

# Programação SFC

Henrique Silva Coutinho

# Vantagens e desvantagens do SFC em relação ao Ladder

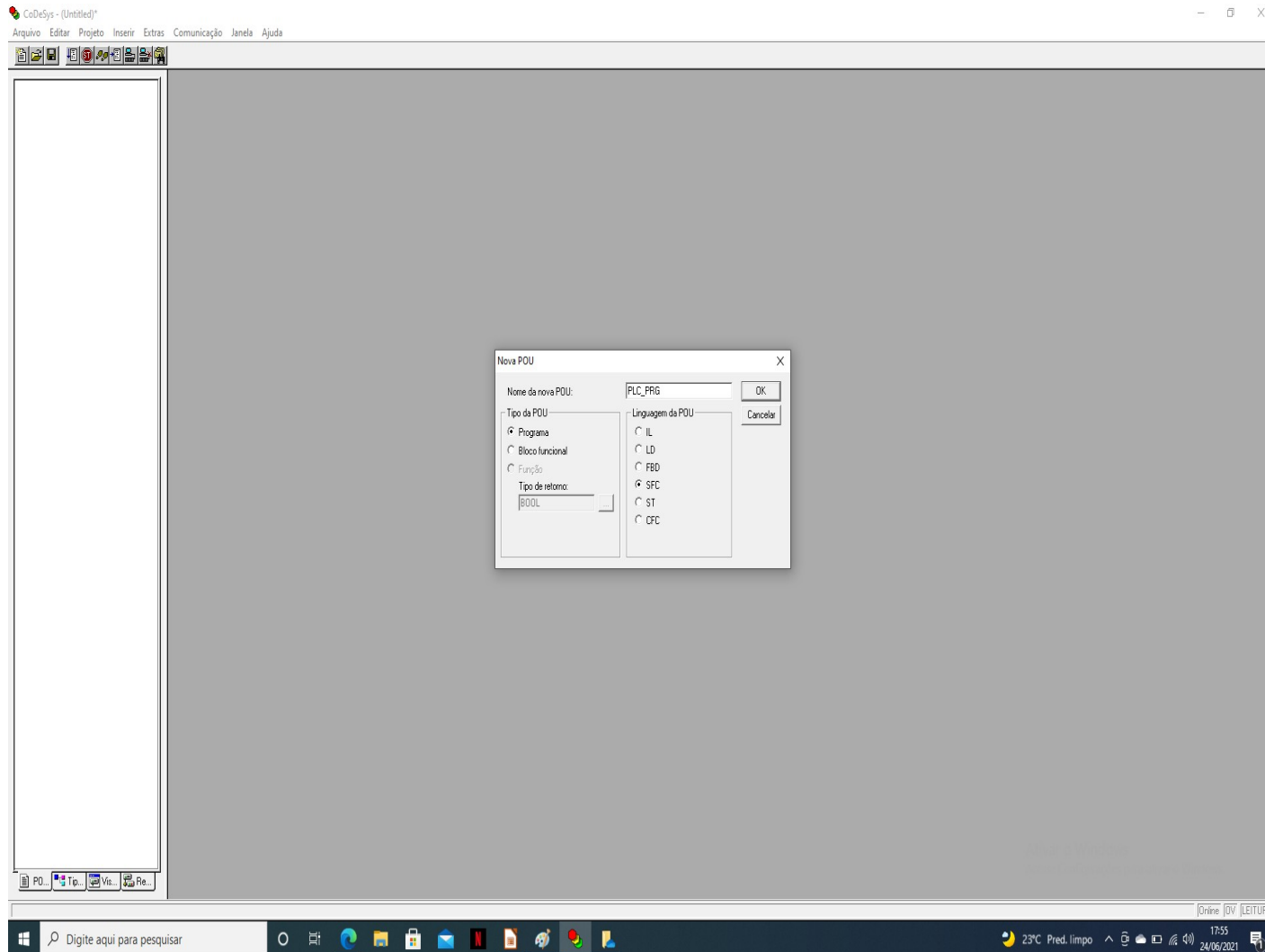
## Vantagens:

- Programação sequencial
- Sincronismo
- Blocos que permitem programação diversa
- Paralelismo de processamento

## Desvantagens:

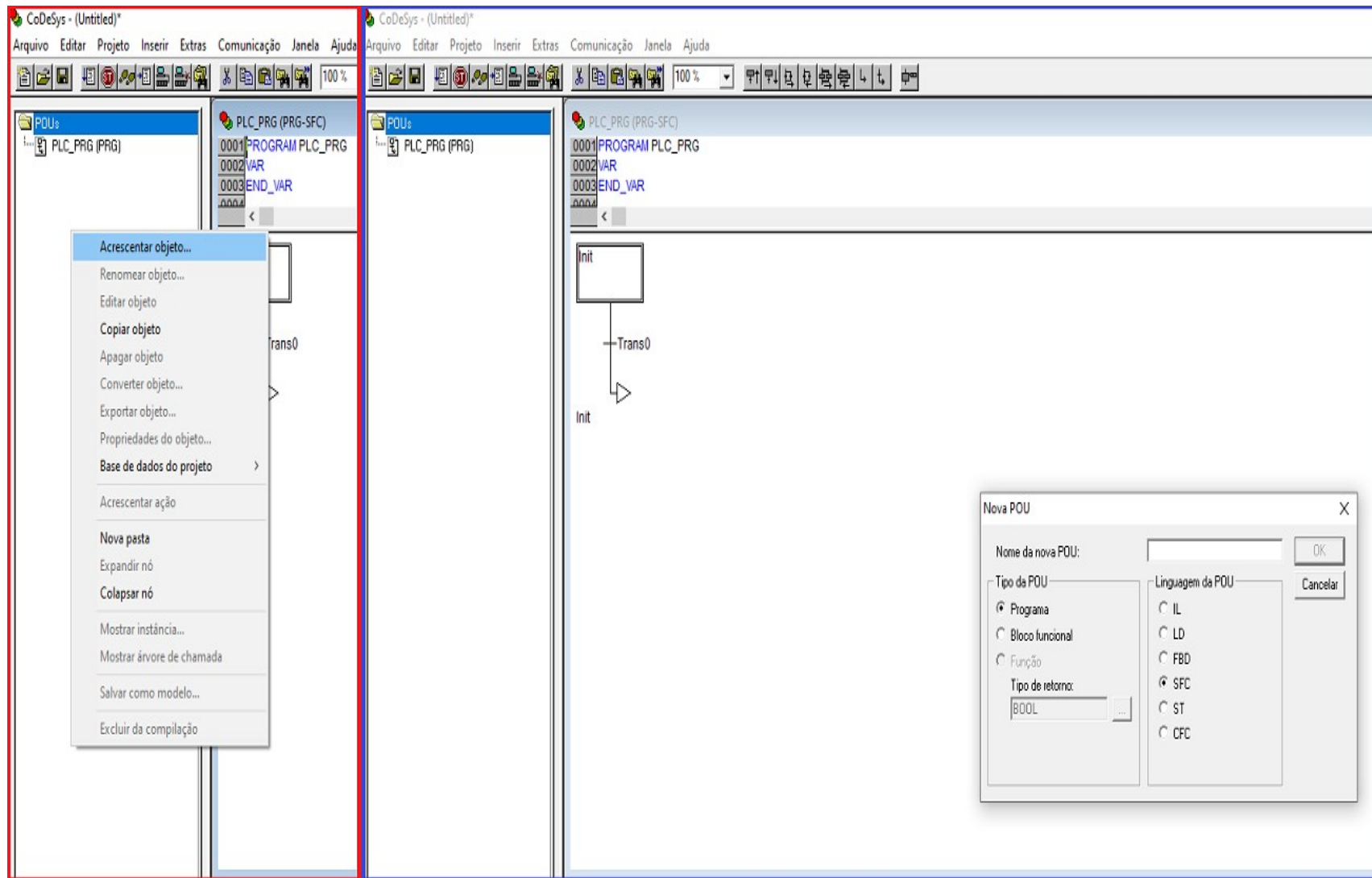
- Não possui algumas funções do Ladder
- Menos simples que o Ladder
- Intertravamento mais complexo que o Ladder

