

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
**DEPARTAMENTO ACADÊMICO DE ELETRÔNICA**  
**CURSO SUPERIOR DE TECNOLOGIA EM AUTOMAÇÃO INDUSTRIAL**

**BRUNO CORADASSI**

**IMPLEMENTAÇÃO DE IHM *MOBILE* PARA PLANTA DIDÁTICA**

**TRABALHO DE CONCLUSÃO DE CURSO**

**PONTA GROSSA**

**2018**

**BRUNO CORADASSI**

**IMPLEMENTAÇÃO DE IHM *MOBILE* PARA PLANTA DIDÁTICA**

Trabalho de Conclusão de Curso apresentado ao Departamento Acadêmico de Eletrônica do Campus Ponta Grossa da Universidade Tecnológica Federal do Paraná para obtenção de grau de Tecnólogo no Curso Superior de Tecnologia em Automação Industrial.

Orientador: Prof. Me. José Jair Alves Mendes Jr.

Coorientador: Prof. Dr. Felipe Mezzadri

**PONTA GROSSA**

**2018**



---

**TERMO DE APROVAÇÃO**

**IMPLEMENTAÇÃO DE IHM *MOBILE* PARA PLANTA DIDÁTICA**

por

**BRUNO CORADASSI**

Este Trabalho de Conclusão de Curso foi apresentado em 02 de Julho de 2018 como requisito parcial para a obtenção do título de Tecnólogo em Automação Industrial. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Me. José Jair Alves Mendes Junior.  
Prof. Orientador

---

Me. Edison Luiz Salgado Silva  
Membro titular

---

Esp. Rafael Schmidt Baumel  
Membro titular

A Deus, que nos criou e foi criativo nesta tarefa. Seu fôlego de vida em mim me foi sustento e me deu coragem para questionar realidades e propor sempre um novo mundo de possibilidades.

## **AGRADECIMENTOS**

A Deus por ter me dado saúde e força para superar as dificuldades.

A esta Universidade, seu corpo docente, direção e administração que oportunizaram a janela em que hoje vislumbro um horizonte superior, eivado pela acendrada confiança no mérito e ética aqui presentes.

Aos meu orientadores Prof. Me. José Jair Alves Mendes Jr. e Prof. Dr. Felipe Mezzadri, pelo suporte no pouco tempo que lhes coube, pelas suas correções e incentivos.

Aos meus pais, pelo amor, incentivo e apoio incondicional.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

“A maior recompensa para o trabalho do homem não é o que ele ganha com isso, mas o que ele se torna com isso.” John Ruskin.

## RESUMO

CORADASSI, Bruno. **Implementação de IHM *Mobile* para Planta Didática**. 2018. 72 f Trabalho de Conclusão de Curso Superior de Tecnologia em Automação Industrial – Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2018.

Este projeto consiste da elaboração, de uma metodologia para implementação, de uma interface IHM, em uma planta didática. Como supervisório foi utilizado o LabVIEW™. Como IHM o *app Data DashBoard for LabVIEW™*. Para realizar a implementação, alguns conceitos foram abordados. Dentre estes conceitos, a Automação Industrial, e sua constante evolução. A necessidade de uma IHM. Programas supervisórios, o que precisam. Comunicação entre servidores e clientes. Interface MPI. Utilizar um tablet como IHM.

**Palavras-chave:** LabVIEW™. Supervisório. IHM. *Mobile. Data DashBoard for LabVIEW™*. Automação Industrial.

## ABSTRACT

CORADASSI, Bruno. **Implementation of Mobile HMI for Didactic Plant**. 2018. 72 f Conclusion Work of Higher Course of Technology in Industrial Automation - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2018.

This project consists of the elaboration, of a methodology for implementation, of an HMI interface, in a didactic plan. As a supervisor, LabVIEW™ was used. How to HMI the Data DashBoard app for LabVIEW™. To implement the implementation, some concepts were addressed. Among these concepts, Industrial Automation, and its constant evolution. The need for an HMI. Supervisory programs, what they need. Communication between servers and clients. MPI interface. Use a tablet like HMI.

**Keywords:** LabVIEW™. Supervisory. HMI. mobile Data DashBoard for LabVIEW™. Industrial automation.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de blocos sobre Processos de Controle. ....	13
Figura 2 – Direção da Comunicação entre Planta, Servidor, Cliente e Supervisório. ....	18
Figura 3 – Fluxograma da Metodologia empregada no projeto. ....	20
Figura 4 – Planta Didática de Processos da Manufatura com as suas seis divisões. ....	22
Figura 5 – Esquemático do CLP usado no projeto. ....	23
Figura 6 – Comunicação CLP-PC. ....	24
Figura 7 – Seleção da Interface de comunicação no STEP7. ....	24
Figura 8 – <i>Project Wizard</i> do software STEP7. ....	25
Figura 9 – Tela Inicial pós Configuração do STEP7. ....	26
Figura 10 – Servidor OPC do LabVIEW™ (NI OPC Server). ....	27
Figura 11 – Seleção do <i>Device Driver</i> do NI OPC Server. ....	27
Figura 12 – Configurações do canal de comunicação do Servidor. ....	28
Figura 13 – <i>Local Station Address</i> do NI OPC Server. ....	29
Figura 14 – <i>Device ID</i> do NI OPC Server. ....	30
Figura 15 – Tela de Definição das Tags do NI OPC Server. ....	30
Figura 16 – Teste das <i>Tags</i> do NI OPC Server por meio do cliente do próprio servidor. ....	31
Figura 17 – <i>Project Explorer</i> do software LabVIEW™, em que a) mostra a seleção do I/O Server e b) a seleção do OPC Client. ....	31
Figura 18 – Janela de seleção do servidor OPC do LabVIEW™. ....	32
Figura 19 – Variáveis Compartilhadas no <i>Project Explorer</i> do LabVIEW™. ....	33
Figura 20 – Selecionando as Variáveis Compartilhadas no <i>Project Explorer</i> do LabVIEW™, onde a) mostra as variáveis selecionadas, b) o botão para adicionar as variáveis que foram selecionadas, e c) apertar OK. ....	33
Figura 21 – Programação da tela do supervisor no LabVIEW™. ....	35
Figura 22 – Janela do CMD. ....	36
Figura 23 – Espaço da Variável no Aplicativo <i>Data Dashboard for LabVIEW™</i> . ....	36
Figura 24 – Opções da seleção de variável do aplicativo <i>Data Dashboard for LabVIEW™</i> . ....	37
Figura 25 – Inserindo o Endereço IPv4 no Aplicativo <i>Data Dashboard for LabVIEW™</i> . ....	37
Figura 26 Seleção da biblioteca no app <i>Data Dashboard for LabVIEW™</i> . ....	38
Figura 27 – Seleção da variável no app <i>Data Dashboard for LabVIEW™</i> . ....	38
Figura 28 – Opções do Indicador <i>Boolean</i> . ....	39
Figura 29 – Tela do Supervisor LabVIEW™ e IHM do <i>Data Dashboard for LabVIEW™</i> . ....	41
Figura 30 – IHM no app <i>Data Dashboard for LabVIEW™</i> . ....	41

## LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

### LISTA DE ABREVIATURAS

a.C.	Antes de Cristo
mA	Miliampère
VDC	Tensão em corrente continua

### LISTA DE SIGLAS

CLP	Controlador Lógico Programável
UTFPR	Universidade Tecnológica Federal do Paraná
IHM ou HMI	Interface Homem Máquina ( <i>Human Machine Interface</i> )
GM	<i>General Motors</i>
I/O	Entrada / Saída (Input/Output)
IEC	Comissão de Eletrotécnica Internacional ( <i>International Electrotechnical Commission</i> )
MPI	Interface de passagem de mensagem ( <i>Message Passing Interface</i> )
OPC	OLE para Controle de Processos ( <i>OLE for Process Control</i> )
OLE	Vinculação e Incorporação de Objetos ( <i>Object Linking and Embedding</i> )
NI	<i>National Instruments</i>
LCD	Display de Cristal Líquido ( <i>Liquid Crystal Display</i> )
OIT	Terminal de Interface do Operador ( <i>Operator Interface Terminal</i> )
IP	Protocolo da Internet ( <i>Internet Protocol</i> )

### LISTA DE ACRÔNIMOS

MODICON	Controlador Digital Modular
GRAFSET	Gráfico de Controle Ação, Etapas e Transições
LED	Diodo Emissor de Luz
SCADA	Controle Supervisório e Aquisição de Dados
LabVIEW™	Bancada de Engenharia de Instrumentação Virtual de Laboratório ( <i>Laboratory Virtual Instrument Engineering Workbench</i> )

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>12</b>
1.1 TEMA E DELIMITAÇÃO DO TEMA .....	13
1.2 OBJETIVO .....	14
1.2.1 Objetivo Geral .....	14
1.2.2 Objetivos Específicos.....	14
1.3 JUSTIFICATIVA.....	14
<b>2 REFERENCIAL TEÓRICO .....</b>	<b>16</b>
<b>3 METODOLOGIA.....</b>	<b>20</b>
<b>4 DESENVOLVIMENTO.....</b>	<b>21</b>
4.1 SELEÇÃO DA PLANTA .....	21
4.2 COMUNICAÇÃO CLP-PC.....	22
4.3 DESENVOLVIMENTO DO GRAFCET E O PROGRAMA EM LADDER .....	26
4.4 COMUNICAÇÃO ENTRE PLANTA, SERVIDOR E CLIENTE.....	27
<b>5 RESULTADOS .....</b>	<b>40</b>
<b>6 CONSIDERAÇÕES FINAIS .....</b>	<b>42</b>
<b>REFERÊNCIAS.....</b>	<b>43</b>
<b>APÊNDICE A - GRAFCET do programa que foi criado no STEP7.....</b>	<b>45</b>
<b>APÊNDICE B - Programação em Ladder do software SIMATIC STEP7.....</b>	<b>47</b>
<b>APÊNDICE C - <i>Symbol Table</i> do programa do STEP7.....</b>	<b>69</b>

## 1 INTRODUÇÃO

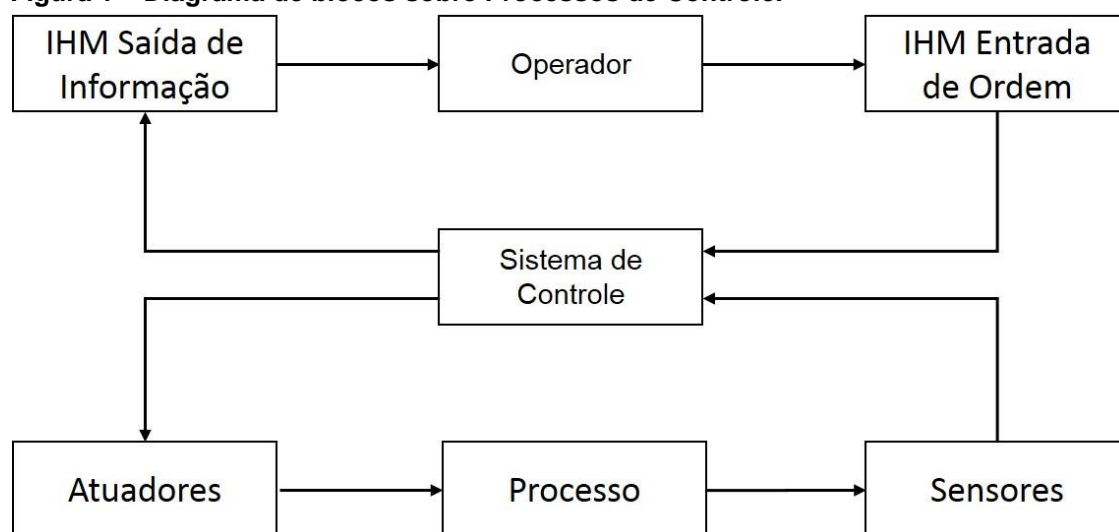
Há muito tempo, as pessoas buscam automatizar as suas tarefas diárias para poupar tempo e esforço. Este pensamento é visível desde a Antiguidade, como exemplo, pode-se citar os registros de 644 a.C. em que moinhos de vento foram utilizados para moer grãos e bombear água para irrigação na China e Pérsia. Por sua vez, no século XIII, surgiram na Europa as primeiras proposta de automação nos ramos das indústrias metalúrgicas, têxteis e madeireiras (FARIAS; SELLITTO, 2011).

No início da industrialização, a grande maioria dos trabalhos oferecidos eram trabalhos braçais e repetitivos, muitas vezes perigosos, os quais forneciam grandes riscos à saúde e ao bem estar do trabalhador. Estes riscos podiam ser imediatos (como acidentes diretos) ou danos a longo prazo (quando o trabalhador era exposto a materiais cancerígenos ou qualquer outro tipo de material nocivo à saúde) (DE ARAÚJO JÚNIOR et al., 2003).

Para os trabalhadores, a Automação Industrial veio para tirá-los destes trabalhos braçais, repetitivos ou com riscos à saúde. Para os empresários ela surgiu para otimizar os processos de produção, de forma a reduzir perdas, aumento da precisão (como em dosadores, por exemplo), e o principal: a redução do erro humano. Onde antes todo o processo era controlado por operadores, agora é dividido em duas seções, a Operacional e a de Controle. A seção Operacional consiste de apenas dos instrumentos necessários para que a máquina se desloque e efetue a ação desejada, sendo que estes instrumentos são os sensores e atuadores. A seção de Controle consiste dos instrumentos programáveis, como os Controladores Lógicos Programáveis, que atualmente, são os instrumentos mais utilizados devido ao alto fluxo de dados, alta confiabilidade e a sua resistência aos ambientes industriais (NATALE, 2011).

A Figura 1 demonstra um esquemático de como funciona um processo automatizado com o operador responsável apenas por operar o processo ou a máquina em questão. Para ele realizar esta interação com o processo, é necessário que exista uma Interface Homem-Máquina (IHM) que será a responsável por transmitir os dados do processo para o operador, e vice-versa

**Figura 1 – Diagrama de blocos sobre Processos de Controle.**



Fonte: Autorial Própria

Qualquer indústria que não seja completamente automatizada, ou seja, não esteja enquadrada nos padrões da Indústria 4.0 necessitará de operadores; e para estes operadores realizarem as interações necessárias com o processo, há a necessidade de uma IHM. Os processos evoluíram e ficaram complexos, porém a IHM deve ser algo simples e extremamente intuitivo para que não haja erros na interpretação do operador. Além disso, como os processos evoluíram, as IHMs também evoluíram de meros leds e botões para visores *touchscreen*, que agora possuem botões, leds, gráficos, desenhos e imagens, que tornam o seu uso mais autoexplicativo e intuitivo (COELHO, 2016; DA SILVA et al., 2011).

Visando esta necessidade, este trabalho aborda o desenvolvimento de uma IHM que possa ser acessada em qualquer dispositivo com acesso à internet para que esses comandos ou visualizações possam ser feitas de maneira mais simples, rápida e até mesmo de forma remota

### 1.1 TEMA E DELIMITAÇÃO DO TEMA

Comunicação de sistemas IHM em plataformas móveis com controladores lógicos programáveis da planta didática de processos de manufatura, alocada no Laboratório de Automação da Manufatura e Robótica, no campus de Ponta Grossa da UTFPR.

## 1.2 OBJETIVO

Desenvolver um manual didático detalhando a implementação de uma IHM *mobile* para a planta didática de Processos de Manufatura através do aplicativo *DataDashboard for LabVIEW™*, utilizando como supervisor o *software* LabVIEW™.

### 1.2.1 Objetivo Geral

Criar um manual didático descrevendo os passos para a Implementação de uma IHM *mobile* para a planta didática de Processos de Manufatura.

### 1.2.2 Objetivos Específicos

- Análise da planta didática;
- Estabelecer a comunicação entre PC e a planta didática de Processos de Manufatura;
- Desenvolver o programa em linguagem Ladder para a respectiva planta;
- Estabelecer a comunicação entre a planta didática de Processos de Manufatura Servidor e Cliente;
- Criar o supervisor com o aplicativo *DataDashboard for LabVIEW™*;
- Efetuar os testes práticos.

## 1.3 JUSTIFICATIVA

A indústria sempre busca por dispositivos em *software* e *hardware* mais atualizados e eficientes para otimizar sua produção, tanto em âmbitos de qualidade como de quantidade. Com essa crescente demanda das indústrias, profissionais capacitados são de extrema importância para efetuar a implementação e reparos desses *hardwares* e *softwares* na indústria.

Com esta necessidade, universidades tendem a propor novos meios de ensino, como as plantas didáticas onde é possível ter uma ideia de como funciona a planta de uma indústria. Com as plantas, surge a necessidade de acompanhar as

principais *Tags* (um marcador, neste projeto relaciona as informações do endereço do CLP a uma variável) dos sistemas, e com essa necessidade, surgem os sistemas supervisórios compatíveis em celulares e tablets, se apresentando assim a necessidade da implementação de Interface IHM *Mobile* em plantas didáticas

## 2 REFERENCIAL TEÓRICO

A Automação Industrial propriamente dita, começou no século XVIII, marcada pela criação da máquina a vapor: uma criação inglesa que inicialmente foi utilizada pra ampliar a produção de produtos manufaturados, sendo considerada a grande invenção que acompanhou a Primeira Revolução Industrial. Logo no século seguinte, houve uma grande expansão industrial, com novos meios de energia e tendo como outra grande invenção os relés mecânicos, sendo considerada a Segunda Revolução Industrial. No século XX surgiu a produção em larga escala através de linhas de montagem, conceito que foi idealizado e utilizado por Henry Ford na *General Motors* (GM). Alguns anos se passaram e a GM já possuía máquinas automatizadas através dos relés mecânicos, porém eram complexas e exigiam alta interconectividade. (SILVEIRA; LIMA, 2003).

