

RAFAEL ALZUGUIR ROSIN

**AVALIAÇÃO DA EFICIÊNCIA EM PROCESSOS DE FABRICAÇÃO
COM BASE NO CONTROLE DE PRODUÇÃO
ESTUDO DE CASO: PROCESSO DE ESTAMPARIA**

*nota final
9,0 (nove e zero)*
WAM

Trabalho de Conclusão de Curso
apresentado à Escola Politécnica
da Universidade de São Paulo

Área de Concentração:
Engenharia Mecatrônica e
Sistemas Mecânicos

São Paulo
2004

RAFAEL ALZUGUIR ROSIN

**AVALIAÇÃO DA EFICIÊNCIA EM PROCESSOS DE FABRICAÇÃO
COM BASE NO CONTROLE DE PRODUÇÃO
ESTUDO DE CASO: PROCESSO DE ESTAMPARIA**

Trabalho de Conclusão de Curso
apresentado a Escola Politécnica
da Universidade de São Paulo

Área de Concentração:
Engenharia Mecatrônica e
Sistemas Mecânicos

Orientador:
Gilmar Ferreira Batalha

São Paulo
2004

FOLHA DE APROVAÇÃO

Eu, Prof. Dr. Gilmar Ferreira Batalha, exercendo a função de Professor Orientador deste trabalho, informo que este Trabalho de Formatura está **APROVADO** para apresentação final perante a Banca Examinadora.

São Paulo, 5 de Dezembro de 2004.

Prof. Dr. Gilmar Ferreira Batalha

Rafael Ronin

RESUMO

Para que fossem feitas melhorias em processos de fabricação das empresas, surgiram fornecedores de pacotes de Planejamento e Controle de Produção (PCP), que possuem seu foco em grandes empresas que podem investir nesta tecnologia. Entretanto, muitas das empresas que possuem estes pacotes, ainda necessitam de informações que não são processadas por tais softwares. Isto justifica muitas vezes a adoção de sistemas específicos, muitos deles em paralelo, que satisfaçam as suas necessidades. Informações como o desempenho da fabricação, tornam-se atualmente premissas básicas de qualquer indústria, entretanto, nem sempre são passadas informações que possibilitem não apenas a leitura de dados, mas também a identificação de problemas no chão de fabrica. O objetivo geral deste trabalho consiste em desenvolver um software que possibilite analisar a eficiência de uma estamparia com base no apontamento de dados do controle de fabricação, e através desta análise propor melhorias no que diz respeito ao chão de fábrica. Como uma forma de realizar tal estudo, foi criada uma folha de Controle de Produção, mediante o registro in-situ da produção e os problemas encontrados no processo. Com estes dados parte-se para determinar o quanto que se perde da eficiência para cada um dos fatores de parada da empresa, tais como, problemas de ferramentaria, falta de material, revezamento de funcionários, manutenção preventiva, transporte de material e trocas de ferramenta.

Palavras chave: processo de fabricação, estamparia, eficiência.

ABSTRACT

To make better in companys fabrication progress, suppliers of Planing and Production Control (PPC) packages have come, and this ones have focused in big companys that can invest in this technology. Although, a lot of companys who have this packages, needs informations that are not processed by this softwares. This justify the adoption of parallel systems, that can satisfy these needs. Information as the performance of the manufacture, become basic premises of any industry. The general objective of this work, consist in develop a software that can analyze the efficiency of a stamping process, using the appointments made in Production Control, and using this analyses propose betterment to the company. To carry through such study, a leaf of Production Control was created, for the register of the problems found in the process. With these data, we determine how much we lose in the efficiency for each one of the stop factors.

AGRADECIMENTOS

Ao professor Gilmar Ferreira Batalha, por orientar este trabalho e minha vida profissional.

À Keiper do Brasil, pelo apoio no trabalho realizado.

Aos mestres e funcionários, pelos ensinamentos passados.

Aos meus pais, por financiarem meus estudos, pela força e tudo mais.

Aos meus irmãos, pela companhia e pelas palavras de apoio.

Aos meus amigos, por tornarem meus dias mais agradáveis.

ÍNDICE

1. INTRODUÇÃO	09
1.1 Contexto	09
1.2 Objetivos	09
2. CONCEITOS DE ADMINISTRAÇÃO DA PRODUÇÃO	11
2.1 Planejamento e Controle do Chão de Fábrica	14
2.2 Balanceamento de Linha	14
2.3 Layout do Chão de Fábrica	15
3. DESENVOLVIMENTO DE UM PROJETO DE SOFTWARE	21
3.1 Introdução	21
3.2 Ciclo de Vida Clássico	22
3.3 Modelo Espiral	23
4. CONTROLE DE PRODUÇÃO EM UMA ESTAMPARIA	25
5. FORMULAÇÃO TEÓRICA	27
6. MOTIVOS DE PARADA	31
6.1 Ferramentaria	31
6.2 Setup	32
6.3 Manutenção Mecânica / Elétrica	38
7. RESULTADOS OBTIDOS	40
7.1 Resultados Gerais	40
7.2 Ferramentaria	41
7.3 Setup	43
7.4 Manutenção Mecânica / Elétrica	46
8. SOFTWARE DE ANÁLISE DA EFICIÊNCIA	47
9. CONCLUSÕES	49
10. ANEXOS	50
11. BIBLIOGRAFIA	68

LISTA DE FIGURAS

<i>Figura 1: Gráfico de Gantt</i>	<i>13</i>
<i>Figura 2: Fatores que influenciam o Layout</i>	<i>16</i>
<i>Figura 3: Os 5 tipos básicos de fluxo</i>	<i>17</i>
<i>Figura 4: Ciclo de Vida Clássico</i>	<i>22</i>
<i>Figura 5: O modelo espiral</i>	<i>23</i>
<i>Figura 6: Elementos básicos de uma ferramenta</i>	<i>31</i>
<i>Figura 7: Ferramenta com STR</i>	<i>34</i>
<i>Figura 8: Perfil Superior de Ferramenta com STR</i>	<i>34</i>
<i>Figura 9: Ferramenta com STR isolada</i>	<i>35</i>
<i>Figura 10: Fixação inferior de Ferramenta com STR</i>	<i>35</i>
<i>Figura 11: Sistema STR simples</i>	<i>36</i>
<i>Figura 12: Sistema STR em máquina Transfer</i>	<i>36</i>
<i>Figura 13: Fixação Inferior de STR – Transfer</i>	<i>37</i>
<i>Figura 14: Sistema STR – Transfer Superior</i>	<i>37</i>
<i>Figura 15: Paradas Gerais da Estamparia</i>	<i>41</i>
<i>Figura 16: Paradas de Ferramentaria</i>	<i>42</i>
<i>Figura 17: Gráfico de Gantt – Ferramentaria</i>	<i>42</i>
<i>Figura 18: Controle Estatístico de Processo para Setup</i>	<i>43</i>
<i>Figura 19: Controle Estatístico de Processo para STR</i>	<i>44</i>
<i>Figura 20: Comparação de tempo para sistema STR</i>	<i>45</i>
<i>Figura 21: Tela de Entrada do Sistema</i>	<i>47</i>
<i>Figura 22: Controle de Produção do Sistema</i>	<i>48</i>

