

**Universidade de Aveiro**  
**Escola Superior de Tecnologia e Gestão de Águeda**

**Relatório de Programação PLC de uma  
máquina de Enchimento de Caixas**

Gonçalo Alexandre Silva Figueira, MEC 109438

**Trabalho Prático 1 da UC de Automação Industrial**  
Curso Técnico Superior Profissional em Instalações  
Elétricas e Automação

**Trabalho efetuado sob a orientação de:**

Professor Luís Moutinho

Dezembro de 2022



# Índice

Introdução .....	5
Descrição do Funcionamento da Máquina .....	6
Tabela de Endereços .....	8
Entradas .....	8
Saídas.....	8
Memórias .....	8
Contadores .....	9
Timers .....	9
Simbologia Utilizada .....	10
Contacto Normalmente Aberto.....	10
Contacto Normalmente Fechado .....	10
Saídas.....	11
Set e Reset .....	11
Função menor que .....	12
Contador .....	12
Temporizadores .....	13
Diferenciadas .....	15
Rungs .....	15
Desenvolvimento do Projeto .....	17
Estado de Repouso .....	17
Arranque da Máquina .....	19
Caixa Detetada.....	20
Estado Caixa Cheia.....	22
Caixa Não Cheia.....	23
Loop de ciclo.....	23
Paragem de Ciclo - Stop.....	25

Limite de Caixas Cheias .....	26
Testes realizados.....	27
Teste de Funcionamento.....	28
Teste de Pressionar Start e Stop ao Mesmo tempo.....	31
Conclusão .....	33

# Introdução

Em prol do primeiro projeto da Unidade Curricular de Automação Industrial foi solicitado que se desenvolvesse a programação em linguagem Ladder para uma máquina de Enchimento de caixas.

Este trabalho tem um peso de 10% da nota final (2 valores) da UC, e foi acompanhado pelo professor *Luís Moutinho* durante o decorrer da disciplina.

O meu projeto foi elaborado na plataforma de programação e simulação *WinProLadder versão 3.30* (simulador na versão 1.2), da *Fatek*, e o autómato selecionado foi o *FBs-20MC*, face à dificuldade que eu tinha em utilizar a plataforma da *Siemens (TIA Portal)* para elaborar o projeto.

Para a realização deste trabalho constaram alguns pontos que tive em atenção:

- O desenvolvimento funcional;
- A apresentação e fácil interpretação da lógica utilizada;
- Os ensaios técnicos do programa;
- A correspondência das entradas e saídas usadas no programa a entradas possíveis a serem usadas numa aplicação real;
- A especial atenção a que tive para que as entradas ativassem variáveis internas que por sua vez realizavam saídas, para melhor diagnosticar algum problema em caso de falha;
- A organização e construção de uma tabela de endereços, e os comentários a identificar a ação de cada linha, à medida que ia desenvolvendo o programa;
- Entre outros, ...

# Descrição do Funcionamento da Máquina

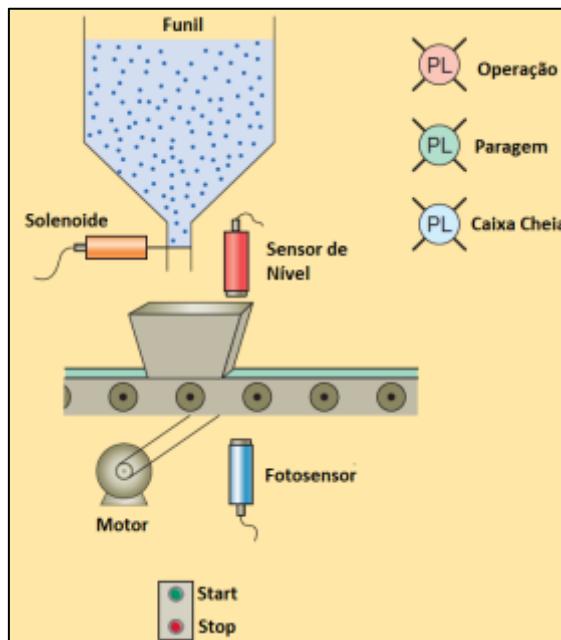
- A máquina é constituída por um botão de Start Normalmente Aberto que inicia o processo ao ser pressionado.

Embora numa condição real, seriam colocados dois botões para realizar um bimaterial, não inclui isso no meu programa por não estar incluído no enunciado.

- Possui também um Botão de Stop que para todo o processo ao ser pressionado, que é do tipo normalmente fechado.

Nessa condição todos os processos em curso são parados e uma luz de paragem é ativada.

Presumi que a luz de paragem seria de cor vermelha. E que esse botão de Stop iria assumir o lugar de uma betoneira de emergência.



- Existe um tapete que é movimentado com um motor, que deve arrancar imediatamente após o processo iniciar.

Assumi que este motor deveria de ser um motor acionado por um contactor (ligado a uma saída) ou um motor DC, embora não faça diferença nenhuma na programação.

- Quando o foto-sensor detetar a face lateral direita de uma caixa envia uma ordem para parar o tapete.
- Com a caixa parada e posicionada por baixo do funil, que é ao lado do foto-sensor. Abre-se uma Solenoide e é ativada a válvula do funil.

Imaginei que as electroválvulas do solenoide fossem ligadas a uma relé, que no bit 1 estaria ativa (Electroválvula YV1 - Aberta) e no bit 0 estaria desativa (Electroválvula YV2 – Desativa).

- Por cima da caixa, junto ao funil está também um sensor de nível que deteta quando a caixa está cheia, caso o sensor não detete, ao fim de 5s da caixa parada, a Solenoide fecha.
- E após 10 segundos da caixa cheia, ou de terem passado os 5 segundos iniciais o tapete arranca automaticamente novamente.
- O processo é parado quando são cheias 10 caixas. E só pode retomar após uma nova ordem de Start do Operador.
- Existe também uma Luz de Operação que deve estar ligada enquanto o processo estiver a operar. A luz de Operação assumi que fosse de cor verde.
- E existe também uma Luz de Caixa Cheia que deve de ser ligada quando a caixa fica cheia. Esta luz deve de permanecer ligada enquanto a caixa estiver a ser deslocada e deixar de ser detetada pelo foto-sensor. A luz de Caixa cheia seria de cor amarela.

# Tabela de Endereços

## Entradas

Endereços	Nomenclaturas	Descrição
X1	BP_Start	Botão de pressão Start NA
X2	BP_Stop	Botão de pressão Stop NF
X3	Q_Fotosensor	Foto-sensor de deteção de Caixa
X4	Q_SensorNivel	Sensor de Nível de Líquido das Caixas

## Saídas

Endereços	Nomenclaturas	Descrição
Y1	H_Operation	Luz de Operação (Verde)
Y2	H_Paragem	Luz de Paragem (Vermelha)
Y3	H_CxCheia	Luz de Caixa Cheia (Amarela)
Y4	MT	Motor do Tapete
Y5	YV1-YV2	Solenóide Enchimento de Caixas

## Memórias

Endereços	Nomenclaturas	Descrição
M0	M_Repouso	Memória Retentiva de Estado de Repouso
M1	M_Start	Memória Retentiva de Estado de Start
M2	M_Stop	Memória Retentiva de Estado de Stop
M10	M_Avançar Ciclo	Memória Retentiva Intermediária de Avanço de Ciclo
M11	M_RetomarCiclo	Memória Retentiva Intermediária de Retomar Ciclo
M20	M_Contador	Memória Retentiva Intermediária do Contador
M30	M_Avançar	Memória Retentiva Intermediária de Avançar
M40	M_Timer200	Memória Retentiva Intermediária do Timer 200

M50	M_CxCheia	Memória Retentiva de Estado de Caixa Detetada
M60	M_Timer255	Memória Retentiva Intermediária do Timer 255

## Contadores

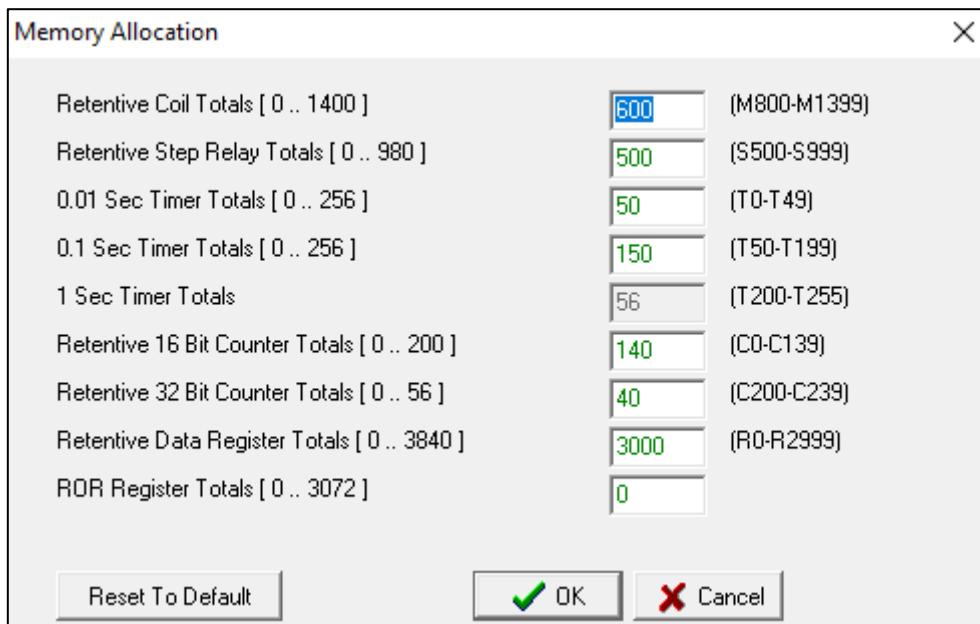
<b>Endereços</b>	<b>Nomenclaturas</b>	<b>Descrição</b>
C1	ContaCxCheias	Contador de Caixas Cheias com Limite de 10 caixas

