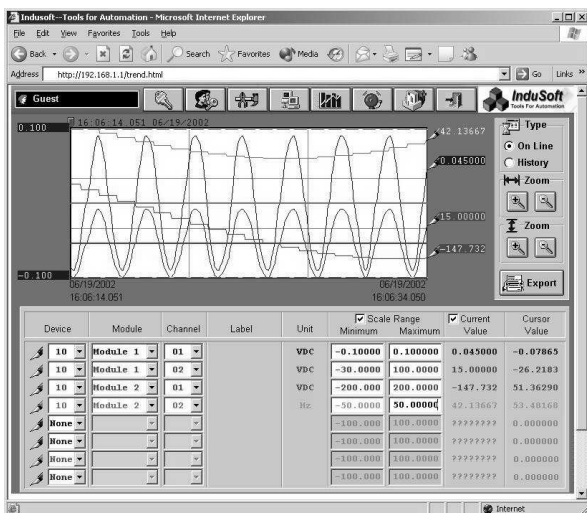
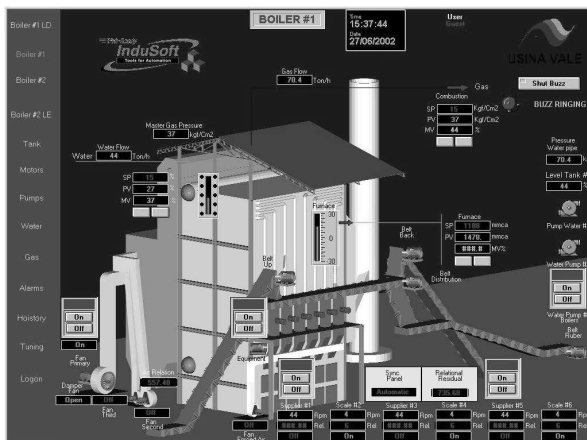
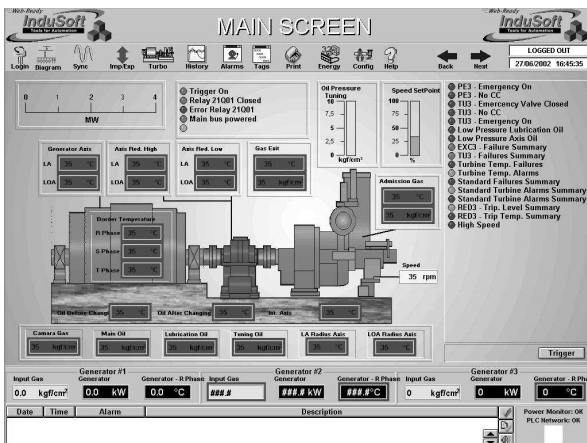


São Paulo, Brasil
Novembro 2004



www.InduSoft.com

Conteúdo

INTRODUÇÃO	5
REQUISITOS DE SISTEMA PARA RODAR O INDUSOFT WEB STUDIO.....	5
INSTALANDO O INDUSOFT WEB STUDIO	6
CRIANDO UM NOVO PROJETO.....	10
CONFIGURAÇÃO DO TAG DATABASE.....	19
TRABALHANDO COM O DATABASE	26
COMUNICAÇÃO.....	27
CONFIGURANDO OS PARÂMETROS DE COMUNICAÇÃO	30
CONFIGURANDO O SISTEMA DE SEGURANÇA.....	39
CRIANDO TAGS NO DATABASE.....	45
CRIANDO NOVAS TELAS	46
CRIANDO A TELA PRINCIPAL A PARTIR DA TELA STANDARD	57
EXPRESSÕES, FUNÇÕES E LINGUAGEM SCRIPT (SCRIPT LANGUAGE)	67
EXEMPLOS PARA ACESSAR O APPLICATION DATABASE.....	68
CONFIGURANDO PLANILHAS (WORKSHEETS).....	80
GRUPO DE ALARMES (ALARMS GROUP)	83
CRIANDO A TELA DE ALARM ON-LINE.....	87
CRIANDO A TELA DE ALARMES HISTÓRICOS (HISTORICAL ALARM SCREEN)	91
EXEMPLO: CONFIGURAÇÃO UTILIZADA NO RUNTIME: TP->HIHILIMIT=70	93
TREND.....	94
CRIANDO UM GRÁFICO DE TENDÊNCIAS HISTÓRICO (TREND HISTÓRICO).....	97
CRIANDO UMA TELA TREND HISTÓRICO	99
CRIANDO RECEITAS (RECIPES).....	104
CRIANDO UMA TELA DE RECEITAS (RECIPE SCREEN)	109
CRIANDO RELATÓRIOS (REPORTS)	112
CRIANDO RELATÓRIOS DO TIPO RTF, COM CABEÇALHO E CORPO.....	115
CONFIGURANDO UMA PLANILHA SCHEDULER	118

CRIANDO EVENTOS NO MÓDULO SCHEDULER.....	119
CRIANDO UMA TELA DE RELATÓRIOS (REPORT SCREEN)	120
O INDUSOFT WEB STUDIO	123

Indusoft LTD.

InduSoft Web Studio

PROGRAMA DE TREINAMENTO BÁSICO



InduSoft Web Studio Para Windows NT e CE

Introdução

Você vai conhecer o **InduSoft Web Studio** uma ferramenta fácil de usar, flexível e poderosa para desenvolver suas aplicações em automação industrial. Durante o treinamento explore o conceito da arquitetura orientada a módulos funcionais, deixando detalhes de configuração para a etapa seguinte, quando estiver de fato desenvolvendo seu projeto. Neste módulo de Apresentação você deve entender como o produto funciona, pois isso é fundamental para a construção de uma aplicação que explora os recursos do produto e ao mesmo tempo preserva ótima performance.

Requisitos de sistema para rodar o InduSoft Web Studio

Hardware

O **InduSoft Web Studio** requer a seguinte configuração mínima:

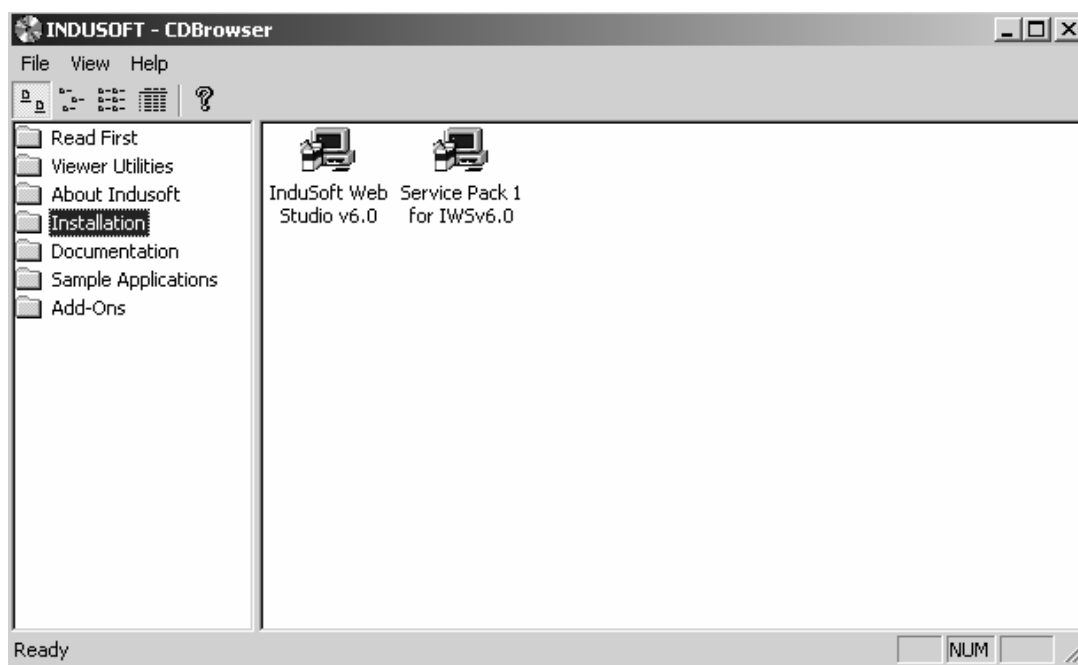
- Microcomputador compatível com Pentium III ou superior.
- Para desenvolvimento, Windows NT (com SP 6), 2000 (com SP 4), ou XP.
- Para runtime, Windows NT/2000/XP, Windows CE 3.00 ou Windows CE.NET.
- Mínimo de 256 MB ou mais, recomendável.
- Mínimo de 350 MB – somente para o **InduSoft Web Studio**. É necessário também ter espaço livre o suficiente para armazenamento do aplicativo desenvolvido e dados como históricos e alarmes conforme o dimensionamento em fase de projeto.
- Teclado padrão.
- Unidade CD-ROM (pode ser utilizado um computador da rede).
- Monitor padrão VGA ou SVGA, compatível com IBM/PC com mínimo de 16 Mb de memória.
- Dispositivo apontador como mouse.
- Portas seriais ou Placas Ethernet ou ainda Placas Especiais para a conexão com os equipamentos de campo e aquisição de dados do processo.

Software

- Windows 9X/ME/NT/2000, até versão 4.1 (Ansi) ou Windows NT/2000/XP para versão 4.4 ou superior (Unicode).
- Para conectividade em rede intranet ou internet, é necessário um cartão Ethernet configurado com o protocolos TCP/IP do Windows.

Instalando o InduSoft Web Studio

O programa de instalação do **InduSoft Web Studio** roda em Windows 32 bits, e se encontra na cópia do CD distribuída pela InduSoft. Quando o CD é inserido no drive de CD ROM, automaticamente é aberto o **CD Browser** que contém a pasta de instalação do InduSoft chamada de Installation, que pode conter o **Service Pack** do produto ou não.



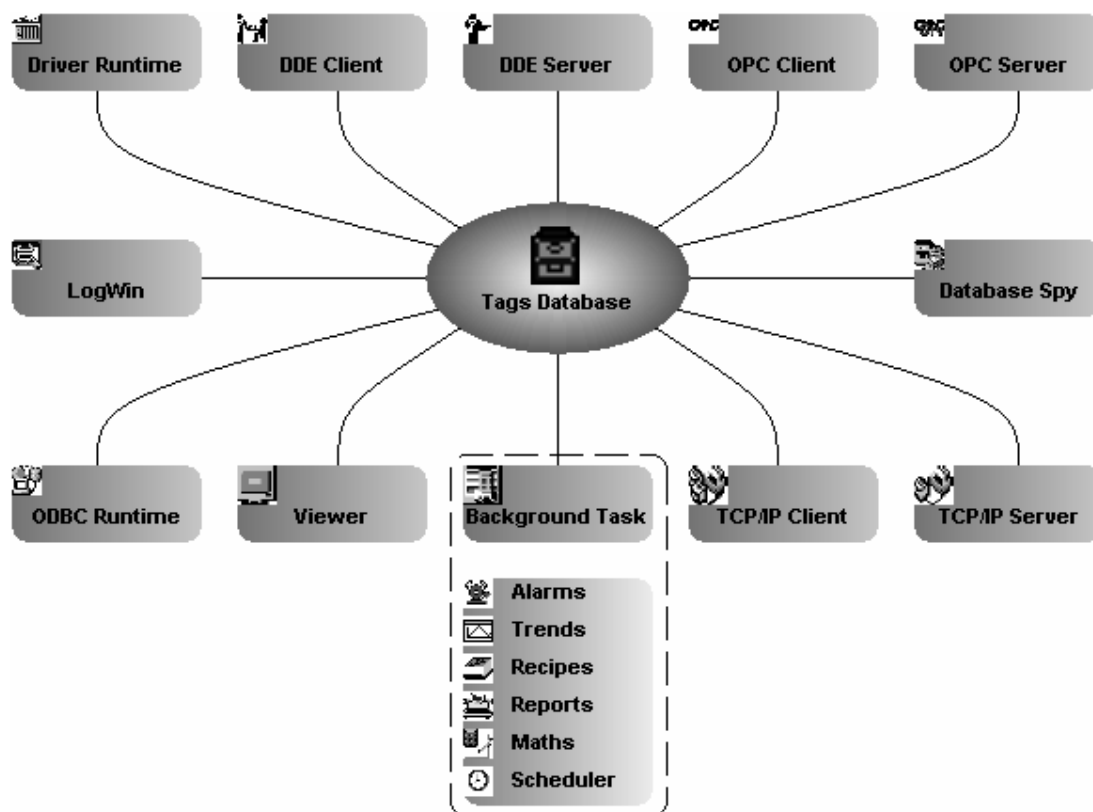
Service Pack é uma atualização da versão corrente que contém pequenas alterações. Qualquer Service Pack contém todas as atualizações dos Service Packs anteriores, de forma que você deve instalar apenas o último deles.

Após a instalação são criados os seguintes diretórios, supondo uma instalação padrão:

- ◆ InduSoft Web Studio\Bin – contém os arquivos executáveis, dlls, ini, etc...
- ◆ InduSoft Web Studio\Demo – diretório com uma aplicação de demonstração
- ◆ InduSoft Web Studio\DRV – contém os arquivos relacionados aos drivers de comunicação
- ◆ InduSoft Web Studio\Lib – telas da biblioteca de símbolos
- ◆ InduSoft Web Studio\Projects – diretório vazio que pode ser usado para conter aplicações construídas pelo usuário
- ◆ InduSoft Web Studio\Redist – Contém as várias versões do programa **CEView** e **CEServer** que são os programas de Runtime para Windows CE.

4. Estrutura Interna

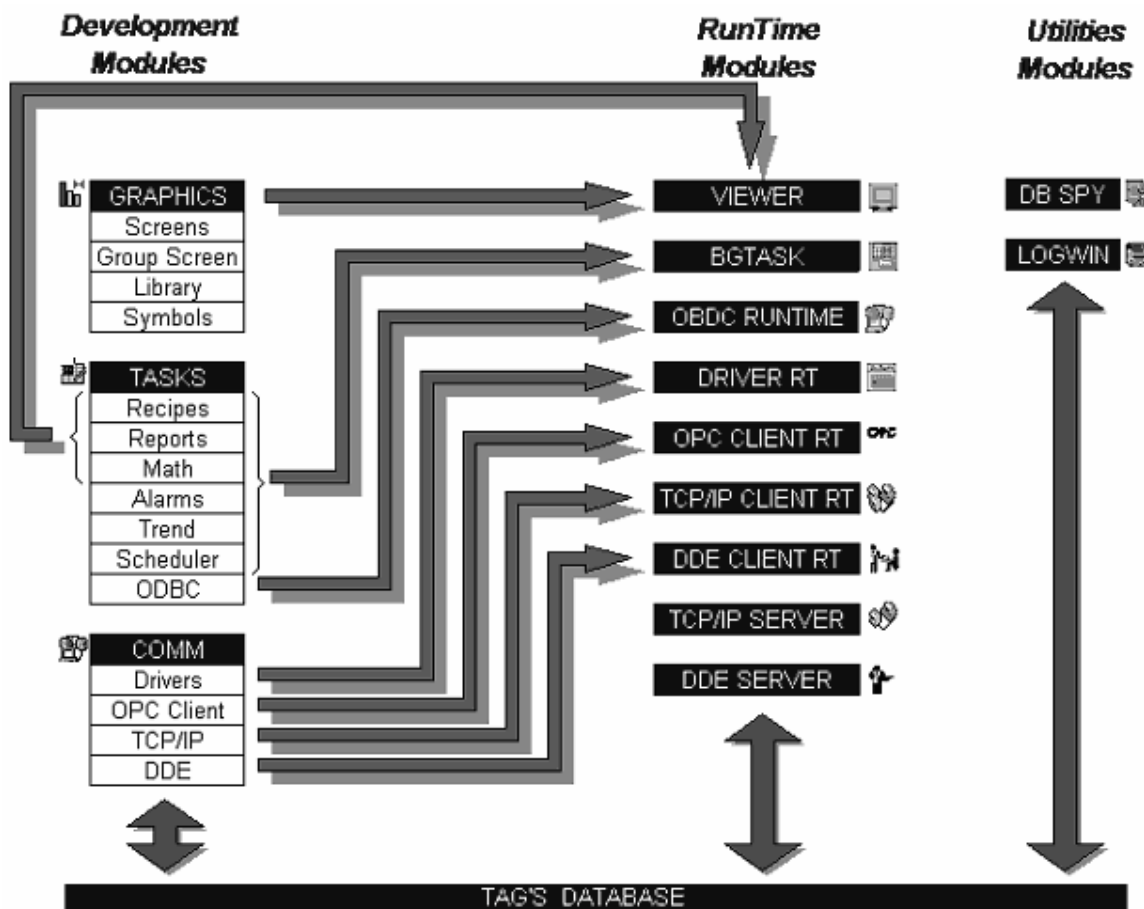
O Database desempenha um importante papel na estrutura de funcionamento do InduSoft Web Studio. É através dele que as informações trafegam. Para mandar dados do driver para uma planilha matemática, o valor do tag gerado pelo driver vai para o Database e só quando necessário é enviado para a planilha matemática. Assim dá-se o processamento por exceção. Os dados não trafegam desnecessariamente.



Todas as aplicações desenvolvidas no InduSoft Web Studio utilizam-se dos módulos de Engenharia para o desenvolvimento e os de Runtime para a execução. Durante a execução todos os módulos trocam dados apenas com o banco de dados (Database) e este com os módulos.

No esquema que segue, pode observar-se a correspondência entre os módulos de Engenharia – Development – e os módulos de Runtime. Há ainda os Utilitários, que embora não tenham sido incluídos no diagrama anterior, operam da mesma forma: trocam dados exclusivamente com o Database.

Note também que os módulos de Engenharia também se comunicam com o Database. Toda configuração em um módulo de engenharia envolve um tag, e este já deve estar criado no Database, ou então o Studio solicitará que você o crie, de modo que a configuração permanece consistente.



- **Background Tasks:** Esta tarefa é responsável pela execução das lógicas (scripts) contidos nas planilhas matemáticas, *scheduler* e gerencia os módulos configurados como nas planilhas de alarmes, Receitas, *Trends*, e relatórios.
- **Database Spy:** Ferramenta de depuração usada para: Ler dados do *TAG Database* (ex.: valores de *TAGs*); Escrever dados no *TAG Database* (ex.: valores de *TAGs*); Executar funções/expressões e verificar seu funcionamento.
- **DDE Client:** Gerencia a troca de mensagens DDE com qualquer *DDE Server* local ou remoto, de acordo com as configurações das planilhas DDE.
- **DDE Server:** Gerencia a comunicação DDE com qualquer *DDE Server* local ou remoto.
- **Driver Runtime:** Gerencia a leitura e escrita de comandos e dados configurados nas planilhas de Driver.
- **LogWin:** Ferramenta de depuração onde é possível verificar mensagens geradas por outras tarefas.
- **ODBC Runtime:** Gerencia a comunicação ODBC com bancos de dados relacionais (SQL), de acordo com as configurações contidas na planilha ODBC.

- **OPC Client Runtime:** Gerencia a comunicação OPC com *OPC Server* local ou remoto, obedecendo aos parâmetros presentes na planilha OPC.
- **OPC Server:** Administra a comunicação OPC com *OPC Client* local ou remoto.
- **TCP/IP Client:** Administra o tráfego de mensagens TCP/IP provenientes de um modulo *TCP/IP Server* remoto (de outro InduSoft Web Studio), respeitando as configurações das planilhas TCP/IP
- **TCP/IP Server:** Gerencia a comunicação de mensagens TCP/IP com módulos TCP/IP Client remotos (provenientes de outro InduSoft Web Studio ou um Web Thin Client).
- **Viewer:** Executa as lógicas de confiuradas em tela (*On Open, On While, On Close, Command, Hyperlink*, etc) e atualiza os objetos em telas.

Criação:

- ◆ Criação de Telas e Biblioteca de Símbolos
- ◆ Criação de Planilhas de tarefas (Scheduler, Alarmes, Matemáticas, Trend, Receitas, Relatórios, ODBC, etc...).
- ◆ Criação de comunicações (Drivers, TCP/IP, DDE, OPC).
- ◆ Criação do Banco de Dados e Sistema de Segurança.

Execução:

- ◆ Visualização de Telas (Módulo Viewer)
- ◆ Execução de Tarefas (Módulo BackGround Tasks executando: Receitas, Relatórios, Planilhas de Alarmes, Matemáticas, ODBC Runtime, etc...).
- ◆ Execução da Comunicação (Módulo de Driver, Módulos TCP/IP Client e Server, Módulo OPC Cliente Runtime, Módulos DDE Client e Server).
- ◆ LogOn e LogOff

Depuração ou Debug:

- ◆ Database Spy – forçar e ler valores do Banco de Dados
- ◆ LogWin - Controle da execução dos módulos, da comunicação serial, variação do valor dos tags, etc...

O objetivo deste treinamento é torna-lo familiar a estas categorias de módulos, para que saiba utiliza-la da melhor forma durante a execução, partida ou manutenção de um sistema desenvolvido com a plataforma **InduSoft Web Studio**.

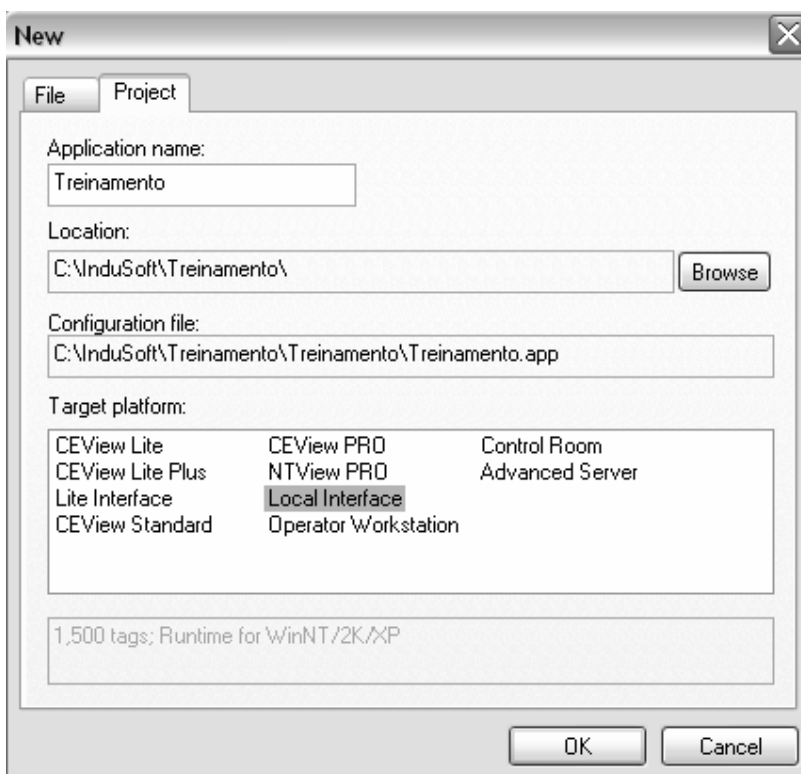
Criando um novo projeto

- Para criar um novo projeto usando o InduSoft Web Studio, rode o **InduSoft Web Studio** clicando no atalho na área de trabalho para este programa.
- .

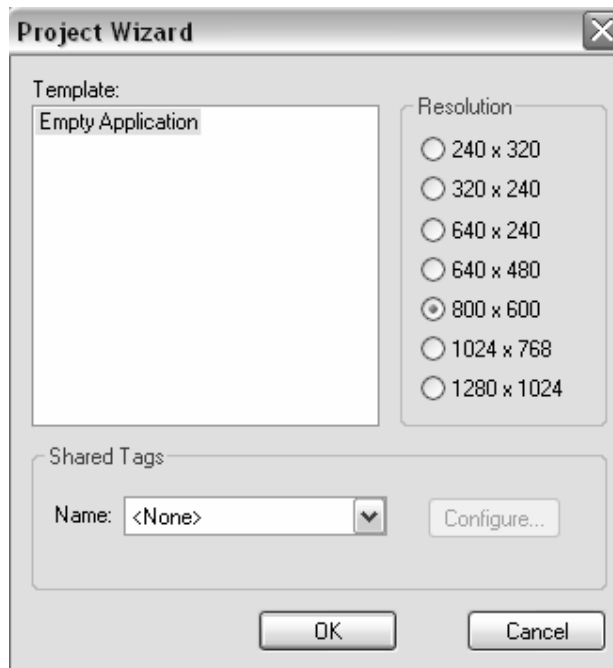


ou usando o botão **Iniciar-> Programas-> InduSoft Web Studio v6.0 -> InduSoft Web Studio v6.0** ou **Start -> Programs-> InduSoft Web Studio v6.0 -> InduSoft Web Studio v6.0** (no caso do Windows em Inglês).

- Já no ambiente do InduSoft Web Studio, selecione **File** e clique em **New** para abrir a janela New.
- Selecione o tab "**Project**" e digite o nome da aplicação no campo "**Application name**".
- Selecione a plataforma para essa aplicação na lista "**Target Platform**", levando em conta a dimensão do projeto a ser criado.



- O botão “**Browse**” proporciona a possibilidade da escolha de um novo caminho para a criação da aplicação.
- Pressione o botão **OK** para abrir a janela seguinte
- Na janela “**Project Wizard**”, é possível selecionar algumas aplicações que podem ser utilizadas como um ponto de partida para a aplicação. Para o treinamento não usaremos essas aplicações, iniciando um projeto.
- É possível também escolher a resolução apropriada para as telas, em pixels.



IMPORTANTE:

- Selecione: **Empty Application** (Aplicação Vazia) e **Resolution = 800 X 600**.



Nota:

A resolução escolhida define somente o tamanho default das novas telas que serão criadas na aplicação. Cada tela inserida pode ter uma resolução totalmente customizável.

- Clique **OK**.

Configurando o Project Settings

- No menu principal, selecione **Project** clique em **Settings...** para abrir a janela "Project Settings." Esta janela possui quatro Tabs:
- O tab "**Identification**" é reservado para identificação e registro dos dados do projeto.

The screenshot shows the 'Project Settings' dialog box with the 'Identification' tab selected. The dialog has a title bar with a close button. Below the title bar are six tabs: 'Identification', 'Options', 'Runtime Desktop', 'Communication', 'Web', and 'Preferences'. The 'Identification' tab is active, displaying the following fields:

- Application: C:\Driver D\InduSoft\Treinamento\TreinamentoSP3\Treinamento.APP
- Description: Programa de Treinamento Básico
- Revision: 1.0
- Company: InduSoft do Brasil Ltda
- Author: InduSoft do Brasil Ltda
- Field Equipment: CLP

Below these fields is a 'Notes' section with a text area containing the text: 'Esse tutorial é de uma aplicação do Treinamento Básico da InduSoft do Brasil Ltda'. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

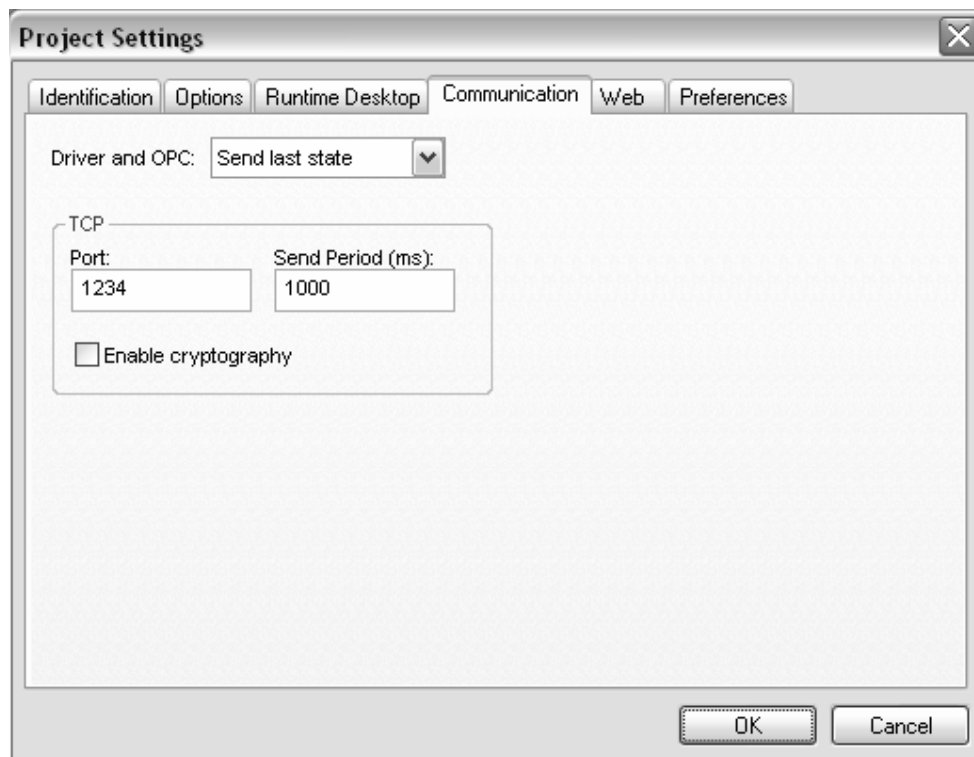
- O Tab "**Options**" permite definir algumas características do ambiente da **APLICAÇÃO FINAL** durante a execução. Tais como: a plataforma a ser utilizada, a porta de comunicação TCP/IP, entre outros.

The screenshot shows the 'Project Settings' dialog box with the 'Identification' tab selected. The 'Target system' section has a dropdown menu set to 'Local Interface' and a text field for 'Resolution' set to '800x600'. Below this is a text field containing '1,500 tags; Runtime for WinNT/2K/XP'. The 'Automatic Translation' section has a text field for 'Translation File name' set to 'C:\GUARANINGUARANI.TRA' and an unchecked checkbox for 'Enable Translation'. The 'Alarm History and Events' section has a text field for 'History Life Time' set to '0', a dropdown for 'History Format' set to 'Proprietary', and two buttons: 'Event Database...' and 'Alarm Database...'. The 'Default Database' section has a 'Configure...' button. The 'Shared Tags' section has a text field for 'Name' set to '<None>' and a 'Configure...' button. At the bottom are 'OK' and 'Cancel' buttons.

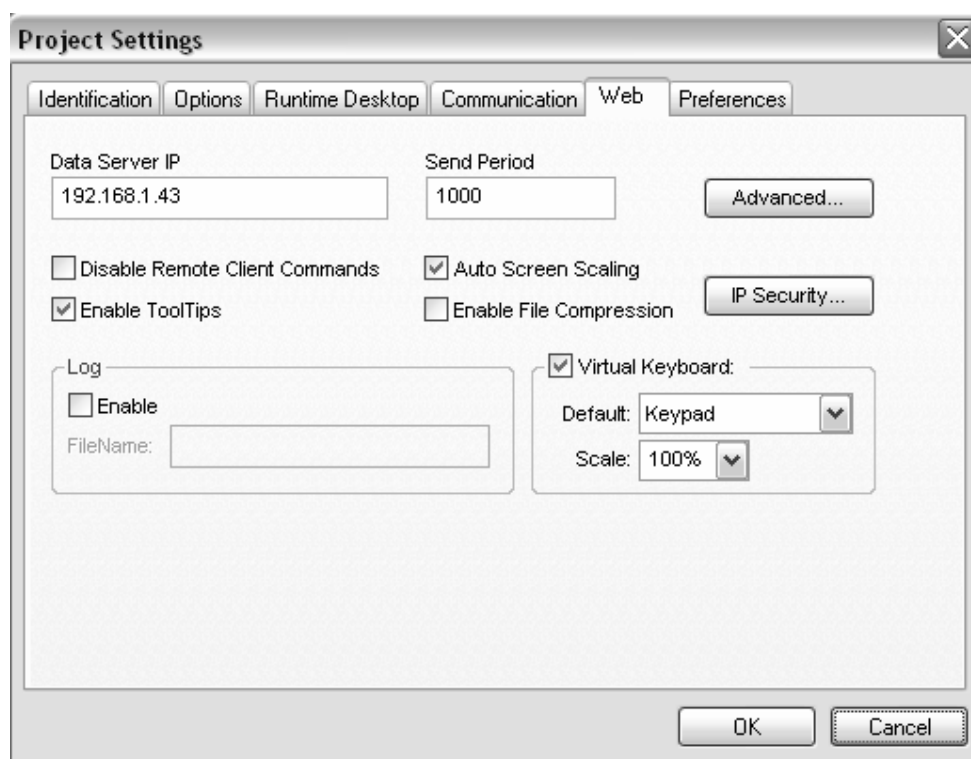
- O Tab "**RunTime Desktop**" permite definir algumas características do ambiente da *APLICAÇÃO FINAL* durante a execução. Neste Tab é importante definir uma tela da aplicação para ser a **startup screen** ou um grupo destas (*.sg). A janela escolhida é aberta automaticamente quando o módulo **Viewer** é iniciado.

The screenshot shows the 'Project Settings' dialog box with the 'Runtime Desktop' tab selected. The 'Titlebar' section has a checked checkbox and a text field set to 'Programa de Treinamento'. The 'Minimize Box', 'Maximize Box', and 'Close Box' are unchecked. 'Start Maximized' and 'Menu' are checked, with an 'Options...' button next to 'Menu'. 'Resize Border' and 'Status Line' are unchecked. The 'Startup screen' text field is set to 'principal.scr'. The 'Show ????' when quality is not GOOD' checkbox is checked, and 'Hide Taskbar' is unchecked. The 'Active area indication' section has unchecked checkboxes for 'Show Object Edge' and 'Change Mouse Cursor'. The 'Mouse Cursor' checkbox is checked. The 'Virtual Keyboard' section has a checked checkbox, a 'Default' dropdown set to 'Keypad', and a 'Scale' dropdown set to '100%'. The 'Enable ToolTips' and 'Auto Screen Scaling' checkboxes are checked. At the bottom are 'OK' and 'Cancel' buttons.

