

DAS Departamento de Automação e Sistemas
CTC **Centro Tecnológico**
UFSC Universidade Federal de Santa Catarina

Sistema de Monitoramento da Produção

*Relatório submetido à Universidade Federal de Santa Catarina
como requisito para a aprovação na disciplina
DAS 5511: Projeto de Fim de Curso*

Erick Massanori Yoshida

Florianópolis, fevereiro de 2014

Sistema de Monitoramento da Produção

Erick Massanori Yoshida

Esta monografia foi julgada no contexto da disciplina
DAS5511: Projeto de Fim de Curso
e aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação

Marcelo Ricardo Stemmer

Assinatura do Orientador

Banca Examinadora:

Arno Bollmann
Orientador na Empresa

Prof. Marcelo Ricardo Stemmer
Orientador no Curso

<nome do orientador na empresa/instituto>
Orientador na Empresa

Prof. <nome do professor avaliador>
Avaliador

<nome aluno 1>
<nome aluno 2>
Debatedores

Agradecimentos

Inicialmente, agradeço aos amigos e colegas de trabalho do grupo NEO Empresarial e nosso tutor, Prof. Dr. Carlos Alberto Schneider, que me acompanharam nos trabalhos durante o semestre e durante boa parte de minha preparação profissional durante a graduação.

Aos orientadores Prof. Dr. Marcelo Ricardo Stemmer e Dr. Arno Bollmann, ao supervisor Leandro Carioni e ao coordenador de estágios do departamento, Prof. Dr. Ricardo José Rabelo, por toda a ajuda para tornar esse projeto possível.

À minha família, que durante esses anos de graduação, fez todo o esforço possível para que eu pudesse focar meus esforços na graduação e em minha formação profissional.

Acrescento também o meu muito obrigado aos mestres do Departamento de Automação e Sistemas pelo excelente trabalho desenvolvido ao longo dos anos, por toda a dedicação na formação de engenheiros competentes e na condução de um dos melhores cursos de engenharia do país.

Resumo

No presente documento é apresentado um relatório técnico de desenvolvimento de um projeto elaborado pelo grupo NEO Empresarial para a empresa WEG S.A., unidade de Motores, localizada em Jaraguá do Sul/SC. O projeto consistiu em um sistema de monitoramento da produção de uma das linhas produtivas de motores da WEG.

A partir do sinal elétrico enviado por 12 diferentes centros de processamento de peças da Fábrica VI da WEG, estruturou-se um sistema para que se fizesse a contagem de ciclos de processamento e o monitoramento das máquinas.

A primeira parte do projeto consistiu na escolha do *hardware* adequado para o trabalho, garantindo a robustez de um sistema que operaria 24 horas por dia, além do correto entendimento do funcionamento do protocolo de comunicação do *hardware* com o futuro software a ser elaborado.

A segunda e mais extensa parte do trabalho foi o desenvolvimento do software, desde a especificação conceitual, passando pelo desenvolvimento em si, até correta documentação para fins de manutenção e utilização.

Além disso, acrescentou-se um trabalho complementar a esse projeto. A tarefa consistia em estudos e testes para se elaborar um método que realizaria a contagem oficial de peças produzidas nessa linha de produção.

Abstract

In the following document it is presented a report of the development of a project designed by the group NEO Empresarial for WEG S.A., unit of Motors, located in Jaraguá do Sul/SC. The project consisted in a Production Monitoring System for one of the assembly lines of WEG.

Through the electrical signals sent by 12 different processing parts centres of the Manufacturing Area VI at WEG, it was structured system that would count the processing cycles and monitor the machines.

The first part of this project consisted in choosing the adequate hardware for the task, providing correct robustness for a system which should operate 24 hours per day. Besides it was understood how the communication protocol of the *hardware* would work with the future *software*.

The second and longest part of the work was the development of the *software*, from the conceptual development, through the development itself, to the correct documentation for the maintenance and the usage.

Moreover, a complementary work was added to this project. The task consisted in studies and tests to create a method, which would make the official count of the parts produced in this assembly line.

Sumário

Agradecimentos.....	4
Resumo	5
Abstract	6
Sumário	7
Capítulo 1: Introdução	10
Capítulo 2: Objetivos	11
2.1. Definição do Problema.....	11
2.2. Solução conceitual	13
2.3. Estruturação do Documento.....	14
Capítulo 3: Hardware.....	15
3.1. Seleção do <i>Hardware</i>	15
3.1.1. CLP WEG Clic 02.....	15
3.1.2. RS-232	16
3.1.3. RS-485.....	17
3.1.4. Conversor RS-232/RS-485	18
3.2. Protocolo Modbus	19
Capítulo 4: Software	20
4.1. Linguagem de programação utilizada	20
4.1.1. O que é LabVIEW?.....	20
4.1.2. Por que LabVIEW?	21
4.2. Visão geral do Software	22
4.2.1. Padrão de <i>design</i>	22
4.2.2. Comunicação entre loops.....	24

4.2.3. Interface	24
4.2.3.3. Aba Relatórios.....	27
4.2.3.4. Aba Configuração.....	28
4.2.4. Vetor Peças.....	28
4.2.5. Outputs.....	29
4.2.5.1. Log1	29
4.2.5.2. Log2	29
4.2.5.3. Arquivo de configuração.....	30
4.3. Loop Mestre	30
4.3.1. Função Timeout	31
4.3.2. Atualização das comboboxes de data para consulta	31
4.3.3. Iniciar consulta	31
4.3.4. Iniciar aquisição.....	32
4.3.5. Parar aquisição	32
4.3.6. Atualização dos comboboxes de data para relatórios	32
4.3.7. Iniciar geração de relatórios	32
4.3.8. Salvar modificações em arquivo de configuração	32
4.4. Loop Aquisição.....	32
4.4.1. Comunicação MODBUS.....	32
4.4.2. Exclusão de interferências	34
4.4.3. Geração de arquivo log1	35
4.5. Loop Processamento	36
4.5.1. Separação de bits	36
4.5.2. Verificação de intervalo de tempo	37
4.5.3. Geração do arquivo log2	37
4.5.4. Projeção gráfica	38

4.6.	Loop Consulta	38
4.6.1.	Localização do arquivo.....	38
4.6.2.	Projeção Gráfica.....	39
4.7.	Loop Relatório.....	39
4.7.1.	Localização do arquivo.....	39
4.7.2.	Geração de relatório.....	40
Capítulo 5: Trabalho complementar: Sistema de contagem de peças		41
5.1.	Solução conceitual	41
5.1.1.	Código de barras unidimensional	42
5.2.	Testes	42
5.2.1.	Primeira etapa	43
5.2.2.	Segunda etapa.....	43
5.2.3.	Análise de possível perda de produtividade.....	44
5.3.	Software extra: Contagem de peças	45
Capítulo 6: Indicadores de processo		47
6.1.	Lead Time de Produção	47
6.2.	Overall Equipment Effectiveness (OEE)	47
Capítulo 7: Conclusões e Perspectivas		50
Capítulo 8: Bibliografia.....		53

Capítulo 1: Introdução

“Se você não pode medir, você não pode gerenciar” é uma das frases mais clássicas já utilizadas pelo famoso guru da administração Peter Drucker. Sua realidade, entretanto, não vale apenas para o mundo de negócios e da administração, mas também é válido para muitas questões na engenharia, dentro da produção, dentro das fábricas.

Ainda que o conceito de sistema integrado de manufatura, na qual a medição e análise de indicadores de produção são tão importantes quanto o potencial produtivo, tenha se popularizado muito nas últimas décadas, os investimentos – mesmo nas grandes empresas – ainda são quase totalmente focado na produtividade, ignorando os métodos de monitoramento, de aquisição e verificação de dados.

O projeto aqui descrito teve origem com uma necessidade bastante básica dos chefes de produção de uma das linhas da WEG Motores: “Como podemos elaborar um método para melhor acompanhar a nossa produção ao final do expediente?”, “Podemos confiar apenas na contagem feita pelos operadores?”, “Será que podemos ter uma acompanhamento em tempo-real do funcionamento dos nossos centros de produção?”

Assim surgiu o Sistema de Monitoramento da Produção. Um projeto inicialmente simples, mas cujos resultados tem potencial para expandir e mostrar aos envolvidos como o monitoramento de um processo produtivo pode levar a um gerenciamento mais adequado e, conseqüentemente, a resultados mais expressivos da produção.

Capítulo 2: Objetivos

A produção da Fábrica VI da WEG é monitorada manualmente. Diariamente, os operadores de cada máquina da linha de produção anotam em um quadro o número de motores que foram processados a cada hora do turno. Este procedimento não é confiável pois é suscetível a erros de contagem do operador. Além disso, não existe um acompanhamento em tempo-real do funcionamento das máquinas. Ou seja, para que seja feito o monitoramento, é necessária a presença física do gerente para essa verificação. Por fim, os relatórios de produtividade são feitos de forma pouco prática: diariamente o gerente de produção compila as anotações dos operadores em uma planilha.

2.1. Definição do Problema

A proposta feita pela WEG para solucionar este problema foi de melhorar este monitoramento automatizando-o em parte da fábrica. Há três tipos de máquinas onde o sistema será implantado, totalizando doze máquinas a serem monitoradas. Estas máquinas pertencem aos seguintes centros: centros de inserção (Figura 1), linhas de bandagem (Figura 2) e estufas de impregnação (Figura 3).



Figura 1 - Centro de Inserção



Figura 2 - Linha de bandagem



Figura 3 - Estufa de impregnação

Foi solicitado então um sistema que pudesse, por meio da aquisição de dados advindos dos três tipos de equipamentos, fazer as seguintes funções:

1. Fazer a contagem de ciclos de processamentos de peças feitos em cada uma das máquinas;
2. Exibir ao gerente de produção um acompanhamento em tempo-real das máquinas;
3. Gerar relatórios de produção a partir de datas selecionadas pelo gerente.

2.2. Solução conceitual

Com o problema exposto, pensou-se então no seguinte conceito de solução, exposto na figura:

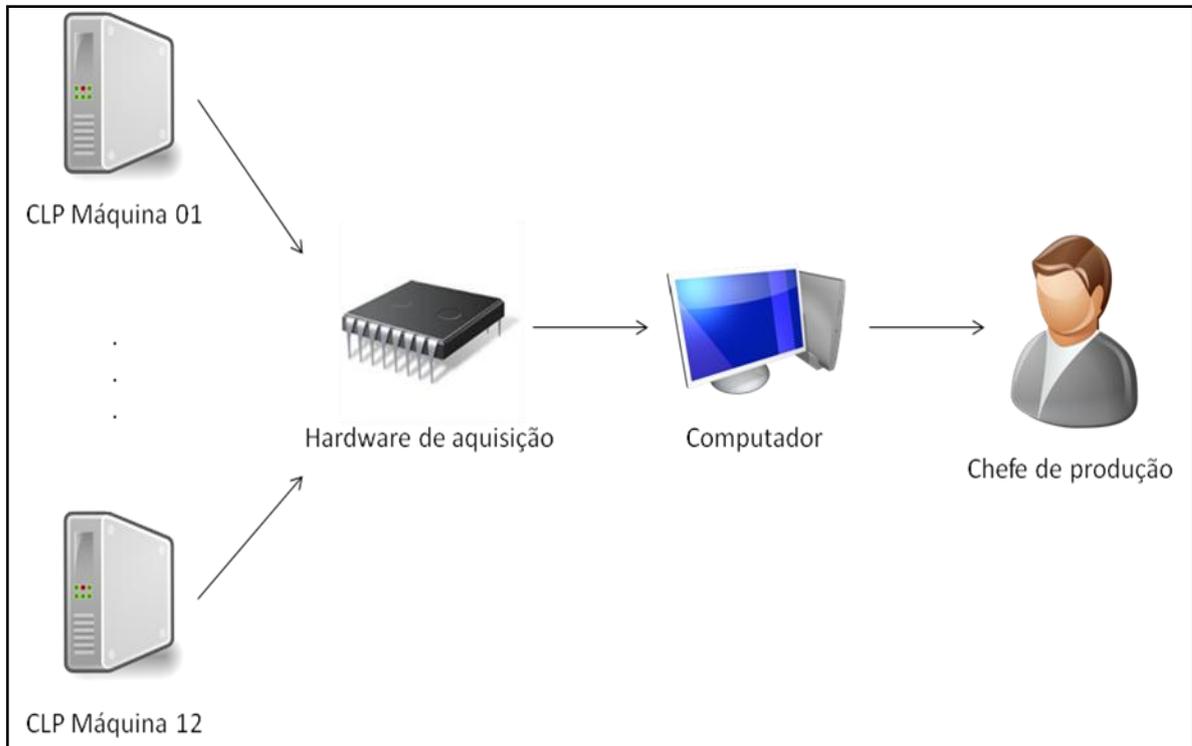


Figura 4 – Solução conceitual

Os Controladores Lógico Programáveis (CLPs) das diferentes máquinas enviariam os dados para um *hardware* de aquisição que, por sua vez, coletaria essas informações e as passaria a um computador, responsável por fazer o processamento das informações, para monitoramento por parte do responsável pela produção e para geração dos relatórios citados.

