

# **Comunicação entre VTSCADA e CodeSys**

Henrique Silva Coutinho

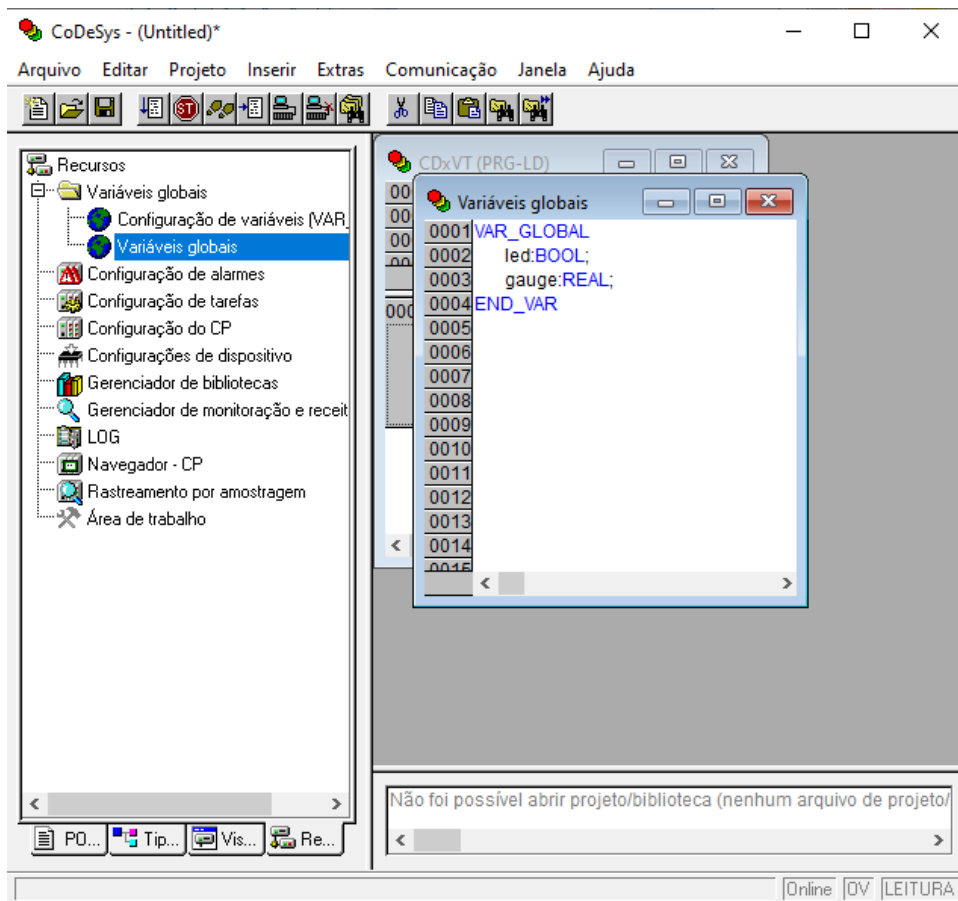
# Comunicação entre o VTSCADA E O CODESYS

Neste tutorial aprenderemos:

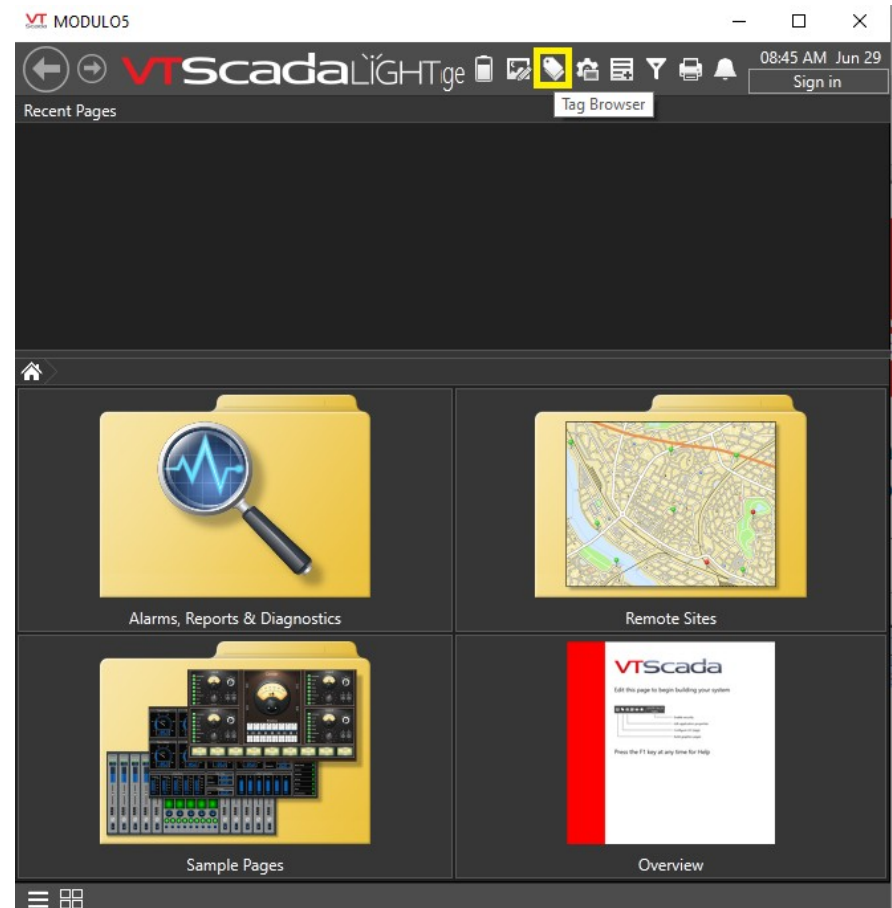
- Como realizar a comunicação entre o VTSCADA e o CODESYS através de uma troca bidirecional;
- Fazer um programa em LADDER no CODESYS comunicar com o supervisor no VTSCADA;
- Implementar uma função de primeira ordem no CODESYS e comunicar com o VTSCADA.

# Criando aplicação no CodeSys e no VTSCADA

Com o projeto criado no CodeSys adicionamos duas variáveis globais, para o led e para o gauge.

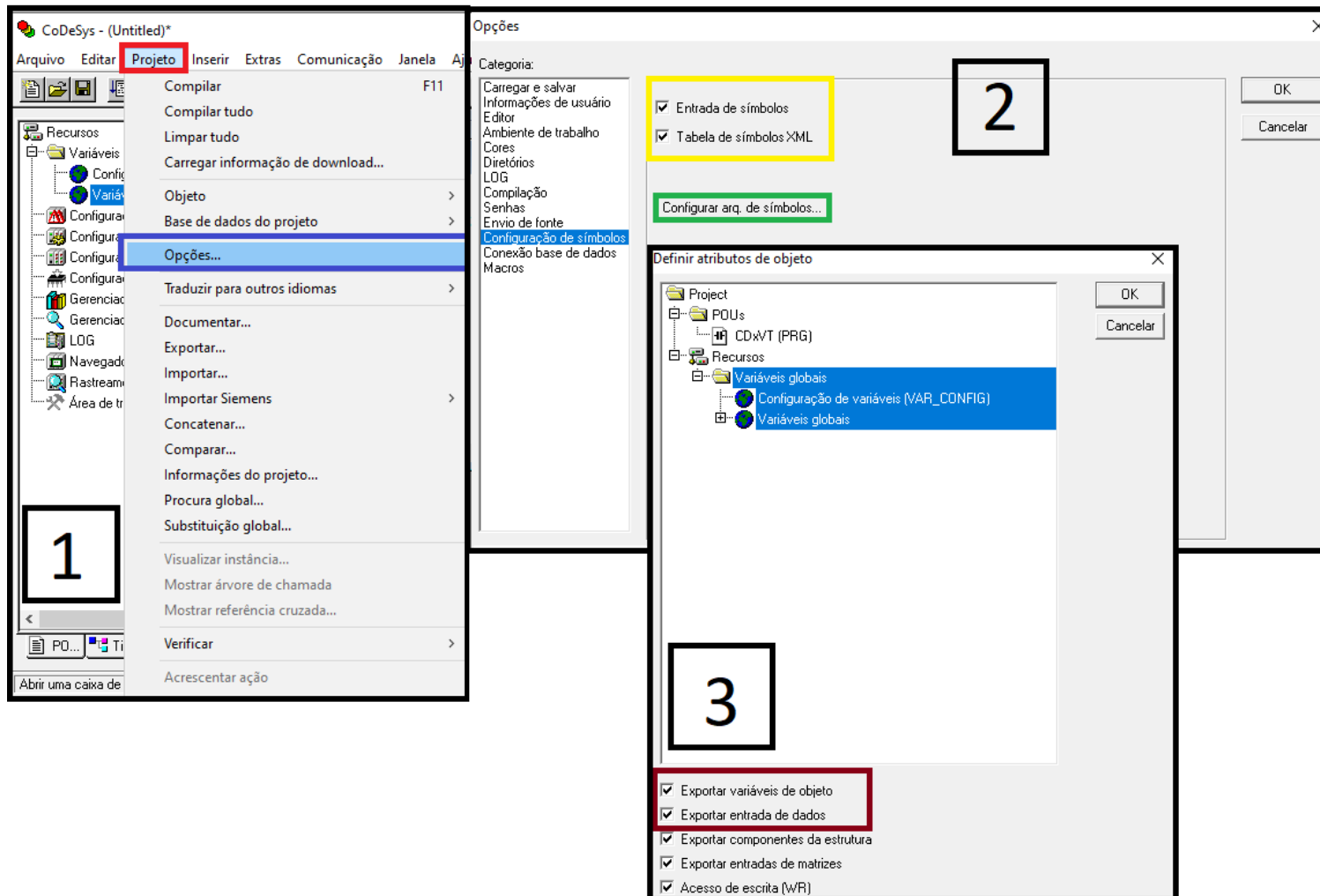


No VTSCADA, com a aplicação criada vamos adicionar uma tag clicando no **Tag Browser**:



# Criando aplicação no CodeSys e no VTSCADA

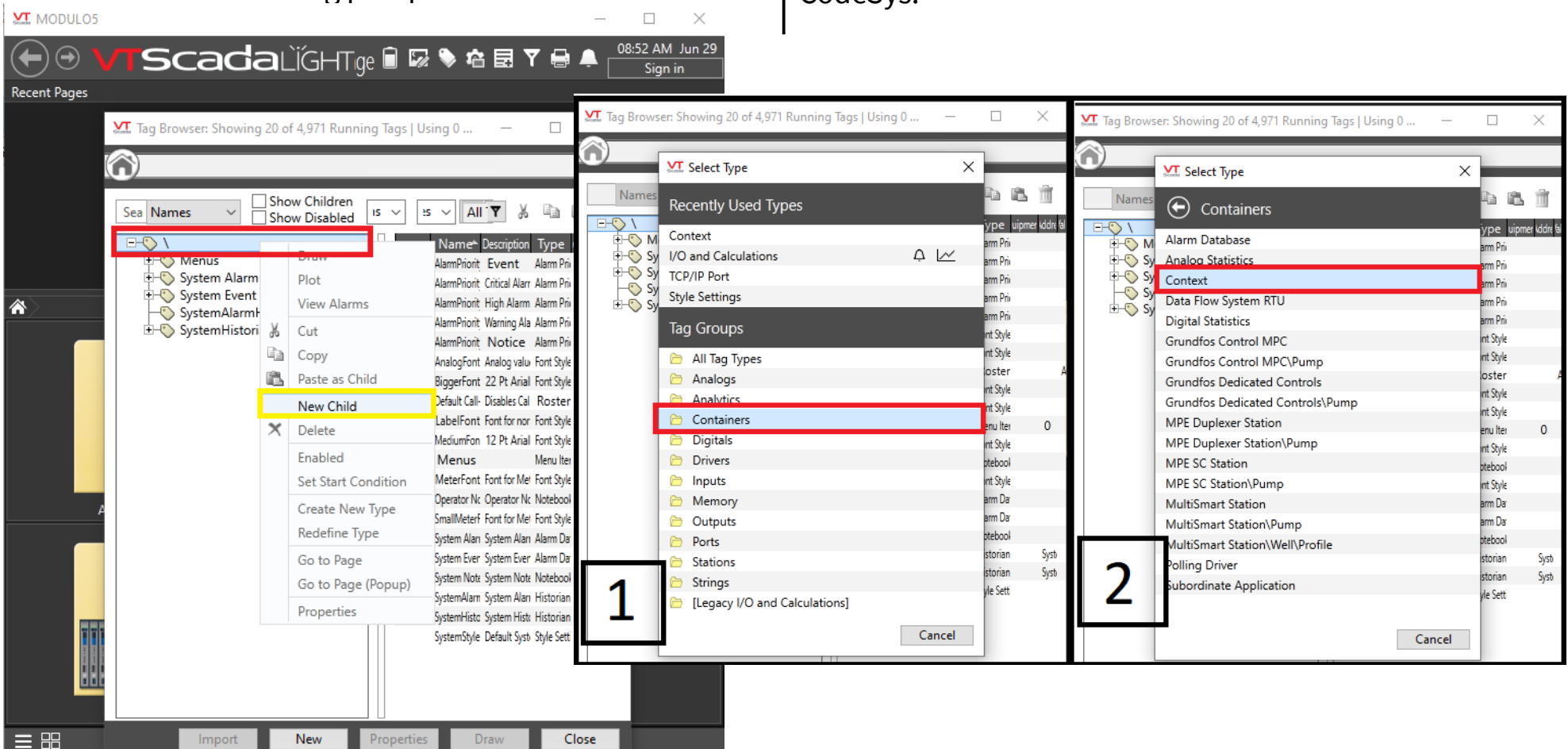
Devemos então estabelecer quais programações e variáveis vamos exportar para o VTSCADA:



# Criando aplicação no CodeSys e no VTSCADA

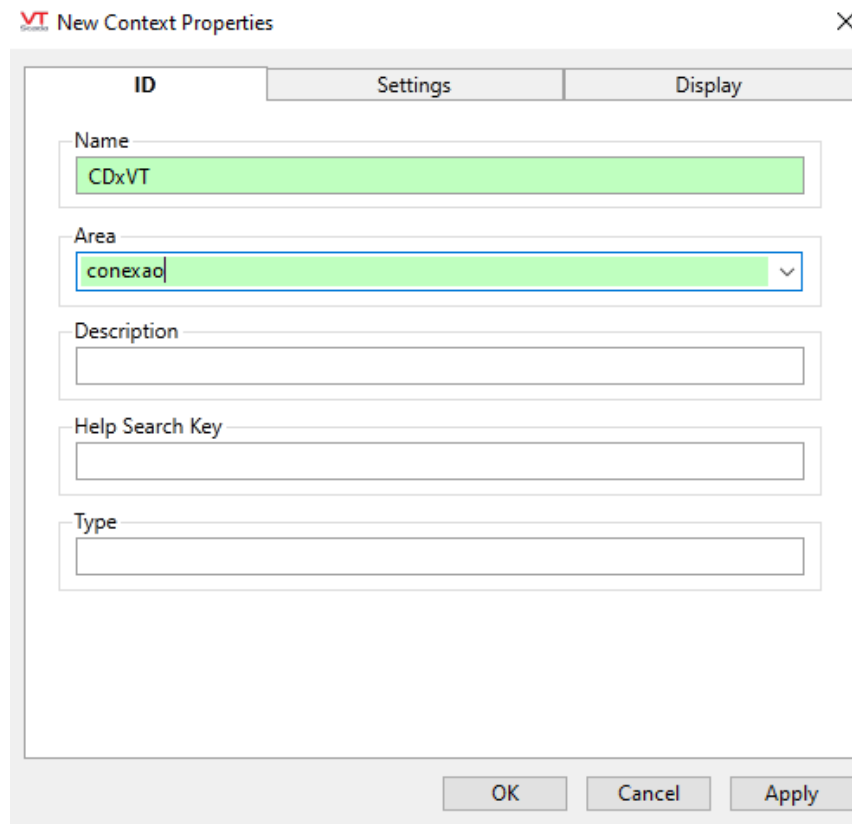
No Tag Browser, clicamos com o botão direito do mouse na área em **vermelho** e clicamos em **New Child** para criarmos um filho da tag principal:

Clicamos então nos botões em **vermelho** (Containers e em seguida Context) para criarmos a conexão com o CodeSys:



# Criando aplicação no CodeSys e no VTSCADA

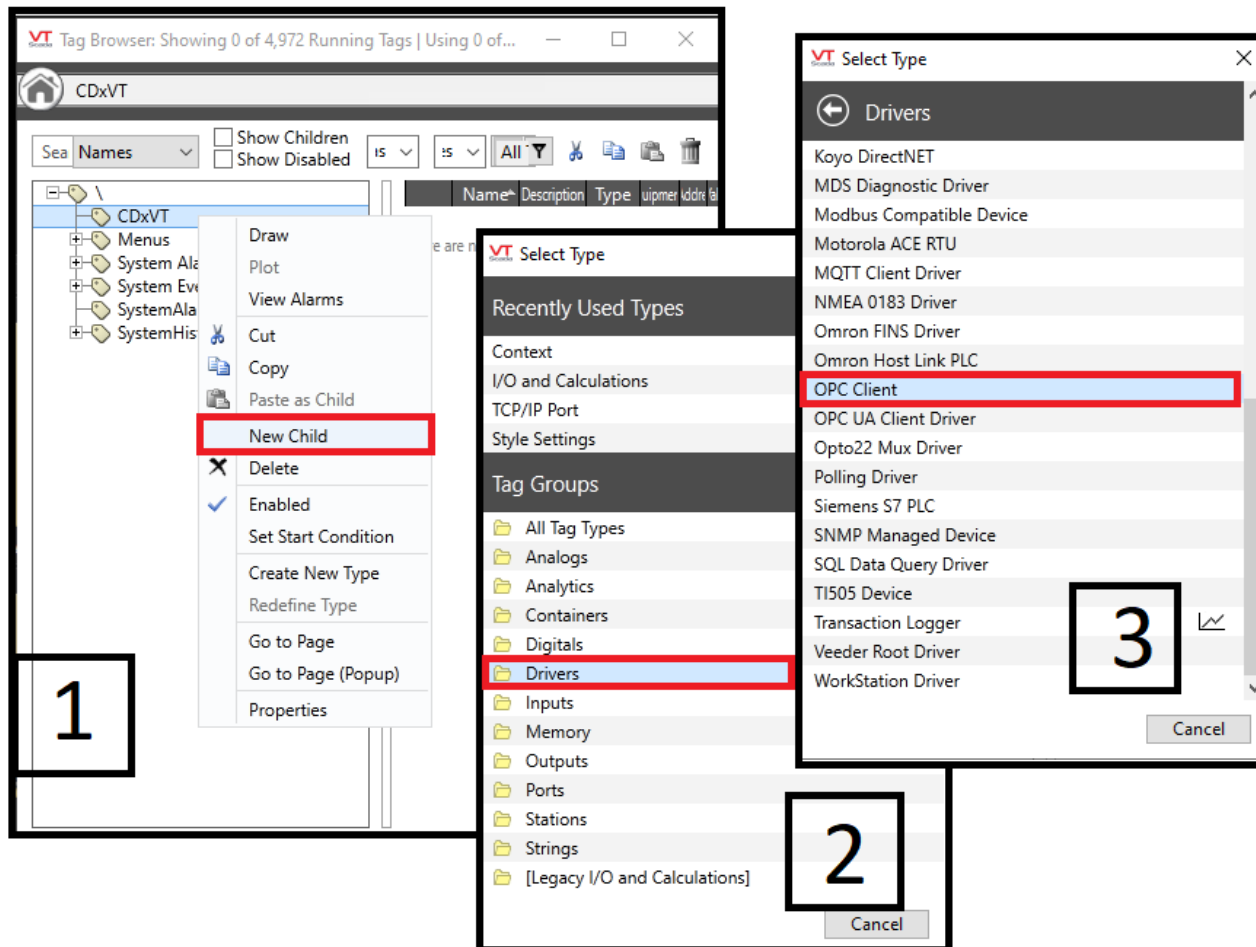
Damos um nome e escrevemos a área:



The screenshot shows the 'New Context Properties' dialog box with the 'ID' tab selected. The 'Name' field is filled with 'CDxVT' and the 'Area' field is filled with 'conexao'. The 'Description', 'Help Search Key', and 'Type' fields are empty. The 'Settings' and 'Display' tabs are also visible but not active. The 'OK', 'Cancel', and 'Apply' buttons are at the bottom right.

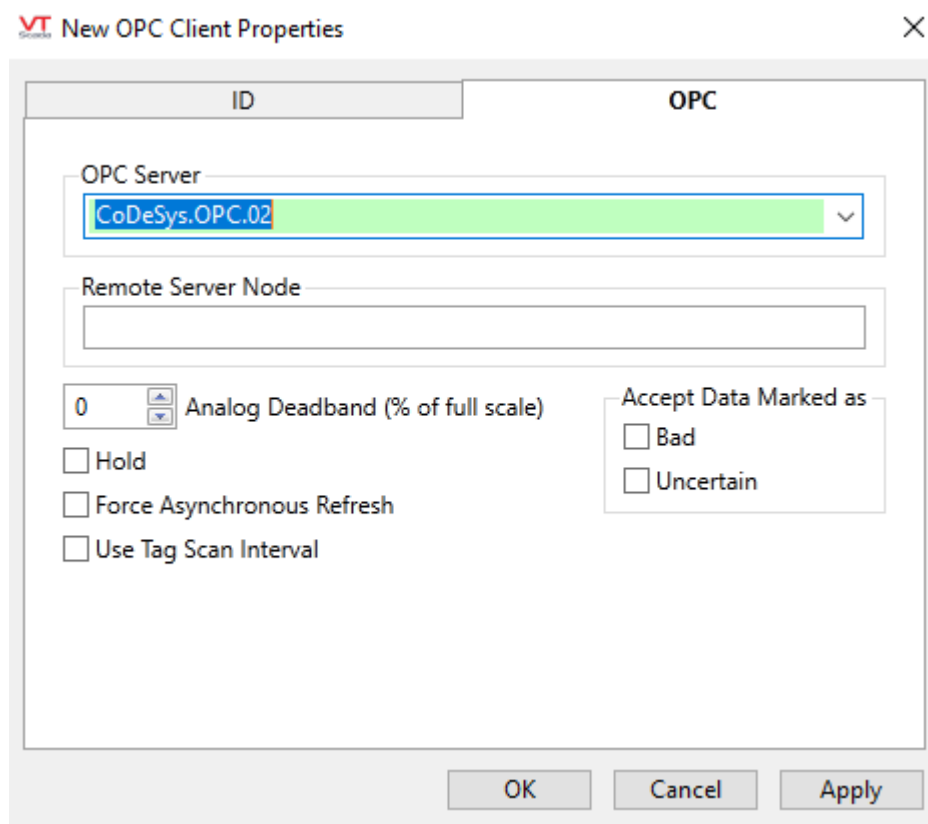
ID	Settings	Display
Name		
CDxVT		
Area		
conexao		
Description		
Help Search Key		
Type		

# Criando aplicação no CodeSys e no VTSCADA



# Criando aplicação no CodeSys e no VTSCADA

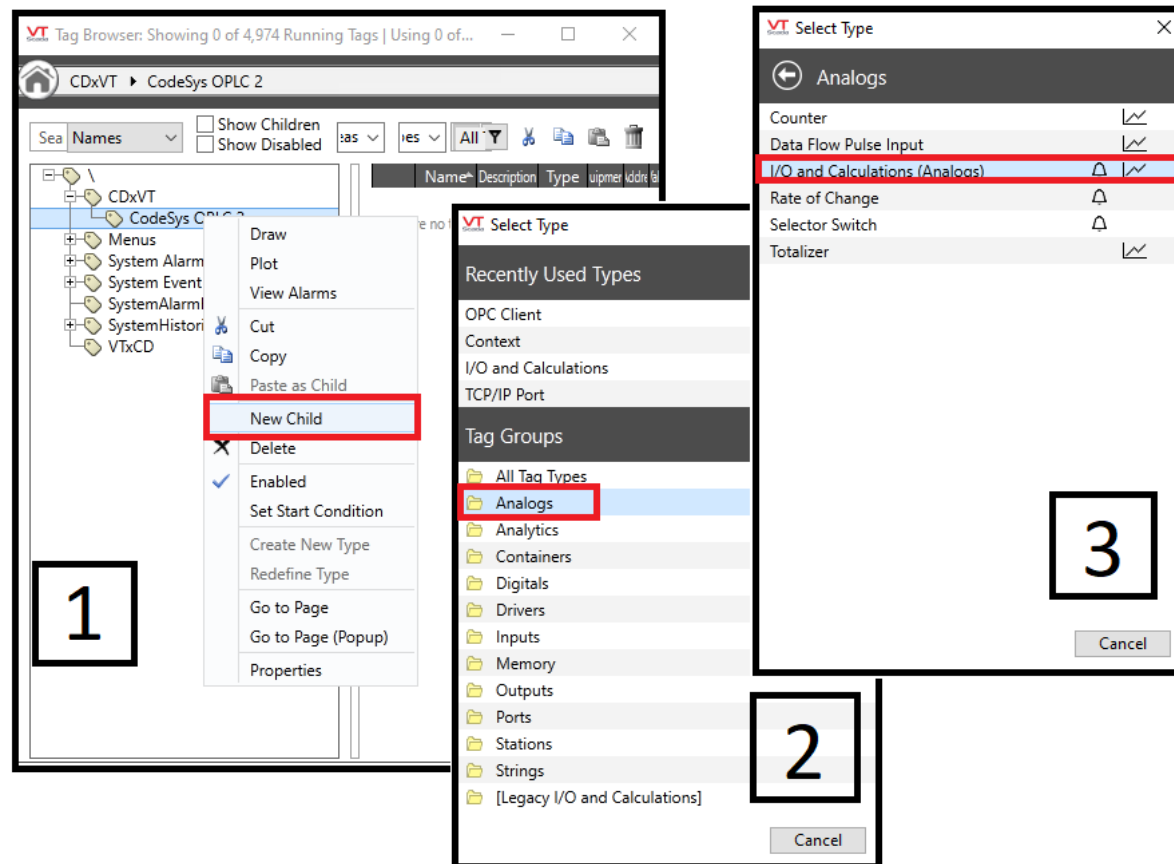
Selecione o OPC Server do CodeSys como abaixo:





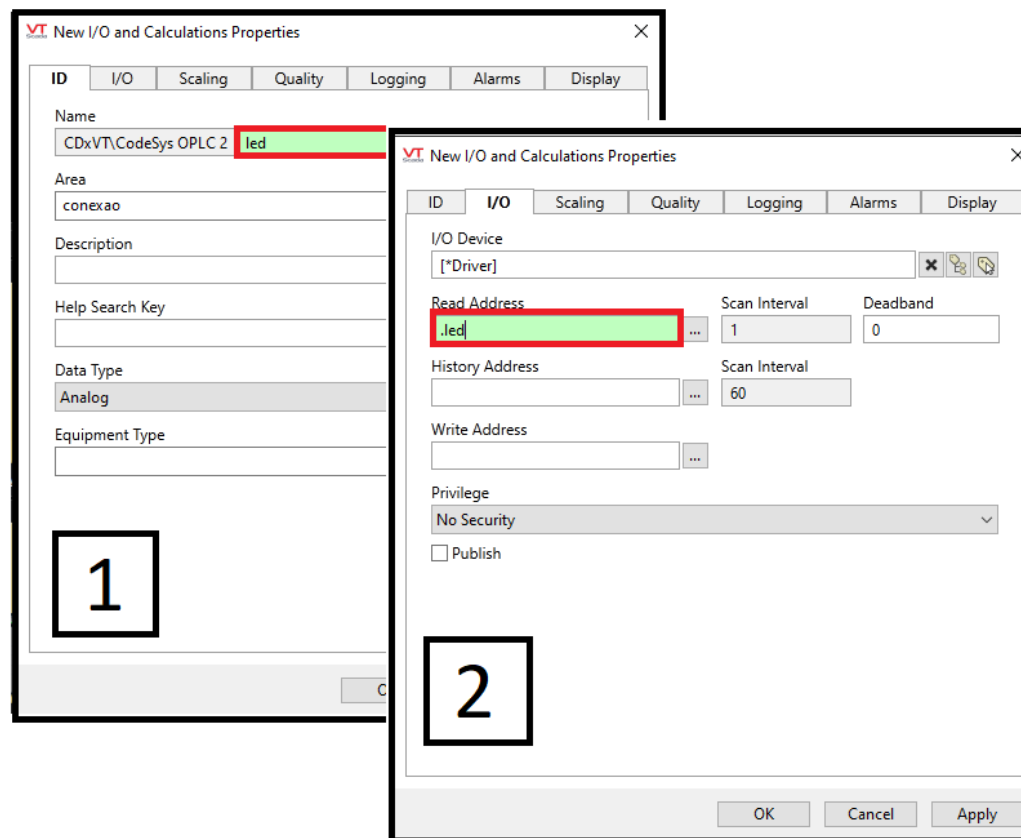
# Criando aplicação no CodeSys e no VTSCADA

Agora devemos criar as tags para nossas variáveis: Para ambas variáveis seguimos os passos a seguir:



# Criando aplicação no CodeSys e no VTSCADA

Para o led:



# Criando aplicação no CodeSys e no VTSCADA

Para o gauge

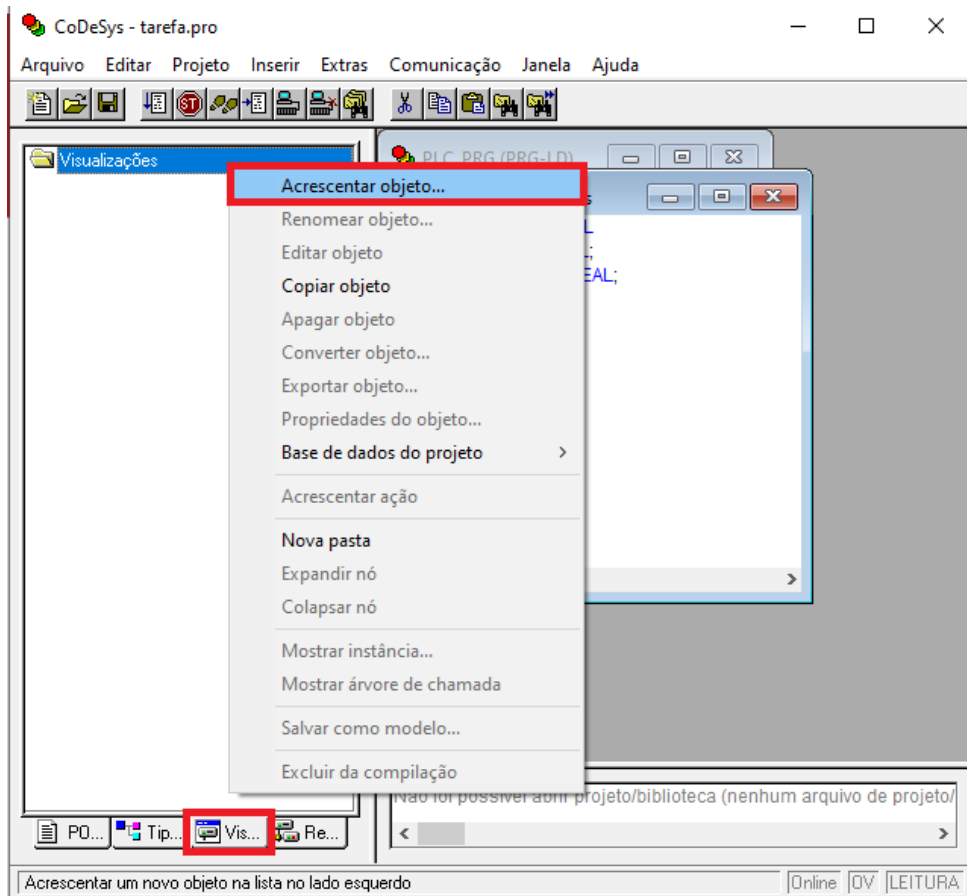
The image shows two screenshots of the 'New I/O and Calculations Properties' dialog box in VTSCADA, illustrating the configuration for a gauge.

**Screenshot 1 (Left):** The 'ID' tab is selected. The 'Name' field contains 'CDxVT\CodeSys OPLC 2 gauge'. The 'Area' field contains 'conexao'. The 'Data Type' is set to 'Analog'. A red box with the number '1' is overlaid on the bottom left of the dialog.

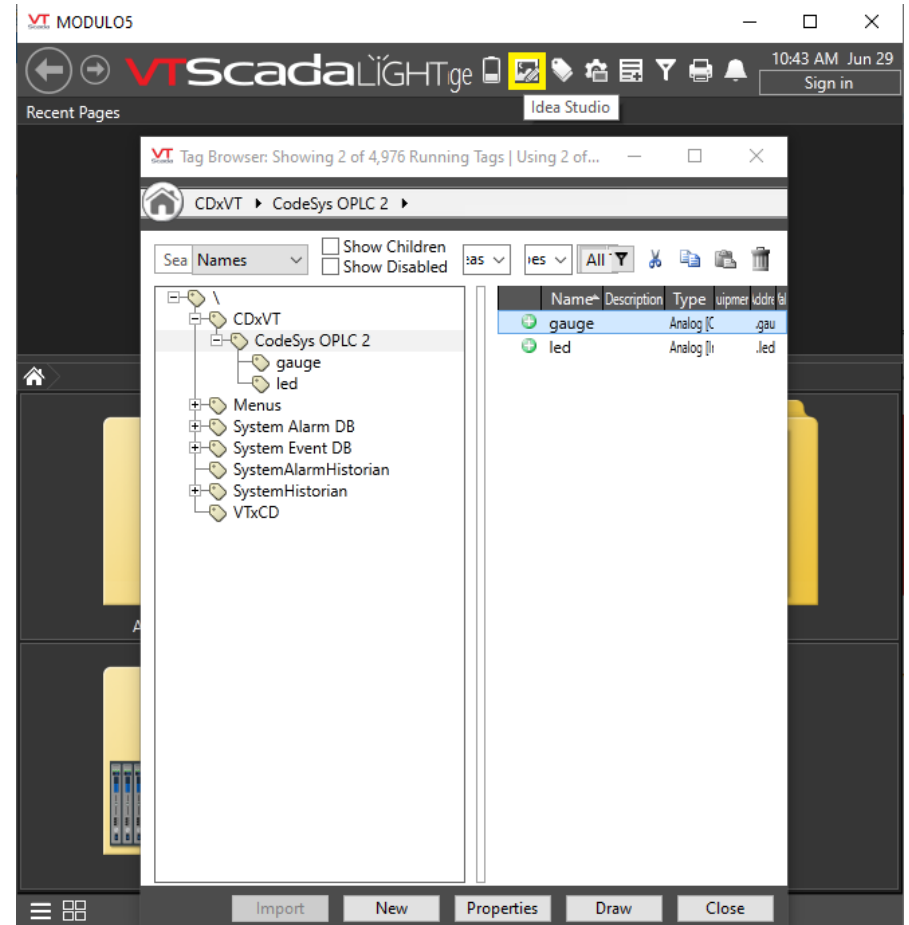
**Screenshot 2 (Right):** The 'I/O' tab is selected. The 'I/O Device' is set to '[\*Driver]'. The 'Read Address' is set to '1'. The 'Scan Interval' is set to '1'. The 'Deadband' is set to '0'. The 'History Address' is set to '60'. The 'Write Address' is set to '.gauge'. The 'Privilege' is set to 'No Security'. A red box with the number '2' is overlaid on the bottom left of the dialog.