LISTA DE TABELAS

<i>Tabela 1: Exemplo de dados de Produção</i>	<i>12</i>
<i>Tabela 2: Vantagens e desvantagens dos tipos de arranjo físico</i>	<i>19</i>
<i>Tabela 3: Horas por período</i>	<i>29</i>
<i>Tabela 4: Tempo Médio de Setup</i>	<i>45</i>

1. Introdução

1.1 Contexto

Com o crescimento da economia globalizada, houve uma mudança na proximidade que existia entre os mercados consumidores e principalmente, entre empresas concorrentes.

Tais mudanças, em um cenário de competição, passaram a exigir que as empresas se aperfeiçoassem cada vez mais em sua produção, de forma a oferecerem os melhores serviços ou produtos frente aos seus concorrentes. Este fator fez com que empresas que se sentiam acomodadas sumissem, por terem sempre trabalhado em um mercado sem concorrentes, onde os clientes não tinham alternativas.

Para que fossem feitas melhorias na produção das empresas, surgiram empresas fornecedoras de pacotes de Planejamento e Controle de Produção (PCP), que possuíam seu foco em grandes empresas que pudessem investir nesta tecnologia.

Entretanto, muitas das empresas que possuem estes pacotes, ainda necessitam de informações que não são passadas por estes, o que acaba exigindo a adoção de sistemas em paralelo que satisfaçam as suas necessidades.

Informações como o desempenho da fabricação, tornou-se uma premissa básica de qualquer industria, entretanto, nem sempre são passadas informações que possibilitem não apenas a leitura de dados, mas também a identificação de problemas no chão de fabrica.

1.2 Objetivos

O objetivo geral deste trabalho consiste em desenvolver um software que possibilite analisar a eficiência de uma estamparia com base no apontamento manual do controle de produção, e através desta analise propor melhorias no que diz respeito ao chão de fabrica.

Este estudo será realizado em uma empresa (Keiper do Brasil) ligada ao setor automotivo, fabricante de estruturas metálicas de assentos e outros dispositivos ligados a estes.

Em suas instalações, tal empresa possui uma estamparia com 60 prensas e cerca de 80 funcionários, e apesar de já possuir um sistema que meça a eficiência do processo, o dado que este informa é apenas uma eficiência global de um período específico, não sendo possível realizar um estudo mais detalhado do que pode ter acontecido ao longo do dia em cada uma das máquinas.

Como uma forma de realizar tal estudo, foi criada uma folha de Controle de Produção, onde cada funcionário aponta manualmente a sua produção e os problemas encontrados no processo, para que o encarregado da linha possa se informar a respeito do que pode estar acontecendo.

Nosso objetivo é criar um sistema que receba os dados apontados pelos funcionários nas máquinas e que processe estes para que possamos analisar a eficiência e as razões de suas grandes, e com base nestes dados propor melhorias que poderiam ser tomadas.

Para atingir este objetivo geral, vários objetivos específicos precisam ser atingidos, entre eles:

- Apresentar alguns dos conceitos de administração da produção.
- Método de Controle de Produção manual apontado pelos funcionários.
- Formulação teórica da eficiência com base no Controle de Produção.
- Método de implantação via software do sistema.

2. Conceitos de Administração da Produção

2.1 Planejamento e Controle do Chão de Fabrica

A conexão entre a programação e o controle da produção e o SAP se dá pelo plano de material e capacidade, uma vez que a capacidade é especialmente crítica no gerenciamento detalhado do chão de fabrica, pois, em sua essência, representa a disponibilidade de recursos que vão de encontro como plano de material. [MOREIRA, 1996].

O objetivo principal desta conexão é cumprir o planejamento de material provendo a parte certa na hora certa, resultando no cumprimento do programa mestre e na satisfação do cliente.

No cumprimento deste objetivo, a programação deve:

- Permitir que os produtos tivessem a qualidade especificada
- Fazer com que máquinas e pessoas operem com níveis desejados de produtividade
- Reduzir estoques e custos operacionais

Isto significa assegurar que as ordens de produção serão cumpridas da forma certa na hora certa, através do relato periódico do material em processo acumulado, do estado atual de cada ordem de produção, das quantidades produzidas de cada produto, da utilização de equipamento, etc...

A Programação envolve o processo de determinação da ordem na qual as operações devem ser processadas, dando o nome de sequenciamento a esta fase. Existem algumas regras para este sequenciamento:

- Prioridade ao consumidor, onde as operações permitem as vezes que os itens para um determinado consumidor sejam processados antes de outros, independentemente da ordem de chegada do consumidor ou item,
- Prioridade pela data prometida, onde o sequenciamento ocorre de acordo com a data prometida de entrega

- FIFO, onde o sequenciamento é feito pela exata chegada dos pedidos,
- Operação mais longa/Tempo total mais longo da tarefa em primeiro, onde se deseja sequenciar as operações pelos mais longos trabalhos em primeiro lugar,
- Operação mais curta/Tempo total mais curto da tarefa em primeiro, onde a maioria das operações podem se tornar limitadas por disponibilidade de caixa e o sequenciamento pode ser ajustado para atacar os trabalhos mais curtos,

Além disso, programação também envolve a distribuição de operações necessárias pelos diversos centros de trabalho, recebendo o nome de alocação de carga. Alguns tipos de programação são:

→Gráfico de Gantt

É o método de programação mais comumente utilizado, o qual representa o tempo como uma barra em um gráfico. Como concepção, o gráfico consiste de uma tabela de dupla entrada. Cada linha corresponde a um recurso produtivo de que se dispõe: máquinas, pessoas, centros de trabalho, etc. Cada divisão vertical corresponde a unidade de tempo, como dias, semanas, meses, etc. No cruzamento coloca-se algum tipo de marcação indicando o trabalho ou operação que será feito com determinado recurso durante certo intervalo de tempo. A grande vantagem consiste na sua simplicidade, mas não é hábil para revelar os custos associados com alocações alternativas de carga e, além disto, necessita revisões continuas para atualização.