O projeto consistiria então de, no mínimo, duas fases. A primeira de especificação e adequada instalação do *hardware* e a segunda, de especificação e desenvolvimento do *software* a ser instalado no computador.

2.3. Estruturação do Documento

O documento é então dividido nos seguintes capítulos:

- Capítulo 1: Breve introdução sobre o trabalho apresentado.
- Capítulo 2: A definição do problema encontrado, uma síntese da solução conceitual a ser relatada.
- Capítulo 3: Definição do *hardware* a ser utilizado, bem como os instrumentos adicionais.
- Capítulo 4: Apresentação do *software* desenvolvido.
- Capítulo 5: Detalhamento de um trabalho complementar realizado, o Sistema de contagem de peças.
- Capítulo 6: Indicadores de produtividade obtidos a partir do projeto realizado.
- Capítulo 7: Conclusão, resumizando os resultados do trabalho apresentado e uma visão crítica do que foi desenvolvido.
- Capítulo 8: Bibliografia.

Capítulo 3: Hardware

3.1. Seleção do *Hardware*

3.1.1. CLP WEG Clic 02

Foi escolhido um CLP Clic 02 da WEG para que se estabelecesse a comunicação entre o computador onde ocorreria o processamento de dados e os CLPs das máquinas utilizadas.

Anteriormente, cogitou-se a utilização de um microcontrolador PIC para essa função, entretanto, escolheu-se o CLP principalmente por sua maior robustez e confiabilidade, haja visto o fato que o sistema iria dentro de um ambiente industrial 24h por dia.

Quanto ao modelo do CLP, optou-se por um modelo da própria WEG, por ser um equipamento já conhecido por parte da equipe envolvida no trabalho, que poderia se conseguir por um custo menor, haja visto que é fabricado internamente.



Figura 5 – CLP WEG Clic 02

Abaixo, uma ilustração do hardware do sistema com suas respectivas ligações (Figura 6).

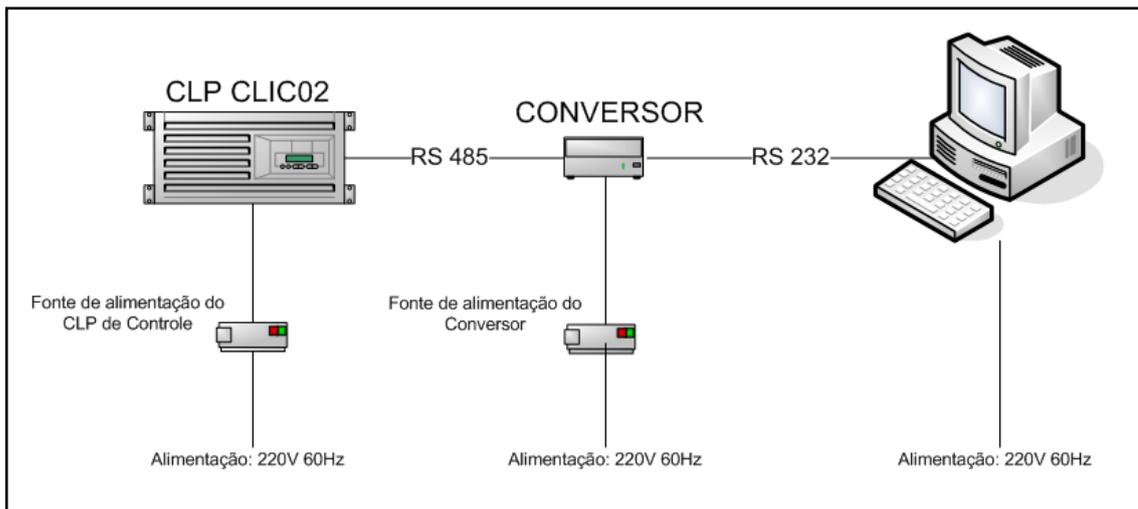


Figura 6 – Visão geral de Hardware

A comunicação entre o CLP e o PC é feita via Protocolo Modbus, sendo necessário um conversor entre eles, pois a saída do CLP é do tipo RS-485 e a do PC RS-232.

Estes conceitos de comunicação e tipos de ligações serão explicados a seguir.

3.1.2. RS-232

RS-232 (também conhecido por [EIA RS-232C](#) ou V.24) é um padrão para troca serial de dados binários entre um DTE (terminal de dados, de Data Terminal Equipment) e um DCE (comunicador de dados, de Data Communication Equipment). É comumente usado nas portas seriais dos PCs.



Figura 7 - RS-232

No protocolo de comunicação RS-232, caracteres são enviados um a um como um conjunto de bits. A codificação mais comumente utilizada (e também utilizada neste projeto) é o "start-stop assíncrono" que usa um bit de início, seguido por sete ou oito bits de dados, possivelmente um bit de paridade, e um ou dois bits de parada sendo, então, necessários 10

bits para enviar um único caractere. O padrão define os níveis elétricos correspondentes aos níveis lógicos um e zero, a velocidade de transmissão padrão e os tipos de conectores.

O RS-232 é recomendado para conexões curtas (quinze metros ou menos), valores próximos de zero não são sinais válidos. O nível lógico um é definido como tensão negativa e o nível lógico zero tensão positiva.

3.1.3. RS-485

A comunicação RS-485 funciona em modo diferencial, ou seja, a diferença entre as tensões na linha indicam se o mestre está transmitindo 1 ou 0. Suporta comunicação *half-duplex* e *full-duplex* sendo que para a primeira há necessidade da utilização de um cabo par trançado e para a segunda dois pares de cabo.

Este tipo de comunicação alcança grandes distâncias, podendo chegar até 1200 m de cabo.

O cabo de comunicação RS-485 é composto de dois fios, sendo estes comumente denominados A e B. A tabela abaixo contém os estados lógicos dos terminais A e B de acordo com o dado que o transmissor deseja enviar.

DRIVER			
INPUT	ENABLE	OUTPUTS	
D	DE	A	B
H	H	H	L
L	H	L	H
X	L	Z	Z

Figura 8 - Estados lógicos na comunicação RS-485

Quando o transmissor (INPUT D) está em alto (H), a linha A fica mais positiva que a B e o inverso ocorre quando o estado inverte. Há também uma linha de controle chamada DE, que quando está em nível lógico baixo (L), o barramento tem alta impedância.

Para que o receptor identifique um sinal válido, a diferença entre os terminais A e B deve ser maior que 200 mV. Entre -200 mV e 200 mV o sinal é indefinido.

Abaixo, um gráfico que mostra a transmissão do conteúdo binário 01001 a uma taxa 9600 bps no barramento RS-485.

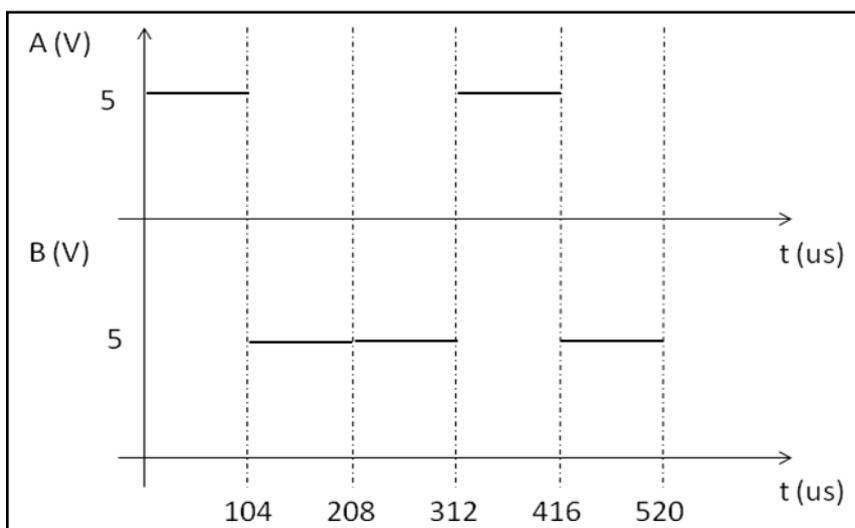


Figura 9 – Exemplo de envio por comunicação RS-485, mensagem: 01001

3.1.4. Conversor RS-232/RS-485

Para transformar o sinal do CLP (RS-485) em um sinal serial para a comunicação com o computador (RS-232), foi utilizado um conversor RS-232/RS-48, modelo IF-3 da Pró-Digital.



Figura 10 - Conversor RS-232/RS-485 da Pró-Digital

Algumas características do conversor:

- Identificação dos sinais através de três *leds* indicativos na parte superior: conversor ligado (ON), tráfego do sinal TX (transmitindo) e tráfego do sinal RX (recebendo).
- Conectores: linha RS-485 com conector com parafusos para a ligação dos fios, linha RS-232 com conector DB9 (serial).
- Funciona em modo assíncrono, bidirecional (half-duplex).

3.2. Protocolo Modbus

O Protocolo Modbus é um dos protocolos de comunicação de mais larga utilização em automação industrial pela sua simplicidade e facilidade de implementação.

Este protocolo é baseado em um modelo de comunicação mestre-escravo (Figura 5) onde um único dispositivo, o mestre, pode iniciar transações denominadas *queries*. Os demais dispositivos da rede (escravos) respondem, suprimindo os dados requisitados pelo mestre ou executando uma ação por ele comandada. Geralmente, o mestre é um sistema supervisor. Os papéis de mestre e escravo são fixos quando se utiliza comunicação serial.

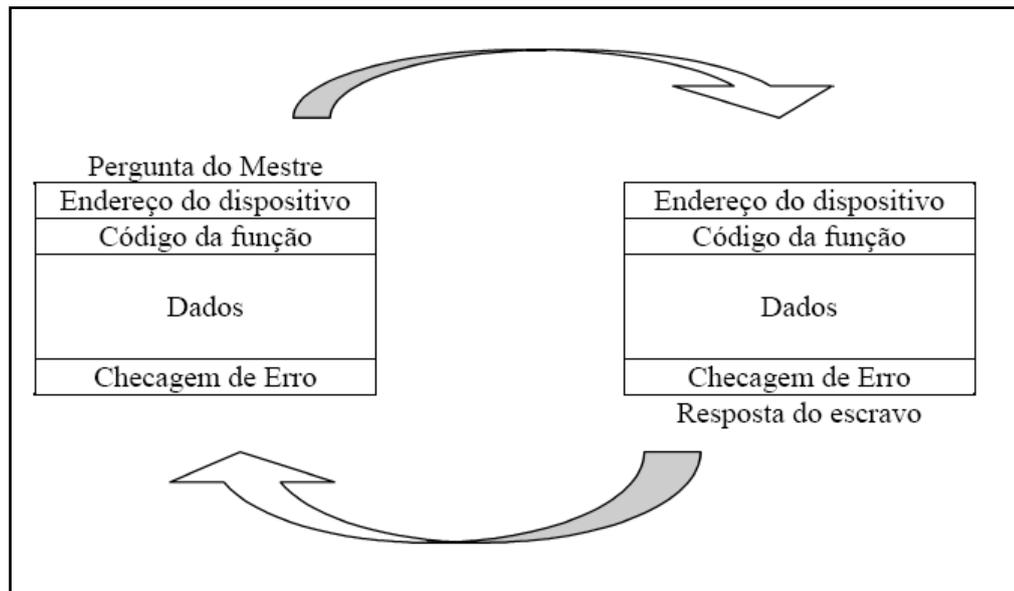


Figura 11 - Modbus

Existem dois modos de transmissão de dados:

ASCII

Cada *byte* de mensagem é enviado como dois caracteres ASCII. Durante a transmissão, intervalos de até um segundo entre caracteres são permitidos sem que a mensagem seja truncada. Algumas implementações fazem uso de tais intervalos de silêncio como delimitadores de fim de mensagem.

