Manual do Software

SCADABR

Fundação Centros de Referência em Tecnologias Inovadoras Financiadora de Estudos e Projetos Serviço de Apoio às Micro e Pequenas Empresas Conselho Nacional de Desenvolvimento Científico e Tecnológico

SCadaba

ScadaBR 0.7 Sistema Open-Source para Supervisão e Controle

> Manual do Software Outubro de 2010

Sumário

1. Introdução	5
1.1. Sobre o documento	5
1.2. Sistemas SCADA	5
1.3. ScadaBR	6
2. Instalação	7
2.1. JDK (Java Development Kit)	7
2.1.1. Instalando a JDK	
2.1.2. Configurando a JAVA_HOME	7
2.2. Instalando o ScadaBR manualmente	8
2.3. Instalando o ScadaBR via Instalador	8
3. Operando o ScadaBR	9
3.1. Apresentação	9
3.1.1. Menu principal	9
3.1.2. Tipos de dados	9
3.1.3. Data sources	10
3.1.4. Data points	11
3.1.5. Monitoramento	11
3.1.6. Controle	12
3.1.7. Eventos	12
3.1.8. Ícones da Aplicação	12
3.1.9. Sons de Alarmes	13
3.1.10. Graficos	14
3.2. Adicionando Data Sources e Data Points	15
3.3. Visualizando os dados: Watch List e Gráficos	17
3.4. Definindo Eventos	20
3.5. Representações Gráficas	22
4. Data Sources	25
4.1. Modbus	25
4.1.1. Visão geral	
4.1.2. Configuração básica	
4.1.3. Configuração do data point	
4.2. OPC DA	

4.2	.1. Visão geral	28
4.2.	2. Configuração básica	28
4.2	.3. Configuração do data point	28
	atórios	
5.1.	Introdução	30
5.2.	Gerando relatórios no ScadaBR	30
5.2	1. Configurando um novo modelo de relatório	30
5.2	2. Agendamento de Relatórios	31
5.2	.3. Envio de relatórios por e-mail	31
5.2	4. Gerenciamento de modelos	33
	5. Fila de relatórios	
5.3.	Gerando relatórios utilizando softwares de terceiros	34
	.1. Pentaho: Configurando a base de dados do ScadaBR	
	.2. Pentaho: Exemplo de criação de um relatório	
5.3	.3. iReport: Configurando a base de dados do ScadaBR	45
	4. iReport: Selecionando os dados da base a serem utilizados	
6. Scri	pts	50
6.1.	Introdução	
6.2.	Criando o Data Source e os Data Points	
6.3.	6.3. Criando um Meta Data Source e seus Data Points	
6.4.	Criando Point Links	
7. Wat	chdog	62
7.1.	Como funciona	62
7.2.	Configuração	
7.3.	Como executar	63
	as	
	Como alterar o banco de dados?	
9. Glos	ssário	66

1. Introdução

1.1. Sobre o documento

Este tutorial tem como propósito auxiliar os usuários no uso do software ScadaBR. Informações adicionais sobre as funcionalidades disponíveis no software podem ser encontradas no menu principal clicando no ícone .

Informações mais específicas sobre cada funcionalidade também estão disponíveis nas telas de configuração de cada uma delas, clicando no ícone similar .

1.2. Sistemas SCADA

A sigla SCADA é uma sigla do inglês para **S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition, o que significa Controle Supervisório e Aquisição de Dados. Sistemas SCADA servem como interface entre o operador e processos dos mais variados tipos, como máquinas industriais, controladores automáticos e sensores dos mais variados tipos. Com sistemas SCADA são construídos desde aplicativos simples de sensoreamento e automação, até os famosos "Painéis de Controle" em empresas de geração e distribuição de energia elétrica, centrais de controle de tráfego e assim por diante.

Um SCADA típico deve oferecer drivers de comunicação com equipamentos, um sistema para registro contínuo de dados ("datalogger") e uma interface gráfica para usuário, conhecida como "IHM" ou Interface Homem-Máquina. Na IHM são disponibilizados elementos gráficos como botões, ícones e displays, representando o processo real que está sendo monitorado ou controlado. Entre algumas das funções mais utilizadas em sistemas SCADA estão:

- Geração de gráficos e relatórios com o histórico do processo;
- Detecção de alarmes e registro de eventos em sistemas automatizados;
- Controle de processos incluindo envio remoto de parâmetros e set-points, acionamento e comando de equipamentos;
- Uso de linguagens de script para desenvolvimento de lógicas de automação ("receitas").

1.3. ScadaBR

O software ScadaBR é desenvolvido em modelo "open-source", **possuindo** licença gratuita. Toda a documentação e o código-fonte do sistema estão à disposição, inclusive sendo permitido modificar e re-distribuir o software se necessário.

O ScadaBR é uma aplicação multiplataforma baseada em Java, ou seja, PCs rodando o Windows, Linux e outros sistemas operacionais e podem executar o software a partir de um servidor de aplicações (sendo o Apache Tomcat a escolha padrão).

Ao executar o aplicativo, ele pode ser acessado a partir de um navegador de Internet, preferencialmente o Firefox ou o Chrome. A interface principal do ScadaBR é de fácil utilização e já oferece visualização das variáveis, gráficos, estatísticas, configuração dos protocolos, alarmes, construção de telas tipo HMI e uma série de opções de configuração.

Após configurar os protocolos de comunicação com os equipamentos e definir as variáveis (entradas e saídas, ou "tags") de uma aplicação automatizada, é possível montar interfaces de operador Web utilizando o próprio navegador. Também é possível criar aplicativos personalizados, em qualquer linguagem de programação moderna, a partir do código-fonte disponibilizado ou de sua API "web-services".

Em nosso site encontra-se um vídeo demonstrativo que aborda exemplos com as funcionalidades básicas do ScadaBR.

2. Instalação

2.1. JDK (Java Development Kit)

O Tomcat exige do sistema uma JDK, com a variável de ambiente JAVA_HOME devidamente declarada. Certifique-se de que você possui uma JDK 1.6 (ou maior) instalada em seu sistema e de que a variável JAVA_HOME esteja corretamente configurada digitando: *java -version* em um console.

2.1.1. Instalando a JDK

Faça o download da JDK no site da Sun: (http://java.sun.com/javase/downloads/index.jsp). Siga as instruções de instalação padrão.

2.1.2. Configurando a JAVA_HOME

- No Windows XP ou 2000 Navegue: Painel de Controle □ Sistema □ Avançado □ Variáveis de Ambiente Clique em adicionar e insira JAVA_HOME com o endereço do diretório de instalação da JDK.
Inicie um console e entre com <i>java -version</i> para certificar-se que tudo funcionou corretamente.
- No Windows Vista ou 7, navegue: Painel de Controle \square Sistema e segurança \square Sistema \square Configurações Avançadas do Sistema \square Avançado \square Variáveis de Ambiente Clique em adicionar e insira JAVA_HOME com o endereço do diretório de instalação da JDK. Inicie um console e entre com <i>java -version</i> para certificar-se que tudo funcionou corretamente.
- No Linux

No Ubuntu, edite o arquivo .bashrc (em /home/<seu usuario>/), inserindo a linha:

export JAVA_HOME=<endereço do diretório de instalação da JDK>

Feche o console, abra novamente e entre com *java -version* para certificar-se que tudo funcionou corretamente.

2.2. Instalando o ScadaBR manualmente

Faça o download da última versão do Apache Tomcat no site oficial (http://tomcat.apache.org/).

Extraia os arquivos para um diretório de sua escolha "<tomcat-home>".

Execute o arquivo **startup.bat** (Windows) ou **startup.sh** (Linux) do diretório <tomcathome>/bin/ para iniciar o Tomcat.

Em um browser, entre com o endereço: **localhost:8080**/. Caso a página de configuração do Tomcat inicie, o mesmo está corretamente configurado. Caso contrário, certifique-se que todos os passos anteriores (principalmente a instalação da JDK e configuração da variável JAVA_HOME) foram executados corretamente.

Para parar o Tomcat, execute o arquivo **shutdown.bat** (Windows) ou **shutdown.sh** (Linux) do diretório **<tomcat-home>/bin/** para iniciar o Tomcat.

Faça o download do arquivo **ScadaBR.war** (http://www.scadabr.org.br/?q=webfm).

Coloque este arquivo no diretório **<tomcat-home>/webapps/**.

Inicie o Tomcat.

Em um browser, entre com o endereço:

localhost:8080/ScadaBR/ (username: "admin" password: "admin").

2.3. Instalando o ScadaBR via Instalador

Para instalar o ScadaBR via instalador, faça o download do arquivo **ScadaBR-Win-Installer_<ver>.exe** e execute-o. Siga os passos clicando em *Next*.. Ao final dos passos, clique em *Install*.

Obs.1: Recomenda-se desinstalar todas os Tomcat já instalados na máquina antes de utilizar o instalador do ScadaBR.

Obs.2: No Windows Vista e no Windows 7, deve-se executar o instalador em modo de administrador, clicando com o botão direito sobre o arquivo e selecionando *Executar como administrador.*

3. Operando o ScadaBR

3.1. Apresentação

3.1.1. Menu principal

As funcionalidades nesta aplicação são acessadas pelos controles no cabeçalho. Dependendo das permissões de sua conta de usuário, vários ícones serão exibidos abaixo do logotipo da aplicação. Quando o cursor do mouse pairar sobre um ícone, será exibida um balão de texto com uma descrição resumida da funcionalidade daquele ícone. Além dos ícones de controle, no lado direito do cabeçalho é mostrado o nome de usuário que está logado no sistema.