# Inicializando o SFC

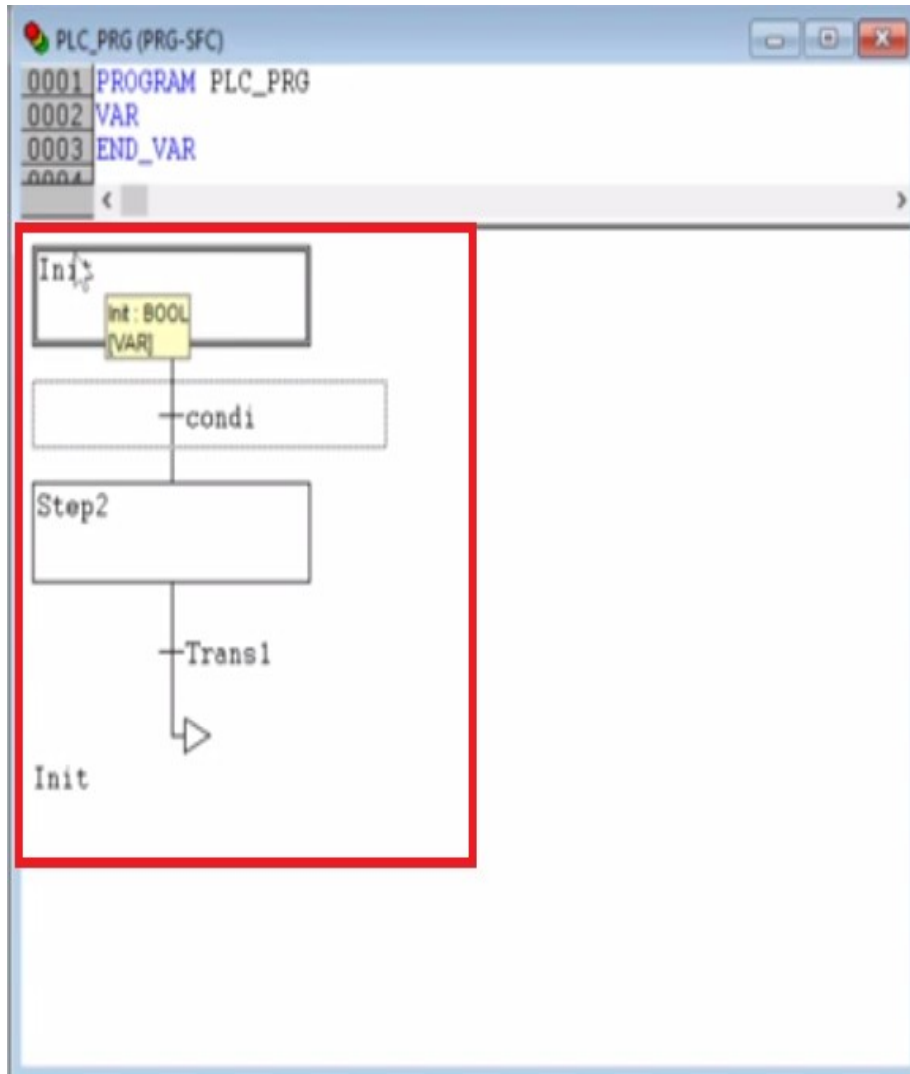


Ao Criarmos um novo projeto selecionamos a linguagem SFC da POU.