Um exemplo de gráfico de Gantt é apresentado adiante, onde desejamos obter no menor intervalo de tempo, o processamento dos itens apresentados na tabela a seguir.

Produtos	Tempos e Rotas		
	#1	#2	#3
1	1/M1	8/M2	4/M3
2	6/M2	5/M1	3/M3
3	4/M1	7/M3	9/M2

Tabela 1 – Exemplo de dados de Produção

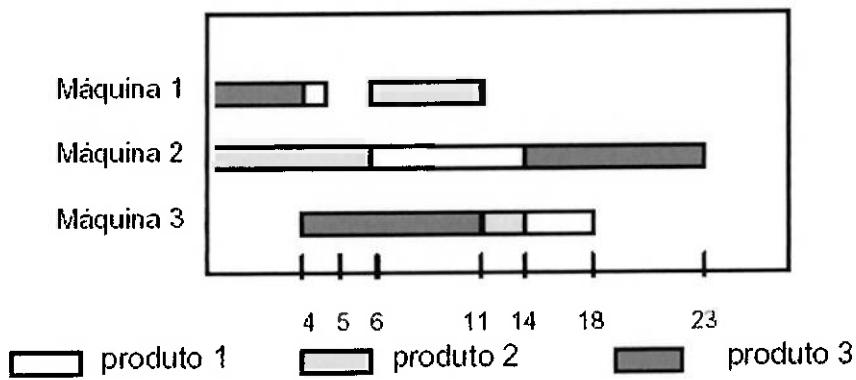


Figura 1 - Gráfico de Gantt

O gráfico então deve ser preenchido com símbolos que podem indicar a data planejada para o início das tarefas, o andamento real das mesmas, o término, explicitar quando a máquina não pode ser usada devido a manutenções ou qualquer outro problema. Desta forma, é possível realizar a programação e passá-la aos funcionários sem que sejam atribuídas mais atividades do que é possível para uma mesma unidade de produção.

↳ Gráfico de Gantt para acompanhamento de trabalho: a única diferença para o item anterior é que o eixo vertical deve conter o número dos trabalhos ao invés das máquinas, para que desta forma a visualização do andamento de cada trabalho seja facilitada.

↳ Gráfico de Gantt para carga de trabalho: esta na verdade é uma simplificação do gráfico de Gantt para distribuição dos trabalhos. São plotadas apenas a porcentagem de tempo de trabalho que alguma unidade de produção realizou, comparando esta porcentagem com a capacidade que tal unidade poderia ter realizado.

ZACCARELLI (1973) afirma que o gráfico para distribuição de trabalhos e o gráfico para andamento de trabalhos apresentam uma maneira clara de estabelecer e representar a programação. Entretanto, o próprio autor afirma que estes não são usados com muita freqüência. Isso se deve ao fato de o procedimento para preparação destes gráficos, apesar de simples, é muito trabalhoso e sujeito a erros. Cada vez que ocorrem eventos imprevisíveis como: quebra de ferramentas, falta de operários, cancelamentos de ordens de produção, entre outros, um grande número de alterações nos gráficos deverá ser feita. Cada alteração afetará um grande número de trabalhos gerando um grande retrabalho para o programador.

→Método de designação

É um modelo de Programação Linear especialmente projetado para assinalar recursos a trabalhos que devem ser, onde estes recursos podem ser pessoas ou máquinas, enquanto que os trabalhos podem ser operações, projetos, tarefas específicas, etc. Na verdade o que se deseja é uma distribuição de recursos pelos trabalhos como forma a satisfazer critérios tais como: minimização de despesas ou de tempos de operação, maximização de lucros ou eficiência, etc..

→Programação para frente e para trás

A programação para frente envolve iniciar trabalho logo que ele chega. A programação para trás envolve iniciar o trabalho no ultimo momento possível sem que ele tenha atraso. A programação par frente possui alta utilização de pessoal e é bastante flexível, uma vez que deixa folgas no sistema; já a programação para trás promove baixo custos de materiais, além de estar menos exposto a riscos no caso de mudança de programação do consumidor, focando as operações nas datas prometidas.

2.2 Balanceamento de Linha

O objetivo do balanceamento de linha é atingir a máxima eficiência ou a mínima porcentagem de tempo ocioso, sendo eficiência definida como o quociente entre o tempo de trabalho efetivo na linha e o tempo total disponível, ambos tomados na confecção de uma unidade.

O problema do balanceamento pode ser resumido nos seguintes pontos:

- Existe um certo numero n de tarefas distintas que devem ser completadas em cada unidade de produto (ou parte dele) que sai da linha;
- O tempo de execução “ t_i ” de cada tarefa “i” é conhecido e constante.
- O conteúdo de trabalho de uma unidade do produto (o tempo que um único posto de trabalho levaria para completa-lo) é dado por $T = \sum t_i$
- O objetivo do balanceamento é organizar as tarefas em grupos, alocando cada um deles a um posto de trabalho;

- O tempo de ciclo ou simplesmente ciclo é o tempo disponível, em cada posto de trabalho, para completar o grupo de tarefas ali alocado. Assim, designado por C o tempo de ciclo, tem-se que o número mínimo N de postos de trabalho é dado por $N = T/C$;

- A eficiência de uma linha de montagem é dada por:

$$Eficiencia = \frac{Volume De Pr oduca Re al}{Capacidade Efetiva}$$

Para problemas práticos, matematicamente complicados, utilizam-se métodos heurísticos que fornecem uma solução razoável do ponto de vista da eficiência, mas não necessariamente ótima. Como exemplos tem-se:

- Método de Helgelson e Birnie: este método consiste em dar peso a cada tarefa, que é igual ao seu tempo de execução somado aos tempos de execução de todas as tarefas que lhe seguem. Em seguida, as tarefas são alocadas aos postos de trabalho na ordem decrescente de seus pesos
- Método de Kilbridge e Webster: é parecido com o anterior, onde para cada tarefa é contado o número total de tarefas precedentes. São alocadas então as tarefas na ordem crescente do número de predecessores. Quando existirem duas ou mais tarefas com mesmo número de predecessores, aloca-se primeiro aquele com maior duração e assim sucessivamente.