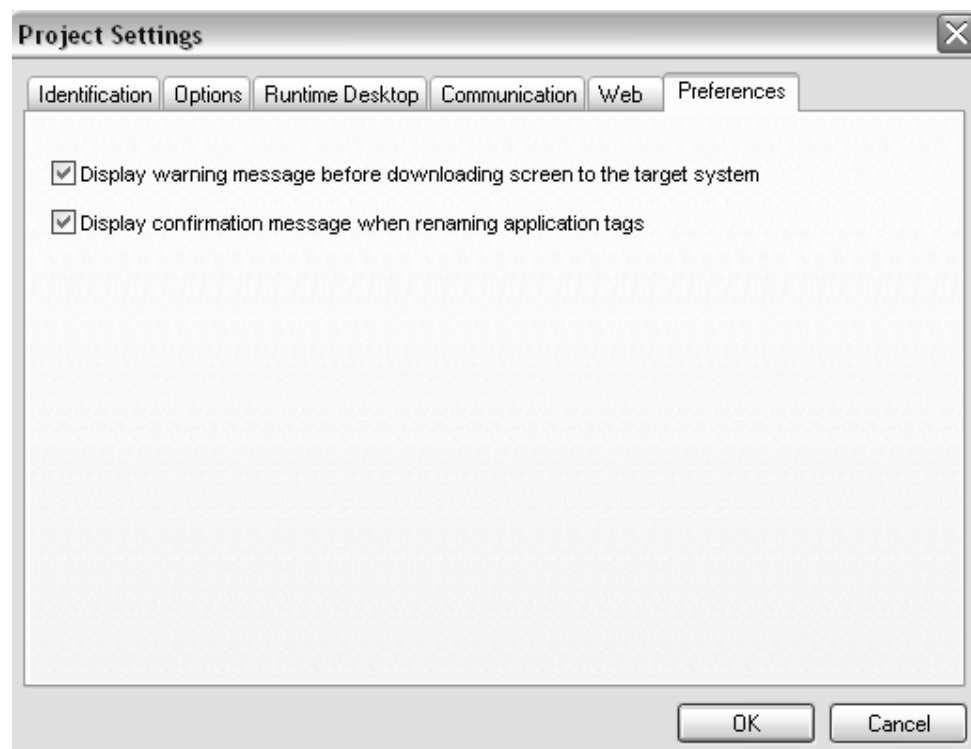
- O Tab "**Communication**" permite definir algumas características do InduSoft Web Studio para trabalhar como Servidor de mensagens criptografadas ou não e ainda determinar a porta de comunicação TCP e o tempo de envio de mensagens.



- O Tab "**Web**" permite definir algumas características do InduSoft Web Studio para trabalhar como Servidor Web da aplicação final. Este Tab será melhor explicado, mais à frente, quando estudarmos configurações Web.

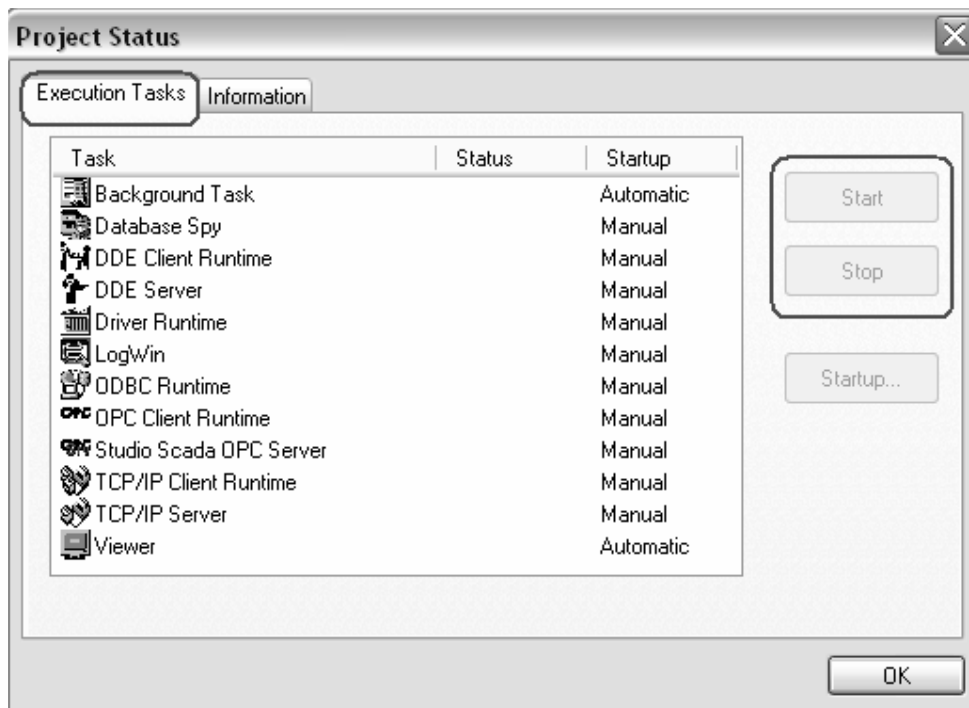


- O Tab "**Preferences**" permite definir como o próprio nome já expressa, preferências sobre a apresentação ou não de algumas mensagens de alerta durante determinadas ações realizadas pelo desenvolvedor, no ambiente do InduSoft Web Studio.

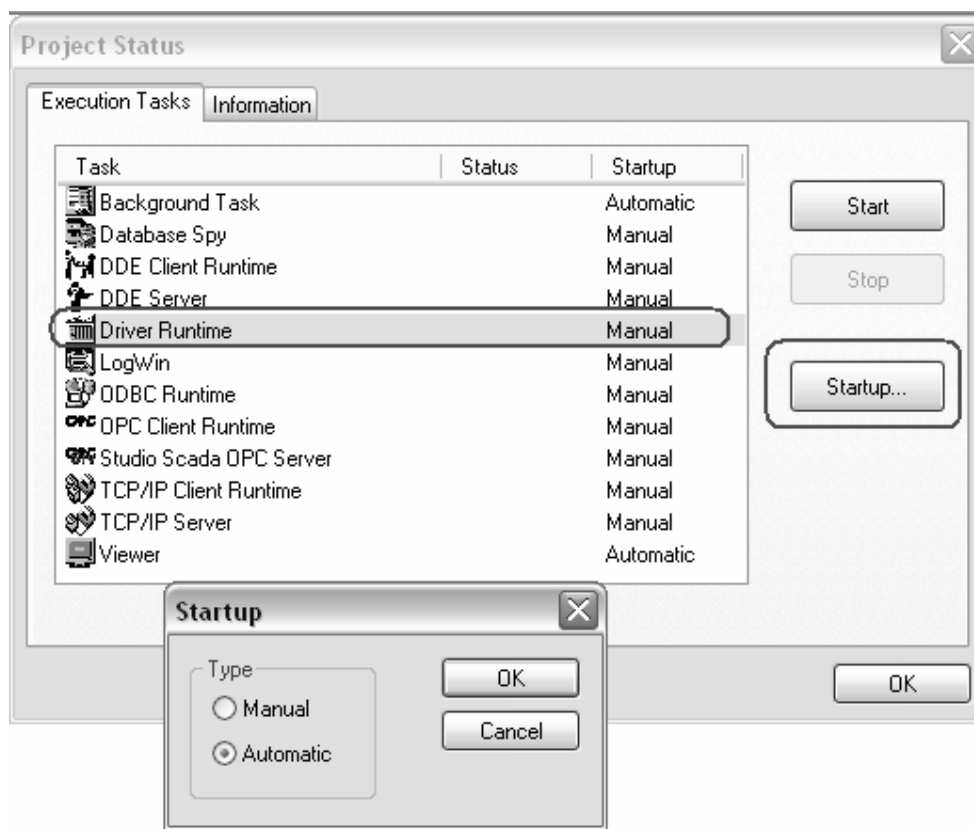


Configurando Project Status

- No menu principal selecione **Project** clique em **Status...** para abrir a janela **Project Status**. Dois tabs compõem a janela Project Status.
- O Tab "**Execution Tasks**" permite controlar a execução de cada módulo de Runtime, iniciando ou finalizando estes módulos, usando os botões "**Start**" e "**Stop**".



- Para configurar os módulos de Runtime que necessitam ser iniciados automaticamente no **STARTUP DA APLICAÇÃO**, clique no módulo em seguida no botão **Startup** selecionando a opção "**Automatic**".



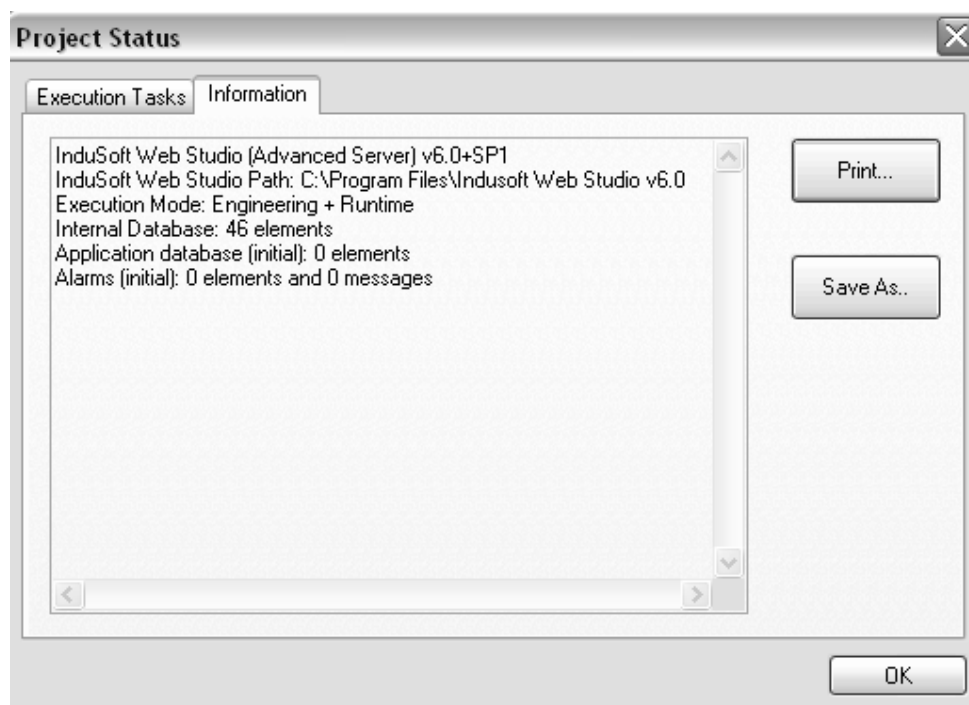
- Se estiver em dúvida sobre quais módulos devem ser configurados para start automático, estude novamente o quadro sinótico do capítulo Apresentação, que mostra quais funcionalidades são executadas por quais módulos de runtime.



Nota:

Quando o sistema operacional para o qual está sendo desenvolvida a aplicação é o Windows CE o tab “**Execution Tasks**” não estará acessível. No CEView, todos os módulos do Runtime (BGTask, Viewer, TCP/IP Client / Server e Driver Runtime) são iniciados simultaneamente quando a aplicação é iniciada. Portanto também não é possível fechá-los individualmente.

- O Tab “**Information**” mostra algumas informações gerais sobre a aplicação. Tais como:
 - Versão do produto;
 - Local onde a aplicação está sendo criada;
 - Tipo de Licença;
 - Número de Tags Internos da Aplicação;
 - Número de Tags Criados Durante a Aplicação;
 - Número de Mensagens de Alarmes Criadas na Aplicação.



- Pressione **OK**.

Configuração do Tag Database

Definição de Tags

Tags são as variáveis utilizadas na aplicação. São utilizados em todas as configurações dos módulos de supervisão. Tags podem ser de comunicação com o equipamento de campo ou utilizados internamente na aplicação, para cálculos, para gerar bases de tempo ou eventos.

No InduSoft Web Studio, os tags criados pelo usuário são chamados *application tags* ou *tags de aplicação*; O InduSoft possui alguns tags pré-definidos que são chamados *internal tags* ou *tags internos*. Esses tags internos possuem informações do sistema como data e hora corrente, usuário logado, entre outros, ou ainda podem ser usados como comando de alguma função como o beepoff, ackalr e outros.

Regras de sintaxe dos Tags

O nome de um Tag tem as seguintes restrições:

- ◆ Só devem ser compostos de letras, números e do caractere “_” (underscore).
- ◆ Não podem ser utilizados os seguintes caracteres especiais, nem acentos:
` ~ ! @ # \$ % ^ & * () - = \ + \ [] { } < > ? / ´ ¨ . ¨ : , ; ç
- ◆ Devem sempre iniciar com uma letra.
- ◆ Sua largura máxima é de 32 caracteres para nome de tags e 16 para tags membros de uma classe
- ◆ O nome de um tag deve ser diferente do nome de um internal tag e de funções matemáticas.

Exemplo:

Temperatura

Pressao

Nivel

InduSoft Web Studio não diferencia letras maiúsculas de minúsculas, portanto NIVEL = Nivel = nivel, sendo assim a interpretação deles é da mesma forma.

Contudo o sistema mantém conforme foi digitado. Adote um padrão que lhe convém para facilitar a leitura a partir da visualização, como em NivelTanque, PressaoTanque, que torna a leitura mais fácil do que em niveltanque, pressaotanque.

Tipos de Tags

O tipo de um tag pode ser de um dos seguintes tipos padrões:

- **Boolean** – (1 bit. Valor “0” ou “1”)

Boolean (digital):

Variável digital ou Booleana (0 ou 1).

- ◆ **Integer** – 4 bytes (Número inteiro)

Integer (analógica):

Números inteiros (Podem ser positivos, negativos ou zero).

Equivalente ao tipo long integer. (De -2147483647 a 2147483648).

Exemplos:

0

50

-200

♦ **Real** – 8 bytes (Número real com ponto flutuante)

Real (ponto flutuante):

Números reais internamente gravados como uma double word.

Exemplos:

25,40

0.00009

-2960

3,141516

♦ **String** – 255 Caracteres alfanuméricos

String (ASCII):

String de caracteres com o comprimento de até 255 caracteres (de 0 a 254) e suporta letras, números, ou caracteres especiais.

Exemplos:

Receita do produto X123

01/01/90

***** On *****

Todos os tags são declarados no Database, na pasta Application Tags.

Além dos quatro tipos previamente listados, você pode definir novos tipos chamados *Classes*.

Classes, Arrays e Apontadores (ou tags indiretos)

Array tags

Os tags do InduSoft Web Studio podem armazenar um valor de um tag ou de um vetor de tags. Um tag com a opção “*array*” é um grupo de tags com o mesmo nome, mas com índices diferentes (uma matriz de *n* linhas e 1 coluna).

Exemplo:

Temperatura[1]

Temperatura[2]

Temperatura[3]
Temperatura[500]

Use um tag array sempre que possível, pois otimiza a memória e simplifica a configuração das tarefas. Suponha que você quer ter um display para monitorar cada tanque. Usando tags array torna possível configurar somente um display que contém tags linkados a qualquer um dos Tanques:

Pressao[tk]
Temperatura[tk]
Temperatura[tk +1]

O tag **tk** é o tag que contém o número do tanque que se deseja visualizar. Um índice para array deve ser um tag, um valor numérico, ou uma expressão incluindo uma adição.

Para se referir a um array usando a operação adição, você deve seguir a seguinte sintaxe: tag_name[tag + N], onde N é uma constante numérica.

Exemplos:

Temperatura[tk+2]
Temperatura[tk+6]

O uso de tags com a opção de *array* economiza tempo no desenvolvimento da aplicação. Suponha que você necessita da temperatura de quatro Tanques.

O método convencional seria:

Tag Name	Description
Temperatura1	Temperatura no Tanque 1
Temperatura2	Temperatura no Tanque 2
Temperatura3	Temperatura no Tanque 3
Temperatura4	Temperatura no Tanque 4

Usando um tag com a opção de array, é possível simplificar essa tarefa:

Temperatura[x] Temperatura no Tanque X

Onde X representa 1,2,3 ou 4.

Quando você cria um tag array de quatro posições, o sistema cria cinco posições (de 0 a 4).

Exemplo:

tag_Exemplo[15] //posição inicial=0 ///posição final=15

Dessa forma, o array tag_Exemplo[15] possui 16 elementos.



IMPORTANTE:

Quando o programa tenta usar um índice inválido, como por exemplo, a posição 20 de um array com 15 posições, a posição 0 é usada para armazenar esse valor em rascunho. Devido a isso NÃO é aconselhável utilizar a posição 0.

Tags indiretos – apontadores

InduSoft Web Studio suporta acesso indireto aos tags do database.

Considere o tag **X** do tipo **string**. Este tag pode conter o nome de outros tags do database, isto é, ele pode funcionar como um ponteiro para qualquer outro tipo de tag. (Incluindo um tag tipo classe).

A sintaxe para um tag indireto é:

@<nome_do_tag_indireto>.

Por exemplo, assumindo que um tag string chamado **X** contem o valor *TEMP*. Ler ou escrever em **@X** permite acessar o valor do tag *TEMP*. Para acessar um tag tipo **Class**.

Para acessar um tag indireto de uma classe, use a seguinte sintaxe **@<tag_indireto>.<membro>**

Exemplo:

@X.Nivel

Neste caso você está acessando o membro *Nivel* do tag para o qual **X** aponta.

Apesar de que qualquer tag string pode ser usado como um ponteiro, quando você criar tags para esse propósito, coloque o @ antes do nome do tag e escolha o tipo de dado que este ponteiro carrega, pois isso permite que o InduSoft faça a consistência para o tipo boolean, integer, real.

Veja o exemplo seguinte:

TagName	Type
@Z	Integer
@X	Class: TANQUE

Qualquer tag que é criado como **string** é um apontador em potencial.

Classes

Em adição aos tipos padrão de tags você também pode definir tags do tipo **Classe**.

Class: Classes são estruturas que permitem um alto grau de organização do application database. Quando um tag do tipo **class** é criado, este não irá conter um simples valor, mas um grupo de valores. Criar tags tipo **class** equivale a agrupar tags simples, que passam a ser *elementos*.

Exemplo:

Class	Member	Type	Description
Tanque	NIVEL	Integer	(Nivel)
	TEMPERATURA	Integer	(Temperatura)
	PRESSAO	Real	(Pressão)
	VALVULAENCHE	Boolean	(EstadoValvula)

Se você criar um novo tag **tk** do tipo **Class Tanque**, você está na verdade criando o tag com as propriedades **Class: Tanque**.

Para acessar os membros de um tag tipo **Class** use o ponto (.) como separador.

Exemplo:

tk.NIVEL
tk.TEMPERATURA

Se o tag **tk** é um array, a sintaxe será:

tk[1].NIVEL
tk[n].TEMPERATURA

Fields – parâmetros do tag

Fields são os parâmetros relativos a cada tag no database.

Estes parâmetros são definidos utilizando o botão Tag Properties localizado na barra superior do InduSoft Web Studio.

Para acessá-los, use a seguinte sintaxe:

NomeDoTag->field

Você pode utilizar os seguintes **fields** no Runtime:

Min valor mínimo para unidade de engenharia.

Max valor máximo para unidade de engenharia.



Nota:

Se a aplicação tentar escrever um valor e este valor estiver fora do range especificado, o database não aceita e uma mensagem de aviso é enviada ao Logwin.

Unit Este field aceita qualquer string (até 8 caracteres) relativo ao tag como uma breve descrição ou referência.

***HiHiLimit** Quando se criam alarmes tipo HighHigh, este field contém o limite.

***HiLimit** Quando se criam alarmes tipo High, este field contém o limite.

***LoLimit** Quando se criam alarmes tipo Low, este field contém o limite.

***LoLoLimit** Quando se criam alarmes tipo LowLow, este field contém o limite.

***RateLimit** Limite da taxa de variação do valor de um tag que gera um alarme.

***DevSetpoint** Refere-se ao setpoint do alarme do tipo Dev para um valor de tag que gera um alarme.

Exemplo: Configuração usada no Runtime:

TP->HiLimit =70

Description: descrição do tag digitada no DataBase.

***AlrDisable** desabilita o alarme:

1 Desabilita alarme

0 Habilita alarme

Exemplo: Configuração usada no Runtime:

TMP->AlrDisable=1

Size tamanho de um array.

HiHi Se maior que zero, um alarme Highhigh está presente.

Hi Se maior que zero, um alarme high está presente.

Lo Se maior que zero, um alarme Low está presente.

LoLo Se maior que zero, um alarme LowLow está presente.

Rate Se maior que zero, um alarme tipo rate está presente.

Dev Se maior que zero, um alarme Dev está presente.

TimeStamp Grava a data e hora da última mudança do valor do Tag.

b0..b31 Permite acesso individual aos bits de um tag inteiro.

Exemplos:

Nivel->Max
Temperatura->Unit
pv101->HiLimit



Nota:

Durante o runtime, é possível modificar valores dos fields relativos aos limites de alarmes, bem como desabilitá-los, utilizando-se dos fields apontados nessa descrição por “*”.

Internal Tags and Shared Database

Internal tags

São tags pré-existent em todas as aplicações, que possuem funções e valores específicos e podem ser utilizados em qualquer parte da aplicação, mas não podem ser apagados, modificados e nem terem seu valor alterado no banco de dados.

Exemplo:

Date: Mostra uma string com a data corrente do computador.

Time: Mostra uma string com a hora corrente do computador.

A maioria dos tags internos são read-only, ou seja, somente leitura.

Não se pode escrever no tag **Time** para se alterar a data do computador. Isto é feito com a função: **SetSystemTime(strTime)**.

Application tags

São todos os tags criados pelo usuário.

Shared tags

Tags adquiridos pelo compartilhamento do database de um Soft-PLC. Podem ser utilizados em toda a aplicação, mas não podem ser apagados.



Nota:

Atualmente este tipo de compartilhamento caiu em desuso. No entanto o InduSoft Web Studio tem a possibilidade de fazer esse tipo de comunicação com os seguintes produtos: First ISaGRAF, ISaGRAF, Think & Do SteepleChase, SixNet e OpenControl.

Trabalhando com o Database

- No Workspace, selecione o tab **Database**. Clique duas vezes na pasta **Application Tags** para expandi-la.
- Clique duas vezes em **Datasheet View**.



O Database será preenchido e alterado durante o desenvolvimento da aplicação. Porém, para iniciarmos uma aplicação de exemplo, podemos já previamente definir um tag que utilizaremos no módulo de comunicação.

Configure a janela **Application Tags** como mostrado a seguir:

Desta forma você estará criando o tag **CONTADOR**, que servirá para a exemplificação da comunicação do InduSoft Web Studio com o PLC.

Application Tags					
	Name	Array Size	Type	Description	Web Data
1	Contador	0	Integer	Tag Contador que será lido pelo PLC	Server
2					
3					
4					
5					
6					

- Feche a janela Application Tags, não é necessário salvar.

Comunicação

Drivers de Comunicação

Os drivers são os responsáveis pela transferência de dados entre os dispositivos de campo (CLP, remotas, controladores, unidades concentradoras, etc.) e o InduSoft Web Studio.

Temos em torno de 150 drivers disponíveis nas versões para Win9x, NT e Windows CE.

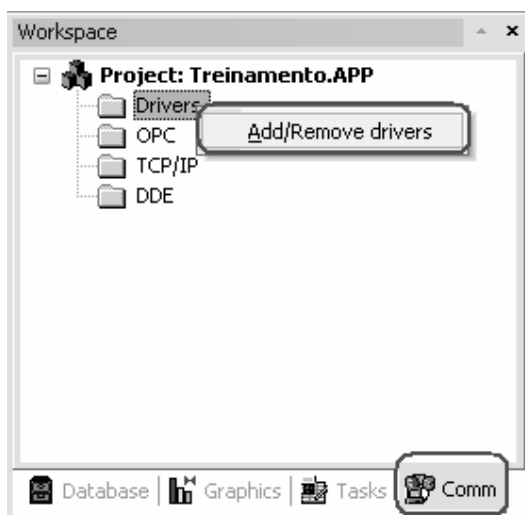
Todos eles ficam disponíveis para download em nosso site, www.indusoft.com.br/download/drivers, sem qualquer restrição.

Seguem alguns dos mais utilizados:

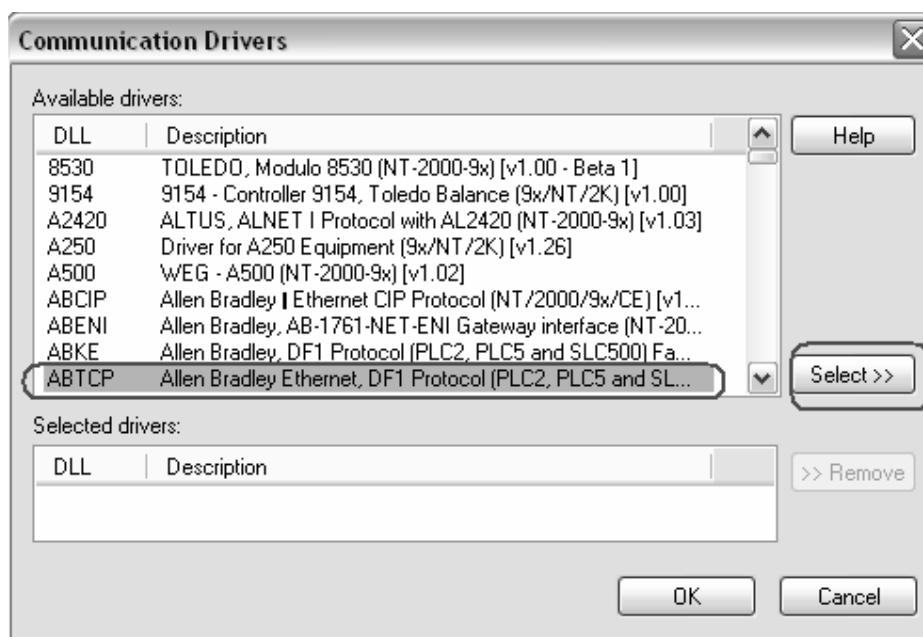
Allen Bradley – DF1 e TCP/IP
Siemens – S5 – AS511 PG Port
Profibus DP Master e Slave (Hilscher)
Allen Bradley - ControlNet Slave
OMRON – Host Link
GE FANUC – SNP, 90-30 90-70 Series
Modbus – Schneider 984 series
Profibus DP Master
Cultler Hammer – D50 – D300 Series
Hitachi – H series
Toshiba – Prosec T1/T2

Selecionando um Driver

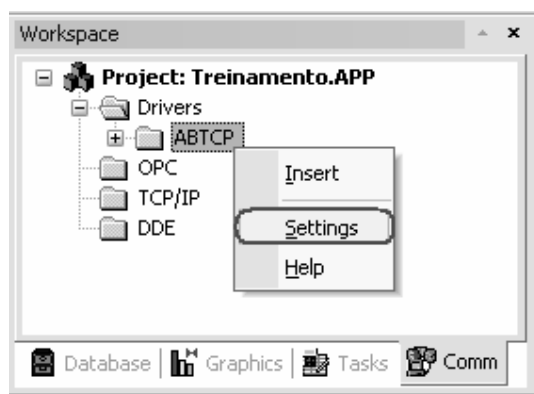
- Quando o InduSoft é instalado, todos os drivers são instalados juntos. Vá a pasta **Drivers** do tab **Comm**, e clique com o botão direito nesta pasta e escolha a opção **Add/Remove Drivers**.



- Escolha para o tutorial o **ABTCP**. Após selecionar o driver (**ABTCP**), clique no botão **Select >>**, clique em **OK**.



- Agora clique com o botão direito do mouse sobre a pasta **ABTCP**, aparecerão as opções **Insert**, que insere uma nova planilha, e **Settings** que configura alguns parâmetros da comunicação.
- Clique na opção **Settings**, que abrirá janela **Communication Parameters**.



Configurando os Parâmetros de Comunicação

Na janela **Communication Parameters**, são configurados os parâmetros da comunicação, sendo ela SERIAL ou como no nosso caso a comunicação TCP/IP.

Campos de Configuração da Janela Communication Parameters

Parâmetro	Valor Default	Valores Válidos	Descrição
COM	COM1	COM1 à COM8	Porta Serial do PC usada para comunicação com o equipamento (se for um driver serial).
Baud Rate	19200	110 à 57600bps	Velocidade de Comunicação
Data Bits	8	5 a 8	Número de data bits usados no protocolo
Stop Bits	1	1or 2	Número do stop bits usado no protocolo
Parity	Odd	even odd none space mark	Paridade do Protocolo
Station	0	0	Número, Nome do Computador, ou Unidade de Rede se o protocolo requerer isto.



Nota:

Os campos da seção Configuration Parameters variam de acordo com o driver utilizado, podendo ser diferentes ou estar desativadas em outros drivers.

- Os quatro campos da parte de baixo desta janela são configurados de forma diferente para cada driver e são configurados com parâmetros diferentes para cada driver. Para o driver ModBus, os campos são: **Signed Value**, **FP swap (0=Byte/1=Word)**, **Protocol (ASCII or RTU)**, e **Custom Command (ERO-xxx)**.

Se você entrar com um tipo de valor inválido em um desses campos, o InduSoft Web Studio pode aceitar o valor, mas quando você for fechar a janela *Communication Parameters*, uma mensagem de erro será apresentada e lhe informará o erro antes de ser fechada.

Especificando o Advanced Settings

Clicando no botão Advanced na janela *Communication Parameters* são apresentadas as possibilidades possíveis para configurar outros parâmetros para a comunicação serial (verifique as possibilidades na tabela abaixo):

Janela Advanced Settings

Parâmetros	Valores Default	Valores Válidos	Descrição
Start message (ms)	1000	0 a 10000	Tempo de resposta do dispositivo (tempo de time-out)
End message (ms)	0	0 a 10000	Tempo para o recebimento do fim da mensagem do dispositivo (Nota: entrando com valor 0 –zero – o driver desconsiderará a verificação destes dados)

Parâmetros	Valores Default	Valores Válidos	Descrição
Interval between char	500	0 a 10000	Tempo Máximo entre caracteres enviados para o equipamento
Wait CTS (ms)	100	0 a 10000	Tempo para recebimento sinal CTS (Clear to Send), após o sinal RTS (Request to Send) signal (Nota: Valido apenas se selecionado Yes no campo Verify CTS).
Control RTS	No	<ul style="list-style-type: none"> no yes yes + echo 	Define o sinal de handshake RTS (Request to Send) antes da comunicação e se ele terá eco nesta comunicação.
Verify CTS	No	<ul style="list-style-type: none"> no yes 	Define se o driver irá aguardar o sinal de handshake <i>antes de enviar a mensagem</i> .
Disable DTR	Not checked	<ul style="list-style-type: none"> Not checked Checked 	Se habilitado, o driver não seta o sinal DTR antes de iniciar a comunicação.
Retries	0	0 a 5	Numero de tentativas de reconexão em caso de time-out.
Tx Buffer (bytes)	512	0 a 512	Tamanho do buffer com o máximo de informação por mensagem de envio.
Rx Buffer (bytes)	512	0 a 512	Tamanho do buffer com o máximo de informação por mensagem de recebimento.

Parâmetros de Comunicação

Note que os primeiros parâmetros desta janela estão desabilitados. Estes se referem a uma comunicação SERIAL. Como essa janela é padrão para todos os drivers de comunicação, eles aparecem, neste caso, porém desabilitados.

LONG1, LONG2, STRING1 E STRING2 FIELDS

Dependendo do driver, são necessários mais alguns parâmetros.

No caso do Modbus, por exemplo, é necessário saber se estamos utilizando o modo RTU ou ASCII.

No Allen Bradley, se é um CLP da família 500, 5, 2 ou 3.

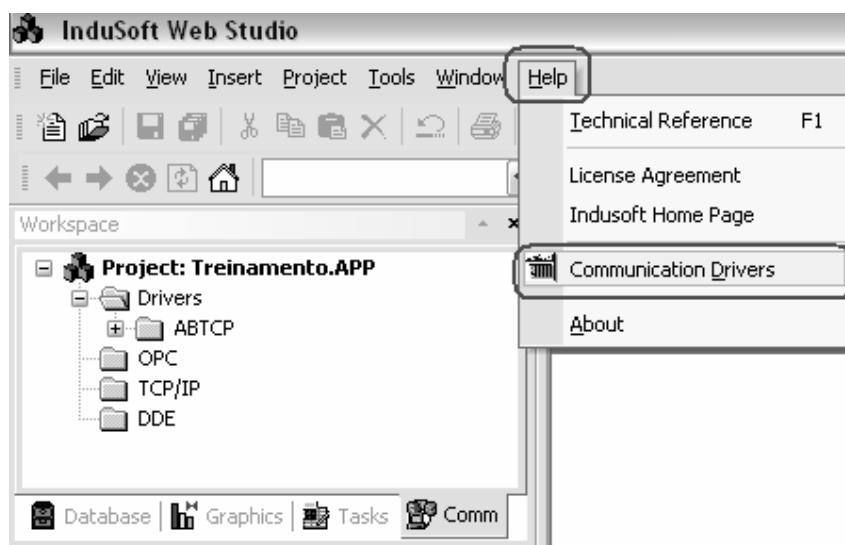
Para tanto esses campos estarão habilitados.

Há ainda um campo para Habilitar ou Desabilitar o Trace de mensagens específicas.

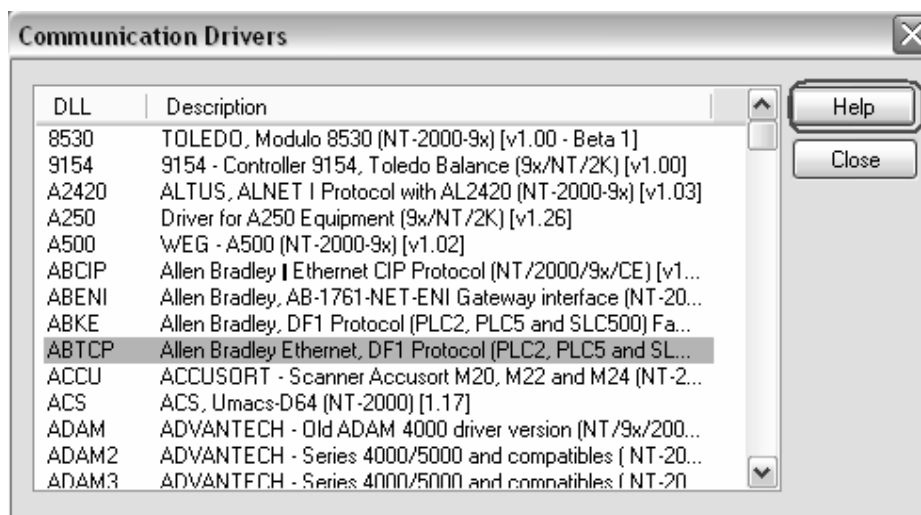
Manual do Driver de Comunicação

Cada driver de comunicação possui sua documentação específica.