A instrumentação teve o seu início na década de 1940 com os instrumentos pneumáticos, em que o operador não precisava efetuar a abertura e fechamento das válvulas manualmente. Alguns anos depois, com o avanço da eletrônica, os instrumentos pneumáticos foram substituídos por dispositivos eletrônicos, que eram mais robustos pois não possuíam partes móveis. Posteriormente, padronizou-se o sinal de corrente em 4-20 mA e 24 VDC de alimentação elétrica para transmissores e posicionadores (RINALDI; GUERRA, 2011).

Em 1968, foram desenvolvidos dispositivos eletrônicos para serem os substitutos dos relés mecânicos, o primeiro Controlador Logico Programável (CLP): o MODICON (*Modular Digital Controller*). Esses CLPs realizavam o chamado Controle Discreto através das variáveis digitais, essas variáveis possuíam dois estados lógicos representando as condições de fechado/aberto e desligado/ligado. Os CLPs eram compatíveis com a Arquitetura de Von Neumann (Programa Armazenado) e são normalmente programados através da Linguagem Ladder. Suas entradas e saídas são conectadas com atuadores, sensores, entre outros. Atualmente, os CLPs têm *firmwares* escritos em várias linguagens (o que contribui para ciclos mais rápidos), possuem sistemas compactos de I/O (*Input/Output*) e ainda podem detectar e diferenciar entre suas próprias falhas e também as falhas de sensores e atuadores (SILVEIRA; LIMA, 2003).

Com os CLPs, surgiu a necessidade de criar *softwares* para fazer a programação deles. Cada CLP tem o seu próprio *software* de programação,



programados em diagramas de blocos e depois traduzidos para a lógica Ladder, seguindo a normatização do Padrão IEC 611313, que define cinco linguagens de programação padrão (SEIXAS FILHO, 2000). Para realizar programas complexos e extensos, é necessário uma ótima esquematização do processo. O GRAFCET, além de uma linguagem de programação, pode ser usado como uma ferramenta para esquematizar estes processos, especialmente processos sequenciais. Em sua lógica existem etapas, transições e ações (MENDES, 2016).

Com todas essas melhorias na infraestrutura da automação industrial, era necessária uma interoperabilidade e uma comunicação mais sofisticada entre o supervisor, o CLP e a instrumentação. No começo, a automatização das indústrias não era completa, pois cada parte da operação tinha o seu sistema isolado sem comunicação com os demais, sendo substituído pelo controle centralizado onde todas as informações partiam de um único local. Por volta de 1960, surgiram as redes industriais, utilizando redes de computadores para unir os dados digitalmente. A hierarquia da rede tinha 3 divisões principais: Administração, Células da Manufatura, e Sensores e Atuadores. O primeiro nível Sensores e Atuadores era o responsável pela comunicação dos dispositivos (sensores e atuadores) utilizados no nível operacional (chão de fábrica). No segundo nível (células de manufatura, com os CLPs) estão os controladores que fazem o gerenciamento do processo. No último nível (administração), tem-se o controle gerencial da produção (FRANCHI; CAMARGO, 2011).

Com a crescente necessidade de melhor comunicação entre o controle, a manufatura e os sensores/atuadores, surgiram as redes industriais, que são formadas por 3 características básicas: meio físico, sistema especialista e unidade de processamento. Cada rede tem suas próprias características, protocolos e aplicações. Dentro das redes as mais conhecidas são: Fieldbus®, PROFIBUS®, Ethernet, Modbus, Interbus, e Devicebus (LUGLI; SANTOS, 2011).

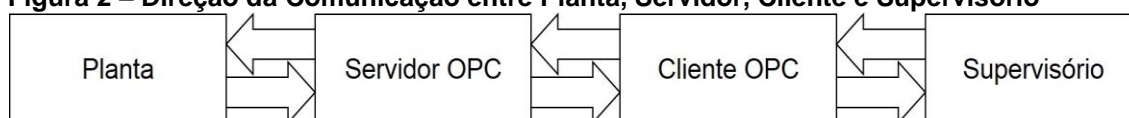
O padrão MPI (interface de passagem de mensagem) é um conjunto de interfaces pré-estabelecidas que existem para padronizar a passagem de mensagens. A sua primeira versão foi inicialmente lançada em 1994, após uma série de reuniões com fabricantes e desenvolvedores, esta versão era aplicada a comunicação ponto a ponto (WAKU, 2012).

Para realizar a comunicação entre planta e supervisor, além do padrão da interface de passagem de mensagem, é necessário estabelecer protocolos, para que

esta comunicação ocorra de forma segura e sem erros, assim surge o OPC (*Object Linking and Embedding for Process Control*) que é o nome dado a uma interface de comunicação padronizada para evitar problemas como a inconsistência dos *drivers*. A OPC Foundation é a responsável pelos padrões, normas e regras que regem o padrão OPC. Existem quatro elementos principais para que esta comunicação ocorra. O primeiro item é a planta, que deve possuir um CLP, onde todos os sensores e atuadores devem estar devidamente instalados. O segundo item é o servidor (*software OPC Server*). O terceiro item é o OPC *Client*. O quarto e último item é o Supervisório (ELIPSE, 2018).

A Figura 2 demonstra a hierarquia de comunicação em uma linha sendo que cada item só pode se comunicar com os que estão ao seu lado. Esta comunicação é *full-duplex*, ou seja, ambos os itens tanto enviam como recebem. O último item da comunicação é o supervisório; neste projeto será utilizado o sistema *Supervisory Control and Data Acquisition* (SCADA), é o responsável por coletar os dados da planta em tempo real. Estes programas evoluíram com o passar dos anos, sendo que inúmeras empresas baseiam os relatórios de ferramentas gerenciais nos dados do supervisório. Os supervisórios possuem quatro componentes físicos: sensores e atuadores, estações remotas de aquisição, rede de comunicação e as estações de monitoramento. A estação remota de aquisição é onde os CLPs se encontram, e na estação de monitoramento é onde estão alocadas as IHMs.

**Figura 2 – Direção da Comunicação entre Planta, Servidor, Cliente e Supervisório**



**Fonte: Autoria Própria**

O termo IHM ou HMI (*Human Machine Interface*) é uma interface na qual possibilita ao homem interagir com a máquina. Existem três tipos principais de IHM: *Operator Interface Terminal* (OIT), *Embedded*, e baseada em PC. A IHM OIT, também conhecida como a substituta das botoeiras, é usada principalmente para substituir chaves, leds, botoeiras e dispositivos eletromecânicos. Todos os instrumentos substituídos são representados graficamente em um display LCD. A IHM *Embedded* é utilizada quando há a necessidade de um *feedback* ou um monitoramento constante. A IHM de Supervisão ou Baseada em PC são as que possuem mais características

como flexibilidade, acesso remoto, gráficos e conectividade. Nesta categoria além do CLP é necessário também um *Software* Supervisor (SCADA). As principais características a serem levadas em consideração na escolha entre estes três tipos são: vida útil, custo, tamanho da aplicação, armazenamento de dados, conectividade, segurança, acesso remoto, gráficos, entradas e operações e flexibilidade. A partir destas características é escolhido qual interface é a ideal para o processo (DA SILVA et al., 2011).

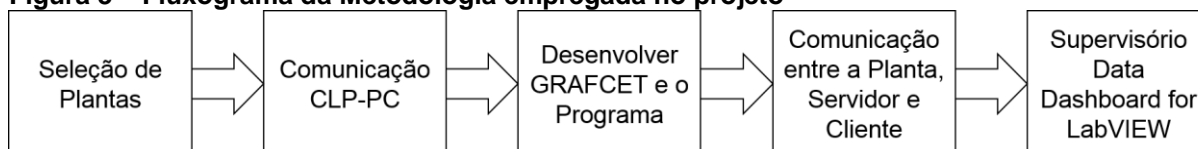
O programa a ser utilizado como supervisor é o LabVIEW™ (*Labaratory Virtual Instrument Engineering Workbench*), que é um programa da *National Instruments*, comumente usados no desenvolvimento de sistemas ou programas, onde é necessário o controle ou monitoramento de um determinado processo. Sua linguagem de programação gráfica surgiu em 1986, a qual deixa a programação muito mais fácil e dinâmica. A programação neste *software* é feita de acordo com o fluxo de dados (NATIONAL INSTRUMENTS, 2018).

Segundo Johnson e Jennings (2006), a *National Instruments* através do *software* LabVIEW™, iniciou e concretizou o conceito de *Virtual Instrument* (VI), ou instrumentação virtual. A instrumentação virtual possibilita levar ao computador todos os recursos dos elementos da instrumentação como *displays* e controles. Neste VI, podem ser inseridos os elementos gráficos dos dispositivos, além de ter a possibilidade de efetuar outras programações usando como base o valor adquirido pelos instrumentos. Um elemento virtual possui vários benefícios quando comparado a um real, tais como preço/performance, flexibilidade e customização.

### 3 METODOLOGIA

Para a realização deste projeto ocorrer corretamente, alguns passos devem ser seguidos. A Figura 3 mostra em sequência o passo a passo de execução do projeto.

**Figura 3 – Fluxograma da Metodologia empregada no projeto**



Fonte: Autoria Própria

Primeiro passo consiste em observar todas as plantas didáticas disponíveis na UTFPR campus Ponta Grossa para então selecionar uma planta para a realização do projeto.

Após selecionar a planta didática que receberá a implementação da IHM *mobile*, o próximo passo é realizar uma rápida análise no processo proposto por esta planta, para desta forma, desenvolver um GRAFCET claro e objetivo.

No segundo passo, deve ser realizadas as configurações da comunicação entre o CLP e o PC. Estas configurações serão executadas através do *Project Wizard* do *software* Simatic S7-300.

No terceiro passo será desenvolvido o GRAFCET do processo que será executado na planta didática; com o GRAFCET pronto, dará início a escrita do código em lógica Ladder. Posteriormente com o código finalizado, será compilado e baixado no CLP.

Para o quarto passo a comunicação entre a Planta didática, Servidor e Cliente deve ser instaurada. Neste passo será configurado todos os parâmetros da comunicação entre os mesmos, esta configuração será feita através dos seus respectivos *softwares*.

O último passo é a criação do supervisório no programa LabVIEW™, e realizar a comunicação deste, com o aplicativo de celulares e tablets, *DataDashboard* para criar uma IHM *mobile*.

## 4 DESENVOLVIMENTO

Para o desenvolvimento deste projeto é necessário saber qual planta didática será trabalhada para receber a implementação desta nova IHM. Neste ponto foi necessário realizar um levantamento geral das plantas didáticas, da UTFPR campus Ponta Grossa, disponíveis para a realização deste projeto. Após o levantamento das plantas disponíveis, foi escolhida uma planta que estava apta para a implementação da nova interface.

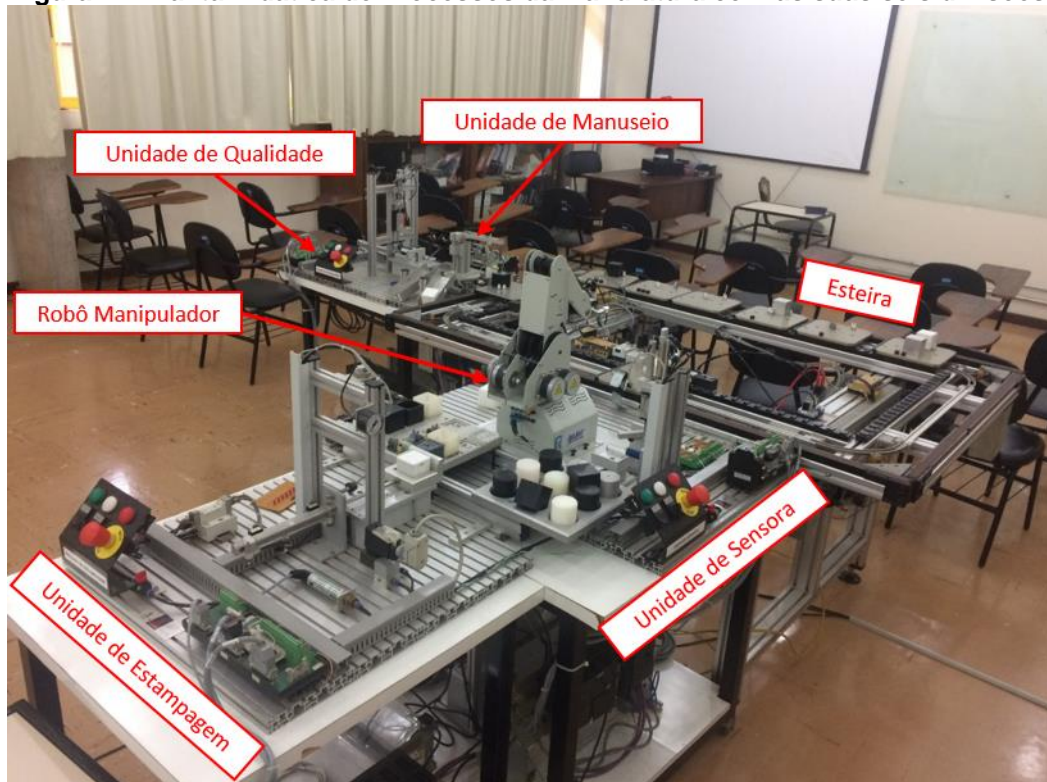
### 4.1 SELEÇÃO DA PLANTA

A planta didática selecionada foi a de Processos de Manufatura. Esta planta didática de Manufatura é dividida em seis seções, são elas: Unidade de Estampagem, Robô Manipulador, Unidade Sensora, Esteira, Unidade de Manuseio, Controle de Qualidade. Cada seção está conectada a um computador utilizando a interface MPI com um cabo DB9 padrão serial RS-232, podendo ainda conectar todas as seções em apenas um computador. Atualmente, é usado como padrão de comunicação a rede PROFIBUS®. Esta planta tem por objetivo demonstrar algumas etapas de um processo de fabricação. Nela, é proposto a montagem de uma caixa com um circuito impresso em seu interior.

O processo começa com o Robô Manipulador montando os três componentes: a caixa, a tampa com quatro parafusos e o circuito impresso. Após o Robô Manipulador ter montado a caixa, ele a coloca na Unidade de Estampagem. Os parafusos da tampa da caixa serão prensados sobre a caixa garantindo que ela fique fechada. Com a caixa montada e prensada, o Robô Manipulador a move para a Unidade Sensora onde será determinado a altura da caixa com a tampa, a cor da caixa (preto ou branco) e o seu material (plástico ou ferro). Com a caixa verificada, o Robô Manipulador a coloca sobre um *pallet* na Esteira. A Esteira possui oito *pallets* numerados. Além dos sensores de presença que checam a numeração dos *pallets*, a esteira ainda possui dois sensores de presença, onde é verificado se há caixa no *pallet* ou não. Após a caixa ser colocada em um dos *pallets*, ela segue caminho na esteira até chegar na Unidade de Manuseio, onde a caixa é movida para a seção do Controle de qualidade. O controle de qualidade possui três pinos para realizar a checagem de

funcionamento do circuito. Após o teste, a Unidade de Manuseio pega a caixa novamente e a coloca em uma de duas rampas, sendo elas, uma para as caixas defeituosas e a outra para as caixas aprovadas. Neste projeto será utilizado apenas a Unidade de Estampagem. A Figura 4 mostra uma foto da planta didática de processos da Manufatura com as suas devidas divisões.

**Figura 4 – Planta Didática de Processos da Manufatura com as suas seis divisões**



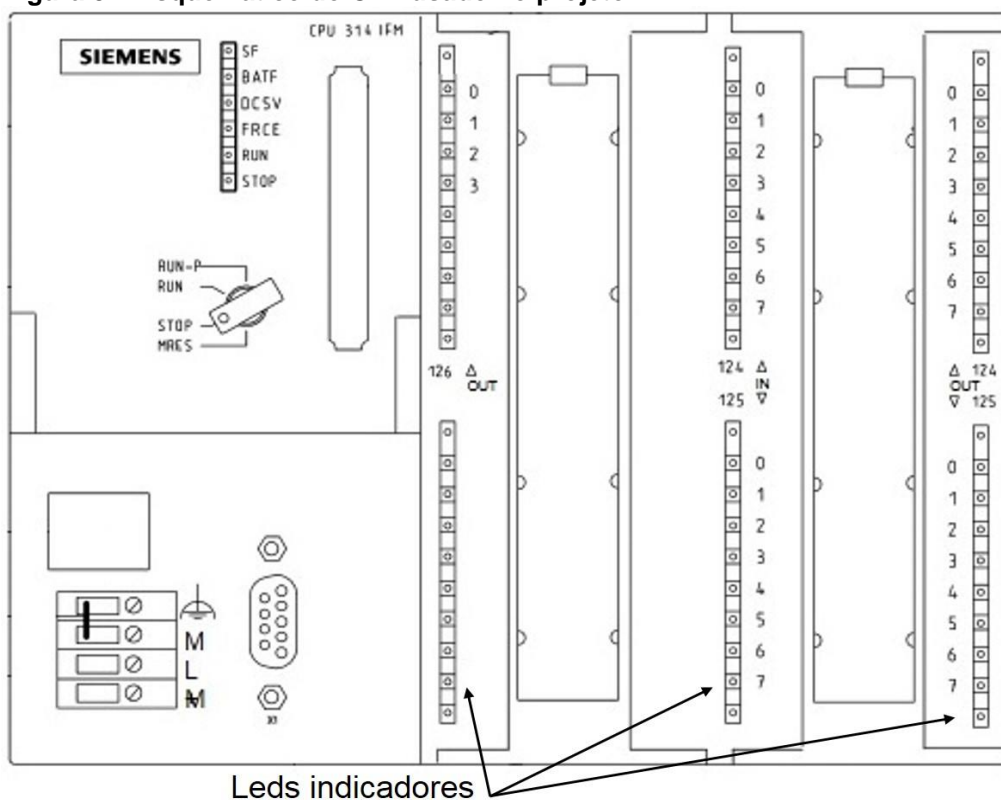
Fonte: Autoria Própria

## 4.2 COMUNICAÇÃO CLP-PC

Como citado anteriormente, o intuito de uma IHM é indicar ou alterar o valor de uma ou mais variáveis. Com isso em mente, é necessário saber quais são e de onde vem as variáveis que serão implementadas na IHM. Partindo desse princípio, é necessário ir mais especificamente na planta didática que receberá a implementação. Todos os sensores e atuadores desta planta devem estar conectados a um ou mais CLP, logo é necessário saber em qual CLP o elemento foi conectado e qual é o respectivo *Address* desse elemento no CLP. Para descobrir o endereço do elemento basta ir ao manual do fabricante da planta e procurar pelos endereços dos elementos

desejados. Caso a planta tenha sido adaptada, modificada ou de qualquer outra forma não esteja operando do modo, ou com os elementos que foi fabricada, a descoberta desse *Address* necessitará de outro método. O método mais comum é mudar o valor do elemento fazendo o seu *set* ou *reset* e ver qual endereço também mudou o seu valor, sendo este método chamado de tentativa e erro. A Figura 5 apresenta um esquemático do CLP da SIEMENS S7-300 CPU 314 IFM, que foi o utilizado no projeto. Na Figura 5 ainda é possível ver os leds indicadores das I/O que ele possui.