Quando sua aplicação tiver um alarme ativo, um ícone de uma bandeira () ficará piscando e será associada uma descrição do alarme próximo ao centro do cabeçalho. A cor da bandeira indicará a severidade do alarme:

- Informação
- 🖣 Urgente
- ¶ Crítico
- Prisco de Vida

É possível clicar no ícone ou na descrição para exibir a lista de alarmes ativos.

3.1.2. Tipos de dados

São suportados cinco tipos de dados:

- Valores Binários (ou booleanos) podem ter apenas dois estados, referidos no sistema com os valores zero (0) e um (1). É possível utilizar conversores para exibir os valores binários em quaisquer rótulos necessários, como "ligado/desligado", "alto/baixo", "iniciado/parado", etc.
- Valores de **Estados Múltiplos** têm múltiplos estados distintos. Por essa abordagem, o tipo binário é um caso particular de um estado múltiplo. Valores são primitivamente representados como inteiros (ex. 0, 1, 2, 7, ...), mas, como nos valores binários é possível atribuir rótulos a cada valor como, por exemplo, "ligado/desligado/desativado", "aquecer/resfriar/desligado", ou quaisquer outros.

- Valores Numéricos (ou analógicos) são valores decimais representados no sistema com uma variável de ponto flutuante. Podem ser citados como exemplos de valores numéricos: temperatura, umidade, preço e altitude. Renderizadores de texto podem ser usados para determinar exibição de características como número de casas decimais, separação de milhar (com pontos ou vírgulas), exibição de sufixos (°F, kW/h, moles, etc.). Renderizadores de faixas podem ser usados para converter faixas de valores em rótulos de texto.
- Valores **Alfanuméricos** são simplesmente seqüências de caracteres, como a descrição O/S de uma fonte SNMP.
- Valores em Imagens são representações binárias de dados de imagens. São armazenados em arquivos no sistema de arquivos do servidor host (e não no banco de dados) e são guardados em memória. Renderizadores podem ser usados para criar imagens escaladas, como thumbnails, para exibição.

3.1.3. Data sources

Data sources (fontes de dados) são parte fundamental para a operação desta aplicação. Um **data source** é um "lugar" de onde os dados são recebidos. Virtualmente, qualquer coisa pode ser um data source, desde que o protocolo de comunicação seja suportado pela aplicação. Alguns exemplos:

- Se você tem uma rede Modbus acessível por RS232, RS485, TCP/IP ou UDP/IP, você pode criar um data source Modbus que irá "pollar" (poll) a rede em um intervalo definido.
- Se você tem equipamentos ou aplicações que podem enviar dados sobre HTTP, você pode iniciar um data source HTTP receiver que irá escutar conexões recebidas e enviar os dados aos pontos apropriados.
- Para hardware que suporta SNMP, inicie um data source SNMP. Valores poderão ser "pollados" em intervalos definidos, ou traps podem ser recebidos para reporton-exception.
- Dados podem ser lidos e atualizados em uma base de dados SQL externa ao sistema.
- Dados podem ser gerados randomicamente ou preditivamente usando um data source Virtual.

Valores de dados recebidos ou coletados por um data source são armazenados em data points.

3.1.4. Data points

Um **data point** é uma coleção de valores históricos associados. Por exemplo, um ponto particular pode ser uma leitura de temperatura de um quarto, enquanto outro ponto poderia ser a leitura de umidade do mesmo quarto. Pontos também podem ser valores de controle, como um indicador para ligar ou desligar um equipamento.

Existem muitos atributos que são usados para controlar o comportamento de pontos. Inicialmente existe o conceito de um **point locator**. *Locators* são usados por data sources para determinar como "achar" os dados para o ponto particular. Por exemplo, um data source SQL tem atributos incluindo onde achar a instância da base de dados; point locators para o data source indicam o nome da tabela e dos campos onde podem ser achados valores específicos. A separação lógica de data source e de data point dependem do protocolo de comunicação em questão.

Atributos de data points também podem determinar muitos outros aspectos do ponto, como seu nome, como deve ser registrado (todos os dados, apenas mudanças no valor, ou nenhum), por quanto tempo manter os dados, como formatar os dados para exibição e como traçar um gráfico com os valores.

Você também pode configurar data points com **detectores de valor**, que são usados para detectar condições de interesse nos valores dos pontos, como por exemplo, se o valor esteve muito alto por muito tempo, se é muito baixo, se muda com freqüência, se não muda, etc.

Pontos podem ser arranjados em um uma hierarquia, ou árvore, para simplificar sua gerência e exibição usando a funcionalidade de Hierarquia.

3.1.5. Monitoramento

Monitoramento de pontos dentro do sistema pode ser feito de duas maneiras. É possível usar uma **watch list** para criar listas dinâmicas de pontos com seus valores, últimos tempos de atualização, e gráficos de informações históricas (se a configuração do ponto permitir). Valores e gráficos são atualizados em tempo real sem ter que atualizar a janela do navegador. Gráficos de múltiplos pontos também podem ser exibidos sob demanda.

Também é possível criar **representações gráficas** de pontos usando a funcionalidade drag and drop para posicionar representações gráficas de pontos sobre uma imagem de fundo arbitrária. Imagens animadas podem ser usadas para criar uma visualização

altamente dinâmica do comportamento do sistema, e, como as watch lists, valores são representados em tempo real sem necessitar de atualização do navegador. Essas visualizações podem ser marcadas como "públicas" para que possam ser utilizadas em web sites públicos.

3.1.6. Controle

O controle de sistemas externos pode ser obtido para pontos que podem ser "setados" (ou que permitem escrita ou saída). Um ponto **setável** pode ser "setado" para um valor definido pelo usuário, como uma configuração de um termostato ou de um controle interruptor para um equipamento. Ambas watch lists e graphical views proporcionam meios simples para determinar a entrada de um valor. O point locator para um ponto "setável" determina como o data source define o valor no equipamento externo.

3.1.7. Eventos

Um **evento** é a ocorrência de uma condição definida no sistema. Existem tanto eventos definidos pelo sistema como definidos pelo usuário. Eventos definidos pelo sistema incluem erros de operação de data sources, logins de usuários, e inicialização e parada do sistema. Eventos definidos pelo usuário incluem detectores de valor, eventos agendados, e eventos compostos que detectam condições sobre pontos múltiplos usando argumentos lógicos. Há também os "eventos autidatos" que ocorrem quando usuários fazem alterações (adições, modificações e remoções) que afetam objetos em tempo de execução, incluindo data sources, data points, detectores de valor, eventos agendados, eventos compostos e tratadores de eventos.

Uma vez que um evento foi detectado, é manipulado por tratadores. Um **tratador de eventos** é um comportamento definido pelo usuário que deve ser executado quando um evento particular ocorre, como envio de email ou "setar" o valor a um ponto setável.

3.1.8. Ícones da Aplicação

- Data source
- Data point
- Marafico

- Set point
- Watch list
- Representação gráfica
- • Atualizar
- Secondary Secondary
 Secondary Secondary<
- W Evento agendado
- Pratadores de eventos
- 🗐 Alarme
- Suário
- P Comentário de usuário
- 🛂 Relatório
- I Listas de envio
- Rublisher
- Aviso
- Dogout

3.1.9. Sons de Alarmes

O ScadaBR pode executar sons quando alarmes estão ativos. Por definição, sons para alarmes são executados para alames urgentes, críticos e de risco de vida e não para alarmes de informação, mas os sons dos alarmes podem ser definidos individualmente. Para habilitar sons de alarmes particulares, arquivos válidos no formato mp3 devem ser colocados em:

- <ScadaBR_home>/audio/information.mp3
- <ScadaBR_home>/audio/urgent.mp3
- <ScadaBR_home>/audio/critical.mp3
- <ScadaBR_home>/audio/lifesafety.mp3

Para desabilitar um som, remova o arquivo de som associado. Uma coleção de sons pode ser achada na pasta **<ScadaBR_home>/audio/lib**.

3.1.10. Graficos

O ScadaBR contém uma pequena biblioteca de gráficos que pode ser achada na pasta <**ScadaBR_home>/graphics**. Cada sub-pasta contém todas as imagens daquela definição de imagem e um arquivo opcional de propriedades chamado **info.txt**. Este arquivo de propriedades contém pares nome/valor para os seguintes atributos (todos opcionais):

- **name**: O nome que será usado para descrever a imagem na interface de usuário. Se não fornecido, a imagem assume por padrão o nome da pasta.
- width: A largura da imagem. Por padrão, todas as imagens são do mesmo tamanho. Se não fornecido, a largura da primeira imagem será utilizada.
- **height**: A altura da imagem. Por padrão, todas as imagens são do mesmo tamanho. Se não fornecido, a altura da primeira imagem será utilizada.
- **text.x**: A posição de texto relativo ao limite esquerdo da imagem, em pixels. Se não fornecido, o valor padrão é 5.
- **text.y**: A posicão de texto relativo ao limite superior da imagem, em pixels. Se não fornecido, o valor padrão é 5.

Uma vez que é utilizada uma definição de imagem nas views a pasta não deve ser renomeada! O nome da pasta é usada internamente como o identificador da definição de imagem.