# Configurando a IHM

No CodeSys criamos uma nova **visualização**:

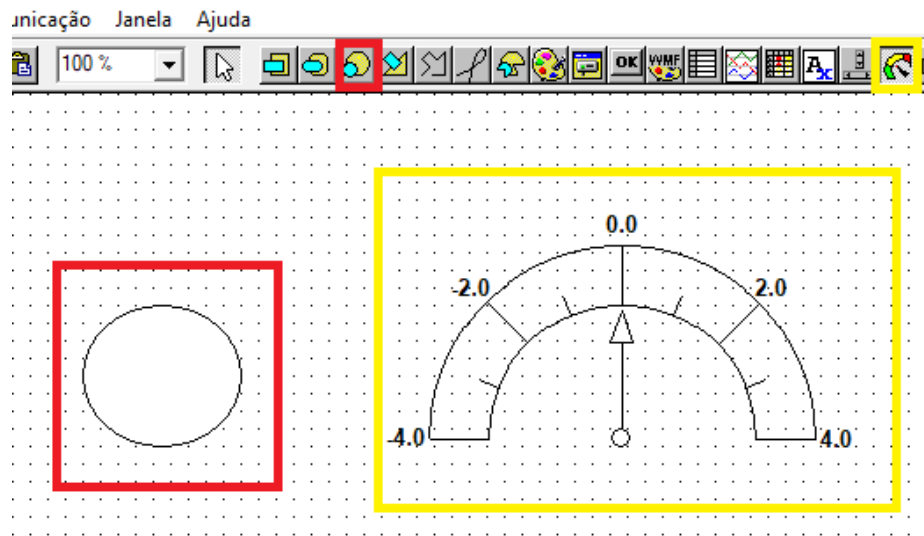


No VTSCADA vamos ao **Idle Studio**:



# Configurando a IHM

Na visualização colocamos algo para representar um botão(nesse exemplo um circulo) e um gauge :

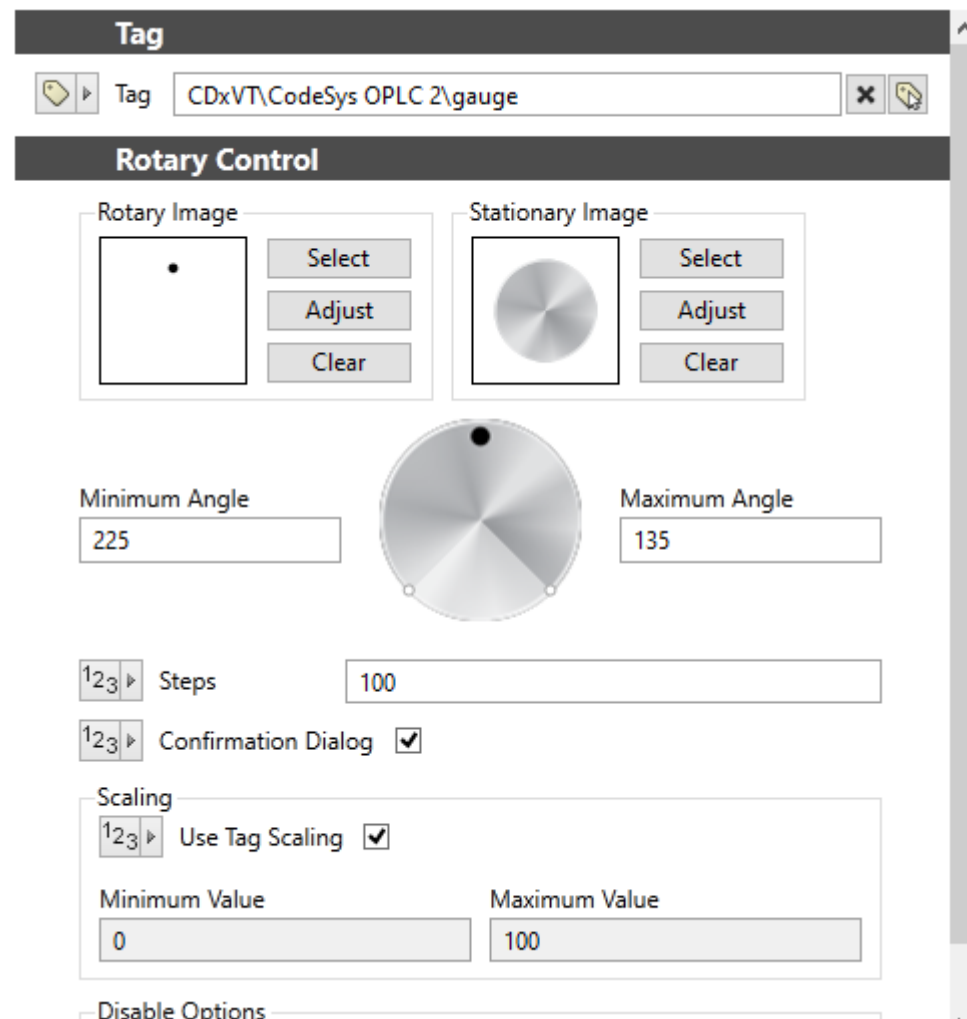
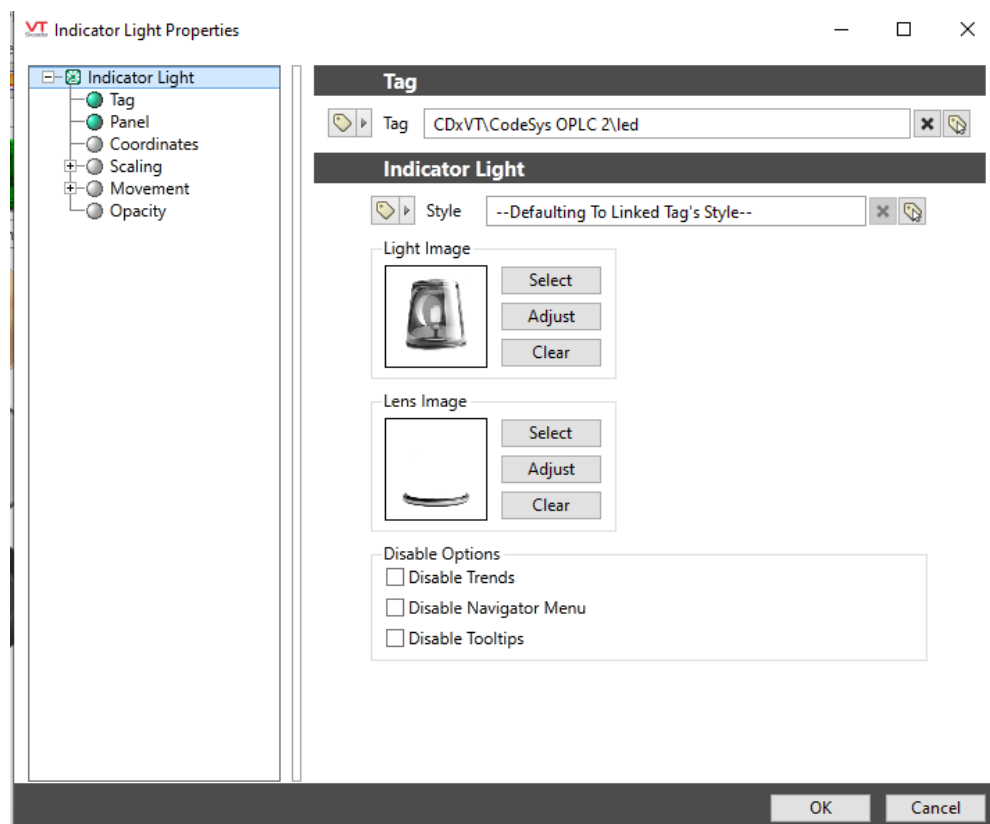


No Idle Studio adicionamos um led e um slide:



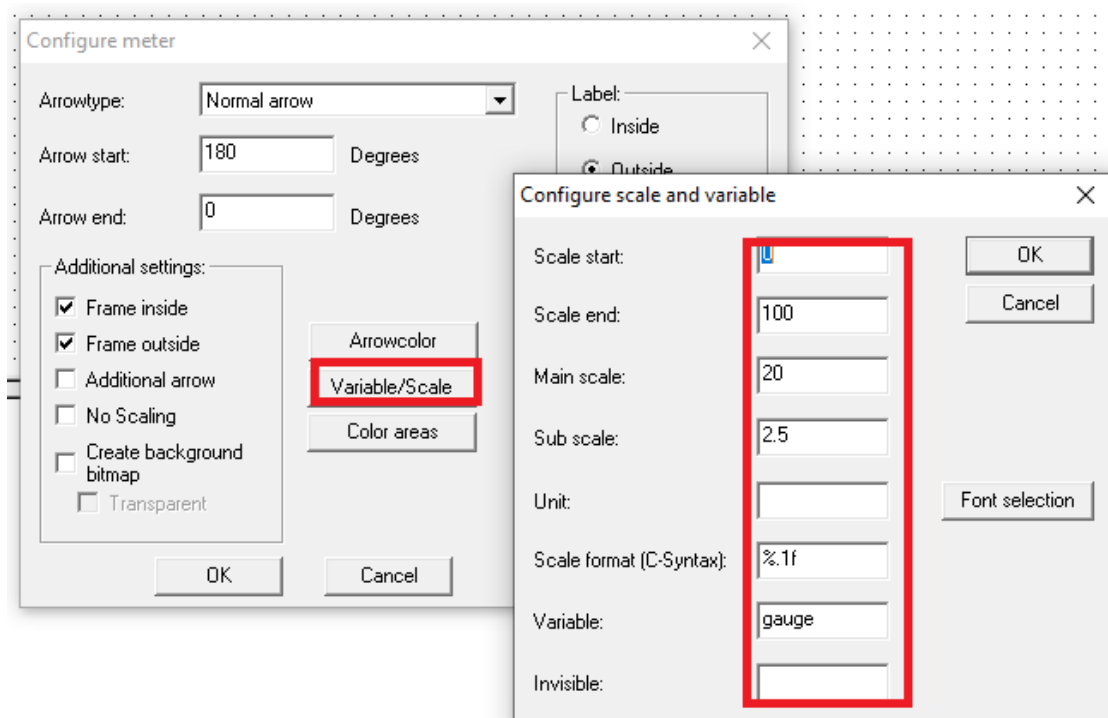
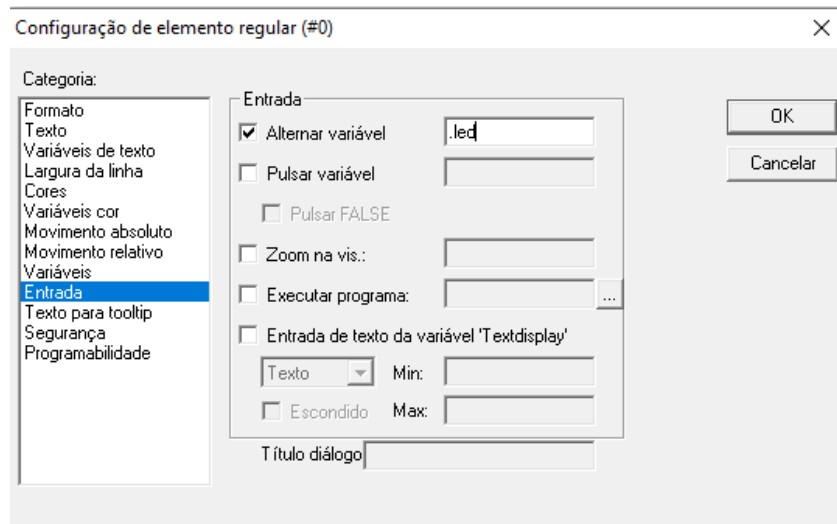
# Configurando a IHM

Configurando o led e o slide:



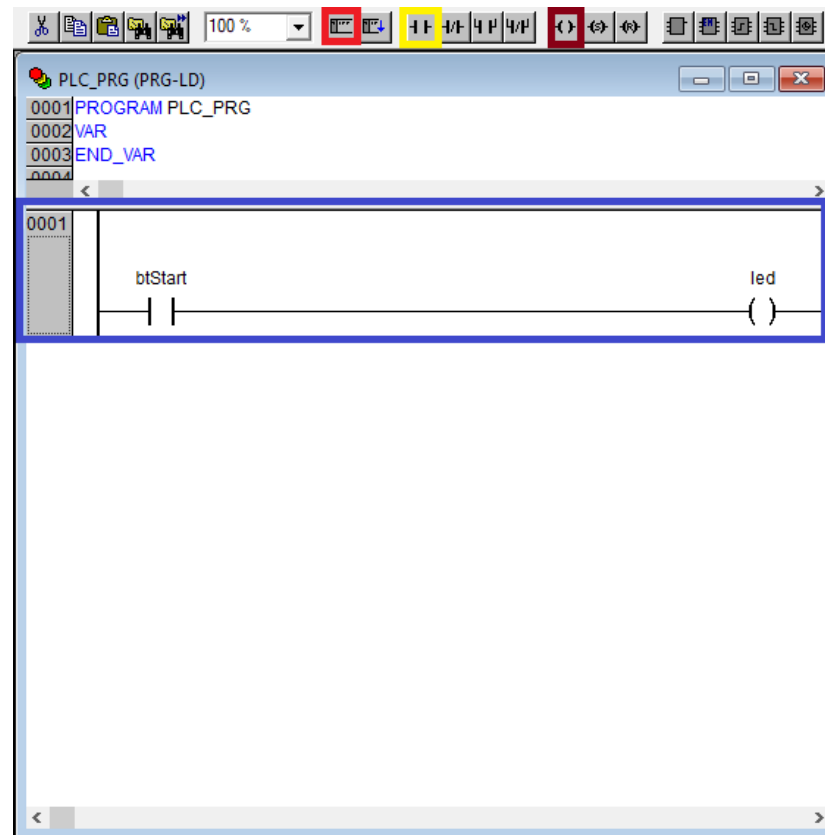
# Configurando a IHM

Configurando o botão e o gauge:



# Comunicação programa LADDER

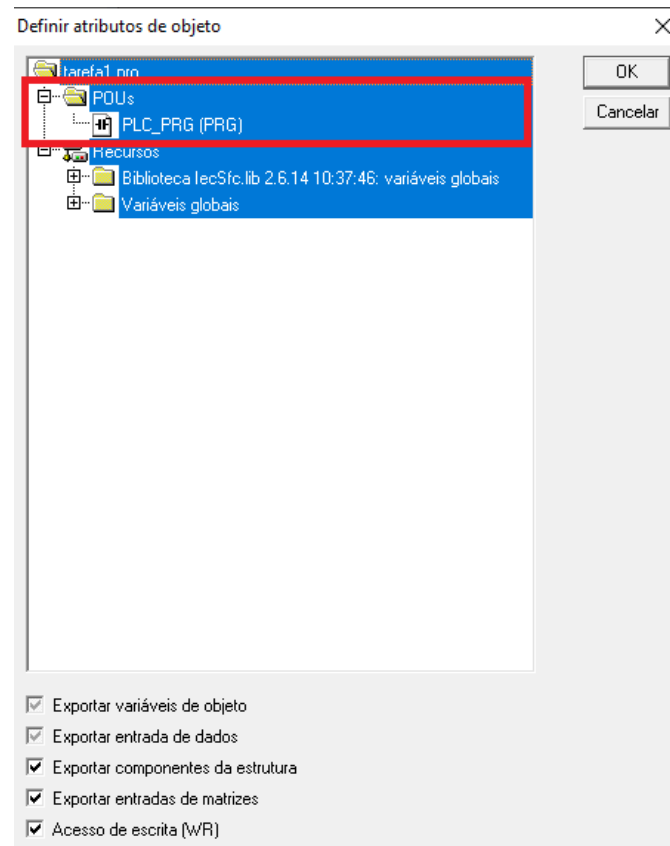
Para realizar a comunicação entre um programa LADDER no CodeSys com o VTSCADA é simples. Primeiramente fazemos um programa LADDER como no primeiro tutorial. (foi adicionado um variável btStart ao nosso programa anterior, desta forma temos que proceder de forma igual para fazer a ligação da variável com o VTSCADA).