2.3 Layout do Chão de Fábrica

O layout de uma operação produtiva preocupa-se com a localização física dos recursos de transformação. Com isto, definir o layout significa decidir onde colocar as instalações, máquinas, equipamentos, e pessoal da produção, definindo sua “forma” e aparência, além do fluxo de material e de pessoas através de uma operação.

2.3.1 Importância do Planejamento de layout

Como pode ser visto na figura seguinte, o planejamento de layout não deve ser feito independentemente do produto, do processo e da programação do chão de fabrica. Assim, é necessário que exista uma coordenação destes fatores para que o chão de fabrica atenda as expectativas desejadas.



Figura 2 - Fatores que influenciam o Layout

Os objetivos do planejamento do layout do chão de fabrica, de forma geral, são:

- minimizar os investimentos em equipamento;
- minimizar o tempo de produção total;
- utilizar eficientemente o espaço disponível;
- prover ao funcionário conveniência, conforto e segurança;
- garantir flexibilidade dos equipamentos e operações;
- minimizar o custo de manuseio de material;
- minimizar as variações e tipos de equipamentos para manuseio e materiais;
- facilitar o processo de manufatura
- facilitar a estrutura organizacional.

Os problemas a serem resolvidos pelo planejamento do layout podem ser dos seguintes tipos:

- planejamento do layout de uma nova fabrica;
- planejamento do layout de uma fabrica em expansão;
- reconfiguração do espaço disponível
- rearranjo do espaço atual.

2.3.2 Metodologia para planejamento do Layout do Chão de Fabrica

Para um correto planejamento do layout do chão de fabrica, devemos seguir os seguintes passos:

2.3.2.1 Determinação do tipo de fluxo padrão

Os tipos de fluxo padrão podem ser classificados em 2 grandes grupos: horizontais e verticais.

Para o tipo de fluxo padrão horizontal existem 5 tipos básicos de fluxo que podem ser vistos na figura seguinte

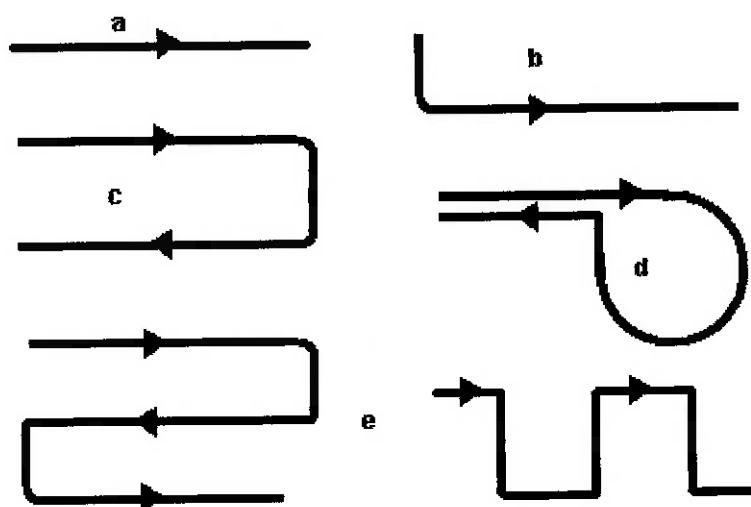


Figura 3 - Os 5 tipos básicos de fluxo

O fluxo em linha reta (a) é um dos mais simples, mas deve ser separado o recebimento de diferentes materiais e a entrega de produtos. O fluxo em forma de L (b) é utilizado quando o fluxo em linha reta não pode ser adotado na fabrica existente ou quando os custos não permitem o fluxo em linha reta. O fluxo em forma de U (c) é bastante popular, devido a sua simples administração e a facilidade de combinar atividades de recebimento de materiais e envio de produtos. O fluxo circular (d) é indicado quando se deseja terminar um fluxo em um ponto próximo a origem do mesmo. Por ultimo temos o fluxo em serpentina (e) quando a quantidade de processamentos para obtenção do produto é tão longa, que eh preciso ter um zigue-zague no chão de fabrica.

Alguns dos fatores que influem na escolha do tipo de fluxo padrão que deve ser adotado são:

- Localização da área de transferência externa;
- Número de partes do produto;
- Número de operações em cada parte;
- Seqüência de operações em cada parte;
- Número de unidades a serem produzidas;
- Quantidade de espaço disponível;
- Local de estoque de matéria prima;
- Flexibilidade desejada;
- Instalações, etc...

2.3.2.2 Determinação do tipo de Arranjo Físico (layout)

Existem 4 categorias básicas de arranjo físico: posicional, por processo, celular e por produto.

No arranjo físico posicional, os recursos transformados não se movem através dos recursos transformadores, mas sim o contrário. Isto acontece em casos onde o produto seja muito grande para ser movido de forma conveniente.

No arranjo físico por processo, o produto não mais fica estacionário. Processos similares são localizados juntos uns dos outros, e os recursos transformados percorrerão um roteiro de processo a processo, de acordo com as suas necessidades. Diferentes produtos percorrerão diferentes roteiros através da operação.

No arranjo físico celular os recursos transformados, entrando na operação, são pré-selecionados para movimentar-se para uma parte específica da operação, na qual todos os recursos transformadores necessários a atender as suas necessidades imediatas de processamento se encontram. Cada célula pode ter um layout específico e outros produtos transformados podem se mover de célula para célula.

No arranjo físico por produto, os recursos transformadores localizam-se segundo a melhor conveniência do recurso que está sendo transformado. Cada recurso transformado segue um roteiro predefinido no qual a seqüência das atividades requerida coincide com a seqüência na qual os processos foram arranjados fisicamente.

É necessário ressaltar que os arranjos físicos citados anteriormente podem ser combinados de forma a atender as necessidades e características da fabrica.

O fluxo de pessoas, informações e materiais através da operação são determinados pelo tipo de arranjo físico escolhido, onde temos num extremo o arranjo físico posicional (fluxo intermitente) e no outro o arranjo físico por produto (fluxo mais contínuo).