Arquivos de imagens são arranjados alfabeticamente por nome e são sensíveis à caixa. Pares nome/valor são separados por '='. Linhas iniciadas por '#' são consideradas comentários. Arquivos 'thumbs.db' são ignorados. Arquivos compactados (zip, gz, tar, etc) não podem ser usados por que os arquivos de imagens devem ser acessados pelo web server. As definições de imagens são carregadas na inicialização do sistema, então quaisquer alterações requerem reinicialização.

Para que as futuras versões do ScadaBR tenham mais gráficos, por favor forneça os gráficos que você criar para o projeto ScadaBR.

3.2. Adicionando Data Sources e Data Points

Este tópico mostra um exemplo de criação de Data Sources e Data Points. Para fins didáticos, criou-se um Data Source do tipo *Virtual Data Source*, que se trata de um data source simulado.

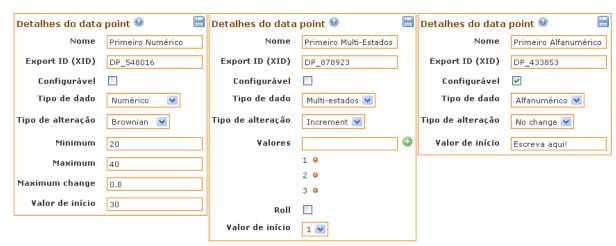
No menu principal, escolha a opção DataSources .

Selecione o tipo *Virtual Data Source* na lista e clique sobre o ícone *Adicionar* . Preencha como abaixo e salve suas configurações.



Ao salvar o data source, as opções para inclusão de data points serão habilitadas. Clique em *Adicionar* e preencha como abaixo e salve suas configurações.





Adicione mais alguns pontos de diferentes tipos.

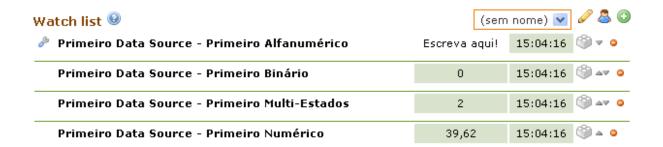
Agora, habilite todos os pontos clicando sobre os ícones em cada um deles e habilite também o data source.



3.3. Visualizando os dados: Watch List e Gráficos

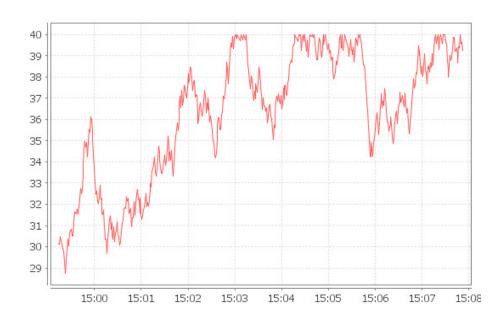
Agora que você possui cadastrado no sistema seus data sources e data points, basta visualizá-los. No menu principal, escolha a opção *Watch List* .

Os data points cadastrados encontram-se à esquerda. Para monitorar seus valores, clique sobre cada um para adicioná-lo à watch list atual.



Você pode acessar e editar mais informações a respeito de cada data point clicando em *Detalhes do data point* . Clique no ícone citado relativo ao data point numérico e observe seu histórico e o gráfico dos seus valores.

Históri	co		Mostrar	10 registro
Valor	Tempo	Comentário		
39,12	15:05:13			
38,94	15:05:12			
38,59	15:05:11			
38,72	15:05:10			
38,22	15:05:09			
38,02	15:05:08			
37,93	15:05:07			
38,31	15:05:06			
38,91	15:05:05			
38,92	15:05:04			



Ainda é possível editar as propriedades de renderização de texto.

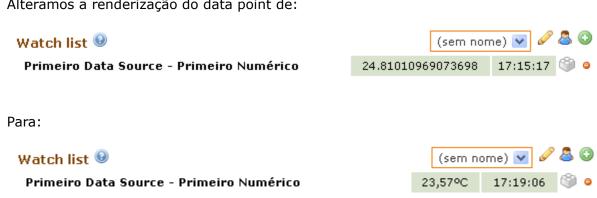
Em sua watch list, clique em *Detalhes do data point* .

Na próxima tela, clique em Editar data point 🦃.

Na área *Propriedades de renderização de texto*, tem-se diferentes opções de renderização. Por exemplo, com as configurações da figura abaixo:



Alteramos a renderização do data point de:



3.4. Definindo Eventos

Em sua watch list, clique em Detalhes do data point . Na próxima tela, clique em Editar data point 👺.



Na lista de Detectores de Eventos, selecione "Limite inferior" e clique em *Adicionar* apara adicionar um novo detector.

Preencha os campos como abaixo e clique no botão *Salvar* , no lado direito inferior da tela.



Pronto. Agora o sistema o informará (na tela de "Alarmes", no menu principal) toda vez que o valor de seu data point ficar abaixo do limite mínimo (22) por mais de 5 segundos.

3.5. Representações Gráficas

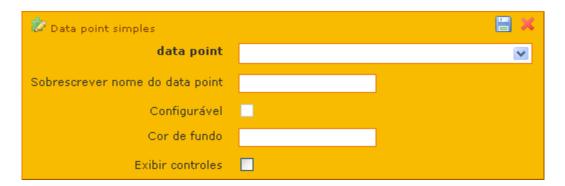
Para criar visualizações mais elaboradas dos dados, podemos construir "Representações Gráficas". No menu principal, escolha a opção *Representação gráfica* ...

Clique em Nova representação ...

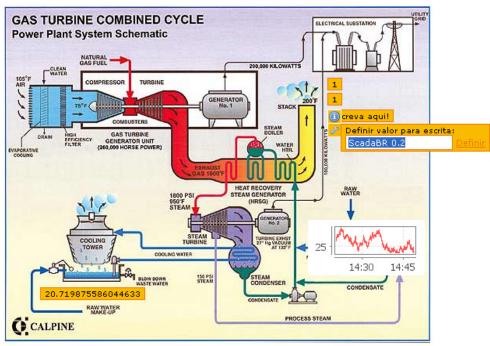
Escolha um nome para sua primeira representação, clique sobre *Escolher arquivo*, selecione uma imagem, confirme e clique em *Fazer upload de imagem*.

Uisualizar propriedades 🚇				
Nome	Primeira Representaçã			
Export ID (XID)	<u>GV_</u> 914596			
Imagem de fundo	Escolher arquivo powerplant.jpg			
Fazer upload de imagem Limpar imagem				
Acesso anônimo none 🕶				

Na lista de componentes, selecione *Data point simples* e clique em *Adicionar componente à representação* apara adicioná-lo.



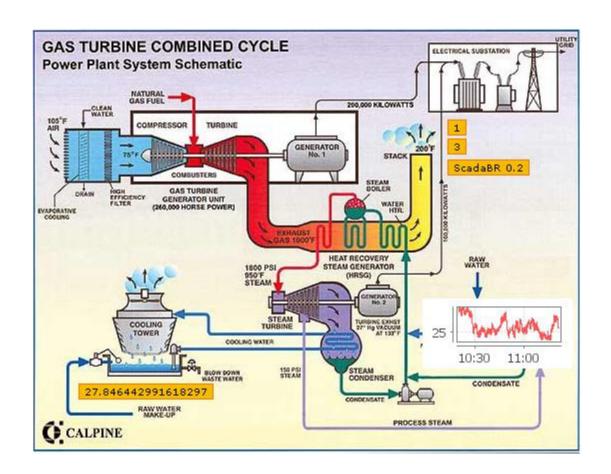
Navegue até o ícone adicionado sobre a figura e clique sobre a segunda opção: *Editar configurações de componente de ponto*.



Escolha um dos seus pontos cadastrados na lista e salve suas configurações.

Adicione outros componentes, associe aos seus data points cadastrados e posicione-os sobre a tela.

Clique sobre o botão "Salvar".



4. Data Sources

4.1. Modbus

4.1.1. Visão geral

O data source Modbus é utilizado para adquirir dados tanto de um equipamento modbus acessível por meio de uma rede I/P – podendo estar em uma rede local ou intranet, ou então estar em qualquer lugar na internet – (Modbus IP), quanto de uma rede local modbus acessível via RS232 ou RS485 (Modbus Serial). O mecanismo de controle de acesso é do tipo mestre-escravo ou Cliente-Servidor.

4.1.2. Configuração básica

O campo **Nome** define o nome do data source, e pode ser qualquer descrição. O **Período de atualização** determina a frequencia com que são solicitados os dados do equipamento Modbus. O campos **Timeout** e os **Retries** determinam o comportamento do sistema no caso de uma solicitação falhar. O data source aguarda por uma resposta pelo tempo (em milisegundos) definido como **timeout**. Se não receber resposta, o data source tentará novamente quantas vezes estiver definido em **retries**.

A opção **Contiguous batches only** pode ser usada para especificar que a implementação de modbus não deve tentar otimizar solicitações por valores diferentes numa única solicitação. Ativando esta opção fará com que a implementação somente realize solicitações por valores múltiplos quando estes valores formarem um espaço contínuo no registrador.

Pode-se scanear nodos escravos na rede utilizando a funcionalidade **Node scan**. Esta funcionalidade itera sobre os escravos de 1 a 240, enviando a cada um um ReadExceptionStatus (código de funcionamento 7). Se uma resposta for recebida (considerando as configurações de timeout e retries), o nodo é considerado disponível. Note que nem todos os equipamentos suportam este código de funcionamento, então falsos-negativos são possíveis.