Na pasta Drivers você encontrará um arquivo em formato PDF do mesmo nome do driver. Para visualiza-lo basta ter o Adobe Acrobat Reader instalado – uma versão é distribuída no CD de instalação do InduSoft Web Studio – e pode-se acessar do próprio ambiente do InduSoft Web Studio, como mostra a figura que segue:

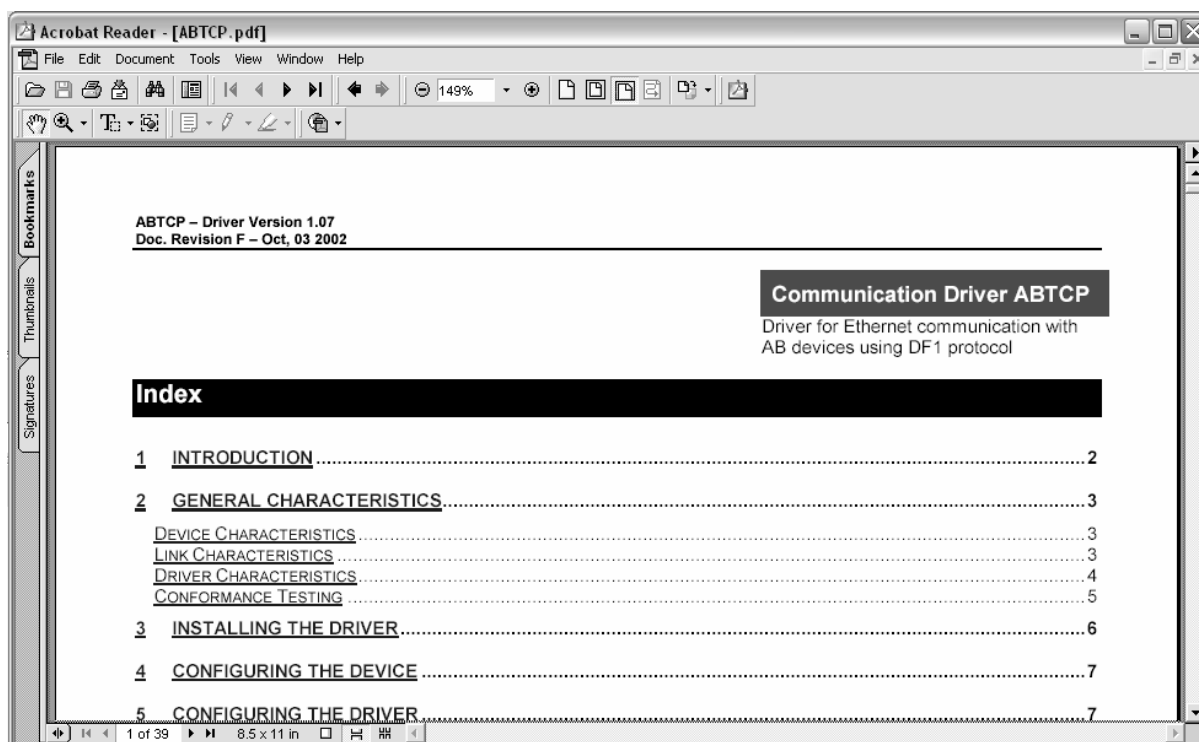


- Selecione o driver que você deseja consultar e clique em **Help**.



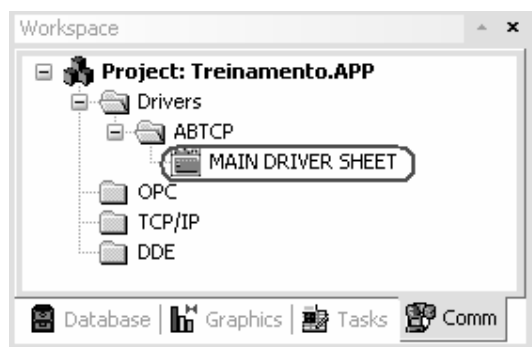
Você abrirá o Adobe Acrobat Reader com a documentação completa do driver.

Lá estão detalhadas todas as opções de configuração, notas explicativas e precauções que devem ser observadas. Também estão exemplos utilizados, pela InduSoft, durante o teste de homologação do driver.



Main Driver Sheet – A configuração mais simples

- Clique sobre a pasta ABTCP no Workspace e clique duas vezes sobre Main Driver Sheet:



Você verá um documento como o que segue. Preencha conforme a figura:

	Tag Name	Station	I/O Address	Action	Scan	Div	Add
1	Contador	192.168.1.88	N7:W0	Read+Write	Always		
2							
3							
4							
5							
6							

Disable: Permite a configuração de um tag que, em um desabilita toda a comunicação configurada em Main Driver Sheet, em zero permite a comunicação com o PLC.

Read Completed: O tag deste campo apenas recebe um feedback se a operação de leitura já foi efetuada, trigando o seu valor entre 0 e 1.

Read Status: Este tag recebe o Status da última comunicação. Zero significa **Ok**. Qualquer outro valor, positivo ou negativo é considerado **ERRO**. (Veja tabela de erros na documentação específica do driver).

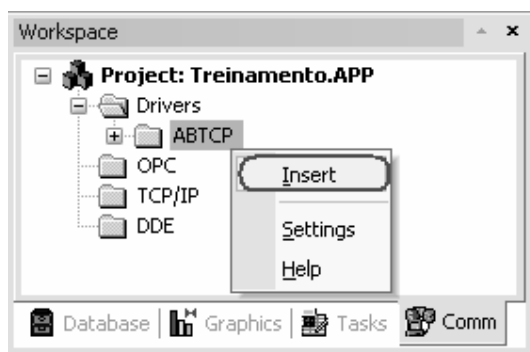
Write Complete: Feedback se a operação de escrita foi completada ou não, alternando os valores entre 0 e 1.

Write Status: Feedback do status da última comunicação. O tag deste campo receberá o valor 0 se a comunicação está Ok ou diferente de zero se não está. (Veja tabela de erros na documentação específica do driver).

Check Box Min e Max: Habilita os campos Max e Min, onde colocamos a faixa de validade dos dados.Exemplo, valores de memória variam de 0 to 4095 equivalem de 0% a 100% na interface com o usuário. Esta configuração é válida para TODOS TAGs da planilha, porém podemos colocar valores independentes de Max e Min em cada linha. No exemplo os valores seriam 0 e 100 para o Min e Max respectivamente.

Driver Worksheets – Otimizando a Comunicação

Adicione uma nova planilha de Driver clicando com o botão direito do mouse na pasta **ABTCP**, Insert.



As planilhas são compostas de duas partes: O cabeçalho e o corpo

Cabeçalho

No cabeçalho temos o controle da comunicação, enquanto no corpo os endereços dos tags que estão operando no CLP.

ABTCP001.DRV

Description: Contador do PC ☐ Increase priority

Read Trigger: Enable Read when Idle: 1 Read Completed: Read Status:

Write Trigger: Enable Write on Tag Change: 1 Write Completed: Write Status:

Station: 192.168.1.88 Header: N7:0 ☐ Min: Max:

É no **Cabeçalho** que configuramos várias informações, incluindo o tipo de operando que será usado na comunicação.

Campo Description: Descrição que será mostrada na árvore da pasta **ABTCP**

Increase priority - check Box: Em caso de ser uma planilha de leitura aumenta a prioridade desta planilha.

Read Trigger: Coloque aqui um tag que, sempre que seu valor for mudado será efetuada uma leitura no CLP. Isso permite um maior controle do fluxo de comunicação

Enable Read when Idle: Se o tag colocado aqui possuir o valor maior que zero, a leitura desta planilha será efetuada continuamente.

Read Completed: O tag deste campo apenas recebe um feedback se a operação de leitura já foi efetuada, trigando o seu valor entre 0 e 1.

Read Status: Já este tag recebe o Status da última comunicação. Zero significa **Ok**. Qualquer outro valor, positivo ou negativo é considerado **ERRO**.

Write Trigger: A mudança do valor do tag deste campo efetua uma escrita de toda a planilha no CLP.

Enable Write on Tag Change: Se este campo contiver um TAG com o valor maior que zero, sempre que algum tag do corpo da planilha mudar de valor somente este tag será escrito no CLP.

Write Complete: Feedback se a operação de escrita foi completada ou não, alternando os valores entre 0 e 1.

Write Status: Feedback do status da última comunicação. O tag deste campo receberá o valor 0 se a comunicação está Ok ou 1 se não está.

Station: Endereço do equipamento na rede. Pode ser um tag entre chaves { }.



Nota:

Este campo é totalmente variável em função do equipamento que estiver sendo utilizado.

Header: Cada driver tem a sua própria sintaxe para este campo, uma vez que cada dispositivo tem o seu próprio método de nomenclatura de operandos. Mas, na maioria dos casos, colocamos neste campo o tipo de operando com o qual queremos comunicar e o endereço inicial da planilha.

Veja alguns exemplos:

Driver	Cabeçalho	Significado
MODBUS	4X:100	4X indica que esta planilha comunica com os Registros de Propriedade, do endereço em 100. No caso AEG 984, o endereço em 400100.
OMPLC (Host Link)	IR:0	IR indica que esta planilha comunica com o I/O e Revezamento Internos, do endereço em 0. No caso de C200H, do endereço em IR00000.
FANUC (SNP)	%M	%M indica que esta planilha comunica com o operando %M. Não há nenhum endereço inicial para este driver.
ABKE (DF1)	N7:0	N7 indica que esta planilha comunica com o arquivo de N7, do endereço em 0. No caso de PLC-5/40, do endereço N7: 0.
AS511 (Siemens PG Port)	DB5:10	DB5 indica que esta planilha comunica com o Bloco de Dados número 5, palavra 10.

Corpo

O corpo da planilha de drivers é composto de 4 colunas: Tag Name, Address, Add e Div.

	Tag Name	Address	Div	Add
1	Contador	W0		
2				
3				
4				
5				
6				
7				

Tag Name: Nome do tag que terá a função de leitura ou de escrita de dados do PLC.

Address: Normalmente é o offset em relação ao endereço inicial declarado no campo **Header** do **Cabeçalho**, mas varia de acordo com o driver. Veja alguns exemplos a seguir:

DRIVER	Operando Interno do CLP	Header	Adress
OMPLC (OMRON)	IR0	IR:0	0
OMPLC (OMRON)	IR00001 (IR bit 1)	IR:0	0.1
OMPLC (OMRON)	IR10	IR:0	10
OMPLC (OMRON)		IR:10	0
FANUC (GE SNP)	%I00001	%I	1
FANUC (GE SNP)	%R00100	%R	100
ABKE (AB DF-1)	N7: 0/5	N7: 0	W0/5
ABKE (AB DF-1)	T4: 0. ACC	T4:0	W0.ACC
ABKE (AB DF-1)	T4:10.CON/15	T4:0	W10.ACC/15
MODBUS	100001	1X:1	0
MODBUS	400100	4X:50	50
AS511 (Siemens PG Port)	DB5, DW10	DB5:0	10
AS511 (Siemens PG Port)	DB5, DW10, bit 2	DB5:10	0.2

Nos exemplos dos drivers, você ajusta os Offsets relativos ao endereço inicial configurado no **Header**. Isto não é válido teclando um valor negativo. O valor zero 0 será subscrito.

Div / Add / Max / Min:

Column	Range of Values	Mean
Div	Todos Integer e Real	Em comando de leitura: Tag = (Valor de campo) / DIV Em comando de escrita: Valor em campo = Tag * DIV
Add	Todos Integer e Real	Em comando de leitura: Tag = (Valor de campo) + ADD Em comando de escrita: Valor em campo = Tag - ADD
Min	Todos Integer e Real	Faixa de validade dos dados independentes de Max e Min do cabeçalho da planilha
Max	Todos Integer e Real	Faixa de validade dos dados independentes de Max e Min do cabeçalho da planilha

Configurando o Sistema de Segurança

Crie três grupos de segurança: **Operacao**, **Manutencao** e **Engenharia**.

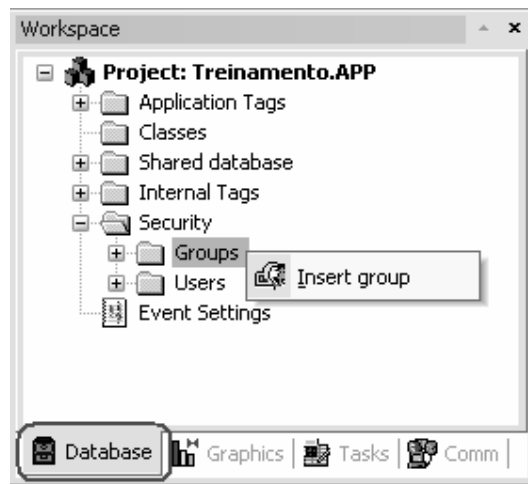


Nota:

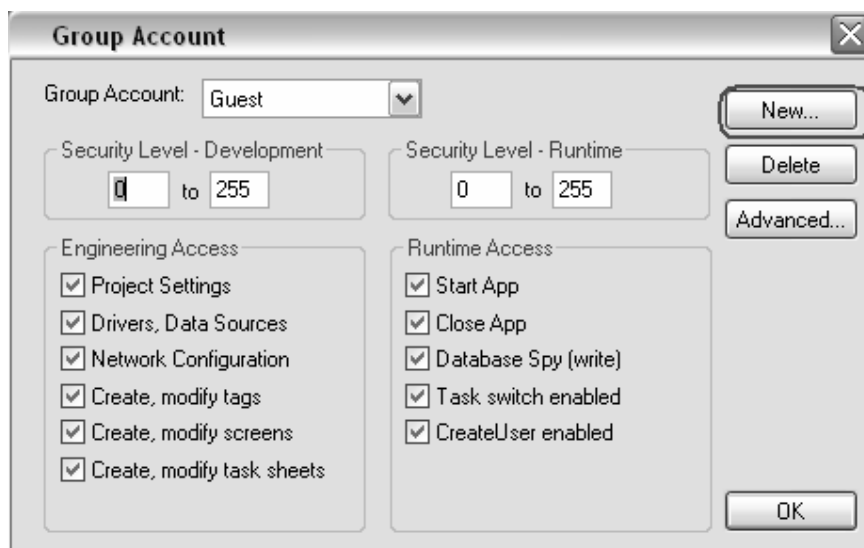
- Apesar de o InduSoft Web Studio suportar caracteres acentuados, de acordo com a configuração International do Control Panel, do Windows, é recomendável não usa-los, pois se incorre no erro de declarar com acento e tentar referenciá-los sem acento em outros pontos do projeto. Contudo, isso é uma opção do desenvolvedor do sistema.

Inserindo Grupos de Usuários

- No tab **Database**, clique na pasta **Security** e com o botão direito clique na sub-pasta Groups, e então clique em **Insert group**.

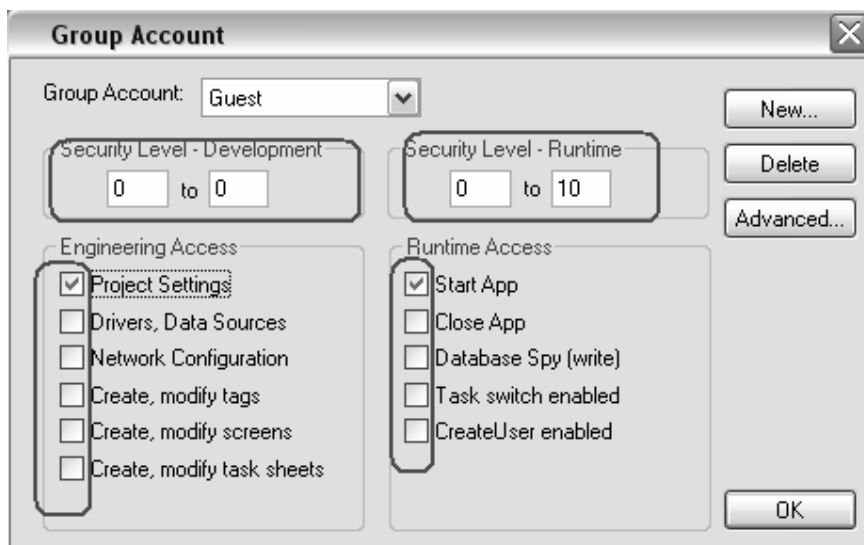


Quando aparecer a tela Group Account como mostrado abaixo: clique em **New...** para poder abrir o menu New Group Account

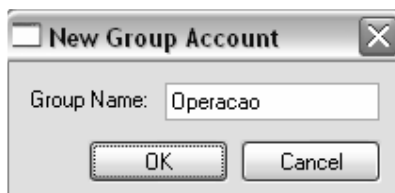


O Group Account Guest é padrão, não pode ser apagado (assim como o usuário Guest) e corresponde ao Group Account que fica válido sempre que uma operação Logoff ocorre. Esse grupo deve ser restrito ao mínimo acesso que seu projeto vai permitir a quem não é usuário do sistema.

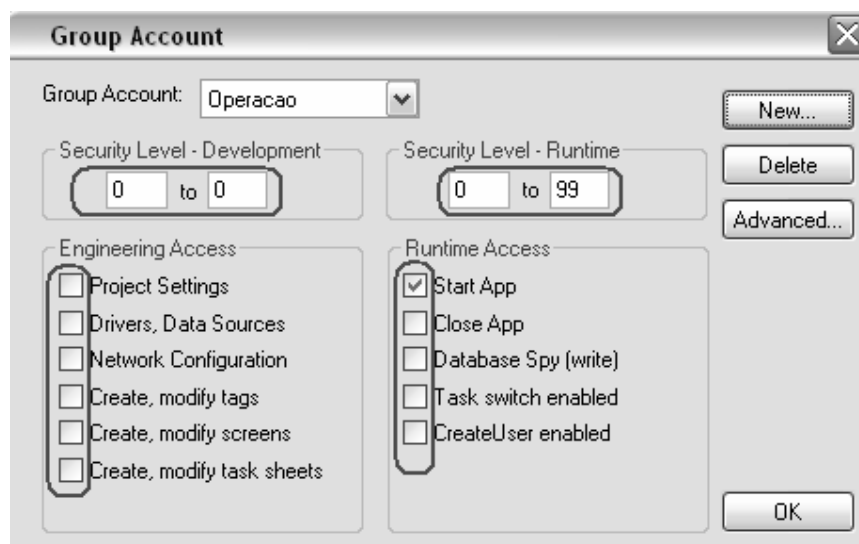
- Preencha conforme sugerido a seguir:



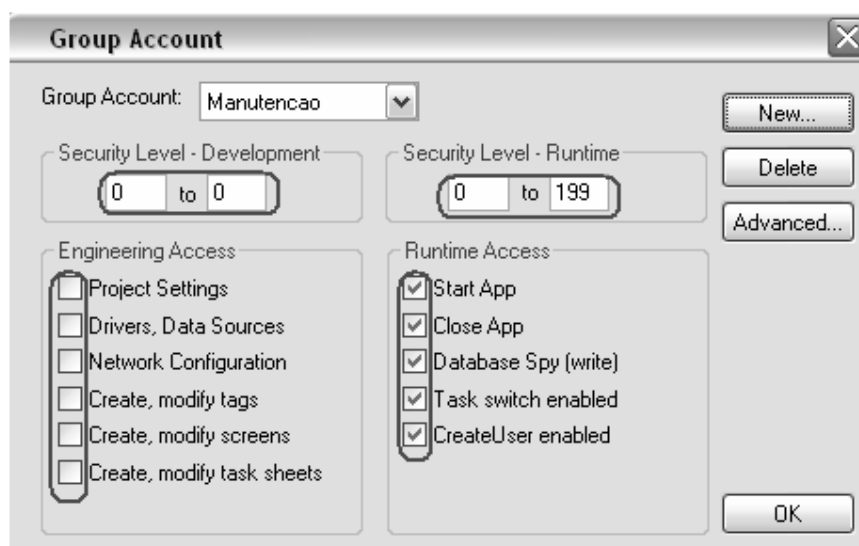
- Configure a tela New Group Account, com o nome de **Operacao** como mostrado abaixo:



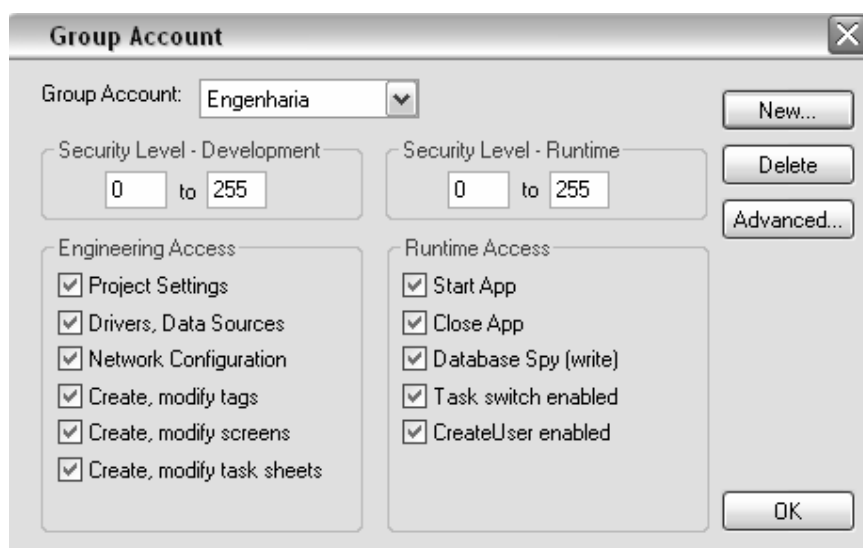
- Pressione OK, o menu Group Account abrirá novamente, configure a janela Group Account como mostrado abaixo:



- Clique em OK e pronto ele será salvo automaticamente.
- Repita o processo, mas configure com o nome de **Manutencao**.
- Configure a janela Group Account como mostrado abaixo:



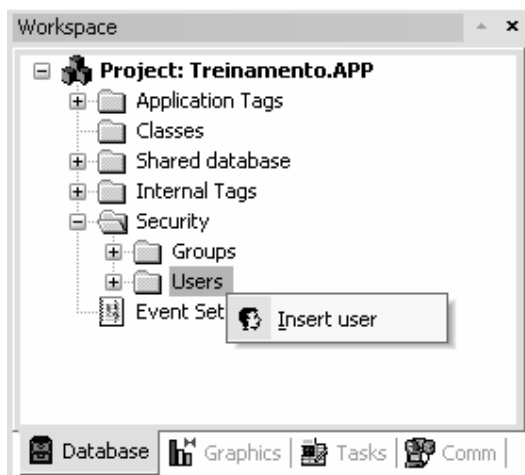
- Por último, repita o processo e configure o grupo com o nome de Engenharia.
- Configure a janela Group Account como mostrado abaixo:



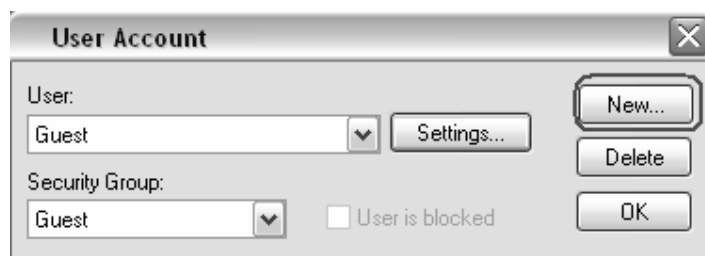
- Pressione OK.

Inserindo Usuários ao Sistema de Segurança

- Agora crie usuários para cada grupo. Clique com o botão direito na sub-pasta **User** e clique em **Insert user**



Aparecerá a janela **User Account**, como mostrado abaixo:



- Clicando em New na janela **User Account**, Crie um novo usuário chamado: **Joao**.
- Associe-o ao grupo **Operacao**;
- Preencha o campo em **Password**, configurando assim uma senha para **Joao**;



- Pressione **OK**.
- O próximo usuário será **Jose**. Repita todo o processo, mas crie-o no grupo **Manutencao**.
- Por último, o mesmo deve ser feito com o usuário **Maria**, no grupo **Engenharia**.

Nas configurações de telas, botões e outros objetos de “input” do operador, você vai usar os níveis de acesso que acaba de criar nesse módulo.

Lembre-se que os Grupos ficaram assim:

Guest 0-10, Operacao 0-99, Manutencao 0-199, Engenharia 0-255.

Configurando a Password Principal

Todo o Security System é protegido por uma Main Password, o que previne usuários do sistema de entrarem e alterarem seus próprios acessos ou criarem novos usuários com habilitações indevidas.

- Clique com o botão da direita sobre a pasta Security, para ter acesso a esta janela de diálogo:



- Clique no botão Main Password para configurar a senha principal do sistema



IMPORTANTE:

- Essa senha só poderá ser “quebrada” enviando o sistema a InduSoft e este processo desconfigura todo o Security System, ou seja, perdem-se as configurações de todos os grupos e usuários.

Criando Tags no Database

- No Workspace, selecione o tab **Database**. Clique duas vezes na pasta **Application Tags** para expandi-la.
- Clique duas vezes em **Datasheet View**.
- O Database será preenchido e alterado durante o desenvolvimento da aplicação. Porém, para iniciarmos uma aplicação de exemplo, podemos já previamente definir alguns tags que sabemos que serão utilizados. Criaremos três Tanques cada qual controlado por duas válvulas.
- A primeira válvula enche o tanque e a segunda o esvazia.

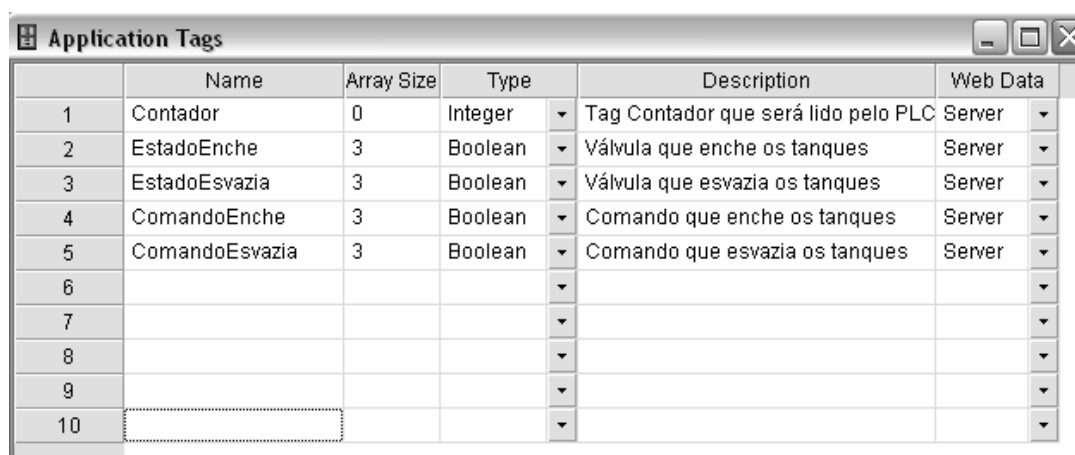
Para sabermos o estado da primeira válvula, teremos o tag **EstadoEnche** e para a segunda **EstadoEsvazia**.

Como cada válvula só pode estar aberta ou fechada, estes tags devem ser booleanos.

Sendo três Tanque, precisaremos de três válvulas para enche-los e outras três para esvaziá-los. Portanto usaremos Arrays para rapidamente criarmos as válvulas.

Vamos criar agora os tags que enviarão os comandos para abrir ou fechar as válvulas.

Esses comandos, num sistema real, seriam enviados aos CLPs através do driver, endereçando esses tags para serem pontos de I/O.



	Name	Array Size	Type	Description	Web Data
1	Contador	0	Integer	Tag Contador que será lido pelo PLC	Server
2	EstadoEnche	3	Boolean	Válvula que enche os tanques	Server
3	EstadoEsvazia	3	Boolean	Válvula que esvazia os tanques	Server
4	ComandoEnche	3	Boolean	Comando que enche os tanques	Server
5	ComandoEsvazia	3	Boolean	Comando que esvazia os tanques	Server
6					
7					
8					
9					
10					

- Feche o DataBase, não é necessário salvar.

Criando novas telas

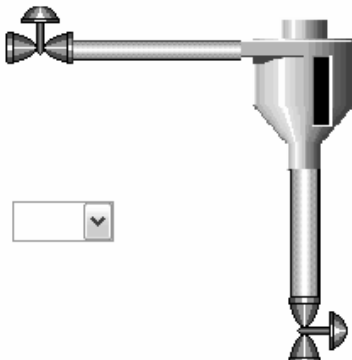
Antes de criar uma tela, vamos entender o que o programa nos oferece para a criação de um bom aplicativo. Vamos deixar alguns comandos e animações sendo mostrados o tempo todo na tela e em outro espaço vamos variar o que está sendo mostrado. Assim criaremos uma tela para os comandos, animações constantes e variações:

- **CABEÇALHO:** Mostrará data e hora o tempo todo, bem como terá os links para as outras telas.
- **PRINCIPAL:** As telas de Trend, Alarmes, Processo, Receitas, etc... terão as mesmas medidas, que será dada por essa tela.

#####	#####	Trend OnLine	Alarme OnLine	Relatórios	Principal
Usuário :#####		Trend Histórico	Alarme Histórico	Receitas	Sair

Tanque

v



Temperatura

#####

Pressão

#####

Nível

#####

Diretório do aplicativo: #####

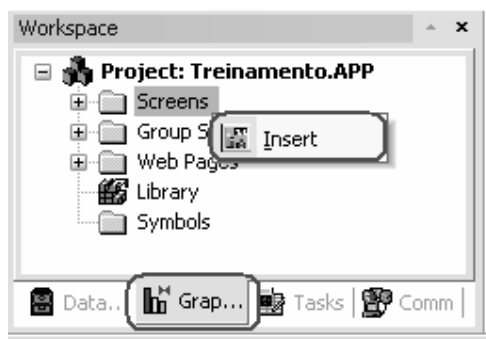
Nome do computador: #####

Endereço do computador: #####

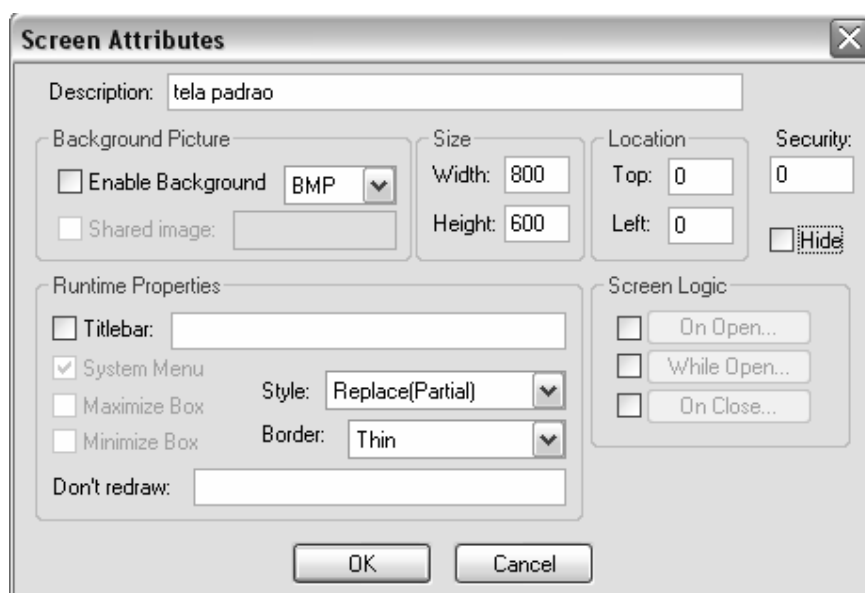
Acima a tela: **PRINCIPAL** já com **CABEÇALHO**.

Usando a estrutura acima recomendada, vamos começar a criar nossas telas. Primeiramente criaremos o **CABEÇALHO** em uma tela (STANDARD) que servirá de padrão para as demais.

- Selecione o tab **Graphics** no Workspace e selecione a pasta **Screens**.
- Agora, clique com o botão direito na pasta **Screens** e clique na opção **Insert**.



- Na janela **Screen Attributes** devemos configurar alguns atributos gerais sobre a tela que está sendo criada. Primeiro configure a tela **STANDARD** como mostrado na figura abaixo:



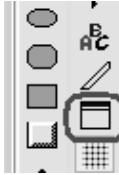
Pressione o botão **OK** e uma tela vazia deverá aparecer com os atributos configurados para ela.

Desenhando o CABEÇALHO e os primeiros objetos

Embora não haja um único objeto em nossa tela, é uma boa prática de programação não trabalhar com arquivos cujo nome tenha sido automaticamente atribuído pelo software, como seu documento que no momento chama-se Display1. Esse é um nome padrão e se você distrair-se e salvar a sua tela padrão como Display1, no futuro você pode vir a salvar outro documento substituindo este.

- Vá ao menu File, Save As, e salve a tela com o nome **Standard**.

- Depois de criar a tela, mudemos a cor de fundo usando o *background color*. Selecione uma



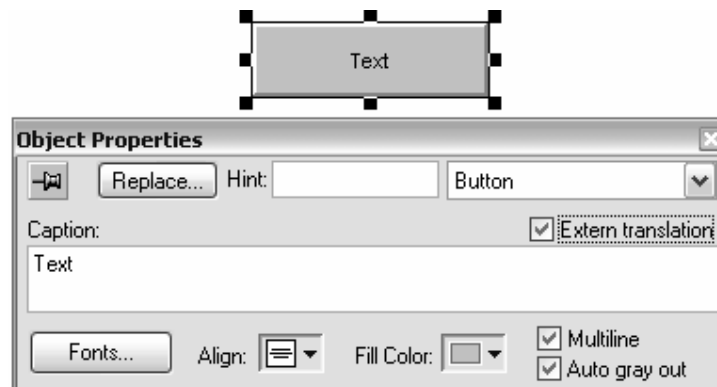
opção de cor desejada, utilizando o ícone do toolbar no canto direito da tela.