A tabela seguinte indica as vantagens e desvantagens dos tipos básicos de arranjo físico:

	Vantagens	Desvantagens
Posicional	Flexibilidade de mix e produto muito alta. Produto ou cliente não movido ou perturbado. Alta variedade de tarefas para a mão-de-obra	Custos unitários muito altos Programação de espaço ou atividades pode ser complexa. Pode significar muita movimentação de equipamentos e mão-de-obra
Processo	Alta flexibilidade de mix e produto relativamente robusto em caso de interrupção de etapas Supervisão de equipamento e instalações relativamente fácil	Baixa utilização de recursos Pode ter alto estoque em processo ou filas de cliente Fluxo complexo pode ser difícil de controlar
Celular	Pode dar um bom compromisso entre custo e flexibilidade para operações com variedade relativamente alta Atravessamento rápido Trabalho em grupo pode resultar em melhor motivação	Pode ser caro reconfigurar o arranjo físico atual Pode requerer capacidade adicional Pode reduzir níveis de utilização de recursos
Produto	Baixos custos unitários para altos volumes Da oportunidade para especialização de equipamento Movimentação de clientes e materiais conveniente	Pode ter baixa flexibilidade de mix Não muito robusto contra interrupções Trabalho pode ser repetitivo

Tabela 2 - Vantagens e desvantagens dos tipos de arranjo físico

2.3.3 Considerações finais sobre o planejamento do arranjo físico

O planejamento de arranjo físico é uma atividade importante na maioria dos tipos de produção, pois:

- É uma atividade difícil e de longa duração devido às dimensões físicas dos recursos de transformação movidos;
- O re-arranjo físico de uma operação existente pode interromper seu funcionamento levando a perdas na produção;
- O desenvolvimento de um arranjo físico pode levar a padrões de fluxos longos e confusos, estoques de materiais, etc... o que gera alto custo.

3. Metodologia de desenvolvimento de Projeto de Software

Para o desenvolvimento do software que será elaborado como forma de implementarmos o estudo desejado, é necessária a utilização de uma metodologia baseada nos paradigmas da Engenharia de Software.

Assim, este capítulo visa o estudo destes diversos paradigmas, assim como a escolha de um deles para o futuro desenvolvimento do software em questão.

3.1 Introdução

A engenharia de software visa o estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais. Assim sendo, a engenharia de software abrange um conjunto de três elementos fundamentais: métodos, ferramentas e procedimentos.

Os métodos de engenharia de software proporcionam os detalhes de como fazer para construir o software. Os métodos envolvem um amplo conjunto de tarefas que incluem: planejamento e estimativa de projeto, análises de requisitos de software e sistemas, projeto de estrutura de dados, arquitetura de programas e algoritmo de processamento, codificação, teste e manutenção.

As ferramentas de engenharia de software proporcionam apoio para sustentar todos os métodos citados anteriormente. Como exemplo temos as ferramentas CASE (Computer-Aided Software Engineering), que combina software, hardware e um banco de dados de engenharia de software para criar um ambiente de engenharia de software.

Os procedimentos da engenharia de software constituem o elo de ligação entre os métodos e as ferramentas para o desenvolvimento do software. Os procedimentos definem as seqüências em que os métodos serão aplicados, os produtos que deverão ser entregues, os controles que ajudam a assegurar a qualidade e a coordenar as mudanças, e os marcos de referência que possibilitam os gerentes de software a avaliar o progresso.

Assim sendo, a engenharia de software compreende um conjunto de etapas que envolvem os métodos, ferramentas e procedimentos citados anteriormente. Estas etapas podem também ser chamadas de paradigmas de engenharia de software, que deve ser escolhido para o projeto do mesmo.

3.2 O Ciclo de vida Clássico

Este paradigma, requer uma abordagem sistemática, seqüencial ao desenvolvimento do software, que se inicia no nível do sistema e avança ao longo da análise, projeto, codificação, teste e manutenção.

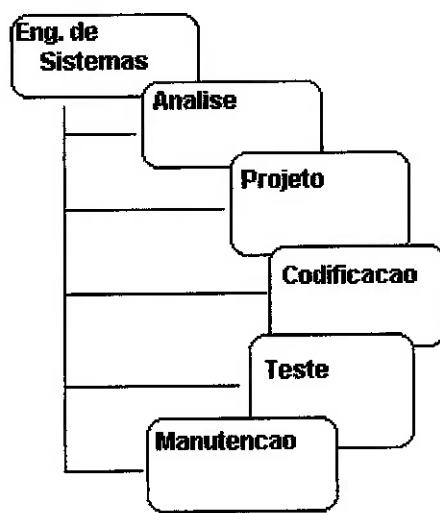


Figura 4 - Ciclo de Vida Clássico

Ele é tratado a partir do ciclo de engenharia convencional, e requer um tratamento segmentado e hierarquizado das seguintes atividades:

- Análise e engenharia de sistemas, que envolve a análise de requisitos em nível do sistema, com uma pequena quantidade de projeto e análise em alto nível;
- Análise de requisitos de software, que envolve o entendimento do programa a ser construído;
- Projeto que inclui a escolha da estrutura de dados, arquitetura de software, detalhes procedimentais e caracterização da interface.
- Codificação, que envolve a tradução do projeto em uma forma legível para a máquina;
- Testes que garantem que todas as funções implantadas tenham sido testadas, identificando erros que devem ser corrigidos para garantir a qualidade do software;

- Manutenção, que envolve a melhoria contínua do software de acordo com as exigências do cliente, assim como a adaptação do software a variações do ambiente externo.

Este é o paradigma de engenharia de software mais utilizado, pois possui alta aplicabilidade e garante que o projeto realmente chegue ao seu final. Porém, algumas dificuldades podem surgir com este paradigma:

- Os projetos freqüentemente têm interações tornando difícil manter um fluxo seqüencial;
- Para o paradigma funcionar é necessário que o usuário defina todas as suas necessidades no inicio;
- O custo da manutenção de problemas é relativamente alto, pois o usuário somente verá o sistema depois de pronto.