**Figura 5 – Esquemático do CLP usado no projeto**

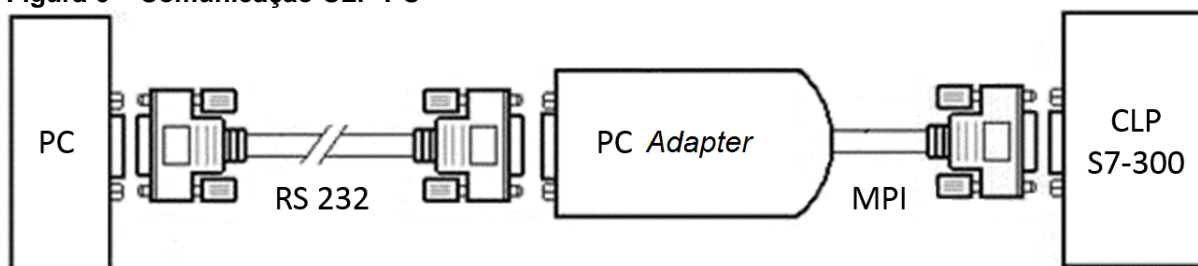


**Fonte: Adaptado de SIEMENS (2001)**

Depois de obtido o *Address* dos elementos desejados, o próximo passo é checar qual é a interface de comunicação, que neste CLP é a MPI.

A Figura 6 demonstra um esboço do meio de transmissão físico utilizado entre o CLP e o PC. Basicamente consiste de um cabo DB9 serial padrão RS232 e um PC *adapter* PC/MPI que realiza a conversão de MPI para RS232.

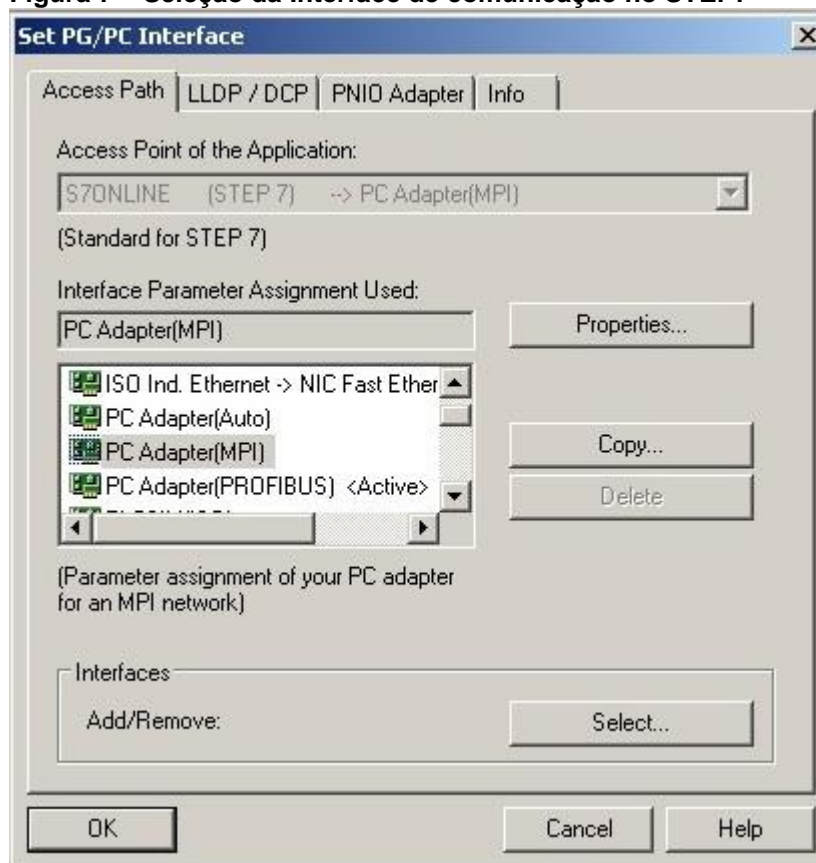
**Figura 6 – Comunicação CLP-PC**



Fonte: Adaptado de DHgate

A Figura 7 mostra como é a tela de seleção de interface do *software* STEP 7, nela é possível optar entre algumas interfaces como PROFIBUS® e MPI.

**Figura 7 – Seleção da Interface de comunicação no STEP7**



Fonte: Autoria Própria

Após o cabo de comunicação devidamente instalado, o próximo passo é ir para o *software* responsável por programar o CLP, o SIMATIC STEP 7. Nele é necessário escolher a interface, que no caso deste projeto é a PC Adapter (MPI). Depois desta interface devidamente instalada e selecionada, inicia-se a configuração da programação. Neste ponto, existem tópicos que devem ser escolhidos com cuidado



como: *baud rate*, *COM port*, *MPI Address*, CPU, entre outros. Esta configuração é de extrema importância para que a comunicação entre o PC e o CLP ocorra de forma segura e sem erros.

A Figura 8 demonstra o *Project Wizard* do programa STEP 7. Esta ferramenta existe para simplificar o passo a passo da configuração da comunicação entre CLP e PC. Nela, é possível observar duas importantes opções da configuração, que são a CPU que o CLP possui e a seleção do canal *MPI address*. O cabo serial normalmente é identificado como COM PORT1, mas sempre é importante checar para confirmar em qual porta ele foi reconhecido. Para saber qual endereço está sendo utilizado pela porta serial, basta ir ao Painel de Controle no menu iniciar do seu computador, posteriormente clicar em Hardware e Sons, e por último clicar em Gerenciador de Dispositivos. Por fim, basta procurar pelo cabo serial e ver qual porta foi designada para ele. O STEP7 e todos os outros programas que serão usados usam por padrão a COM PORT1, caso a máquina tenha reconhecido o cabo serial em outra porta basta ir nas propriedades do programa e alterá-la para a porta correta.

**Figura 8 – Project Wizard do software STEP7**

CPU Type	Order No
CPU313 C	6ES7 313-5BF03-0AB0
CPU313 C-2 DP	6ES7 313-6CF03-0AB0
CPU313 C-2 PtP	6ES7 313-6BF03-0AB0
CPU314	6ES7 314-1AG14-0AB0
CPU314 IFM	6ES7 314-5AE03-0AB0
CPU314 C-2 DP	6ES7 314-6CG03-0AB0

CPU name: CPU314 IFM(1)

MPI address: 2

Work memory 32 KB; 0.3ms/1000 instructions; DI 20/DO 16; AI 4/AO 1 integrated; MPI connection; multi-tier

Preview >>

< Back   Next >   Finish   Cancel   Help

**Fonte: Autoria Própria**

### 4.3 DESENVOLVIMENTO DO GRAFCET E O PROGRAMA EM LADDER

A Figura 9 mostra como fica a interface do *software* SIMATIC após a configuração inicial da comunicação.

**Figura 9 – Tela Inicial pós Configuração do STEP7**



**Fonte: Autoria Própria**

A utilização deste *software* segue com a criação do programa que será compilado no CLP. Para criar um programa que será executado em uma planta, alguns passos devem ser seguidos tais como a esquematização do processo (GRAFCET) e o levantamento dos sensores e atuadores e seus respectivos endereços. Após ter adquirido estes dados e feito o GRAFCET, como mostrado no Apêndice A, pode-se começar a escrever o programa separando em três partes fundamentais: ação (efetua os comandos como Set e Reset, *timer*, entre outros), etapa (realiza a ação) e transição (a condição necessária para passar para a próxima etapa), a elaboração e escrita do programa deve ser feita com cautela pois se projetado de forma errônea possivelmente acarretará em danos físicos aos funcionários e danos financeiros a empresa. O programa que será escrito seguirá a lógica Ladder, como mostrado no Apêndice B, e a lista de símbolos (*Symbol table*) no Apêndice C.

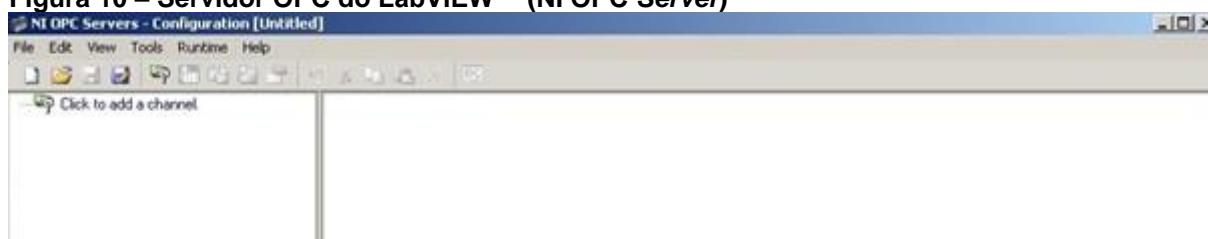
A linguagem Ladder se assemelha muito aos diagramas de lógica dos relés; é uma linguagem muito intuitiva para quem possui um conhecimento básico de eletroeletrônica.

Com o programa escrito, vem a compilação do programa no CLP. Para esta etapa ocorrer, a comunicação entre CLP e PC deve estar funcionando corretamente; com o CLP conectado no PC, basta fazer o *download* do programa.

#### 4.4 COMUNICAÇÃO ENTRE PLANTA, SERVIDOR E CLIENTE

Com o programa compilado, o próximo passo é a criação do Servidor OPC, como citado anteriormente o *software* que será usado como *server* é o LabVIEW™, para isto é necessário ter instalado dois módulos da *National Instruments*, são eles: Real Time e DSC. A Figura 10 mostra como é a interface do *software*.

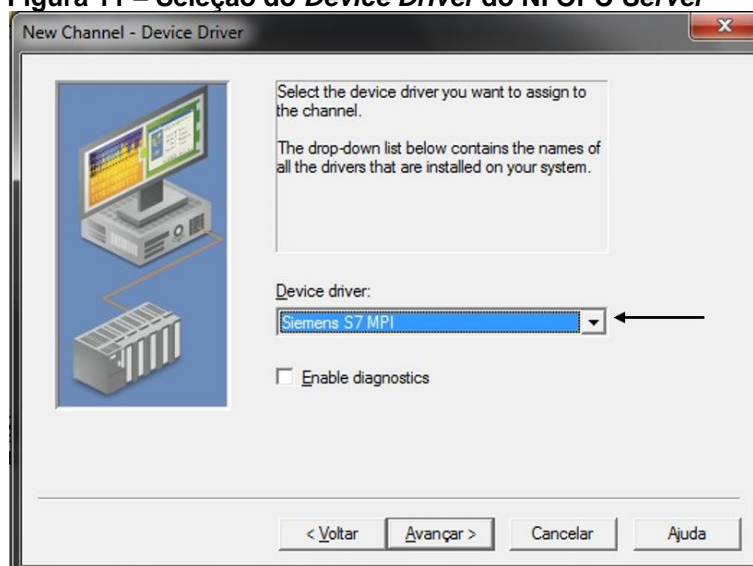
**Figura 10 – Servidor OPC do LabVIEW™ (NI OPC Server)**



Fonte: Autoria Própria

O primeiro passo para a criação do servidor é a criação de um canal (*channel*). Ele pode ser criado clicando em “*Click to add a channel*”, após esta opção ser selecionada, o *software* abrirá a tela de criação do canal de comunicação. A primeira operação é a edição do nome do canal, após esta definição a próxima operação é a seleção do “*device driver*” que consiste da definição do CLP, ou do dispositivo que será usado. Neste projeto o dispositivo usado é um CLP modelo Siemens S7 MPI. A Figura 11 mostra esta tela de seleção.

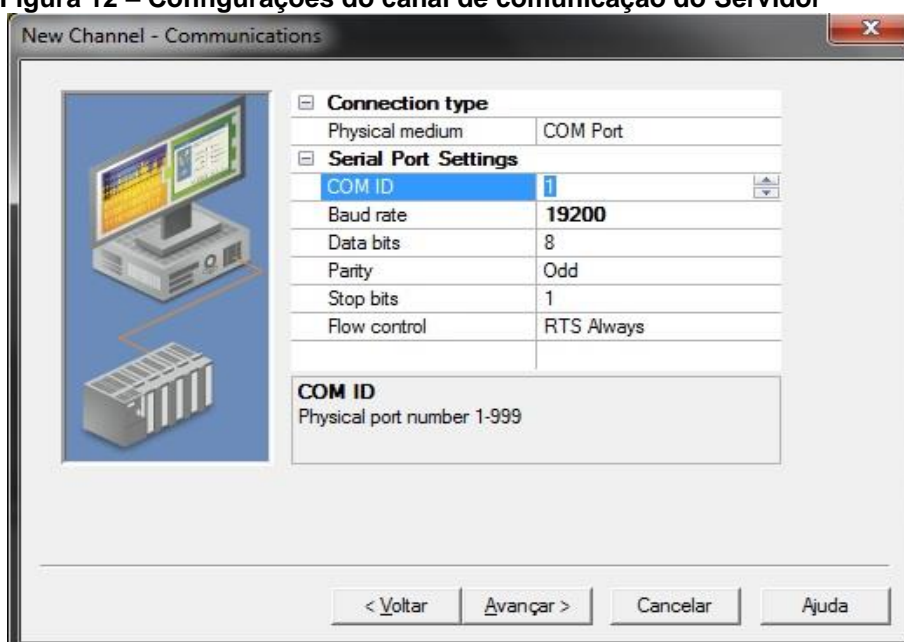
**Figura 11 – Seleção do Device Driver do NI OPC Server**



Fonte: Autoria Própria

Com o dispositivo corretamente selecionado, o próximo passo de configuração da comunicação é selecionar a porta de comunicação física (Ethernet ou COM). Caso seja Ethernet, o dispositivo que estará conectado, ao cabo ou a rede, também deverá ser escolhido. Caso a comunicação seja através de uma porta COM, algumas opções deverão ser selecionadas: endereço da porta, taxa de transmissão de dados, *data bits*, *parity*, *Stop bits*, *Flow control*. Neste projeto, será utilizado um cabo serial como meio de transmissão físico, logo ele será conectado a uma porta COM. Ao utilizar uma porta COM é necessário definir algumas opções, a primeira é o endereço da porta COM, que será o mesmo inserido anteriormente. O próximo item é a taxa de transmissão de dados, ou *baud rate*, que consiste em determinar a velocidade que a comunicação ocorrerá. Neste passo, é necessário ir ao manual do CLP para verificar qual é o seu *baud rate* padrão. Após ter adquirido este valor específico, basta selecioná-lo (é importante lembrar de sempre inserir o mesmo valor de *baud rate* quando solicitado), no caso do CLP utilizado neste projeto é 19200. Quanto as outras configurações oferecidas, aconselhasse a mantê-las no padrão em que foram fornecidas. Caso alguma mudança seja necessária, é importante mantê-las em mente caso esta opção seja solicitada em outro *software*. A Figura 12 demonstra como é esta tela de seleção.

