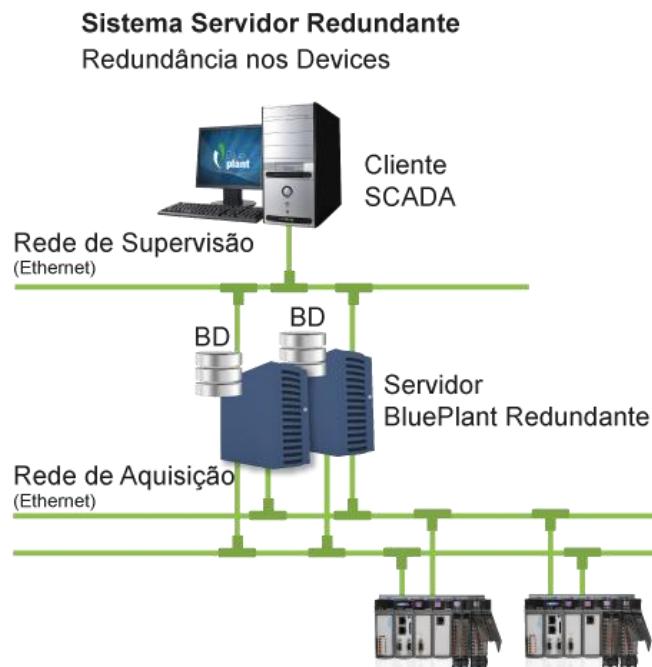
**Figura 7-4. Sistema Servidor Redundante Terceira Máquina**



**Figura 7-5. 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 onde 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) tem opções de 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 supervisório.