RTU

Cada byte de mensagem é enviado como um byte de dados. A mensagem deve ser transmitida de maneira contínua, já que pausas maiores que 1,5 caractere provocam truncamento. Este é o modo de transmissão utilizado neste projeto.

Capítulo 4: Software

4.1. Linguagem de programação utilizada

Para o desenvolvimento do projeto, foi escolhido o ambiente de desenvolvimento LabVIEW, originária da National Instruments.

4.1.1. O que é LabVIEW?

Como definido pelos desenvolvedores da National Instruments: “LabVIEW é um ambiente de desenvolvimento altamente produtivo, para a criação de aplicações customizadas e que interagem com os dados ou sinais do mundo real, em áreas como ciência e engenharia.” (1)

O ambiente de desenvolvimento consiste em duas janelas principais: o painel frontal, na qual se constrói a interface gráfica do programa sem a necessidade de escrever qualquer linha de código, e o diagrama de blocos (cujo nome advém dos tradicionais diagramas de blocos utilizados nas mais diversas áreas da engenharia), na qual o fluxo de dados é programado por meio de linhas, funções primitivas e sub-VIs (*Virtual Instruments* ou instrumentos virtuais), que nada mais são que sub-programas que podem ser executados isoladamente.

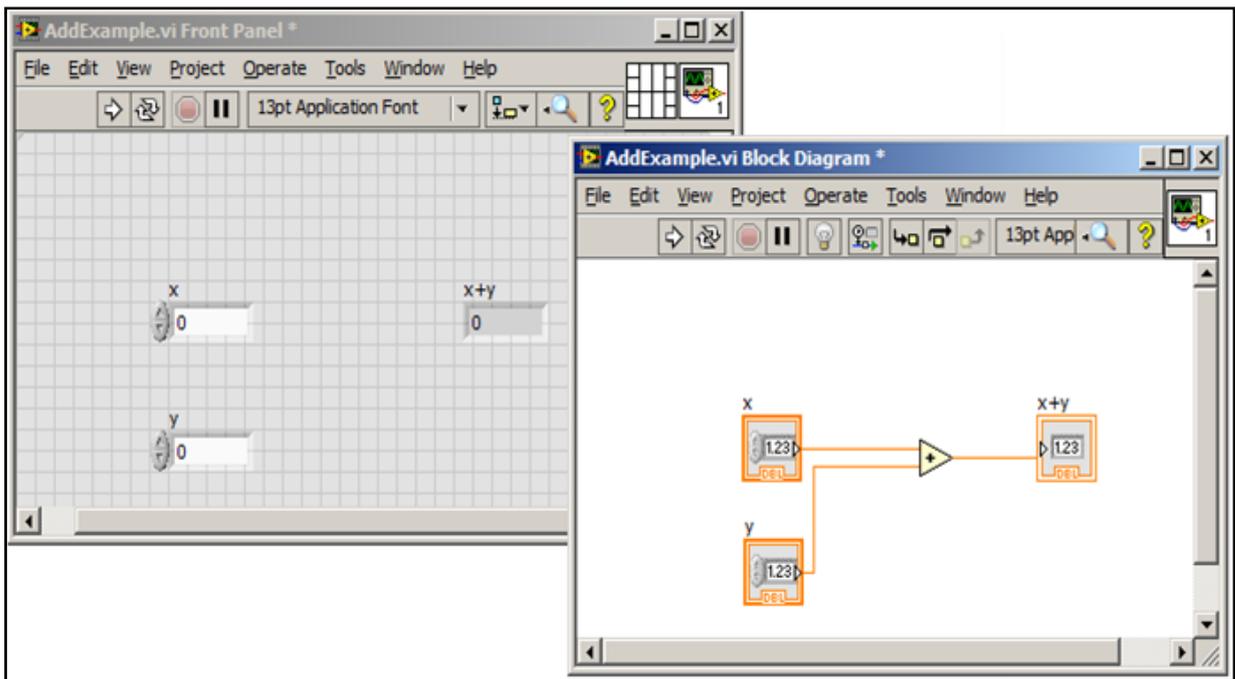


Figura 12 - Exemplo de ambiente de desenvolvimento LabVIEW. Painel Frontal ao lado direito e Diagrama de blocos ao lado esquerdo.

4.1.2. Por que LabVIEW?

As razões que levaram a equipe envolvida a optar pela linguagem LabVIEW foram:

1. Fácil interação com *hardware*: por ser um ambiente de programação voltado para um desenvolvimento simples e de fácil interação com o mundo real, as ferramentas apresentadas tornam mais fácil o processo de integração com o *hardware*. Assim, os protocolos de mensagem em baixo nível são dispensados, podendo o desenvolvedor focar em outras questões do desenvolvimento do programa.
2. Fácil desenvolvimento de interface: A simplicidade para o desenvolvimento da interface que, como previamente citado, não precisa da utilização de linha de códigos para criação foi outro fator de peso para a escolha desse sistema de desenvolvimento.

3. Familiaridade por parte da equipe de desenvolvimento e de outros envolvidos no projeto: por fim, o prévio entendimento do ambiente LabVIEW por parte do desenvolvedor, bem como a prévia utilização por parte de diferentes setores da WEG Motores (incluindo o setor de Metrologia que possivelmente se responsabilizaria pela manutenção do sistema, caso isso fosse necessário), foram um dos maiores fatores que contribuíram para essa decisão.

4.2. Visão geral do Software

4.2.1. Padrão de *design*

O *software* foi desenvolvido a partir da arquitetura *master-slave*, frequentemente utilizada quando múltiplos processos ocorrem simultaneamente (2). Desse modo, permite-se que a aquisição e processamento de dados que ocorrem durante todo o tempo de execução permanecessem ativos, quando o usuário utilizasse outra função do programa (consulta, por exemplo).

Assim, o programa ficou dividido em cinco *loops* (ou *threads*): um *loop* mestre e quatro escravos. Uma visão geral da estruturação pode ser vista na figura a seguir:

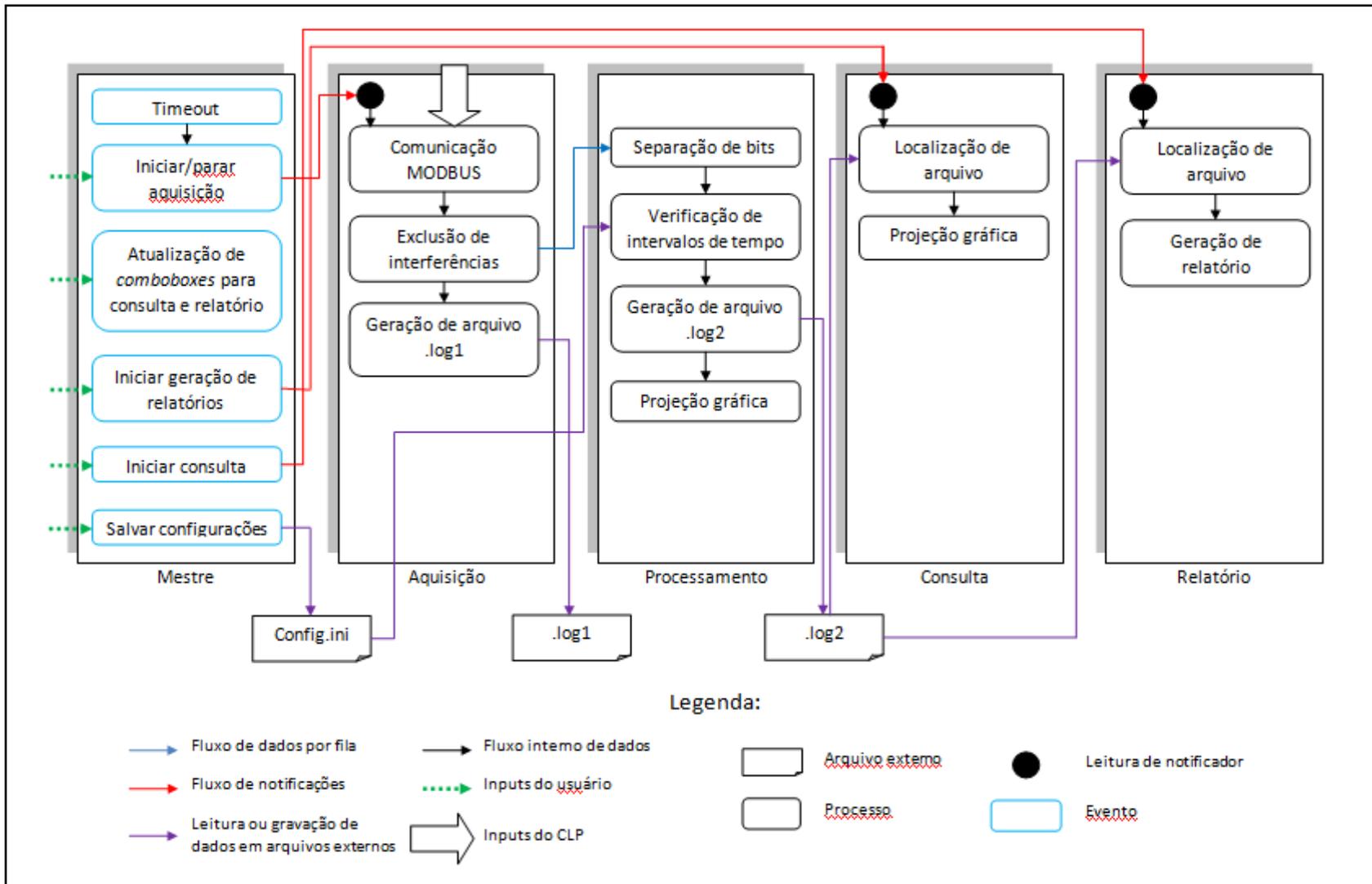


Figura 13 – Visão geral do software

4.2.2. Comunicação entre loops

A comunicação entre os *loops* foi estabelecida através dos seguintes recursos:

4.2.2.1. Notificadores

Notificadores são basicamente uma ferramenta utilizada para que duas partes independentes de um mesmo ambiente possam se comunicar. Funcionam como um correio, um processo envia o notificador e outro recebe. **(3)**

Neste projeto, notificadores são utilizados para que os *loops* escravos recebam as instruções para execução de comando enviadas a partir do *loop* mestre.

4.2.2.2. Filas

Filas mantém ordenação First In First Out (FIFO¹) dos itens. São úteis em situação do tipo “produtor/consumidor”, onde uma porção do código gera as entradas que serão utilizadas em outra porção do código. Uma de suas vantagens é que a taxa de produção e consumo não precisam ser idênticas. (4)

No programa, esse recurso foi utilizado na passagem de dados entre *loop* de aquisição (produtor) e *loop* de processamento (consumidor).

4.2.3. Interface

Nesta seção, apresenta-se a interface do programa, juntamente com uma breve explicação dos elementos presentes.

¹ FIFO: Acrônimo para First In, First Out. Estruturas nas quais os elementos enfileirados, vão sendo utilizados e retirados de acordo com sua ordem de chegada.

