

Programa o de Sistemas Supervis rios

Aula 04 - Scripts, Importa o e Exporta o de Dados no ScadaBR



Apresentação



Olá!
Você está preparado para aprender
sobre Scripts, Importação e Expor-
tação de Dados no ScadaBR?
Então aproveita essa aula!

Hoje, você aprenderá como inserir códigos de programação no *ScadaBR*, conhecerá os mecanismos usados para importar e exportar as aplicações desenvolvidas no software e entenderá como o *ScadaBR* funciona em uma utilização distribuída.



Objetivos

Aprender a inserir códigos no *ScadaBR*;

Conhecer os mecanismos de importação e exportação de dados;

Entender o funcionamento da utilização distribuída.

Scripts

O principal objetivo dos sistemas *SCADA* não é possibilitar a inserção de códigos para a realização de determinadas ações. No entanto, muitas vezes você precisará fazer o cálculo de uma variável, tratar de um valor, etc. Desse modo, você irá recorrer ao *ScadaBR*. É importante salientar que essa programação é bem simples, então, caso você esteja pensando em usá-la como IDE, aviso, desde já, que isso não é oportuno.

O *ScadaBR* usa *javascript* como linguagem de programação em seu ambiente, a qual é usada apenas para algumas funções bem simples. Sendo assim, não é necessário um estudo aprofundado a respeito dela. Conhecendo as principais funções do *javascript* e dominando uma outra linguagem de programação (C, C++, entre outras), você não terá nenhum problema para implementar os seus códigos.



Saiba Mais

Se você quiser conhecer mais sobre a linguagem de programação *javascript*, poderá ver vários tutoriais na internet. Indico, especialmente, o livro [“Thinking in JavaScript”](#), de *Aravind Shenoy*, disponibilizado gratuitamente.

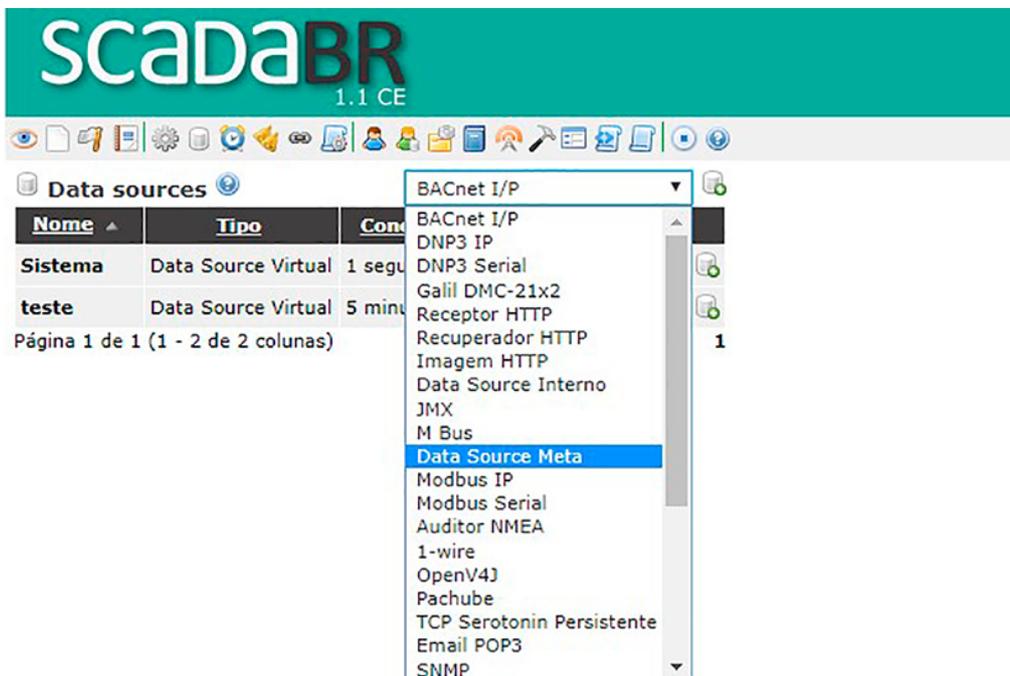
Para entender melhor como inserir scripts no *ScadaBR*, você estudará, primeiramente, o *data source* meta e, assim, aprenderá como inserir os códigos.