Crie agora a navegação para as telas. Inicie criando um botão como todos os seus elementos: texto, comando, tamanho correto. Só então copie os outros sete botões a partir deste.

- Utilizando o ícone Button do Toolbar, arraste o mouse traçando o botão no tamanho adequado.



- Ao terminar o desenho do botão, dê um duplo clique sobre ele. Você terá a janela de Object Properties como esta:



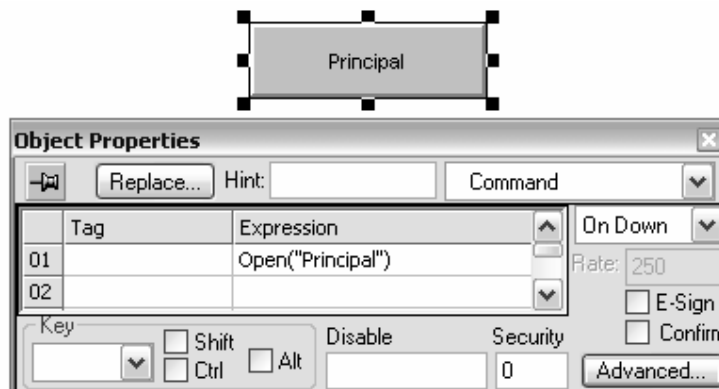
- Modifique o campo Caption para **Principal**.
- A este botão vamos aplicar a propriedade Command.
- Clique no ícone Command do Toolbar.



Ao clicar no ícone Command, você notará que a janela Object Properties modificou-se. Se ela não estiver visível, dê um duplo clique no botão.

- Digite no campo Expression a função Open(“Principal”) como mostra a figura. Você está programando este botão para abrir uma tela cujo nome do arquivo é Principal.

- Você verá a Object Properties como mostra a figura que segue:



- Execute esse mesmo procedimento para, criar todos os outros botões digitando as suas descrições:
 - Trend Online
 - Trend Histórico
 - Receitas
 - Relatórios
 - Alarme Online
 - Alarme Histórico
 - Sair



Nota:

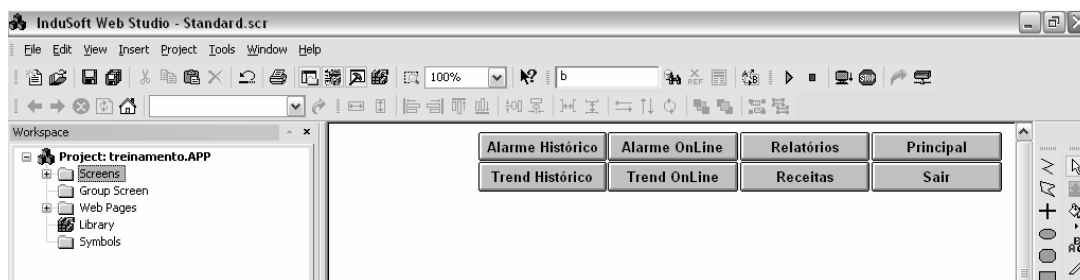
- É possível copiar e colar objetos das telas utilizando-se do recurso Ctrl+C Ctrl+V. Os objetos copiados serão colocados sempre no canto superior esquerdo da tela.
- Outra forma é pressionar a tecla Ctrl e clicar no objeto a ser duplicado, depois disso é só arrastá-lo.



Nota:

Pode ser interessante utilizar-se do recurso Disable Drag através do atalho Ctrl+D. Este recurso faz com que os objetos não possam mais ser movimentados na tela utilizando o mouse. Para desabilitar esta opção, basta pressionar Ctrl+D novamente.

- Clique em cada objeto, para mudar sua propriedade **Caption**, modificando para o nome da tela adequado e modificando também no parâmetro da função **Open**(). Em cada botão você fará duas alterações, de modo que a tela fique como esta:



IMPORTANTE:

- Selecione cada botão, adicione a propriedade **Command**, e configure em Expression na Janela Object Properties as seguintes funções.

- | | |
|----------------------------|--|
| ▪ Trend Online: | <code>Open("TrendOnline")</code> |
| ▪ Trend Histórico: | <code>Open("TrendHistórico")</code> |
| ▪ Receitas: | <code>Open("Receitas")</code> |
| ▪ Relatórios: | <code>Open("Relatórios")</code> |
| ▪ Alarme Online: | <code>Open("AlarmeOnline")</code> |
| ▪ Alarme Histórico: | <code>Open("AlarmeHistórico")</code> |
| ▪ Sair: | <code>Open("Sair")</code> |



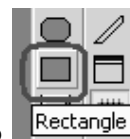
Nota:

A função `Open()`, abre uma tela. Não é necessário que a janela tenha sido criada para que programemos um botão para chamá-la. Mas para que a função funcione aí sim a tela já deve ter sido criada. O nome da tela a ser chamada, deve estar entre aspas por tratar-se de uma constante do tipo String. Se colocarmos ao invés disso um tag do tipo string, sendo que este tag possui como valor, o nome de uma tela, aí então podemos colocar o nome do tag sem aspas.

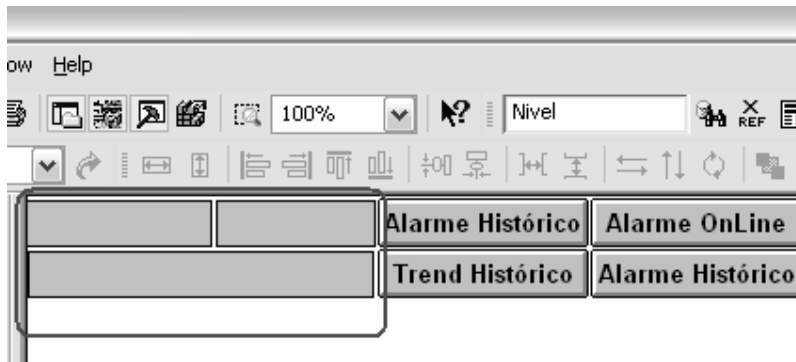
Agora desenharemos alguns retângulos e textos que farão parte da tela no canto superior esquerdo. Esses campos mostrarão a data e hora atual, bem como o nome do usuário logado.

Para isso, aprenderemos mais algumas coisas do Toolbar. Quando mudamos algum dado (setting), como por exemplo, cor e espessura das linhas, cor de preenchimento de objetos, formatação de fontes de texto etc., essas mudanças passam a ter efeito no próximo objeto desenhado e em todos que estejam selecionados.

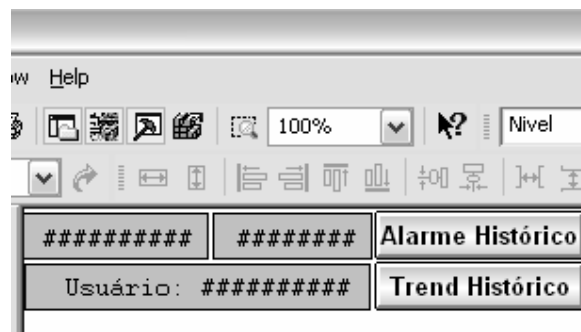
- Desenhe agora três retângulos, com a sua cor desejada.



- Para desenhar os três retângulos clique no ícone mostrado ao lado e desenhe como mostra a figura abaixo:



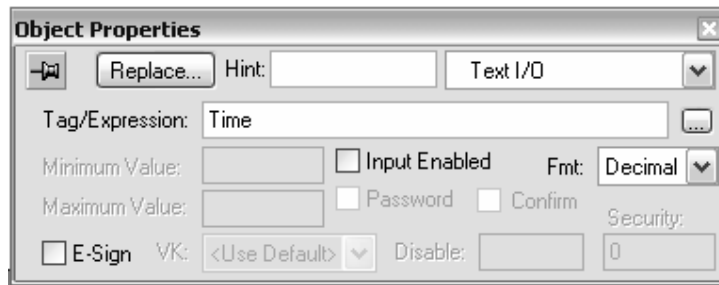
- Para inserir caracteres de texto nos retângulos, usaremos o ícone mostrado ao lado .
- Clique no ícone e depois clique dentro de um dos retângulos que foram desenhados.
- Para mostrarmos o valor de um tag na tela, usamos o caractere # (*sustenido, cerquinha, jogo da velha etc.*). O objeto texto pode combinar texto fixo com a máscara para dinâmica Text, como é o caso em Usuário: #####. O mesmo número de caracteres sustenido (#) que for digitado, será o mesmo número de caracteres do tag que serão mostrados. Como queremos mostrar a data utilizando o formato **MM/DD/AAAA**, precisamos de 10 caracteres no retângulo da esquerda, 8 caracteres para o retângulo da direita e 10 (ou mais) caracteres no retângulo inferior.



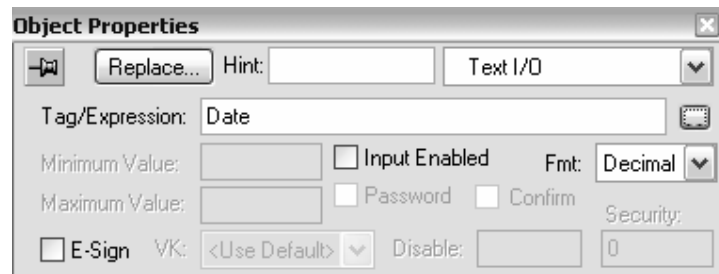
- Para mostrar o valor de um tag ou mesmo expressões numa tela, aplicaremos nos textos onde houver os caracteres #####. a propriedade **Text I/O**, através do ícone:



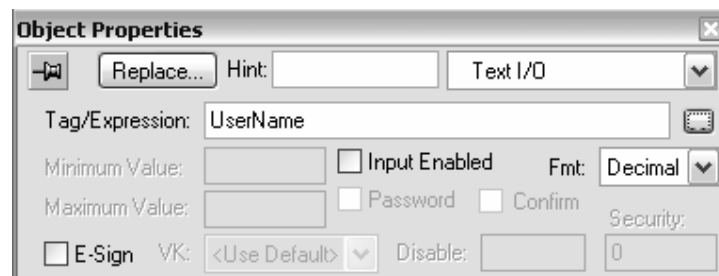
- Clique duas vezes no retângulo com 8 caracteres sustenido (#) para hora, e em seguida no ícone **Text I/O**, a janela **Objects Properties** aparecerá. Note que o check -list superior direito terá além da opção **Text**, a propriedade **Text I/O**.
- No campo **Tag/Expression** preencha com **Time**. Como na figura abaixo;



- Selecione o outro retângulo com texto #####, com 10 caracteres para data; aplique a propriedade **Text I/O**.
- Ao abrir a janela Objects Properties digite Date no campo Tag/Expression. Como na figura abaixo:



- No último retângulo com o texto **Usuário: #####** aplique a propriedade **Text I/O** e digite **UserName** no campo **Tag/Expression**.

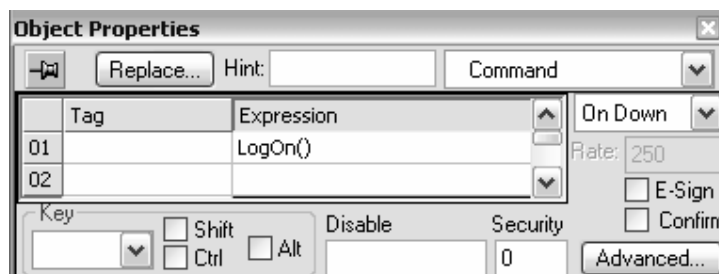


Anteriormente, você aplicou propriedade de comando a um botão. Porém a propriedade de comando pode ser aplicada a outros tipos de objetos, como uma figura geométrica, no nosso caso o retângulo que faz a moldura para a informação de Usuário.

- Clique no retângulo para selecioná-lo; em seguida no ícone **Command** mostrado na figura:



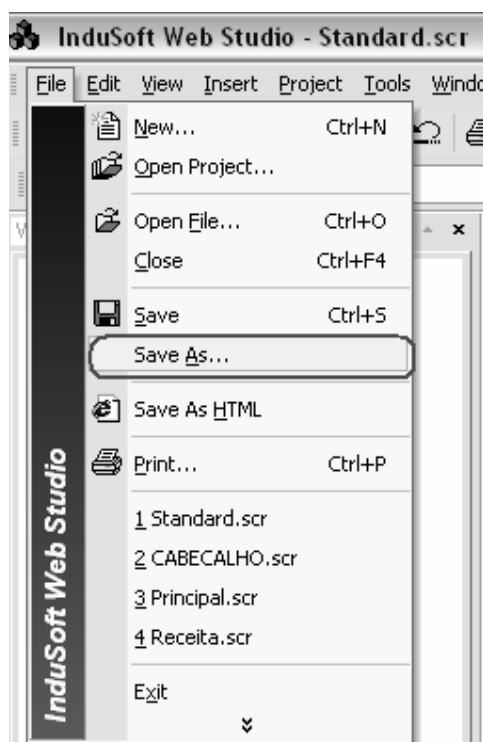
- Configure a propriedade **Command** como mostrado abaixo.



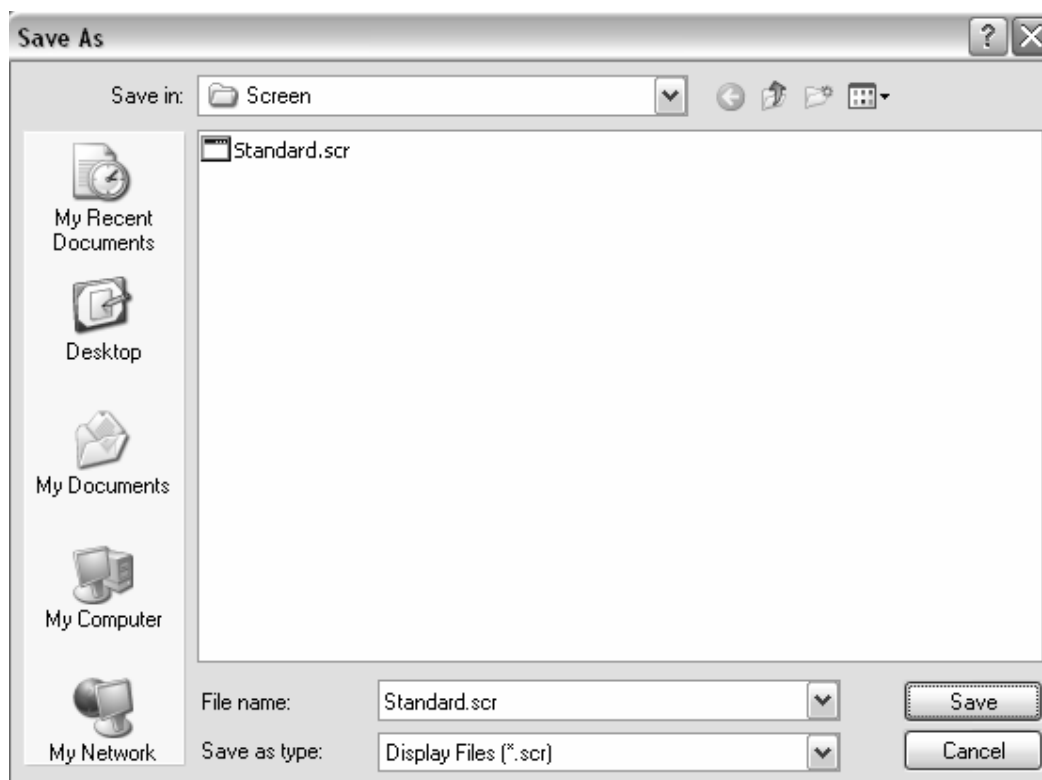
Nota:

A função **LogOn()**, chama um diálogo onde se é solicitado o nome e a senha de um usuário.

- Salve a tela como **Standard**. Se você já a salvou no início, basta clicar no ícone Save do Toolbar.
- Para **salvar**, clique no menu **File** na barra superior, e no item **Save As**.



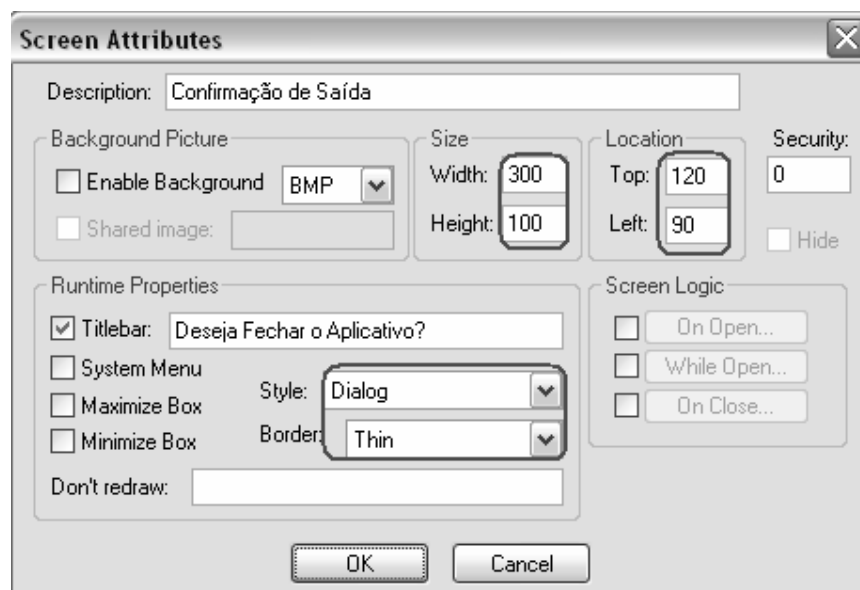
- Digite o nome STANDARD no campo File Name, e clique em Save.



- Feche a tela STANDARD.

Iniciaremos a construção de uma nova tela.

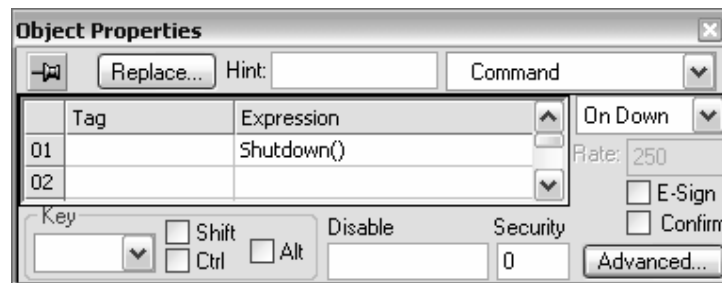
- Agora crie a tela **Sair**, para inserir uma nova tela clique com o botão direito na pasta **Screens**, clique em **Insert** e configure como na figura abaixo:
- Verifique as medidas e as configurações da tela.



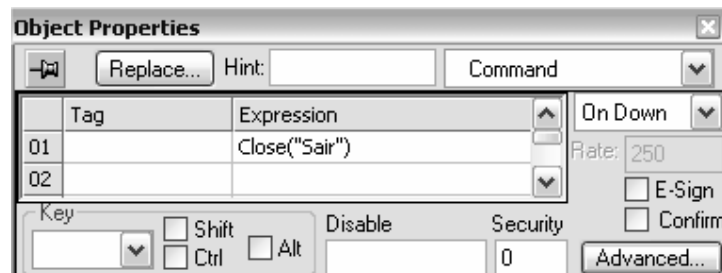
- Como você já aprendeu a criar botões com dinâmicas de comando, não vamos repeti-las. Na nova tela desenhe os dois botões e aplique dinâmica Command:



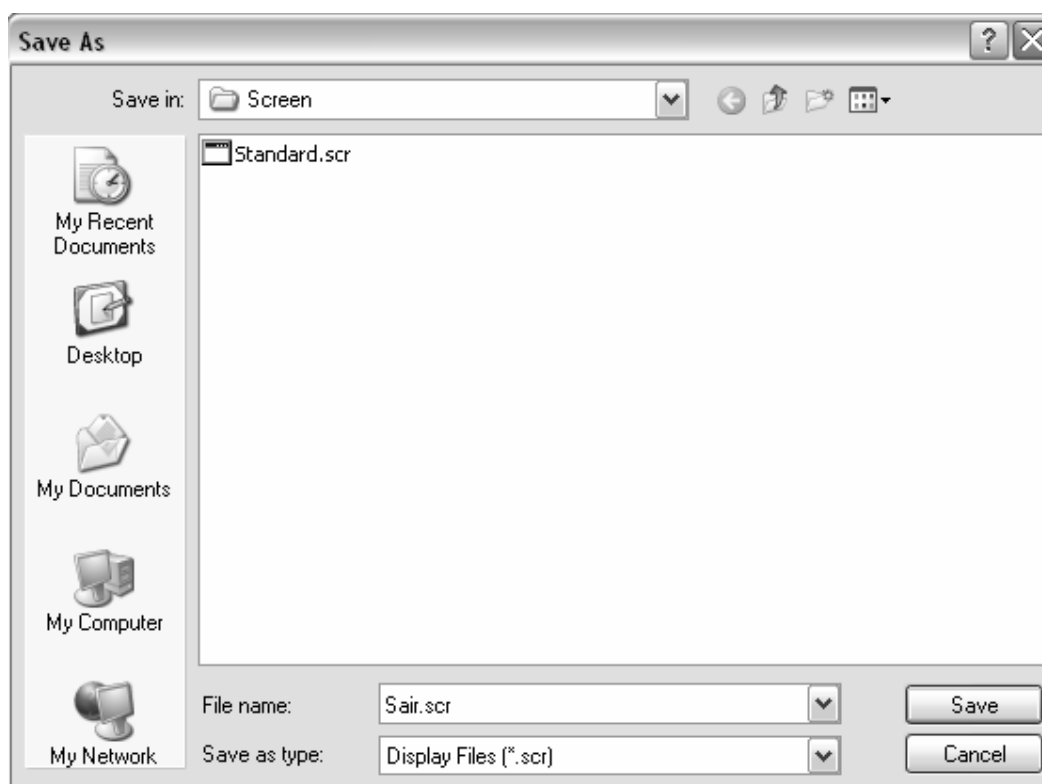
- No botão OK, a propriedade Command terá a Expression **Shutdown()**, como mostra a figura abaixo.



- No botão Cancela, a propriedade Command terá a Expression **Close("Sair")**, como mostra a figura abaixo.



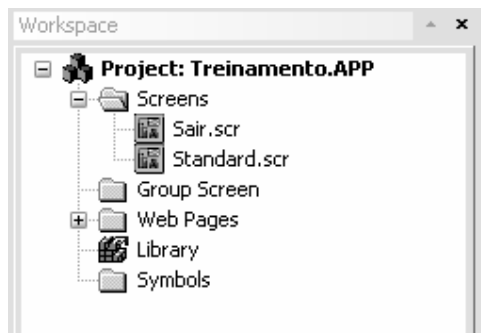
- Salve a tela como Sair
- Para **salvar**, clique no menu **File** na barra superior, e no item **Save As**.



- Agora feche a tela **Sair**.
- Para fechar a tela confirmação clique com o botão direito em algum lugar vazio da tela, em seguida clique na opção close, ou pela tecla de atalho CTRL+F4.

Criando a tela PRINCIPAL a partir da Tela STANDARD

- Abra a tela **Standard**.
- Para abrir uma tela no Workspace, selecione o tab **Graphics**. Clique duas vezes na pasta **Screen** para expandi-la.
- Clique duas vezes na tela **Standard**.



A tela **Principal** será a primeira tela a ser exibida na partida do sistema. Vamos aproveitar o cabeçalho já desenvolvido na tela Standard e inserir novos objetos a partir desta.

- Clique em File, Save As, para salvar esta tela como **Principal**.



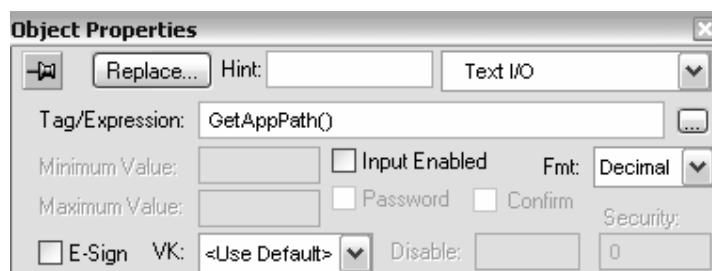
- Clique no ícone para criação de textos e configure o texto mostrado abaixo na parte inferior esquerda da tela:

```
.....
: Diretório da Aplicação: #####
: Nome do Computador: #####
: Endereço IP: #####
: .....
```

- Para sabermos o diretório da aplicação, usamos a função **GetAppPath()**.



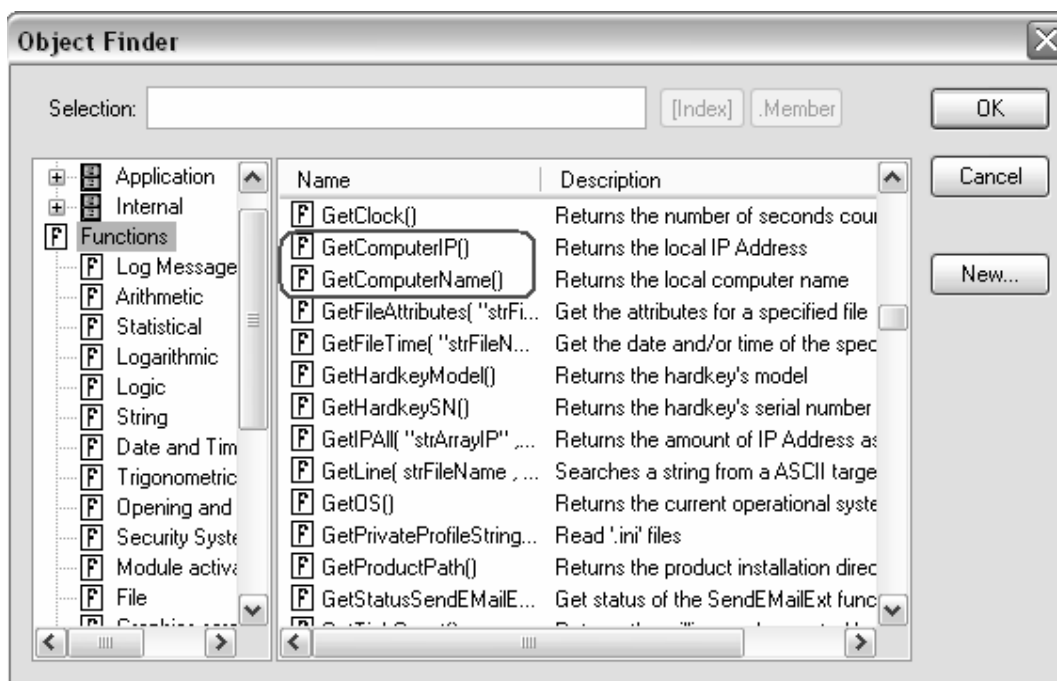
- Clique duas vezes no objeto texto, clique no ícone da propriedade de **Text I/O** e digite **GetAppPath()**, no campo Tag/Expression:, como na figura abaixo:



Nota:

Observe que ao final do campo Tag/Expression há um botão com reticências... Este chama a janela denominada **Object Finder**, que entre outras facilidades, proporciona uma busca de todas as funções de Script Language do InduSoft Web Studio.

Experimente inserir as funções **GetComputerName()** no campo Nome do Computador e **GetComputerIP()** no campo Endereço IP, utilizando o Object Finder.



Desenhando itens de processo e sinóticos

Nesta tela principal, iremos:

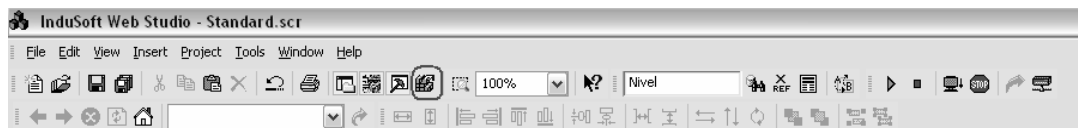
- Mostrar as propriedades de três Tanques (**Temperatura, Pressão e Nível**) em formato gráfico e numérico.
- Mostrar o estado das válvulas dos Tanques através de objetos da biblioteca.
- Inserir comandos de abertura e fechamento das válvulas.



Nota:

Partindo do princípio que os três Tanques serão idênticos, aproveitaremos esta aplicação para entender o conceito de Vetores. Portanto criaremos uma tela com tags indexados onde mostraremos os valores das variáveis de um Tanque por vez.

- Para criar o tanque, os canos e as válvulas, usaremos objetos da biblioteca. Abra a janela da biblioteca de símbolos, clicando no ícone mostrado na figura abaixo:



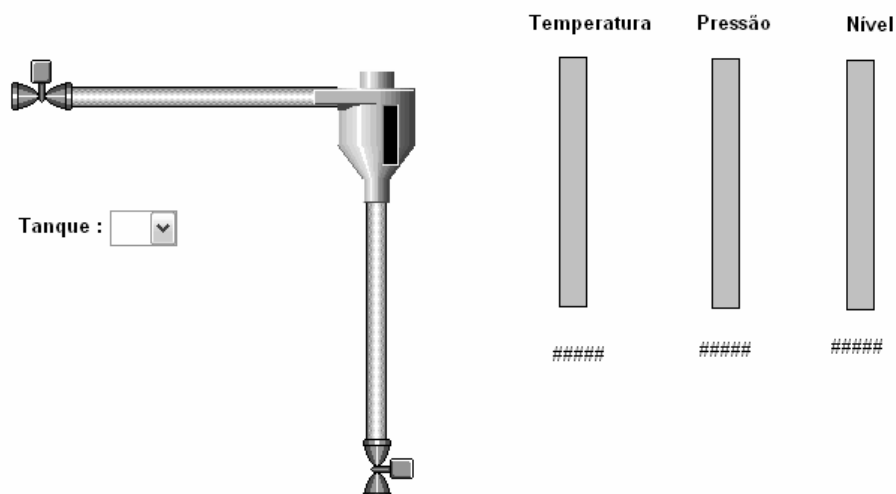
- Para importar um objeto da biblioteca para a tela da aplicação, simplesmente clique no objeto e depois na tela.
- Há outras formas de se desenhar além de utilizar a biblioteca, mas o propósito deste tutorial é também que você perceba como pode otimizar o desenvolvimento de seus projetos através da manutenção de uma biblioteca com seus símbolos típicos.
- Selecione objetos como válvulas (na janela Valves da biblioteca) e o tanque (em Tanks) como mostrado abaixo.
- Desenhe também os três retângulos da posição vertical.



Nota:

Desenhe os retângulos, e não utilize os objetos de bargraph da biblioteca, pois eles têm configurações diferentes das propostas nesse tutorial.

#####	#####	Alarme Histórico	Alarme OnLine	Relatórios	Principal
Usuario: #####		Trend Histórico	Trend OnLine	Receitas	Sair



Diretório da Aplicação:#####
Nome do Computador:#####
Endereço IP:#####



IMPORTANTE:

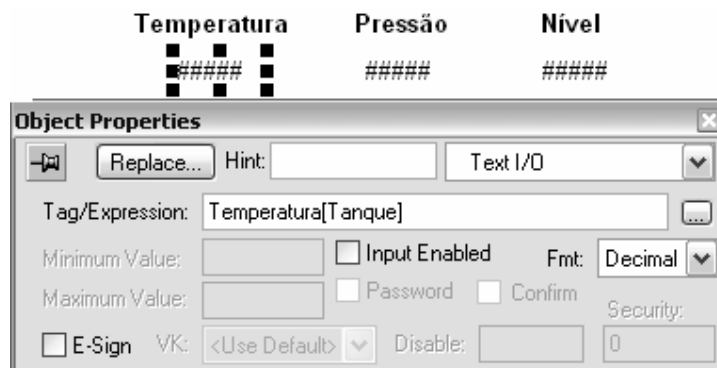
- Crie os tags Temperatura, Nivel, Pressao e Tanque.

Application Tags					
	Name	Array Size	Type	Description	Web Data
1	Contador	0	Integer	Contador do PLC	Server
2	EstadoEnche	3	Boolean	Válvula que enche os tanques	Server
3	EstadoEsvazia	3	Boolean	Válvula que esvazia os tanques	Server
4	ComandoEnche	3	Boolean	Comando que enche os tanques	Server
5	ComandoEsvazia	3	Boolean	Comando que esvazia os tanques	Server
6	Nivel	3	Real	Nivel do Tanque	Server
7	Temperatura	3	Real	Temperatura do tanque	Server
8	Tanque	0	Integer	Número do Tanque Selecionado	Server
9	Pressao	3	Real	Pressão do tanque	Server

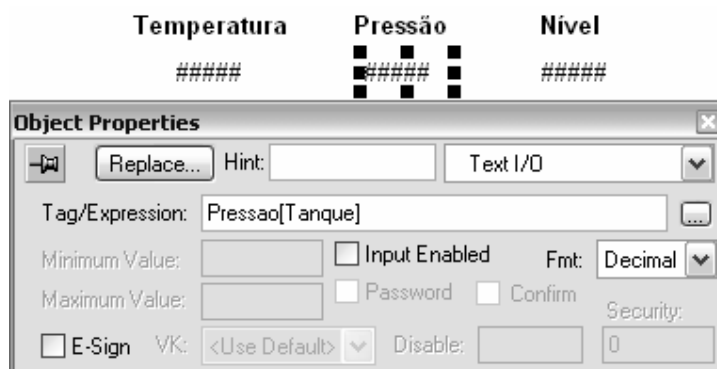
- Clique duas vezes no texto ### abaixo da palavra **Temperatura**, clique no ícone da propriedade



do ícone **Text I/O** e configure como mostrado abaixo:



- Selecione o texto **###** abaixo de da palavra **Pressão**, atribua propriedade de **Text/IO** e configure como mostrado abaixo:



- Por último, selecione o texto **##** abaixo da palavra **Nível**. Configure seu **Text I/O** como os anteriores, seguindo o exemplo abaixo:



A fim de mostrar os valores de Temperatura, Pressão e Nível graficamente, usaremos a propriedade **Bargraph** nos três retângulos desenhados acima dos textos.