A interface com o usuário é dividida em quatro abas:

1. Monitoramento: na qual se acompanha a produção em tempo-real.
2. Consulta: utilizada para se verificar como foi a produção de peças em uma data passada.
3. Relatórios: Na qual se geram relatórios em excel.
4. Configurações: na qual se modificam alguns parâmetros utilizados para o processamento de dados.

4.2.3.1. Aba Monitoramento

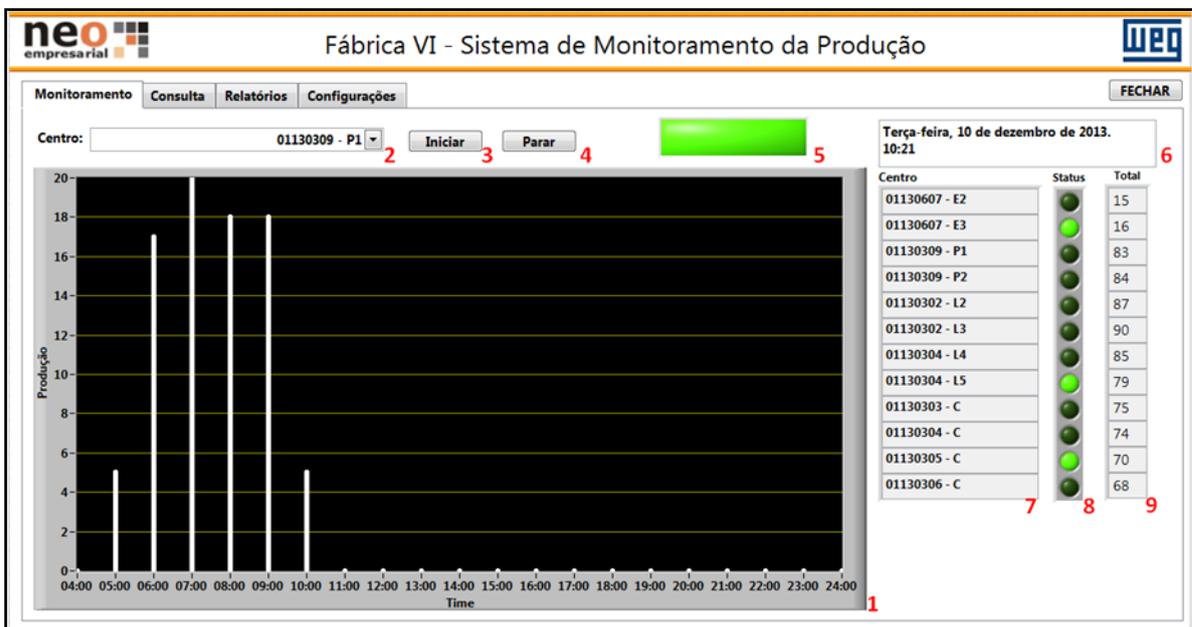


Figura 14 – Aba Monitoramento

A descrição de cada um dos elementos acima enumerados é vista a seguir:

1. Gráfico: mostra a produtividade de um centro ao longo do dia
2. Centro: *combobox* utilizada para se escolher qual centro precisa ser visualizado no gráfico
3. Iniciar: botão que inicia a aquisição de dados do CLP
4. Parar: botão que interrompe a aquisição de dados do CLP
5. Status: indicador *on/off* que aponta se a aquisição de dados está ocorrendo ou não

6. Data e hora: indicador de data e hora atual
7. Lista de centros: Lista com o nome de todos os centros monitorados
8. Lista de status: aponta quais centros estão em atividade
9. Lista total: aponta a quantidade de peças processadas por cada um dos centros.

4.2.3.2. Aba Consulta

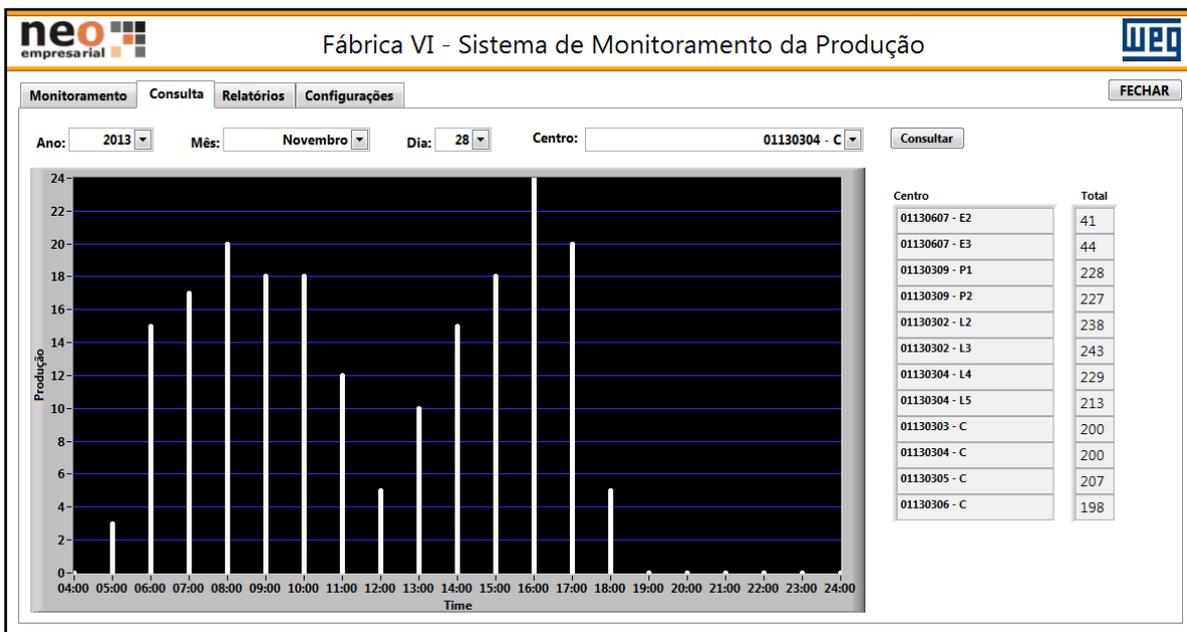


Figura 15 – Aba Consulta

A aba consulta se assemelha bastante à aba Monitoramento. A figura central é a do gráfico que mostra a produtividade de cada máquina durante todo o dia.

Além da *combobox* “Centro”, estão as de “ano”, “mês” e “dia”, que permitem ao usuário escolher a data da qual ele deseja ver a produtividade, e o botão “Consultar” que dá início à pesquisa.

4.2.3.3. Aba Relatórios

The screenshot displays the 'Relatórios' (Reports) tab within the 'Fábrica VI - Sistema de Monitoramento da Produção' application. The interface includes a header with the 'neo empresarial' logo on the left, the system title 'Fábrica VI - Sistema de Monitoramento da Produção' in the center, and the 'WEG' logo on the right. Below the header is a navigation bar with tabs for 'Monitoramento', 'Consulta', 'Relatórios', and 'Configurações', along with a 'FECHAR' (Close) button. The main content area features three dropdown menus for selecting the report date: 'Ano' (Year) is set to 2013, 'Mês' (Month) is set to Novembro (November), and 'Dia' (Day) is set to 28. A 'Gerar Relatório' (Generate Report) button is positioned below these dropdowns.

Figura 16 – Aba Relatórios

Aba extremamente simples também, na qual o usuário apenas seleciona valores nas *comboboxes* para que seja gerado um relatório de produtividade de um determinado dia.

4.2.3.4. Aba Configuração

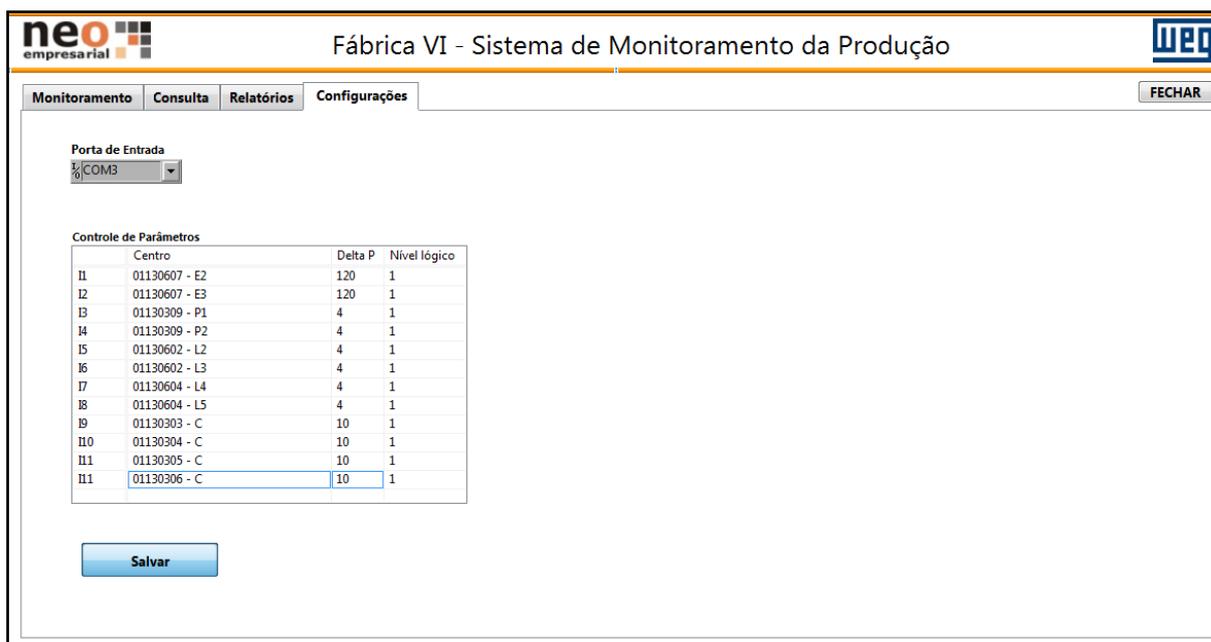


Figura 17 – Aba Configuração

Outra aba bastante simples. Contém uma tabela na qual deve ser preenchida com o nome dos centros, o tempo mínimo de operação de cada um (ver 4.5.2) e o nível lógico de operação (isto é, se o centro, durante o funcionamento, passa ao CLP um sinal de nível lógico alto ou baixo).

Acima da tabela, o usuário escolhe a porta de entrada no computador para a comunicação com o CLP.

4.2.4. Vetor Peças

A principal variável do programa. Nela, contabiliza-se todas as peças processadas durante o período de um dia. É composta por duas dimensões:

- 1ª dimensão: vetor de tamanho 12, representando cada uma das máquinas com a qual o programa trabalha.
- 2ª dimensão: vetor de tamanho 21, que representa as 21h do dia em que o programa se encontra em atividade.

Assim, o valor $v[2][5]$, por exemplo, representa a quantidade de peças produzidas pela máquina 3 no sexto horário de funcionamento do programa (9h da manhã, no caso)².

4.2.5. Outputs

4.2.5.1. Log1

O arquivo log1, salvo durante a execução do *loop* Aquisição, contém um histórico completo dos sinais adquiridos pelo programa ao longo de um dia. Por este fato, apesar de o arquivo tipo log1 não ser utilizado pelo usuário em qualquer momento, ele é essencial para desenvolvedores, caso estes precisem analisar o comportamento do programa a partir do sinal de entrada do CLP.