Data Source Meta

O *data source* meta, como o próprio nome já diz, é um *data source*. Ele é chamado de “meta” devido à sua capacidade de combinar pontos existentes em novos. Ao invés de obter sua informação de uma fonte externa, esse *data source* utiliza valores de outros pontos que permitem ao usuário manipulação de maneiras arbitrárias (SCADABR, 2015).

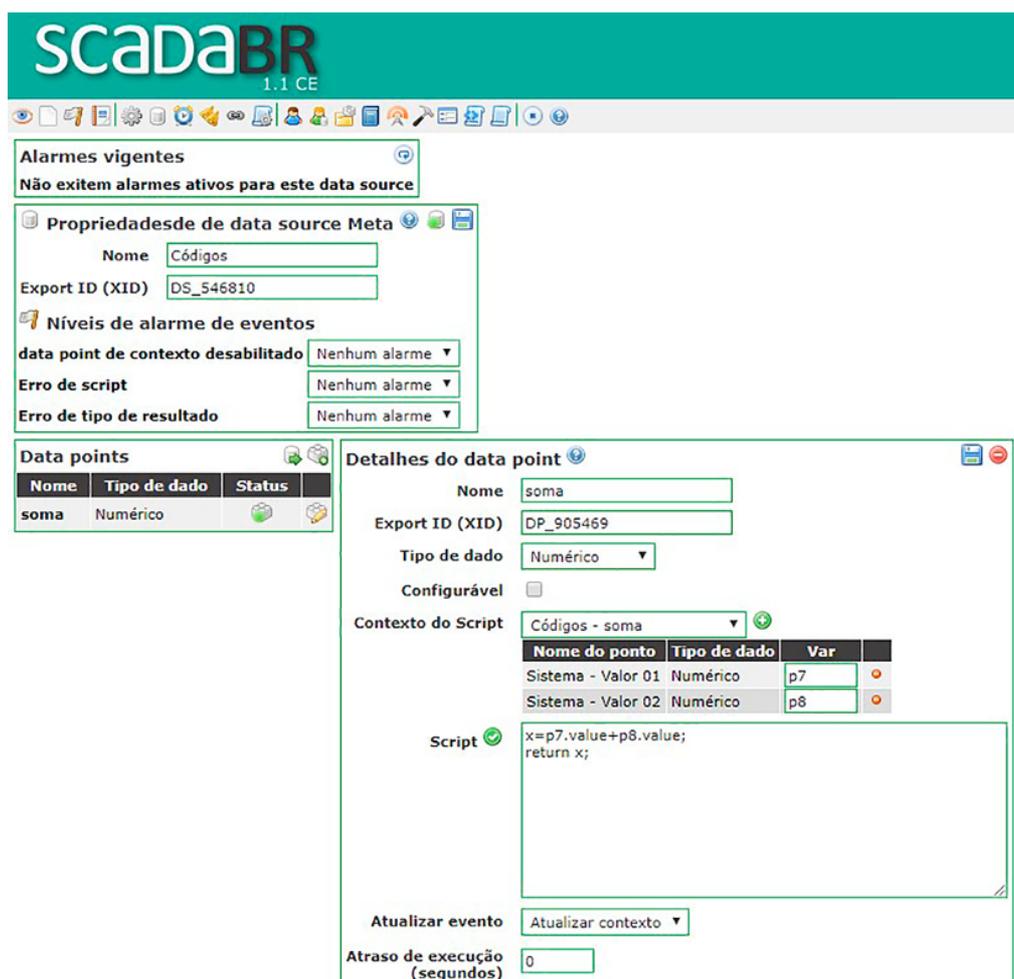
Para inserir os códigos no *data source* meta é necessário colocá-los na aplicação. Então, basta seguir o passo demonstrado na Figura 01.