3.3 Modelo Espiral

Este paradigma, conforme pode ser visto na figura seguinte, define quatro importantes atividades representadas pelos quatro quadrantes da figura:

- Planejamento, que inclua a determinação de objetivos, alternativas e restrições;
- Analise de riscos, que inclua a analise de alternativas e identificação/resolução dos riscos;
- Engenharia, que é o desenvolvimento do produto em um próximo nível;
- Planejamento do Ciclo seguinte, que abrange a avaliação do cliente, que é a avaliação dos resultados da engenharia

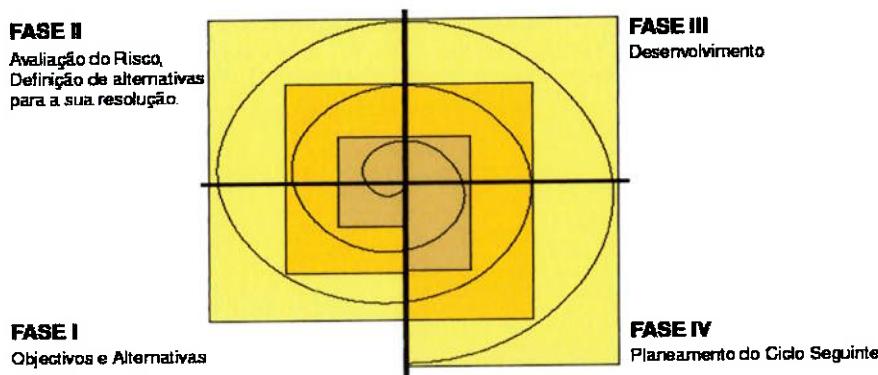


Figura 5 - O modelo espiral

Assim, este paradigma tem ênfase em um processo iterativo de desenvolvimento do software, onde ele introduz um ciclo de prototipagem iterativa. A cada iteração ao redor da espiral são construídas versões mais completas do software, e a evolução obtida é examinada para verificar se os progressos estão de acordo com os objetivos (análise de riscos).

Como nos paradigmas estudados anteriormente, algumas dificuldades podem ocorrer com a sua utilização:

- Pode ser difícil convencer grandes clientes de que o tratamento evolutivo é controlável.
- É necessária uma grande experiência na avaliação dos riscos, de forma a se obter sucesso no emprego deste paradigma.

4. Controle de Produção em uma Estamparia

Nosso trabalho será desenvolvido com base em um Controle de Produção já existente, onde é feito apenas o apontamento da produção dos funcionários manualmente, sendo o nosso objetivo aqui obter posteriormente via software, dados da produção que nos possibilite encontrar erros e propor melhorias no sistema.

O Controle de Produção ao qual nos referimos e que encontra-se em anexo, possui os seguintes campos a serem preenchidos pelos funcionários (exemplo para o primeiro turno):

N.º da Maquina _____ (1) _____ TON. _____ (2) _____

Data ____ / ____ / ____

Produto	Pcs/bat	Hora	RE	Pcs/hora	Produção	Parada	Cod. Par.	Acumulado
		05:50 - 07:00						
		07:00 - 08:00						
		08:00 - 09:00						
(3)	(4)	09:00 - 10:00	(5)	(6)	(7)	(8)	(9)	(10)
		10:00 - 11:00						
		11:00 - 12:00						
		12:00 - 13:00						
		13:00 - 14:10						

- 1- Número da máquina
- 2- Quantidade de toneladas da máquina (prensa)
- 3- Código do Produto que esta sendo batido no período indicado
- 4- Quantidades de peças que saem da ferramenta por batida
- 5- Registro do Funcionário
- 6- Quantidade de peças por hora que o funcionário deve bater para atingir uma eficiência de 100%, segundo cronometragem anteriormente feita e registrada no sistema.
- 7- Quantidade de batidas que o funcionário realizou no período correspondente.
- 8- Tempo que o funcionário permaneceu parado sem estar operando na máquina
- 9- Código da parada do funcionário, que indica o porquê da parada.
- 10- Produção acumulada, considerando períodos anteriores

Estas folhas são colocadas sempre no início do primeiro turno e retiradas no final do terceiro, e depois de feito isso é feita uma análise visual das mesmas para que seja identificado algum eventual problema em uma máquina ou mesmo a baixa eficiência de um funcionário.

Nosso objetivo aqui é fazer uma análise via software destes dados para verificarmos a eficiência da empresa de forma mais detalhada.

Via software é possível criar também um banco de dados que possibilite estender a análise do dia para uma análise do mês, e verificar quais das ocorrências são tidas como sazonais e quais podem ser encontradas de forma freqüente no chão de fabrica.

Assim, por este método, poderemos identificar através da eficiência o efeito das seguintes atividades no processo de estamparia:

- Troca de ferramenta: identificar o tempo gasto e a ocorrência de trocas simultâneas de ferramenta
- Manutenção Preventiva: efeito da aplicação da manutenção preventiva nas máquinas, reduzindo suas paradas o que abaixa a eficiência do sistema.
- Transporte de materiais: efeito da falta de material na reposição de matéria prima ou mesmo na retirada de material acabado.
- Ferramentaria: paradas na produção por defeitos na ferramenta

Tais dados poderão ser retirados com base nos dados apontados pelos funcionários no Controle de Produção, conforme iremos descrever de forma mais detalhada no capítulo seguinte, a ser implementado via software.

5. Formulação teórica da eficiência com base no Controle de Produção

Inicialmente definimos como Capacidade da Ferramenta (CF) como sendo o número de peças processadas (P) pela ferramenta por batida (B), assim temos:

$$CF = \frac{P}{B} \text{ (peças/batida)}$$

E possuímos também a capacidade da prensa (CP), definida como o número de peças batidas por um dado intervalo de tempo, assim temos:

$$CP = \frac{P}{t} \text{ (peças/hora)}$$

Entretanto, no processo de estamparia e no Controle de Produção, não obtemos de forma direta o número de peças processadas, mas sim o número de batidas realizadas pela máquina. Substituindo temos:

$$CP = CF \cdot \frac{B}{t} \text{ (peças/hora)}$$

Assim, a capacidade da prensa nos indicará quantas peças podem ser processadas em determinada prensa com base na capacidade da ferramenta utilizada e no tempo requerido.

Feito isso, definimos agora a Necessidade da Peça (NP) como sendo:

$$\begin{aligned} NP &= (CP)^{-1} \\ NP &= \left(CF \cdot \frac{B}{t} \right)^{-1} \\ NP &= \frac{t}{CF \cdot B} \text{ (horas / peça)} \end{aligned}$$

Assim, NP nos indica o tempo gasto para o processamento de uma peça.