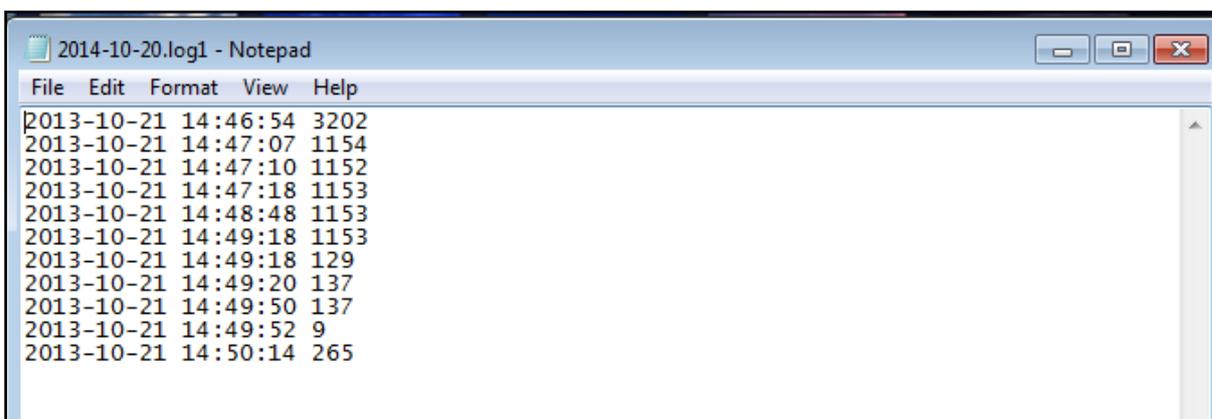


Figura 18 – Exemplo de arquivo log1

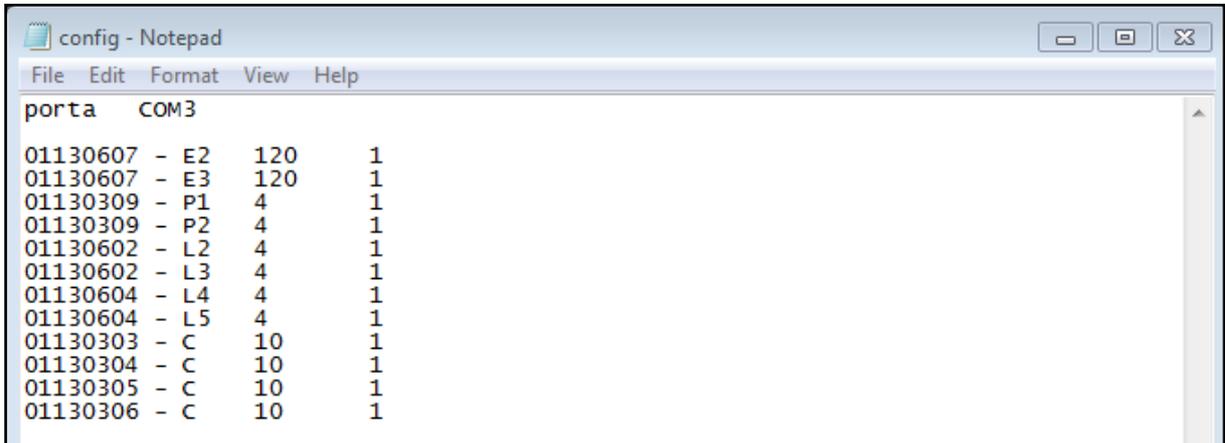
4.2.5.2. Log2

O arquivo log2 é simplesmente o vetor peças (ver 4.2.4) de um dia de produção salvo em um arquivo ASCII para posterior utilização na função de consulta do programa.

² Obs.: Como vetores de programação iniciam no valor “0”, no exemplo, 2 equivale à terceira máquina e 5 ao sexto horário.

4.2.5.3. Arquivo de configuração

O arquivo de configuração config.ini, contém as informações básicas de configuração do programa. As mesmas que podem ser vistas na aba de Configurações.



```
config - Notepad
File Edit Format View Help
porta COM3
01130607 - E2 120 1
01130607 - E3 120 1
01130309 - P1 4 1
01130309 - P2 4 1
01130602 - L2 4 1
01130602 - L3 4 1
01130604 - L4 4 1
01130604 - L5 4 1
01130303 - C 10 1
01130304 - C 10 1
01130305 - C 10 1
01130306 - C 10 1
```

Figura 19 – Exemplo de arquivo de configuração

4.3. Loop Mestre

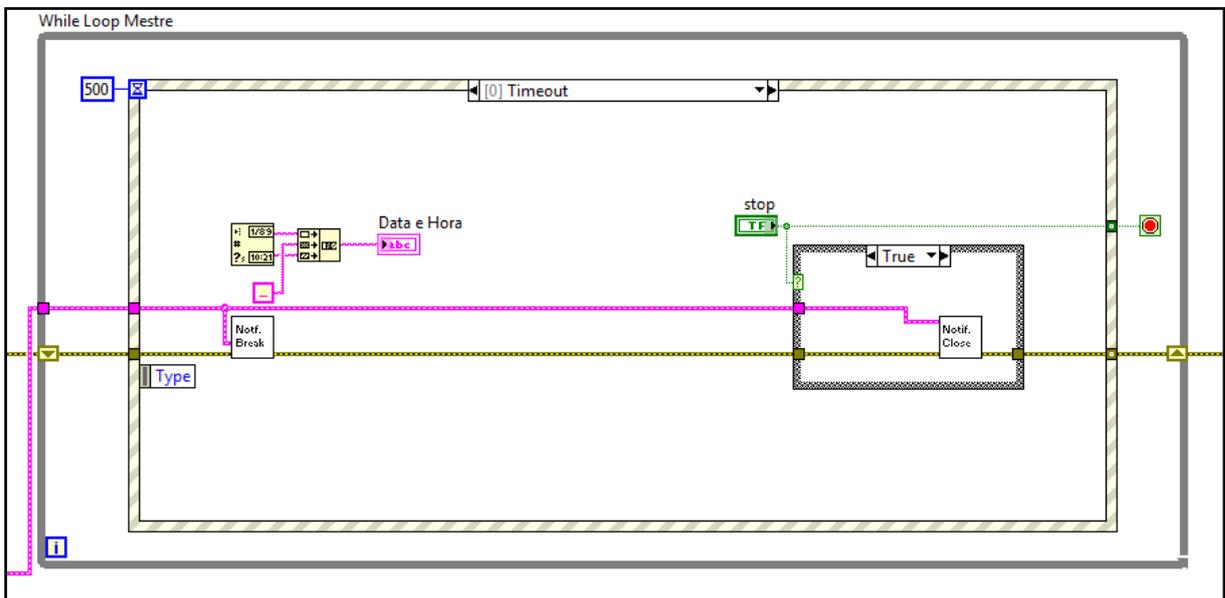


Figura 20 – Loop Mestre

O loop mestre era responsável por responder aos comandos dados pelo usuários pela interface do LabVIEW. Estruturado com um *event structure* dentro de

um *loop while*. Dessa forma, as sub-funções aqui presentes, com exceção do *timeout*, que será explicado em seguida, só ocorrem com comandos do usuário.

As sub-funções do Loop Mestre podem ser divididas em:

4.3.1. Função Timeout

O Timeout ocorre a cada 0,5 segundos e o que se faz é simplesmente uma atualização do horário na interface do programa, além de uma interrupção do funcionamento dele, caso já se tenha passado das 24h. E o início de funcionamento do programa a partir das 5h da manhã. Dessa forma, o programa é capaz de ligar e desligar o funcionamento da aquisição de dados automaticamente, após o término do último turno de funcionamento e antes do início do primeiro.

4.3.2. Atualização das comboboxes de data para consulta

As consultas são feitas a partir de arquivos previamente salvos (logs). Para evitar erros de execução do programa como, por exemplo, a procura por dados em uma data futura ou em data em que não se trabalhou, é exigido que o usuário selecione os campos de pesquisa na ordem ano, mês, dia.

Assim, a *combobox* mês, por exemplo, só carrega as informações após a escolha do ano.

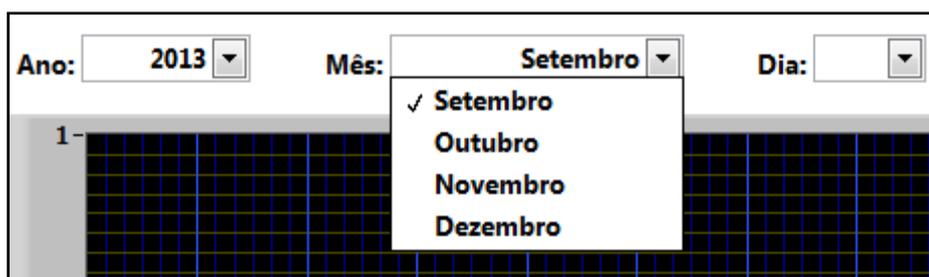


Figura 21 – Comboboxes de consulta

4.3.3. Iniciar consulta

Inicia a consulta com a solicitação do usuário a partir dos parâmetros escolhidos dados nas *comboboxes* ano, mês e dia, descrita na seção anterior.

A consulta em si, isto é, o processamento de dados a partir dos parâmetros e dos arquivos de input não são feitas aqui, mas realizado dentro do loop Consulta.

4.3.4. Iniciar aquisição

Função auto-explicativa. Inicia a aquisição a partir de solicitação do usuário.

4.3.5. Parar aquisição

Função auto-explicativa. Inicia a aquisição a partir de solicitação do usuário.

4.3.6. Atualização dos comboboxes de data para relatórios

Função idêntica à descrita em 4.2.2. que atua porém para a função de geração de relatórios.

4.3.7. Iniciar geração de relatórios

Inicia a produção de relatórios com a solicitação do usuários a partir de parâmetros de data dados pelo usuário.

4.3.8. Salvar modificações em arquivo de configuração

No caso de modificações de parâmetros de processamento ou da porta de comunicação com o CLP, apresentados na aba “Configurações”, o loop Mestre salva os novos valores dentro do arquivo config.ini (ver 4.2.5.3).

4.4. Loop Aquisição

Como descrito pelo nome, é nesse *loop* do software em que ocorrem as aquisições de dados provenientes dos sensores instalados nas máquinas

4.4.1. Comunicação MODBUS

O programa comunica-se com o CLP instalado na fábrica através da comunicação MODBUS.

Ao iniciar o programa, é aberta a porta de entrada serial e estabelecida a comunicação Visa MODBUS. Dentro do loop de aquisição, há verificação de que

não existem problemas com a comunicação Visa. Caso ela falhe, o programa reabre a comunicação para que a aquisição inicie novamente.

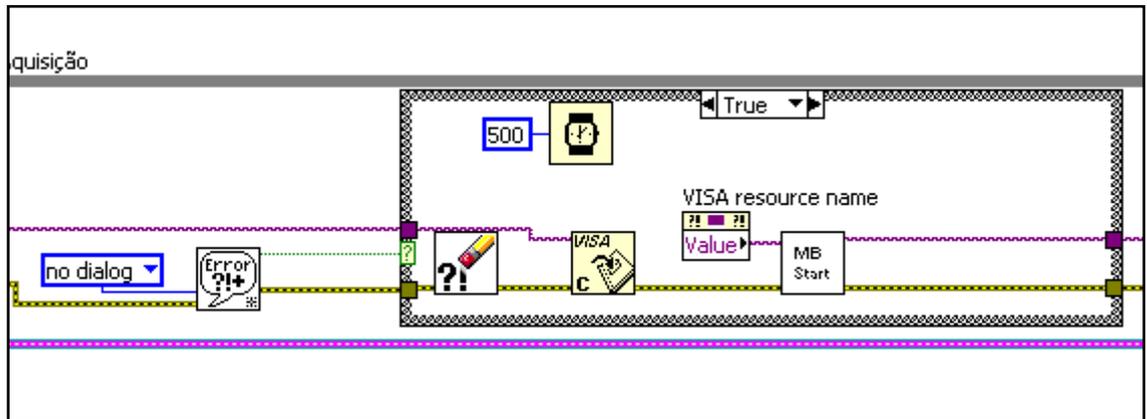


Figura 22 – Subfunção Comunicação MODBUS

Para que a VI MB Serial Master Query faça as aquisições corretas, devem-ser dados os seguintes parâmetros:

- Modo de transmissão: RTU
- Endereço do escravo: 1
- Código da função: 3 (Read Holding Registers)
- Endereço inicial: 560
- Quantidade: 4