# Ou podemos acrescentar um objeto ao nosso projeto já iniciado

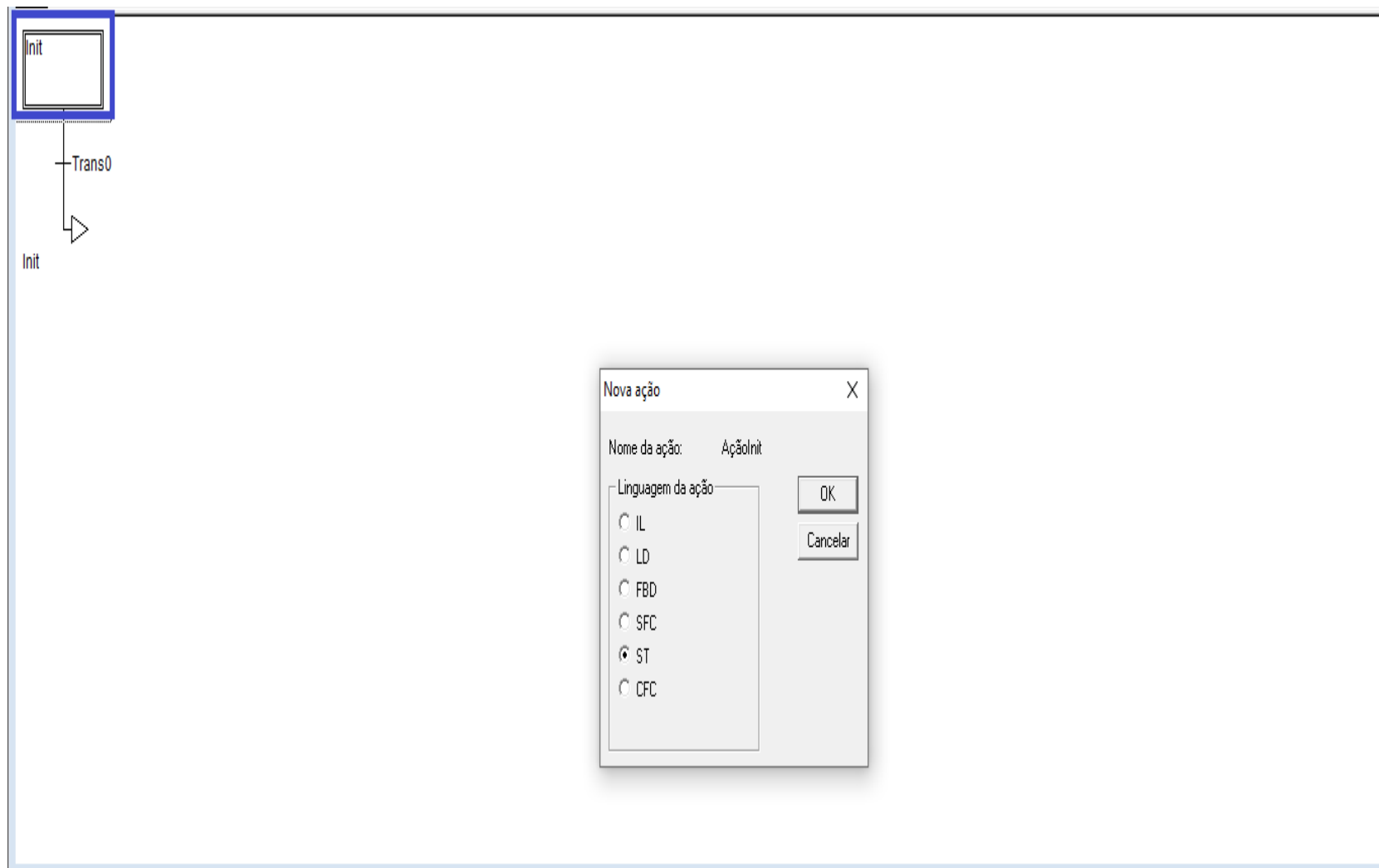


# Diagrama SFC

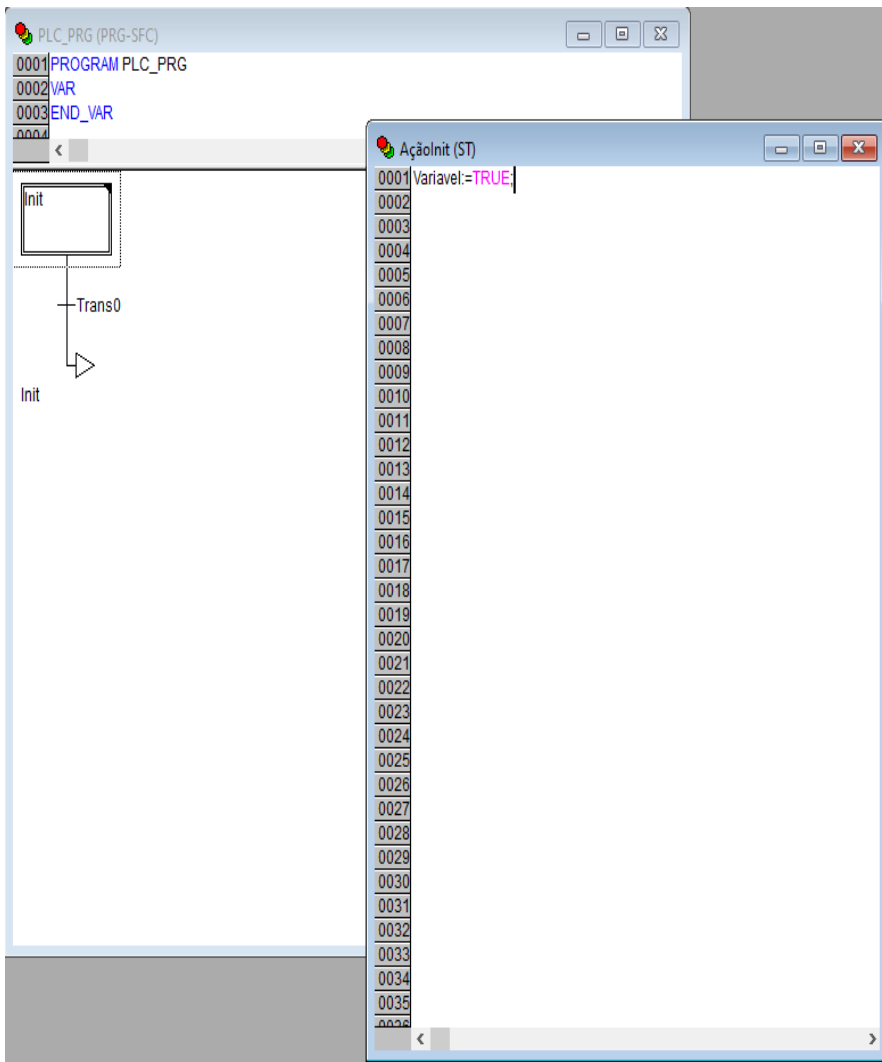


Um **diagrama SFC** possui o aspecto ao lado em **vermelho**, no qual há uma caixa que mostra o estado do sistema e dentro delas podemos inserir um programa.

# Programando



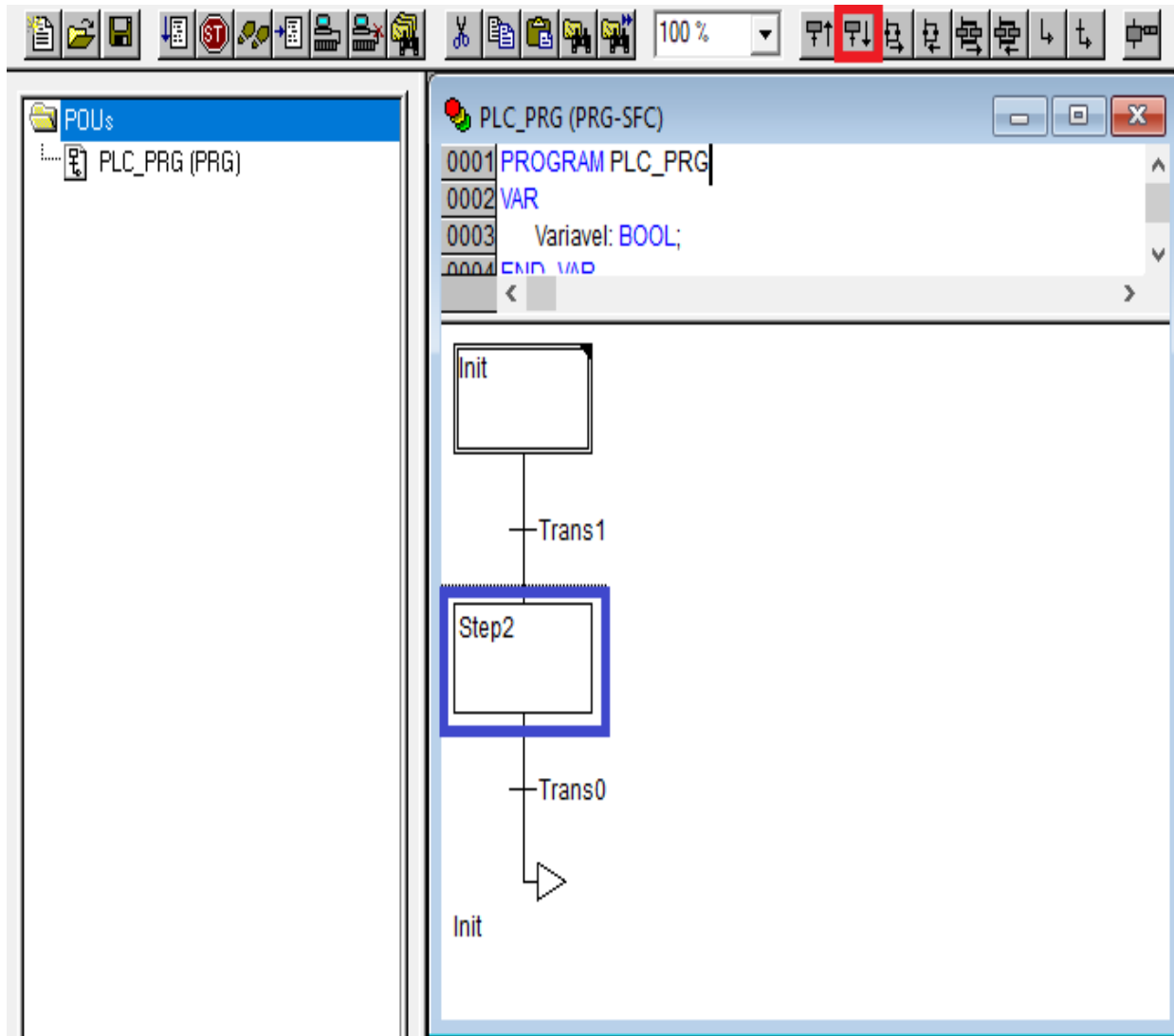
# Programando



Abrirá uma janela para programarmos com a linguagem selecionada (No nosso caso ST).

Podemos então interagir com o diagrama, nesse caso pegamos uma variável e colocamos seu estado para TRUE.