Entretanto, o número de peças produzidas nem sempre é o ideal e daí a importância de tirarmos a eficiência do sistema.

A eficiência da estamparia será dada portanto por:

$$Ee = \frac{\sum NP_i \cdot ni}{Tc}$$

onde:

Npi – necessidade de cada uma das peças processadas

Ni – número de peças processadas para cada tipo de peça

Tc – horas contratadas para a estamparia

O valor de E em geral varia entre 70 e 85%, entretanto nosso objetivo aqui é extrair não apenas a eficiência geral do dia da estamparia dado por E, mas sim outras eficiências com base no apontamento do Controle de Produção

Considerando o tempo que o funcionário permanece efetivamente trabalhando na máquina (Tf), como sendo o tempo de funcionamento da máquina (Tm) menos os tempos de paradas (Tp), temos:

$$Tf = Tm - Tp$$

Assim, temos a eficiência do funcionário dada por:

$$Ef = \frac{\sum NP_i \cdot ni}{Tf}$$

$$Ef = \frac{\sum NP_i \cdot ni}{Tm - Tp}$$

Para cada um dos períodos indicados na folha de Controle de Produção, possuímos os dados necessários para se calcular portanto a Eficiência do Funcionário, sendo eles:

NP – necessidade da peça retirada das equações acima descritas

ni – número de peças produzidas

Tm – Tempo da máquina

Tp – Tempo de paradas indicado juntamente com seu respectivo código de parada

Entretanto, percebemos pela folha de Controle de Produção, que nem todos os períodos de trabalho correspondem à uma hora de trabalho, tendo assim que ser normalizados para efeito de cálculo. Assim, normalizados, teríamos os seguintes tempos para cada um dos períodos:

Período		Duração (horas)
1	05:50 - 07:00	1.167
2	07:00 - 08:00	1
3	08:00 - 09:00	1
4	09:00 - 10:00	1
5	10:00 - 11:00	1
6	11:00 - 12:00	1
7	12:00 - 13:00	1
8	13:00 - 14:10	1.167
9	14:10 - 15:00	0.833
10	15:00 - 16:00	1
11	16:00 - 17:00	1
12	17:00 - 18:00	1
13	18:00 - 19:00	1
14	19:00 - 20:00	1
15	20:00 - 21:00	1
16	21:00 - 22:26	1.433
17	22:26 - 23:00	0.567
18	23:00 - 00:00	1
19	00:00 - 01:00	1
20	01:00 - 02:00	1
21	02:00 - 03:00	1
22	03:00 - 04:00	1
23	04:00 - 05:00	1
24	05:00 - 05:46	0.767

Tabela 3 - Horas por período

Ainda podendo incluir mais uma equação que diz respeito à utilização da máquina, que poderá ser dada por:

$$\text{Utilização} = (\text{horas trabalhadas}) / (\text{horas disponíveis})$$

Sendo horas disponíveis igual ao número de horas trabalhadas mais o tempo de paradas da máquina.

A seguir apresentamos uma descrição dos principais motivos de parada encontrados em uma estamparia.

6. Motivos de Parada

6.1 Ferramentaria

6.1.1 Constituição de uma ferramenta

Como forma de analisarmos as principais falhas que apresentam uma ferramenta no processo de estamparia iremos apresentar um esquema simplificado, de modo que possamos visualizar seus constituintes básicos e que serão citados adiante como motivos de parada da máquina.

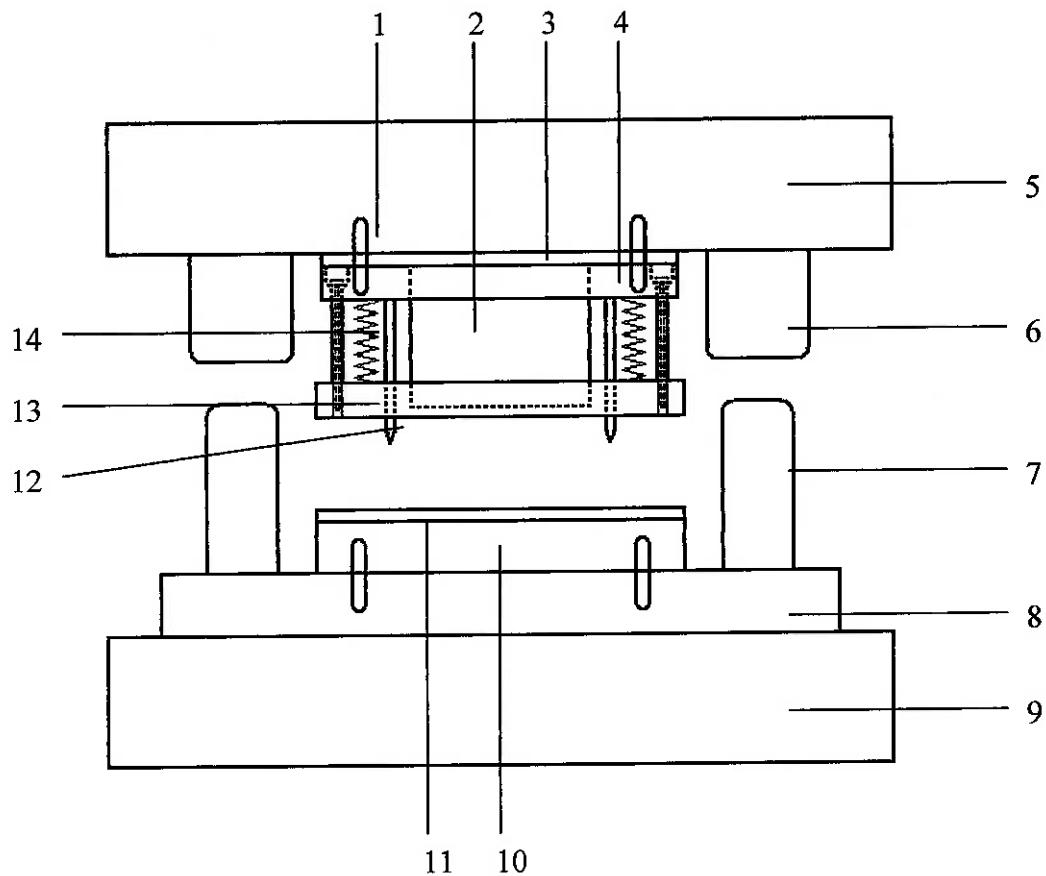


Figura 6 – Elementos básicos de uma ferramenta

Sendo:

1. Pino Guia
2. Punção
3. Placa de Coque
4. Porta Punção

5. Base Superior
6. Bucha
7. Coluna
8. Base Inferior
9. Paralelo
10. Matriz
11. Guia da Tira
12. Piloto
13. Prensa-chapa (pisador)
14. Mola

6.1.2 Motivos de Parada por Ferramentaria

Tendo demonstrado as principais partes de uma ferramenta, podemos agora apresentar quais delas estão relacionadas com maior freqüência a paradas da máquina por motivos de ferramentaria.