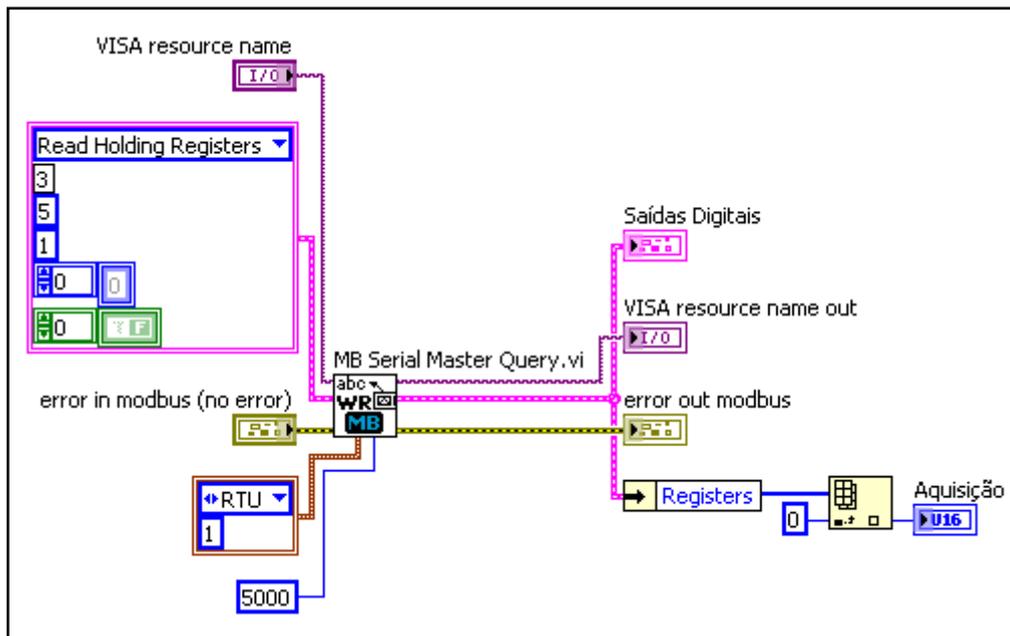


Figura 23 – Parâmetros da Comunicação MODBUS

A entrada dada é um número decimal que representa as 12 entradas booleanas do CLP (sendo a entrada I12 representada pelo dígito menos significativo e, I1 pelo mais significativo).

Exemplo: o valor decimal 3202 representa o binário 11001000011 que, por sua vez, indica que as entradas I1, I2, I8, I11 e I12 estão ativas. O que significa que as máquinas ligadas a essas entradas encontram-se em funcionamento.

4.4.2. Exclusão de interferências

Durante a aquisição, o programa faz 10 aquisições de dados em um intervalo de 100 ms e verifica o valor modal³ desses valores, para que assim se exclua qualquer ruído excedente.

³Valor modal é o valor que detém o maior número de observações, ou seja, o valor ou valores mais frequentes. (fonte : http://pt.wikipedia.org/wiki/Moda_%28estat%C3%ADstica%29)

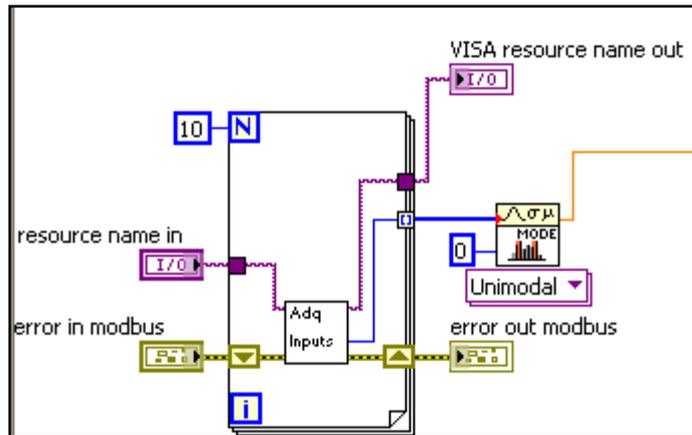


Figura 24 – Exclusão de interferências por valor modal

Desse modo, tem-se as seguintes correções no caso de ruídos anômalos, como se vê na figura a seguir:

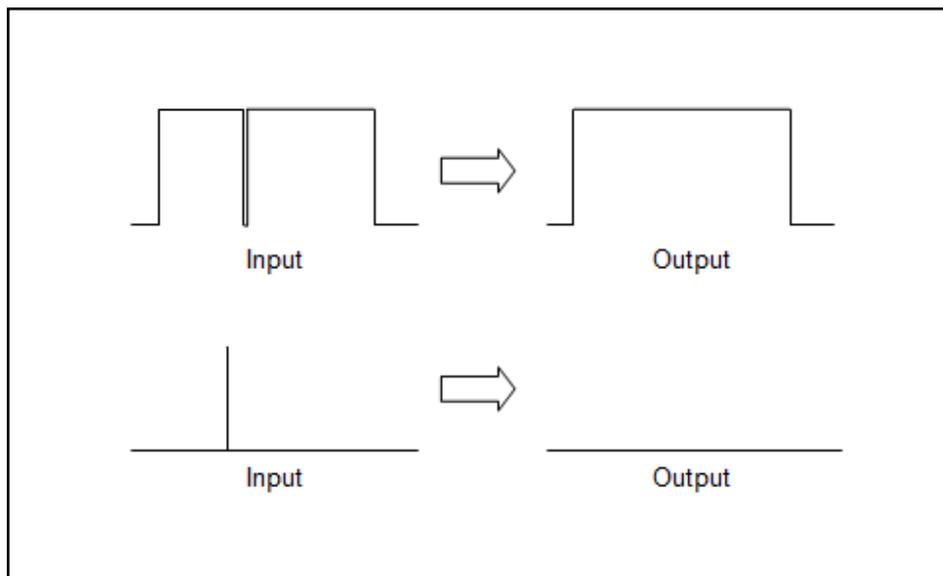


Figura 25 – Exemplo de ruídos anômalos

4.4.3. Geração de arquivo log1

O valor é comparado com o outro anteriormente adquirido. Se eles forem iguais, o sistema apenas descarta o novo valor; se forem diferentes, o valor é repassado em fila do tipo FIFO para o *loop* de processamento e adicionado para ser salvo em um arquivo de extensão log1 (arquivo que não é utilizado, mas que se

decidiu por guardar visto que ele poderia ser útil no caso de manutenção do programa).

Após 30s somente com aquisições repetidas, é repassado o mesmo valor por fila, para que se mantenham os arquivos log1 constantemente atualizados.

Além disso, há uma lógica criada dentro do loop mestre para evitar que qualquer aquisição realizada entre 0h e 4h nos dias de trabalho da semana, ou nos sábados e domingos, seja ignorada.

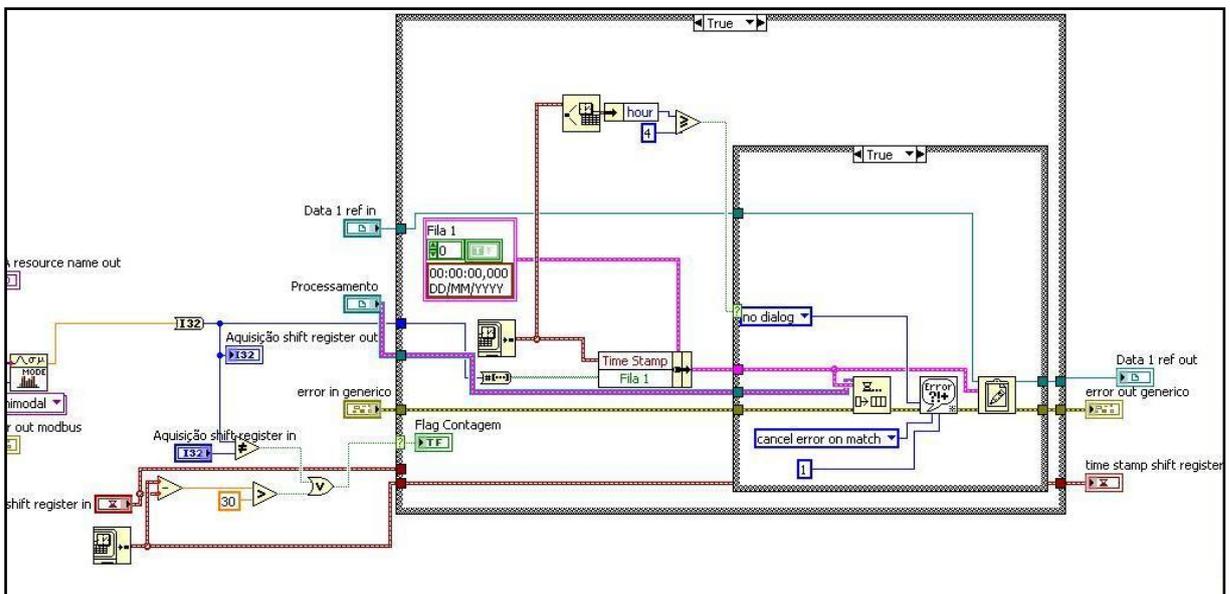


Figura 26 – Parte do loop de Aquisição

4.5. Loop Processamento

É nesse loop é que os dados adquiridos e já filtrados são processados para posteriormente serem contabilizados ou não como peças.

4.5.1. Separação de bits

Como previamente mencionado, o valor recebido do hardware é um número decimal que representa o estado das 12 entradas digitais do CLP. Assim, o primeiro procedimento a se fazer no processamento é a conversão desse dado em um array de bits.

4.5.2. Verificação de intervalo de tempo

No momento em que há uma mudança de nível lógico de uma máquina, indicando o início ou o término do processamento de uma peça, o programa salva o timestamp⁴ em uma variável.

Dessa forma, o intervalo de tempo entre duas timestamps em que se tinha nível lógico alto nada mais é que o tempo de processamento de uma peça.

Para que se contabilizar uma peça, é verificado se esse tempo de processamento é maior que um tempo mínimo de referência. Dessa forma, evita-se que se contabilize peças no caso da máquina ser ativada por um curto período de tempo devido a um falso acionamento, por exemplo.

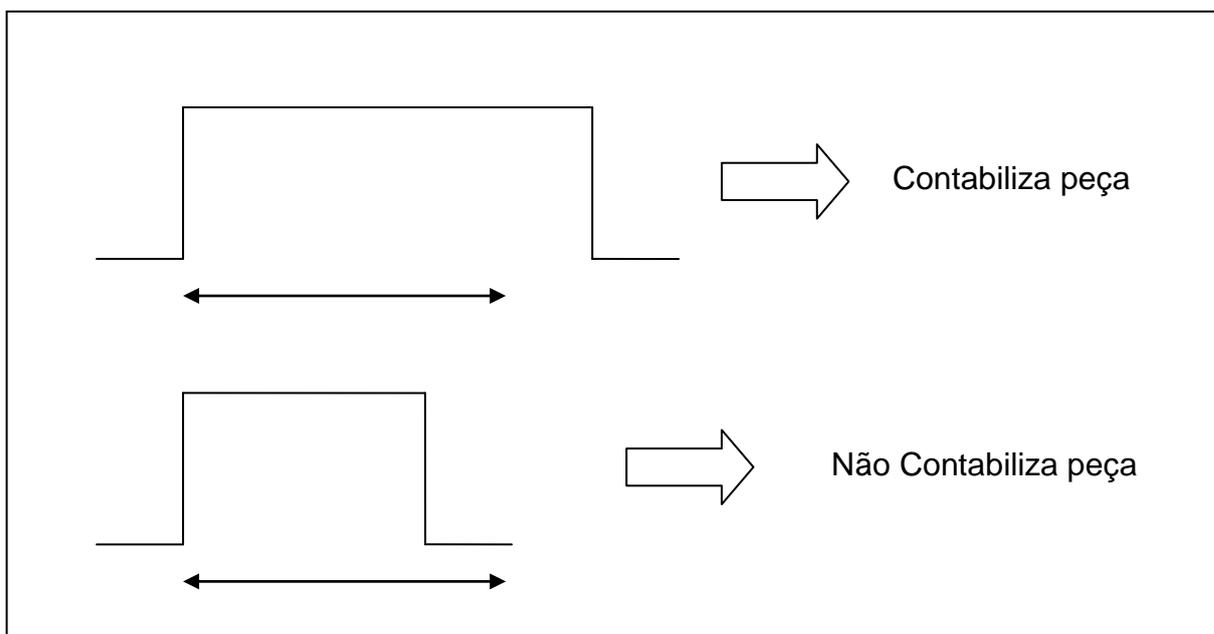


Figura 27 – Verificação do intervalo de tempo

4.5.3. Geração do arquivo log2

Por fim, é gerado o arquivo log2, posteriormente utilizado para a função consulta.