**Figura 12 – Configurações do canal de comunicação do Servidor**

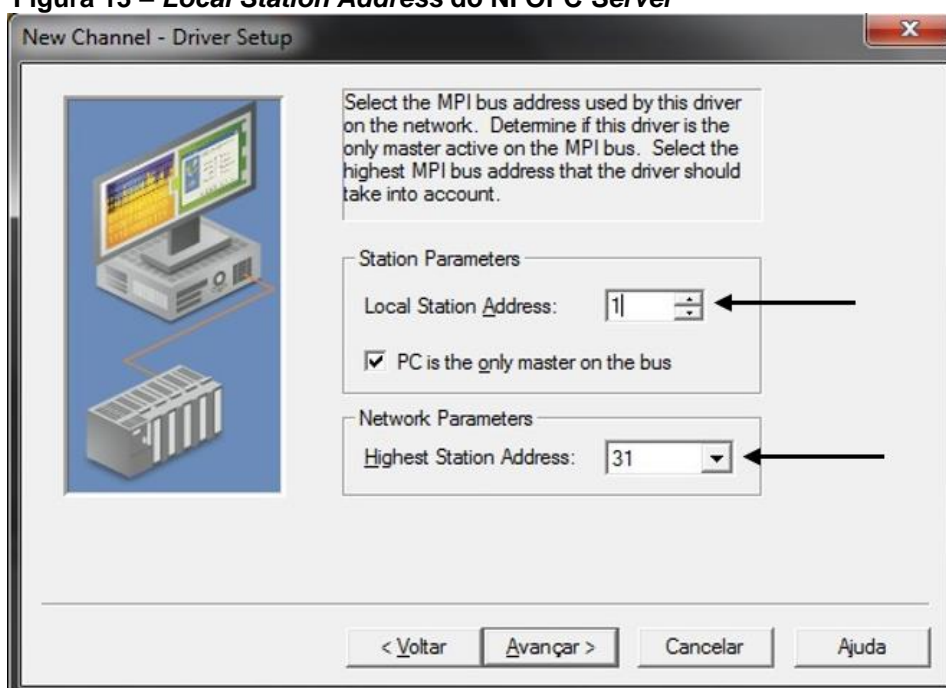


**Fonte: Autoria Própria**

A próxima etapa consiste em definir o endereço da estação local, ou seja, em qual canal de comunicação da MPI o computador será alocado e se ele será o único

mestre da rede. Ainda há a possibilidade de definir o número de estações que compõe a rede; sendo que para este projeto, serão necessários três estações (também pode ser chamado de três nós). Na primeira estação, ou *Station Address 0*, ficará o CLP; na segunda estação, ou estação endereço 1, está o computador (que será o mestre da rede); na terceira e última estação (endereço 2) ficará a IHM, que no caso é o *software* que será usado como supervisor. É importante definir em qual estação cada elemento do sistema ficará, para que, além de não haver conflito entre elas, os *softwares* saibam discernir sobre elas. A Figura 13 mostra como é a tela de definição da estação do computador, se ele será o mestre da rede ou não e quantas estações existirão na rede.

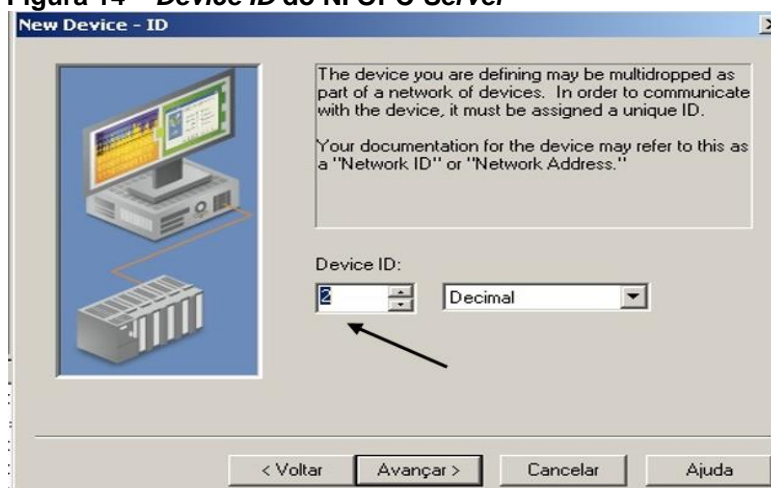
**Figura 13 – Local Station Address do NI OPC Server**



**Fonte: Autoria Própria**

O próximo passo da criação do servidor é a definição do *Device* (ou dispositivo). Primeiramente, como no passo anterior, inicia-se pela definição do nome do dispositivo. Com o nome estabelecido, é necessário definir o endereço da estação, sendo que, como citado anteriormente, este dispositivo ficará alocado no terceiro canal, (estação 2). A Figura 14 mostra como é esta definição do endereço da estação.

**Figura 14 – Device ID do NI OPC Server**



**Fonte: Autoria Própria**

Com o canal e o dispositivo corretamente criados e definidos, o próximo tópico da criação do servidor, é a definição das variáveis (*tags*). Neste ponto é necessário definir todas as *tags* que serão usadas no *software* SCADA, porque se uma *tag* necessária não for inserida, ela não poderá ser usada no supervisório. Para esta *tag* poder ser utilizada, ela deve estar declarada no *software* de programação (STEP 7), no servidor, no cliente e também no supervisório. A Figura 15 mostra como é a tela de declaração de *tags*. Como pode ser observado, esta tela apresenta algumas opções, como: Nome; endereço da *tag*; descrição; o tipo do dado que esta *tag* possui; se é uma *tag* de leitura, escrita ou ambos; e a taxa de atualização desta *tag*.

**Figura 15 – Tela de Definição das Tags do NI OPC Server**



**Fonte: Autoria Própria**

Com as *tags* devidamente criadas, deve-se checar se elas estão funcionando corretamente. Para isso, é necessário abrir o *software* OPC *Quick Client*, o qual



possibilita ao usuários saber quais *tags* estão funcionando corretamente, qual o formato e valor, além de quantas vezes ela mudou de valor. Para isso basta adicionar uma nova conexão de servidor e ir até o canal criado. A Figura 16 mostra como fica o teste quando as *Tags* estão funcionais.

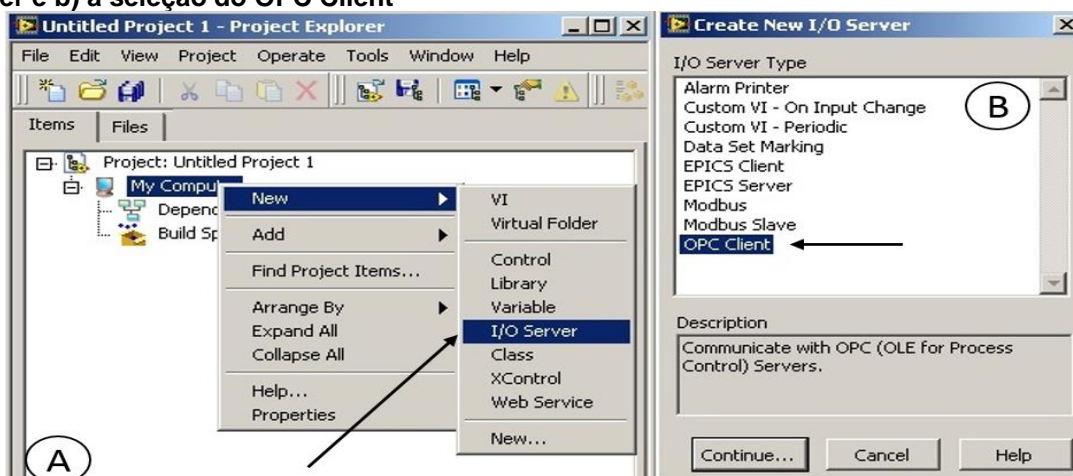
**Figura 16 – Teste das *Tags* do NI OPC Server por meio do cliente do próprio servidor.**

Item ID	Data Type	Value	Timestamp	Quality	Update Count
lab.vars.B6	Boolean	0	13:49:25.861	Good	1
lab.vars.B7	Boolean	1	13:49:25.861	Good	1
lab.vars.B8	Boolean	1	13:49:25.861	Good	1
lab.vars.B9	Boolean	0	13:49:25.861	Good	1
lab.vars.pressao	Word	16	13:49:37.158	Good	4
lab.vars.S10	Boolean	1	13:49:25.861	Good	1

**Fonte: Autoria Própria**

Após testar as variáveis, o próximo passo é a criação do cliente e para isso é necessário ir ao *software* LabVIEW™. A primeira etapa é criar um novo projeto dentro do LabVIEW™. Abrindo o programa e clicando em *Create Project*, tem-se um novo projeto. Após ter clicado nesta opção, o LabVIEW™ abrirá uma janela com o nome de *Project Explorer*. Clicando em *My Computer* com o botão direito do *mouse*, cria-se a comunicação com o servidor NI OPC, abrindo a opção *New* e clicando em *I/O Server*. Após isso o LabVIEW™ abrirá uma nova janela com o nome de *Create New I/O Server*, sendo necessário selecionar a opção *OPC Client* para iniciar os procedimentos da criação do cliente, sendo que estes passos são demonstrados na Figura 17.

**Figura 17 – *Project Explorer* do *software* LabVIEW™, em que a) mostra a seleção do I/O Server e b) a seleção do OPC Client**

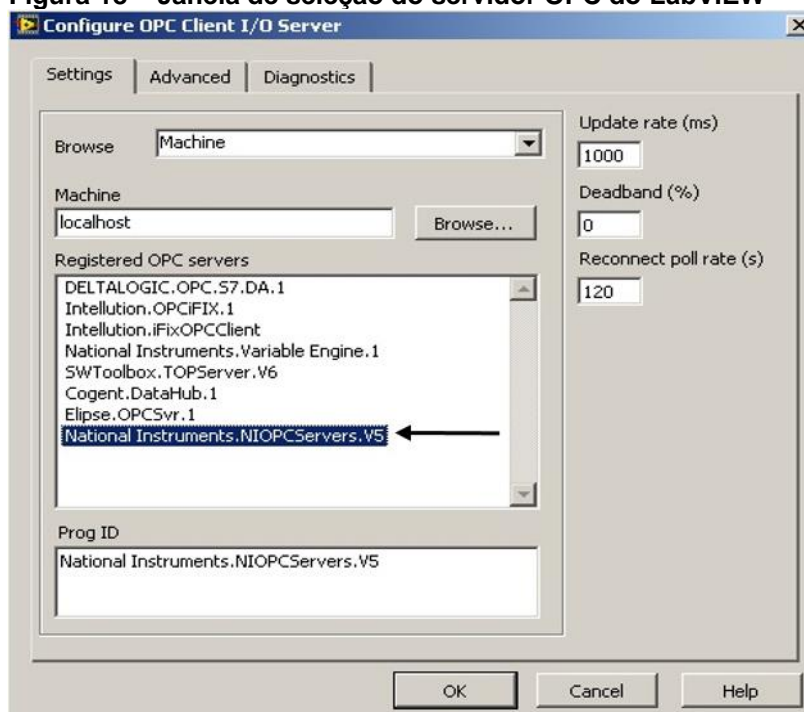


**Fonte: Autoria Própria**

Após efetuar os procedimentos citados acima e mostrados na Figura 16, o *software* irá abrir outra janela (*Configure OPC Client I/O Server*). Esta janela apresenta

algumas configurações que o usuário pode escolher, dentre as opções estão: taxa de atualização, onde o servidor está, em qual máquina ele está alocado, qual o *software* utilizado como servidor e o tempo de reconexão. Todas estas opções citadas são mostradas na Figura 18. Nela também pode ser observado a lista dos *softwares* que poderiam ser utilizados como servidores. Esta lista contempla apenas os *softwares* instalados no computador, logo o *software* utilizado como cliente pode ser compatível com outros servidores. Dos servidores da lista, será utilizado o National Instruments.NIOPCServers.V5, que corresponde ao servidor do LabVIEW™ criado anteriormente.

**Figura 18 – Janela de seleção do servidor OPC do LabVIEW™**



**Fonte: Autoria Própria**

Após efetuar os procedimentos da Figura 18, o LabVIEW™ voltará para a janela do *Project Explorer*, porém nesta janela agora haverá uma biblioteca. Nesta biblioteca, ficará todas as *tags* que serão importadas do *client*. Para importar basta clicar com o botão direito sobre o ícone que foi adicionado dentro da biblioteca. Neste projeto, o item que foi adicionado, “OPC1”, corresponde ao cliente do LabVIEW™. Após do *click* com o botão direito basta selecionar a opção “*Create Bound Variables*”. Estes passos e a tela são mostrados na Figura 19.



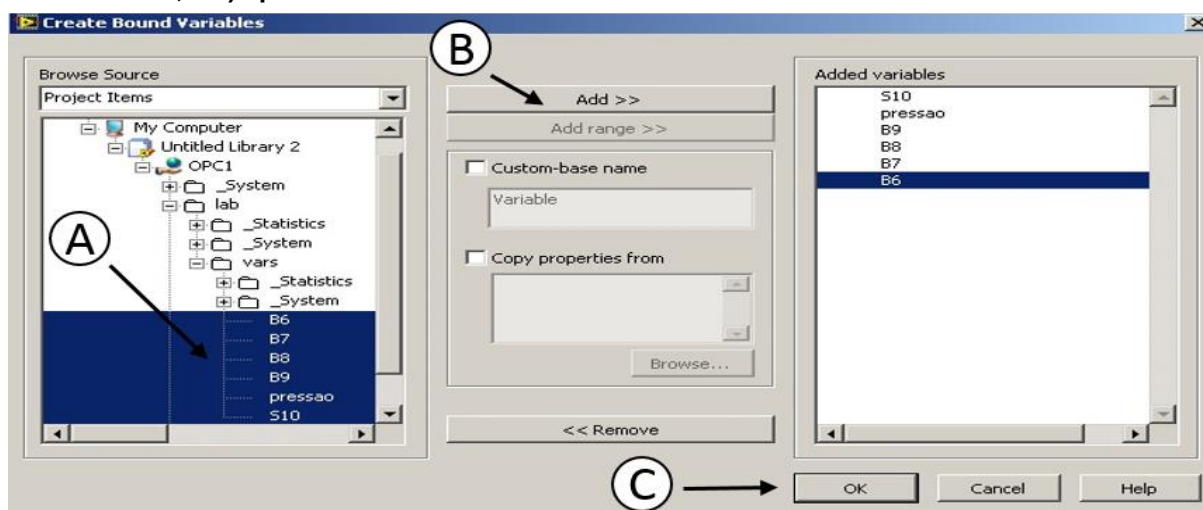
**Figura 19 – Variáveis Compartilhadas no *Project Explorer* do LabVIEW™**



**Fonte: A autoria Própria**

Após ter selecionado a opção citada acima o *software* abrirá uma nova janela, a qual possui o mesmo nome da opção selecionada, “*Create Bound Variables*”. Nesta janela deverão ser selecionadas todas as *tags* que serão importadas do Servidor, e consequentemente adicionadas à biblioteca do projeto. Para poder selecionar as variáveis é necessário expandir alguns menus da janela, a Figura 20, mostra os menus expandidos e a variáveis pretendidas já selecionadas. Após selecionar as variáveis o próximo passo, ainda nesta janela, é clicar no botão “*Add*” para adicioná-las a biblioteca do projeto, e por último basta clicar no botão “*OK*”.

**Figura 20 – Selecionando as Variáveis Compartilhadas no *Project Explorer* do LabVIEW™, onde a) mostra as variáveis selecionadas, b) o botão para adicionar as variáveis que foram selecionadas, e c) apertar OK.**



**Fonte: A autoria Própria**

Depois de ter efetuado estes passos, a janela da seleção das variáveis irá fechar, o LabVIEW™ voltará para a janela do *Project Explorer*, sendo que as variáveis

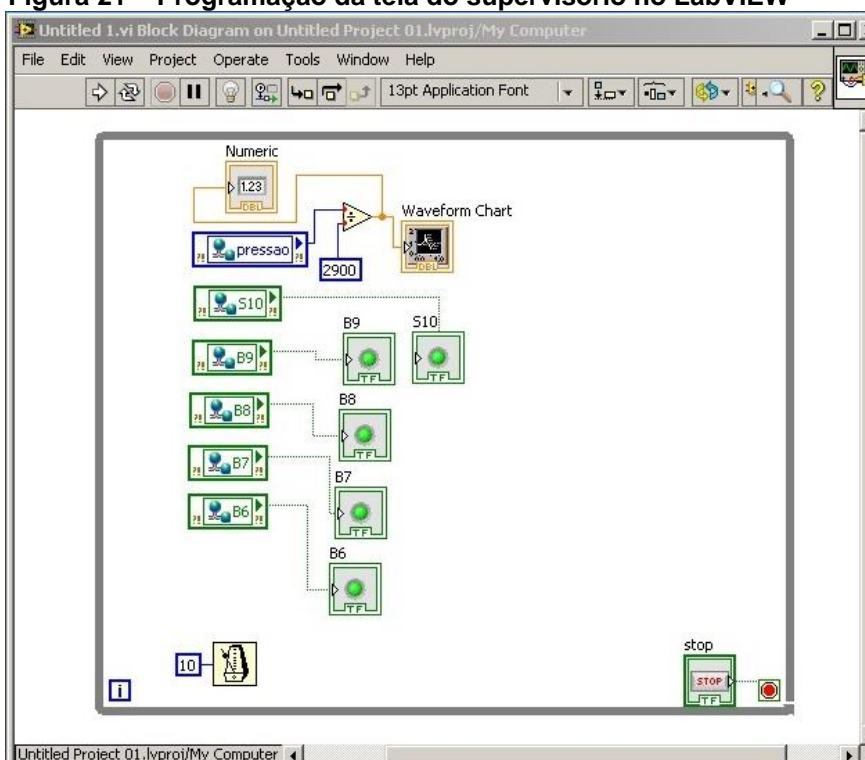
que foram selecionadas no passo anterior estarão na biblioteca. O próximo passo é criar um novo VI, no qual basta clicar na biblioteca com o botão esquerdo, e selecionar a opção *new VI*. Após estes procedimentos, o *software* abrirá duas telas: *Block Diagram* (Diagrama de Blocos) e *Front Panel* (Painel Frontal). No Painel Frontal é onde ficará todas as IHMs, já no Diagrama de Blocos fica toda a programação do programa. Para dar sequência ao projeto é necessário selecionar as variáveis na janela do *Project Explorer* e arrastá-las até a tela do Diagrama de Blocos. Com as variáveis que serão utilizadas na janela do diagrama de blocos, o próximo passo é a elaboração do programa do supervisor. Para isso, primeiro é necessário colocar um *loop While*, para que o programa que será feito execute além de um ciclo. O *While* se encontra no menu *Structures*, na aba *Programming* do Diagrama de Blocos.