## Timers

<b>Endereços</b>	<b>Nomenclaturas</b>	<b>Descrição</b>
T200	Timer5s	Timer de Enchimento de Caixas – 5 segundos
T255	Timer10s	Timer de Espera – 10 segundos

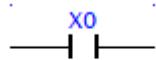
# Simbologia Utilizada

No meu programa utilizei alguns símbolos pré-disponibilizados pelo WinProLadder, que estão associados a endereçamentos. Conforme a nomenclatura associada, têm diferentes objetivos. Os números alocados para essas memórias, podem ser aumentados ou diminuídos conforme a necessidade através a opção mostrada a baixo.



## Contacto Normalmente Aberto

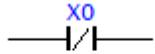
A instrução de Contacto Aberto traduz no programa o mesmo nível lógico do contacto do meio físico.



Ou seja, numa instrução de contacto aberto endereçada para um botão de pressão, é verdadeira quando o botão é pressionado.

## Contacto Normalmente Fechado

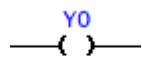
A instrução de Contacto Fechado traduz no programa o inverso do nível lógico do contacto do meio físico endereçado.



Ou seja, quando o botão estiver pressionado o contacto no programa é 0, e quando não estiver é 1, e este contacto força a lógica inversa. Este contacto é mais frequente ser usado com dispositivos normalmente fechados, como seguranças.

## Saídas

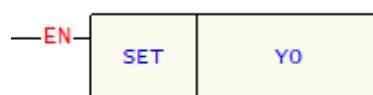
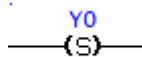
As saídas são usadas para atuadores, tem uma variável booleana associada ao endereçamento físico.



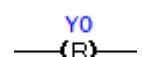
Quando o símbolo é ativado no programa, a saída é ativa fisicamente, por exemplo, se tiver um motor, este começa a trabalhar. Se estiver inativa a saída física fica também inativa e o motor desliga-se.

## Set e Reset

Os blocos lógicos de Set e Reset, podem ser representados por uma saída padrão com uma letra S (Set) ou R (Reset) no meio, ou por meio de um bloco.



Este tipo de bloco força um estado (verdadeiro, se a bobina for SET e falso se a bobina for RESET) a uma variável booleana, toda a vez que for ativa.



O estado então é mantido mesmo com a desenergização da instrução, até que uma outra instrução altere o seu valor.

### Função menor que

Esta função é uma função matemática alocada a um bloco de função, que quando a variável selecionada é menor que o valor selecionado ativa a saída.

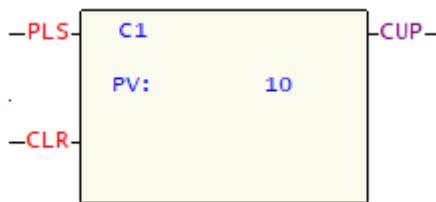
Neste caso da imagem abaixo, enquanto o contador C1 for menor que 1, a saída do bloco de função está ativa, assim que o valor aumentar a saída fica inativa.



### Contador

Os contadores são usados para incrementar ou decrementar um valor até alcançar o número pré-selecionado. No meu caso era 10, que era o número máximo de caixas até o processo parar.

O contador que eu usei incrementa a cada sinal externo, não importando o tempo, e sim a transição.



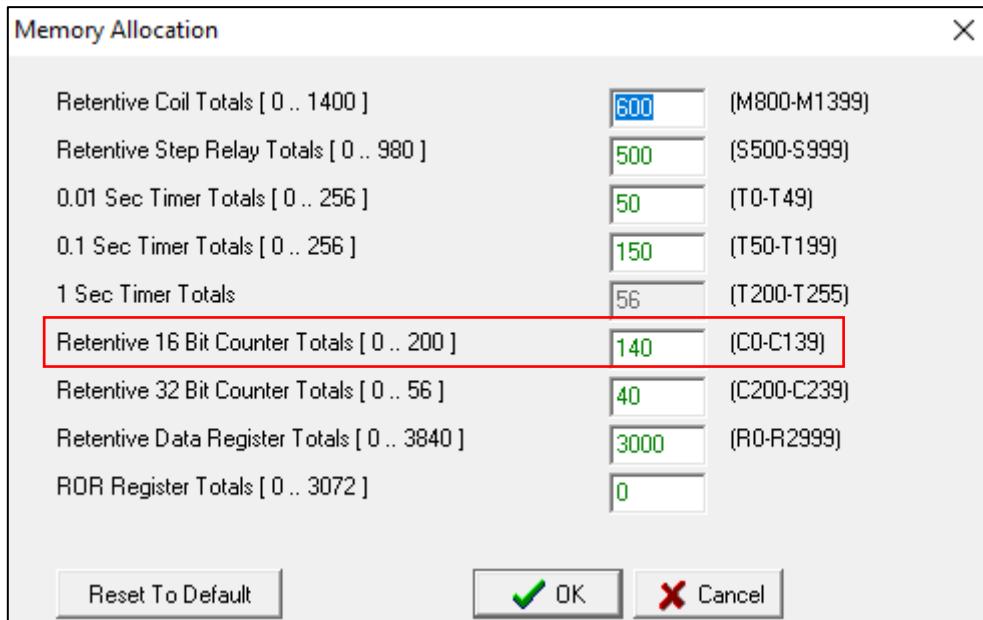
Quando o valor de 10 pulsos é alcançado a sua saída é ativada. Este contador é retentivo, ou seja, no caso da falta de “energia” o valor acumulado não é perdido.

O contador que selecionei é do tipo CTU (Contagem crescente), para aumentar o seu valor o sinal de entrada que é colocado no contacto PLS quando fizer a transição de 0 para 1 adiciona ao seu acumulador interno um valor.

E caso um pulso seja enviado para o contacto CLR o valor é resetado.

Quando chegar ao limite, conforme dito anteriormente, o contacto CUP é ativado até que o contador seja resetado.

Escolhi o C1, por necessitar apenas de uma contagem até 10. E um contador que organizasse essa informação em 16 bits ( $2^4$ ) chegaria, uma vez que 10 é igual a 0000 1010 e cabe nos 16 bits (0001 0000).



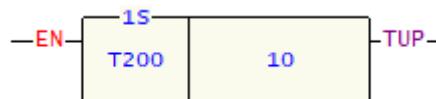
## Temporizadores

Os temporizadores são dispositivos utilizados para medir o tempo e atuar ao final do ciclo.

Na linguagem Ladder os temporizadores levam alguns argumentos para funcionarem, são eles:

- Entrada para ativação;
- Tempo a ser atingido (para ativação da saída auxiliar).

Para os temporizadores com delay na subida (TON), que foi os que eu utilizei no meu programa, no momento que houver uma entrada verdadeira, o tempo programado começará a contar.

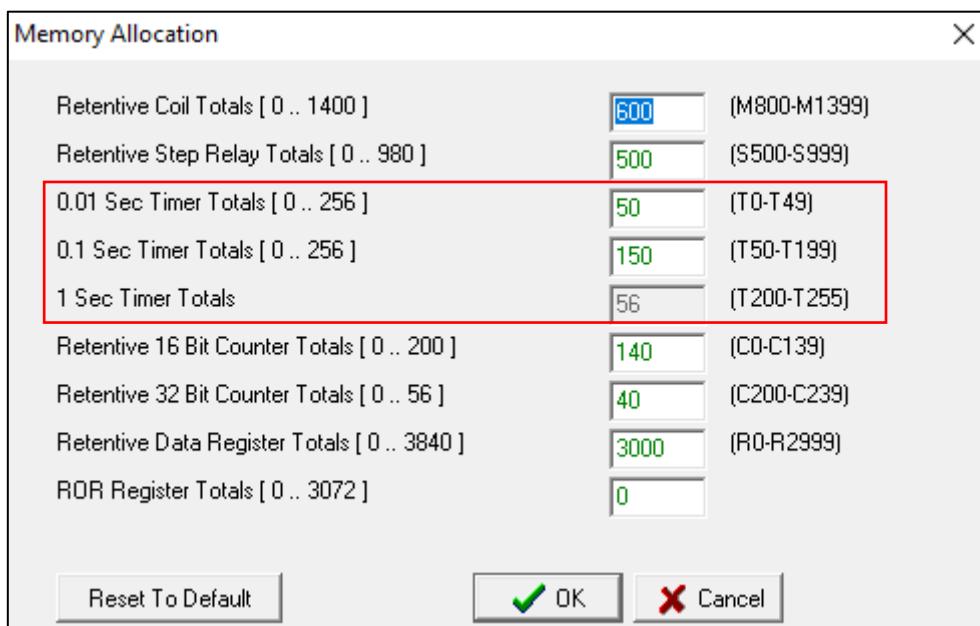


Após o tempo determinado ser atingido, a saída do temporizador será verdadeira e permanecerá nesse estado enquanto a entrada for verdadeira. E continua a contar o tempo com a saída ativa.