O campo **Create slave monitor points** faz com que "monitoradores de escravos" sejam criados automaticamente. Um "monitorador de escravo" é um ponto binário que

indica o atual estado de um escravo. Se uma solicitação de poll a um escravo falhar devido ao timeout ou a um erro, o escravo é considerado *off line*. Estes pontos podem ser utilizados para controle como qualquer outro ponto.

Modbus IP

Três **Transport types** são suportados. Verifique a documentação do seu equipamento para determinar qual configuração pode ser usada.

A configuração **TCP** usa um novo socket TCP a cada poll. O número de retries definido aplica-se quando ocorrem exceções na conexão durante a abertura do socket. O timeout usado para a conexão depende de se a pilha TCP está sendo usada.

A configuração **TCP with keep-alive** cria uma conexão TCP no primeiro poll, o qual é deixado aberto para re-uso. Se a conexão é fechada por algum motivo, uma nova é criada quando necessário. O comportamento para exceções na conexão para esta opção é a mesma para o **TCP**. Esta é a configuração recomendada para a maioria dos usuários, visto que ela provê uma comunicação eficiente e robusta enquanto evita muitos dos problemas de configuração típicos ao **UDP**.

A configuração **UDP** utiliza pacotes UDP para comunicação. Esta configuração provê a maior eficiente de rede, mas tipicamente requer uma configuração maior da rede, visto que tanto o ScadaBR quanto o equipamento precisam ser visíveis na rede (em oposição ao TCP, no qual o ScadaBR pode estar protegido por uma firewall.

As configurações de **Host** e **Port** determinam como encontrar o equipamento Modbus na rede. O host pode ser o nome de um domínio ou um endereço IP.

Modbus Serial

A comunincação Serial é controlado com os valores de **Baud Rate**, **Flow control in**, **Flow control out**, **Data bits**, **Stop bits**, and **Parity**. O **Echo** pode ser usado com a rede RS485 se for apropriado.

O valor de **Encoding** determina como as solicitações Modbus são formatadas. A maioria parte dos equipamentos utilizam mensagens formatadas em RTU. Verifique a documentação do seu equipamento Modbus para determinar como configurar este campo.

4.1.3. Configuração do data point

Ambos, serial e redes IP, utilizam os mesmos atributos de ponto. O **Slave id** é o id com o qual o nodo Modbus foi configurado e é um número entre 1 e 240.

O **Register range** determina em qual dos quatro *ranges* o valor será encontrado. Consulte a documentação do seu equipamento para determinar qual deve ser usado.

- **Coil status** representa o *range* hexadecimal de 0x00000 a 0x0FFFF. Cada registrador contém um único e configurável bit. Valores neste *range* são sempre Binários.
- Input status representa o range hexadecimal de 0x10000 a 0x1FFFF. Cada registrador contém um único e read-only bit. Valores neste range são sempre Binários.
- **Holding register** representa o *range* hexadecimal de 0x40000 a 0x4FFFF. Cada registrador possui 2 bytes (ou uma "palavra") e é configurável. Valores neste *range* podem ser Binários ou Numéricos dependendo das demais configurações.
- **Input register** representa o *range* hexadecimal de 0x30000 a 0x3FFFF. Cada registrador possui 2 bytes (ou uma "palavra") e é *read-only*. Valores neste *range* podem ser Binários ou Numéricos, dependendo das demais configurações.

Valores de **Coil status** e **Input status** são sempre Binários. Entretanto, fornecedores de Modbus são frequentemente muito criativos nas maneiras com que os **Holding** e **Input registers** são usados. O campo **Modbus data type** reflete as várias maneiras nas quais os dados podem ser codificados. Consulte a documentação de seu equipamento Modbus para determinar a configuração apropiada.

Valores específicos estão localizados na configuração do **Offset**. Este campo é um valor *O-indexed*, o que significa que sua contagem inicia em 0. Alguns fornecedores de Modbus fornecem documentação no qual ele é *1-indexed*, no qual a contagem inicia em 1. Portanto, às vezes é necessario subtrair 1 do índice documentado para determinar o offset *O-indexed*. Quando os registradores incluem seus *ranges*, como por exemplo, escrito como 0x30001, *1-indexed* está tipicamente implícito. O campo **Bit** é usado quando valoers binários são codificados em bits de registradores individuais.

O campo **Settable** pode ser usado para tornar um ponto, que normalmente seria configurável (de acordo com seu *range*), não configurável.

Os campos Multiplier e Additive podem ser usados quando conversões triviais de

valores são necessárias. Valores numéricos lidos da rede são calculados da seguinte maneira: (valor bruto) * multiplier + additive. O contrário é aplicado quando um valor numérico é escrito na rede.

4.2. OPC DA

4.2.1. Visão geral

A especificação **OPC DA** (OPC Data Access) é a primeira de um grupo de especificações conhecido como **OPC Specifications**, e nasceu da colaboração de diversos líderes mundiais em fornecimento de equipamentos de automação trabalhando em cooperação com a Microsoft.

Originalmente baseado nas tecnologias **OLE COM** (*component object model*) e **DCOM** (*distributed component object model*), a especificação define um conjunto padrão de objetos, interfaces e métodos para utilizar em aplicativos de controle de processos e de automação de manufaturas para facilitar a interoperabilidade. As tecnologias COM/DCOM fornecem o *framework* para os softwares serem desenvolvidos. Atualmente existem centenas de servidores e clientes OPC DA.

4.2.2. Configuração básica

O campo **Nome** define o nome do data source, e pode ser qualquer descrição. O **Período de atualização** determina a frequencia com que são solicitados os dados do servidor OPC DA.

Os campos **Host**, **Domínio**, **Usuário** e **Senha** definem o endereço e as "credenciais" do servidor a ser utilizado. Após preencher estes campos, clique em *Atualizar*, e uma lista com os servidores disponíveis no endereço definido será criada.

4.2.3. Configuração do data point

Após selecionar o servidor OPC DA, clique em *Listar Tags* na caixa à direita das configurações do data source. As tags disponíveis no servidor serão listadas, juntamente com seus tipos, se são configuráveis ou não e com um *checkbox* na coluna Add. Para adicioná-los como data points, basta dar *check* nas tags desejadas e em seguida

clicar em *Adicionar Tags*. As tags selecionadas irão aparecer numa caixa abaixo das configurações do data source, podendo ser habilitadas ou desabilitadas.

Obs.1: Antes de adicionar as tags, é necessário salvar o data source.

Obs.2: Tags adicionadas que já foram previamente adicionadas, aparecerão na lista duplicadas.

5. Relatórios

5.1. Introdução

O ScadaBR possui um gerador de relatórios próprio, além de ser compatível com os principais geradores de relatórios customizados.

Neste item serão abordadas as duas maneiras de gerar relatórios a partir do ScadaBR.

5.2. Gerando relatórios no ScadaBR

5.2.1. Configurando um novo modelo de relatório

No menu principal, escolha a opção *Relatórios* .

Modelos de relatórios proporcionam uma definição de como criar instâncias de relatórios.

Para adicionar um novo modelo de relatório, clique em *Novo relatório* .



Nesta tela, deve-se definir o modelo do relatório:

O **Nome** do modelo é usado como uma referência visual do modelo. Recomenda-se a utilização de um nome único para cada modelo.

Utilize a lista de **Data points** para selecionar os pontos que devem ser incluídos no relatório. Para adicionar data points, selecione o ponto desejado na lista e clique no ícone . Para remover um ponto existente, clique no ícone ⁹ associado ao ponto.

O valor **Faixa de datas** é utilizado para determinar uma faixa de tempo utilizada para filtrar valores que serão utilizados no relatório.

5.2.2. Agendamento de Relatórios

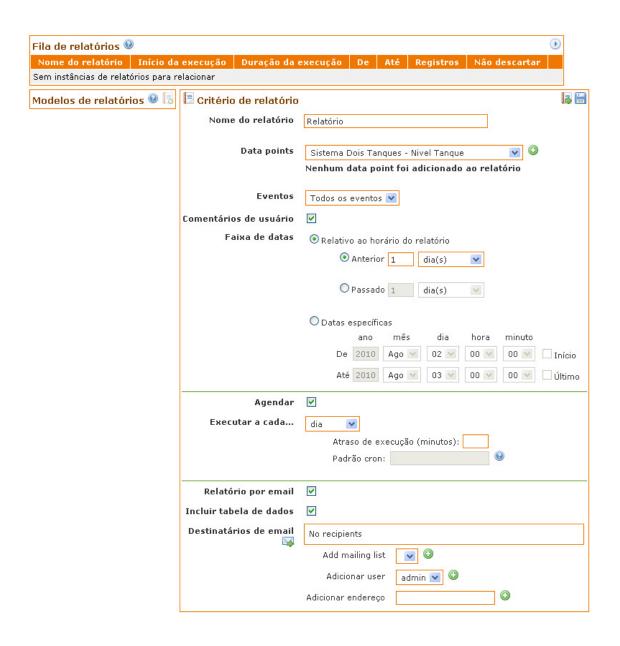
Relatórios podem ser "agendados" para serem gerados automaticamente. Use a seleção Executar a cada... para determinar um simples evento de tempo a partir do qual o relatório será gerado, ou defina uma rotina de acordo com o padrão cron para controle mais específico (veja a documentação de "Cron patterns" para mais informação). Eventos de tempo ocorrem no início do período compreendido. Um Atraso de execução pode ser aplicado se os dados esperados para o relatório tendem a serem coletados após o tempo de execução absoluto do relatório.