⁴ Timestamp: uma cadeia de caracteres denotando a hora ou data que certo evento ocorreu

Os arquivos tipo log2 são utilizados para que a função de consulta possa elaborar os gráficos de produção de um dia. Com intuito de impedir que existam erros de gravação e acesso de dados simultâneos, o controle é feito por semáforos.

4.5.4. Projeção gráfica

A projeção gráfica consiste em simplesmente projetar os dados contidos no vetor principal no gráfico presente na aba de monitoramento do programa.

4.6. Loop Consulta

Para realizar uma consulta, deve-se escolher uma data preenchendo as *comboboxes* da aba consulta e clicar no botão “consultar”. Ao fazer isso, o loop mestre envia um notificador para o loop consulta.

O *loop* de consulta, tem lógica idêntica à lógica do *loop* de gráfico. Os dados recebidos por fila são lançados em um gráfico para visualização do usuário.

Como todos os dados foram adquiridos anteriormente, entretanto, o desenho do gráfico é feito instantaneamente.

Quando uma nova consulta é realizada, o gráfico da aba consulta é reinicializado e preenchido com os novos valores.

4.6.1. Localização do arquivo

Este, por sua vez, procura o arquivo correspondente à data escolhida pelo usuário

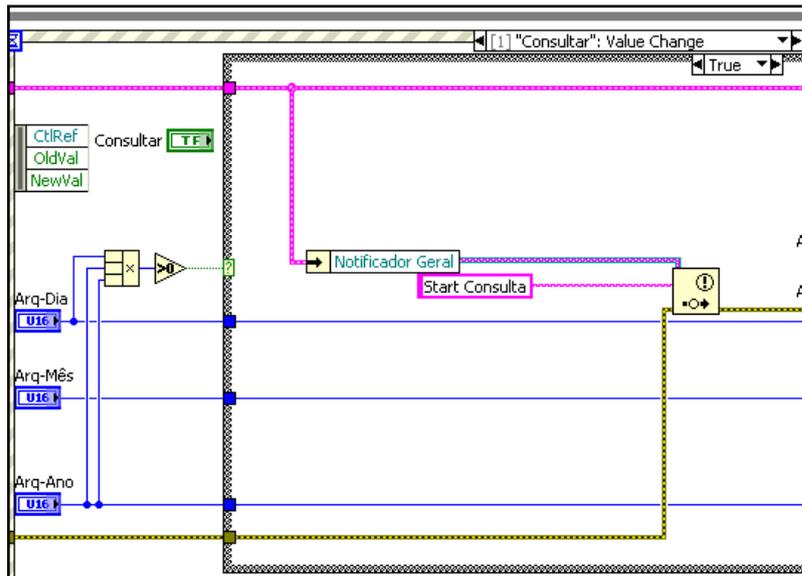


Figura 28 – Subfunção de localização de arquivo

4.6.2. Projeção Gráfica

Com os dados advindos dos arquivos log2, projeta-se então o gráfico da produtividade de peças, de maneira idêntica à sub-função executada no *loop* Processamento.

4.7. Loop Relatório

Por fim, o loop relatório tem o objetivo de trazer todos os dados contidos no vetor peças para um relatório em Excel, mais facilmente utilizado pelos gerentes de produção.

4.7.1. Localização do arquivo.

Subfunção do *loop* consulta reutilizada.

4.7.2. Geração de relatório

Consiste em gerar um arquivo com extensão .xlsx contendo o vetor peças organizado na forma de um relatório de produção.

Capítulo 5: Trabalho complementar: Sistema de contagem de peças

Com o término do trabalho de desenvolvimento do sistema de monitoramento da produção, percebeu-se que os objetivos propostos pelo projeto haviam sido cumpridos.

Como anteriormente listados:

1. Fazer a contagem de processamentos de peças feitos em cada uma das máquinas;
2. Exibir ao gerente de produção um acompanhamento em tempo-real das máquinas;
3. Gerar relatórios de produção a partir de datas selecionadas pelo gerente.

Entretanto, a contagem de quantidade de processamentos realizados em cada máquina não podia ser tratado como o *output* total da linha de produção. Isto porque, no caso de um reprocessamento de peça, uma mesma peça é contabilizada duas vezes. Esse evento, infelizmente, não poderia ser contornado.

Por esse motivo, decidiu-se por incluir um trabalho adicional ao projeto já desenvolvido com o seguinte objetivo:

- 4. Fazer a contagem oficial de peças produzidas na linha de produção.**

5.1. Solução conceitual

O primeiro passo para se elaborar um método confiável de contagem de peças, pensou-se em diferentes soluções, a mais aceita por apresentar um equilíbrio entre preço e confiabilidade foi a utilização de etiquetas de códigos de barras nas peças processadas na linha.

Assim, cada peça seria identificada com um código de barras exclusivo e, por meio de dois pontos de checagem, um localizado na entrada de peças na linha de manufatura e outro posicionado na saída, ter-se-ia uma contagem mais confiável da quantidade processada.

Em um momento posterior, seria também possível adicionar pontos de checagem em setores intermediários para se ter uma maior rastreabilidade de peças dentro do sistema.

Esse capítulo então descreve o desenvolvimento dessa concepção e os testes realizados para verificar a plausibilidade de se utilizar esse sistema.

5.1.1. Código de barras unidimensional



Figura 29 – Exemplo de código de barras unidimensional

A utilização de códigos de barras unidimensionais consiste em um método de identificação amplamente utilizado no mundo, principalmente devido ao seu baixo custo de implementação quando comparado com as demais tecnologias de representação.

Baseia-se em uma representação gráfica de dados, codificados em barras pretas e brancas. A decodificação é feita através de um scanner emissor de um feixe de luz vermelha que, percorrendo todo o código, é absorvida nas barras pretas e refletida nas brancas.

5.2. Testes

Foram realizados então testes que visavam verificar a viabilidade da instalação do sistema de contagem de peças em alguns pontos da linha utilizando leitores comerciais.

No primeiro momento do projeto, a utilização de leitores industriais foi descartada, devido ao elevado preço dos equipamentos.

5.2.1. Primeira etapa

A primeira etapa de testes teve por objetivo verificar a possibilidade de implantação do sistema sem a necessidade de manipulação do leitor comercial por um operador. Buscou-se uma maneira de posicionar o leitor, de maneira fixa, em um dos pontos da linha, e assegurar que ele fizesse a leitura do código de barras de todas as peças que por ali passavam.

Foram encontradas as dificuldades listadas abaixo:

- Colagem manual de etiquetas, impossibilitando um posicionamento padrão e preciso dos códigos de barras;
- Inserção manual de peças na esteira, resultando em diferente posicionamento de cada peça e, conseqüentemente, de cada código de barra;
- Lotes de peças com diferentes alturas.

Como alguns desses problemas seriam extremamente difíceis ou impossíveis de serem contornados e, como as leituras foram realizadas com sucesso em apenas 60% de cerca de 300 peças, concluiu-se que sem a manipulação humana, não seria possível utilizar esse sistema.

5.2.2. Segunda etapa

Em seguida, buscou-se observar a possibilidade de implantação do sistema com a manipulação do leitor comercial por um operador. O uso dos leitores foi realizado por funcionários da fábrica, sob supervisão e auxílio de responsáveis pelo projeto.

Concluiu-se que essa implantação é possível, entretanto algumas dificuldades ainda existem:

- Falta de conhecimento e hábito para utilização do leitor por parte do operador;

- Falhas aleatórias em leituras, devido a problemas técnicos (fato ocorrido em apenas 1% de cerca de 300 casos);
- Dificuldade na manipulação do instrumento, devido à constante necessidade do operador de segurar o leitor, fazer a leitura e, em seguida soltá-lo para realizar outra tarefa;
- Dificuldade em ouvir o sinal sonoro emitido pelo aparelho no instante da leitura, devido ao barulho na fábrica.

Acredita-se, entretanto, que essas dificuldades podem ser satisfatoriamente solucionadas por meio das seguintes atitudes:

- Instalação de um suporte mecânico fixo para o leitor nos pontos necessários, facilitando a manipulação do mesmo por parte do operador;
- Instalação de um software que permitiria ao operador assegurar a realização da leitura e digitar manualmente o código, caso ela não seja possível por meio do leitor;
- Uso do software supracitado para notificar ao operador o momento de leitura da peça;
- Treinamento prévio dos operadores.

5.2.3. Análise de possível perda de produtividade

Estudou-se também se existiria uma possível perda de produtividade na linha devido ao uso do sistema. Concluiu-se que a instalação do sistema na entrada e saída da linha nada afetaria na produção total, visto que esses dois pontos não são dados como gargalos produtivos.

Além disso, verificou-se que o aumento do tempo médio em ambos os postos de operação seria de 22% a partir dos testes realizados, contudo, afirma-se que a instalação de um suporte mecânico para facilitar no manuseio do equipamento reduziria essa taxa consideravelmente.

5.3. Software extra: Contagem de peças

Como anteriormente citado, a necessidade de um *software* para auxílio dos colaboradores na correta utilização dos leitores seria primordial. Assim, foi também desenvolvido um programa, cuja interface pode ser visualizada abaixo.

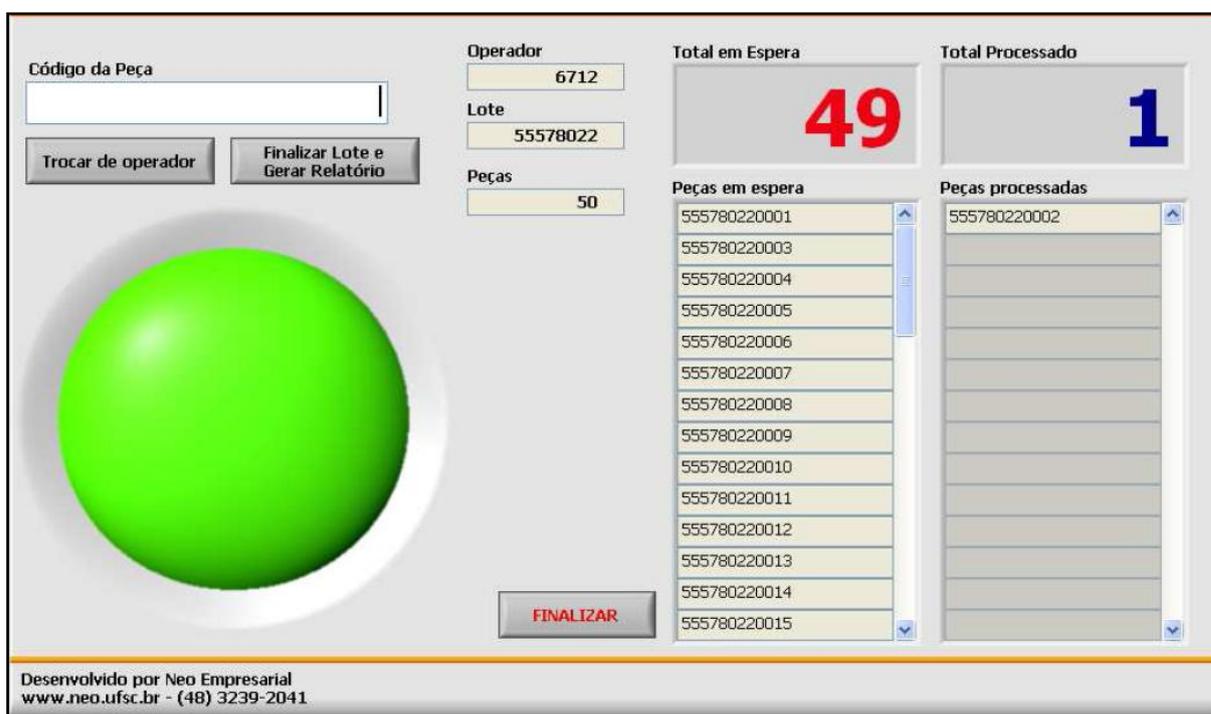


Figura 30 – Interface do software para contagem de peças

Devido a sua simplicidade, decidiu-se por não se acrescentar os detalhes do desenvolvimento desse programa.