Quando o valor da entrada for falso o temporizador volta para o estado falso, sendo que caso outra entrada verdadeira seja aplicada o processo ocorrerá novamente.

Caso o sinal se torne falso antes de o temporizador atingir o valor máximo, a contagem é resetada e a saída permanece falsa.

Na Fatek os temporizadores organizam-se em grupos pelo endereçamento deles, esses grupos podem ser aumentados ou diminuídos conforme a necessidade com recurso ao WinProLadder.



Para o meu programa para o autómato FBs-20MC, aloquei de T0 a T49 a funcionarem em 0,01s multiplicados pelo valor de entrada selecionado. Supondo que quero dez segundos seria necessário criar o bloco de Temporizador T0 com valor 1000.

De T50 a T199 para 0,1s. Este grupo funcionaria da mesma forma, o valor de 0,1s a multiplicar pelo valor inserido na segunda variável, criariam o tempo real. Por exemplo para um timer de 10 segundos seria T50 com valor 100.

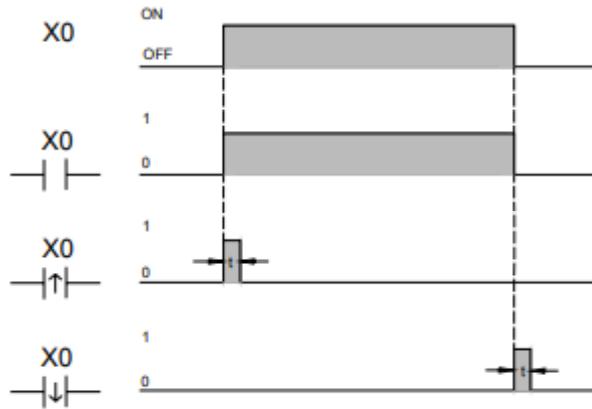
E de T200 a T255 para 1 segundo.

Eu usei dois Timers no meu programa, um para os 5 segundos e outro para os 10 segundos. Ou seja, T200 com valor 5 e T255 com valor 10.

Eu escolhi estes endereçamentos por ser mais fácil de trabalhar com eles, embora pudesse ter escolhido um que se encaixasse num dos outros grupos.

## Diferenciadas

As instruções diferenciadas são como contactos associados a endereços que quando impulsionados dão apenas um pulso que dura um tempo de ciclo ( $t$ , na imagem abaixo).

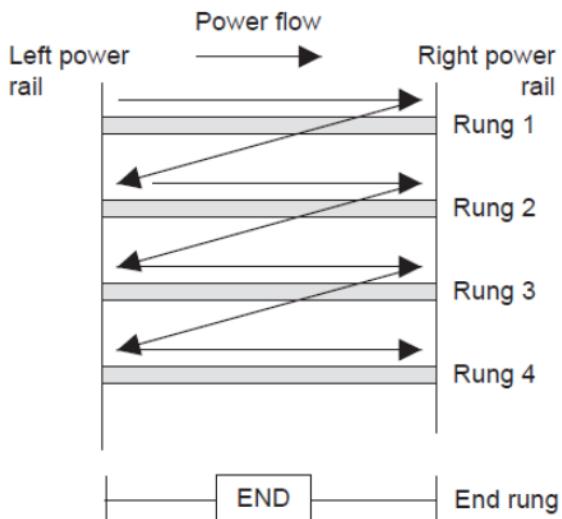


Existem diferenciais de flanco positivo, que são ativas quando detetam uma mudança de estado da memória do endereço de 0 para 1, são simbolizadas com um contacto com uma seta para cima. E existe o oposto, chamadas diferenciais de flanco negativo, que enviam um pulso quando a mudança é de 1 para 0, que são simbolizadas por um contacto com uma seta para baixo.

## Rungs

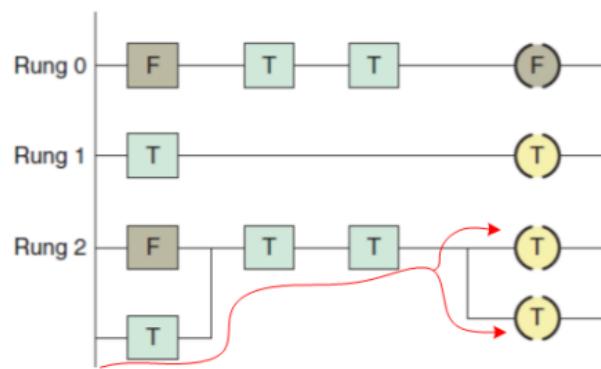
Normalmente, a execução dos programas é feita rung a rung, de cima para baixo.

Cada rung é executado da esquerda para a direita

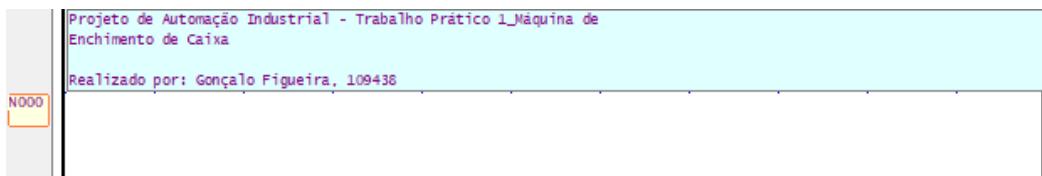


O resultado de cada contacto é determinado durante a iteração da respetiva rung.

Saídas são ativadas quando existe um caminho desde o rail esquerdo até à saída com todos os resultados lógicos “Verdadeiro”.



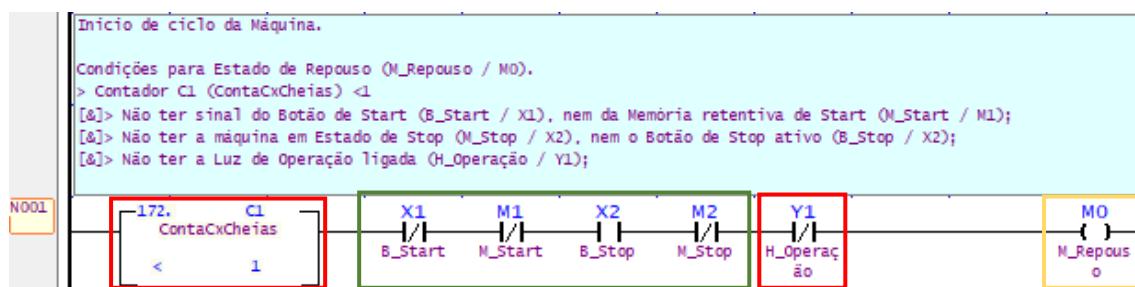
# Desenvolvimento do Projeto



Na linha N000 criei um cabeçalho do programa através de um comentário. No qual inseri o contexto do programa: Projeto de Automação Industrial – Trabalho Prático 1\_Máquina de Enchimento de Caixa” e identifiquei-me com o primeiro e último nome, e em seguida o meu número mecanográfico.

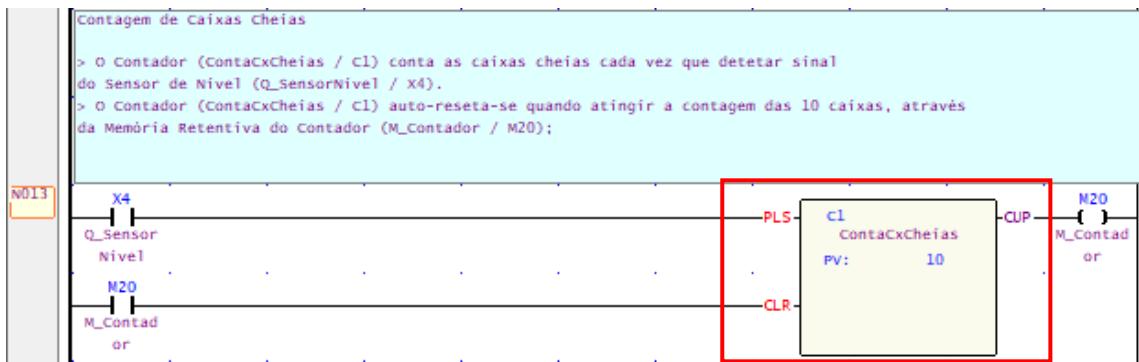
## Estado de Repouso

Criei o estado de repouso, com o objetivo de ser o estado no qual a máquina está assim que é energizada. O objetivo era ser o estado inicial, ou seja, se a máquina já tivesse estado a funcionar, e simplesmente fosse colocada em suspensão, retomaria o ciclo normal, se fosse desligada, quando fosse novamente ligada entraria neste estado de repouso.