Depois do *Loop*, é necessário colocar um botão para interromper o programa. Com o botão de “*Stop*” criado o próximo passo é criar os indicadores para as variáveis, para elas serão inseridos leds para as variáveis do tipo *Boolean*, com exceção da variável pressão que é *Uint16*, para esta será inserido um gráfico.

Por último, ainda na tela de programação, é preciso colocar um *Timing*, um tempo pré-estabelecido que, quando dentro de um *loop while* tem a função de colocar um *delay* entre um *loop* e outro. Dentre as opções do *timer*, a escolhida para este projeto foi a “*Wait Until Next ms Multiple*”. Com ele selecionada basta inserir o valor da constante, neste projeto o valor usado foi 1000, caso necessite de uma atualização maior basta reduzir este valor.

Além dos instrumentos já citados, foi adicionado um indicador numérico na variável “pressão”, e uma função de divisão na programação. Esta divisão ocorre entre o valor desta variável e o valor de uma constante. Neste caso, o valor da constante usada foi “2900”, sendo que este procedimento foi realizado devido a diferença de valores entre o valor direto da variável pressão e o valor demonstrado no pressostado da planta. A Figura 21 mostra como fica a programação do supervisor, pode-se observar o gráfico, os leds, o *loop While* e sua condição de parada, o indicador numérico e também a adequação dos valores através do processo de divisão citado anteriormente.

**Figura 21 – Programação da tela do supervisor no LabVIEW™**



**Fonte: Autoria Própria**

Com o supervisor criado, o próximo passo é a implementação da interface IHM *mobile*. Para este projeto foi optado usar um *tablet*, modelo TB700 da marca Orange. O primeiro passo neste *tablet* é instalar o aplicativo do LabVIEW™. O nome deste aplicativo é *Data Dashboard for LabVIEW™*. Sua instalação se dá pela loja de aplicativos do *Android*. Outra opção é procurar pelo apk (instalador de aplicativos da plataforma operacional *Android*) do aplicativo na internet e transferi-lo para o dispositivo desejado. Após a instalação do aplicativo, inicia-se a configuração da comunicação entre ele e o supervisor. O primeiro passo desta configuração, é descobrir qual o endereço IP da máquina onde está o supervisor. Há várias maneiras de descobrir o IP do computador, a que foi utilizada neste projeto segue os seguintes passos:

- 1º abrir o executar (comando do teclado: Windows + R).
- 2º digitar “CMD” na campo de texto e apertar o botão “Enter”.
- 3º digitar o comando “ipconfig”, e pressionar novamente o botão “Enter”

Após estes passos o CMD exibirá diversas informações sobre o IP da máquina. Para este projeto, a única informação que será utilizada é o endereço do IPv4. A Figura 22 mostra a tela do CMD com as informações do IP da máquina que foi criado o supervisor.

Figura 22 – Janela do CMD

```

C:\Windows\system32\cmd.exe
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\QUALIDADE>ipconfig

Configuração de IP do Windows

Adaptador Ethernet Conexão local:

    Sufixo DNS específico de conexão. . . . . : pg.utfpr.edu.br
    Endereço IPv6 de link local . . . . . : fe80::7c:38df:2655:b518%11
    Endereço IPv4. . . . . : 172.25.3.99
    Máscara de Sub-rede . . . . . : 255.255.0.0
    Gateway Padrão. . . . . : 172.25.1.1

Adaptador de túnel isatap.pg.utfpr.edu.br:

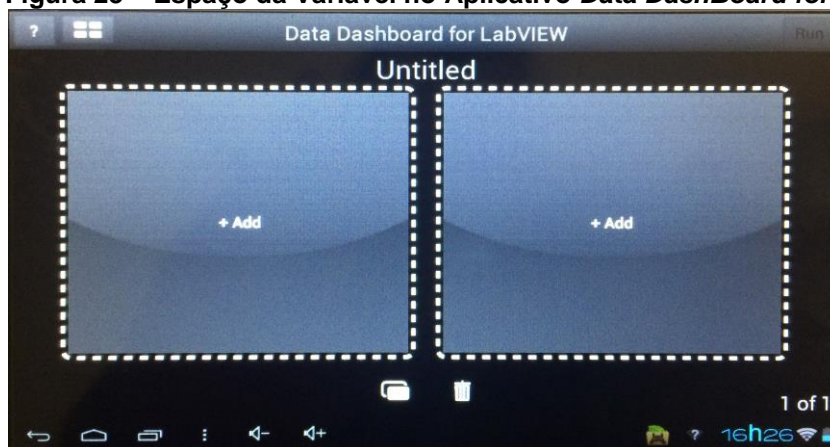
    Estado da mídia. . . . . : mídia desconectada
    Sufixo DNS específico de conexão. . . . . : pg.utfpr.edu.br

C:\Users\QUALIDADE>_

```

Fonte: Autoria Própria

O segundo passo da configuração é ter certeza que tanto o computador quando o dispositivo (*tablet*) estejam na mesma rede de internet. Sabendo o endereço IPv4, o próximo passo é abrir a aplicativo no dispositivo. Com o aplicativo aberto, ele apresentará duas opções de *layout*: uma que apresentará uma variável por tela e outra que apresentará duas por tela. Neste projeto, como serão monitoradas várias variáveis, foi optado pela tela com duas variáveis. Após selecionar a opção do *layout*, o aplicativo mostrará o espaço da IHM porém ainda sem uma variável anexada, para adicionar uma variável a este espaço, basta clicar nele. A Figura 23 mostra esta tela da seleção do espaço que será inserido a variável.

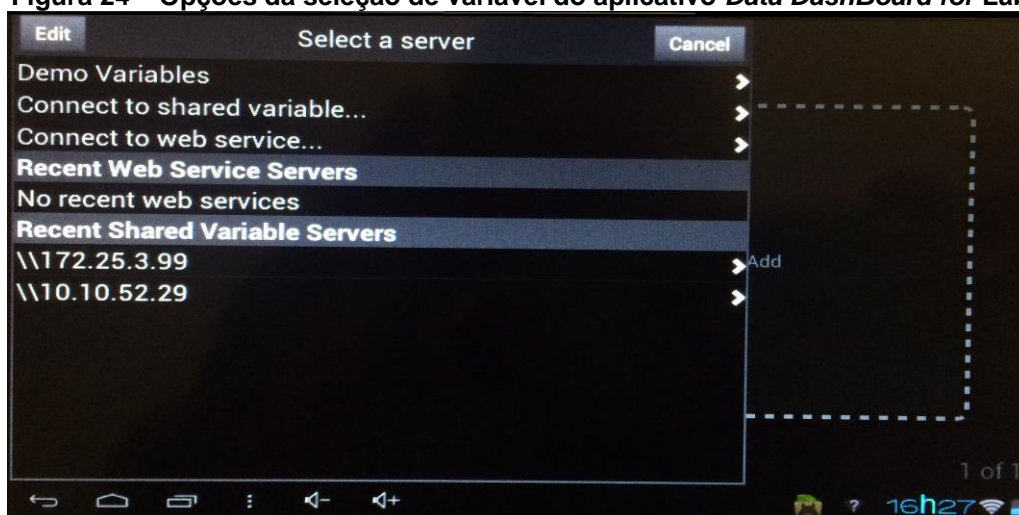
Figura 23 – Espaço da Variável no Aplicativo *Data DashBoard for LabVIEW™*

Fonte: Autoria Própria

Após clicar no espaço desejado, o aplicativo abrirá a tela de configuração de comunicação, nesta tela haverá várias opções como: *Demo Variables*, *Connect to*

*shared variable*, *Connect to web service*. Neste projeto foram usadas *shared variables* devido a comunicação entre o cliente e supervisor, logo a opção que deve ser selecionada é a *Connect to shared variable*. Estas opções são mostradas na Figura 24.

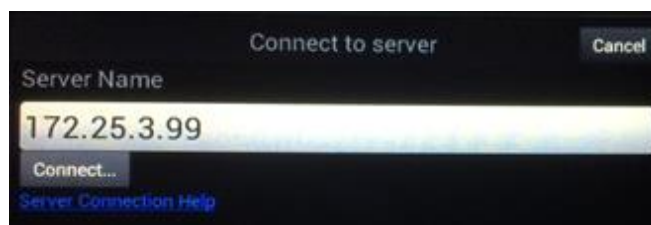
**Figura 24 – Opções da seleção de variável do aplicativo *Data DashBoard for LabVIEW™***



**Fonte: Autoria Própria**

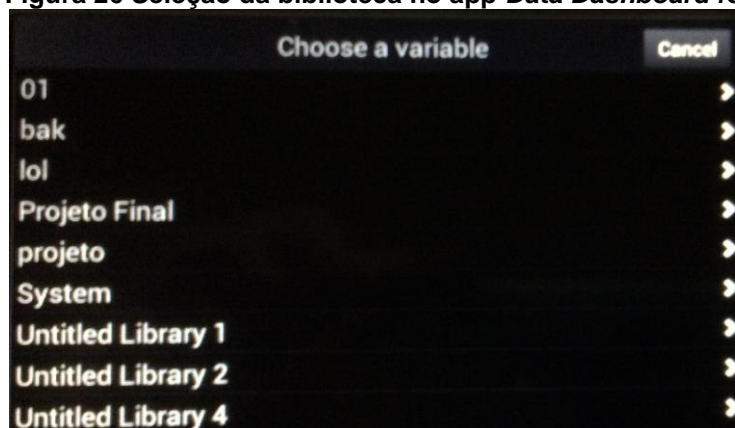
Após selecionar esta opção, o aplicativo vai solicitar o endereço IP da máquina, onde foi criado o supervisor, os passos para a aquisição do endereço IP já foram citadas e demonstradas na Figura 21. Neste ponto basta digitar os dígitos já adquiridos do IPv4, sendo que neste, projeto os dígitos são os seguintes: 172.25.3.99. A Figura 25 mostra esta tela de digitação já com o endereço IPv4 inserido.

**Figura 25 – Inserindo o Endereço IPv4 no Aplicativo *Data DashBoard for LabVIEW™***



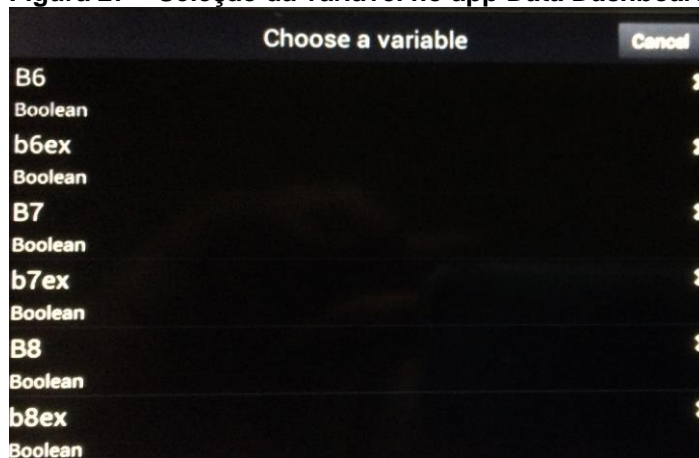
**Fonte: Autoria Própria**

Após digitar corretamente o endereço IP, o aplicativo apresentará todas as bibliotecas que já foram criadas no LabVIEW™, além de ser necessário selecionar a biblioteca que foi criada no projeto do supervisor. Esta tela de seleção é demonstrada na Figura 26.

Figura 26 Seleção da biblioteca no app *Data Dashboard for LabVIEW™*

Fonte: Autoria Própria

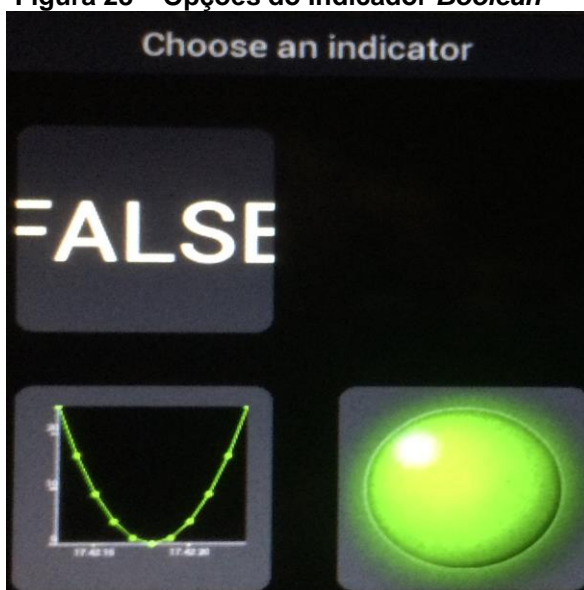
Após selecionar a biblioteca, o aplicativo apresentará todas as variáveis compartilhadas disponíveis, todas elas poderão ser adicionadas, porém serão divididas em mais de uma tela, pois o *layout* permite no máximo duas variáveis por tela. A Figura 27 demonstra esta tela de seleção das variáveis.

Figura 27 – Seleção da variável no app *Data Dashboard for LabVIEW™*

Fonte: Autoria Própria

Ao selecionar uma variável o aplicativo exibirá algumas opções de indicadores ou controladores. A Figura 28 demonstra as opções de indicadores para uma variável *Boolean*.



Figura 28 – Opções do Indicador *Boolean*

Fonte: Autoria Própria

Para este projeto, os indicadores escolhidos para as variáveis *boolean* foram leds e para a variável Uint16 foi optado por um gráfico. O processo para inserir outras variáveis é o mesmo que o que já foi demonstrado.

## 5 RESULTADOS

Com a sequência tomada neste projeto, duas formas de apresentação dos dados vindos do processo, foram obtidos ao longo do processo de desenvolvimento. A primeira delas foi obtida com o supervisório finalizado e a segunda é a interface IHM desenvolvida no *tablet*.

Como já descrito, o supervisório foi desenvolvido no *software* LabVIEW™. Neste supervisório, existem cinco LEDs indicadores, “B6”, “B7”, “B8”, “B9”, “S10”. O nome destes indicadores corresponde aos sensores fim de curso dos atuadores, e um fim de curso que atua como sensor de presença. Abaixo uma lista da relação dos nomes e suas respectivas funções.

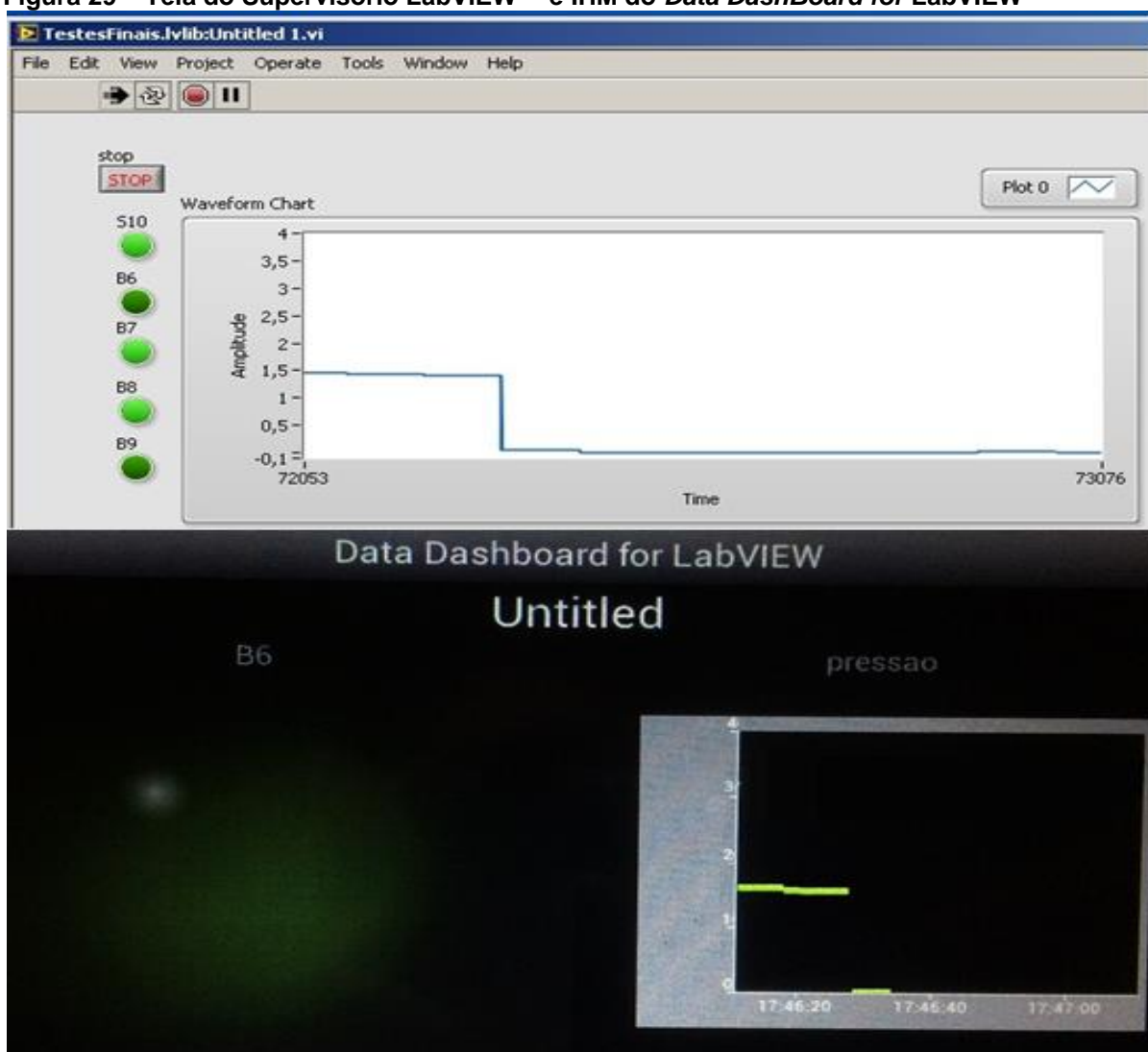
- B6 – Sensor fim de curso, recuo do pistão posicionador.
- B7 – Sensor fim de curso, avanço do pistão posicionador.
- B8 – Sensor fim de curso, recuo do pistão estampador.
- B9 – Sensor fim de curso, avanço do pistão estampador.
- S10 – Sensor fim de curso, posicionado para indicar quando há uma caixa no pistão posicionador.