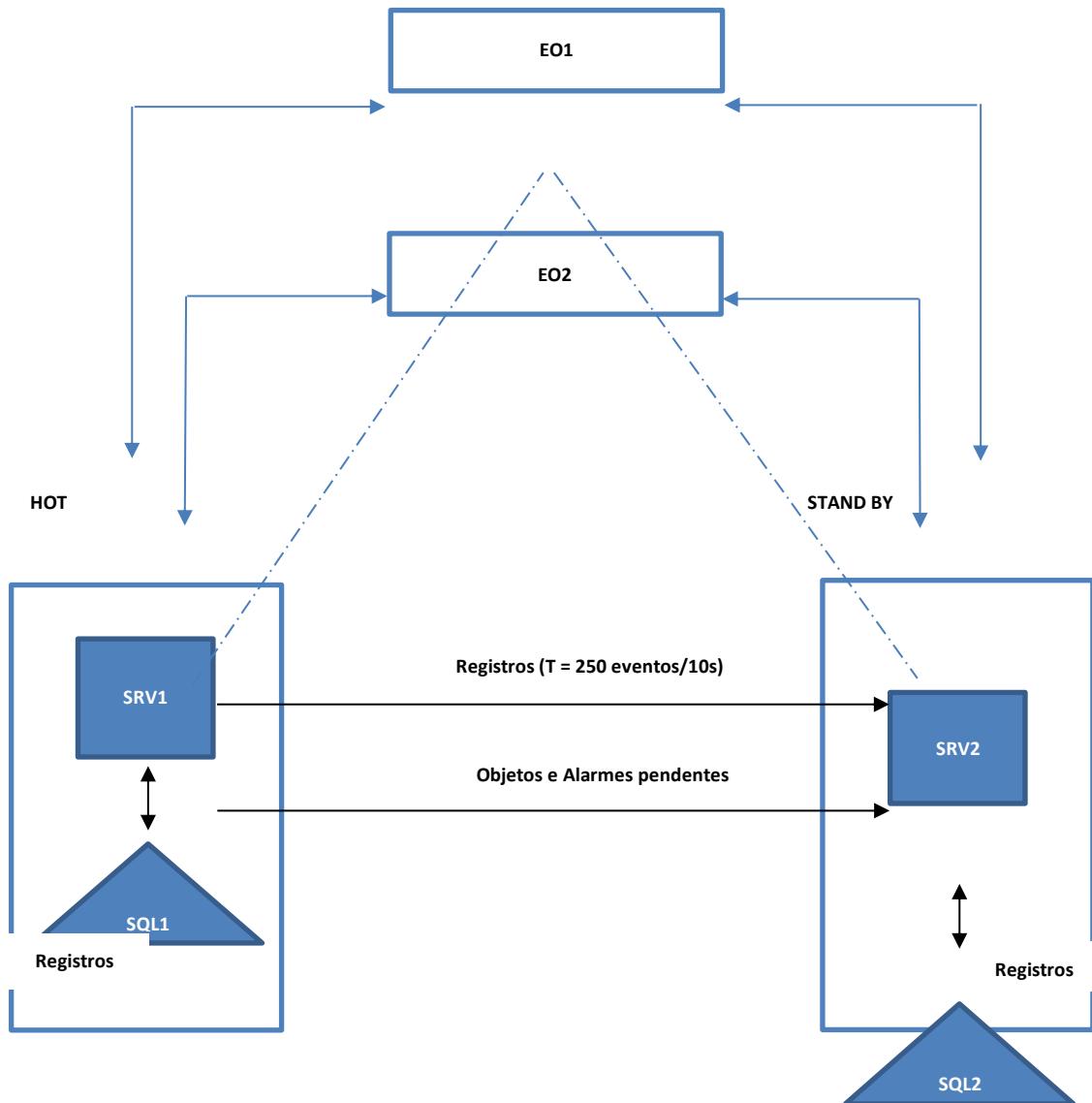


**Figura 7-6. Sistema Servidor Redundante com Redundância nos Dispositivos**

*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 7-7. Mecanismo de funcionamento do Servidor Redundante**

**Notas:**

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

### Operação normal

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 e 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

Quando retirados 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 e 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 é:

- 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 tempo estipulado para poder ser realizado este procedimento é de:

$$\text{taxa média} \left[ \frac{\text{eventos}}{\text{segundo}} \right] = \frac{\text{eventos} [\text{eventos}]}{\text{tempo} [\text{segundos}]}$$

$$\frac{250 [\text{eventos}]}{10 [\text{segundos}]} = \frac{35121 [\text{eventos}]}{\text{tempo} [\text{segundos}]}$$

$$\text{tempo} = \frac{35121 [\text{eventos}]}{25 [\text{eventos/segundo}]}$$

$$\text{tempo} \simeq 23 \text{ minutos}$$

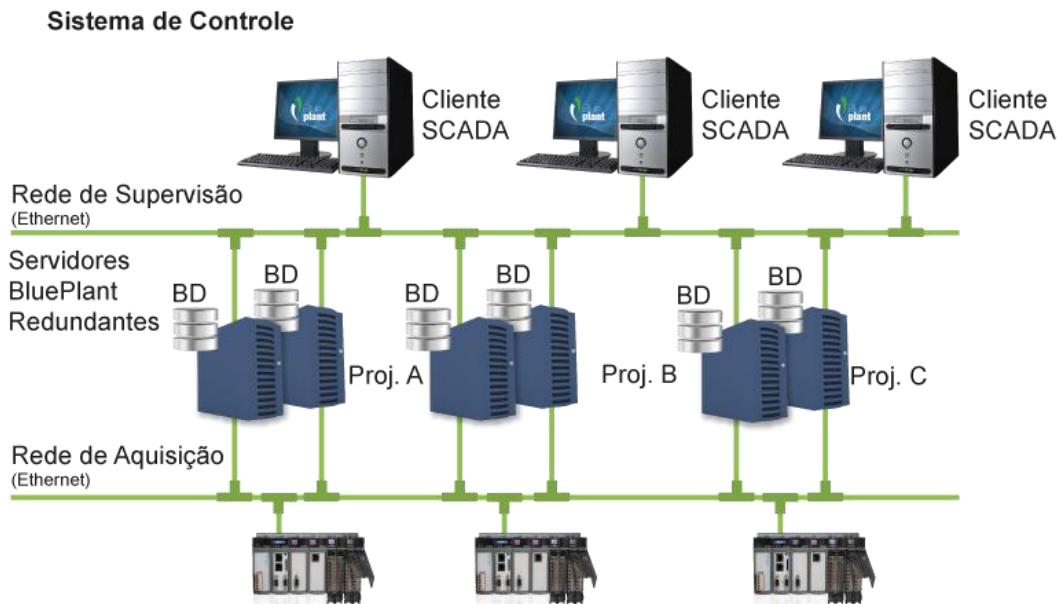
### Recuperação do Sistema

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 o mesmo 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 7-8 ilustra esse sistema.