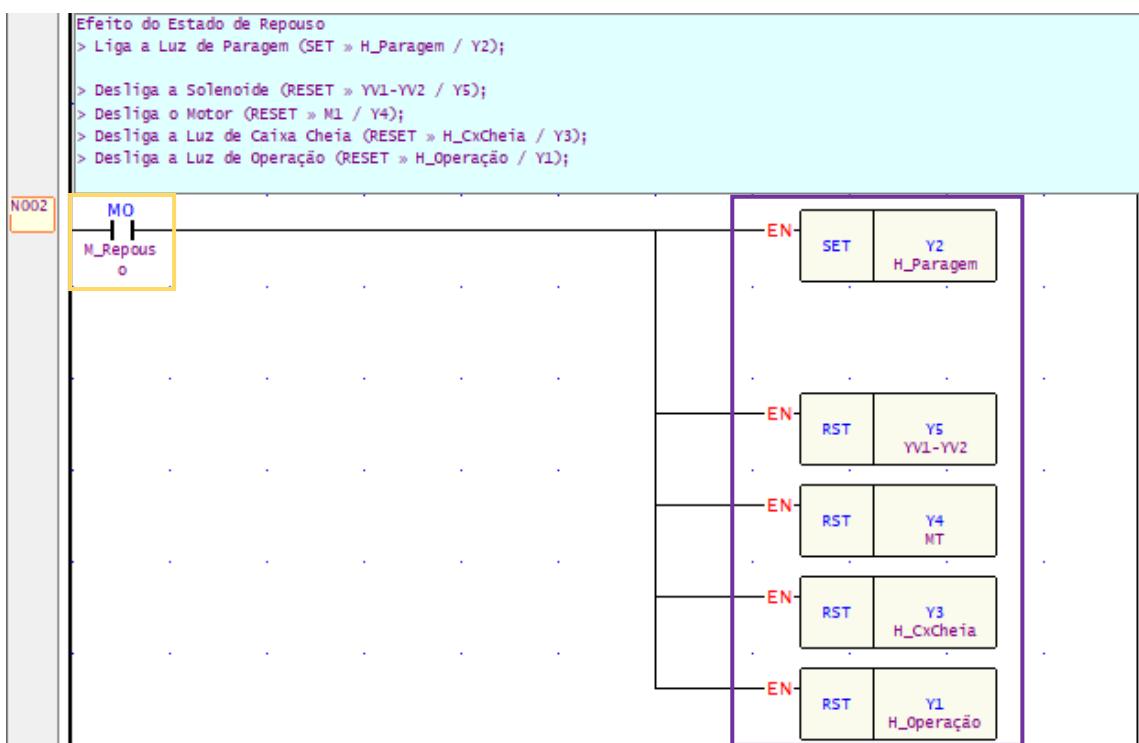
Para isso criei algumas condições, a primeira foi que o contador associado à contagem de caixas fosse menor que 1, ou seja 0 (Contador este que está na imagem abaixo a cor vermelha). E que a luz de Operação estivesse desligada.

Criando a obrigatoriedade de ter a máquina a entrar no estado de Repouso se estivesse a ser energizada nesse momento, e não apenas tirada fora de suspensão. Sinalizadas acima com a cor vermelha.



E assim que a máquina fosse posta em curso, ou seja passa-se para o Estado de Start, o estado de Repouso era desativado. Isso resultaria também se fosse colocada em estado de Paragem (Stop). Condições estas assinaladas a verde na primeira imagem.

Para isso criei uma variável do tipo retentiva, para associar as condições a esse estado, chamei-lhe “*M\_Reposo*”, e dei-lhe o endereço M0. Que está assinalada a cor amarela.



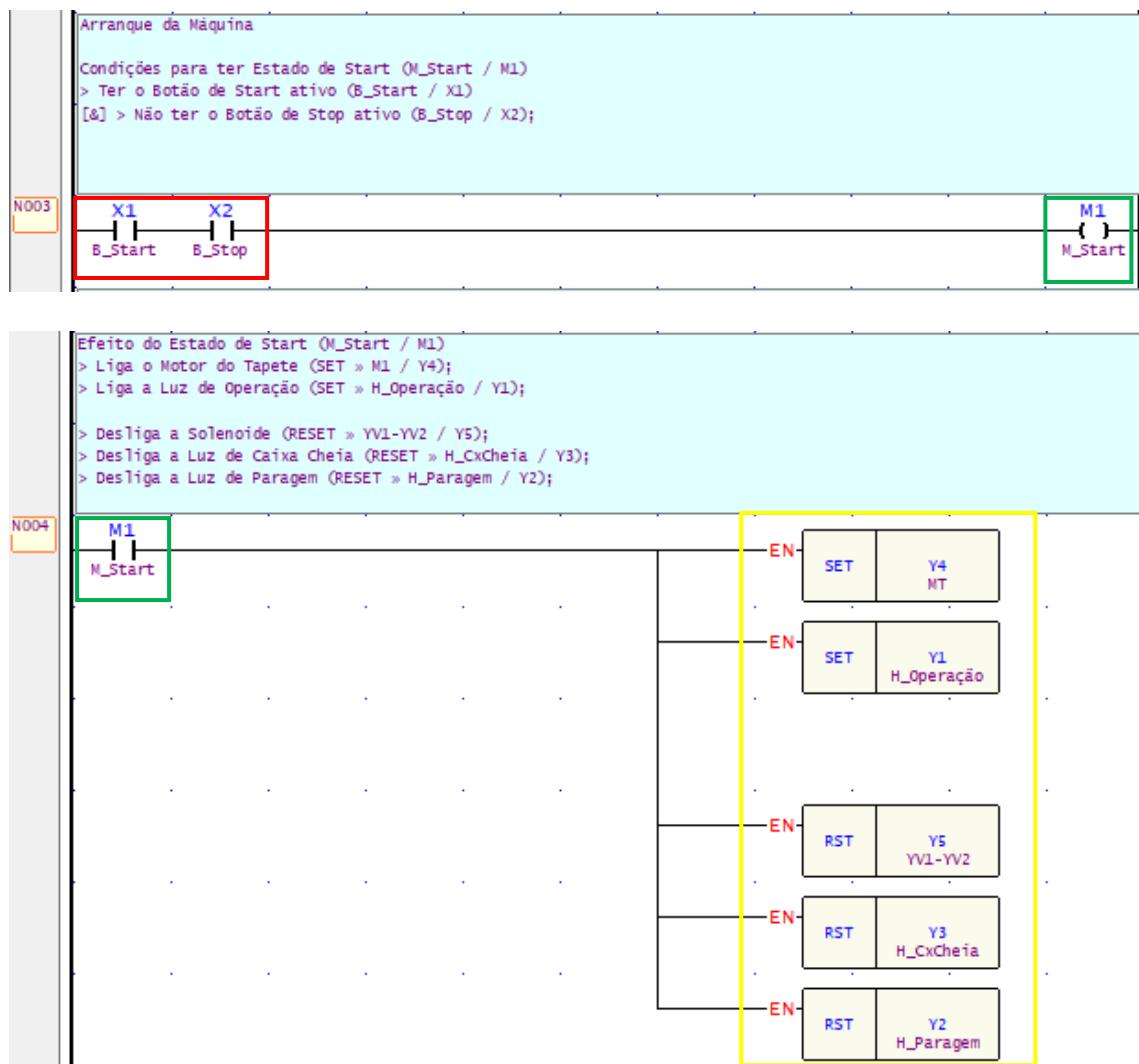
E depois criei blocos de função Set/Reset para ativar ou desativar os atuadores da máquina conforme a necessidade. Ou seja, para o estado de repouso, queria que apenas a Luz de Paragem estivesse ligada, e tudo o resto desligado. Usei um bloco de Set para a Luz de Paragem e blocos de Reset para desligar: a solenoide, o motor do tapete, a luz de caixa cheia, e a luz de operação. Isto para no caso de algum deles estar ligado, por algum motivo, ser forçado a desligar-se. Estes atuadores estão acima assinalados a roxo.

## Arranque da Máquina

Para o arranque da minha máquina criei também uma variável retentiva para guardar esse estado, que é ativada quando o Botão de Pressão de Start for pressionado, e o Botão de Pressão de Stop não. Uma vez que o botão de pressão de stop é normalmente fechado, funciona com a lógica inversa. Coloquei estes dois, ao invés de apenas o de Start para caso estes sejam pressionados ao mesmo tempo, a memória de estado não seja ativada.

A esta memória retentiva de Start dei-lhe o nome de “*M\_Start*”, e enderecei-a em M1, que está assinalada nas seguintes imagens a verde.

Para os botões de pressão Start/Stop enderecei-os em X1 e X2, consecutivamente, abaixo a vermelho.



Acima a amarelo tenho as ações a serem tomadas quando o estado de Start for ativado.

Feito por blocos de Set/Reset. Nos blocos de Set tenho a ativação do Motor do Tapete (Endereçado em Y4 e com a nomenclatura de MT). Tenho a Luz de Operação a ser ligada, endereçada em Y1, com o nome de H\_Operação.