Basicamente, ao ser iniciado, o programa solicita três informações básicas:

- Código do operador no posto de trabalho
- Número do lote de peças
- Tamanho do lote

A partir dessas informações, o programa preenche uma lista com o código de todas as “peças em espera”, ou seja, todas as que pertencem a esse lote e ainda não tiveram o código lido pelo programa.

Cada código de peça lido com o leitor de código de barras (ou inserido por meio de teclado numérico) é retirado de “peças em espera” e passa então para a lista de “peças processadas”.

Ao término do programa, seria gerado um relatório com cada código lido e seu respectivo *timestamp*, como visto na figura a seguir:

A	B	C	D	E	F
Relatório gerado em:	22/01/2014				
Operador	6712				
Lote	55578022				
Peças totais	50				
Peças processadas	50				
ID	Timestamp				
555780220002	22/01/2014 14:21				
555780220007	22/01/2014 14:21				
555780220014	22/01/2014 14:21				
555780220012	22/01/2014 14:22				
555780220001	22/01/2014 14:22				
555780220037	22/01/2014 14:23				
555780220002	22/01/2014 14:23				
555780220015	22/01/2014 14:24				
555780220011	22/01/2014 14:24				
555780220020	22/01/2014 14:24				
555780220022	22/01/2014 14:26				

Figura 31 - Relatório do software para contagem de peças

Como o programa seria, em um primeiro momento, utilizado na entrada e na saída da linha, ter-se-ia um registro do horário do início e término de processamento de cada peça.

Capítulo 6: Indicadores de processo

Com os trabalhos executados e descritos nos capítulos anteriores seria então possível passar a fazer a coleta de determinados indicadores por parte da gerência da linha de produção, o que permitiria um melhor controle e operação da produção. Neste capítulo, os indicadores são explicados.

6.1. Lead Time de Produção

É o período entre o início da atividade (nesse caso, entrada da matéria-prima na linha de produção) e o seu término (a saída da peça pronta) (5).

Como relatado, o *software* para contagem de peças (ver 5.3), teria em seus relatórios a *timestamp* de cada uma das partes processadas. Assim, a diferença entre as *timestamps* do computador posicionado no posto de saída da linha e do outro, localizado na entrada, forneceria o *lead time* de cada item.

A média desses tempos constitui um valioso indicador para o departamento de controle da produção, haja visto que é uma informação básica para que se possa planejar a produção dos pedidos solicitados.

6.2. Overall Equipment Effectiveness (OEE)

O Overall Equipment Effectiveness (OEE), ou Eficácia Global do Equipamento é um método que vem se popularizando no mundo industrial para avaliar quão eficaz é um equipamento. Baseia-se em três diferentes aspectos de avaliação de desempenho:

- Tempo disponível para utilização do equipamento;
- Qualidade do produto manufaturado;
- Velocidade, ou taxa de produção, do equipamento. (6)

Assim, tem-se:

$$OEE (\%) = ID(\%) * IP(\%) * IQ(\%)$$

Abaixo a explicação de cada um dos fatores que compõem a equação:

ID – Índice de disponibilidade: leva em conta as paradas não planejadas originadas por quebra do equipamento, *setup*, necessidades de ajustes no equipamento durante a troca do produto, etc. Avalia o fator tempo disponível. É calculado por:

$$ID(\%) = \frac{TPP - TPnP}{TPP}$$

Onde:

- TPP: Tempo de produção planejado
- TPnP: Tempo de paradas não-planejadas

IP – Índice de performance: leva em conta as perdas de velocidade, aumento dos ciclos das operações, os atrasos, etc. Mede a velocidade do equipamento. O índice de performance é dado por:

$$IP(\%) = \frac{TCU * n}{TPP - TPnP}$$

Onde:

- TCU: tempo de ciclo unitário
- n: quantidade de ciclos unitários de processamento
- TPP: Tempo de produção planejado
- TPnP: Tempo de paradas não-planejadas

É neste fator que se faz presente a utilização do Sistema de Monitoramento da Produção, pois é por meio dele que se tem obtém a quantidade de ciclos unitários de processamento (n) realizados por cada máquina durante um determinado período de tempo.

IQ – Índice de qualidade: leva em conta os refugos que tiveram origem em falhas nos recursos físicos. Avalia a qualidade do equipamento.

$$IQ(\%) = \frac{n - r}{n}$$

Onde:

- n: quantidade de ciclos unitários de processamento
- r: quantidade de refugos

Na avaliação do OEE geral, ou seja, de toda a linha de produção (e não de um único equipamento), o *software* de Contagem de Peças permite a contagem de ambos os valores, n e r.

A diferença na quantidade de peças contabilizadas entre a entrada e saída da linha representaria o valor r de peças que não chegaram ao fim do processamento por apresentarem algum tipo de defeito durante um dos sub-processos intermediários.

Capítulo 7: Conclusões e Perspectivas

A característica mais marcante do projeto aqui apresentado foi a simplicidade. Focou-se em questões-chave do projeto Sistema de Monitoramento da Produção e evitou-se complexidades desnecessárias. Assim, obteve-se uma solução simples e eficaz.

Dessa maneira, pôde-se executar um trabalho dentro do prazo determinado e, ao longo de seu desenvolvimento, já começar a colher os frutos do trabalho realizado.

Outra vantagem, foi o aproveitamento do tempo economizado para solucionar uma questão complementar ao projeto, como descrito no Capítulo 5.

Além disso, com uma solução de menor complexidade, pôde-se garantir o essencial usuário, sem criar um produto que falha pelo excesso de funções. E o mais primordial, como consequência de uma baixa complexidade, tem-se uma maior robustez e maior facilidade para entendimento por futuros usuários e desenvolvedores, caso um dia o programa necessite de manutenção ou precise ser atualizado.

De trabalhos futuros, envolvendo o Sistema de Monitoramento da Produção, que aqui poderiam ser incluídos, pode-se citar:

- Modificação do programa para que, um dia, seja possível se fazer a inclusão de novas máquinas dentro do sistema, sem que haja a necessidade de interferência de um desenvolvedor.
- Sistemas de alertas, com a função de identificar falhas no funcionamento das máquinas e aí possam automaticamente comunicar um responsável (caso uma máquina se encontre fora de operação por razão não proposital, por exemplo, e a gerência de produção ainda não tenha identificado o fato).
- Simplificação do sistema de consulta. Para que se possa realizar consultas aos dados de maneira mais ágil, sem selecionar parâmetros em

três diferentes *comboboxes*, mas em um único calendário (similar ao utilizado no Sistema Operacional Windows, por exemplo).

- Alocação do sistema em uma plataforma *online*, para que possa ser utilizado e/ou visualizado a partir de qualquer computador da rede WEG.

Como já citado os trabalhos desenvolvidos até o momento trouxeram diversos benefícios, como:

- A elaboração de relatórios de produção de maneira mais rápida pela gerência de produção.
- O acompanhamento em tempo-real das máquinas.
- Contagem eficiente das peças processadas em cada máquina.
- Contagem oficial de peças processadas na linha.

Ressalta-se que, todos esses objetivos atendem a necessidades de pessoas diretamente ligadas à produção, entretanto, em um segundo momento, o projeto passa a auxiliar no desenvolvimento de indicadores de produtividade, como Lead Time e OEE, para uso em um nível hierárquico superior. Assim, não apenas os controladores da produção são beneficiados, mas também engenheiros de controle da produção. Desse modo, aumenta-se o retorno real dos resultados do projeto em relação ao retorno inicialmente esperado.

O sistema apresentado já é um dos modelos de base utilizados por uma equipe da WEG responsável por futuramente padronizar o acompanhamento e monitoramento dos processos produtivos dentro da empresa. Essa é uma necessidade que ocorre não apenas na Fábrica VI, onde o projeto foi desenvolvido, mas em outros pontos de manufatura da WEG.

Por fim, nota-se que um dos elementos essenciais que permitiu essa expansão na utilidade do projeto foi o caráter multidisciplinar do conhecimento fornecido pelo curso de Engenharia de Controle e Automação. Tratou-se de um projeto que, inicialmente e principalmente, envolveu tópicos da Informática industrial e, ao fim, trouxe ganhos para questões utilizadas na Engenharia de Produção.

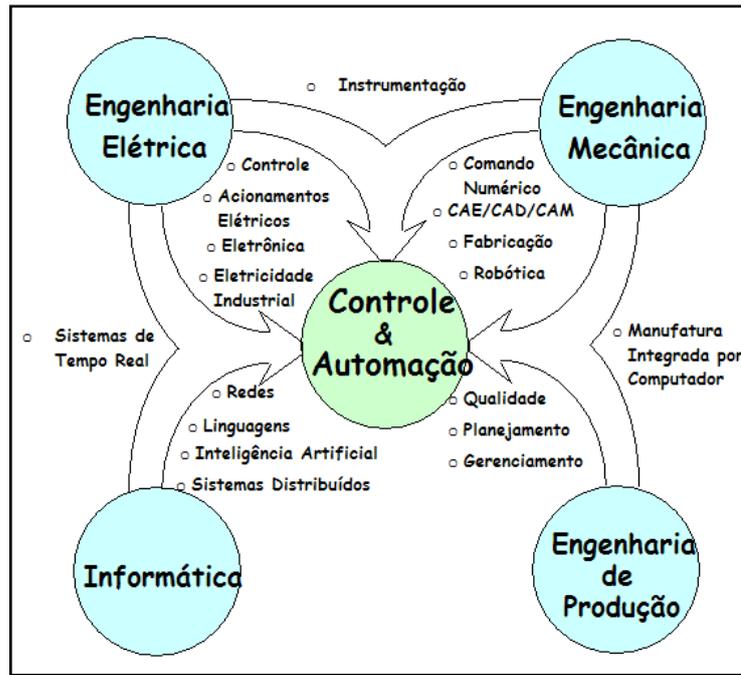


Figura 32 - Inter-relação do curso de Engenharia de Controle e Automação com outras engenharias

Capítulo 8: Bibliografia

(1) **National Instruments.** Application Design Patterns: Master/Slave. *National Instruments Website*. [Online] [Citado em: 15 de Dezembro de 2013.] <http://www.ni.com/white-paper/3022/en/#toc3>.

(2) —. O que é LabVIEW. *National Instruments Website*. [Online] [Citado em: 23 de February de 2014.] <http://www.ni.com/newsletter/51141/pt/>.

(3) —. What is a notifier? *National Instruments Website*. [Online] [Citado em: 16 de February de 2014.] <http://digital.ni.com/public.nsf/allkb/B9398355D9550EAF862566F20009DE19>.

(4) —. What is a queue? *National Instrumens Website*. [Online] <http://digital.ni.com/public.nsf/allkb/DD7DBD9B10E3E537862565BC006CC2E4>.

(5) **LAMBERT, Douglas M., STOCK, James R. e ELLRAM, Lisa M.** *Fundamentals of logistics management*. New York : McGraw-Hill, 1998.

(6) **SLACK, Nigel, Chambers, Stuart e Johnston, Roberts.** *Operations Management*. s.l. : Prentice Hall, 2007.