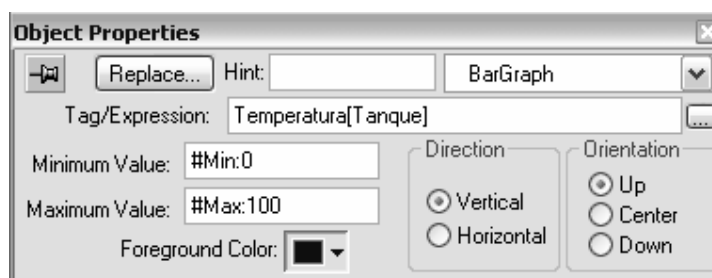
No nosso exemplo, o retângulo vai ser preenchido de acordo com o valor do tag associado ao seu bargraph.

- Clique duas vezes no retângulo acima do texto **Temperatura** e clique no ícone da propriedade

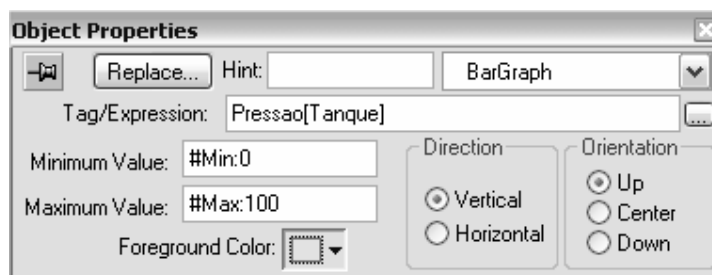


Bargraph

- Configure o retângulo como mostrado abaixo:



- Clique duas vezes no retângulo sobre o texto **Pressão** e clique no ícone **Bargraph**. Configure como mostra o exemplo abaixo:



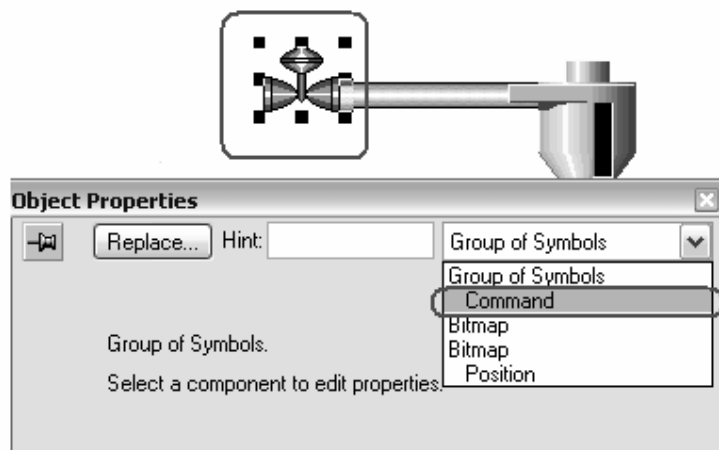
- Por último, clique duas vezes no retângulo sobre o texto **Nível** e clique a propriedade **Bargraph** e configure como mostrado abaixo.



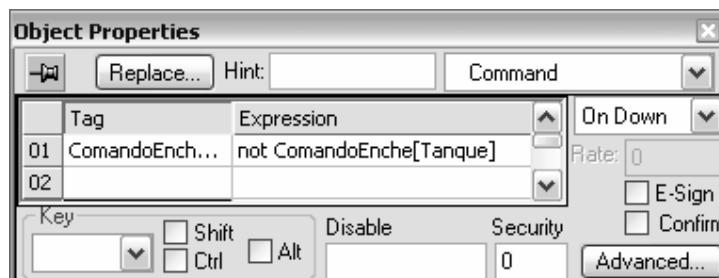
Agora vamos configuras as válvulas.

Chamaremos a válvula que se encontra na horizontal de **ValvulaEnche**.

- Clique sobre a válvula e aplique a propriedade de Command, da mesma forma como fez com os botões ou a moldura para nome do Usuário.
- Dê um duplo clique na válvula e localize a propriedade de Command, como mostra a figura abaixo:



- Clique na propriedade Command, apague o conteúdo da mesma e preencha como mostra a figura abaixo:



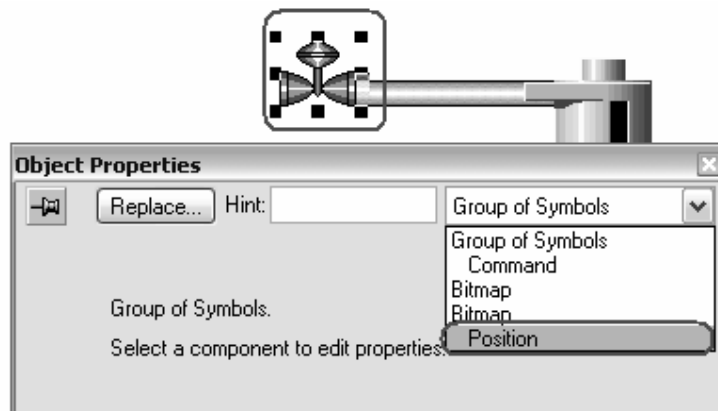
Tag

ComandoEnche[Tanque]

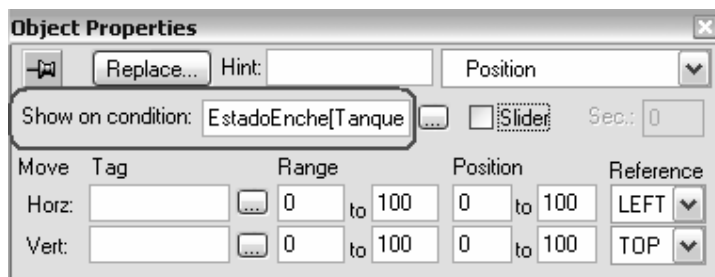
Expression

not ComandoEnche[Tanque]

- Identifique a propriedade **Position** que se encontra no final da lista de propriedades da válvula, como na figura abaixo:



E no campo **Show on condition**, apague o conteúdo da mesma e digite **EstadoEnche[Tanque]**, conforme mostrado abaixo:



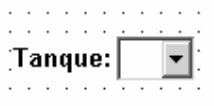
Numa aplicação real, este tag enviaria comandos para abrir e fechar as válvulas do sistema. Portanto este tag seria um tag de **ESCRITA** no campo.

Chamaremos de **ValvulaEsvazia** a válvula que está na posição vertical. Para a ValvulaEsvazia você deve proceder tal qual configurou a ValvulaEnche, primeiro selecionando a propriedade Command e configurando-a, depois configurando a propriedade Position, ambas já existente no objeto.

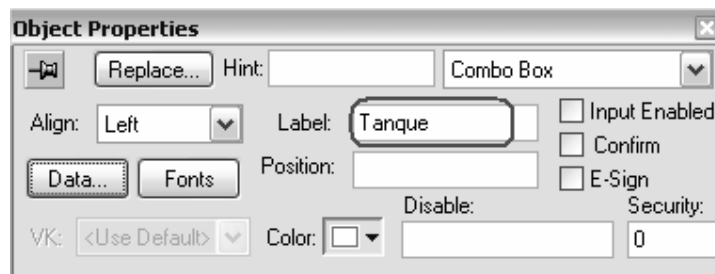
Finalmente crie os comandos para o tag **Tanque**. Observe a figura abaixo. Nela há um objeto texto e um objeto ComboBox. O objetivo é criar uma interface para selecionar o número do tanque que se vai monitorar.



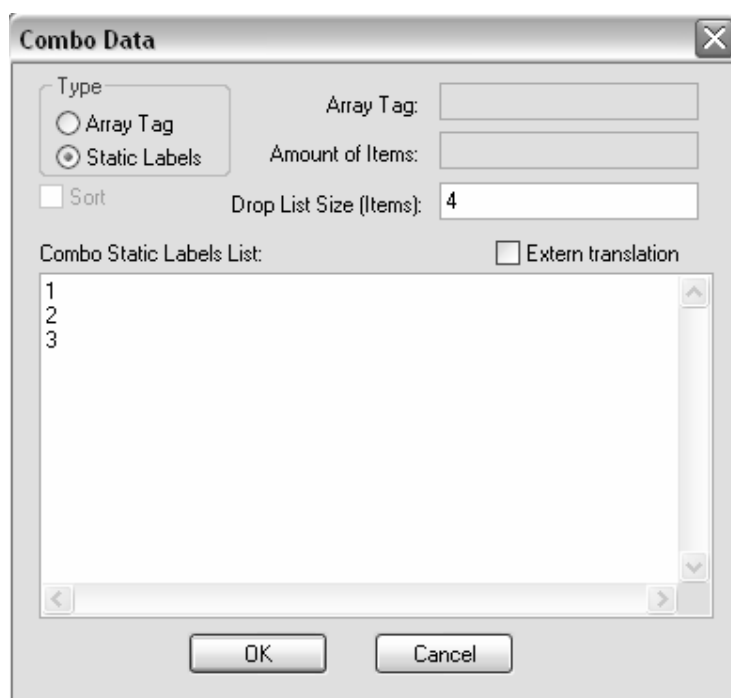
- Insira o objeto ComboBox através do ícone no toolbar
- Arraste o mouse formando um retângulo, para que fique do tamanho adequado



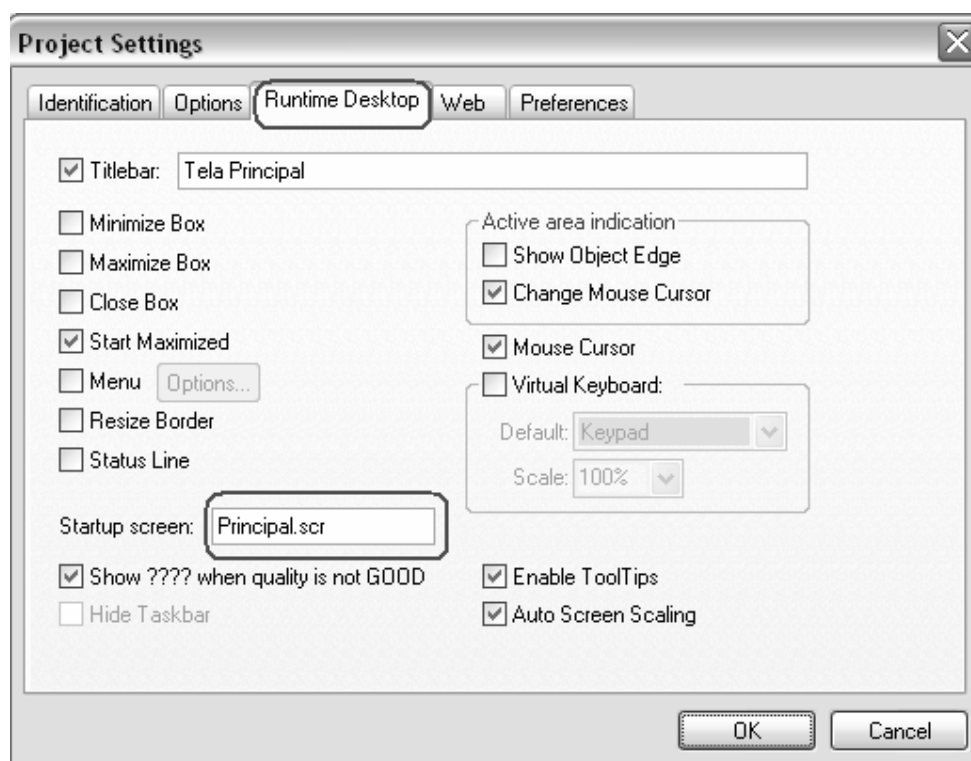
- Duplo clique sobre o ComboBox para configurar suas propriedades dinâmicas, conforme as figuras que seguem:



- Clique no botão Data... e configure como mostra a figura:



- Clique em OK.
- Salve a tela como Principal.
- Para **salvar**, clique no menu **File** na barra superior, e no item **Save As** e digite o nome **Principal** no campo **Nome do arquivo**, e clique em Salvar.
- É interessante configurarmos a tela que dará início, quando a aplicação for Startada em modo de RunTime.
- Para isso vá ao Menu Project / Settings, Aba RunTime Desktop e Configure o campo Startup Screen, como mostrado abaixo:



- Em seguida clique em OK.

A tela Principal já está pronta para ser executada. Clique no ícone **Run Application** do Toolbar:



Expressões, Funções e Linguagem Script (Script Language)

Este capítulo visa apresentar a linguagem do programa InduSoft Web Studio e como utilizá-la em diversas partes da construção de sua aplicação, como por exemplo:

- Propriedades dinâmicas na construção das telas
- Lógica de background nas telas
- Planilhas de Scheduler
- Planilhas matemáticas
- Comando de Botões

A linguagem matemática InduSoft Web Studio procura facilitar ao máximo e agilizar a programação de Scripts. Por isso utilizamos o conceito de uma planilha com duas colunas: **Tag Name** e **Expression**:

- **Tag Name**: tag que recebe o valor do cálculo da coluna **Expression**.
- **Expression**: expressões matemáticas criadas a partir das funções da linguagem InduSoft Web Studio

Exemplo:

	Tag Name	Expression
1	a	Temperatura - Nivel

O tag **a** recebe o resultado do cálculo da expressão **Temperatura - Nivel**



IMPORTANTE:

- Diferentemente de outras linguagens baseadas em textos estruturados, não se faz atribuição de valores na coluna **Expression**.
- Se você digitar A=2 nesta coluna, simplesmente está comparando o tag **A** ao valor numérico 2. O resultado dessa expressão, 1 para verdadeiro ou 0 para falso, será carregado no tag que estiver na coluna **Tag Name**.
- Se desejar carregar o valor **2** no tag **A**, apenas digite **A** na coluna **TagName** e **2** na coluna **Expression**.



NOTA:

- O InduSoft Web Studio não é sensível a letras maiúsculas ou minúsculas. Por tanto **Nivel** é igual a **NIVEL**
- Para poder fazer comentários nas linhas de programa utilize os caracteres de barras “//” no inicio do comentário a ser feito.

Tipos de dados

Números Inteiros: 1, 23, 45, -123

Ponto Flutuante: 1.234, 3.1416, -775.344

Números inteiros hexadecimais: 0x5, 0xA0, 0xBC4.

Strings: “demo”, “new demo”

Números inteiros são números de 32 bits, pontos flutuantes de 8 bytes e strings suportam até 255 caracteres.

Exemplos para acessar o application database

Para ler um valor corrente do tag no database, use o nome tag diretamente.

Exemplo 1:

Na figura abaixo, o tag X recebe a soma de 2 tags, *Temperatura* e *Nível*:

	Tag Name	Expression
1	x	Temperatura + nivel

Exemplo 2:

InduSoft Web Studio permite leitura e escrita de tags indiretamente através de apontadores (ponteiros ou *pointers*). Um tag usado como ponteiro para outro tag pode ser criado de duas formas: como qualquer tag **string** (um ponteiro para qualquer tipo de tag) ou como um ponteiro para um tipo específico de tag.

Na figura abaixo, @pointer_to_integer é um tag que é um ponteiro para valores inteiros.

Application Tags				
	Name	Array Size	Type	Description
1	EstadoEnche	0	Integer	(Válvula que enche os tanques)
2	@pointer_to_integer	0	Integer	(// Ponteiro para valor inteiro)

O InduSoft Web Studio suporta as seguintes operações.

Operando Aritméticos

- + adição
- subtração
- * multiplicação
- / divisão
- > maior que

< menor que
= igual a
>= maior que ou igual a
<= menor que ou igual a
<> diferente de

Operandos Lógicos

AND E lógico
NOT NOT lógico.
OR OU lógico
XOR OU EXCLUSIVO lógico
& AND, bit a bit.
| OR, bit a bit.
~ NOT, bit.
^ XOR, bit a bit
>> rotate right - rotaciona n bits para a direita.
<< rotate left - rotaciona n bits para a esquerda

Usando Funções

As funções do InduSoft Web Studio seguem a seguinte estrutura de sintaxe:

- **num[Name]:** Tag do tipo Numérico ou valor
- **str[Name]:** Tag do tipo String ou valor
- **tag[Name]:** Nome do Tag
- **optNum[Name]:** Tag do tipo Numérico Opcional ou valor
- **optStr[Name]:** Tag do tipo String Opcional ou valor
- **optTag[Name]:** Nome do Tag Opcional

Estas sintaxes identificam o tipo do argumento requerido por cada parâmetro das funções do InduSoft Web Studio.

Lista de Funções

O InduSoft Web Studio tem mais de uma centena de funções, que podem ser usadas em toda a aplicação, abaixo estas funções estão descritas resumidamente. Dentro dos parênteses há uma descrição resumida do tipo de operação que deve ser passado como parâmetro. Exemplo: TRACE(strOutputMessage), a função TRACE envia uma string mostrada no módulo **LogWin**, chamada de mensagem de saída (Output message).

Funções de Mensagens de Log	Execução	NT/2K/XP	Win CE	Web Client
Trace()	Synchronous	✓	✓	✗

Funções Aritméticas	Execução	NT/2K/XP	Win CE	Web Client
Abs()	Synchronous	✓	✓	✓
Div()	Synchronous	✓	✓	✓
Format()	Synchronous	✓	✓	✓
GetBit()	Synchronous	✓	✓	✓
Mod()	Synchronous	✓	✓	✓
Pow()	Synchronous	✓	✓	✓
ResetBit()	Synchronous	✓	✓	✓
Round()	Synchronous	✓	✓	✓
SetBit()	Synchronous	✓	✓	✓
Sqrt()	Synchronous	✓	✓	✓
Swap16()	Synchronous	✓	✗	✓
Swap32()	Synchronous	✓	✗	✓
Trunc()	Synchronous	✓	✓	✓

Funções Estatísticas	Execução	NT/2K/XP	Win CE	Web Client
Avg()	Synchronous	✓	✓	✓
Max()	Synchronous	✓	✓	✓
Min()	Synchronous	✓	✓	✓
Rand()	Synchronous	✓	✗	✓

Funções Logarítmicas	Execução	NT/2K/XP	Win CE	Web Client
Exp()	Synchronous	✓	✓	✓
Log()	Synchronous	✓	✓	✓
Log10()	Synchronous	✓	✓	✓

Funções Lógicas	Execução	NT/2K/XP	Win CE	Web Client
False()	Synchronous	✓	✓	✓
If()	Synchronous	✓	✓	✓
True()	Synchronous	✓	✓	✓

Funções de Manipulação de Strings	Execução	NT/2K/XP	Win CE	Web Client
Asc2Str()	Synchronous	✓	✓	✓
CharToValue()	Synchronous	✓	✓	✓
CharToValueW()	Synchronous	✓	✓	✓
ClassMembersToStrVector()	Synchronous	✓	✓	✓
NCopy()	Synchronous	✓	✓	✓
Num()	Synchronous	✓	✓	✓
Str()	Synchronous	✓	✓	✓
Str2Asc()	Synchronous	✓	✓	✓
StrGetElement()	Synchronous	✓	✓	✓
StrLeft()	Synchronous	✓	✓	✓
StrLen()	Synchronous	✓	✓	✓
StrLower()	Synchronous	✓	✓	✓
StrRChr()	Synchronous	✓	✓	✓
StrRight()	Synchronous	✓	✓	✓
StrSetElement()	Synchronous	✓	✓	✓

Funções de Manipulação de Strings	Execução	NT/2K/XP	Win CE	Web Client
StrStr()	Synchronous	✓	✓	✓
StrStrPos()	Synchronous	✓	✓	✓
StrTrim()	Synchronous	✓	✓	✓
StrTrimAll()	Synchronous	✓	✓	✓
StrUpper()	Synchronous	✓	✓	✓
ValueToChar()	Synchronous	✓	✓	✓
ValueWToChar()	Synchronous	✓	✓	✓

Funções de Data & Hora	Execução	NT/2K/XP	Win CE	Web Client
ClockGetDate()	Synchronous	✓	✓	✓
ClockGetDayOfWeek()	Synchronous	✓	✓	✓
ClockGetTime()	Synchronous	✓	✓	✓
DateTime2Clock()	Synchronous	✓	✓	✓
GetClock()	Synchronous	✓	✓	✓
Hour2Clock()	Synchronous	✓	✓	✓
SetSystemDate()	Synchronous	✓	✓	✓
SetSystemTime()	Synchronous	✓	✓	✓

Funções Trigonômicas	Execução	NT/2K/XP	Win CE	Web Client
Acos()	Synchronous	✓	✓	✓
Asin()	Synchronous	✓	✓	✓
Atan()	Synchronous	✓	✓	✓
Cos()	Synchronous	✓	✓	✓
Cot()	Synchronous	✓	✓	✓
Pi()	Synchronous	✓	✓	✓
Sin()	Synchronous	✓	✓	✓
Tan()	Synchronous	✓	✓	✓

Funções de Manipulação de Telas	Execução	NT/2K/XP	Win CE	Web Client
Close()	Asynchronous	✓	✗	✗
Open()	Asynchronous	✓	✓	✓
OpenPrevious()	Asynchronous	✓	✓	✓

Funções de Segurança	Execução	NT/2K/XP	Win CE	Web Client
BlockUser()	Synchronous	✓	✓	✓
CreateUser()	Synchronous	✓	✓	✓
GetUserNames()	Synchronous	✓	✓	✓
GetUserState()	Synchronous	✓	✓	✓
RemoveUser()	Synchronous	✓	✓	✓
SetPassword()	Synchronous	✓	✓	✓
UnblockUser()	Synchronous	✓	✓	✓

Funções de Ativação de Módulo	Execução	NT/2K/XP	Win CE	Web Client
AppActivate()	Asynchronous	✓	✓	✓
AppIsRunning()	Synchronous	✓	✓	✓
AppPostMessage()	Synchronous	✓	✓	✓
AppSendKeys()	Synchronous	✓	✗	✗
CleanReadQueue()	Synchronous	✓	✗	✗
CloseSplashWindow()	Synchronous	✓	✗	✗
DisableMath()	Asynchronous	✓	✗	✗
EnableMath()	Asynchronous	✓	✗	✗
EndTask()	Asynchronous	✓	✗	✗
ExitWindows()	Asynchronous	✓	✗	✗
IsScreenOpen()	Synchronous	✓	✓	✗
IsTaskRunning()	Synchronous	✓	✗	✗
IsViewerInFocus()	Synchronous	✓	✗	✗
KeyPad()	Asynchronous	✓	✓	✓
LogOff()	Asynchronous	✓	✓	✓
LogOn()	Asynchronous	✓	✓	✓
Math()	Synchronous	✓	✓	✓

Funções de Ativação de Módulo	Execução	NT/2K/XP	Win CE	Web Client
PostKey()	Synchronous	✓	✓	✓
Recipe()	Synchronous	✓	✓	✗
Report()	Synchronous	✓	✓	✓
SendKeyObject()	Synchronous	✓	✗	✗
SetAppPath()	Synchronous	✓	✗	✗
SetKeyboardLanguage()	Synchronous	✓	✓	✓
SetViewerInFocus()	Synchronous	✓	✗	✗
SetViewerPos()	Synchronous	✓	✗	✗
ShutDown()	Synchronous	✓	✓	✗
StartTask()	Asynchronous	✓	✗	✗
ViewerPostMessage()	Asynchronous	✓	✓	✓
Wait()	Synchronous	✓	✓	✓
WinExec()	Asynchronous	✓	✓	✓
WinExecIsRunning()	Synchronous	✓	✓	✓

Funções de Manipulação de Arquivos	Execução	NT/2K/XP	Win CE	Web Client
DeleteOlderFiles()	Synchronous	✓	✓	✓
DirCreate()	Synchronous	✓	✓	✓
DirDelete()	Synchronous	✓	✓	✓
DirLength()	Synchronous	✓	✓	✓
DirRename()	Synchronous	✓	✓	✓
FileCopy()	Synchronous	✓	✓	✓
FileDelete()	Synchronous	✓	✓	✓
FileLength()	Synchronous	✓	✓	✓
FileRename()	Synchronous	✓	✓	✓
FileWrite()	Synchronous	✓	✓	✓
FindFile()	Synchronous	✓	✓	✓
FindPath()	Synchronous	✓	✓	✓
GetFileAttributes()	Synchronous	✓	✓	✓
GetFileTime()	Synchronous	✓	✓	✓
GetLine()	Synchronous	✓	✓	✓

Funções de Manipulação de Arquivos	Execução	NT/2K/XP	Win CE	Web Client
Hst2Txt()	Asynchronous	✓	✗	✗
Hst2TxtIsRunning()	Synchronous	✓	✗	✗
Print()	Asynchronous	✓	✓	✗
RDFFileN()	Synchronous	✓	✓	✓

Funções para Impressão de Telas	Execução	NT/2K/XP	Win CE	Web Client
PrintSetup()	Asynchronous	✓	✓	✓
PrintWindow()	Asynchronous	✓	✓	✗

Funções de Tradução	Execução	NT/2K/XP	Win CE	Web Client
Ext()	Synchronous	✓	✓	✓
SetTranslationFile()	Synchronous	✓	✓	✓

Funções Multimedia	Execução	NT/2K/XP	Win CE	Web Client
Play()	Synchronous/Asynchronous	✓	✓	✓

Funções de Informações do Sistema	Execução	NT/2K/XP	Win CE	Web Client
DbVersion()	Synchronous	✓	✓	✓
GetAppHorizontalResolution()	Synchronous	✓	✗	✗
GetAppPath()	Synchronous	✓	✗	✗
GetAppVerticalResolution()	Synchronous	✓	✗	✗
GetComputerIP()	Synchronous	✓	✓	✓
GetComputerName()	Synchronous	✓	✗	✓
GetHardKeyModel()	Synchronous	✓	✗	✗
GetHardKeySN()	Synchronous	✓	✗	✗
GetIPAll()	Synchronous	✓	✓	✓
GetNetMACID()	Synchronous	✓	✗	✗
GetOS()	Synchronous	✓	✓	✓
GetPrivateProfileString()	Synchronous	✓	✓	✓
GetProductPath()	Synchronous	✓	✓	✓
GetServerHostName()	Synchronous	✓	✗	✗
GetTickCount()	Synchronous	✓	✓	✓

Funções de Informações do Sistema	Execução	NT/2K/XP	Win CE	Web Client
InfoAppAlrDir()	Synchronous	✓	✓	✓
InfoAppHSTDir()	Synchronous	✓	✓	✓
InfoDiskFree()	Synchronous	✓	✗	✓
InfoResources()	Synchronous	✓	✓	✓
IsActiveXReg()	Synchronous	✓	✗	✗
NoInputTime()	Synchronous	✓	✗	✗
ProductVersion()	Synchronous	✓	✓	✓
SaveAlarmFile()	Synchronous	✓	✓	✗
SetAppAlarmPath()	Synchronous	✓	✓	✗
SetAppHSTPath()	Synchronous	✓	✓	✓
SetDateFormat()	Synchronous	✓	✓	✓
SetWebConfig()	Synchronous	✓	✓	✗

Funções de Acesso a Banco de Dados	Execução	NT/2K/XP	Win CE	Web Client
ForceTagChange()	Synchronous	✓	✓	✓

Funções de Loop	Execução	NT/2K/XP	Win CE	Web Client
For() & Next	NA	✓	✓	✗

Funções ODBC	Execução	NT/2K/XP	Win CE	Web Client
ODBCBeginTrans()	Synchronous	✓	✗	✗
ODBCBindCol()	Synchronous	✓	✗	✗
ODBCCanAppend()	Synchronous	✓	✗	✗
ODBCCanTransact()	Synchronous	✓	✗	✗
ODBCCanUpdate()	Synchronous	✓	✗	✗
ODBCClose()	Synchronous	✓	✗	✗
ODBCCommitTrans()	Synchronous	✓	✗	✗
ODBCDelete()	Synchronous	✓	✗	✗
ODBCExecutesQL()	Synchronous	✓	✗	✗
ODBCInsert()	Synchronous	✓	✗	✗
ODBCIsBOF()	Synchronous	✓	✗	✗
ODBCIsDeleted()	Synchronous	✓	✗	✗

Funções ODBC	Execução	NT/2K/XP	Win CE	Web Client
ODBCIsEOF()	Synchronous	✓	✗	✗
ODBCIsFieldNull()	Synchronous	✓	✗	✗
ODBCIsFieldNullable()	Synchronous	✓	✗	✗
ODBCMmove()	Synchronous	✓	✗	✗
ODBCMmoveFirst()	Synchronous	✓	✗	✗
ODBCMmoveLast()	Synchronous	✓	✗	✗
ODBCMmoveNext()	Synchronous	✓	✗	✗
ODBCMmovePrev()	Synchronous	✓	✗	✗
ODBCOpen()	Synchronous	✓	✗	✗
ODBCQuery()	Synchronous	✓	✗	✗
ODBCRollback()	Synchronous	✓	✗	✗
ODBCSetFieldNull()	Synchronous	✓	✗	✗
ODBCSetFilter()	Synchronous	✓	✗	✗
ODBCSetSort()	Synchronous	✓	✗	✗
ODBCUnbindCol()	Synchronous	✓	✗	✗
ODBCUpdate()	Synchronous	✓	✗	✗

Funções de E-mail	Execução	NT/2K/XP	Win CE	Web Client
CNFEmail()	Synchronous	✓	✓	✓
GetStatussendEmailExt()	Synchronous	✓	✓	✓
SendEmail()	Synchronous	✓	✓	✓
SendEmailExt()	Asynchronous	✓	✓	✓

Funções Dial-Up	Execução	NT/2K/XP	Win CE	Web Client
DialError()	Synchronous	✓	✓	✗
DialGetClientIP()	Synchronous	✓	✓	✗
DialGetServerIP()	Synchronous	✓	✗	✗
DialStatus()	Synchronous	✓	✓	✗
DialUp()	Asynchronous	✓	✓	✗
DialUpToCE()	Asynchronous	✓	✗	✗
FindAllDevices()	Synchronous	✓	✓	✗

Funções Dial-Up	Execução	NT/2K/XP	Win CE	Web Client
FindModem()	Synchronous	✓	✓	✗
HangUp()	Synchronous	✓	✓	✗
PhoneDialUp()	Asynchronous	✓	✗	✗
PhoneDisableListen()	Synchronous	✓	✗	✗
PhoneEnableListen()	Synchronous	✓	✗	✗
PhoneHangUp()	Asynchronous	✓	✗	✗
PhoneStatus()	Synchronous	✓	✗	✗

Funções de ActiveX	Execução	NT/2K/XP	Win CE	Web Client
XGet ()	Asynchronous	✓	✓	✓
XRun ()	Asynchronous	✓	✓	✓
XSet ()	Asynchronous	✓	✓	✓

Funções de Log de Eventos	Execução	NT/2K/XP	Win CE	Web Client
SendEvent ()	Synchronous	✓	✓	✓

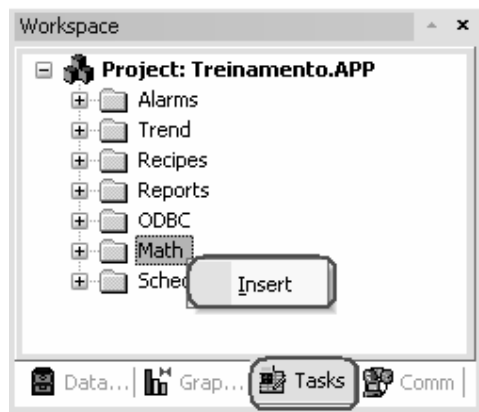
Funções de FTP	Execução	NT/2K/XP	Win CE	Web Client
CNFFtp ()	Asynchronous	✓	✓	✓
FtpGet ()	Asynchronous	✓	✓	✓
FtpPut ()	Asynchronous	✓	✓	✓
FtpStatus ()	Asynchronous	✓	✓	✓

Funções de Banco de Dados	Execução	NT/2K/XP	Win CE	Web Client
SyncAlarm ()	Asynchronous	✓	✓	✓
SyncAlarmStatus ()	Asynchronous	✓	✓	✓
SyncEvent ()	Asynchronous	✓	✓	✓
SyncEventStatus ()	Asynchronous	✓	✓	✓
SyncTrend ()	Asynchronous	✓	✓	✓
SyncTrendStatus ()	Asynchronous	✓	✓	✓

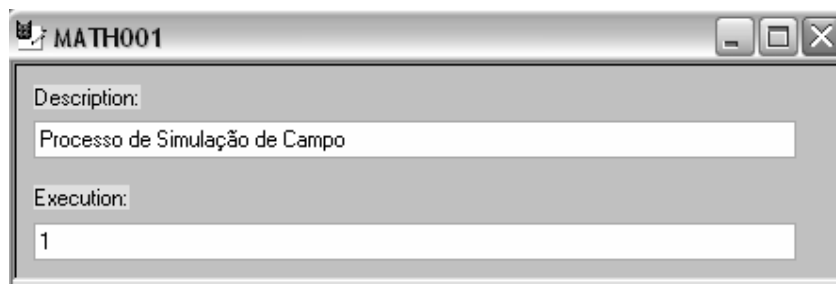
Configurando Planilhas (Worksheets)

Configurando uma Planilha Matemática (Math Worksheet) para simular valores de processo

- Clique com o botão direito na pasta **Math** localizada no tab **Tasks**. Clique em **Insert** para criar uma nova planilha matemática, como mostra a figura abaixo.



- Ao fazer isso, surgirá a figura abaixo:



- O campo **Description** determina o nome da Planilha Matemática que será apresentado no Workspace.
- O campo **Execution** controla a execução da planilha. Pode-se digitar um valor numérico, um tag, uma expressão ou uma função que, resultando verdadeira (diferente de zero), executa a planilha matemática.
- Portanto o campo **Execution** será preenchido com o valor **1**. Isto habilita a execução contínua desta planilha uma vez que **1** será sempre um valor verdadeiro (TRUE). Como na figura abaixo.
- No corpo da planilha, estaremos simulando:
 - O estado das válvulas, de acordo com o comando criado.
 - O Nível, a Pressao e a Temperatura dos três Tanques.