5.2.3. Envio de relatórios por e-mail

Apesar dos relatórios não poderem ser explicitamente compartilhados pelo sistema, podem ser implicitamente compartilhados por meio da criação de uma lista de envio de emails para a qual o sistema enviará instâncias de relatórios gerados. O conteúdo desse email é o mesmo da janela "gráfico de relatório" aberta no painel de Relatórios. Para incluir o arquivo de exportação com o formato CSV no email, selecione a caixa **Incluir tabela de dados**.

Selecione os **Destinatários de email** para os quais enviar o email com o relatório. Os destinatários podem ser listas de discussão, usuários do sistema ou endereços de email digitados. Clique no ícone *Enviar e-mail de teste* para enviar uma mensagem de teste para os destinatários selecionados.

Importante: Instâncias de relatórios enviadas por email são automaticamente apagadas depois de serem enviadas.



5.2.4. Gerenciamento de modelos

Para salvar um modelo de relatório clique no ícone no canto direito superior do painel de criação de modelos. Para apagar um modelo existente, clique no ícone • . Para gerar um relatório imediatamente a partir de um modelo, clique em "Executar agora". Note que alguns modelos de relatórios podem incluir uma grande quantidade de informações e, por causa disso, podem levar algum tempo para serem gerados. Assim, todos os relatórios são gerados de maneira assíncrona da interface de usuário. Quando um ícone é clicado, a instância de relatório será incluída em uma fila onde o usuário pode monitorar seu progresso.

5.2.5. Fila de relatórios

A Fila de relatórios lista todas as instâncias de relatórios para o usuário em questão. Esta lista não é atualizada automaticamente. Uma atualização pode ser invocada ao clicar em ①. Alguns eventos, entretanto, também irão invocar uma atualização desta lista, como a geração de um relatório.

Instâncias de relatórios em qualquer estado serão listadas, estejam completos ou em execução. As colunas da lista irão refletir o estado atual da instância.

As colunas da lista têm as seguintes definições:

- **Nome do relatório**: o nome herdado do modelo de relatório. Uma vez criada a instância, o nome do relatório não é alterado mesmo se o nome do modelo mudar.
- Início da execução: o momento de tempo em que a instância começou a ser gerada.
- Duração da execução: quanto tempo a geração da instância consumiu.
- **De**: o tempo a partir do qual os registros foram selecionados..
- Até: o tempo até o qual os registros foram selecionados.
- **Registros**: o número total de registros selecionados.

A caixa **Não Descartar** permite que usuários previnam que instâncias de relatórios importantes sejam descartadas (veja a documentação "Outras configurações" para mais informações a respeito do processo de descarte de relatórios). Esta funcionalidade deve apenas ser usada quando estritamente necessária, pois as instâncias de relatórios podem consumir quantidades consideráveis de armazenamento.

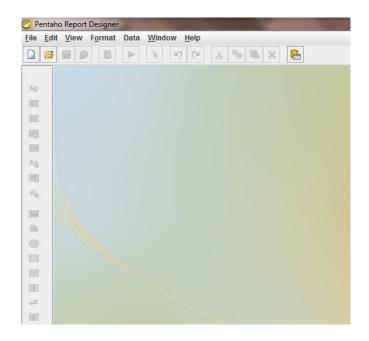
A coluna final da tabela oferece controles para o gerenciamento das instâncias. O ícone inicia o download de um arquivo CSV com os dados do relatório, para importação em softwares de planilha eletrônica. O ícone informações da instância de relatório, dados estatísticos e uma imagem do gráfico dos dados do relatório. Clicando no ícone instância do relatório é apagada. Instâncias de relatório particularmente grandes podem levar algum tempo para serem apagadas.

5.3. Gerando relatórios utilizando softwares de terceiros.

5.3.1. Pentaho: Configurando a base de dados do ScadaBR

Após fazer o download do RPD, basta descompactar o arquivo e executar o arquivo report-designer.bat no Windows ou o report-designer.sh no Linux que estará dentro da pasta descompactada.

A tela inicial do programa ira abrir.

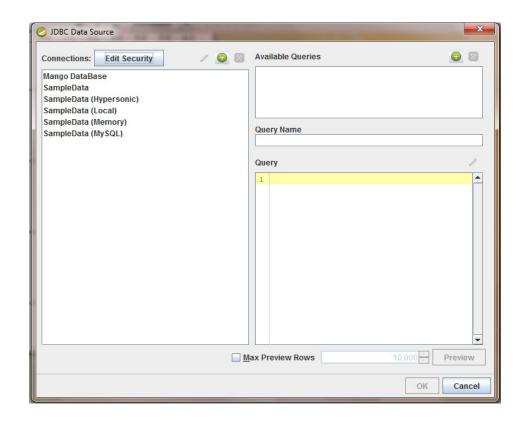


Vá em File -> New, para criar um novo Report.

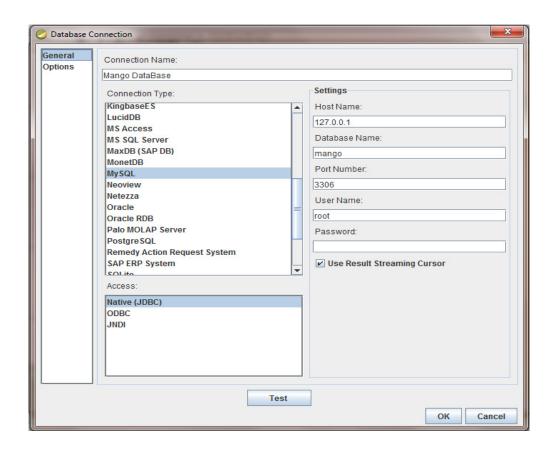
Agora é hora de criar um *DataSource*. O *DataSource* será a fonte dos dados utilizados no *Report*.

Neste exemplo, será utilizado o MySQL como DataSource. Para utilizar utilizar o Apacha Derby, basta trocar a $Connection\ Type$.

Então vá em *Data* □ *Add Data Source* □ *JDBC*



Agora, criamos uma nova Connection, basta clicar no ícone "+" ao lado de Edit Security



Configure conforme figura acima e clique em *OK*.

Pronto, agora está tudo configurado e a base de dados já pode ser utilizada. A configuração do *Report* é drag-and-drop, ou seja, basta arrastar e soltar.

Desse modo, basta escolher o que deseja adicionar na paleta, arrastar e soltar no *Report*.

5.3.2. Pentaho: Exemplo de criação de um relatório

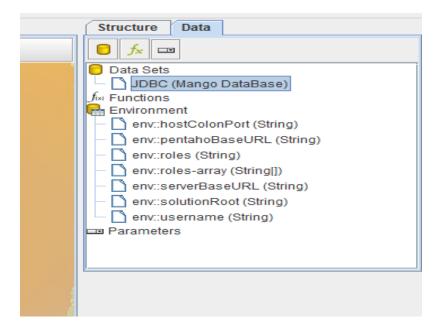
Neste tópico é apresentado um passo-a-passo para criar um Report básico:

Criar um Novo Report

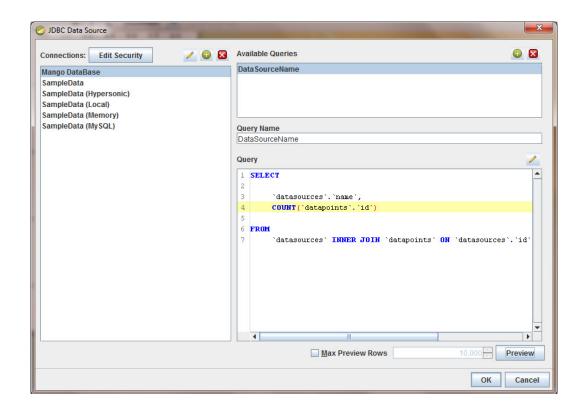
Configurar o *MySQL* como *DataSource* usando a base de dados do ScadaBR como colocado anteriormente

Criar as Queries que serão utilizadas no andamento do Report:

As Queries são as tabelas que estarão disponíveis do banco de dados . Para criar uma nova Query, verique ao lado direito da Janela Principal e vá na aba "Data":

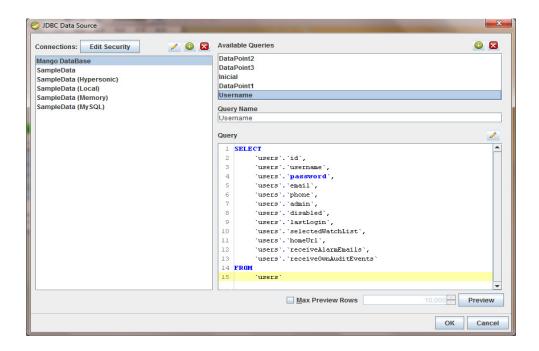


Dê dois cliques no *Data Set* selecionado e então adicionamos uma *Query*. A primeira *Query* que criamos é para pegar o nome do *DataSource* e a quantidade de *DataPoints* pertencente ao *DataSource*:

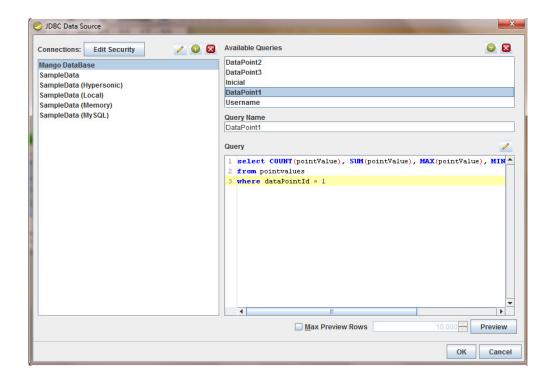


Clique em OK.