Figura 01 - Inserindo o *data source* meta



A configuração do *data source* meta é fácil. É necessário, apenas, inserir um nome, como mostra a Figura 02. Para colocar os códigos, é preciso inserir um *data point* e configurá-lo. Depois você deve escolher o tipo de dado de saída, bem como as variáveis que estarão no contexto do código. Perceba que o sistema automaticamente nomeia essas variáveis (p7 e p8), porém, caso você ache conveniente, pode inserir outros nomes, mas é importante salientar que essas variáveis são locais e, como se tratam de códigos bem simples, isso nem sempre é necessário.

Figura 02 - Configurando o *data source*



Ainda de acordo com o exemplo da Figura 02, na área destinada ao script deve ser inserido o código por meio da linguagem *javascript*. Para testar o código, basta clicar no ícone  e o resultado será apresentado. Você ainda pode escolher como será a atualização do seu código, decidindo se será no contexto (quando houver modificação nas variáveis) ou determinando dia, hora, minuto, semana, mês, etc. É recomendável a utilização da primeira opção, tendo em vista que só é relevante o envio de dados quando houver alguma modificação, pois, caso não haja, o dado anterior já deverá estar sendo mostrado na tela.

Ao final da configuração você poderá ver o *data point* criado na *watch list*, como mostra a Figura 03, e conseguirá realizar qualquer tipo de configuração e renderização nesse dado, assim como já viu nas aulas anteriores.

Figura 03 - Dado calculado

Watch list			
(sem nome)			
Códigos - soma	7.0	12:49:38	✓
Sistema - Valor 01	2.0	12:49:38	✓
Sistema - Valor 02	5.0	12:49:38	✓

Agora,
assista um vídeo que ensina como
criar um *data source* meta com
alguns exemplos de códigos.



Vídeo 01 - Criar um *data source* meta

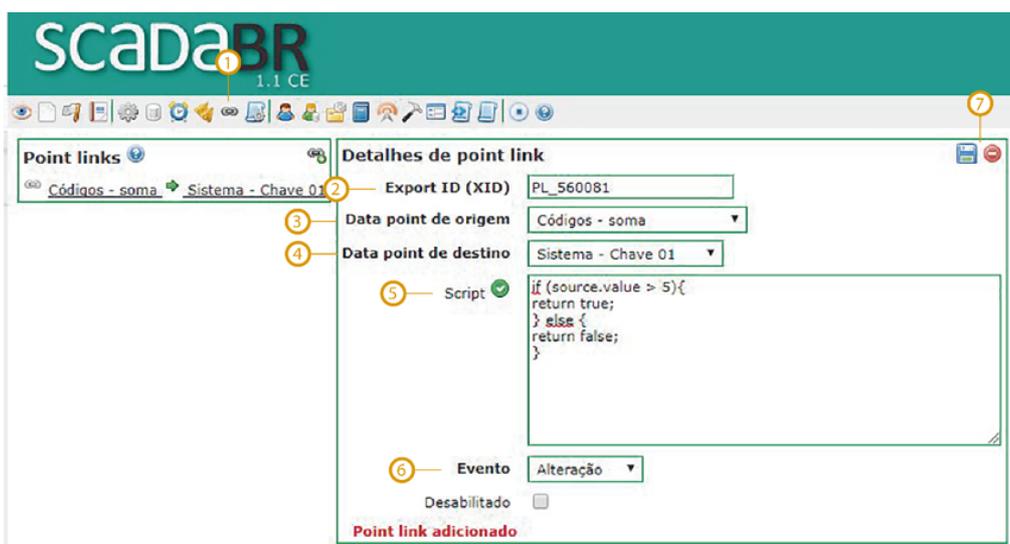
Point links

Você pode estar se perguntando: “Como posso fazer uma modificação em um valor externo?”. Perceba que até agora só foram estudados os valores e a criação de *data points* baseados em alguma lógica com outros *data points*, no entanto, ainda não foi possível, por exemplo, acender uma lâmpada com base em outro *data point*. Para fazer isso, é preciso usar os *point links*. Os *point links* são responsáveis por

fazer a “ligação” entre dois point links através de um script, onde a entrada do script seria o data point que se deseja fazer alguma verificação e a saída do *script* e o *data point* em que se deseja atuar. Assim, as entradas do *point link* são os *data points* a serem interligados e o script que implementará a lógica desejada entre esses data points.