IMPORTANTE:

- Crie um novo tag chamado de **K**, do tipo **Integer** com **Array Size 0**

- Para simular o estado de cada válvula, vamos transferir o valor do Comando para o Estado, uma vez que não estamos numa aplicação real.
- Para os valores de temperatura e pressão utilizaremos as funções trigonométricas de seno e cosseno (sin e cosin).
- Para simular a propriedade **Nível** de cada Tanque, usaremos os estados das válvulas de Encher e Esvaziar para incrementar ou decrementar o valor da variável **Nível**.
- Utilizaremos a função de Laço – **FOR/NEXT** – para acessar todas as posições do vetor (tanques 1, 2 e 3), sem precisar digitar repetidamente as simulações para os três tanques.
- Veja na segunda listagem, como poderia ser a mesma configuração sem a utilização da função FOR().



IMPORTANTE:

- Configure a sua planilha matemática como mostrado abaixo:

Tag Name	Expression
K	FOR (1 , 3 , 1)
EstadoEnche[K]	ComandoEnche[K]
EstadoEsvazia[K]	ComandoEsvazia [K]
Temperatura[K]	(Sin ((Second / (K * 10)) * PI()) + 1) * 50
Pressao[K]	(Cos ((Second / (K * 10)) * PI()) + 1) * 50
Nivel[K]	if ((Not EstadoEsvazia[K] and EstadoEnche[K]) and Nivel[K] < 100, Nivel[K] + 1)
Nivel[K]	if ((Not EstadoEnche[K] and EstadoEsvazia[K]) and Nivel[K] > 0, Nivel[K] - 1)
NEXT	// Comentário: Retorna para a linha da função FOR()



IMPORTANTE:

- A planilha a seguir **NÃO** deve ser configurada, serve somente como exemplo.

Sem a utilização da função FOR():

Tag Name	Expression
EstadoEnche[1]	ComandoEnche[1]
EstadoEsvazia[1]	ComandoEsvazia [1]
EstadoEnche[2]	ComandoEnche[2]
EstadoEsvazia[2]	ComandoEsvazia [2]

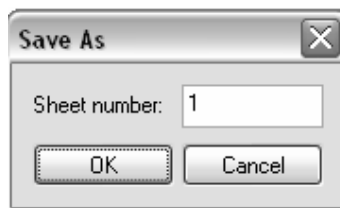
EstadoEnche[3]	ComandoEnche[3]
EstadoEsvazia[3]	ComandoEsvazia [3]
Temperatura[1]	$(\sin((\text{Second}/10) * \pi) + 1) * 50$
Temperatura[2]	$(\sin((\text{Second}/20) * \pi) + 1) * 50$
Temperatura[3]	$(\sin((\text{Second}/30) * \pi) + 1) * 50$
Pressao[1]	$(\cos((\text{Second}/10) * \pi) + 1) * 50$
Pressao[2]	$(\cos((\text{Second}/20) * \pi) + 1) * 50$
Pressao[3]	$(\cos((\text{Second}/30) * \pi) + 1) * 50$
Nivel[1]	if ((Not EstadoEsvazia[1] and EstadoEnche[1]) and Nivel[1] < 100, Nivel[1] + 1)
Nivel[1]	if ((Not EstadoEnche[1] and EstadoEsvazia[1]) and Nivel[1] > 0, Nivel[1] - 1)
Nivel[2]	if ((Not EstadoEsvazia[2] and EstadoEnche[2]) and Nivel[2] < 100, Nivel[2] + 1)
Nivel[2]	if ((Not EstadoEnche[2] and EstadoEsvazia[2]) and Nivel[2] > 0, Nivel[2] - 1)
Nivel[3]	if ((Not EstadoEsvazia[3] and EstadoEnche[3]) and Nivel[3] < 100, Nivel[3] + 1)
Nivel[3]	if ((Not EstadoEnche[3] and EstadoEsvazia[3]) and Nivel[3] > 0, Nivel[3] - 1)



IMPORTANTE:

- Salve e Feche a Planilha

- Para salvar a Planilha clique em File, Save As.
- Salve como 1.



- Clique em OK e Feche a Planilha.
- Agora já podemos rodar a aplicação usando o ícone **Run Application** e verificar o comportamento do nosso processo.



Grupo de Alarmes (Alarms Group)

Criando uma planilha

- Para criar um grupo de alarmes, vamos ao tab **Tasks**, e clique com o botão direito na pasta **Alarm**, em seguida clique no botão **Insert**.



- Configure a planilha como mostrado abaixo:

ALARM001.ALR

Group Name: Indusoft Description: Simulação de Alarmes

Disable: Remote Ack: Email Settings...

Total Alarms: Total Unack: Dead Band Time (sec.) Active: Norm:

Display/Save
☒ Summary ☐ To Printer
☒ Ack ☒ Save To Disk
☐ Beep ☒ Generate Ack Messages
☒ Generate Norm Messages

Colors
☒ Enable
 Start FG BG
 Ack FG BG
 Norm FG BG

	Tag Name	Type	Limit	Message	Priority	Selector
1	Nivel[1]	LoLo	10.000000	Nivel Baixo no Tanque 1	1	a
2	Nivel[1]	HiHi	80.000000	Nivel Alto no Tanque 1	1	a
3	Nivel[2]	Lo	27.000000	Nivel Baixo no Tanque 2	2	b
4	Nivel[2]	Hi	67.000000	Nivel Alto no Tanque 2	2	b
5	Nivel[3]	Lo	2.000000	Nivel Baixo no Tanque 3	3	c
6	Nivel[3]	Hi	19.000000	Nivel Alto no Tanque 3	3	c
7						
8						
9						

- Criando esta planilha, você está informando ao sistema quais tags devem ter a função de alarmes, que tipo de alarmes, quais os seus limites, as mensagens para tais, prioridades e filtros (dados)
- O check box **Disk** permite escolher salvar o histórico destes alarmes em disco, em formato ASCII na pasta *alarm* da aplicação.
- O manual explica o Header (cabeçalho), da planilha de Alarme é explicada abaixo:

ALARM WORKSHEET HEADER

Define algumas características em comum para todos os alarmes.

CAMPO GROUP NAME

Nome utilizado para distinguir os grupos de alarme.



IMPORTANTE:

- Antes de mudar o conteúdo do campo Group Name salve a configuração ou então os dados não salvos podem ser perdidos

CAMPO DESCRIPTION

Ajuda na descrição do grupo.

CAMPO DISABLE

É um tag que, se seu valor for maior do que zero, os alarmes serão desabilitados.

CAMPO REMOTE ACK

Tag que permite o reconhecimento dos alarmes deste grupo.3

CAMPO TOTAL ALARMS

Neste campo um tag receberá o número de alarmes ainda ativos, mesmo que não reconhecidos.

CAMPO TOTAL ALARMS OU TOTAL UNACK

O tag deste campo receberá o número de alarmes que ainda não foram reconhecidos. Como estes alarmes podem não estar mais ativo, ou apenas não reconhecido, isto, justifica o nome do campo.

Group Box

Summary Check-box – Quando selecionada permite que estes alarmes sejam mostrados em um objeto de alarme.



IMPORTANTE:

- Se esta opção **Summary** não estiver escolhida, os alarmes deste grupo não serão mostrados nos grupos de alarme. Pode ser útil para o controle de eventos.

Check-box Ack– Permite o reconhecimento do alarme. Alguns alarmes, usados apenas por alerta, não precisam de reconhecimento. Só é habilitado se o campo **Summary** também estiver habilitado.

Check-box Beep– Faz soar o Bip do computador quando há alarmes não reconhecidos. Disponível apenas se **Ack** e **Summary** estiverem habilitados.

Check-box Printer– Envia o alarme para uma impressora. Não tem sido mais utilizado, pois as impressoras de hoje não permitem a impressão linha a linha, e só página a página. Já pensou se acontecer 5 alarmes simultâneos e forem impressas 5 páginas de uma linha?

Check-box Save to Disk– Habilita a gravação dos alarmes em disco, permitindo o seu uso em objetos de alarme histórico. Se não for escolhido não será registrado nenhum alarme desta planilha em disco.

Check-box Generate Ack Messages e **Check-box Generate Norm Message**– Se a opção **Disk** estiver habilitada, permite decidir que se grave em disco as informações de quando e se o alarme ocorrido foi reconhecido e/ou normalizado .

Group Box Colors– Define as cores dos alarmes no objeto de alarme, tanto da fonte quanto do seu background (fundo). Só tem efeito se a opção *radio* estiver em **Custom**. No caso de **Default**, a cor do alarme que será mostrada será a mesma que configurada no objeto de alarme da tela

Criando a tela de Alarm on-line

- Abra a tela *Standard* e desenhe um objeto de **Alarm** como se fosse um botão, utilizando o ícone



Alarm / Event Control

#####	#####	Alarme Histórico	Alarme OnLine	Relatórios	Principal
Usuário: #####		Trend Histórico	Alarme Histórico	Receitas	Sair

Active Time	Tagname	Message
03/25/2004 11:11:35	TTTTTTTTT	MMMMMMMMMMMMMMMMMM...
03/25/2004 11:11:35	TTTTTTTTT	MMMMMMMMMMMMMMMMMM...
03/25/2004 11:11:35	TTTTTTTTT	MMMMMMMMMMMMMMMMMM...
03/25/2004 11:11:35	TTTTTTTTT	MMMMMMMMMMMMMMMMMM...

Reconhecer Um
Reconhecer Todos

- Duplo-clique no objeto de **Alarm** e configure a janela **Object Properties** como mostrado abaixo.
- Esteja certo de que a opção **On-line** está selecionada.
- Clique no botão **Filters** que possui alguns itens a serem configurados.

Object Properties

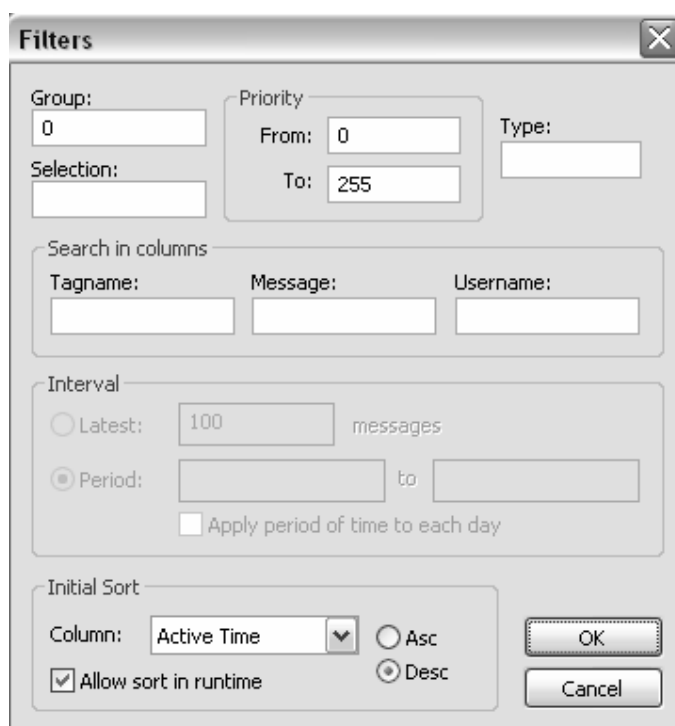
Replace... Hint: Alarm/Event Control

Type: Alarm Online

Font Filters Win: E-Sign

Columns Advanced

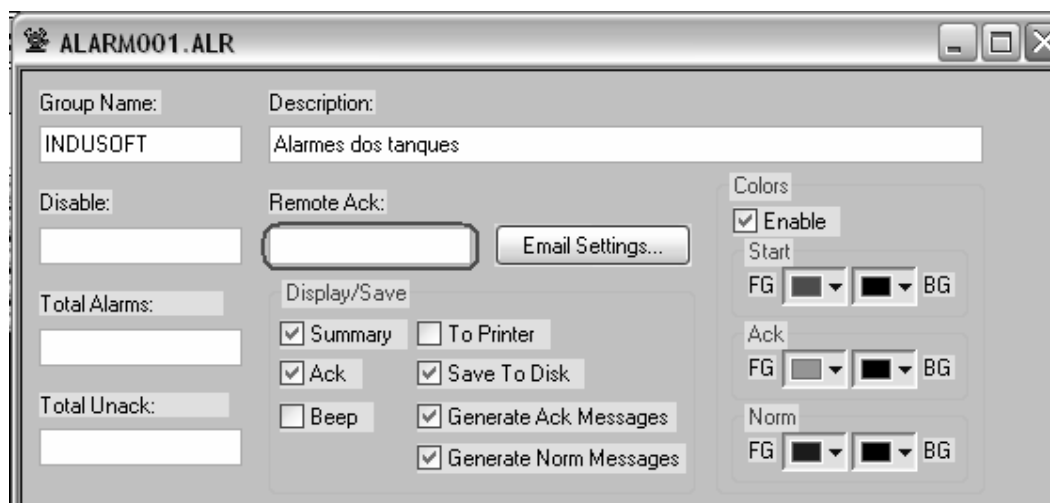
☐ Show gridlines ☒ Show header ☒ Ext translation VK: <Use Default>



The 'Filters' dialog box contains the following fields and controls:

- Group:** Text box with value '0'.
- Priority:** Section with 'From: 0' and 'To: 255' text boxes.
- Type:** Text box.
- Selection:** Text box.
- Search in columns:** Section with 'Tagname:', 'Message:', and 'Username:' labels above three text boxes.
- Interval:** Section with two radio buttons: 'Latest:' (unselected) and 'Period:' (selected). The 'Period:' option has two text boxes for a range and a checkbox 'Apply period of time to each day'.
- Initial Sort:** Section with 'Column:' (dropdown menu showing 'Active Time'), 'Asc' (radio button, unselected), 'Desc' (radio button, selected), and a checkbox 'Allow sort in runtime' (checked).
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

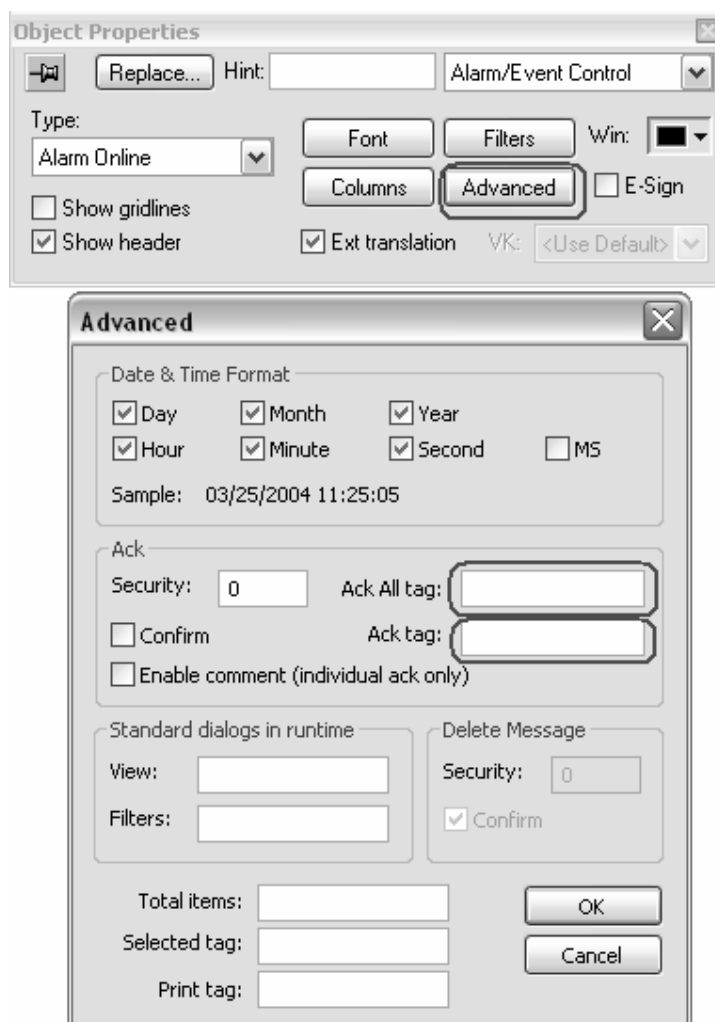
- Estudaremos estes detalhes quando falarmos dos alarmes históricos.
- Existem duas maneiras de se reconhecer os Alarmes.
- Trigando o tag que estaria no campo **Remote Ack**, da planilha de alarmes:



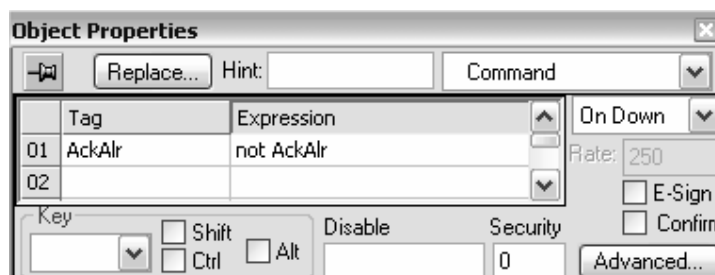
The 'ALARM001.ALR' dialog box contains the following fields and controls:

- Group Name:** Text box with value 'INDUSOFT'.
- Description:** Text box with value 'Alarmes dos tanques'.
- Disable:** Text box.
- Remote Ack:** Text box with a red border, and an 'Email Settings...' button next to it.
- Colors:** Section with 'Enable' (checked checkbox), and three color selection groups for 'Start', 'Ack', and 'Norm'. Each group has 'FG' and 'BG' color pickers.
- Display/Save:** Section with checkboxes for 'Summary' (checked), 'To Printer' (unchecked), 'Ack' (checked), 'Save To Disk' (checked), 'Beep' (unchecked), 'Generate Ack Messages' (checked), and 'Generate Norm Messages' (checked).
- Total Alarms:** Text box.
- Total Unack:** Text box.

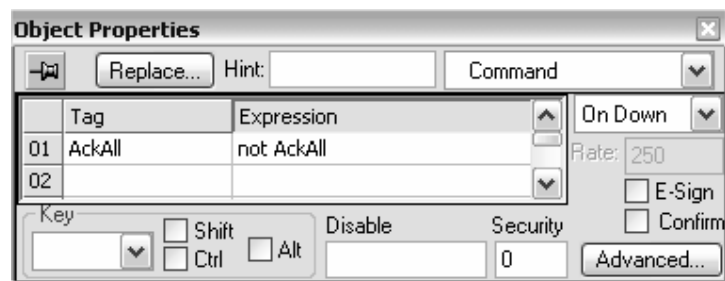
- Ou trigando um tag que estaria no campo **Ack Tag** dentro do Object Properties/Advanced no objeto de alarme.



- Mas ainda há outra maneira. Há três tags internos que estudaremos agora: **Alarm**, **AckAlr** e **AckAll**.
- O tag **AckAlr** reconhece o alarme que está ativo no Sistema.
- Trigando **AckAlr** este alarme é reconhecido.
- Já o tag interno **AckAll** reconhece todos os alarmes não reconhecidos, independentes do grupo que eles pertencem.
- Para efeito de uso neste tutorial, usaremos estes tags internos.
- Clique duas vezes no botão **Reconhece Um** e utilize a opção **Command** para configurar o tag interno **AckAlr** como mostrado abaixo:



- Clique duas vezes no botão **Reconhecer Todos** e configure o tag interno **AckAll** como mostrado abaixo:.



IMPORTANTE:

- Salve a tela com o nome de *AlarmeOnLine* e feche-a, em seguida rode a aplicação

Criando a tela de Alarmes Históricos (Historical Alarm Screen)



IMPORTANTE:

- Crie os tags indicados como segue:

Application Tags					
	Name	Array Size	Type	Description	Web Da
9	Pressao	3	Real	Pressão do tanque	Server
10	AlarmIni	0	String	Data Inicial dos Alarmes Históricos	Server
11	AlarmFin	0	String	Data Final dos Alarmes Históricos	Server
12	AlarmSel	0	String	Tag de Seleção dos Alarmes Históricos	Server

- Abra a tela **Standard** e desenhe um objeto de **Alarm** como se fosse um botão, utilizando o ícone



Alarm / Event Control e configure como abaixo.

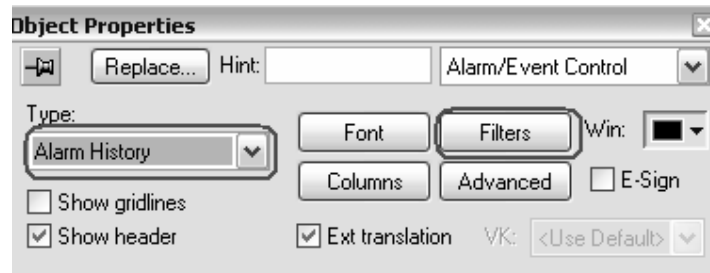
#####	#####	Alarme Histórico	Alarme OnLine	Relatórios	Principal
Usuário: #####		Trend Histórico	Alarme Histórico	Receitas	Sair

Active Time	Tagname	Message
03/25/2004 12:15:54	TTTTTTTTT	MMMMMMMMMMMMMMMMMM...
03/25/2004 12:15:54	TTTTTTTTT	MMMMMMMMMMMMMMMMMM...
03/25/2004 12:15:54	TTTTTTTTT	MMMMMMMMMMMMMMMMMM...
03/25/2004 12:15:54	TTTTTTTTT	MMMMMMMMMMMMMMMMMM...

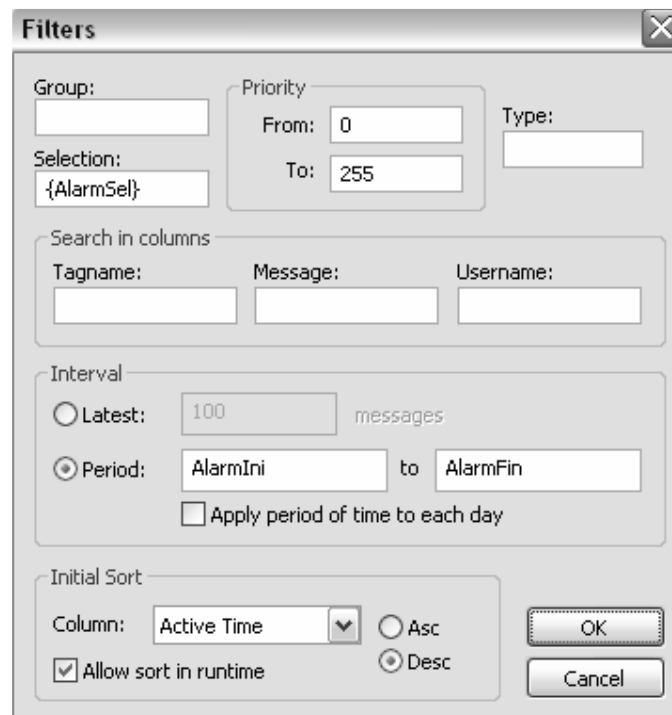
Seleção : ##	Data Inicial : #####	Data Final : #####
--------------	----------------------	--------------------

- Esta tela de histórico dos alarmes mostrará todos os alarmes que aconteceram e foram gravados em disco, de acordo com um período definido como parâmetro.

- Clique duas vezes no objeto de alarmes e configure a janela **Object Properties** como mostrado abaixo.
- Não se esqueça de mudar a opção de **On-line** para **History**



- Pressione o botão **Filters** e vamos configurar como mostrado abaixo



IMPORTANTE:

- Note o campo Selection: o tag **AlarmSel** foi configurado entre chaves { }. Isso acontece em todo campo que pode ser preenchido com uma constante string. Se você deseja configurar com um tag, este deve estar entre chaves: **{AlarmSel}**.

Agora vamos configurar as entradas de texto para seleções e filtros por data.



IMPORTANTE:

- Todos os textos ##### devem conter a propriedade **Text I/O**, e para permitir a entrada de valor no tag devem estar com o **Check Box “ Input Enabled” habilitado**.
- Configure os demais campos da mesma forma, utilizando na sequência os tags **AlarmIni, AlarmFim e AlarmSel**.
- Salve a tela como **AlarmeHistórico** e execute a aplicação.



IMPORTANTE:

- Salve a tela como **AlarmeHistórico** e execute a aplicação.

Os Tag Fields de Alarmes

Alguns dos Tag fields (TagName->Field) são diretamente relacionados a alarmes. Os limites dos alarmes, por exemplo, podem ser mudados dinamicamente escrevendo um novo valor no tag field. Veja abaixo os tag fields relacionados aos alarmes (* significa que podem ser alterados dinamicamente)

***HiHiLimit**
***HiLimit**
***LoLimit**
***LoLoLimit**
***DevLimit**
***RateLimit**
***DevSetpoint**

Exemplo: configuração utilizada no Runtime: TP->HiHiLimit=70

***AlrDisable** Desabilita o alarme:

1 Desabilita alarme

0 Habilita alarme

Exemplo: TMP->AlrDisable=1

HiHi Se maior que zero, há alarme tipo HiHi

Hi Se maior que zero, há alarme tipo Hi

Lo Se maior que zero, há alarme tipo Low

LoLo Se maior que zero, há alarme tipo LowLow

Rate Se maior que zero, há alarme tipo Rate

Dev Se maior que zero, há alarme tipo Deviation

Trend

A tarefa **Trend** armazena amostras de determinadas variáveis da aplicação, a intervalos de tempo configuráveis. Você pode armazená-las em disco e simultaneamente apresentá-las em forma de gráficos. Vamos criar objetos para apresentar gráficos de tendências na tela tanto **On-Line** (que é atualizado a cada X segundos com os novos valores das variáveis como se fosse um registrador gráfico), quanto **Histórico** (que inclui recuperar dados dos arquivos gravados).

Trend On line

Nosso Trend On Line mostrará algumas variáveis e será atualizado uma vez por segundo.

- Abra a tela **Standard.scr**.



- Clique no ícone mostrado ao lado e desenhe um objeto de **Trend**, como se fosse um botão e configure-o com mostrado a seguir:

#####	#####	Alarme Histórico	Alarme OnLine	Relatórios	Principal
Usuário: #####	Trend Histórico	Alarme Histórico	Receitas	Sair	

###

###

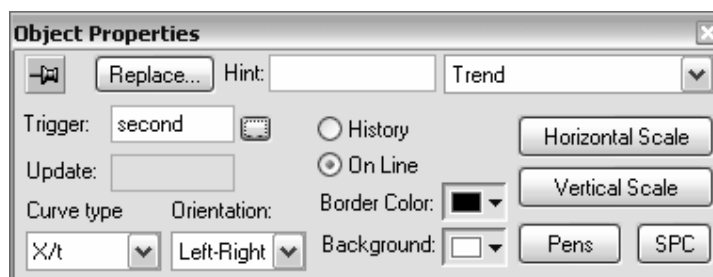
###

###

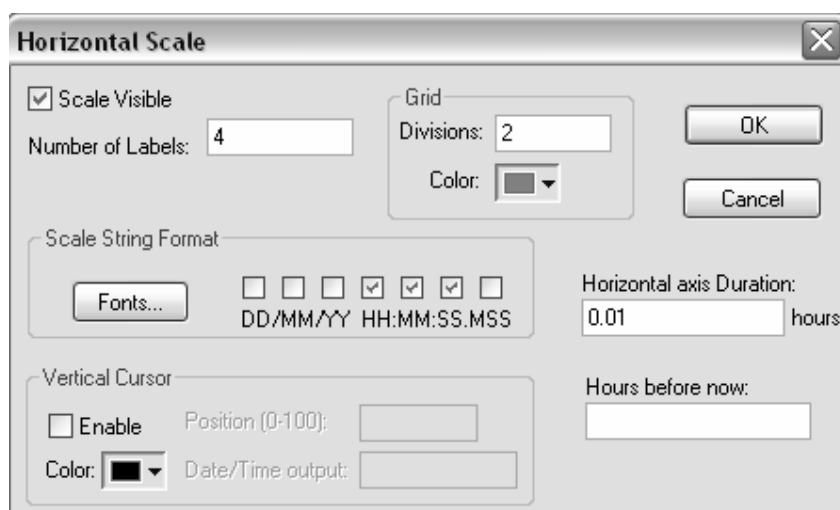
###

###:###:##
##:###:##
###:###:##
###:###:##

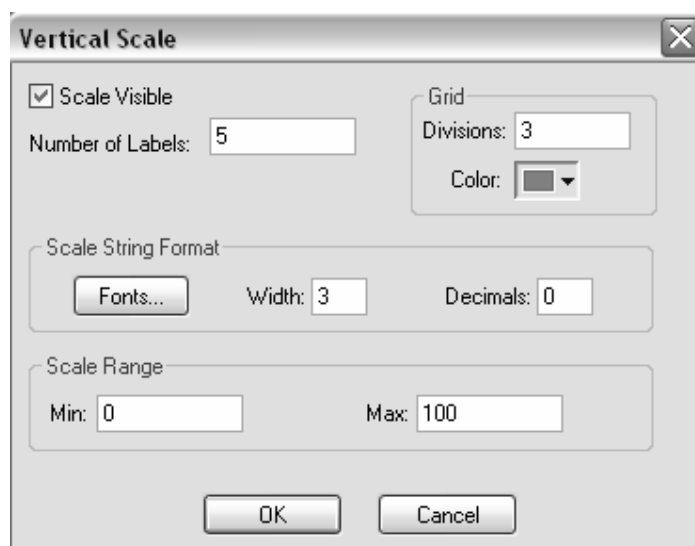
- Clique duas vezes no objeto Trend recém desenhado (tela branca) e configure a janela Object Properties. Na tela inicial, nenhuma alteração deve ser feita. As configurações default atendem nossa aplicação.



- Clique no botão **Horizontal Scale** e edite como mostrado abaixo:



- Clique em **OK** . Agora clique no botão **Vertical Scale** e configure como mostrado a seguir:



- Clique em **OK** . Agora clique no botão Pens e edite como mostrado abaixo:

Tag	Line	Min	Max	Cursor value
Temperatura[1]		0	100	
Temperatura[2]		0	100	
Temperatura[3]		0	100	

OK Cancel



IMPORTANTE:

- Salve a tela como **TrendOnLine** e execute a aplicação.

Criando um gráfico de tendências histórico (Trend Histórico)

Nesta parte do tutorial mostraremos vários recursos ao mesmo tempo, incluindo tags indiretos e outras operações usadas nos controles do gráfico.

Criando um Grupo de Trend

A princípio criaremos tags que controlarão o objeto de Trend histórico.



IMPORTANTE:

- Crie os tags abaixo.

13	Lalto	0	Integer	▼	Limite alto do gráfico	Server	▼
14	Lbaixo	0	Integer	▼	Limite baixo do gráfico	Server	▼
15	Duracao	0	Real	▼	Tamanho da janela do gráfico - em Horas	Server	▼
16	Dataini	0	String	▼	Data do início do gráfico	Server	▼
17	Tempoini	0	String	▼	Hora de início do gráfico	Server	▼
18	Pcursor	0	Real	▼	Posição do cursor no gráfico	Server	▼
19	CursorPen	3	Real	▼	Valor da Interposição curva/cursor	Server	▼
20	Gravar	0	Boolean	▼	Trigger de gravação das amostras	Server	▼
21	Scursor	0	String	▼	Saída de data/hora do cursor	Server	▼
22	Pena	3	String	▼	Apontador para tag traçado na pena	Server	▼

- Agora informaremos ao sistema quais os tags de processo queremos amostrar seus valores em disco, e de quanto em quanto tempo.
- Clique com o botão direito na pasta **Trend** (tab "Tasks") e clique em **Insert**.



- Configure a planilha como mostrada a seguir.
- Assim você está preparando o sistema para salvar os valores dos tags de temperatura dos Tanques, toda vez que o tag **Second** mudar de valor, e guardando isso em disco por até 700 dias.

	Tag Name	Dead Band	
1	Temperatura[1]		
2	Temperatura[2]		
3	Temperatura[3]		
4			
5			

- O check box **Save on Trigger** define a taxa de gravação.



IMPORTANTE:

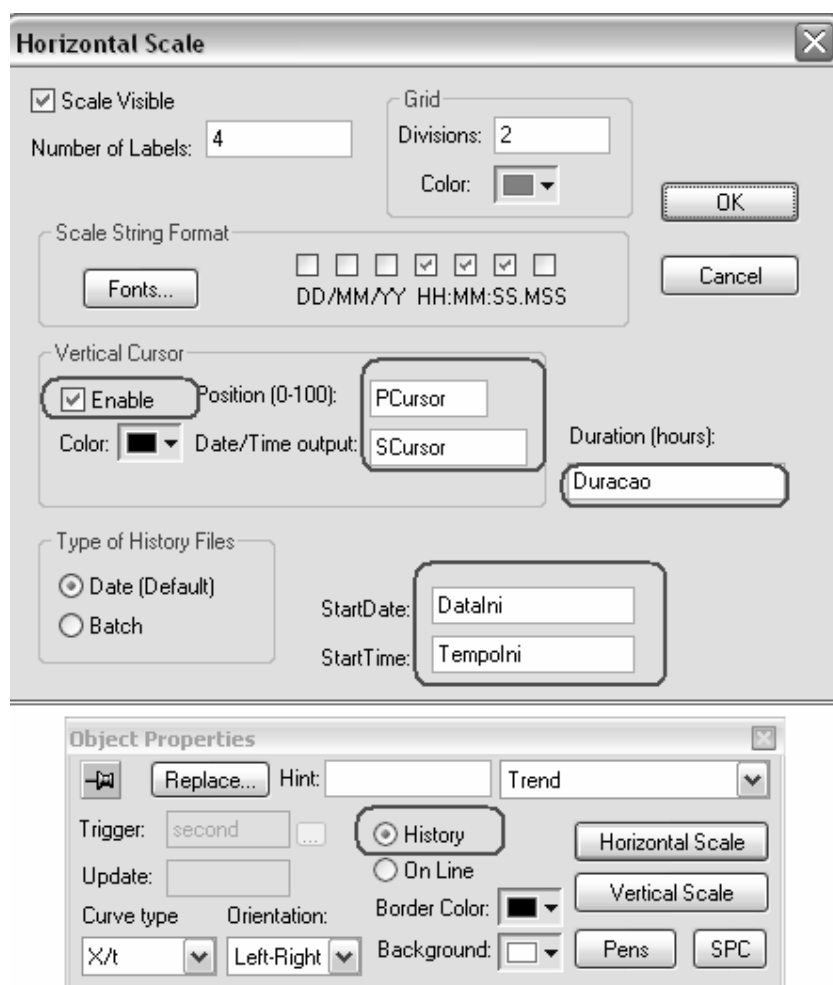
- O módulo do InduSoft Web Studio que controla a execução das funções matemáticas, bem como de outras tarefas de Runtime, como Alarmes, Trends, etc... é o **BackGround Tasks**. Se uma função matemática aparenta não funcionar verifique se este módulo está rodando. Se ele estiver, um ícone deve estar no canto direito inferior da tela (Status Bar) do Windows NT (onde fica normalmente o relógio).