A segunda é para pegar os dados dos usuários do sistema ScadaBR:



E, por fim, a terceira é para pegar os dados dos DataPoints:



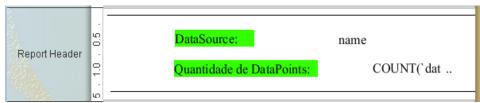
A *Query* anterior foi criada para o *DataPoint* 1, porém é criada mais duas Queries para os outros *DataPoints* apenas mudando na *Query* o número do data point desejado em: where dataPointId = <DataPointDesejado>

A configuração do layout do Report se dá da seguinte maneira:

O Layout é subdividido em: *Page Header, Report Header, Details, Report Footer, Page Footer.* Iniciamos com o *Page Header*.



Nessa parte adicionamos um "label" para escrever "Report" e um "image" para colocar a imagem do ScadaBR. Para adicionar bastar puxar o item desejado da paleta e soltar no local desejado.



Report Header

Nessa parte adicionamos dois "label" um para "DataSource:" e outro para "Quantidade de DataPoints:" e selecionamos da Query anteriormente criada o nome do DataSource e a quantidade de datapoints.

Details



Nessa parte adicionamos três Sub-Report. O Sub-Report é adicionado porque em cada Report é permitido o uso de somente uma Query, então para utilzarmos mais Queries adicionamos Sub-Reports.

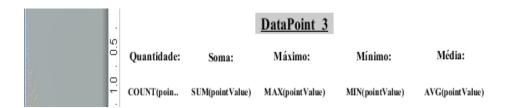


Nesse Sub-Report são adicionados 5 "label" : Quantidade, Soma, Máximo, Mínimo e



Média. Também são adicionados dados da Query DataPoint1.

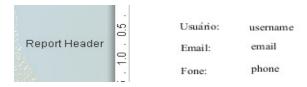
Nesse *Sub-Report* são adicionados 5 "label" : Quantidade, Soma, Máximo, Mínimo e Média. Também são adicionados dados da Query DataPoint2.



Nesse Sub-Report são adicionados cinco "label": Quantidade, Soma, Máximo, Mínimo e Média. Também são adicionados dados da Query DataPoint3.

Report Footer

Nessa parte adicionamos um Sub-Report.



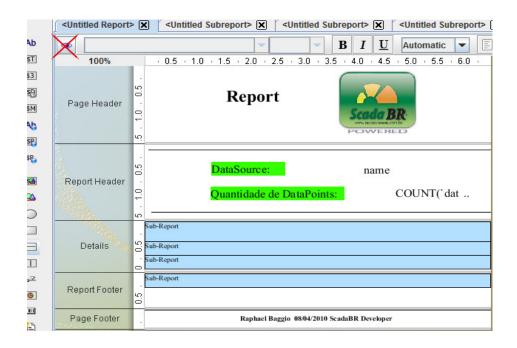
Nesse *Sub-Report* são adicionados três "label": Usuário, Email e Fone. Também são adicionados dados da Query Username.



Page Footer

Nessa parte adicionamos um "label".

Para visualizar um preview do relatório, clique no ícone destacado com um X.

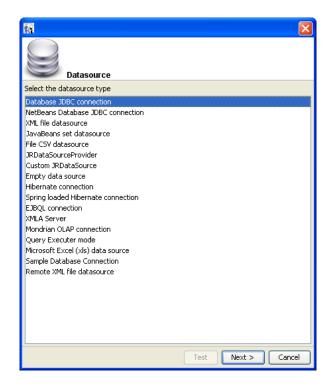


5.3.3. iReport: Configurando a base de dados do ScadaBR

Obs.: Se você estiver utilizando o Derby, é necessário adicionar a lib **derby.jar** (**ScadaBR\WebContent\WEB-INF\lib\derby.jar**) ao Classpath do iReport, em Ferramentas > Opções, na aba Classpath, *Add JAR*. Selecione a opção *Report Datasources*.



Clique em *New*. Escolha a opção Database JDBC connection.



Defina um nome (p.ex. ScadaBR).

- Caso utilize MySQL

Em JDBC Driver, escolha "MySQL(com.mysql.jdbc.Driver)" Em JDBC URL, digite: jdbc:mysql://localhost/mango

- Caso utilize Derby

Em JDBC Driver, digite: org.apache.derby.jdbc.EmbeddedDriver

Em JDBC URL, digite: jdbc:derby:<localização do seu tomcat>/bin/mangoDB

Deixe os campos JDBC URL Wizard em branco.

Preencha os campos de Username e Password de acordo com a configuração de seu banco de dados.



Para verificar se você configurou corretamente, clique em *Test*. Caso a mensagem *Connection test successful* seja mostrada, a base já está pronta para ser utilizada.

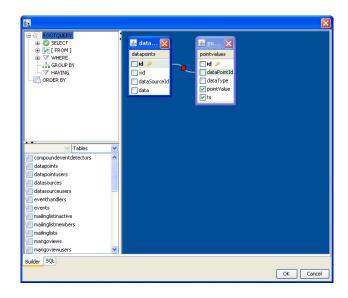
5.3.4. iReport: Selecionando os dados da base a serem utilizados

Selecione Report query

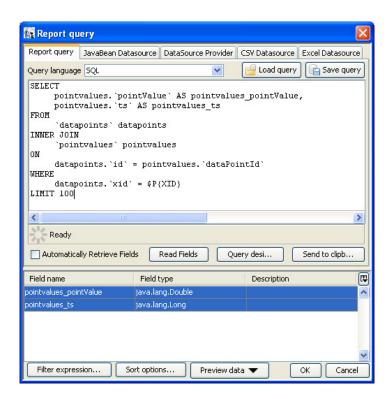


Na aba Report query, você pode digitar diretamente a query que desejar, ou clicar no botão *Query...*.

Em *Query...*, você pode montar a query visualmente, escolhendo as tabelas, criando cláusulas e selecionando os campos que deseja obter.



Montada a query, clique em *Read Fields*. Os campos selecionados irão aparecer na tabela abaixo, com seus tipos já definidos.



Você pode clicar em *Preview data* para verificar se os dados retornados são realmente aqueles que deseja.

Clique em OK, e os campos serão adicionados a lista de *Fields* do relatório, podendo então ser utilizados.

6. Scripts

6.1. Introdução

Para criar scrips no ScadaBR, utiliza-se o Meta Data Source. Ele tem esse nome por sua capacidade de combinar pontos existentes em novos. Ao invés de obter sua informação de uma fonte externa, utiliza valores de outros pontos e permite manipulação de maneiras arbitrárias pelo usuário.

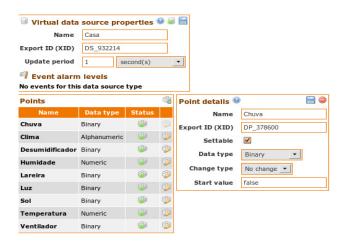
Para conhecer mais funcionalidades de scripts no ScadaBR, busque a *Ajuda*, no menu Meta Data Source.

Segue abaixo um exemplo de como criar um script básico.

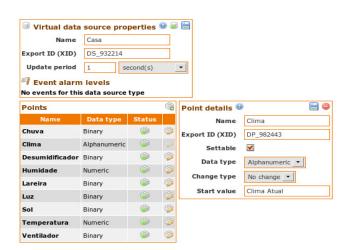
6.2. Criando o Data Source e os Data Points

Neste exemplo, criamos 1 Virtual Data Source: Casa e 9 Data Points: Chuva, Clima, Desumidificador, Humidade, Lareira, Luz, Sol, Temperatura e Ventilador.

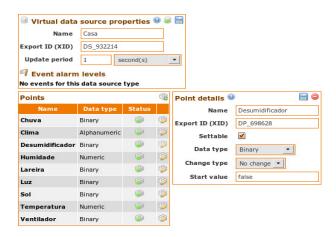
Data Point Chuva



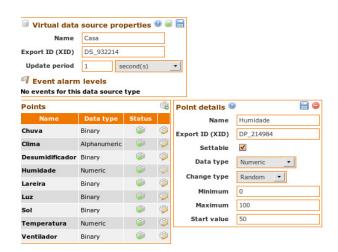
Data Point Clima



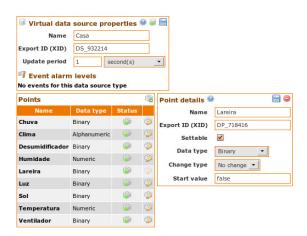
Data Point Desumidificador



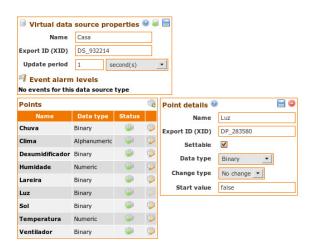
Data Point Humidade



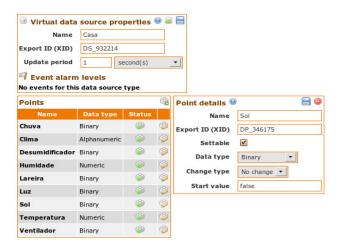
Data Point Lareira



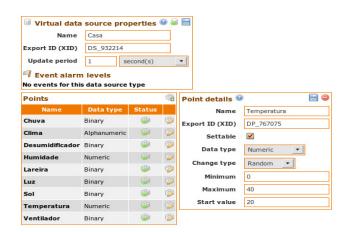
Data Point Luz



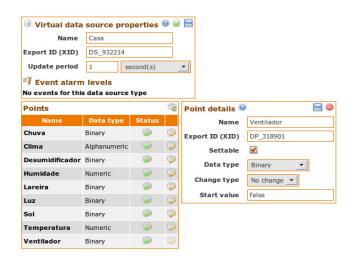
Data Point Sol



Data Point Temperatura



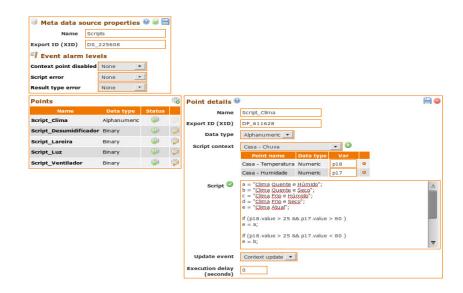
Data Point Ventilador



6.3. 6.3. Criando um Meta Data Source e seus Data Points

Agora vamos adicionar os Scripts. Para isso, adicionamos 1 Meta Data Source e 5 Data Points : Script_Clima, Script_Desumidificador, Script_Lareira, Script_Luz e Script_Ventilador.