E tenho a desativação da Luz de Paragem (Endereço Y2), a desativação da Luz de Caixa Cheia (Endereçada a Y3), e a desativação da Solenoide, endereçada a Y5.

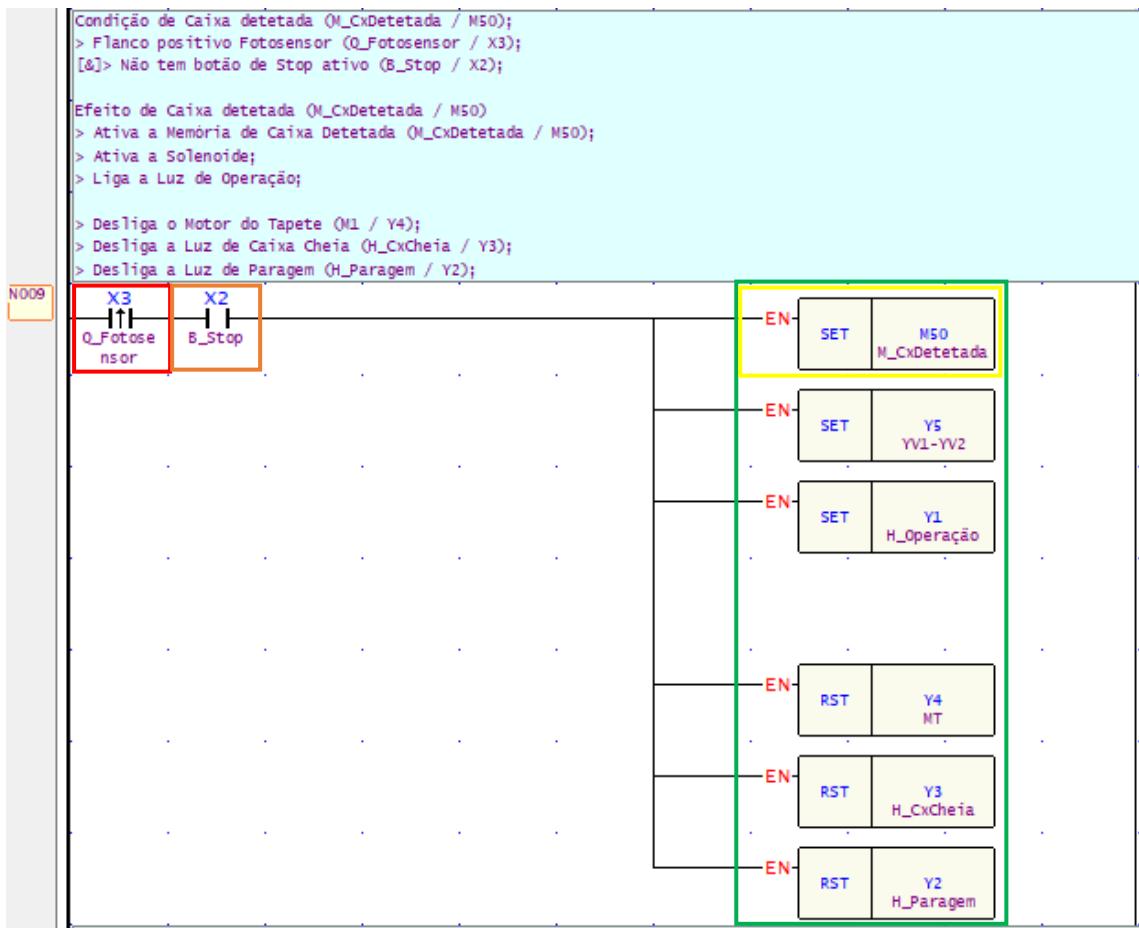
Criei este estado, apenas para o arranque da máquina. E um outro estado para quando a máquina for para voltar a entrar em funcionamento após o enchimento das caixas, documentado mais à frente neste relatório.

### **Caixa Detetada**

Quando a caixa que está no tapete e chega perto da solenoide existe um foto sensor que deteta a presença dela.

Este foto sensor está endereçado em X3, e utilizei um contacto ao flanco positivo, porque a caixa será detetada mesmo enquanto estiver a ser cheia e for para arrancar novamente com o tapete depois disso, e um contacto normal não serviria para o caso. Este está abaixo sinalizado a vermelho.

Esta condição só funciona se o Stop não estiver ativo. Serve de segurança e identificação de problemas futuros. E está abaixo a laranja.

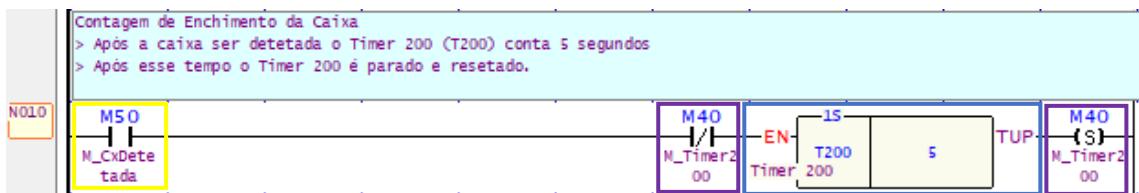


Para estas condições tenho ações a serem tomadas, que estão sinalizadas a verde, que da mesma forma que anteriormente utilizei blocos de função Set/Reset para as realizar.

Quando a caixa for detetada, o Motor do Tapete (Y4) deve parar. E a solenoide abrir Y5, para começar o enchimento da caixa.

Assim como a Luz de Operação é ligada, e a Variável M50 ativa, que serve para criar um estado de Caixa Detetada.

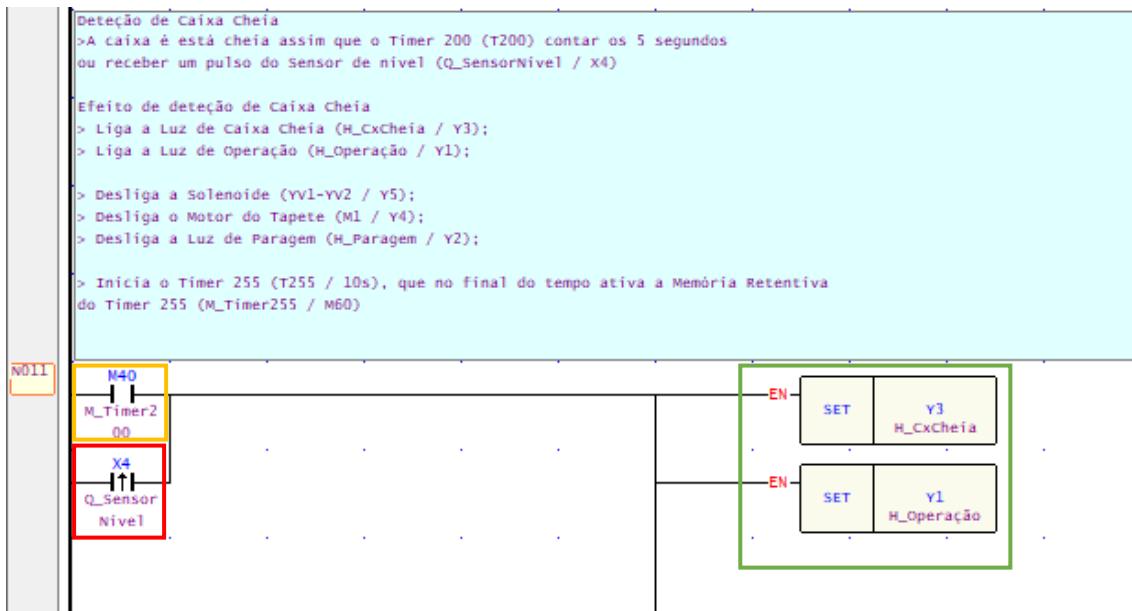
Da mesma forma que a Luz de Caixa Cheia (Y3) e a Luz de Paragem (Y2) são desligadas.



Quando a variável M50 – Caixa Detetada (A amarelo) for ativa inicia um timer, para parar de encher a caixa. Este timer é um T200 (do conjunto de 1 segundo – sinalizado a azul) que multiplicado pelo valor inserido de 5 dá 5 segundos. Este timer pára-se a ele mesmo com recurso à variável M40, que lhe chamei M\_Timer200 (Sinalizada a roxo).