Como você pode ver na Figura 04, para adicionar um novo *point link*, é preciso clicar no ícone , e configurar os seguintes pontos:

Figura 04 - *Point link*.



1 O **Export ID** é o valor utilizado para identificar o *point link* em diferentes instâncias do *ScadaBR*.

2 O **ponto de origem** é o ponto que terá seu valor monitorado.

3 O **ponto de destino** é o ponto que terá seus valores alterados.

4 O **script** é um valor opcional que transformará os valores utilizados conforme a necessidade. Você pode testar o código clicando em .

5 **Evento** determinará se o data point de destino será alterado somente quando o seu valor mudar ou sempre que o *data point* de origem for alterado.

6 Um *point link* pode ser **Desabilitado**, para não ser executado. Isso pode ser útil caso você não queira excluí-lo, mas também não queira habilitá-lo no momento.

7 Para salvar um **point link**, clique no ícone . Para excluir um existente, clique no ícone .

O código inserido na Figura 04 mostra o acionamento da chave para valores maiores do que 5.

Na Figura 05 você pode ver o teste dessas condições na *watch list*.

Figura 05 - Funcionamento do *point link*

Watch list 		(sem nome)     
Códigos - soma	8.0	13:11:06    
 Sistema - Valor 01	5.0	13:11:06    
 Sistema - Valor 02	3.0	13:11:06    
 Sistema - Chave 01	1	13:11:06    

Watch list 		(sem nome)     
Códigos - soma	4.0	13:11:34    
 Sistema - Valor 01	1.0	13:11:34    
 Sistema - Valor 02	3.0	13:11:34    
 Sistema - Chave 01	0	13:11:34    

Assistindo ao Vídeo 02 você aprenderá como criar os *point links*. Veja-o!



Vídeo 02 - Criando os *point links* meta

Importação e Exportação

Agora que você aprendeu a utilizar as principais ferramentas do *ScadaBR*, chegou a um ponto muito importante: como colocar o sistema em outra máquina ou servidor, após a aplicação dele ter sido concluída. Você já percebeu que a implementação de um sistema de automação dura um bom tempo devido à quantidade de variáveis. Desse modo, para que você não fique durante dias no domicílio do cliente, o ideal é montar toda a aplicação em sua própria casa ou escritório e, após a conclusão, inseri-la na máquina do contratante. Para isso, há duas maneiras pelas quais optar a fim de realizar essa ação: *Json* ou *Zip*.

Na Figura 06 veja como fazer a importação e exportação via *json*.

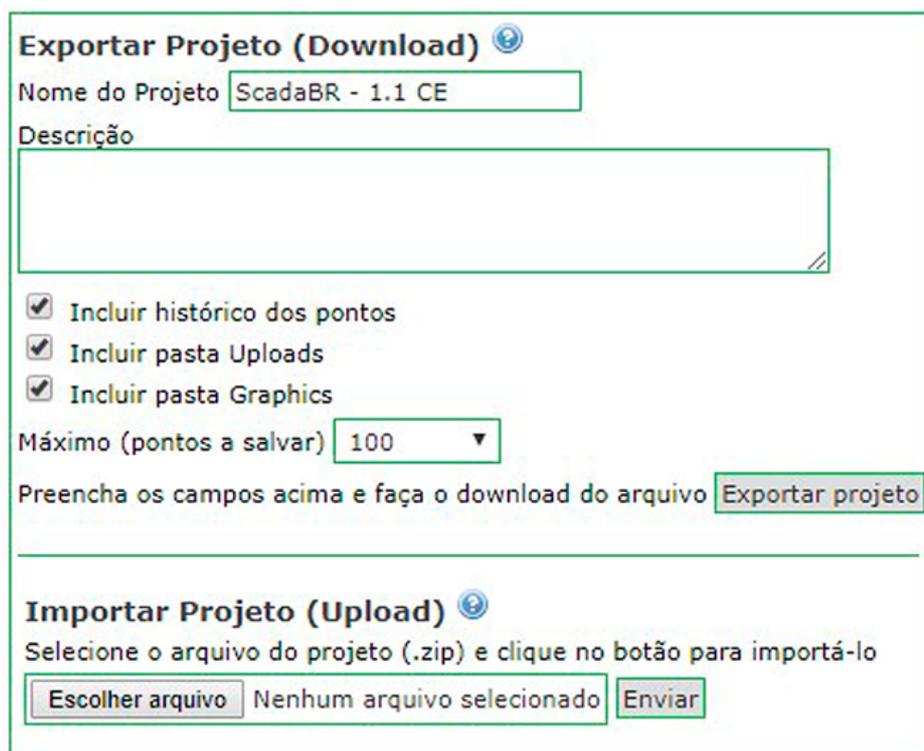
Figura 06 - Importando e exportando via *json*