Script_Clima



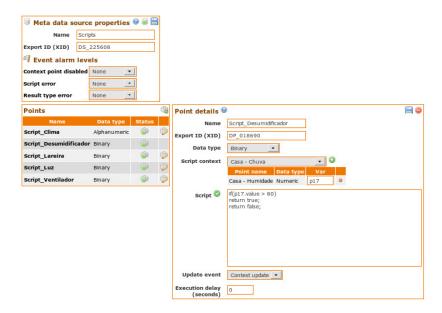
EmTipo de Dado, escolha Alfanumérico. Em Script Context, escolha os datapoints Casa – Temperatura e Casa – Humidade e então clique no ícone para adicioná-los ao script. No campo Var, escolhe-se o nome pelo qual o data point será referenciado no script. Neste exemplo, definiu-se temp_var e humi_var para a temperatura e humidade respectivamente.

No campo Script insira o seguinte código :

```
a = "Clima Quente e Húmido";
b = "Clima Quente e Seco";
c = "Clima Frio e Húmido";
d = "Clima Frio e Seco";
e = "Clima Atual";
if (temp_var.value > 25 && humi_var.value > 80 )
```

```
e = a;
if (temp_var.value > 25 && humi_var.value < 80 )
e = b;
if (temp_var.value < 25 && humi_var.value > 80 )
e = c;
if (temp_var.value < 25 && humi_var.value < 80 )
e = d;
return e;</pre>
```

Script_Desumidificador

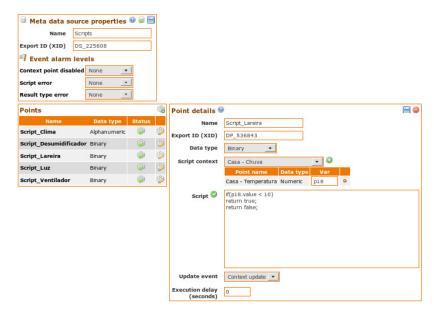


Em*Tipo de Dado*, escolha *Binario*. Em *Script Context*, escolha os datapoints *Casa – Humidade* e então clique no ícone para adicioná-lo ao script. No campo *Var*, definiuse humi_var para a humidade.

No campo Script insira o seguinte código:

```
if(humi_var.value > 80)
return true;
return false;
```

Script_Lareira

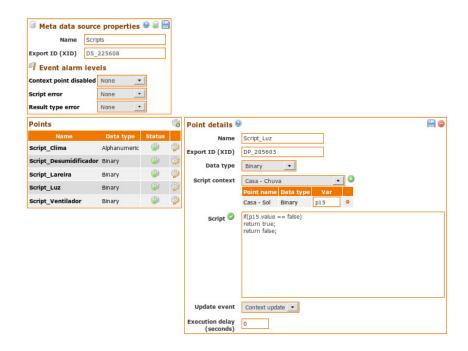


Em*Tipo de Dado*, escolha *Binario*. Em *Script Context*, escolha os datapoints *Casa – Temperatura* e então clique no ícone para adicioná-lo ao script. No campo *Var*, definiu-se temp_var para a humidade.

No campo Script insira o seguinte código:

```
if(temp_var.value < 10)
return true;
return false;</pre>
```

Script_Luz

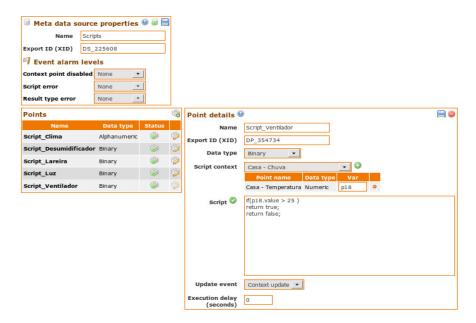


Em*Tipo de Dado*, escolha *Binario*. Em *Script Context*, escolha os datapoints *Casa – Sol* e então clique no ícone para adicioná-lo ao script. No campo *Var*, definiu-se sol_var para a humidade.

No campo Script insira o seguinte código:

```
if(sol_var.value == false)
return true;
return false;
```

Script_Ventilador



Em*Tipo de Dado*, escolha *Binario*. Em *Script Context*, escolha os datapoints *Casa – Temperatura* e então clique no ícone para adicioná-lo ao script. No campo *Var*, definiu-se temp_var para a humidade.

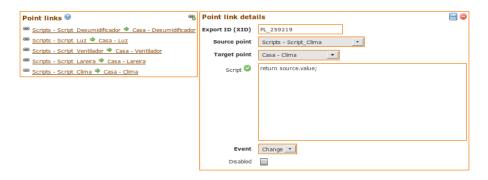
No campo Script insira o seguinte código :

```
if(temp_var.value > 25 )
return true;
return false;
```

6.4. Criando Point Links

Depois de inseridos os Data Points e os Scripts vamos inserir os Point Links. Eles irão fazer a "ligação" entre a saída do Script com o Data Point em que se quer atuar. Assim, a entrada do Point Link é o Script e a saída é o Data Point desejado.

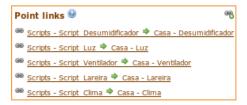
Foram adicionados 5 Point Links, 1 para cada Script, como por exemplo, para "Clima" e "Script_Clima":



OBS: Em todos os Point Links no campo Script é adicionado:

return source.value;

Depois de todos adicionados a página de Point Links fica assim :



Pronto, agora basta verificar o funcionamento na Watch List. Verifique que com a mudança dos estados de alguns Data Points outros são alterados.

7. Watchdog

7.1. Como funciona

O watchdog verificará periodicamente o status do ScadaBR por meio do método *getStatus* de sua API SOAP. Caso o ststus indique alguma falha, o sistema reiniciará o Tomcat a fim de corrigi-la.

O watchdog finalizará após um número especificado de tentativas consecutivas de reiniciar o Tomcat.

Obs.: Caso o sistema operacional da máquina seja Microsoft Windows, é necessário definir a variável de ambiente **CATALINA_HOME**, que aponta para o diretório onde se encontra o ScadaBR. Por exemplo, se o ScadaBR estiver instalado no diretório **C:\Arquivo de Programas\ScadaBR**, a variável **CATALINA_HOME** seria definida como:

set CATALINA_HOME=C:\\Arquiv~1\\ScadaBR

7.2. Configuração

Para configurar o watchdog, edite o arquivo **config.properties**. Todas as configurações possuem valores padrão que serão utilizados caso o arquivo ou o parâmetro em questão não seja especificado.

- 1. Configure o watchdog no arquivo config.properties. Todas as configurações possuem valores padrão que serão utilizados caso o arquivo ou o parâmetro em questão não seja especificado.
 - api.address: Endereço da API do ScadaBR.
 - catalina.home: Pasta base da instalação do Tomcat onde roda o ScadaBR. Caso não seja fornecido, o watchdog utilizará a variável de ambiente CATALINA_HOME (caso esteja definida).
 - watchdog.period: Período em segundos em que o watchdog verificará o status do ScadaBR. Seu valor mínimo é de 30 segundos e seu valor padrão é de 60 segundos.
 - watchdog.retries: Número máximo de tentativas consecutivas com que o watchdog reiniciará o Tomcat a fim de recuperar o ScadaBR.

• watchdog.fileLogging: para habilitar ou desabilitar o log em arquivo (log.txt).

7.3. Como executar

Caso tenha instalado no diretório padrão do ScadaBR não é necessário editar o config.properties (arquivo de configuração do watchdog).

Caso tenha instaldo em outro diretório por favor edite o arquivo config.properties com a seguinte mudança:

Mude o caminho da catalina.home com o local onde foi instalado o ScadaBR:

Exemplo do caminho para diretório padrão:

catalina.home=C:\\ScadaBR

IMPORTANTE : Não esqueça de deixar duas "backslashes" entre os nomes do endereço! Para iniciar o watchdog abra um prompt de comando na pasta onde foi instalado o ScadaBR e digite:

java -jar watchdog

IMPORTANTE:

- Em Windows, nunca esqueça de deixar duas "backslashes" entre os nomes do endereço:
 - Usar **CATALINA_HOME** sem espaços
 - Usar JAVA_HOME sem espaços

8. Dicas

8.1. Como alterar o banco de dados?

O ScadaBR atualmente suporta dois sistemas gerenciadores de banco de dados: MySQL e Derby.