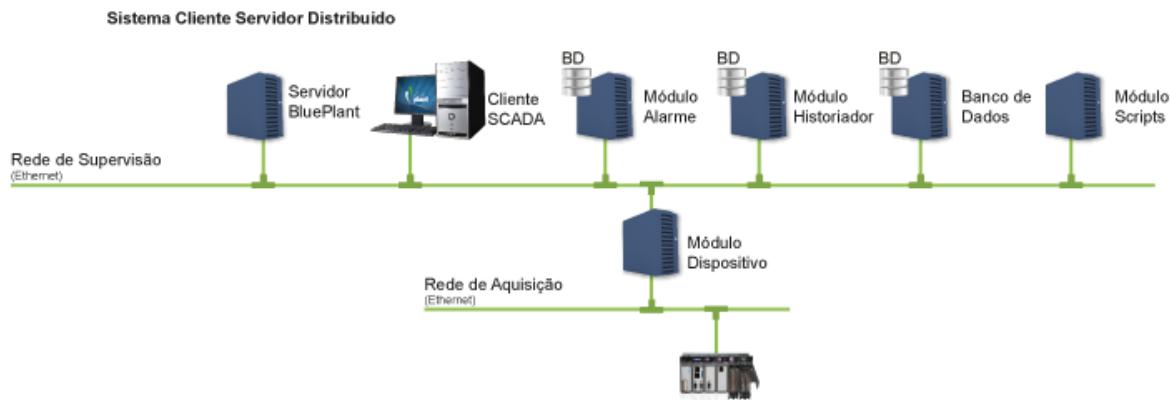


**Figura 7-8. Sistema de Controle**

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

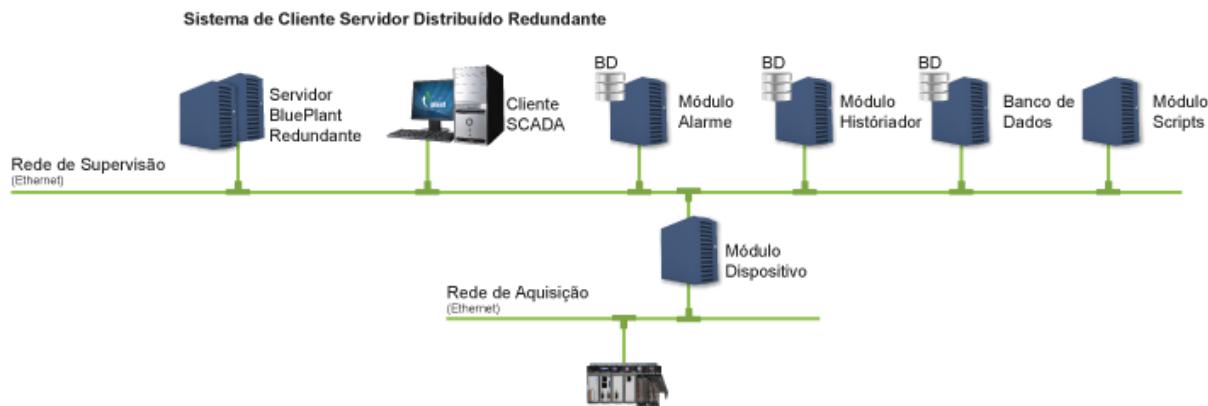
Esse sistema contempla uma máquina BluePlant servidora e módulos Alarme, Historiador, Banco de Dados e Clientes SCADA em outros computadores da rede LAN, de maneira que cada módulo é executado em um computador distinto. A Figura 7-9 mostra esse sistema.



**Figura 7-9. 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 7-10 mostra essa configuração.



**Figura 7-10. 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 os mesmos operem como unidades redundantes entre si reduzindo o número de computadores exigidos.