Além destes indicadores também existe um indicador numérico e um gráfico do valor da pressão.

A interface homem máquina que foi desenvolvida no *tablet* possui simplicidade em sua aparência para a fácil interpretação dos dados. Todas as variáveis presentes no supervisório também foram inseridas nesta IHM, sendo que as variáveis estão divididas em telas, cada tela possuindo duas variáveis. A Figura 29 mostra, tanto o supervisório, quanto a IHM implementada, e a Figura 30 demonstra a segunda tela da IHM implementada no *tablet*.

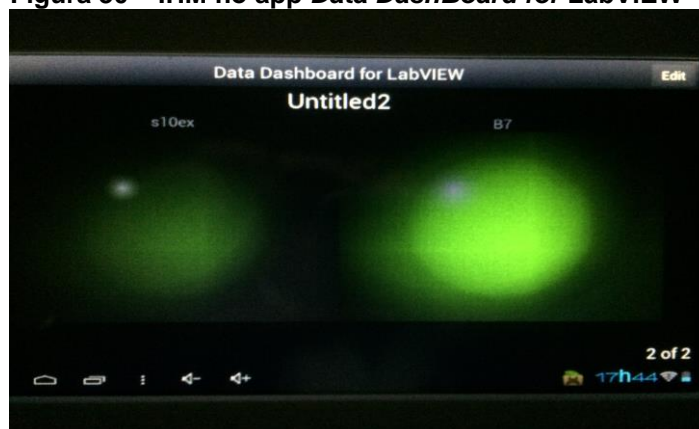


Figura 29 – Tela do Supervisor LabVIEW™ e IHM do *Data DashBoard for LabVIEW™*



Fonte: Autoria Própria

Figura 30 – IHM no app *Data DashBoard for LabVIEW™*



Fonte: Autoria Própria

## 6 CONSIDERAÇÕES FINAIS

O desenvolvimento da presente implementação possibilita aos futuros alunos, terem outra perspectiva das atualizações industriais, podendo usar os procedimentos mostrados neste projeto, ou desenvolver novos métodos com outros *softwares* supervisórios.

Uma possibilidade não abordada neste projeto é a criação desta IHM em uma página da web, ainda utilizando como supervisório o *software* LabVIEW™, porém em vez de usar um aplicativo, seria utilizada uma ferramenta do LabVIEW™ chamada *Web Service*. Essa é uma das propostas de trabalhos futuros nesta linha de desenvolvimento.

A implementação desta nova interface apresentou uma série de dificuldades, especialmente relacionadas a comunicação dos dispositivos. Estas dificuldades de comunicação ocorreram durante a conexão de elementos diferentes do sistema, como por exemplo, entre o supervisório e o *tablet*. O problema encontrado na comunicação entre o supervisório e o *tablete* foi referente a rede Wi-Fi da UTFPR: o *tablet* usado não conectava nesta rede. Para contornar esta barreira foi utilizado um roteador de Wi-Fi comercial, de forma ao *tablet* estar conectado a mesma rede do supervisório.

Por fim este projeto agregou vários conhecimentos, principalmente relacionados ao *software* LabVIEW™, já que com este projeto percebeu-se que o leque de atuação deste *software* é bem extenso. Outras áreas de conhecimento também foram estudadas ou aprimoradas pois, este projeto consiste desde a configuração e programação de um CLP até a implementação de uma IHM *mobile*.

## REFERÊNCIAS

COELHO, Pedro Miguel Nogueira. **Rumo à indústria 4.0**. Dissertação de Mestrado. FCTUC – Faculdade de Ciências e Tecnologia Universidade de Coimbra, 2016.

DA SILVA, Gustavo Peloi; BERTONCELO, Valdecir; COGO, Filipe. **Análise de Interfaces Homem-Máquina (IHM) em Equipamentos Controlados Eletronicamente**. Centro Universitário de Maringá, 2011.

DE ARAÚJO JÚNIOR, Antônio Pereira; DAS CHAGAS, Christiano Vasconcelos; FERNANDES, Raphaela Galhardo. **UMA RÁPIDA ANÁLISE SOBRE AUTOMAÇÃO INDUSTRIAL**. Universidade Federal do Rio Grande do Norte, 2003.

DHGATE. **PC / MPI +: RS232 isolado Siemens S7-300 adaptador de Programação / 400 PLC, substitua 6ES7 972-0CA23-0XA0**. Disponível em: <<https://pt.dhgate.com/product/pc-mpi-rs232-isolated-siemens-s7-300-400/72591575.html>> Acesso em: 03/07/2018.

ELIPSE. **Elipse lança duas ferramentas OPC**. Disponível em: <<https://www.elipse.com.br/drivers-development/elipse-lanca-duas-ferramentas-opc/>> Acesso em: 28/03/2018.

FARIAS, Leonel Marques; SELLITTO, Miguel Afonso. **Uso da energia ao longo da história: evolução e perspectivas futuras**. Revista Liberato, v. 12, n. 17, p. 01-106, 2011.

FRANCHI, Claiton Moro; CAMARGO, Valter Luís Arlindo de. **Controladores Lógicos Programáveis – Sistemas Discretos**. São Paulo: Érica, 2011.

JOHNSON, Gary W. JENNINGS, Richard. **LabVIEW™ Graphical Programming**. 4ed McGraw-Hill Pg 4 – 5. 2006.

LUGLI, Alexandre Baratella; SANTOS, Max Mauro Dias. **Redes Industriais Para Automação Industrial: As-i, PROFIBUS® e Profinet**. São Paulo: Érica, 2011.

MENDES, Cleiton Rodrigues. **Conceitos básicos da linguagem GRAFCET**. Institutografcet, 2016.

NATALE, Ferdinando. **Automação Industrial**. São Paulo: Érica, 2011.

NATIONAL INSTRUMENTS. **O que é LabVIEW™?** Disponível em: <<http://www.ni.com/pt-br/shop/LabVIEW™.html>> Acesso em 22/03/2018.

RINALDI, Enoque; GUERRA, Andreia. **História da ciência e o uso da instrumentação: construção de um transmissor de voz como estratégia de ensino**. *Caderno Brasileiro de Ensino de Física*, 2011, 28.3: 653-675.

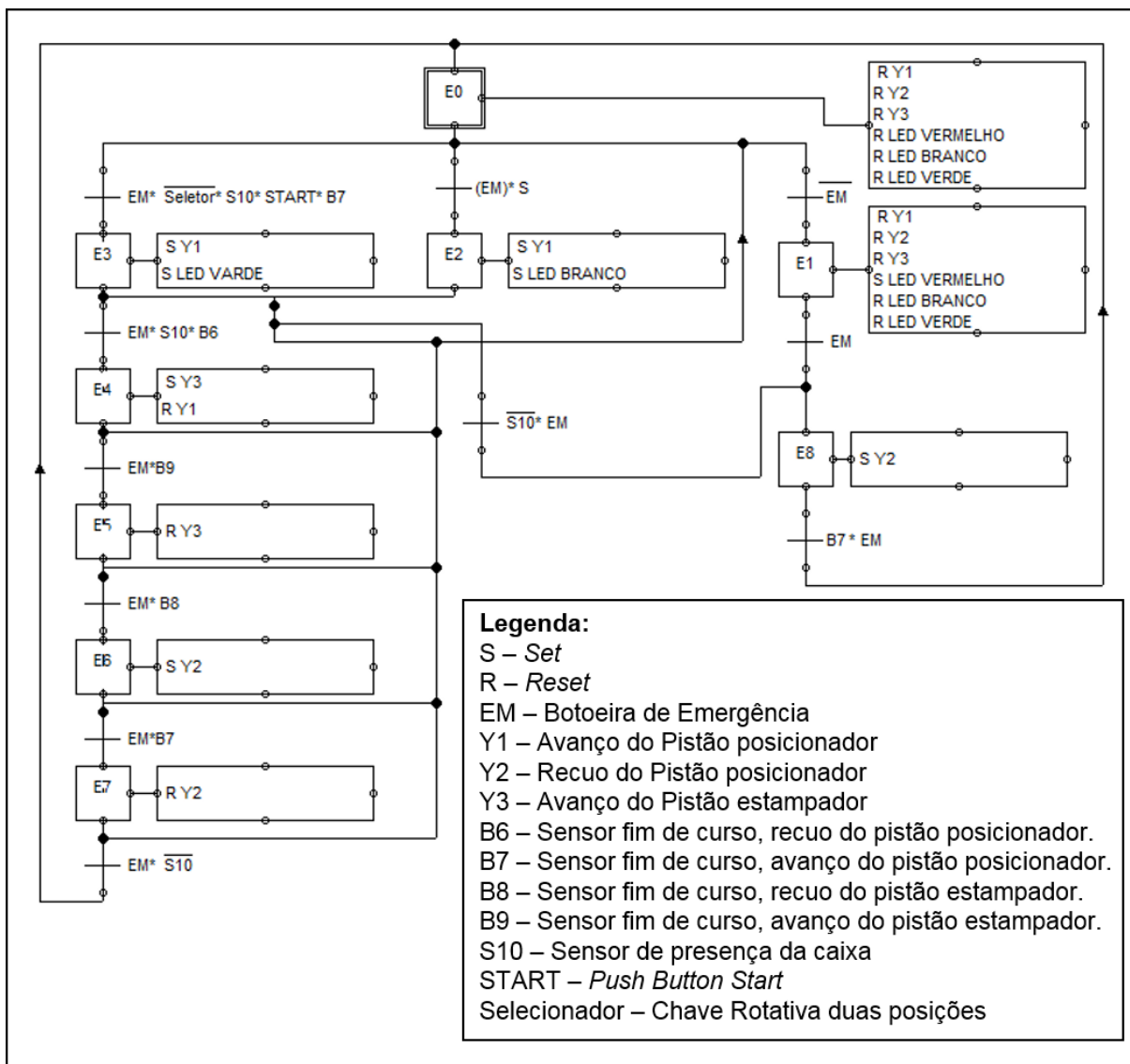
SEIXAS FILHO, Constantino. **A automação nos anos 2000: uma análise das novas fronteiras da automação**. ATAN Sistemas de Automação, 2000.

SIEMENS. **SIMATIC PLC S7-300, CPU Specifications CPU 312 IFM to CPU 318-2 DP**. Reference Manual, Edição 10/2001.

SILVEIRA, Leonardo; LIMA, Weldson Q. **Um breve histórico conceitual da automação industrial e redes para automação industrial**. Redes para Automação Industrial. Universidade Federal do Rio Grande do Norte, 2003.

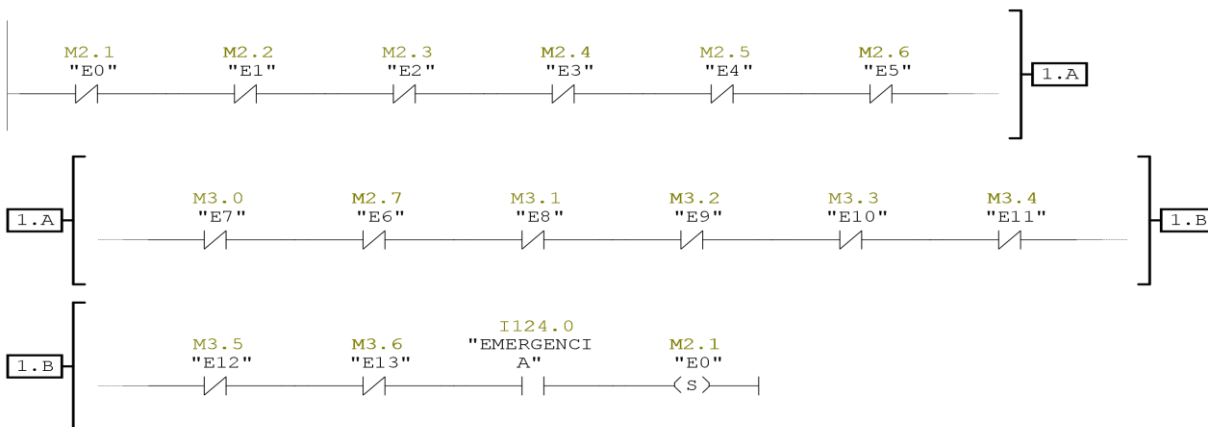
WAKU, Gustavo Mitsuyuki. **Message Passing Interface (MPI)**. Instituto de Computação (IC) – Universidade Estadual de Campinas (UNICAMP) Campinas, SP – Brazil, 2012.

**APÊNDICE A** - GRAFCET do programa que foi criado no STEP7.



**APÊNDICE B** - Programação em Ladder do *software* SIMATIC STEP7.

Network: 1 INICIO



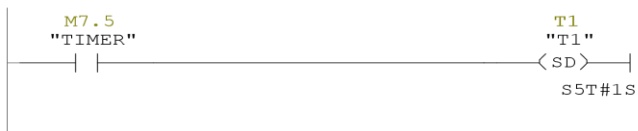
Network: 2

L PIW 128  
 T MW 1500  
 T "pressao" MW120

Network: 3

L 5000  
 T MW 130  
 T PQW 128

Network: 4

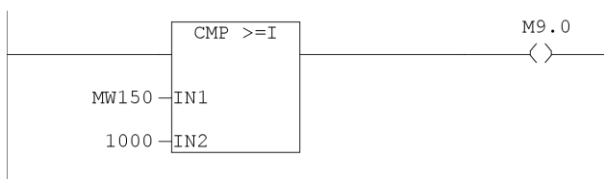




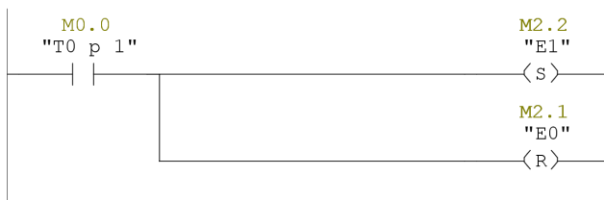
Network: 5



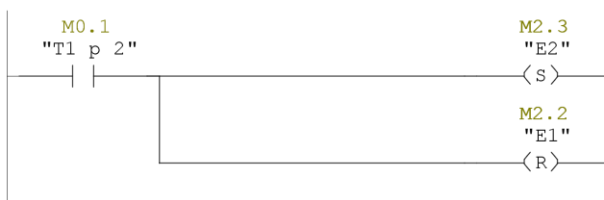
Network: 6



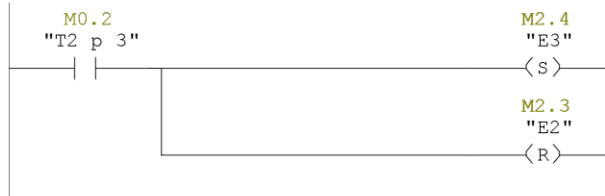
Network: 7 Transição 0 para 1



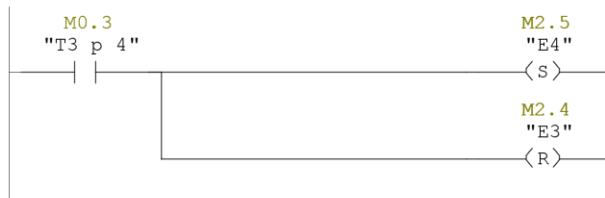
Network: 8 Transição 1 para 2



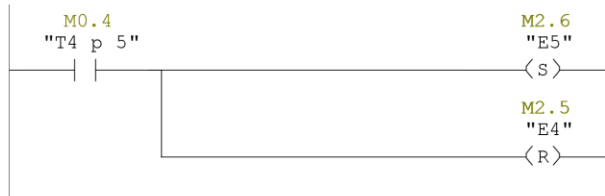
Network: 9 Transição 2 para 3



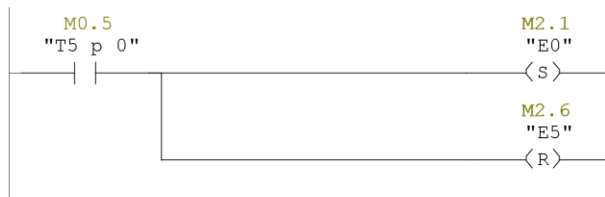
Network: 10 Transição 3 para 4



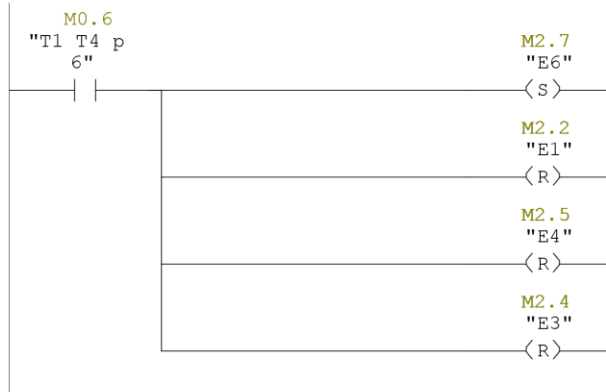
Network: 11 Transição 4 para 5



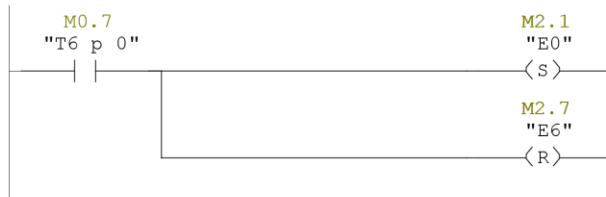
Network: 12 Transição 5 para 0



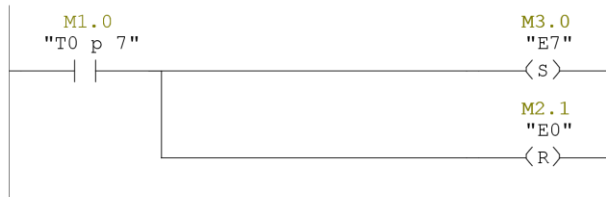
Network: 13      Transição 1 ou 4 para 6