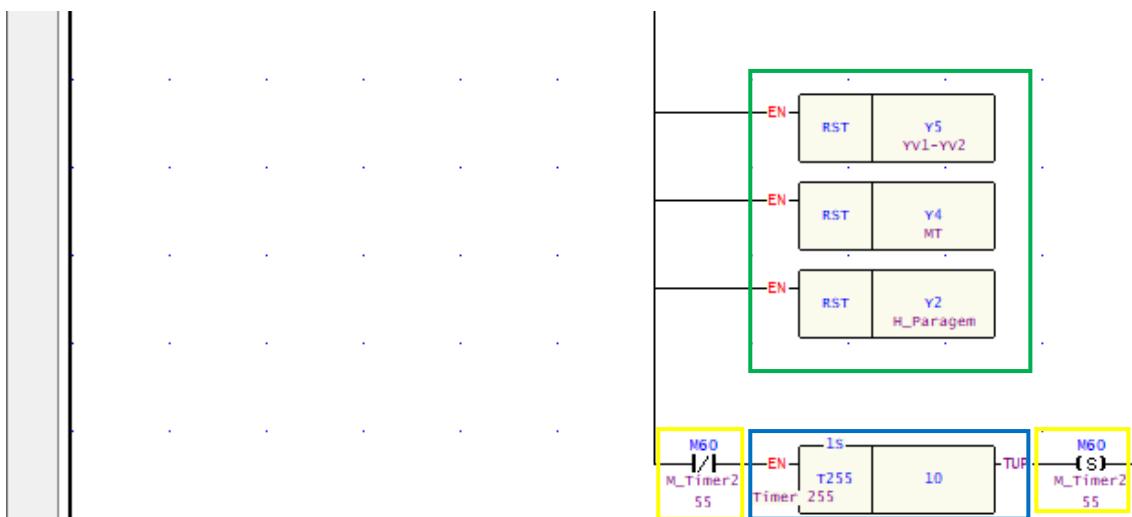
## Estado Caixa Cheia

Com a caixa parada por baixo da Solenoide e a encher. Existem duas formas detetar que ela está cheia, ou através do Sensor de Nível (A vermelho abaixo), ou através do Timer 200 que estava documentado na etapa anterior (A laranja abaixo).



Quando uma destas duas condições for detetada algumas ações são tomadas através de blocos de Set/Reset.

Como a ativação da Luz de Caixa Cheia (Utilizei esta luz como sinalização do processo de enchimento de caixa concluído, mesmo que exista falta de líquido no funil e a caixa não seja realmente cheia). A ativação da Luz de Operação, ambas acima assinaladas a verde.

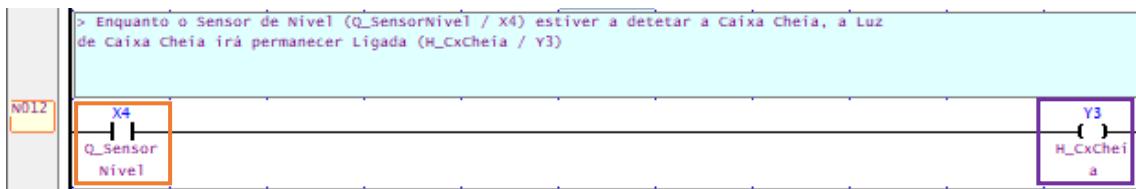


Quando o processo de enchimento das caixas terminar deve ser também fechada a solenoide para parar de encher a caixa. Isto é feito com um Reset ao endereço da Solenoide Y5 como pode ser visto acima a verde.

O motor do tapete e a luz de paragem embora já estivessem desligados são novamente forçados a estarem desligados neste momento através de um bloco de Reset, também assinalado a verde acima.

E ao mesmo tempo que estes processos, um Timer de 10 segundos é iniciado. O Timer faz parte do mesmo conjunto dos anteriores (1 segundo), e é o Timer 255, com um valor de 10, destacado a azul acima.

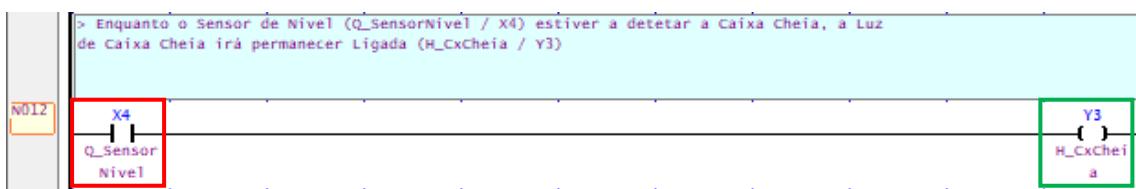
Este reinicia-se a ele mesmo através da variável M60 – M\_Timer255, acima assinalada a amarelo.



De forma a garantir que a Luz de Caixa Cheia permanece ligada mesmo após a caixa avançar novamente no tapete, e enquanto estiver a ser detetada pelo sensor de nível criei esta linha que me garante isso. A laranja está o sensor de nível e a roxo a luz de Caixa Cheia.

### Caixa Não Cheia

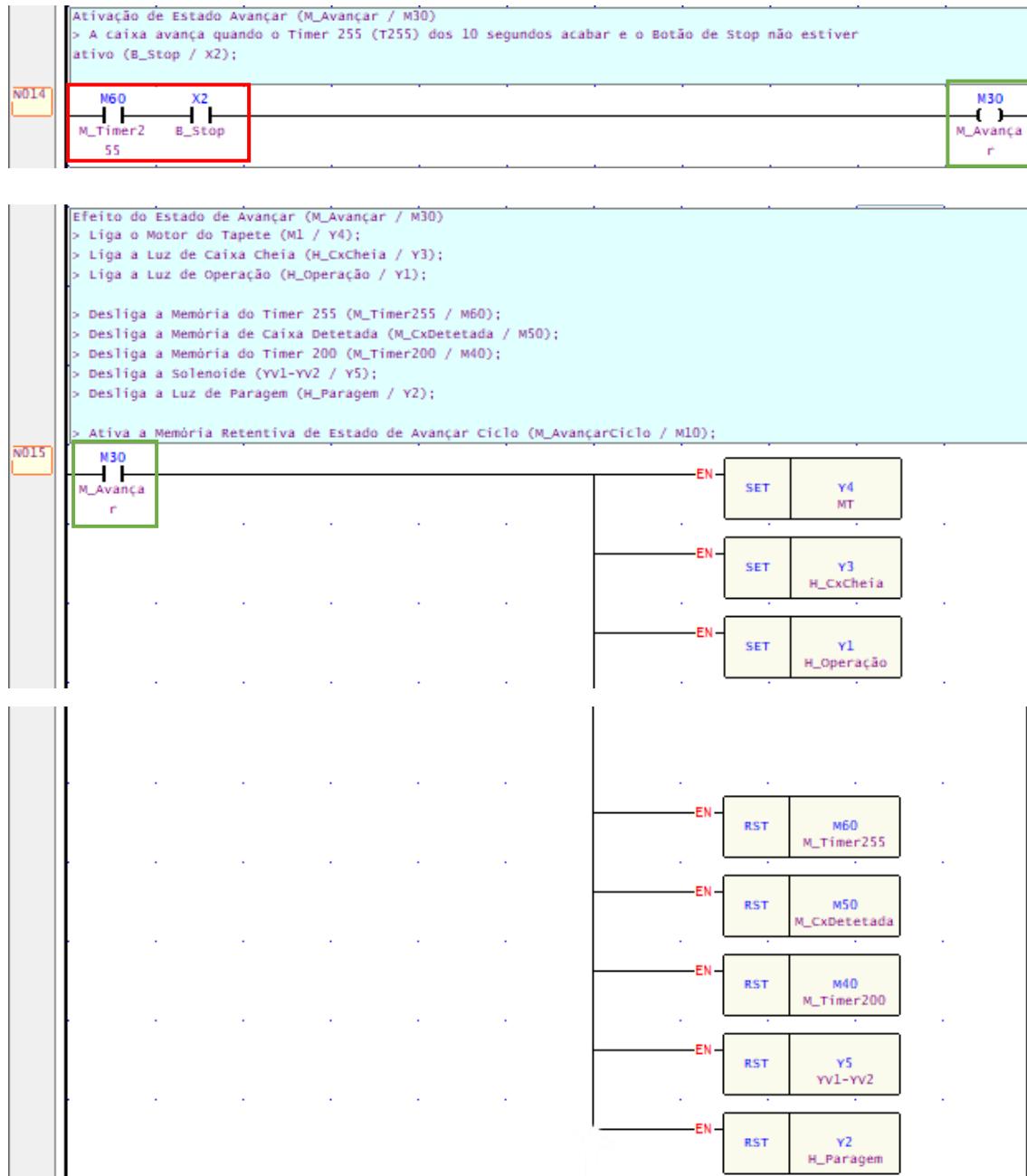
Quando a caixa não for cheia, por falta de líquido, o sensor de nível (X4 - Vermelho) a Luz de Caixa Cheia (Y3 - Verde) não irá permanecer ligada porque não vai estar a detetar nível.



### Loop de ciclo

Este estado (Sinalizado abaixo a verde) serve para quando o ciclo de enchimento de caixas está concluído, ou seja, só será ativado após o Temporizador 255 fazer a sua contagem.

Como é óbvio este estado não é possível se o stop estiver ativo, conforme pode ser visto abaixo a vermelho.