## 8. Glossário

<b>AppDomain</b>	Limite fornecido pelo CLR (Common Language Runtime) englobando os objetos criados a partir do mesmo escopo da aplicação.
	Os domínios da aplicação ajudam a isolar os objetos criados em uma aplicação de outros objetos criados em outras aplicações. O TRuntime.exe é executado em um AppDomain diferente (módulo do dispositivo ou visualizador) dos outros módulos. Termo relacionado ao Microsoft dotNet.
<b>Assembly</b>	"Assembly" refere-se a um arquivo executável (.EXE) ou biblioteca (.DLL) criada com o uso de código gerenciado e framework Microsoft .NET. Termo relacionado ao Microsoft dotNet.
<b>Designer</b>	Componente de desenvolvimento do gerenciador do programa usado para projetar telas sinóticas. Telas sinóticas são representações gráficas de processos industriais gerados no ambiente de desenho.
<b>Device Module</b>	Módulo que permite a implementação de protocolos de comunicação (Dispositivos) em máquinas remotas com o servidor no qual o projeto está sendo executado, mantendo o Runtime.
<b>Domain, Server Domain, Client Domain</b>	Refere-se aos valores e à localização de objetos quando em execução. "Objeto de domínio de servidor em execução no servidor" significa o objeto em execução no servidor. Valores associados a esse objeto abrangem todo o sistema. "Domínio do cliente" significa que o objeto está sendo executado na estação do cliente e cada máquina cliente pode apresentar valores diferentes.
<b>Internal Module</b>	Características e programas que implementam funções do sistema interno que são executadas sem nenhuma configuração do usuário. Exemplos: tarefa de sincronização de rede e gerador de relatório de plano de fundo. Algumas ferramentas BluePlant, por exemplo, ModuleInformation.exe, podem exibir informações de status destes módulos internos. No entanto, estas informações são exigidas somente para otimizações de sistemas avançados. Exemplo: módulo de sincronização.
<b>Main Project File</b>	"Main Project File" refere-se à base de dados SQL do BluePlant-codificado que contém a configuração do projeto. A extensão .TPROJ mostra o projeto atual em desenvolvimento. A extensão .TENG mostra os projetos Somente Leitura publicados.
<b>Modifier (Device/Points)</b>	Parâmetro auxiliar para marcar pontos de leitura e escrita em um dispositivo, tratamento de arrays, máscaras de bit, strings, swap e outras operações nas quais a definição de Tipo de Dado não pode caracterizá-los completamente (parâmetros adicionais são exigidos para definir o Tipo de Dado).
<b>Module</b>	Programa que acessa o banco de dados em tempo real (RtDB), podendo ser composto de um ou mais Assemblies. Exemplo: Manager.XBAP é um módulo de configuração de projeto que fornece ao usuário o acesso ao seu nome online através do URL do navegador.
<b>Namespace</b>	Espaço de endereço. Todos os objetos criados com Namespace apresentam nomes exclusivos. O Namespace também pode ser entendido como uma configuração para criar hierarquia entre os objetos. Exemplo: todas as variáveis de processo são agrupadas no Tag, namespace, todos os relatórios criados são agrupados no relatório namespace.
<b>ObjectType (RunObj type)</b>	Determina o(s) tipo(s) de objeto(s) de Runtime. Por exemplo, um objeto pode apresentar os seguintes tipos de Runtime: display, report e ScriptFunction, Tela, Relatório ou Função Script.
<b>Objects, RunTime Objects</b>	Objetos de Runtime são os objetos visíveis (através dos seus nomes) acessados via scripts ou exibidos no projeto. Exemplos: tags, Relatórios, Telas e outros. Também deve ser usado pelos desenvolvedores de módulo. Um módulo de alarme, por exemplo, inicializará e atualizará os valores dos objetos de Runtime associados com os alarmes.
<b>ObjectValues, PropertyValue</b>	Objetos de Runtime (tags, Nós de dispositivos, etc.) podem ter uma ou mais propriedades definidas. O objeto TAG (e seu valor) apresenta propriedades, tais como: mínimo, máximo e descrição. Durante a configuração específica do projeto são exigidos os itens "PropertyValue" ou "ObjectValue". Especifique o nome completo até o valor da propriedade final, isto é, TAG.myTag.Min ou Tag.MyTag.Description. Para algumas configurações como itens de alarme, historiador ou pontos do dispositivo especificando o OBJETO principal (Tag.MyTag neste exemplo), o sistema assumirá, por padrão, a propriedade de valor para executar a operação.
<b>ObjectWatch</b>	Utilidade de diagnóstico para verificar e modificar valores dos objetos quando em execução.
<b>Parameters (TagConfiguration)</b>	Definição dos parâmetros de comportamento e valores dos tags de processamento. Exemplo: CounterUP para contadores, ou DeadBand smoothing tags para Inteiros.
<b>Project</b>	Conjunto de configurações BluePlant, desenhos de telas, relatórios, documentos e notas do usuário criadas e editadas como uma entidade única.
<b>projectDB or ProjectDatabase</b>	ProjectDB (ou ProjectDatabase) é uma base de dados proprietária com informações de configuração. Corresponde aos arquivos com extensão <projeto>.Tproj (por exemplo, Atual) ou <projeto>_VERSION.Teng (tipo Release).