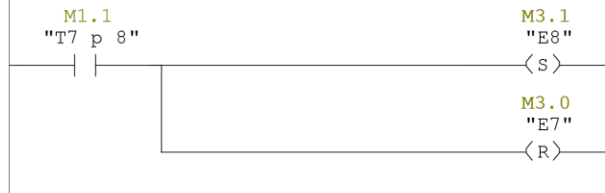
Network: 14      Transição 6 para 0



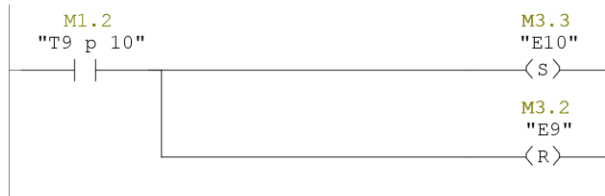
Network: 15      Transição 0 para 7



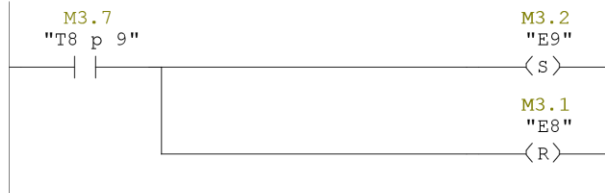
Network: 16      Transição 7 para 8



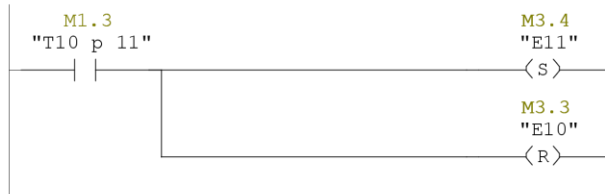
Network: 17      Transição 9 para 0



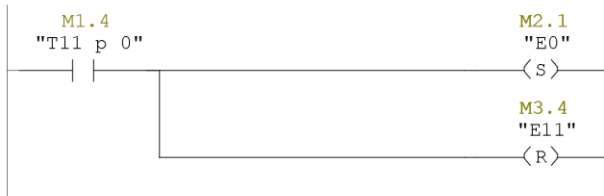
Network: 18      Transição 8 para 9



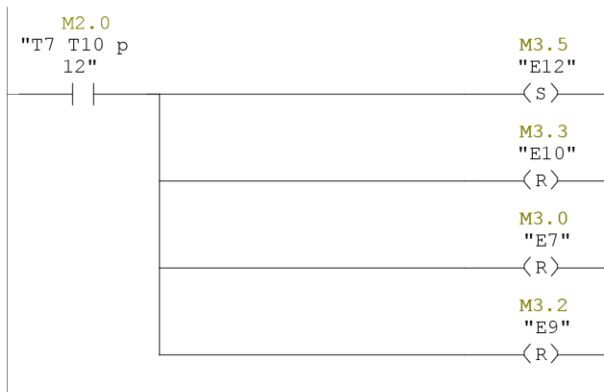
Network: 19      Transição 10 para 11



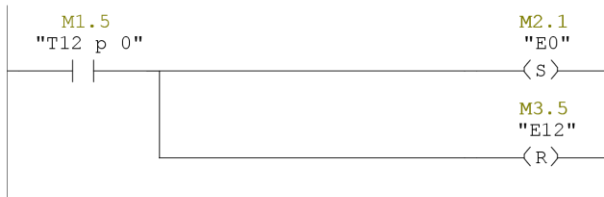
Network: 20      Transição 11 para 0



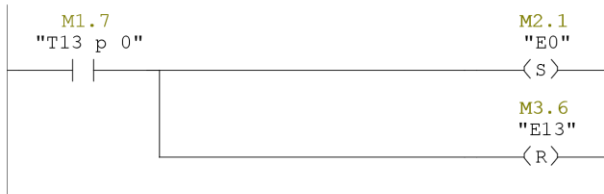
Network: 21      Transição 7, 9 ou 10 para 12



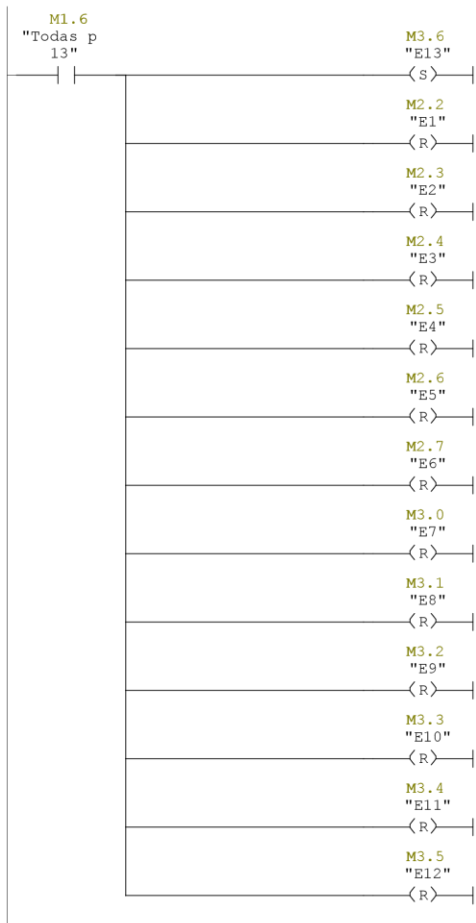
Network: 22      Transição 12 para 0



Network: 23      Transição 13 para Zero



Network: 24      TODAS PARA 13



SIMATIC

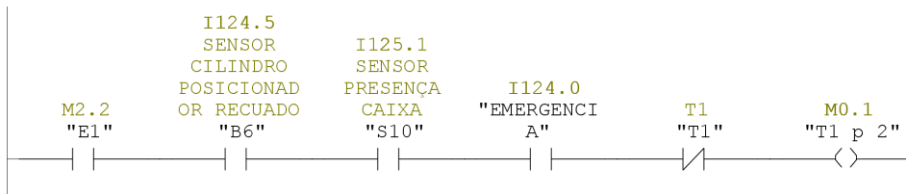
Bruneli2017\  
SIMATIC 300(1)\CPU 314IFM\...\OB1 - <offline>

06/05/2018 05:45:39 PM

Network: 25      Etapa Zero FALTA ACRESCENTAR BRAÇO ROBO



Network: 26      Etapa 1 - Verificar se tem caixa



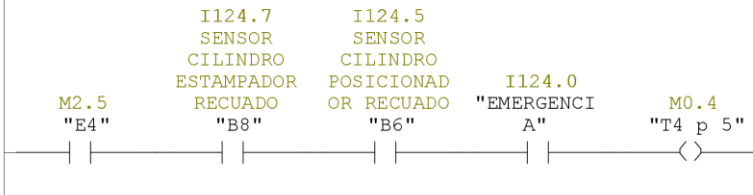
Network: 27      Etapa 2 - Recuo cilindro



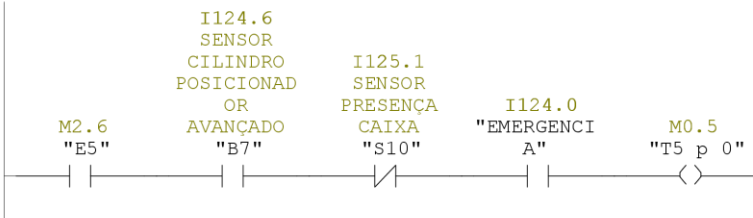
Network: 28      Etapa 3 - Ativa o cilindro