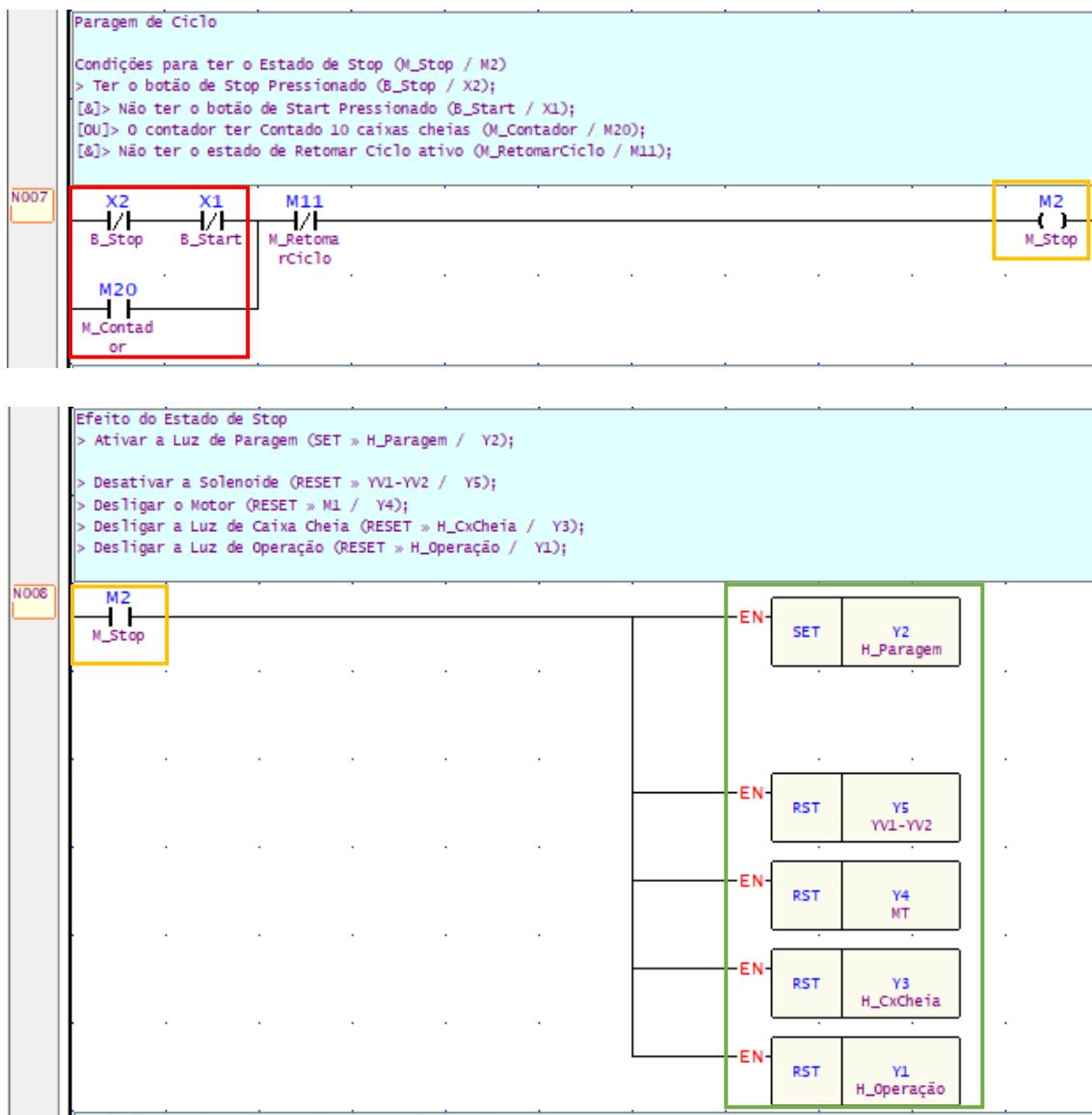
Na ativação deste estado são tomadas algumas ações. Como a ativação do Motor do Tapete (Y4), a ativação da Luz de Caixa Cheia (Y3) (Que utilizei novamente para simbolizar o estado atual da máquina), e a Luz de Operação (Y1).

Da mesma forma que a posição da Memória Retentiva associada ao Temporizador 255, a da Memória Retentiva de Estado de Caixa Detetada, a da Memória do Timer 200, a da Solenoide e a da Luz de Paragem são colocadas a 0, de forma a garantir que tudo está conforme deve estar nesta etapa de curso da máquina.

## Paragem de Ciclo - Stop

Este estado serve para parar a máquina por completo. É suposto todos os processos serem parados e a Luz de Paragem ficar ativa.

Para isso coloquei o botão de Stop a fazer a ativação do estado (Junto com o de Start para evitar a situação de ter os dois pressionados realizar a ativação deste estado) ou a Memória do Contador que ao fim das 10 caixas para a máquina, conforme está assinalado a vermelho abaixo.

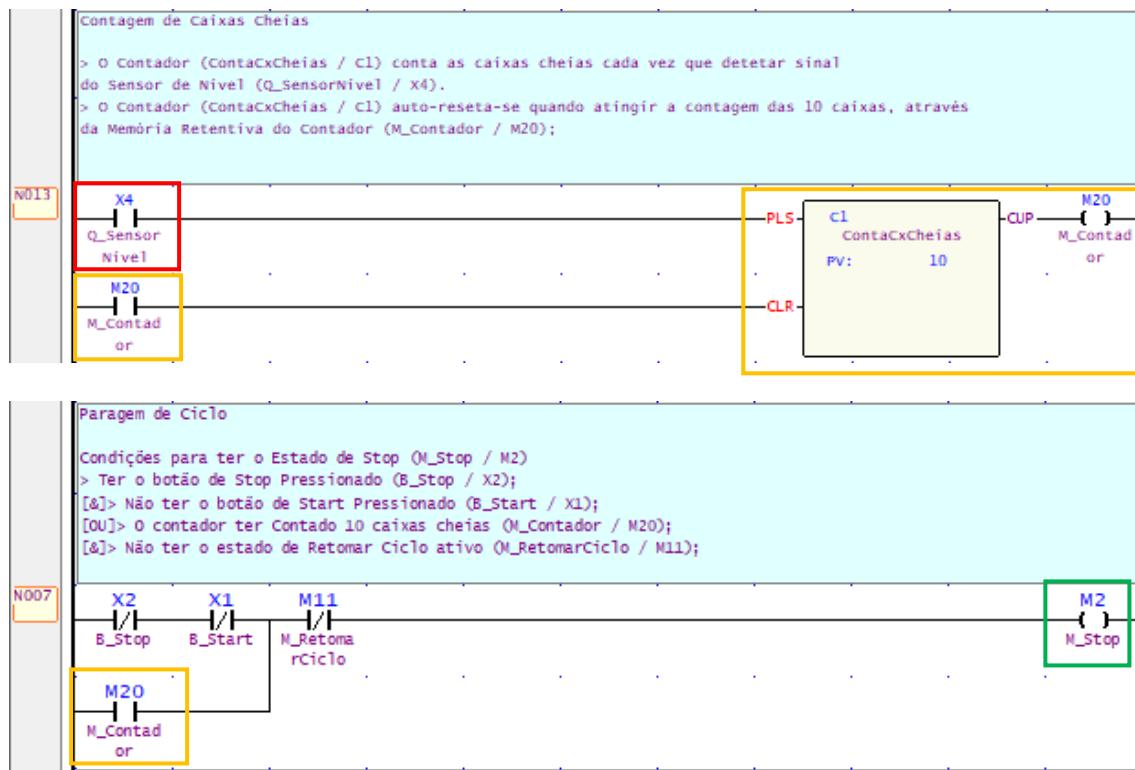


A paragem da máquina ativa então a Luz de Paragem e desativa/desliga a Solenoide, o Motor do Tapete, a Luz de Caixa Cheia e a Luz de Operação conforme pode ser visto acima a verde, colocando os processos que decorrem na máquina totalmente em suspensão.

## Limite de Caixas Cheias

Este bloco foi criado para fazer a contagem das caixas, o objetivo é parar o processo ao fim da contagem das 10 caixas. A contagem é feita pelo Sensor de Nível, uma vez que só me interessa contar as caixas cheias e desprezar as caixas não cheias (Abaixo a vermelho).

O contador auto reseta-se (Como visto abaixo a amarelo) no final da contagem e é usado para parar a máquina através do estado de Stop.



# Testes realizados

A documentação dos resultados dos testes é um passo crucial na avaliação de uma máquina industrial.

Os testes permitiram analisar os resultados com mais detalhes, identifique quaisquer problemas ou falhas e tomar medidas para corrigi-los.

Além disso, uma frequente documentação dos resultados dos testes é importante para manter um registo para futuras intervenções.

Abaixo está a página de endereços que utilizei para realizar os testes. Entre esses endereços estão as entradas, as saídas, as posições de memória associadas a estados das máquinas, o estado dos Temporizadores e do Contador.

Status Monitoring											
Comment	Status	Data	Comment	Status	Data	Comment	Status	Data	Comment	Status	Data
B_Start	Enable	OFF	M_ReposEnable	OFF							
B_Stop	Enable	OFF	M_Start	Enable	OFF						
Q_Fotosensor	Enable	OFF	M_Stop	Enable	OFF						
Q_SensorNivel	Enable	OFF									
			M_Avanc	Enable	OFF						
			M_Retom	Enable	OFF						
H_Operação	Enable	OFF	M_Contac	Enable	OFF						
H_Paragem	Enable	OFF	M_Avanc	Enable	OFF						
H_CxCheia	Enable	OFF	M_Timer2	Enable	OFF						
MT	Enable	OFF	M_CxDet	Enable	OFF						
YY1-YY2	Enable	OFF	M_Timer2	Enable	OFF						
Timer 200	OFF	0									
Timer 255	OFF	0									
ContaCxCheias	OFF	0									

## Teste de Funcionamento

1. Ao ligar a máquina apenas o botão de Stop ficará ativo, uma vez que é um botão normalmente fechado, a posição de memória associada a essa entrada permanece ativa. E a máquina está no estado de repouso.