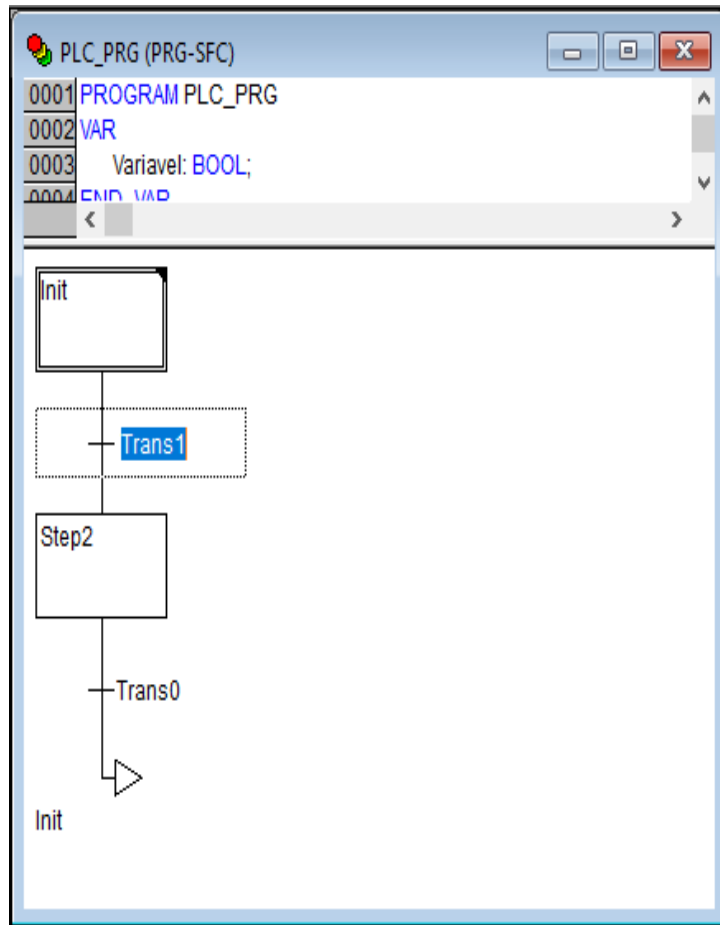
# Adicionando outro bloco e programação



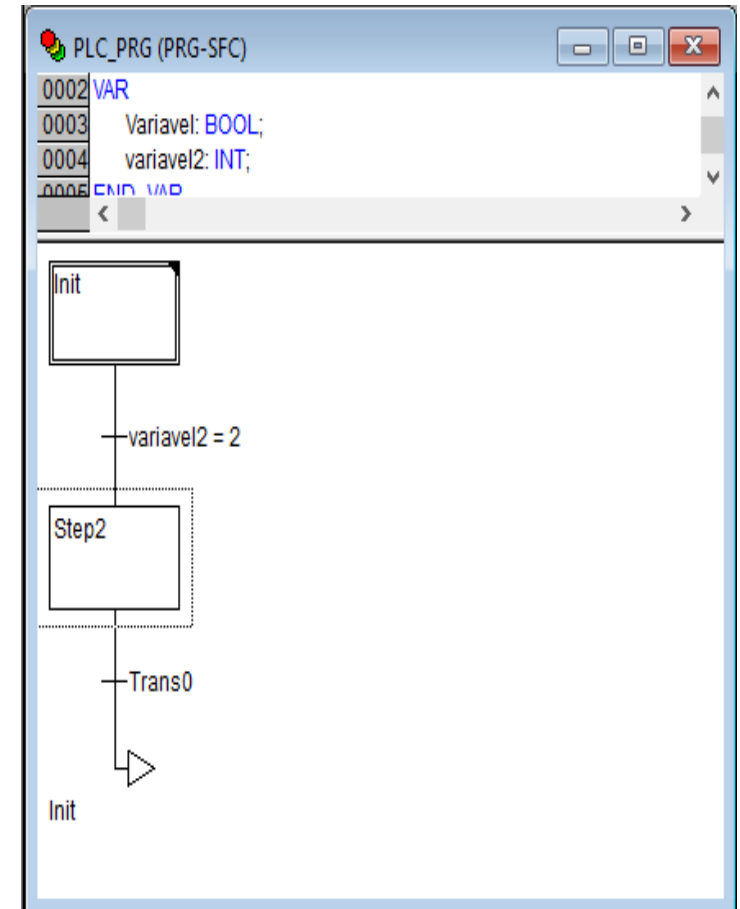
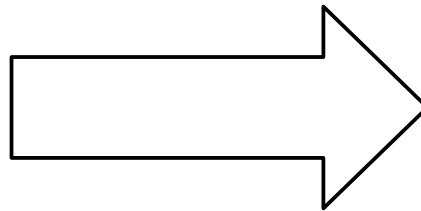
- Ao clicarmos no **ícone em vermelho** adicionamos um novo **bloco de programação em azul**.
- Observe que entre o primeiro e o segundo bloco há uma transição.
- A programação do segundo bloco só começará a operar quando a condição de transição for satisfeita.