O arquivo **ScadaBR.war** vem por padrão configurado com o Derby por questões de facilidade para o usuário, pois o mesmo roda embarcado e não exige a instalação do SGBD ou a configuração de parâmetros como login e senha.

A alteração dos dados referentes ao banco utilizado é prevista para ser realizada durante a "construção" (building) do sistema, porém, caso seja necessário, pode-se alterá-las de uma outra maneira a partir do arquivo war disponibilizado:

Descompactar o arquivo **ScadaBR.war** para um diretório qualquer <scadabr>. (pode-se renomear o arquivo para **ScadaBR.zip** e então extraí-lo, caso desejado).

Abra com um editor de texto o arquivo:

<scadabr>/WEB-INF/classes/env.properties e o edite conforme desejado:

MySQL*:

db.type=mysql db.url=jdbc:mysql://localhost/scadabr db.username=root db.password= db.pool.maxActive=10 db.pool.maxIdle=10

* Atenção: para utilizar o MySQL, um banco com o nome "scadabr" deve ser criado antes de iniciar o ScadaBR. (apenas o banco, as tabelas são criadas automaticamente pelo sistema)

Derby (padrão):

db.type=derby
db.url=<pasta do bd>**

db.username=
db.password=

** A pasta onde ficará o banco de dados. Pode ser um caminho relativo, como: ../scadabrDB/.

Salve o arquivo, compacte toda a pasta <scadabr> novamente em um arquivo **ScadaBR.war** e realize a instalação.

9. Glossário

Α

- A&E "Alarms and Events" Alarmes e Eventos.
- Alarmes Ocorrências, usualmente assíncronas, de uma medição com valor fora de limites pré-estabelecidos, ou eventos que necessitem atenção de um operador (por exemplo, visualizar e "reconhecer" o alarme). Alarmes usualmente possuem um grau de criticidade associado, por exemplo de "0 = warning" até "5 = critical alert"
- **API** Application Programming Interface, um conjunto de especificações que permite que diferentes softwares (ou diferentes módulos de um software) comuniquem-se entre si.

В

- Batch Conhecido também como Receita ou Batelada, é uma seqüência de ações a serem executadas sequencialmente para obter um determinado resultado em processo de manufatura. Por exemplo: uma receita de produto químico especificando passos de reações, destilação, enchimento e esvaziamento de tanques etc. Nesse caso, cada "composto" ou volume de produção diferente possui uma receita diferente.
- Builder ver HMI Builder

C

- Configurador no contexto de sistemas SCADA estamos chamando de "configurador" a um aplicativo, com interface de usuário (não apenas arquivos de configuração), que permite definir parâmetros de comunicação dos protocolos, configurações de "tags", períodos de amostragem dos registradores e assim por diante.
- **Custom App** (ou aplicação personalizada) qualquer aplicativo externo ao SCADA que comunique-se com o mesmo obtendo dados ou enviando comandos através de uma API padronizada. Por exemplo, uma nova HMI, um historian avançado, ou alguma aplicação de terceiros como um ERP, BI (Business Intelligence) e assim por diante.

D

• **DA** - Data Access - nome comum que se dá à leitura de tags ou envio de comandos. "DA" usualmente ocorre em alto nível, em contraste com o IO que

- ocorre em baixo nível; desta forma, o DA referencia variáveis físicas através do nome da TAG, e não em termos de registradores ou endereçamento de barramentos.
- **DA Server** Software que traduz as solicitações de alto nível em requisições para os drivers de protocolos; o DA server pode ter funções adicionais como "cache" (manter o último valor em memória) ou "buffer" (permitindo IO nãobloqueante pelo gerenciamento de filas de alta velocidade, por exemplo).
- **Datalogger** Registrador de dados, produz "time-series" a serem registradas em um sistema como SGBD (banco de dados).
- Driver Componente de software que implementa um protocolo de comunicação, devendo estar compatível com a API de desenvolvimento definida para o DA Server.

Ε

- Eventos são ocorrências na planta supervisonada que não necessariamente representam um valor numérico, ocorrendo de maneira assíncrona e "empurrada", ou seja, partindo da dos dispositivos em direção ao software que os supervisiona. Exemplos de eventos seriam "início de operação do estágio 1 do motor", "operação em regime permanente atingida", "relé de proteção XYZ-14 acionado". Eventos diferem-se de alarmes no sentido de não possuirem uma criticidade associada, nem de exigir uma ação do operador.
- **EPA** Enhanced Performance Architecture.

F G H

- HDA History Data Access, conjunto de funcionalidades relacionadas à análise de dados registrados. Usualmente refere-se a uma API HDA como sendo um conjunto de métodos padronizados para se extrair séries temporais, informandose o nome da tag desejada, o período de início e o período de fim; também pode oferecer funcionalidades como extração de máximas/mínimas, médias e assim por diante.
- **Historian** Software capaz de trabalhar com grandes volumes de dados, atuando como um datalogger avançado com características como compressão de dados, armazenamento por deadband etc.

• **HMI Builder** - Interface de desenvolvimento onde um integrador de sistemas pode desenvolver graficamente (no estilo "wysiwyg") as telas da HMI, adicionando componentes como indicadores, gráficos (plot) e botões de comando.

Ι

- IDE Integrated Development Environment, é um ambiente de desenvolvimento (similar ao Visual Studio ou Eclipse, por exemplo) onde um integrador de sistemas encontra tanto funcionalidades de HMI Builder como funcionalidades de Configurador, permitindo desenvolver todos os aspectos de uma aplicação SCADA completa.
- IEC International Electrotechnical Commission.
- I/O Input/Output ou Entrada e Saída. Utilizamos esse termo para identificar operações de comunicação em baixo nível, ou no nível dos protocolos (como Modbus, Profibus, Fieldbus etc.) No nível de I/O são importantes as definições peculiares de cada protocolo, como representação física do dado (inteiro, double, byte[]), taxas de comunicação (baud rate), identificador de porta ou endereçamento de rede, ao contrário da camada DA que "abstrai" essas informações.

J K L M

- Módulo Um módulo do ScadaBR será o nome dado a um componente de software que tenha uma função "macro" bem delimitada, como por exemplo: Módulo de Históricos, Módulo de Alarmes, Módulo de Batch & Scripting e assim por diante, comunicando-se com o restante do sistema através do Middleware ScadaBR.
- **Módulo de Processamento** um componente de software capaz de processar grandes volumes de informação e gerar novos resultados a partir deles, por exemplo: calcular a média entre duas temperaturas gerando uma "terceira" temperatura, ou identificar padrões de comportamento (formas de onda específicas, por exemplo) e executar um processo externo.

- **Middleware** conjunto de especificações ou softwares dedicados especialmente à função de Interoperabilidade, ou seja, comunicação entre diversos componentes de software.
- **Middleware ScadaBR** conjunto de API's para comunicação nos contextos de DA, HDA e A&E, permitindo que diferentes módulos do ScadaBR comuniquem-se entre si e/ou com softwares externos. O Middleware ScadaBR deve "abstrair" questões de baixo nível (I/O), sistema operacional ou linguagem de programação, sendo o mais "neutro" possível.

N O

- **OPC** OLE for Process Control, middleware padrão da automação industrial largamente adotado em indústrias do mundo inteiro; possui API 's DA, HDA, A&E e outras, implementadas sobre tecnologia DCOM da microsoft. Possui um modelo de dados que descreve componentes e operações comuns no ambiente de
- **OSI** Open System Interconnection.

manufatura.

Ρ

- **Protocolo** Padronização de comunicação especificando camada física, transporte, modelo de dados e formato dos pacotes na comunicação entre equipamentos e o PC onde roda o software SCADA.
- **Poll** Formato de comunicação do tipo "request-response" onde cada novo dado é solicitado pelo consumidor.
- **Push** Formato de dados empurrado onde o produtor envia os dados sequencialmente (de maneira síncrona ou assíncrona) para o consumidor.

Q R

• **Real-Time** - Requisitos de tempo-real, referem-se ao fato de que pode haver exigência que determinada operação de comunicação ou processamento ocorra em uma fatia de tempo determinística, não tolerando atrasos.

S

- **SCADA** Supervisory Control and Data Acquisition.
- **Scripting** -Capacidade de inserir lógicas computacionais no SCADA, atuando em determinadas tags com base no valor de outras. Por exemplo, "se nível do tanque acima de 3 metros, abrir válvula em 70%". Usualmente os scripts são definidos

em linguagens interpretadas de alto nível (como basic, javascript), ou linguagens gráficas padronizadas para o ambiente industrial (como ladder, grafcet)

Т

- **Tag** variável do sistema supervisionado. Uma tag tem um nome (identificador exclusivo em todo o sistema) pelo qual é referenciado a partir de todos os módulos, "escondendo" detalhes de baixo nível referentes ao protocolo ou dispositivo. Exemplos de Tags seriam "temperatura1", "tensao_alternador", "nivel_tanque_A" e assim por diante. Tags podem ser "lidas" (obter último valor com timestamp) ou "escritas" (enviar comando ou set-point para equipamentos). Tags podem implementar diversos tipos de dados como numérico inteiro, pontoflutuante, string, booleano on/off, binário 8-bits, 16-bits e assim por diante.
- Tempo-Real ver Real-Time

U V W X Y Z