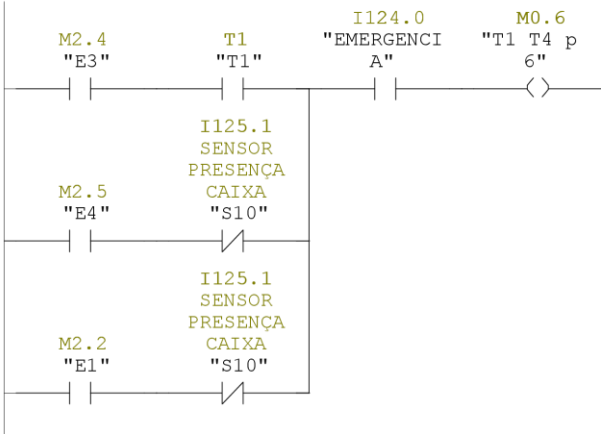
Network: 29      Etapa 4



Network: 30      Etapa 5

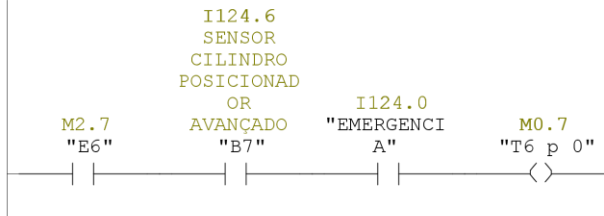


Network: 31      Etapa 6

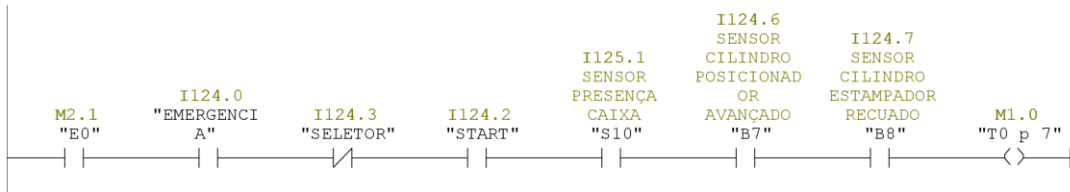




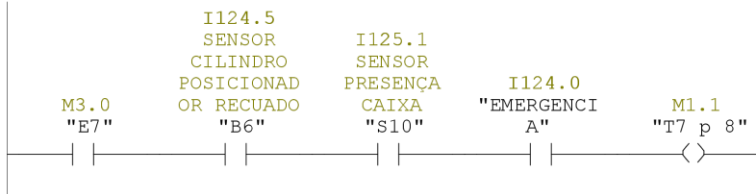
Network: 32      Etapa 6 para 0



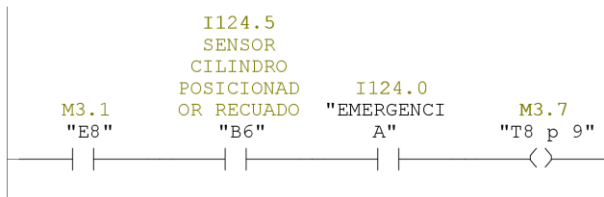
Network: 33      Etapa Zero para 7



Network: 34      Etapa 7



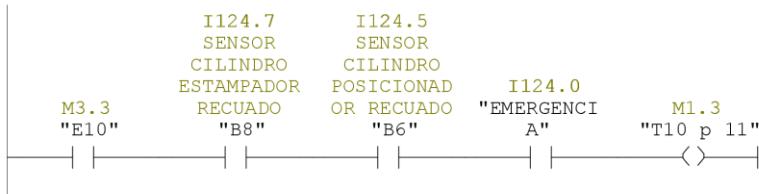
Network: 35



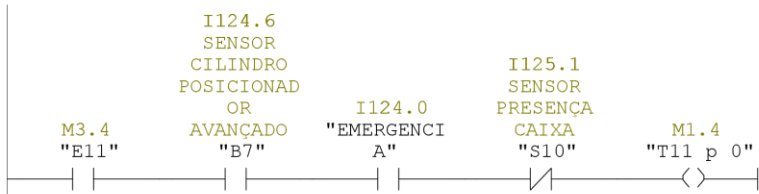
Network: 36



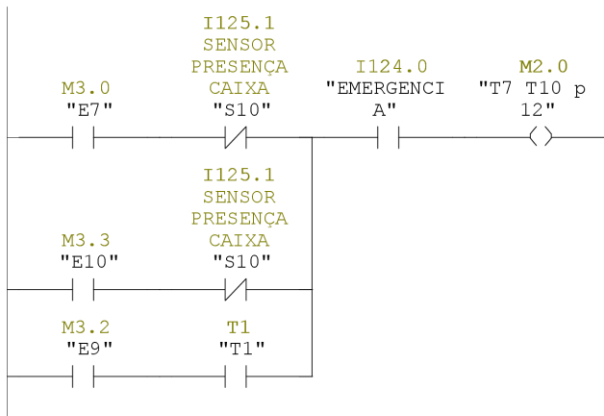
Network: 37



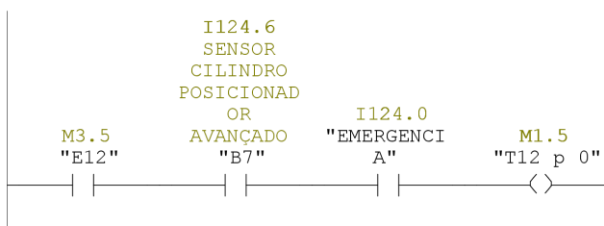
Network: 38



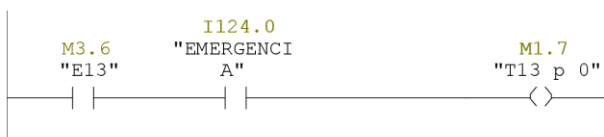
Network: 39



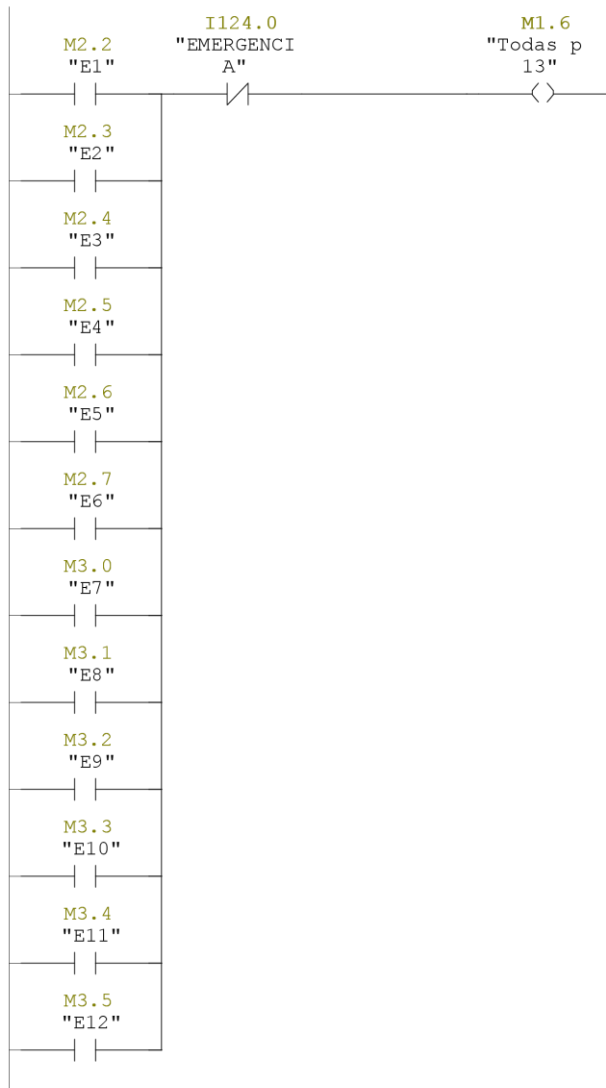
Network: 40



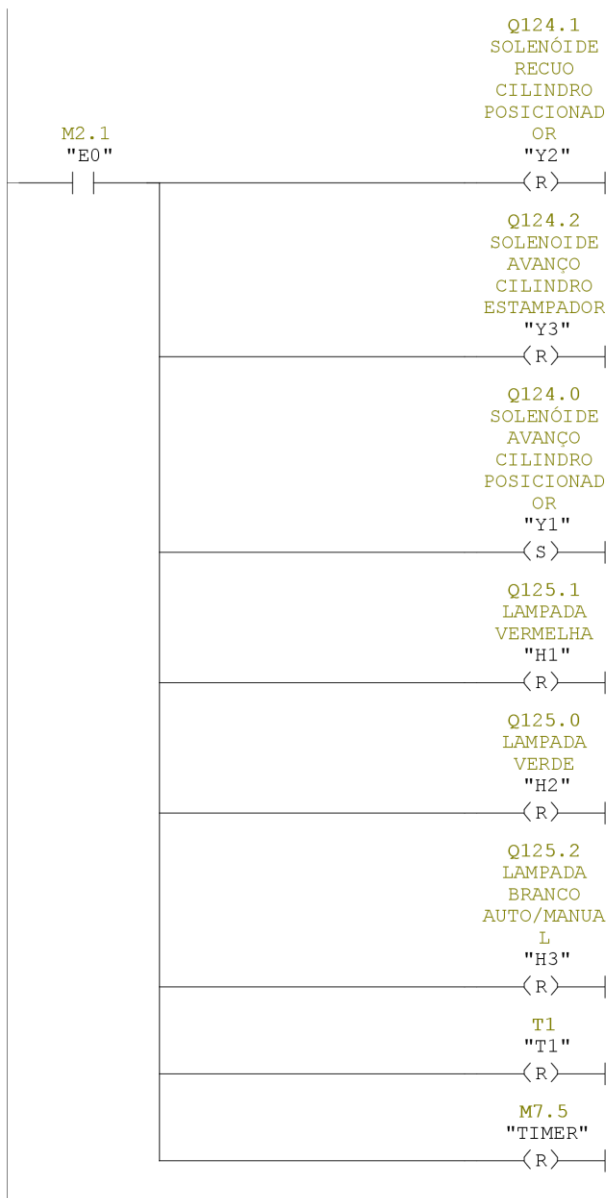
Network: 41



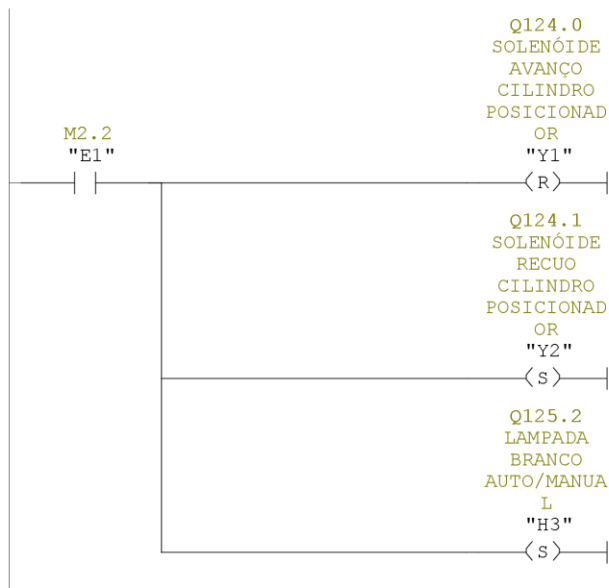
Network: 42



Network: 43      Inicio Etapas



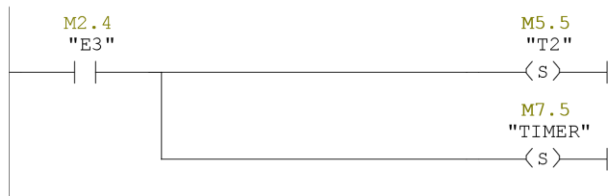
Network: 44 SOLENÓIDE AVANÇO CILINDRO POSICIONADOR



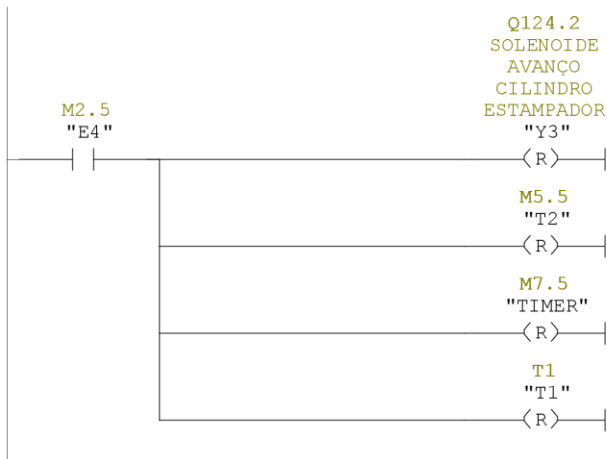
Network: 45 SOLENOIDE AVANÇO CILINDRO ESTAMPADOR



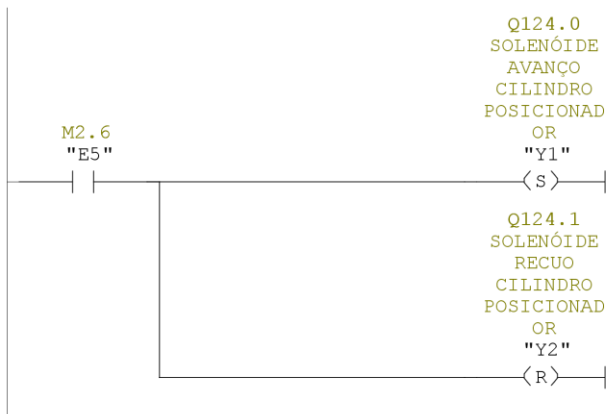
Network: 46 Etapa 3



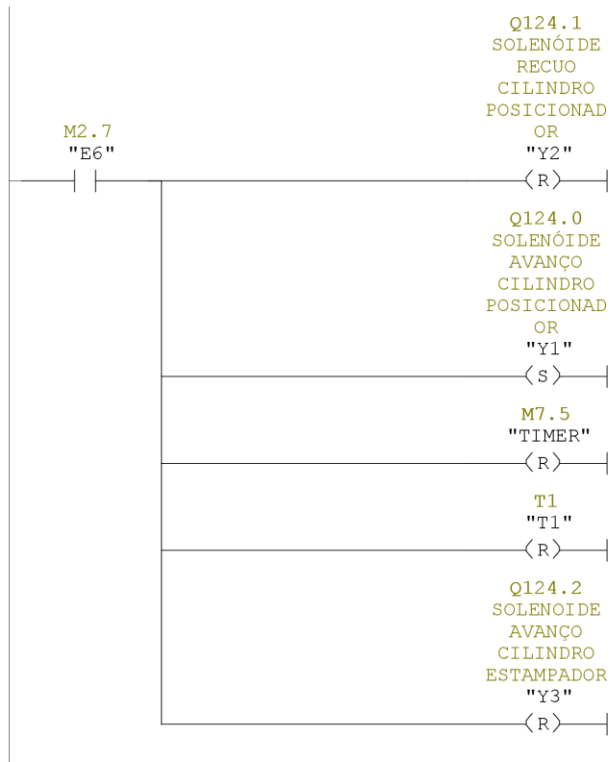
Network: 47 ETAPA 4



Network: 48 SOLENÓIDE AVANÇO CILINDRO POSICIONADOR

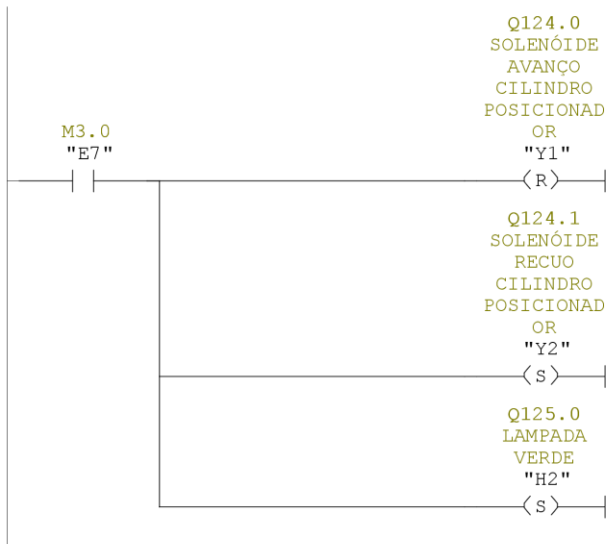


Network: 49 SOLENÓIDE RECUO CILINDRO POSICIONADOR





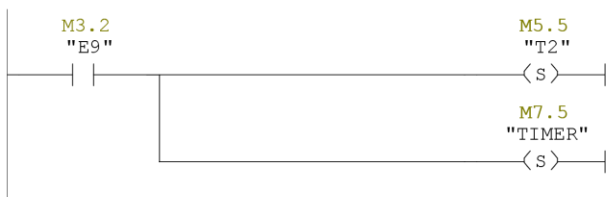
Network: 50 SOLENÓIDE AVANÇO CILINDRO POSICIONADOR



Network: 51



Network: 52

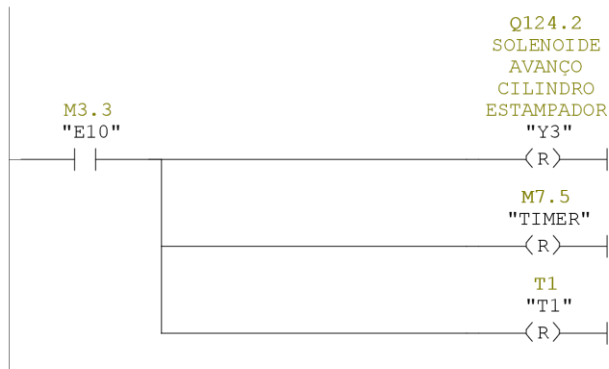


SIMATIC

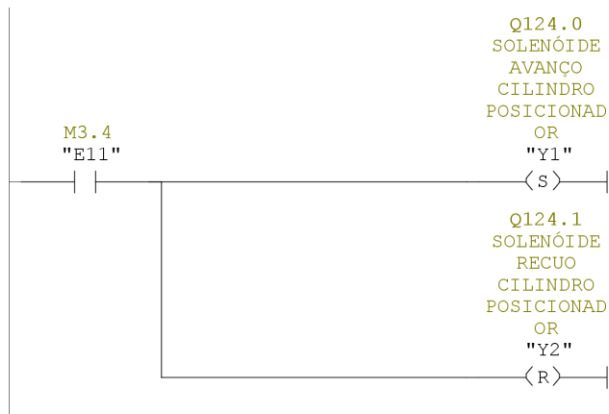
Bruneli2017\  
SIMATIC 300(1)\CPU 314IFM\...\OB1 - <offline>

06/05/2018 05:45:40 PM

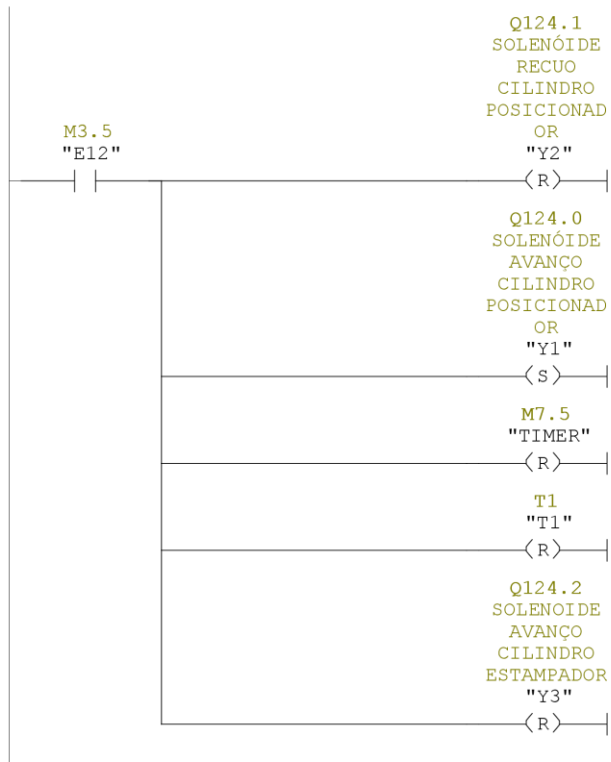
Network: 53 SOLENOIDE AVANÇO CILINDRO ESTAMPADOR



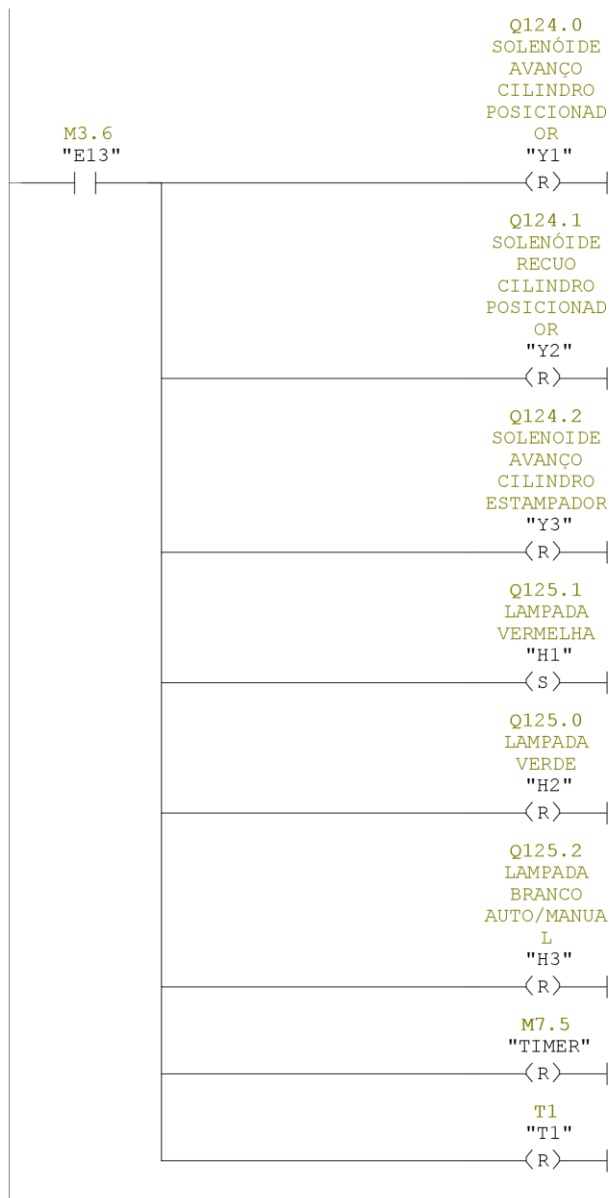
Network: 54 SOLENÓIDE AVANÇO CILINDRO POSICIONADOR



Network: 55	SOLENÓIDE RECUO CILINDRO POSICIONADOR
-------------	---------------------------------------



Network: 56 SOLENÓIDE AVANÇO CILINDRO POSICIONADOR



**APÊNDICE C** - *Symbol Table* do programa do STEP7.

**Properties of symbol table**

Name: Symbols  
 Author:  
 Comment:  
 Created on: 04/26/2018 02:32:05 PM  
 Last modified on: 12/07/2017 08:10:34 PM  
 Last filter criterion: All Symbols  
 Number of symbols: 56/56  
 Last Sorting: Symbol Ascending

Status	Symbol	Address	Data type	Comment
	B6	I 124.5	BOOL	SENSOR CILINDRO POSICIONADOR RECUADO
	B7	I 124.6	BOOL	SENSOR CILINDRO POSICIONADOR AVANÇADO
	B8	I 124.7	BOOL	SENSOR CILINDRO ESTAMPADOR RECUADO
	B9	I 125.0	BOOL	SENSOR CILINDRO ESTAMPADOR AVANÇADO
	E0	M 2.1	BOOL	
	E1	M 2.2	BOOL	
	E10	M 3.3	BOOL	
	E11	M 3.4	BOOL	
	E12	M 3.5	BOOL	
	E13	M 3.6	BOOL	
	E14	M 5.0	BOOL	ETAPA REPETE PISTÃO
	E2	M 2.3	BOOL	
	E3	M 2.4	BOOL	
	E4	M 2.5	BOOL	
	E5	M 2.6	BOOL	
	E6	M 2.7	BOOL	
	E7	M 3.0	BOOL	
	E8	M 3.1	BOOL	
	E9	M 3.2	BOOL	
	EMERGENCIA	I 124.0	BOOL	
	H1	Q 125.1	BOOL	LAMPADA VERMELHA
	H2	Q 125.0	BOOL	LAMPADA VERDE
	H3	Q 125.2	BOOL	LAMPADA BRANCO AUTO/MANUAL
	M_EM	M 6.6	BOOL	
	m10	M 10.0	BOOL	IOT
	pressao	MW 120	WORD	
	RT	M 7.6	BOOL	
	S10	I 125.1	BOOL	SENSOR PRESENÇA CAIXA
	SELETOR	I 124.3	BOOL	
	START	I 124.2	BOOL	
	T0 p 1	M 0.0	BOOL	
	T0 p 7	M 1.0	BOOL	
	T1	T 1	TIMER	
	T1 p 2	M 0.1	BOOL	
	T1 T4 p 6	M 0.6	BOOL	
	T10 p 11	M 1.3	BOOL	
	T11 p 0	M 1.4	BOOL	
	T12 p 0	M 1.5	BOOL	
	T13 p 0	M 1.7	BOOL	
	T2	M 5.5	BOOL	
	T2 p 14	M 5.1	BOOL	

SIMATIC

Bruneli2017\

06/05/2018 05:46:59 PM

SIMATIC 300(1)\CPU 314IFM\S7 Program(1)\Symbols

Status	Symbol	Address	Data type	Comment
	T2 p 3	M 0.2	BOOL	
	T3 p 4	M 0.3	BOOL	
	T4 p 5	M 0.4	BOOL	
	T5 p 0	M 0.5	BOOL	
	T6 p 0	M 0.7	BOOL	
	T7 p 8	M 1.1	BOOL	
	T7 T10 p 12	M 2.0	BOOL	
	T8 p 9	M 3.7	BOOL	
	T9 p 10	M 1.2	BOOL	
	TIMER	M 7.5	BOOL	
	Todas p 13	M 1.6	BOOL	
	U_SENSORA	I 125.6	BOOL	BRAÇO UNIDADE SENSORA
	Y1	Q 124.0	BOOL	SOLENOIDE AVANÇO CILINDRO POSICIONADOR
	Y2	Q 124.1	BOOL	SOLENOIDE RECUO CILINDRO POSICIONADOR
	Y3	Q 124.2	BOOL	SOLENOIDE AVANÇO CILINDRO ESTAMPADOR