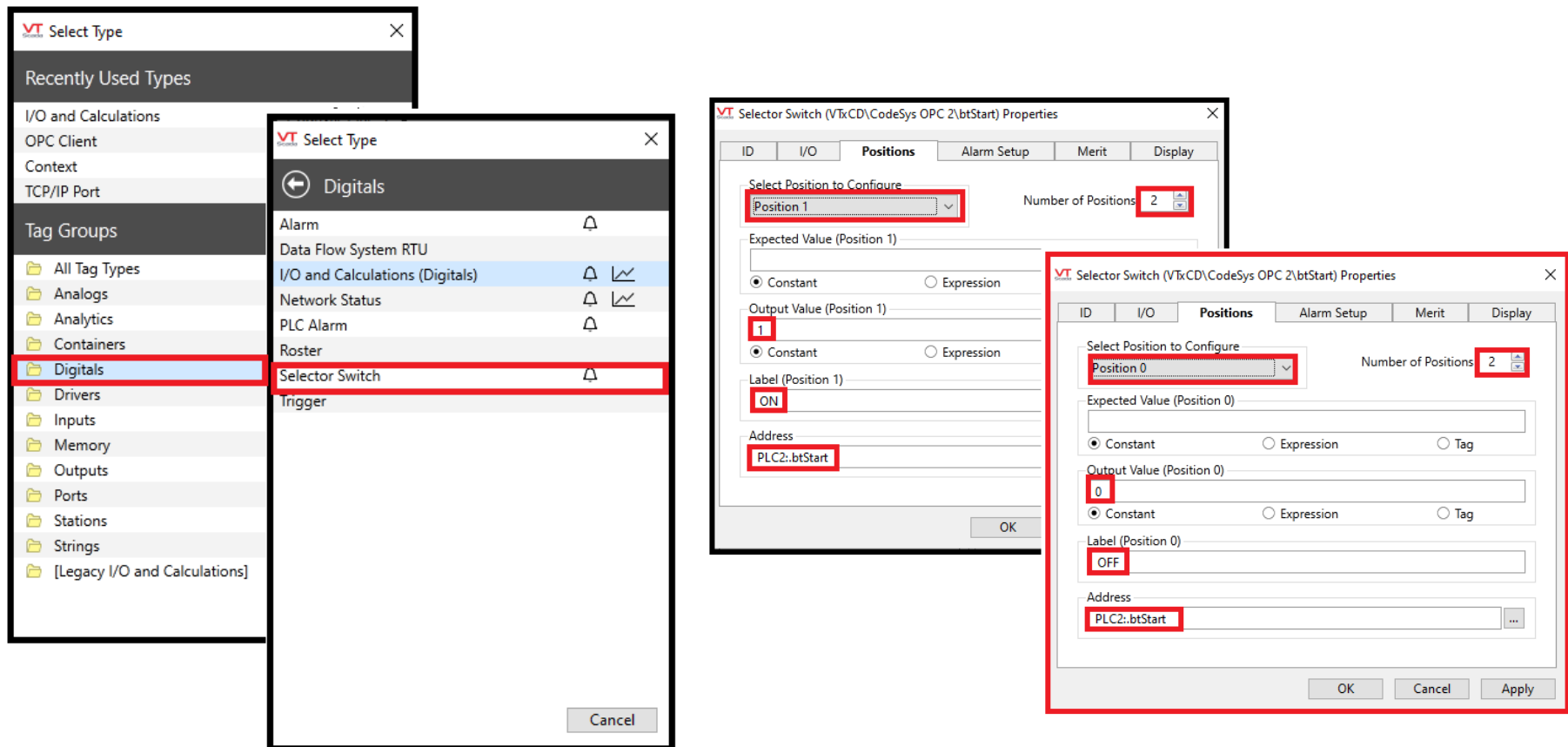
# Comunicação programa LADDER

Quando formos exportar, basta selecionar também o programa LADDER no POU para exportar junto com as variáveis:



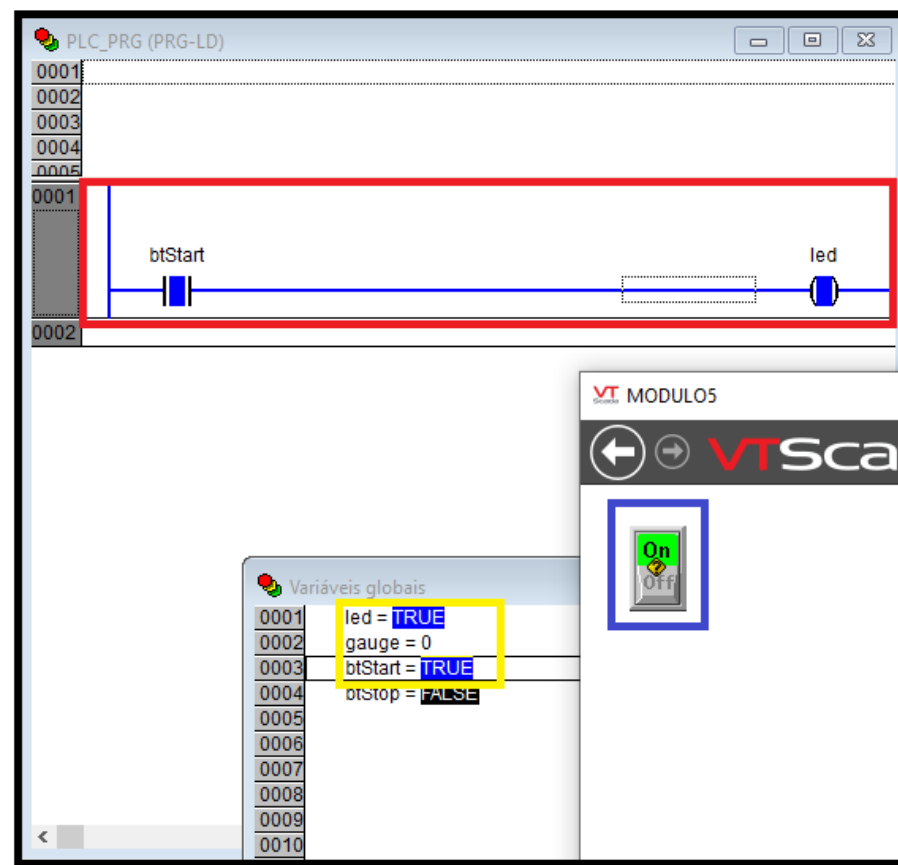
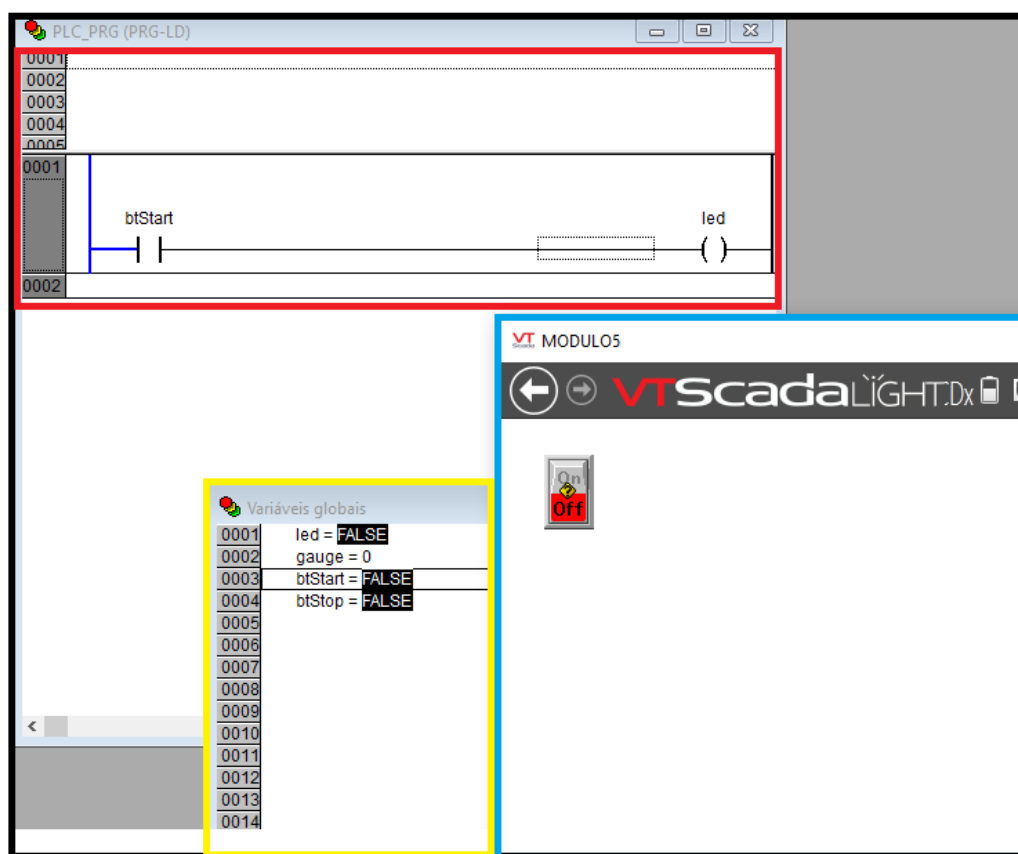
# Comunicação programa LADDER

Fazemos a tag para nova variável (desta vez digital e selector switch) e um botão para ela:



# Comunicação programa LADDER

O programa funcionará normalmente:



# FT primeira ordem

Agora iremos implementar um sistema de primeira ordem no nosso programa LADDER. Utilizaremos como base o programa anterior para mais facilidade:

Um sistema de primeira ordem tem a seguinte função de transferência:

$$Y(s)/X(s) = k.1/(s+a)$$

Aplicando a função Z inversa para sistema discretos temos que:

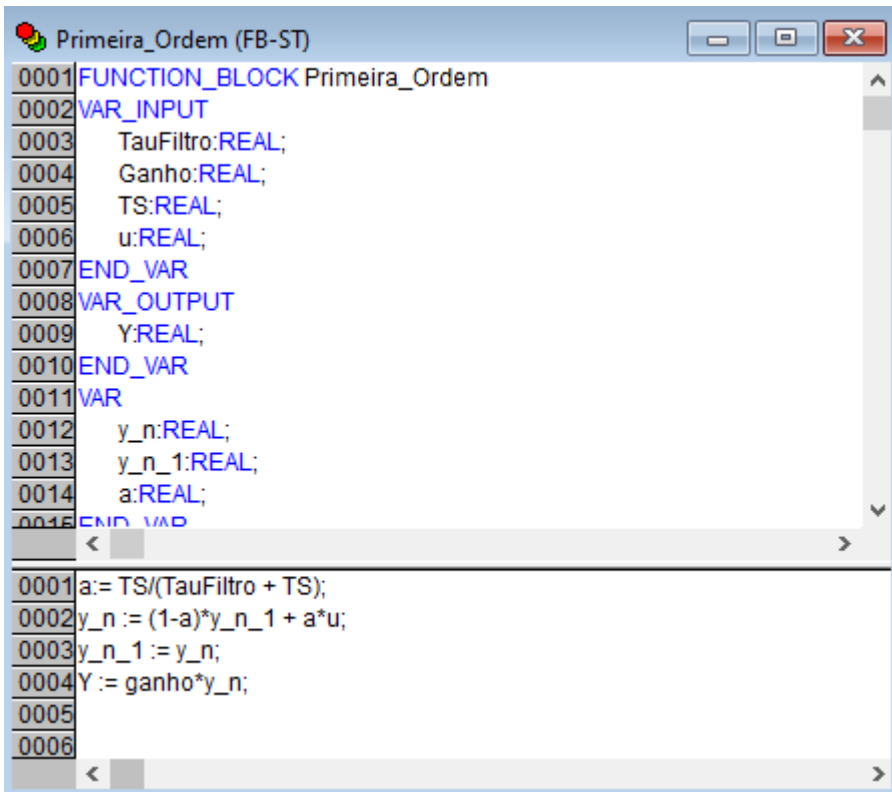
$$Y(t) = K.((1-p)y_{n-1}(t) + p.U(t))$$

Com  $p = \Delta T/(1/a + \Delta T)$

# FT primeira ordem

Vamos utilizar tudo que aprendemos até aqui para essa parte, por isso o passo a passo de criar a IHM no VTSCADA, além de declaração de variável globais e outros serão omitados.

Começamos criando o bloco funcional com o sistema de primeira ordem:



```
0001 FUNCTION_BLOCK Primeira_Ordem
0002 VAR_INPUT
0003     TauFiltro:REAL;
0004     Ganho:REAL;
0005     TS:REAL;
0006     u:REAL;
0007 END_VAR
0008 VAR_OUTPUT
0009     Y:REAL;
0010 END_VAR
0011 VAR
0012     y_n:REAL;
0013     y_n_1:REAL;
0014     a:REAL;
0015 END_VAR
0001 a:= TS/(TauFiltro + TS);
0002 y_n := (1-a)*y_n_1 + a*u;
0003 y_n_1 := y_n;
0004 Y:= ganho*y_n;
0005
0006
```

Lembrando que nosso sistema é discreto, logo precisamos sempre armazenar os valores de saída.

Usamos a função do nosso sistema no tempo como obtido no slide anterior.

# FT primeira ordem

Criamos as variáveis globais e a IHM no VTSCADA:

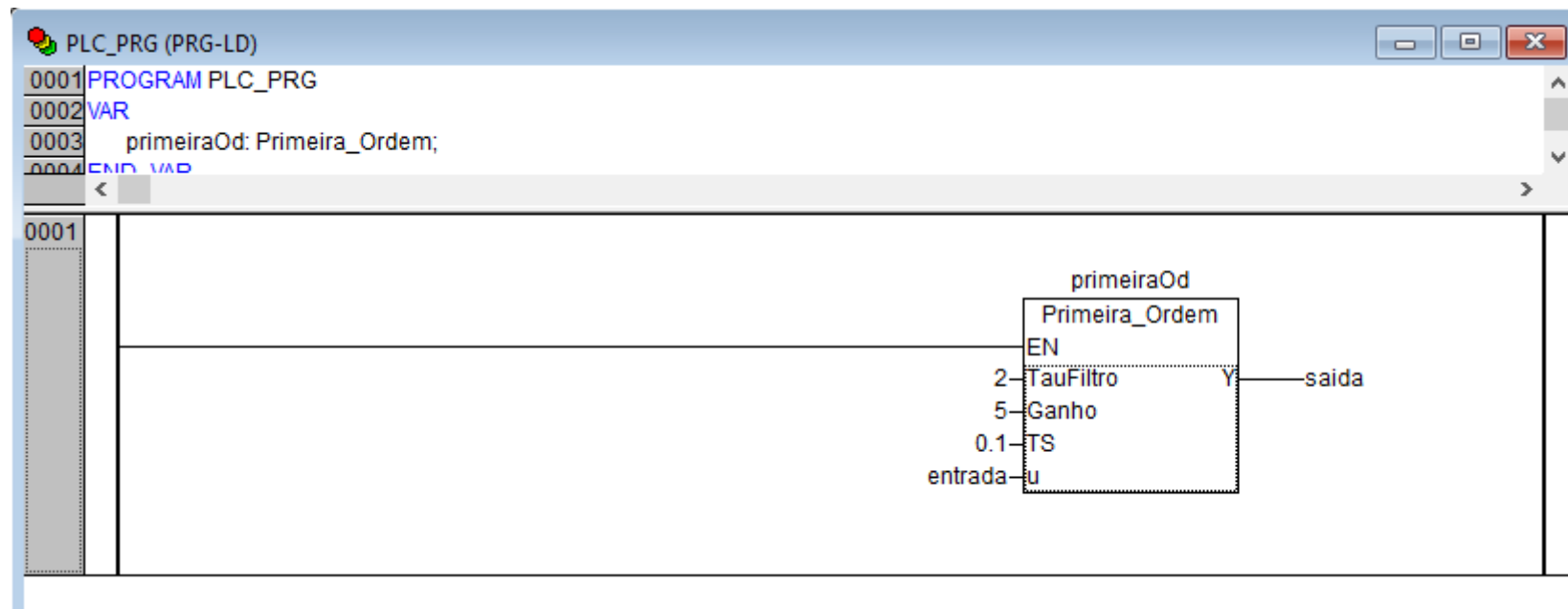
Um bloco para inserir o valor de entrada e um gauge para visualizarmos a saída.



0.0 ?

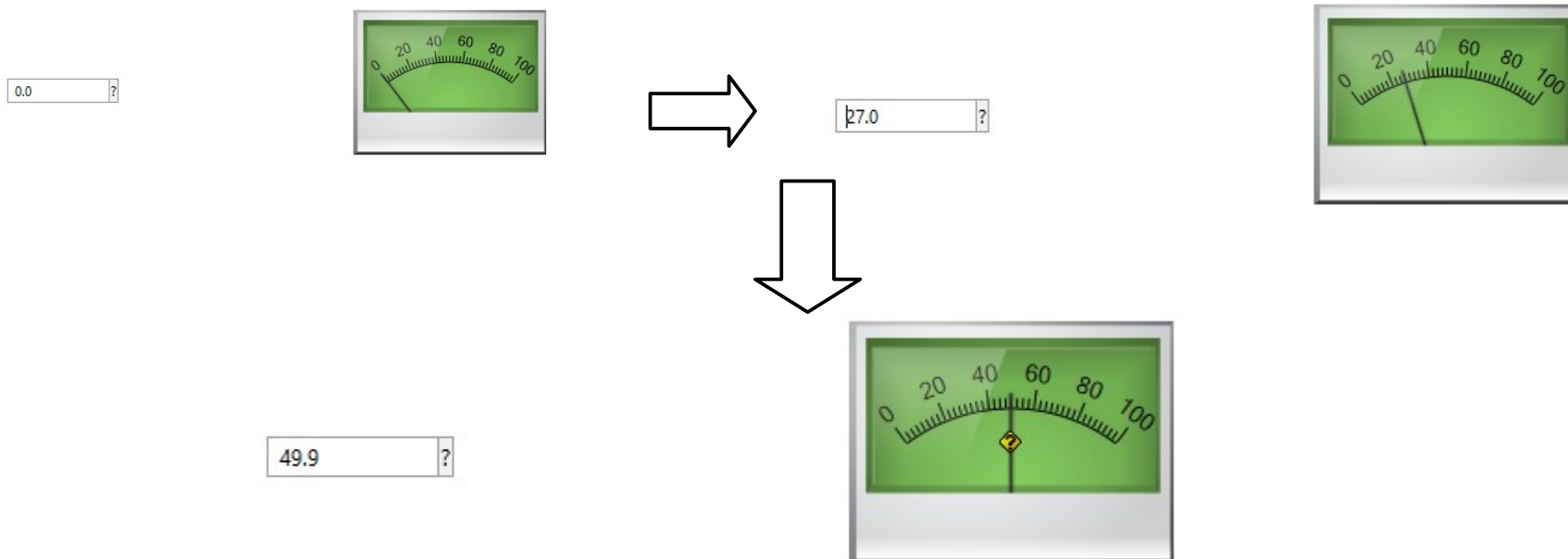


Criamos nosso programa em LADDER com uma instrução do nosso bloco programável:  
A entrada é o valor objetivo e a saída e o valor atual da nossa função. O ganho, Tau e DeltaT (TS) são valores definidos pelo usuário.



# FT primeira ordem

Depois de fazer todas conexões no OPC, PLC, configurar as saídas do Codesys e criar as tags no VTSCADA, o sistema está pronto: (Com o ganho 5, adicionamos a entrada 10 e a saída tende a chegar ao valor 50 com o tempo:





**Fim**