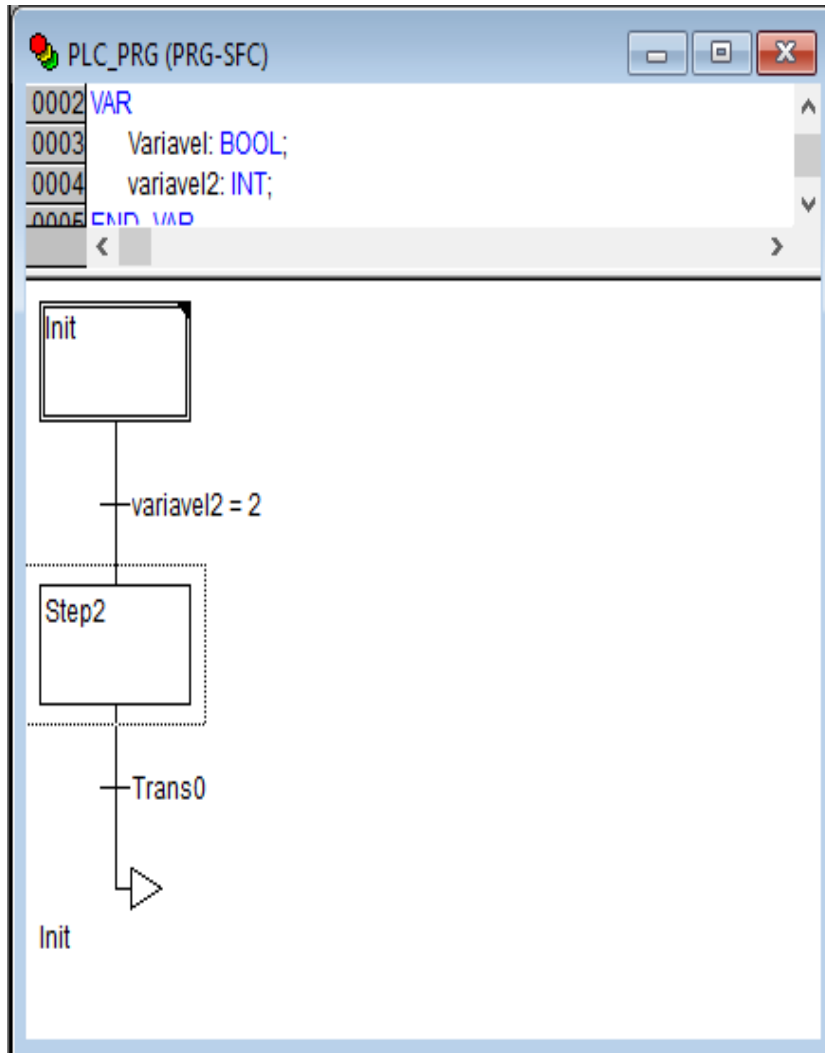
# Alterando a transição



Inserimos uma condição para a passagem do bloco 1 (init) e o bloco 2

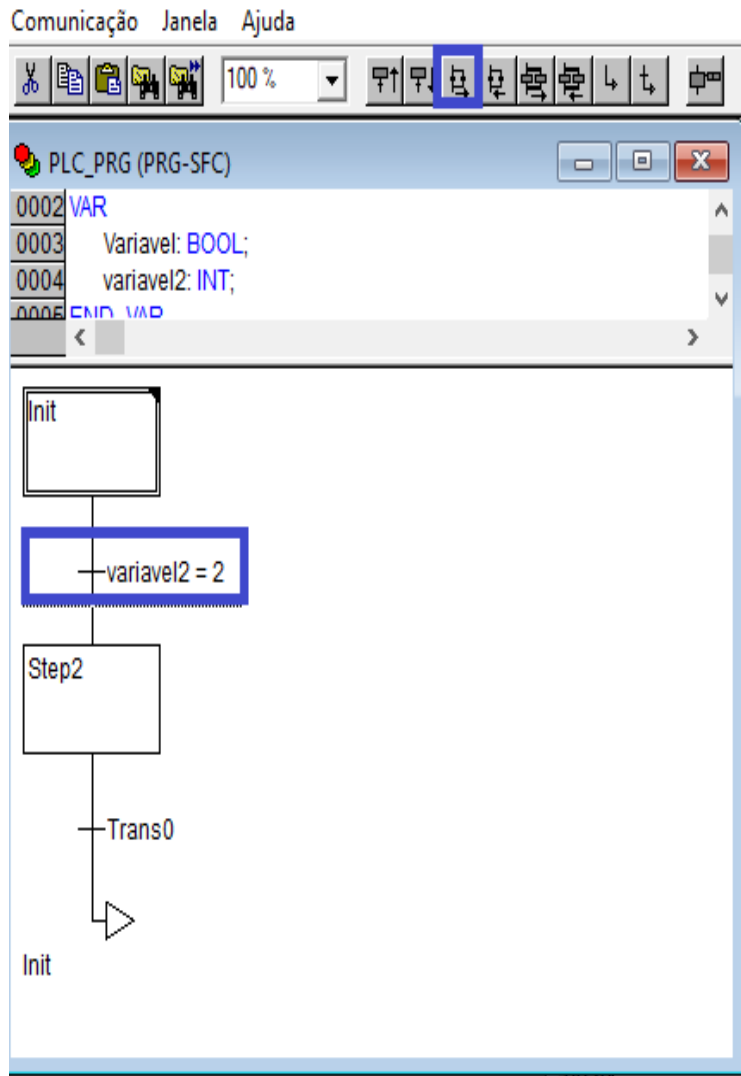


# Repetição do programa

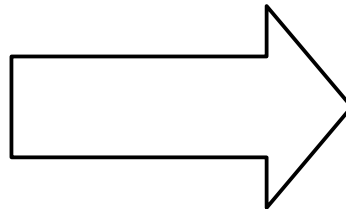


Abaixo do ultimo bloco há uma ultima transição que após ser cumprida faz com que o programa retorne ao bloco INIT e repita o processo.