Fazer a exportação via *json* é bem simples: basta selecionar os campos que deseja exportar e, então, gerar o *json*. Em seguida, os dados gerados serão apresentados no campo **dados**. Para importar, basta colar os dados do *json* no campo de **dados** e clicar em “importar”.

Agora, observe a Figura 07 e veja como fazer a importação e exportação via *Zip*.

Figura 07 - Importando e exportando via *zip*



The screenshot displays two sections of the ScadaBR interface. The top section, titled "Exportar Projeto (Download)", includes a text input for "Nome do Projeto" containing "ScadaBR - 1.1 CE", a large empty text area for "Descrição", three checked checkboxes for "Incluir histórico dos pontos", "Incluir pasta Uploads", and "Incluir pasta Graphics", and a dropdown menu for "Máximo (pontos a salvar)" set to "100". Below these is a button labeled "Exportar projeto". The bottom section, titled "Importar Projeto (Upload)", contains the instruction "Selecione o arquivo do projeto (.zip) e clique no botão para importá-lo" and two buttons: "Escolher arquivo" and "Enviar". The text "Nenhum arquivo selecionado" is positioned between these two buttons.

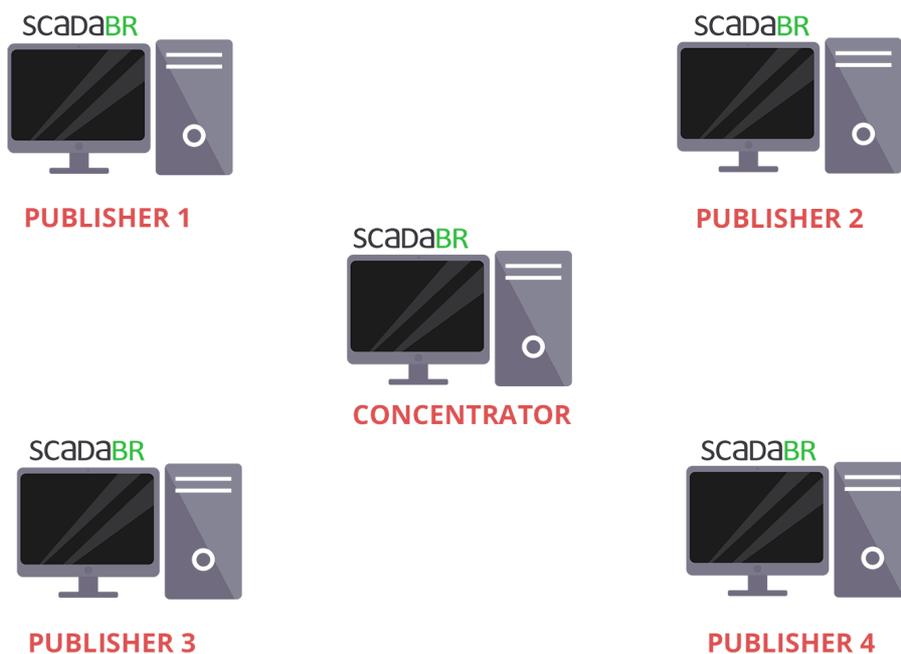
A exportação via *Zip* também é simples: é necessário apenas selecionar os itens e a quantidade de pontos que deseja salvar. Ao clicar em “exportar projeto”, o sistema vai gerar um arquivo do tipo *.zip*. Para a importação, basta escolher o arquivo *.zip* e enviá-lo.