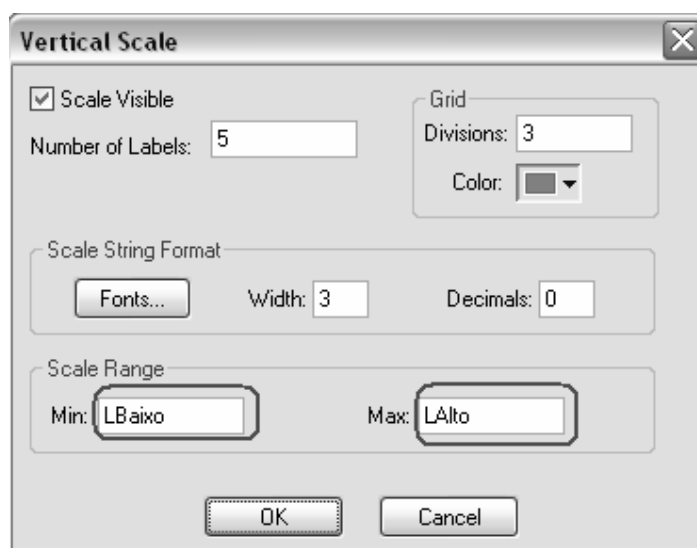
Criando uma tela Trend Histórico

Nesta tela de histórico criaremos alguns campos para entrada de valores como data inicial do gráfico, hora inicial, duração, etc... O gráfico estará apresentando um cursor vertical que, sempre em intersecção com as curvas, indicará o valor daquela curva naquele momento. Além disso, mostraremos as penas com as opções de escondê-las ou não, utilizando-se de tags indiretos para tal.

- Utilize a tela do **TrendOnline** e modifique a configuração do objeto Trend como mostra a figura que segue.
- Atenção para a modificação na primeira janela de Object Properties, onde se seleciona Trend History.



- Clique OK e passe à configuração de **Vertical Scale**. Configure como segue:




Vertical Scale

☒ Scale Visible

Number of Labels: 5

Grid Divisions: 3

Color: 

Scale String Format

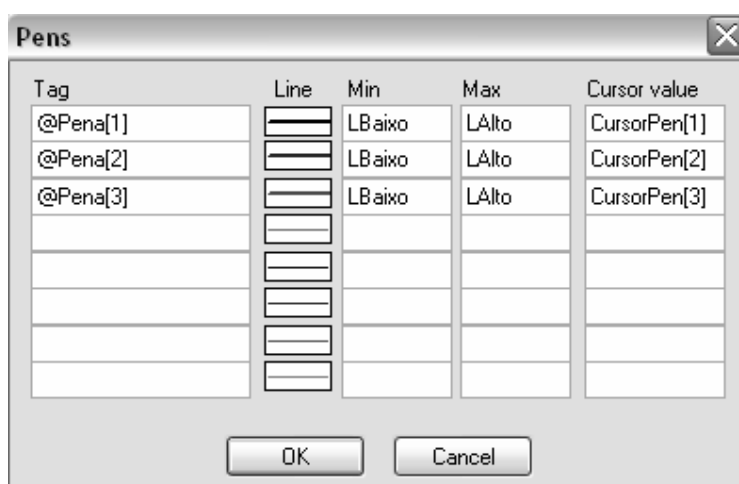
Fonts... Width: 3 Decimals: 0

Scale Range










Min: LBaixo Max: LAlto

OK Cancel

- Clique OK e passe à configuração de **Pens**. Configure como segue:



Pens

Tag	Line	Min	Max	Cursor value
@Pena[1]		LBaixo	LAlto	CursorPen[1]
@Pena[2]		LBaixo	LAlto	CursorPen[2]
@Pena[3]		LBaixo	LAlto	CursorPen[3]
				
				
				
				
				
				

OK Cancel

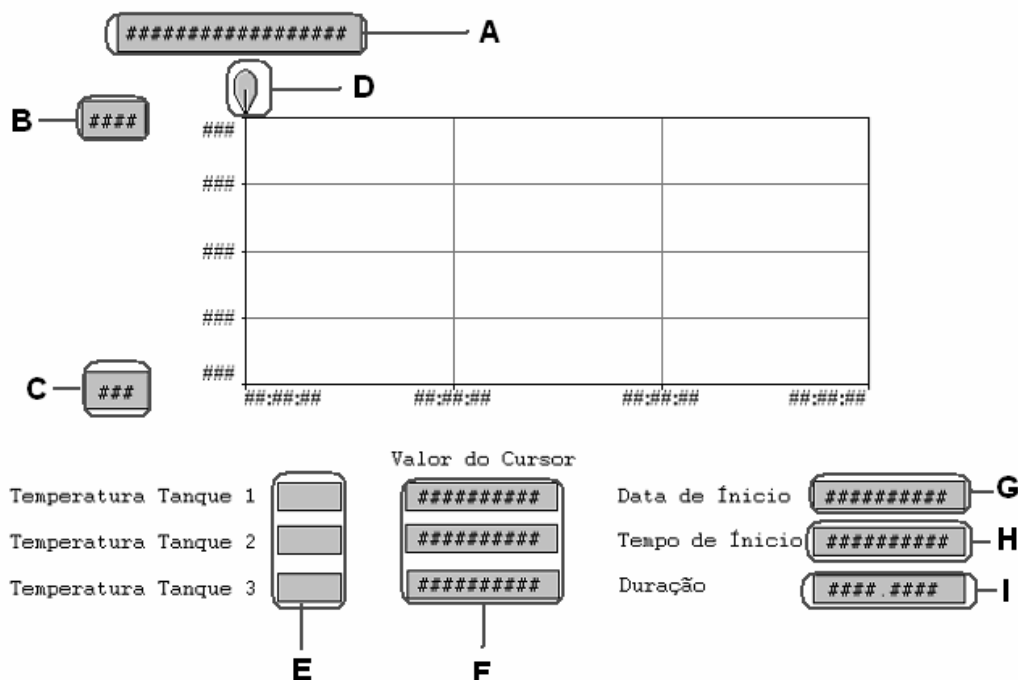
- Clique OK.
- Clique com o botão direito em qualquer lugar vazio da tela e escolha a opção **Screen Attributes**,
- Selecione o check-box On Open e clique sobre ele em seguida.

- Configure como mostrado a seguir:

	Tag	Expression
01	LAlto	100
02	LBaixo	0
03	Pena[1]	Temperatura[1]
04	Pena[2]	Temperatura[2]
05	Pena[3]	Temperatura[3]
06		
07		
08		
09		
10		
11		
12		

- Desenhe a tela como mostrado abaixo:

#####	#####	Alarme Histórico	Alarme OnLine	Relatórios	Principal
Usuário: #####		Trend Histórico	Alarme Histórico	Receitas	Sair



- **A.** Text I/O – Tag =Scursor
Position – Horizontal
Tag = Pcursor
Slider = Habilitado
Range Position
0 to 100 0 to 350
- **B.** Text I/O – Input Enabled Habilitado
Tag = Lalto
- **C.** Text I/O – Input Enabled Habilitado
Tag = Lbaixo
- **D.** Bitmap da Biblioteca (Buttons)
Position – Horizontal
Slider = Habilitado
Tag = Pcursor
Range Position
0 to 100 0 to 350
- **E.** Três Retângulos com a propriedade “Command”
Tag = Pena [X]
Expression:
IF(Pena[X]="Temperatura[X]", "", "Temperatura[X]")

Onde X = Número 1, 2 ou 3 referente ao tanque. Se for Temperatura [1], substitua o X por 1, e assim por diante.

- **F**. Três Text I/O – Input Enabled Desabilitado
Tag = CursorPen[X]

Onde X = Número 1, 2 ou 3 referente ao tanque. Se for Temperatura [1], substitua o X por 1, e assim por diante.

- **G**. Text I/O – Input Enabled Habilitado
Tag = DataIni
- **H**. Text I/O – Input Enabled Habilitado
Tag = TempoIni
- **I**. Text I/O – Input Enabled Habilitado
Tag = Duracao



IMPORTANTE:

- Salve a tela como **TrendHistórico** e execute a aplicação.

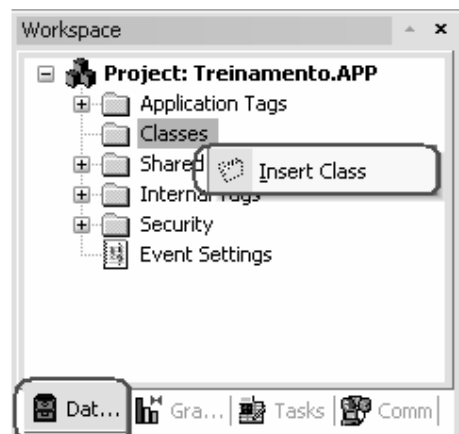
Criando Receitas (Recipes)

O módulo **Recipes** permite criar, recuperar e apagar receitas. Receitas são arquivos que contêm qualquer conjunto de valores de tags, sejam de fato uma receita de processo ou parâmetros para o sistema operar em determinadas situações. Na indústria alimentícia, química, podemos facilmente imaginar receitas compostas por matérias-primas e suas respectivas quantidades. Isso compõe uma receita. Mas também um forno pode precisar deste módulo para armazenar rampas de aquecimento, desaquecimento onde as receitas são compostas de patamares de tempos e temperaturas respectivas. Utilizaremos este último exemplo para a nossa aplicação de receita.

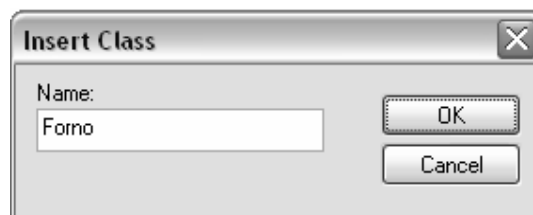
Criando a planilha de Receita (Recipe Worksheet)

Primeiramente serão criados os tags com os quais estaremos trabalhando nesta parte do treinamento.

- Abordaremos agora o conceito de CLASSE, para isso crie uma classe com o nome FORNO com os membros: TEMPERATURA, PRESSAO, QUEIMADOR, VALVULAABERTURA e VALVULAFECHAMENTO.
- Faremos um exemplo como se fosse a preparação de um Forno.
- Para criar a CLASSE, clique com o botão direito na pasta *Classes* na Aba *Database* e escolha *Insert*.



- Na caixa de dialogo, digite o nome da classe, como, por exemplo, Forno e clique em OK.



- Na planilha da classe, crie os nomes dos membros da Classe. Configure como mostrado abaixo:



	Name	Type	Description
1	Temperatura	Real	
2	Pressao	Real	
3	Queimador	Real	
4	ValvulaAbertura	Boolean	
5	ValvulaFechamento	Boolean	
6			
7			
8			

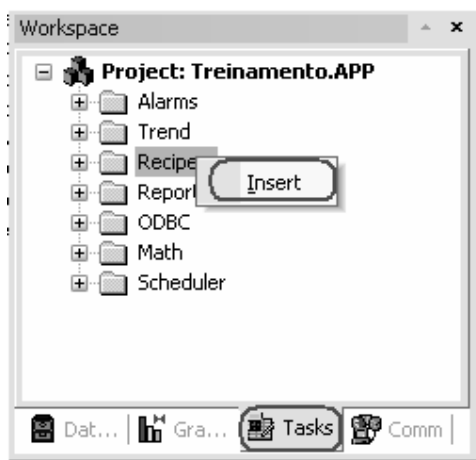
- A planilha pode ser fechada sem ser salva.



IMPORTANTE:

- Crie o Tag SETPOINT do tipo Class: Forno com Array Size 0.
- Crie o Tag RdFileNStatus do tipo Inteiro com Array Size 0.
- Crie o tag Forno do tipo String com Array Size 0

- Crie agora uma planilha de Receita.
- Para criar uma Receita, clique com o botão direito na pasta **Recipes** e depois clique em **Insert**, como na figura abaixo:



A Planilha de Recipe é dividida em duas áreas:

Área de Cabeçalho, o qual contém a informação para o grupo inteiro.

Área de Corpo, onde você define cada Tag no grupo.

Use os parâmetros do cabeçalho nesta planilha como segue:

- **Campo Description:** Digite uma descrição da planilha para documentação posterior.

- **Check-Box Save As XML** – Se você habilitar a opção de salvar como XML, serão gerados arquivos de Receita no formato XML, os arquivos gerados em XML incluem todos os valores do Tag, junto com o nome do Tag de que aqueles valores originaram. Caso a opção não seja checada os arquivos serão salvos no formato DAT.



IMPORTANTE:

- Você pode carregar a informação em um arquivo do tipo DAT em Tag diferentes usando uma segunda planilha de Receita, mas você deve carregar a informação em um arquivo do tipo XML em Tag com o mesmo nome que o Tag dos dados.



Nota:

Como com páginas do HTML, você deve startar o web server para ver dados de XML via Web. Ao contrário das páginas do HTML no sistema run-time, as páginas de XML não necessitam ter a aplicação funcionando para ver os dados de XML. (você deve ter a versão 5.0 ou superior do Internet Explorer para ver dados de XML).

- **Campo File Name** - é colocado o nome do arquivo de receita que será salvo.



IMPORTANTE:

- O Nome de arquivo **Recipe1.rcp** é o nome do arquivo de configuração das receitas, que contém em seus dados o nome do tag que receberá o nome da receita (no campo **File Name**) e os tags que serão gravados.
- Os arquivos criados pelas receitas têm a extensão padrão ***.dat**
- Digite em File Name o tag que dará nome ao arquivo: { **NomeForno** }.
- Colocando apenas o nome do Tag **entre chaves**, os arquivos de receita serão gravados no diretório do documento. Se forem criados novos diretórios para as receitas, o caminho destes precisa ser colocado neste campo. Por exemplo, se você tiver no campo File Name algo como: *c:\AppName\Recipe\NomeForno*. Neste caso o valor dentro do tag **NomeForno** será colocado no diretório *c:\AppName\Recipe*.

- **Campo Register Number** – Digite um Tag para definir o número do registro a ser lido ou escrito em um arquivo de DBF. O InduSoft permite habilitar este campo para que algumas aplicações mais velhas criadas para usar arquivos de DBF, mas é inutilizados para os projetos mais novos.
- **Check-Box Unicode:** Habilite este check-box para salvar o arquivo de receita no formato UNICODE (dois bytes por caracter) ou desabilite para salvar o arquivo de receita no formato ANSI (um byte por caracter).



Nota:

Quando salvamos uma planilha, você pode salva-lá usando algum nome que você escolher (você não precisa usar um nome predefinido do arquivo). Um arquivo de configuração que usa a extensão padrão RCP (ou o XSL se você especificar Salvar como XML) contém a configuração da receita e o arquivo que o campo File Name contém o nome do arquivo de dados a ser lido ou escrito.

Use os parâmetros do corpo nesta planilha como segue:

- **Coluna Tag Name:** Digite o nome do tag para atualizar o conteúdo do arquivo ou com valores ou criar um arquivo. Se o tag é um array, você pode especificar a primeira posição a ser utilizada.
- **Coluna Number of Elements:** Usado para tags do tipo vetor, para evitar a digitação de todas as posições. Configura-se a primeira posição do array e o número de posições subsequentes.



IMPORTANTE:

- Quando você define um Tag array, sua posição inicial é zero, embora IWS use o Tag no caso de uma configuração inválida de posição.

- Preencha os campos como mostrado abaixo:

Recipe1.rcp

Description: Preparação do Forno

File Name: {NomeForno}

Register Number:

Options: ☐ Save As XML, ☒ Unicode

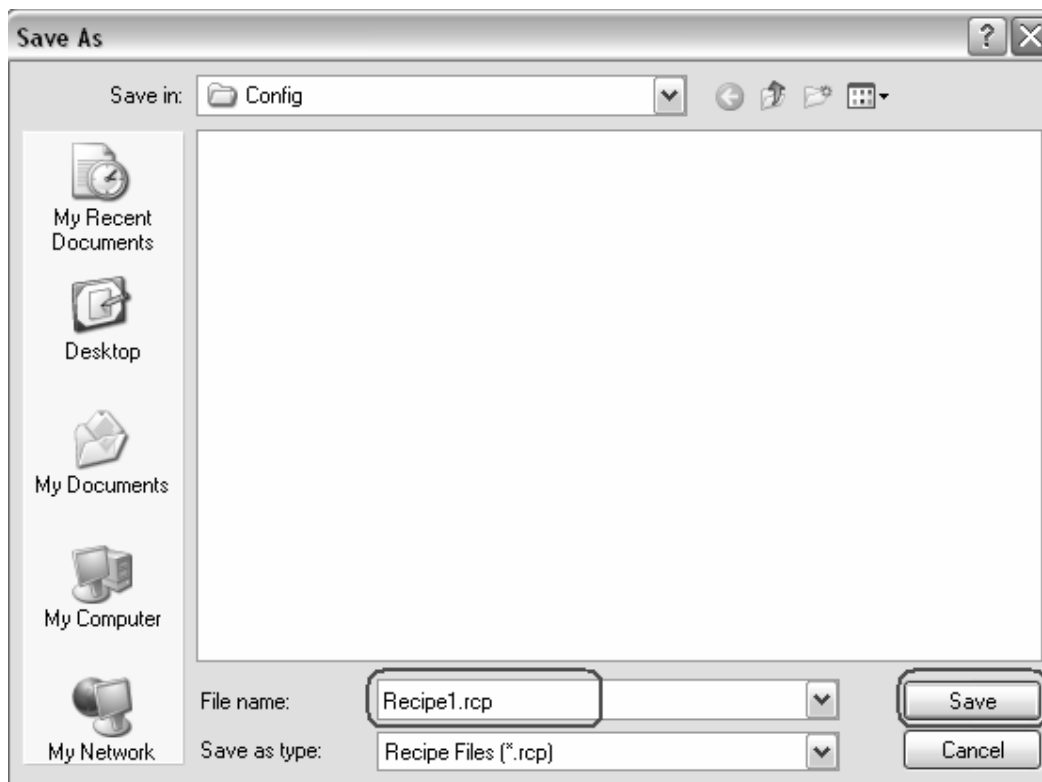
	Tag Name	Number of Elements
1	SetPoint.Temperatura	
2	SetPoint.Pressao	
3	SetPoint.Queimador	
4	SetPoint.ValvulaAbertura	
5	SetPoint.ValvulaFechamento	
6		



IMPORTANTE:

- Salve a receita como Recipe1.

- Para **salvar**, clique no menu **File** na barra superior, e no item **Save As**
- Digite o nome **Recipe1** no campo **Nome do arquivo**, e clique em Salvar.



Nota:

- Lembre-se que a sintaxe usada para acessar um valor de um Tag do tipo classe é:
< nome_tag>.<nome_membro >
(para o exemplo, **SetPoint.Temperatura**, ou **SetPoint.Pressao**, etc.).

Criando uma tela de Receitas (Recipe Screen)

- Abra a tela Standard no tab Graphics na pasta Screens, e desenhe os objetos como mostrado na figura a seguir e SALVE-A como Receita.



IMPORTANTE:

- No passo a seguir deixe sempre a opção **Input Enabled** habilitada em todos eles. Isso permitirá digitar valores.

- Junto ao texto **Nome do Forno**, coloque um texto com propriedade Text I/O para o tag **NomeForno**
- Junto ao texto **Temperatura**, coloque um texto com propriedade Text I/O para o tag **SetPoint.Temperatura**
- Junto ao texto **Pressão** coloque um texto com propriedade Text I/O para o tag **SetPoint.Pressao**
- Junto ao texto **Queimador** coloque um texto com propriedade Text I/O para o tag **SetPoint.Queimador**
- Junto ao texto **Válvula de Abertura** coloque um texto com propriedade Text I/O para o tag **SetPoint.ValvulaAbertura**
- Junto ao texto **Valvula de Fechamento** coloque um texto com propriedade Text I/O para o tag **SetPoint.ValvulaFechamento**

#####	#####	Alarme Histórico	Alarme OnLine	Relatórios	Principal
Usuario: #####		Trend Histórico	Trend OnLine	Receitas	Sair

Nome do Forno

Temperatura

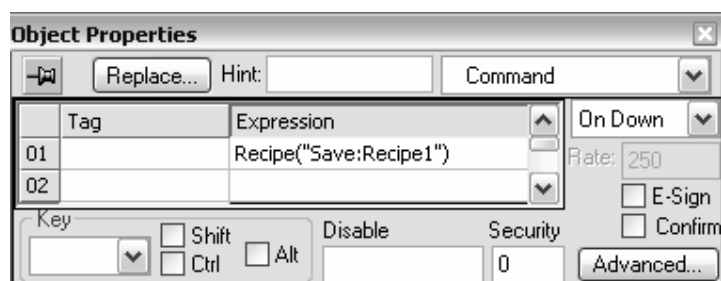
Pressão

Queimador

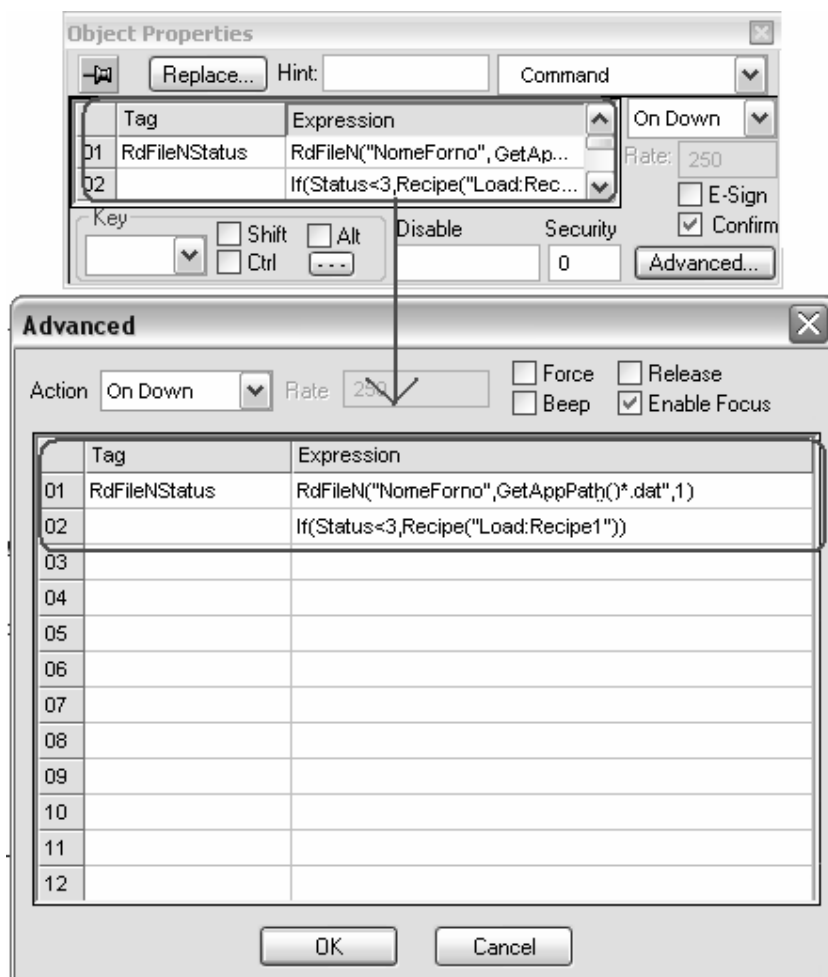
Válvula de Abertura

Válvula de Fechamento

- Insira dois botões, um com **Caption** com valor **Salvar** e outro **Carregar**.
- Clique duas vezes sobre o botão **Salvar**. Insira a propriedade Command. Em Expression digite o seguinte comando como na figura abaixo:



Faça o mesmo com o botão **Carregar** e insira o comando abaixo:



- Clique em Ok.



IMPORTANTE:

- Salve a tela como **Receitas**

- Salve a tela clicando no ícone ou através do menu **File, Save**.
- Rode a aplicação usando o ícone **Run Application**.



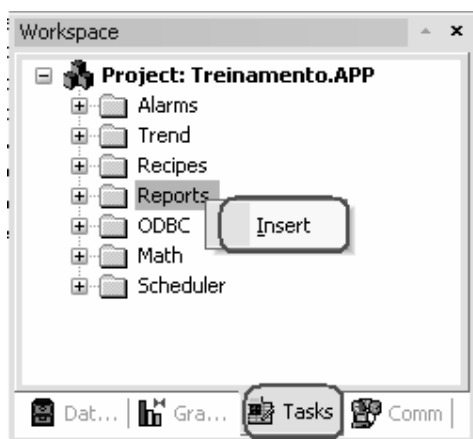
- Agora brinque com os valores e nomes das Receitas.

Criando Relatórios (Reports)

A ferramenta de relatórios **InduSoft Report** permite a criação e impressão de relatórios sem a necessidade de nenhuma outra ferramenta que não o próprio InduSoft Web Studio (não necessita VB, Pascal, C++ etc.). Tudo o que se necessita é a criação da máscara do relatório em formato ASCII ou RTF (que utiliza o nosso Report Writer), mesclando textos e valores de tags, e executando os comandos de **Reports** a partir de um botão de tela, ou scheduler etc.

Criando Relatório tipo ASCII

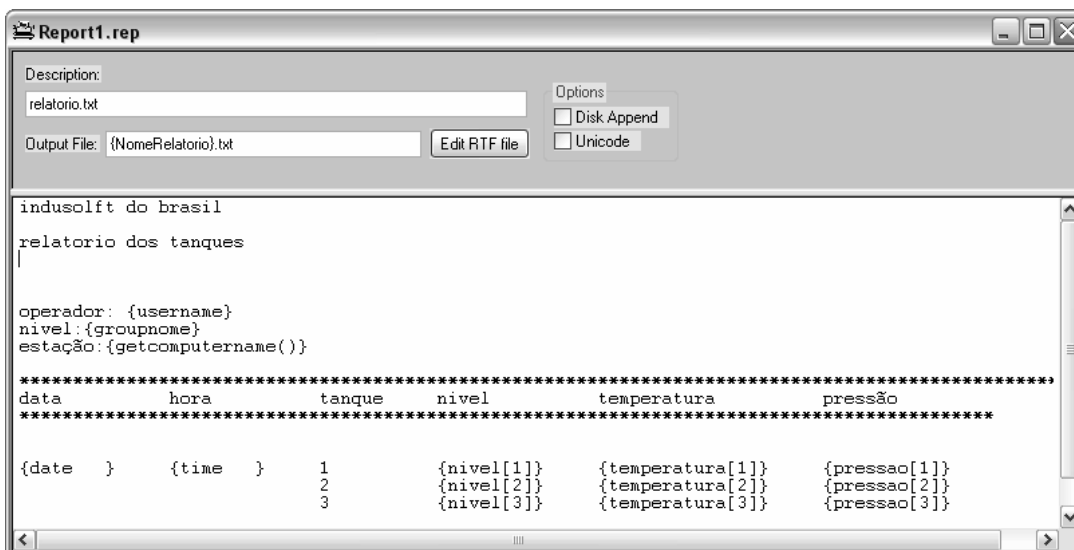
- Primeiramente vamos criar um relatório no formato TXT.
- Clique com o botão direito na pasta **Reports** (tab **Tasks**) e selecione a opção **Insert** para criar uma nova planilha de Relatório



NOTA:

- No caso de relatório em ASCII, é importante saber quantos caracteres serão mostrados (valor do tag) e prever esse espaço dentro das chaves {}, inclusive os caracteres abre-chave e fecha-chave. Ou seja, para mostrar o tag **Date**, no caso de relatório **ASCII**, {**Date** } afim de que caibam todos os caracteres de uma data formato DD/MM/AAAA
- No caso do relatório **RTF**, isto não é considerado. No entanto, nem sempre o número de caracteres do tag vai permitir um alinhamento correto de colunas no relatório. A função **Format** ajuda muito e, além disso, na hora de escolher a fonte, lembre-se que há fontes proporcionais, em que caracteres diferentes ocupam espaços diferentes.

- Configure a planilha como mostrado abaixo:

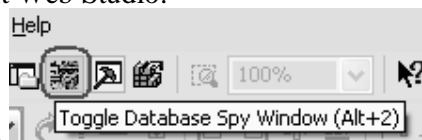


- Salve esta planilha com o nome de **Report1**

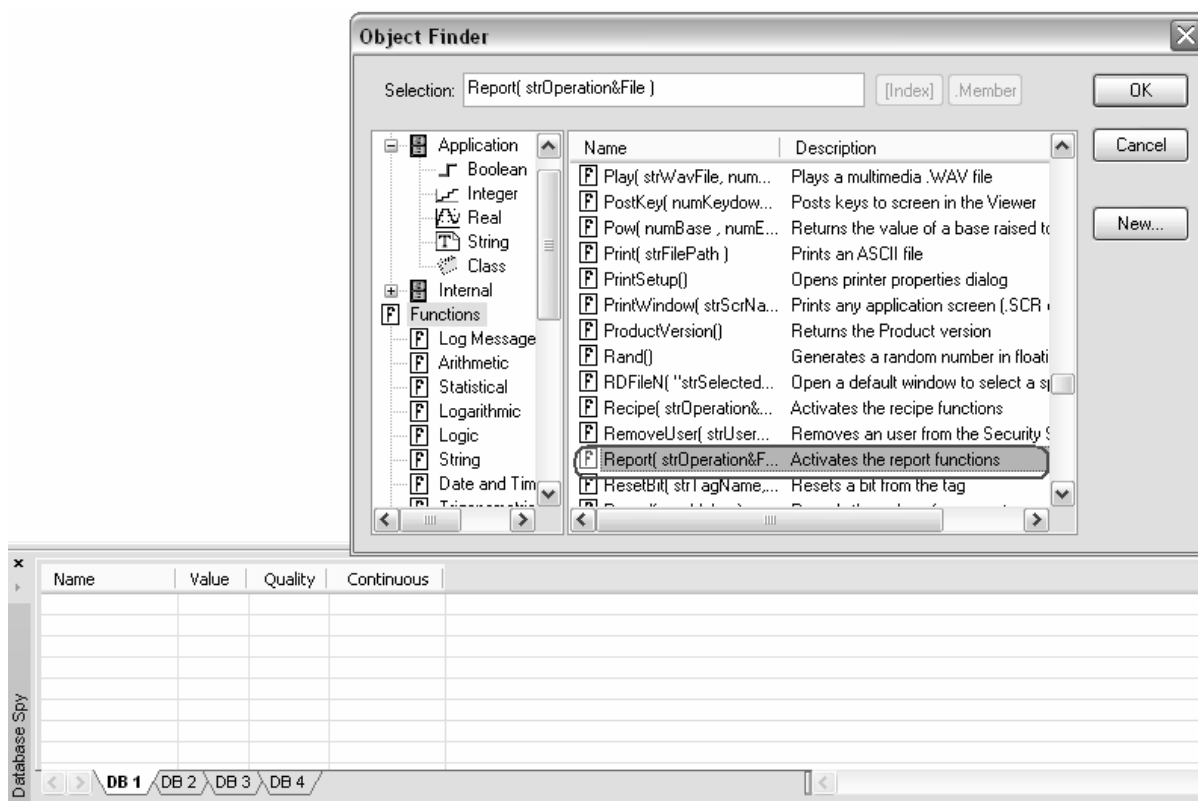
Utilizando o Database Spy para testar partes da aplicação

Tipicamente, um relatório deve ser disparado em um horário determinado, por um evento do processo ou por solicitação do operador. Porém como o desenvolvimento da sua aplicação não requer uma seqüência rígida, vamos por hora utilizar a ferramenta de debugger do InduSoft Web Studio, o Database Spy para testar a configuração de Report que acabamos de executar.

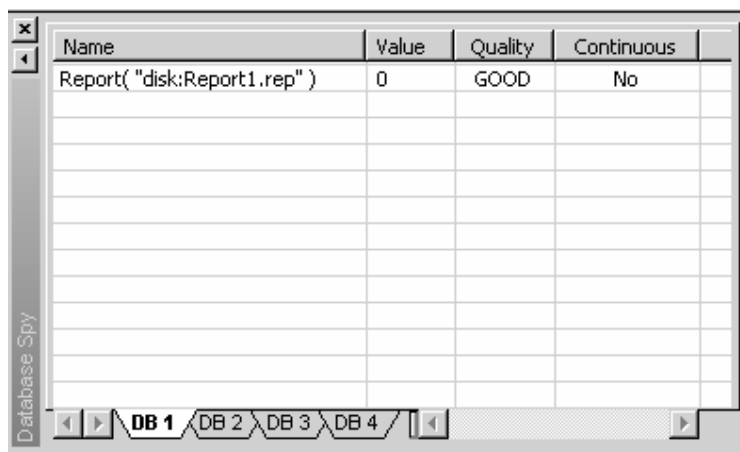
- Certifique-se que sua aplicação esteja rodando. Se não estiver coloque-a em execução através do botão Run Application do Toolbar.
- Volte ao mode de Engenharia InduSoft Web Studio.



- Vá ao Database Spy clicando no ícone
- Dê um duplo-clique na primeira linha vaga, na coluna Name. A janela do Object Finder deve abrir como mostra a figura abaixo.



- Selecione a função Report.
 - Vá ao campo Selection e edite a função: **Report("disk:Report1.rep")** e clique OK.
- Sua janela Database Spy deve apresentar-se tal como a que segue. Value 0 é o retorno da função, onde 0 indica execução normal.



- Verifique, através do Explorer, no diretório da sua aplicação o arquivo Relatorio.TXT.