Comment	Status	Data	Comment	Status	Data
B_Start	Enable	OFF	M_RepouEnable	ON	
B_Stop	Enable	ON	M_Start	Enable	OFF
Q_Fotosensor	Enable	OFF	M_Stop	Enable	OFF
Q_SensorNivel	Enable	OFF			
			M_Avanç Enable	OFF	
			M_Retom Enable	OFF	
H_Operation	Enable	OFF	M_Contac Enable	OFF	
H_Paragem	Enable	ON	M_Avanç Enable	OFF	
H_CxCheia	Enable	OFF	M_Timer2 Enable	OFF	
MT	Enable	OFF	M_CxDet Enable	OFF	
YV1-YV2	Enable	OFF	M_Timer2 Enable	OFF	
Timer 200	OFF	0			
Timer 255	OFF	0			
ContaCxCheias	OFF	0			

2. O arranque da máquina é feito com a ativação do botão de Start. Nesse momento o motor liga e a luz de operação liga, e a máquina está em estado de funcionamento.

Comment	Status	Data
B_Start	Enable	OFF
B_Stop	Enable	ON
Q_Fotosensor	Enable	OFF
Q_SensorNivel	Enable	OFF
H_Operation	Enable	ON
H_Paragem	Enable	OFF
H_CxCheia	Enable	OFF
MT	Enable	ON
YV1-YV2	Enable	OFF

3. Depois é detetado sinal do foto-sensor, que significa que foi detetada uma caixa na zona de enchimento de caixas. Nesse momento começa a contagem do temporizador 200 de 5 segundos para a seguinte etapa, é possível também avançar para o próximo passo se for detetado sinal do sensor de nível. Nesse momento

estará também a acontecer o enchimento de uma caixa, que é o possivelmente fará o sensor de nível ativar.

Comment	Status	Data	Comment	Status	Data
B_Start	Enable	OFF	B_Start	Enable	OFF
B_Stop	Enable	ON	B_Stop	Enable	ON
Q_Fotosensor	Enable	ON	Q_Fotosensor	Enable	ON
Q_SensorNivel	Enable	OFF	Q_SensorNivel	Enable	ON
H_Operação	Enable	ON	H_Operação	Enable	ON
H_Paragem	Enable	OFF	H_Paragem	Enable	OFF
H_CxCheia	Enable	OFF	H_CxCheia	Enable	ON
MT	Enable	OFF	MT	Enable	OFF
YV1-YV2	Enable	ON	YV1-YV2	Enable	OFF
Timer 200	OFF	4	Timer 200	OFF	0
Timer 255	OFF	0	Timer 255	OFF	4

4. Assim que um dos sinais é dado arranca o temporizador dos 10 segundos. Caso seja o sinal do temporizador a caixa não é contabilizada, caso seja o sinal do sensor de nível a caixa entra para a contagem de caixas cheias.

Comment	Status	Data
B_Start	Enable	OFF
B_Stop	Enable	ON
Q_Fotosensor	Enable	ON
Q_SensorNivel	Enable	ON
H_Operação	Enable	ON
H_Paragem	Enable	OFF
H_CxCheia	Enable	ON
MT	Enable	OFF
YV1-YV2	Enable	OFF
Timer 200	OFF	0
Timer 255	OFF	4
ContaCxCheias	OFF	3

5. Assim que se passarem os 10 segundos o tapete arranca novamente. E a luz de caixa cheia é desligada uma vez que a caixa deixa de ser detetada pelo sensor de nível (caso esta tenha sido cheia).

Comment	Status	Data
B_Start	Enable	OFF
B_Stop	Enable	ON
Q_Fotosensor	Enable	ON
Q_SensorNivel	Enable	OFF
H_Operação	Enable	ON
H_Paragem	Enable	OFF
H_CxCheia	Enable	OFF
MT	Enable	ON
YV1-YV2	Enable	OFF
Timer 200	OFF	0
Timer 255	OFF	0

6. Quando a caixa é contabilizada, no contador é incrementado um valor.

Comment	Status	Data
B_Start	Enable	OFF
B_Stop	Enable	ON
Q_Fotosensor	Enable	ON
Q_SensorNivel	Enable	ON
H_Operação	Enable	ON
H_Paragem	Enable	OFF
H_CxCheia	Enable	ON
MT	Enable	ON
YV1-YV2	Enable	OFF
Timer 200	OFF	0
Timer 255	OFF	0
ContaCxCheias	OFF	1

7. Após isso o processo retorna ao normal funcionamento pronto para que uma nova caixa chegue à zona de enchimento.

Comment	Status	Data
B_Start	Enable	OFF
B_Stop	Enable	ON
Q_Fotosensor	Enable	OFF
Q_SensorNivel	Enable	OFF
H_Operation	Enable	ON
H_Paragem	Enable	OFF
H_CxCheia	Enable	OFF
MT	Enable	ON
YV1-YV2	Enable	OFF
Timer 200	OFF	0
Timer 255	OFF	0

8. Quando são contabilizadas as 10 caixas o processo entra em paragem (Stop), a contagem é resetada e o processo fica suspenso até nova ordem de início, do operador.

Comment	Status	Data	Comment	Status	Data
B_Start	Enable	OFF	B_Start	Enable	OFF
B_Stop	Enable	ON	B_Stop	Enable	ON
Q_Fotosensor	Enable	ON	Q_Fotosensor	Enable	OFF
Q_SensorNivel	Enable	OFF	Q_SensorNivel	Enable	OFF
H_Operation	Enable	ON	H_Operation	Enable	OFF
H_Paragem	Enable	OFF	H_Paragem	Enable	ON
H_CxCheia	Enable	OFF	H_CxCheia	Enable	OFF
MT	Enable	OFF	MT	Enable	OFF
YV1-YV2	Enable	OFF	YV1-YV2	Enable	OFF
Timer 200	OFF	0	Timer 200	OFF	0
Timer 255	OFF	7	Timer 255	OFF	0
ContaCxCheias	OFF	9			

### Teste de Pressionar Start e Stop ao Mesmo tempo

Ao pressionar ambos os botões ao mesmo tempo, nenhuma ação é tomada. Todos os processos são parados e não existe nenhum conflito de ações na máquina.

Comment	Status	Data	Comment	Status	Data
B_Start	Enable	ON	M_Reparou	Enable	OFF
B_Stop	Enable	OFF	M_Start	Enable	OFF
Q_Fotosensor	Enable	OFF	M_Stop	Enable	OFF
Q_SensorNivel	Enable	OFF			
			M_Avanç	Enable	OFF
			M_Retom	Enable	OFF
H_Operação	Enable	OFF	M_Contac	Enable	OFF
H_Paragem	Enable	OFF	M_Avanç	Enable	OFF
H_CxCheia	Enable	OFF	M_Timer2	Enable	OFF
MT	Enable	OFF	M_CxDetr	Enable	OFF
YV1-YV2	Enable	OFF	M_Timer2	Enable	OFF
Timer 200	OFF	0			
Timer 255	OFF	0			
ContaCxCheias	OFF	0			

# Conclusão

A programação de controladores lógicos programáveis (PLCs) em linguagem Ladder é uma técnica importante na automação industrial.

Este trabalho foi uma oportunidade valiosa para aprender sobre este tipo de programação e compreender as suas vantagens e aplicações direcionadas ao mercado de trabalho.

Algumas das principais vantagens de ter realizado este trabalho incluem:

- O conhecimento aprofundado sobre a linguagem Ladder: o trabalho permitiu compreender a sintaxe e as estruturas de controle utilizadas na linguagem Ladder, bem como as suas aplicações na programação de PLCs (Uma vez que tive que aprender os dois sistemas: Fatek e Siemens).
- Habilidades práticas na programação de PLCs e estímulo de raciocínio lógico.
- Conhecimento dos diferentes programas e metodologias de trabalho em autómatos programáveis.
- Ampliação do conhecimento sobre a automação industrial uma vez que serviu para fazer uma pesquisa geral dos componentes que consistiam o meu sistema.

Embora tenha encarado o desafio entrando logo na programação, acredito que com a elaboração de um gráfet ou de um fluxograma teria perdido muito menos tempo à procura de soluções e a resolver problemas e obstáculos encontrados no decorrer do trabalho.

Descobri também com o uso do programa algumas coisas que me poderão ser úteis no futuro, e que pretendo explorar mais: como a possibilidade de construir o programa, em subprogramas (que acredito que sejam pequenas rotinas), descobri que é possível inserir passwords no meu programa.

Mas, em resumo, este trabalho foi vantajoso não só em prol da disciplina, mas também para me preparar para o futuro mercado que irei enfrentar que é rico cada vez mais de tecnologia PLC.