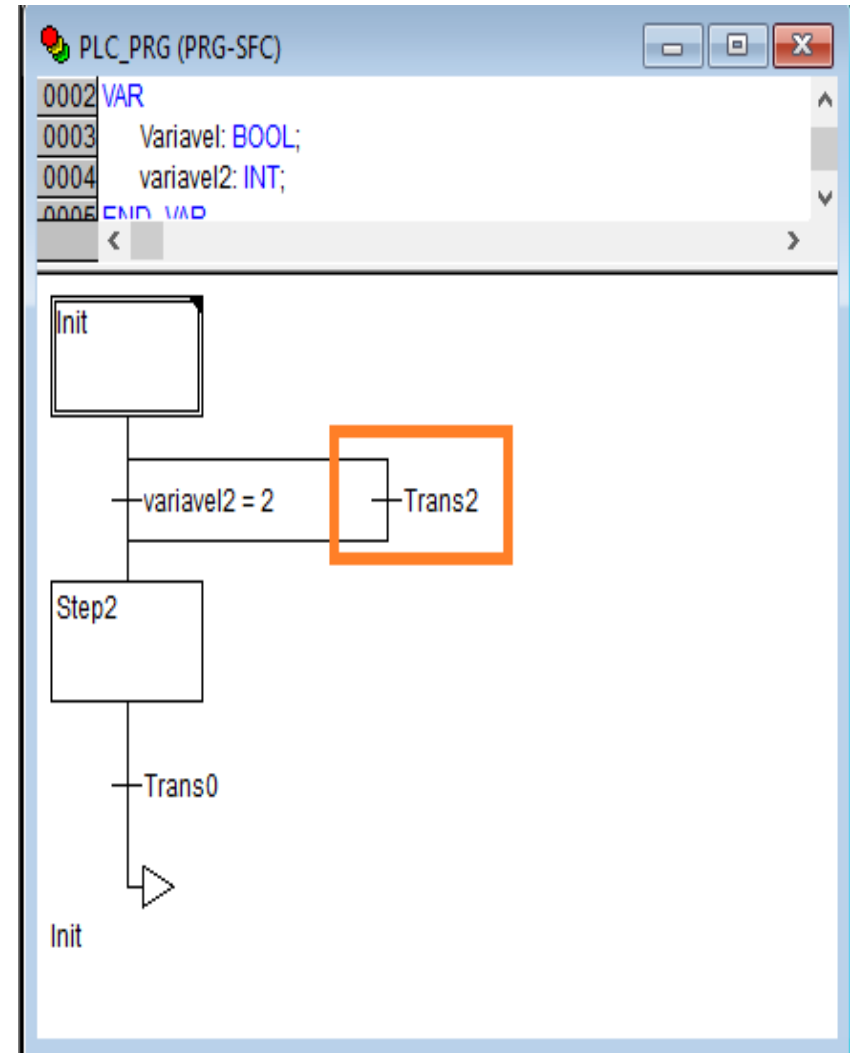
# Ramificação



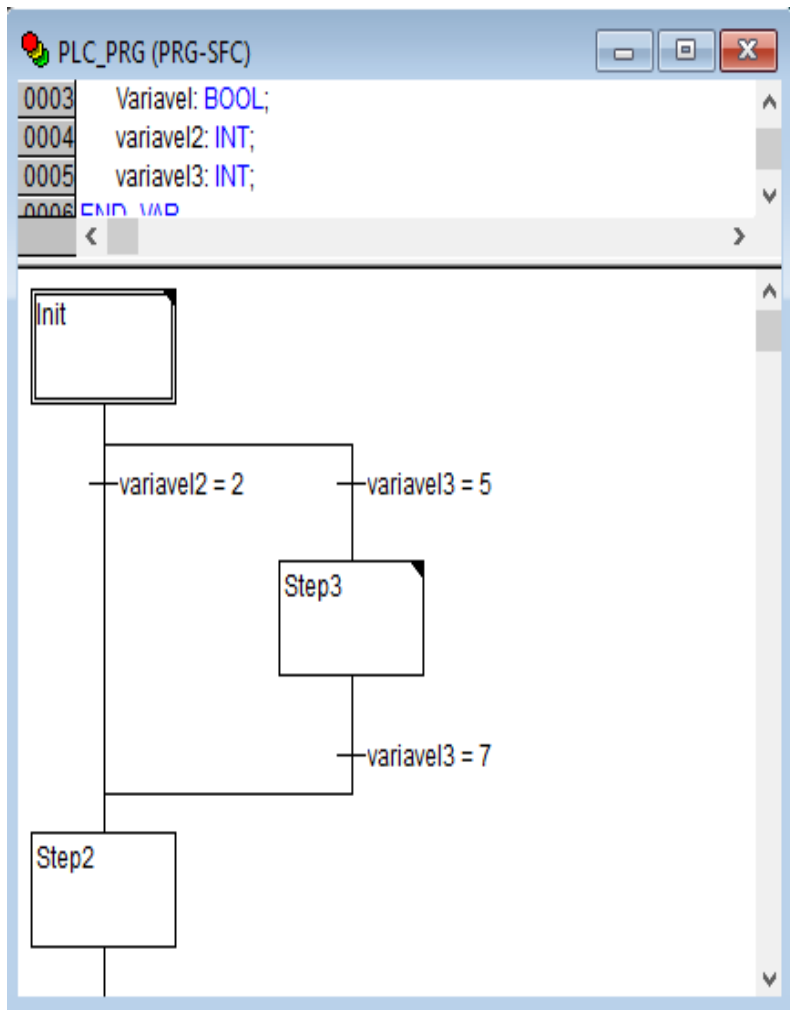
Ao selecionarmos um objeto do diagrama e clicarmos no ícone azul adicionamos um ramo concorrente no nosso diagrama.



Em **laranja** podemos observar o **novo ramo** composto por uma condição concorrente á primeira.



# Exemplo de ramificação com Blocos

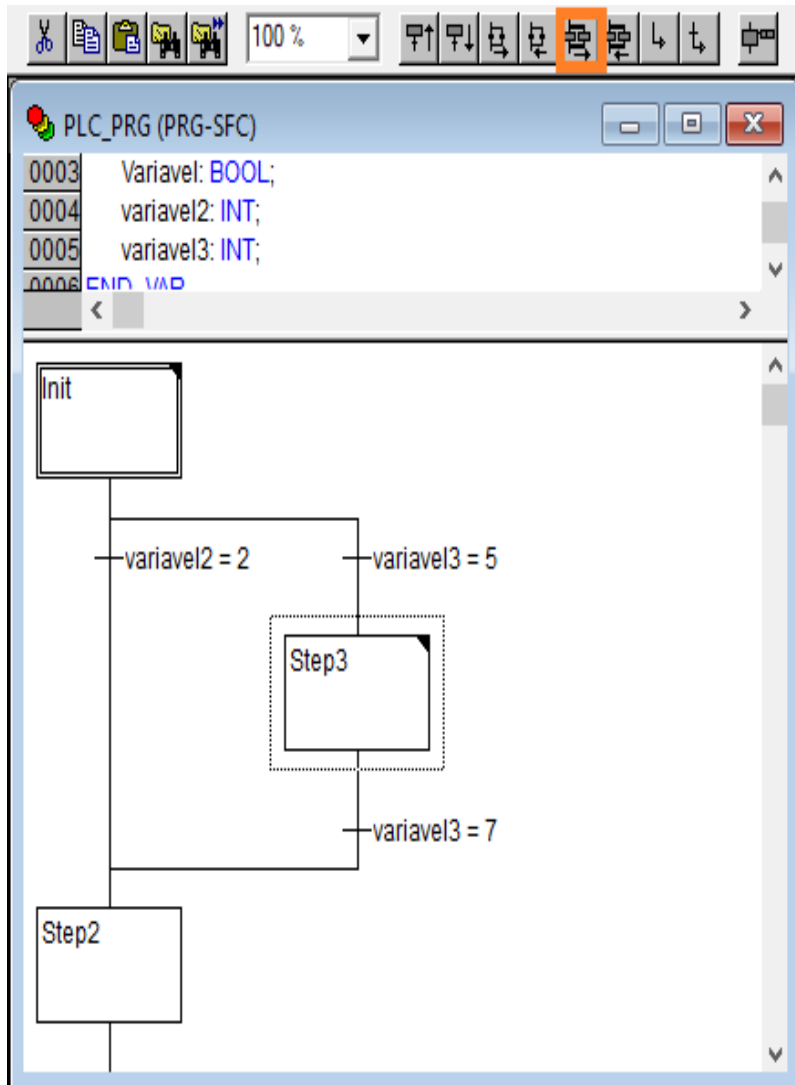


Podemos então adicionar blocos às nossas ramificações, dessa forma o programa passa a ter vários caminhos disponíveis de processamento.

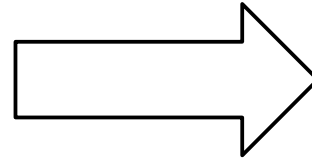
Nesse caso, o programa rodará o init. Se a condição `variavel2=2` for atendida, o programa prosseguirá para o bloco **Step2**, se a condição `variavel3 = 5` for atendida, o programa executará primeiramente o **Step3** até que a condição `variavel3 = 7` for atendida.

Se mais de uma condição for atendida, o programa dará prioridade da esquerda para a direita.