<b>Property</b>	Propriedade (valor) associado a um Tag ou a um objeto de Runtime.
<b>RunDB, Runtime Database</b>	Base de dados de tempo real criada quando o projeto está em execução. Todos os objetos de Runtime acessáveis via script (tais como tags e relatórios) são gerados internamente com código encapsulado e acessáveis através da classe RunDB. O RunDB corresponde ao primeiro nível da árvore de objetos de Runtime.
<b>Runtime</b>	Projeto com módulos de Runtime carregados e em execução.
<b>Runtime Startup</b>	Operação que permite a execução de um projeto. Esta operação pode ser executada a partir do programa TStartup.exe na versão atual do BluePlant, ou a partir do TServer.exe na versão publicada do projeto.
<b>RuntimeDB</b>	RuntimeDB é uma cópia do banco de dados do ProjectDB que contém informações específicas sobre o RuntimeDB quando a versão do Runtime é publicada. Corresponde aos arquivos com <projeto>_<versão>.trun.
<b>Tag</b>	Variável de processo. Nome de um Namespace que inclui todas as variáveis criadas pelo usuário em uma configuração do projeto.
<b>Tag type</b>	Define o tipo de objetos no Tag do Namespace. Exemplo: Digital, Analógico e Texto. Esses tags representam uma classe de propriedades acessadas diretamente, tais como: Min, Max, valor, Qualidade. Cada propriedade é criada internamente como ValueType.
<b>Task (Script.Task)</b>	Programa de tarefas escrito em VB.NET (ou C #) executado no servidor ou no cliente durante o Runtime do projeto. A execução será no servidor ou cliente dependendo da propriedade de domínio configurada no script.
<b>TManager</b>	Programa que realiza a configuração do projeto.
<b>Toggle</b>	Inverte o valor de uma variável. Valores maiores que zero são convertidos a zero; zero é convertido ao valor "1".
<b>Partial Trust, Full Trust</b>	Partial Trust: Ambiente no qual uma aplicação tem acesso limitado aos recursos (acesso restrito a arquivos de pastas particulares, execução de outras aplicações, etc.). Aplicações XBAP que são executadas com um navegador (Internet Explorer, por exemplo) devem utilizar o modo "Partial Trust". Full Trust: Ambiente no qual uma aplicação tem acesso a todos os recursos do sistema. Aplicações instaladas em um computador geralmente são executadas no modo "Full Trust". Termo relacionado ao Microsoft dotNet.
<b>TWelcome</b>	BluePlant inicial e tela de seleção do projeto.
<b>Visibility (Tag Visibility)</b>	Refere-se ao sistema de Tag. Os tags podem ser públicos, protegidos ou privados. Público: o valor do Tag durante a execução está disponível para acesso por programas externos via TCP/IP ou Servidor OPC. O valor do Tag é necessariamente global ou compartilhado em todas as estações cliente (Domínio do Servidor). Protegido: somente leitura. Privado: um Tag definido como "Privado" não pode ser acessado por aplicações externas (Servidor OPC, TCP/IP) e será executado em um Cliente ou em um Servidor, de acordo com a configuração da aplicação com as seguintes características: <ul style="list-style-type: none"><li>• Tags usados somente em módulos chamados pelo cliente (tais como telas e Relatórios) e que são executados no escopo deste. Podem apresentar valores diferentes em cada máquina cliente (Domínio do Cliente);</li><li>• Tags usados em modulo servidores, tais como dispositivos e alarmes. Apresentam um único valor no sistema (Domínio do Servidor).</li></ul>
<b>Xbap</b>	XAML browser application (XBAP) é uma aplicação executada dentro de um browser (IE). Termo relacionado ao Microsoft dotNet.