Utilização distribuída do *ScadaBR*

Em plantas grandes de automação, relativas, em geral, a indústrias, é comum haver sistemas bem distintos, cujas áreas são bem diferentes, por exemplo sistema de ar condicionado, tratamento de esgoto, controle das caldeiras, monitoramento de energia, monitoramento de pressão, etc. Perceba que todos esses sistemas, embora sejam muito distintos, possuem grande complexidade. Você já deve ter visto que sistemas complexos envolvem muitos *data points*, *data sources* e telas gráficas. Assim, em alguns casos o correto é optar por sistemas *SCADA* diferentes para cada aplicação, mesmo que seja necessária uma interligação entre eles.

Para fazer a comunicação entre diferentes *ScadaBR's*, utiliza-se o conceito de “*Publisher-Receiver*”, como visto na arquitetura da Figura 08. No menu *publishers*  se encontra o mecanismo de envio dos dados para outras máquinas que possuam *ScadaBR*. Esse envio ocorre através do *datasource* receptor HTTP  , que permite receber os dados enviados por outro *ScadaBR* e publicar valores de *data points* por meio do protocolo HTTP pelo *Data Source* HTTP.

Figura 08 - Arquitetura de comunicação distribuída



O *publisher* pode ser considerado o oposto dos *data sources*, pois, ao invés de coletar dados para o sistema, os *publisher's* os distribuem. Isso pode ser útil quando o *ScadaBR* é utilizado a fim de coletar informações de diferentes sistemas para modificar, filtrar ou gerenciar de alguma maneira. Nesse sentido, o *ScadaBR* não é apenas M2M (machine to machine), mas M2M2M.

Uma outra função para a qual também é útil o uso de editores diz respeito ao envio de dados a outras instâncias do *ScadaBR* — por exemplo, para consolidar dados de várias instâncias remotas. Cada instância do *ScadaBR* pode usar editores a fim de controlar, de diversos modos, outras instâncias (SCADABR, 2015).

Veja na Figura 09 a tela de configuração do *publisher*, bem como os campos de configuração dele:

Figura 09 - Configurações do publisher

The image shows a configuration window for a publisher. It is divided into three main sections:

- Propriedades gerais:** Fields for 'Nome' (maquina 1), 'Export ID (XID)' (PUB_104476), 'Habilitado' (checked), 'Alerta para tamanho de armazenamento' (100), 'Atualizar evento' (Todas atualizações), 'Enviar imagem instantânea padrão' (unchecked), and 'Imagem instantânea do período enviada' (5 minutos).
- Propriedades do remetente HTTP:** Fields for 'Método' (GET), 'URL' (http://localhost:8080/ScadaBR/), 'Cabeçalhos estáticos' (None defined), 'Parâmetros estáticos' (None defined), 'Alerta de aumento de resultado' (checked), and 'Formato de data' (Básico).
- Teste de envio HTTP:** Buttons for 'Enviar parâmetros estáticos' and 'Cancelar'.

1 O **Método** determina qual método HTTP será usado para entregar os dados. Os dados enviados em ambos os casos são idênticos, diferenciam-se apenas quanto à estrutura de mensagem.

2 O **URL** define o destino para o qual enviar os dados. Esquemas válidos incluem "http" e "https". O servidor pode ser um nome de domínio ou um endereço IP. Caminhos e strings de consulta também são válidos.

3 Uma lista de **Parâmetros estáticos** também pode ser adicionada. Esses parâmetros são incluídos em toda requisição enviada ao URL. Dessa maneira, este é um local apropriado para incluir credenciais de autorização ou outros artefatos de identificação.

A fim de configurar quais pontos deverão ser publicados, estes devem ser configurados na lista, como você pode ver na Figura 10. A publicação de **pontos** pode ser habilitada () ou desabilitada (). Desabilitar um ponto permite que ele não seja usado, sem ter de explicitamente deletá-lo. O **Nome do parâmetro** é a chave que será usada para o ponto nos parâmetros da requisição. É muito importante que esse nome seja registrado, pois deverá ser o mesmo para sua requisição em outras máquinas. Selecione **Incluir timestamp** para incluir um parâmetro na requisição que contenha o tempo de atualização para o valor. A chave para o tempo será "__time", e a data será codificada mediante o formato padrão "yyyMMddHHmmss".