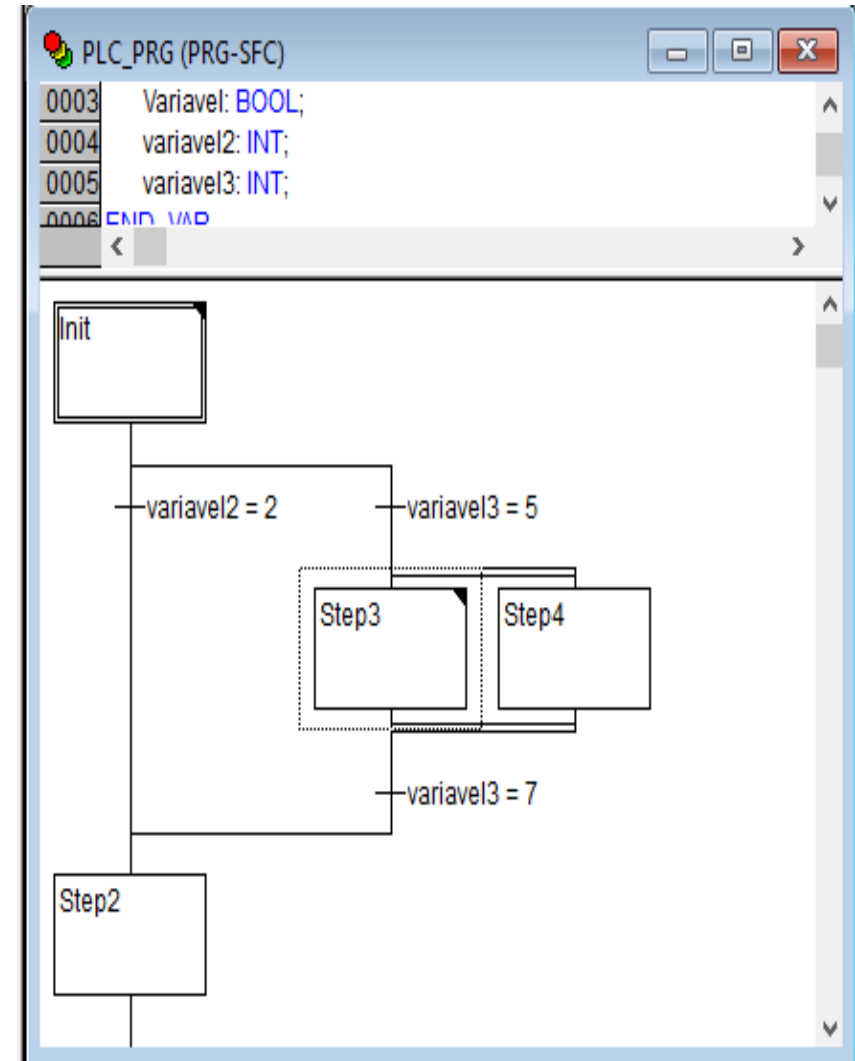
# Paralelismo



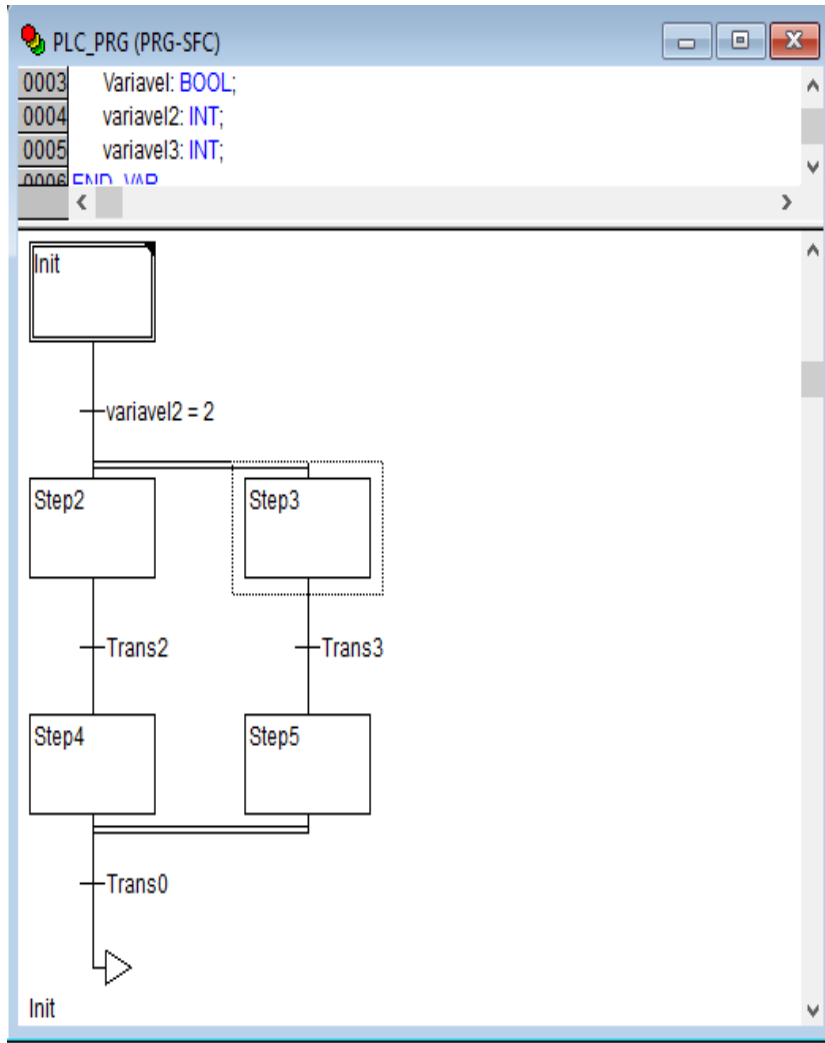
Ao selecionarmos um bloco e clicarmos no ícone em laranja adicionamos um bloco em paralelo com o primeiro.



O novo bloco Step4 está em paralelo ao Step3. Eles serão executados simultaneamente



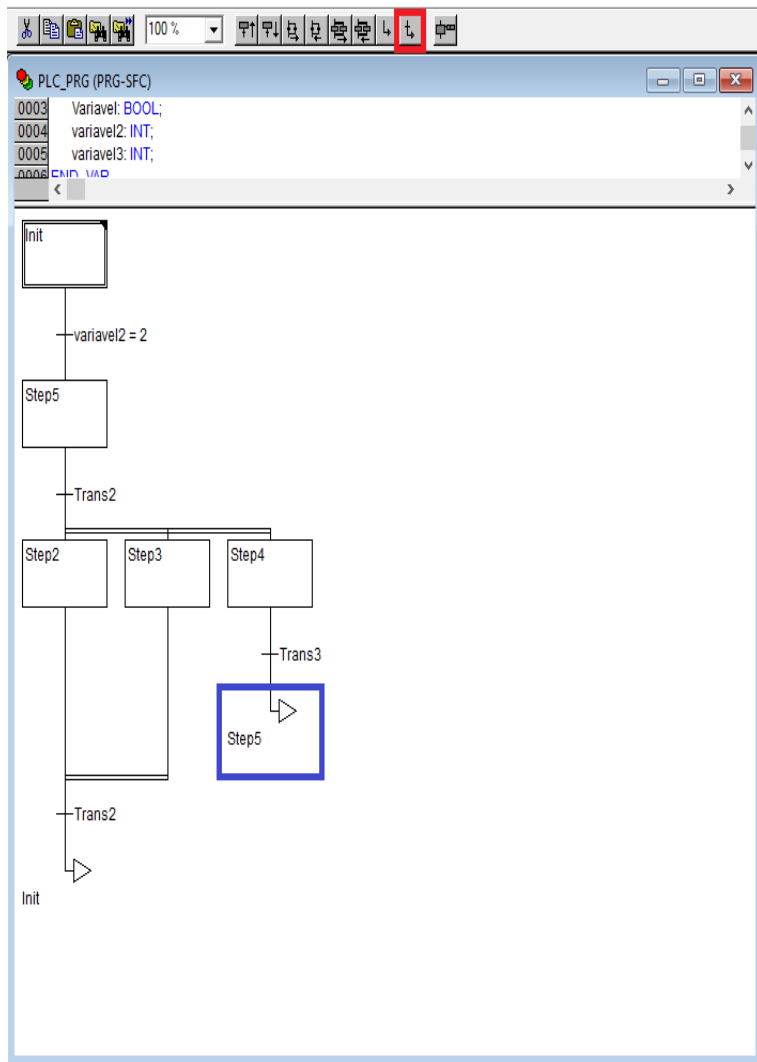
# Sincronismo



Em processos paralelos há um sincronismo. Desta forma o programa só voltará ao ramo principal após a todos os processos do ramo paralelo serem realizados.

Nesse exemplo, a condição Trans0 só passará a funcionar se o processo do ramo esquerdo estiver em Step4 e o processo do ramo direito estiver em Step5.