Criando relatórios do tipo RTF, com Cabeçalho e Corpo

A maioria dos relatórios é baseada em conceito *cabeçalho/corpo*. Isto significa um cabeçalho que pode incluir um bitmap com o nome da empresa, logotipo, nome das variáveis, etc... e a seguir valores das várias coletas de dados.

Fazer isso com o InduSoft é fácil. Serão criadas duas planilhas.

Uma terá o Cabeçalho e será chamada **Report2**.

A outra terá a formatação dos tags que serão gravados a cada X minutos e será chamada **Report3**. Esta segunda planilha terá o check box **Disk Append** habilitado e exatamente o mesmo conteúdo do campo **Output File** do **Report2**.

A opção **Disk Append** salva o relatório sem apagar outro já existente com o mesmo nome, apenas gravando no mesmo arquivo, assim permitindo vários registros da mesma máscara em um único relatório.

Com o exemplo abaixo isso deve ficar mais claro:

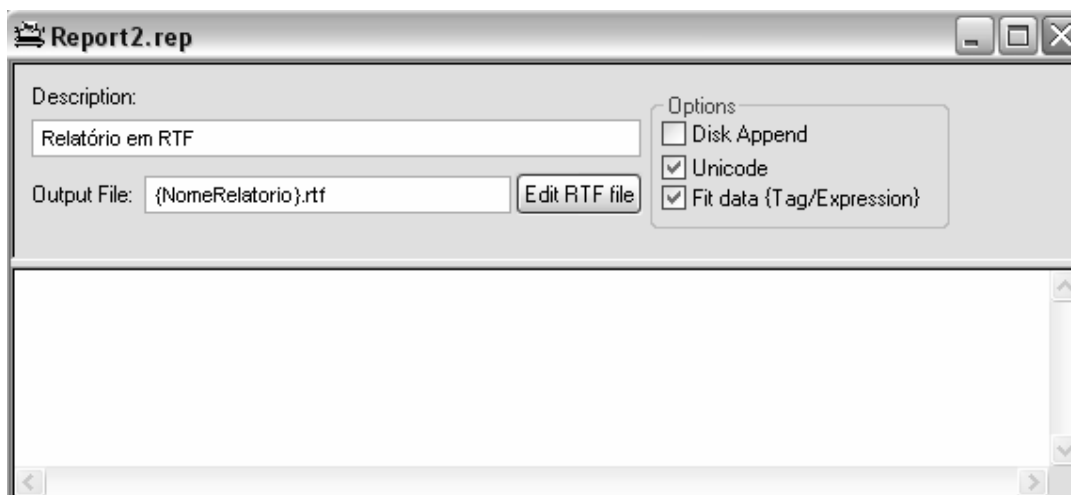



IMPORTANTE:

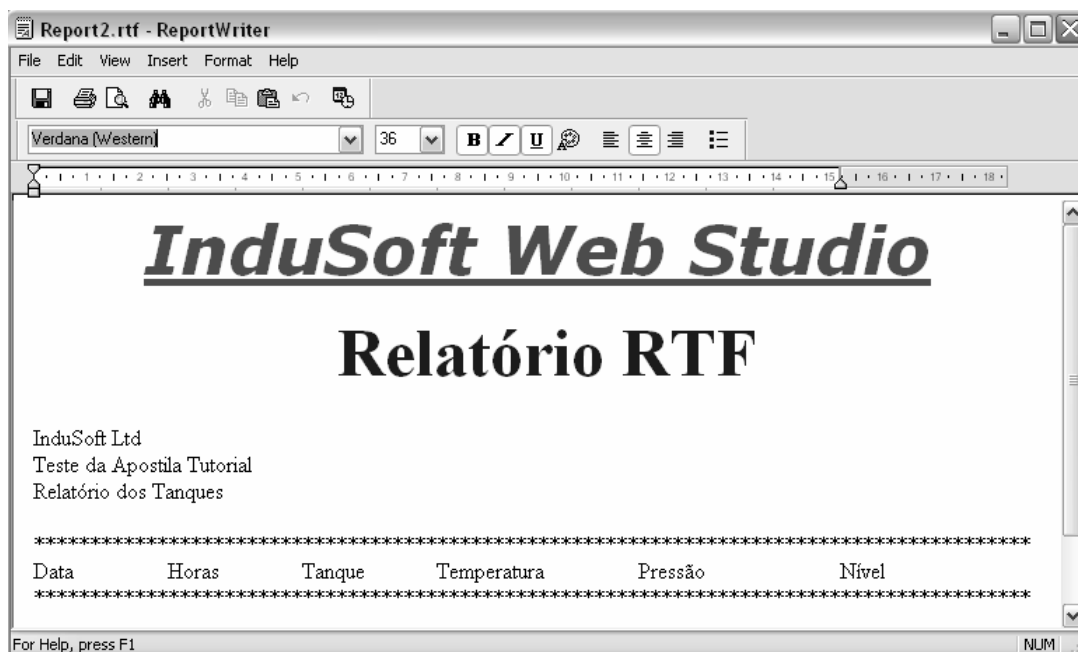
- Crie o tag **NomeRelatorio** do tipo String com Array Size 0.

Crie primeiro a planilha que será o Cabeçalho.

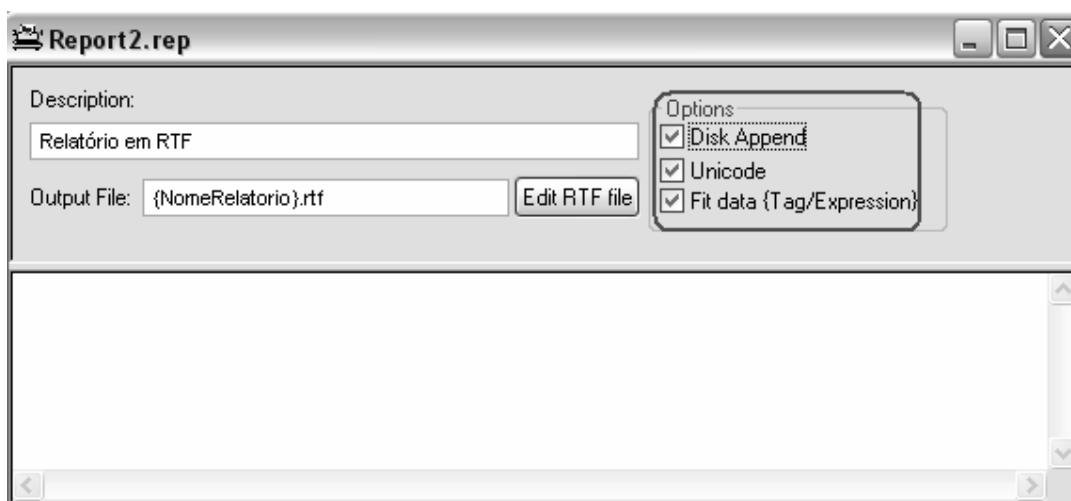
- Vá ao **tab Tasks** e clique com o botão direito na pasta **Reports**, clique em **Insert** e crie uma nova planilha (Report2), e configure como mostrado abaixo,



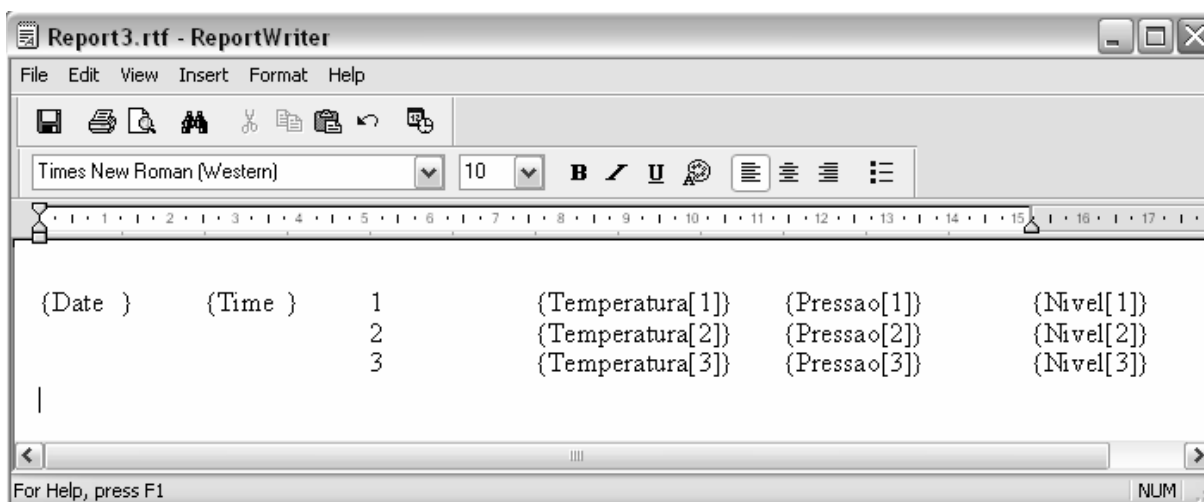
- Salve esta planilha com o nome de Report2
- Clique em  e configure a página RTF que será o cabeçalho do relatório como mostra a figura abaixo:



- Salve o RTF e em seguida feche-o.
- Repare que não há nenhum tag incluído nesta planilha.
- Agora vá ao **tab Task** e clique com o botão direito na pasta **Reports**, clique em **Insert** e crie uma nova planilha (Report3), que será a máscara do corpo do relatório.
- Não esqueça de habilitar a opção **Disk Append**.
- Faça como mostrado abaixo:



- Salve esta planilha com o nome de Report3
- Clique em **Edit RTF file** e configure a página RTF que será a máscara do corpo do relatório como mostra a figura abaixo:



- Salve-a novamente e FECHÉ-A.
- Como pudemos ver acima, as diferenças entre **Report2** e **Report3** se encontram nos textos **RTF** e no check box **Disk Append**.

Configurando uma planilha Scheduler

O evento **Clock** é utilizado para disparar eventos baseados em temporizadores. Na coluna **Time**, configuramos a base de tempo (mínimo de 100ms). Isso significa que, a cada HH: MM: SS.100ms o evento será executado. Na coluna **Tag**, devemos configurar o tag que receberá o valor da expressão da coluna **Expression**. Finalmente, a coluna **Disable** pode conter um tag, função ou expressão que, se verdadeira, impedirá a execução do comando. Desta maneira é possível controlar a base de tempo da execução bem como se deve executar sim ou não. As colunas **Trigger** e **Date** não são utilizadas para este tipo de evento

O tipo **Calendar** dispara eventos em uma data e hora definida, ou sempre no mesmo horário todos os dias. A coluna **Trigger** não é utilizada. Já as colunas **Tag**, **Expression** e **Disable** são utilizadas da mesma forma que é explicado acima para o evento **Clock**.

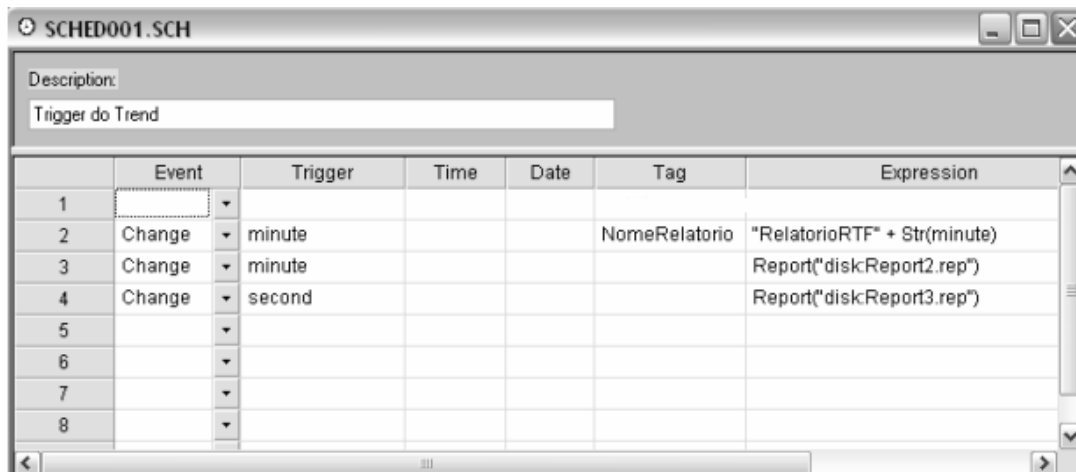
Já o tipo **Change** dispara uma ação sempre que o tag colocado na coluna **Trigger** tem o seu valor alterado. As colunas **Time** e **Date** não são utilizadas e as demais têm o mesmo comportamento descrito para os tipos anteriores.

Criando Eventos no módulo Scheduler

- No tab Tasks, insira uma nova planilha de Scheduler



- Reproduza a configuração conforme indicado a seguir:



Essa configuração faz com que a cada minuto se inicie um novo relatório, que será adicionado de novas informações (Report3), uma vez por segundo. Até que no minuto seguinte se inicie um novo relatório, conforme muda o valor do tag NomeRelatorio.

- Verifique os arquivos que estão sendo gerados a partir dos Report2 e Report3. Se sua aplicação não estiver rodando, coloque-a em execução usando o ícone **Run Application**



Criando uma tela de Relatórios (Report Screen)

- Abra a tela Standard no tab Graphics na pasta Screens, e desenhe os objetos como mostrado na figura a seguir e SALVE-A como Relatórios.
- Junto ao texto **Nome do Relatório**, coloque um texto com propriedade Text I/O para o tag **NomeRelatorio**



IMPORTANTE:

- Deixe sempre a opção **Input Enabled** habilitada em todos eles. Isso permitirá digitar valores.

#####	#####	Trend OnLine	Alarme OnLine	Relatórios	Principal
Usuário :#####		Trend Histórico	Alarme Histórico	Receitas	Sair

Nome do Relatório

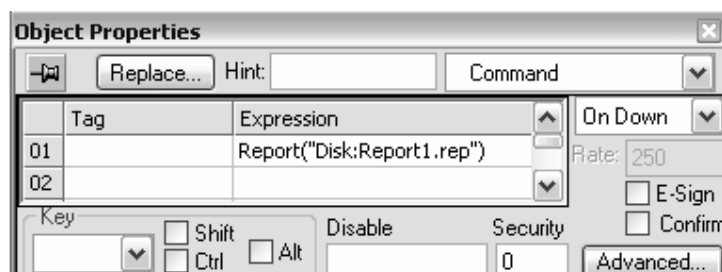
Salvar TXT	Abrir TXT
Salvar Cabeçalho RTF	Abrir RTF
Salvar Valores RTF	



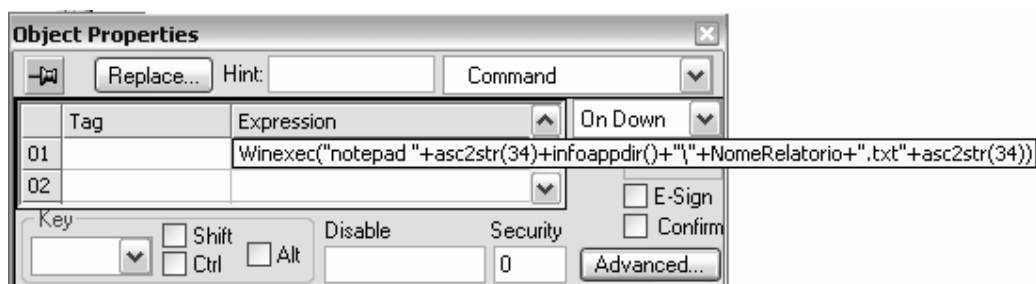
IMPORTANTE:

- Salve a tela com **Relatórios**.

- Clique duas vezes sobre o botão **Salvar TXT**. Insira a propriedade Command. Em Expression digite o seguinte comando como na figura abaixo:



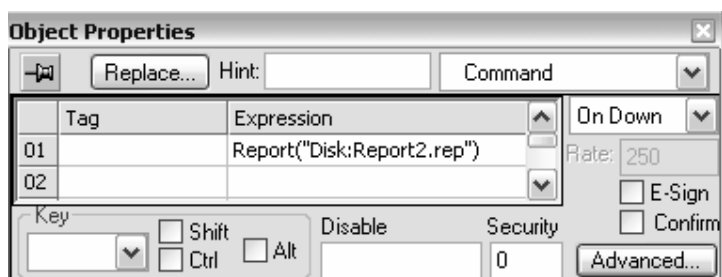
Já para o botão Abrir TXT e insira o comando abaixo:



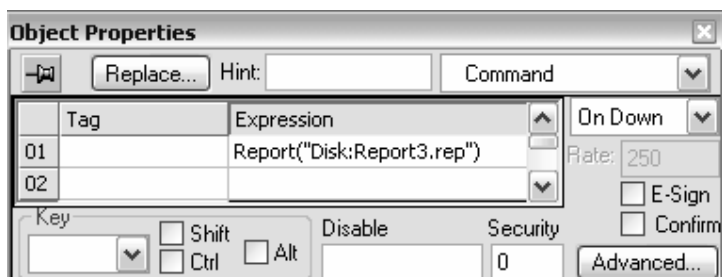
Estaremos dessa forma utilizando o programa Notepad do windows para abrir o relatório em ASCII (formato texto).

A função WinExec permite a execução de programas que podem ser executados com o comando Run do Windows.

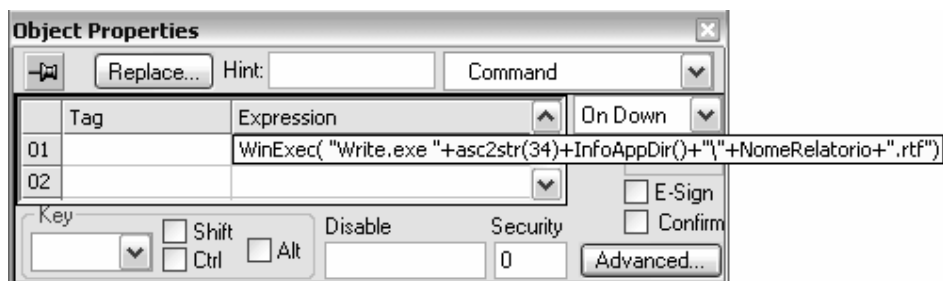
Faça o mesmo com o botão **Salvar Cabeçalho RTF** e insira o comando abaixo:



Repita o procedimento com o botão **Salvar Valores RTF** e insira o comando abaixo:



Já para o botão Abrir RTF e insira o comando abaixo:



- Clique em Ok.
- Salve a tela clicando no ícone ou através do menu **File, Save**.
- Rode a aplicação usando o ícone **Run Application**.



- Agora brinque com os nomes dos Relatórios.

O INDUSOFT WEB STUDIO

O InduSoft Web Studio tem a propriedade de trabalhar como estação Servidora de Dados para acesso de páginas Web.

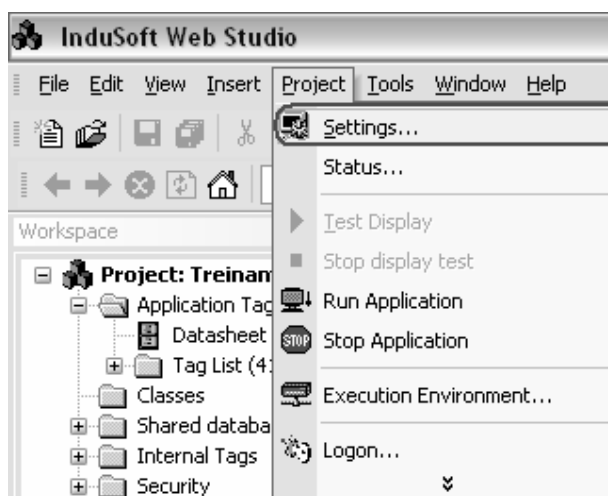
Você pode monitorar ou até comandar de qualquer estação rodando um browser, qualquer aplicação que esteja rodando no InduSoft Web Studio.

Não é necessário que o micro que será utilizado como client tenha o programa do InduSoft Web Studio instalado. A estação Servidora de Dados, que está rodando InduSoft Web Studio precisar ter licenças Web Thin Client, em número suficiente para suportar os usuários Web simultaneamente conectados a esta.

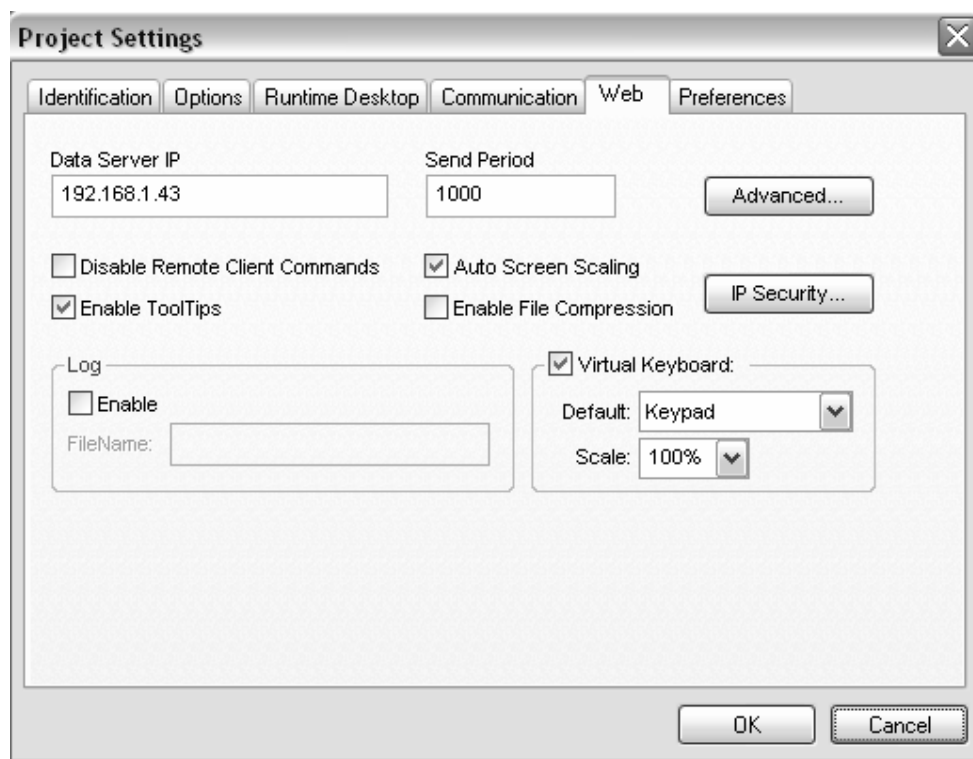
Configurando para Web

Configurando uma aplicação para simulação de valores de processo via Web.

Vá ao menu Project e clique sobre Settings.



Abra a janela Project Settings, clique sobre a aba Web.



Janela Project Settings da aba Web

-Configure a janela com base nas informações relacionadas abaixo:

Data Server IP: É o endereço TCP/IP do micro onde está rodando a aplicação (Server) do InduSoft Web Studio.

Send Period: Período em milissegundos que o InduSoft Web Studio leva para atualizar no micro client, os valores dos tags configurados como **Server**.

Disable Remote Client Commands - Desabilita todos os comandos remotos do client.

Enable ToolTips – Tool Tips é quando paramos por alguns instantes sobre um objeto, e então aparece uma janela dizendo o que é aquele objeto. Esta função habilita e desabilita esta opção na Web.

Enable File Compression – Se esta opção estiver habilitada os arquivos do diretório Web da aplicação serão transferidos do Micro Server para o Micro Client, de modo reduzido facilitando assim o download das telas e atualizações de dados no Micro Client.

IP Security – Abre uma janela de segurança de endereços.

Advanced - Abre uma janela de configurações avançadas para conexões Web.

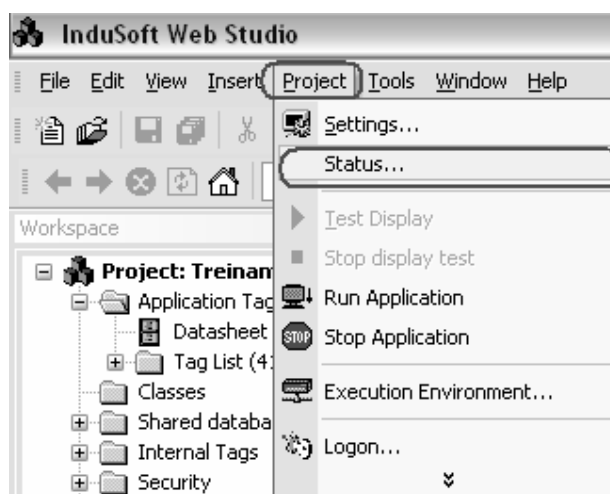
Log – Salva na área de trabalho em arquivo .dat, as mensagens de debug da conexão Web.

Enable: Habilita o campo FileName

FileName: campo para especificação do diretório onde se quer salvar as mensagens debug.

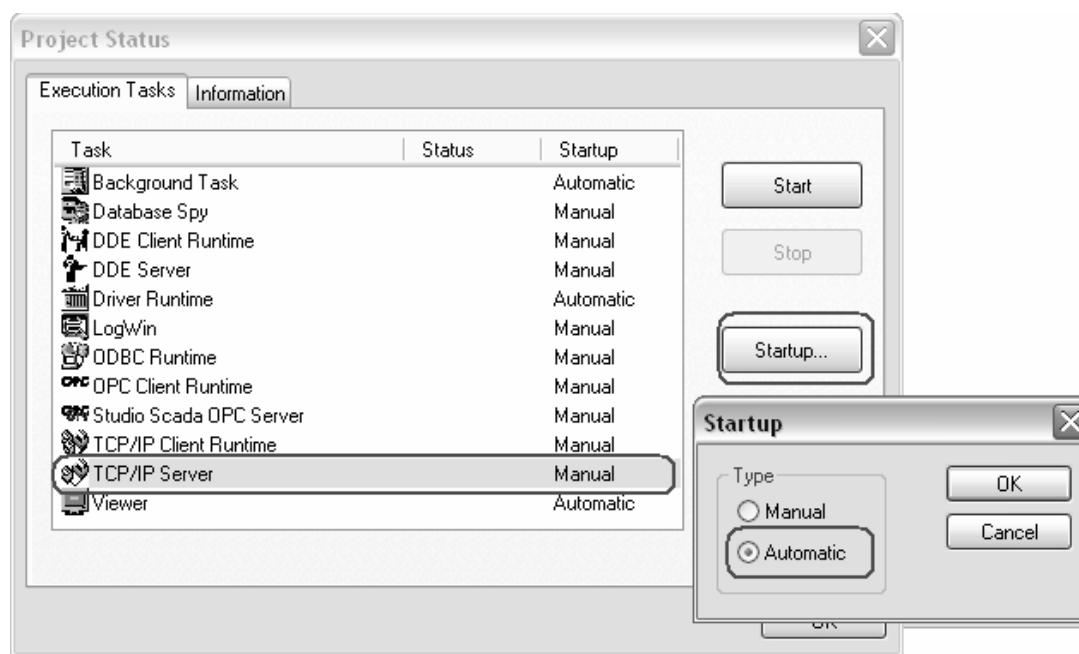
O módulo TCP/ IP server, deve ser configurado como **Automatic**, assim como o **Viewer**.

Para configurar o módulo TCP/ IP server como **Automatic** deve-se clicar no menu Project, Status...



Em seguida conferir se está na aba Execution Tasks, de um clique sobre o módulo TCP/ IP Server, em seguida clique no botão Startup...

Selecione o check- Box **Automatic**, clique em Ok



Clique em Ok novamente.

Para que a aplicação funcione via Web, é preciso fazer algumas configurações no Micro Server em que a aplicação foi desenvolvida.

Vamos a elas:

Clique no menu INICIAR do Windows, PROGRAMAS, ACESSÓRIOS, WINDOWS EXPLORER.

Clique em ARQUIVOS DE PROGRAMAS, INDUSOFT WEB STUDIO, BIN.



IMPORTANTE:

- O aplicativo NTWebServer, **não deve** ser movido da pasta Bin e sim copiado para dentro pasta Web da aplicação.



NTWebServer.exe

Localize e **COPIE** o aplicativo , **COLE** dentro da pasta **Web** da sua aplicação.



IMPORTANTE:

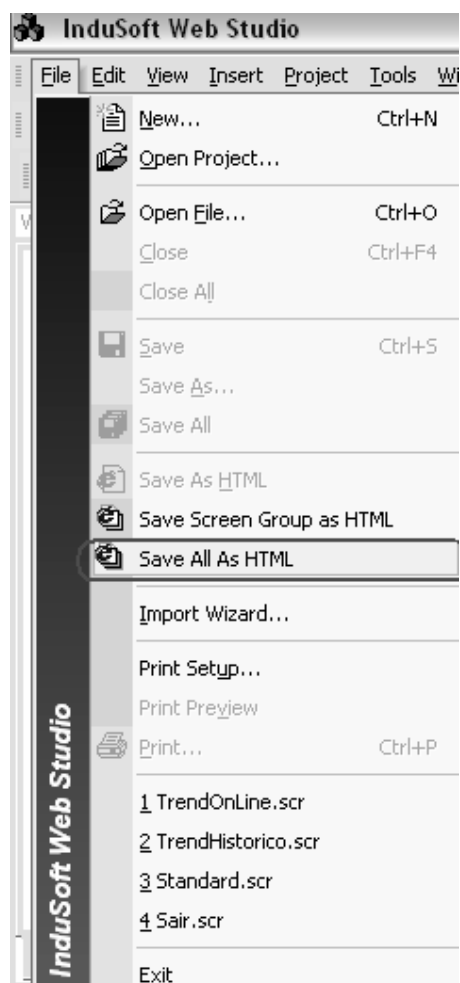
- Não arraste este aplicativo para a sua pasta, pois este procedimento pode provocar erros de funcionamento. Somente copie o NTWebServer.exe.

Depois de ter copiado o aplicativo NTWebService para a pasta Web da sua aplicação, duplo clique sobre o aplicativo e minimize-o .

Feche o Windows Explorer.

De volta ao ambiente de Engenharia do InduSoft Web Studio, vá ao menu superior e clique em **File**.

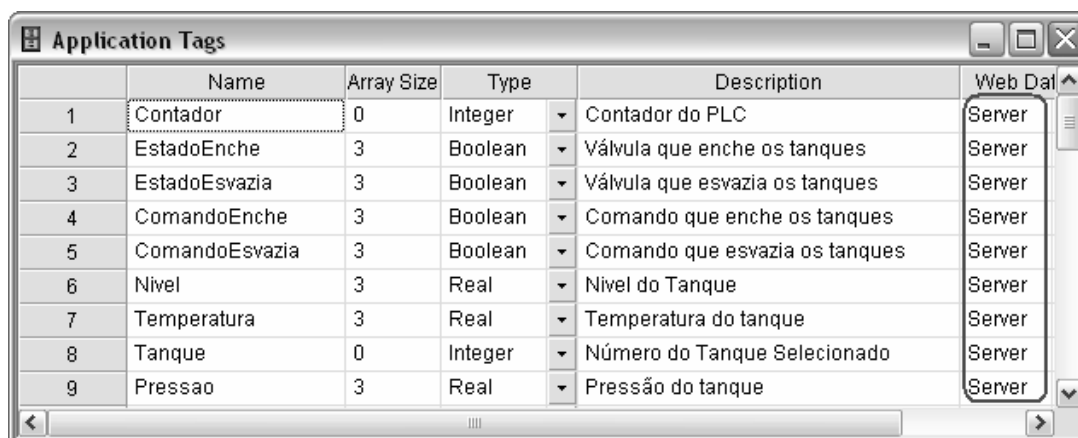
Em seguida clique sobre a opção Save All As HTML, como mostra a figura abaixo.



Perceba que ao clicar sobre a opção Save As HTML, o ponteiro do mouse irá piscar algumas vezes, este fato é normal e depois disso a tela automaticamente será convertida para o formato HTML.

Feche a tela Principal

Vá até o tab DataBase e clique duas vezes sobre a pasta Application Tags, clique duas vezes sobre Datasheet View, aparecerá a janela Application Tags, nela verifique se os tags da aplicação estão configurados como Server no campo Web Data, caso não estejam configure-os já os que se mantiverem como Local, só receberam comando do Micro Server e nenhum comando do Micro Client, como mostra a figura abaixo.

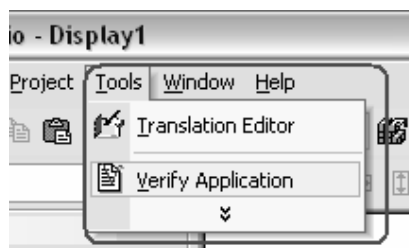


	Name	Array Size	Type	Description	Web Dal
1	Contador	0	Integer	Contador do PLC	Server
2	EstadoEnche	3	Boolean	Válvula que enche os tanques	Server
3	EstadoEsvazia	3	Boolean	Válvula que esvazia os tanques	Server
4	ComandoEnche	3	Boolean	Comando que enche os tanques	Server
5	ComandoEsvazia	3	Boolean	Comando que esvazia os tanques	Server
6	Nivel	3	Real	Nível do Tanque	Server
7	Temperatura	3	Real	Temperatura do tanque	Server
8	Tanque	0	Integer	Número do Tanque Selecionado	Server
9	Pressao	3	Real	Pressão do tanque	Server

Feche a janela Application Tags.

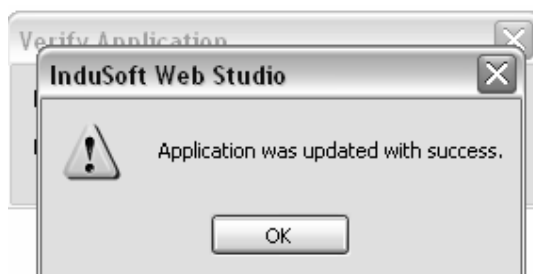
Agora vá ao menu superior e clique em **Tools**.

Em seguida clique sobre a opção **Verify Application**, como mostra a afigura abaixo.



Depois de clicar sobre a opção Verify Application, o sistema faz uma varredura na aplicação procurando por erros.

Caso o sistema não encontre erros será informada uma mensagem de que aplicação foi verificada com sucesso. Como mostrado na figura abaixo.



Clique em OK.

Rode a aplicação utilizando o Run Application do InduSoft Web Studio.