Figura 10 - Configuração dos pontos do *Publisher*

Pontos					Sistema - Chave 01
Nome	Status	Tipo	Nome do parâmetro	Incluir timestamp	
Códigos - soma		Numérico	Códigos - soma	<input checked="" type="checkbox"/>	

Para os dados serem recebidos por outra máquina, é necessário configurar o receptor http (*data source*). Para tanto, é preciso preencher os seguintes campos exibidos na Figura 11:

A configuração do *data point* também pode ser vista na Figura 11. Para configurá-lo, é necessário ajustar os seguintes pontos:

Figura 11 - Configuração do receptor *http*

Propriedades do receptor HTTP

1 Nome: Dados maquina1

Export ID (XID): DS_035400

2 Lista branca de IP remotos: *

3 Lista branca de ID de dispositivo: *

Níveis de alarme de eventos: Não existem eventos para este tipo de data source

Auditor de receptor HTTP

Escutar por dados HTTP Cancelar

Data points

Nome	Tipo de dado	Status	Parâmetro
Soma maquina 1	Numérico		Códigos - soma

Detalhes do data point

Detalhes do data point salvos:

Nome: Soma maquina 1

Export ID (XID): DP_460218

4 Nome de parâmetro HTTP: Códigos - soma

5 Tipo de dado: Numérico

6 Valor 0 binário:

1

Nome: que pode conter qualquer descrição.

2

A **Lista branca de IPs remotos** é uma funcionalidade de segurança que instrui o data source a ignorar requisições por meio de um endereço IP de origem, o qual não corresponde a qualquer máscara de endereço IP fornecida. Essa funcionalidade é de grande utilidade, caso o usuário queira receber requisições apenas de quem ele deseje.

3

A **Lista branca de ID do dispositivo** proporciona segurança adicional, bem como roteamento de dados, através da permissão dos dispositivos para os quais esse data source escuta. A correspondência ao ID do dispositivo independe de maiúsculas e minúsculas em seu nome. Além disso, um '*' pode ser usado no término de uma máscara de ID de dispositivos. Por exemplo, a máscara 'site10*' irá corresponder aos IDs de dispositivos 'site10', 'site10temp', 'site10HUM', etc. O ID do dispositivo é especificado na requisição HTTP com o nome do parâmetro **__dispositivo** (com dois sublinhados).

4

O **Nome do parâmetro HTTP** é a chave usada para entregar os dados ao data source auditor HTTP. Esse campo tem de ser preenchido do mesmo modo como foi utilizado para publicar o valor.

5

O **Tipo de dado** é qualquer tipo de dado válido.

6

Se o tipo do dado for Binário, um **Valor Binário 0** deve ser fornecido. Se o valor do ponto na requisição compatibilizar com esse valor, o valor atribuído ao ponto será 0; caso contrário, será 1.

A comunicação pode ser feita de forma bem simples, mas perceba que deve haver configuração em duas máquinas distintas. Primeiramente, devem ser configurados na máquina 1, usando o ***publisher***, os dados que deseja enviar e o local para onde serão enviados. Posteriormente, deve ser criado um ***datasource receptor http*** para recebê-los, de modo que cada dado esteja em seu respectivo *data point* e com o nome **exatamente igual** ao que foi utilizado para o envio no *publisher*.



Resumo

Nesta aula você aprendeu como inserir códigos de programação no *ScadaBR*, conheceu os mecanismos para importação e exportação das aplicações desenvolvidas no software e entendeu o funcionamento e as aplicações do *ScadaBr* em uma utilização distribuída.



Leitura complementar

Thinking in JavaScript de *Aravind Shenoy* disponível em: <https://www.amazon.com.br/Thinking-JavaScript-Aravind-Shenoy-ebook/dp/B00JUI6LUQ>.



Autoavaliação

E aí, aprendeu o conteúdo estudado hoje? Teste agora seus conhecimentos fazendo em sua aplicação todos os passos apresentados na aula. Procure, também, experimentar o código em várias condições.



Referências

SCADABR automação para todos, versão 1.0. 2015. Disponível em: <http://www.scadabr.com.br/?q=downloads>> Acesso em: 30.jan.2018.