# Salto

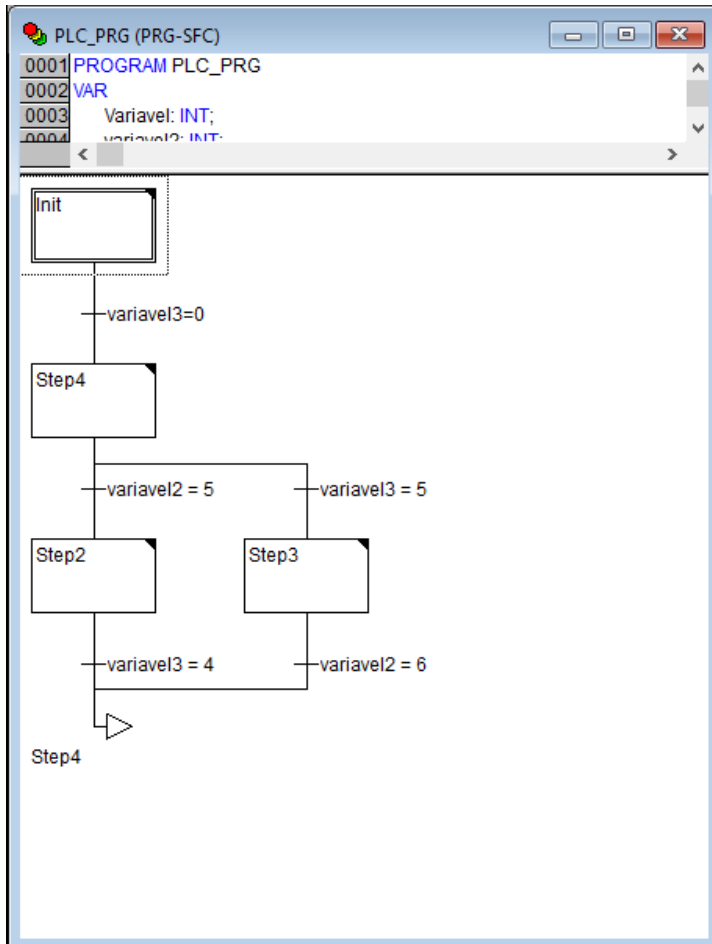


Podemos também fazer saltos entre processos em nosso programa

Ao selecionarmos um bloco e pressionarmos o **ícone em azul** adicionamos **um salto** que leva até outro bloco.

Nesse exemplo, ao executarmos o Step4 e a condição Trans3 for atendida o programa retornará ao Step5.

# Exemplo final



Bloco init:  
Variavel2 := 10;  
Variavel3 := 0;

Bloco Step4:  
Variavel2 := variavel2 - 1;  
Variavel3 := variavel3 + 1;

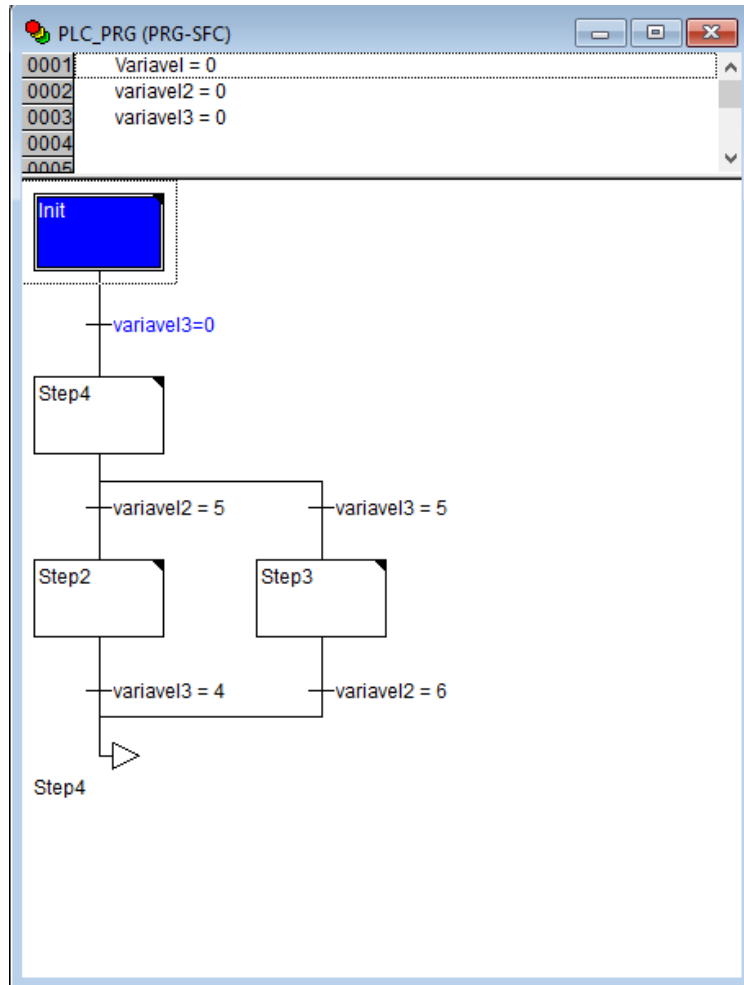
Bloco Step2:  
Variavel3 := 4;

Bloco Step3:  
Variavel2 := 6;

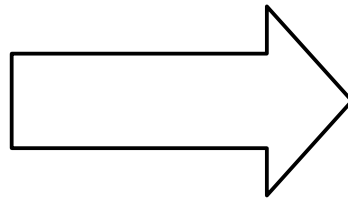


# Exemplo

## Clock 1

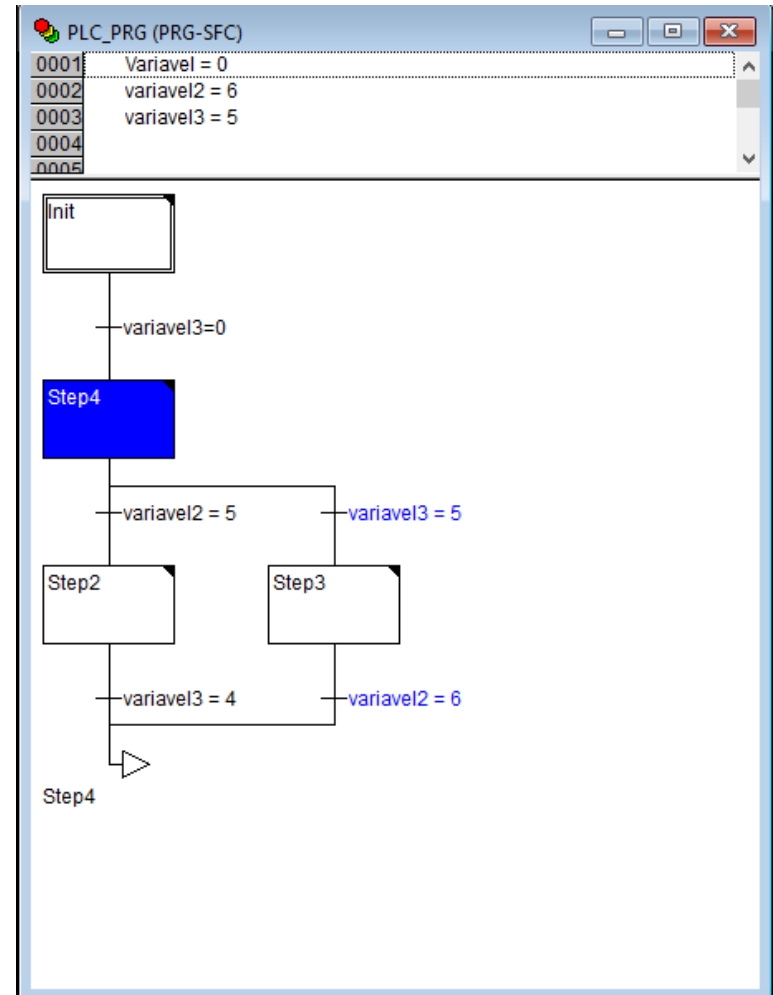


Apesar de em init o valor de variavel2 ser setado para 10, como a condição  $\text{variavel3} = 0$  já está sendo satisfeita, o programa ignora o que está em init



Assim variave3 chega em 5 primeiro e é executada primeiro.

## Clock 2



## Exemplo

Após isso o programa ficara infinitamente alternando entre Step2 e Step3.