Na Folha de Controle de Produção, existe em seu verso um espaço para o preenchimento das justificativas das paradas encontradas, e para as paradas por ferramentaria são apontadas as seguintes justificativas:

- Peça fora de especificação
- Rebarba na peça
- Matriz Quebrada
- Punção quebrado
- Pino Guia Quebrado
- Mola Quebrada
- Outros

Estes motivos servirão como alternativa de motivos de parada por ferramentaria no desenvolvimento do software proposto.

6.2 Setup

Grande parte das pesquisas considera os tempos de setup como não relevantes ou de pequena variabilidade e, geralmente, os incluem nos tempos de processamento. Esse

procedimento simplifica muito a análise em determinadas aplicações, principalmente quando os tempos de setup são consideravelmente menores que os tempos de processamento, ou em casos em que o setup destina-se à produção de lotes e é executado somente uma vez para um grande lote de produção.

No entanto, para os casos em que os tempos de setup apresentam razão significativa diante dos tempos de processamento, há a necessidade de tratá-los diferenciadamente, uma vez que eles têm relação direta com a disponibilidade de equipamentos e acarretam custos específicos, como a necessidade de pessoal especializado para sua execução. O tratamento em separado dos tempos de setup pode levar, com a otimização do critério de desempenho adotado, a melhorias no atendimento à demanda e à facilidade no gerenciamento do sistema de produção.

6.2.1 Sistemas de Troca Rápida (STR)

Uma forma de reduzirmos o tempo de Setup é adotando os chamados Sistemas de Troca Rápida (STR), onde a forma como é presa a ferramenta, é modificada utilizando-se dispositivos que facilitam esta tarefa, e que por consequência melhoram o ajuste da ferramenta, aumentando a sua vida e reduzindo o número de falhas

Existem diversos tipos de Sistemas de Troca Rápida, sendo que na empresa onde foi feito este estudo de caso, encontramos para uma das prensas o seguinte sistema.

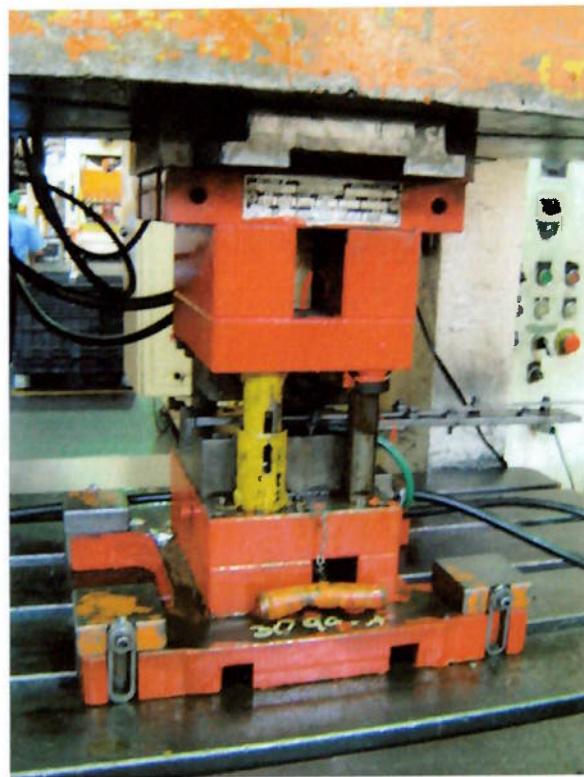


Figura 7 – Ferramenta com STR

Nesta figura, podemos reparar que o sistema de fixação tradicional por parafusos da ferramenta foi substituído, sendo que agora ela é fixada através de um perfil na parte superior e através de blocos localizadores na parte inferior.

Esta ferramenta trabalha em uma Prensa Guttmann de 160 Toneladas, E o STR superior da ferramenta pode ser visto em maior detalhe na figura seguinte.



Figura 8 – Perfil Superior de Ferramenta com STR

A Ferramenta fora da prensa, pode ser vista na figura seguinte, onde observamos o perfil de encaixe da mesma.



Figura 9 – Ferramenta com STR isolada

E como foi dito anteriormente, não apenas a parte superior tem o seu dispositivo de fixação facilitado, mas também a inferior, como mostrado na figura seguinte.



Figura 10 – Fixação inferior de Ferramenta com STR

Os Sistemas de Troca Rápida podem variar de forma significativa, e como já dissemos anteriormente, apenas recebem esta definição por possuírem algum mecanismo que facilite a troca da ferramenta e que aumente o seu ajuste.

Um STR simples pode ser visto na figura seguinte, onde a fixação da ferramenta na prensa é feita através de um fixador que é acionado por um único movimento do preparador da máquina.



Figura 11 – Sistema STR Simples

Estes dois dispositivos aos quais nos referimos, fazem parte de prensas onde o tempo de Setup pode atingir um máximo de uma hora e meia.

Entretanto, existem ferramentas de maior porte onde o tempo de Setup pode atingir proporções bem maiores. Tais ferramentas também possuem os chamados Sistemas de Troca Rápida.

Um exemplo de STR para prensas de grande porte pode ser visto na figura seguinte.



Figura 12 – Sistema STR em máquina Transfer

Nesta figura observamos que a base da prensa é móvel, e assim, a ferramenta pode ser preparada quando a máquina ainda está em operação com outra ferramenta e em outra base.

Desta forma, o tempo que estariamos perdendo por setup, está sendo gasto com a máquina operando e batendo outro produto.

Quando a base móvel chega no local de trabalho, esta é fixada por dispositivos pneumáticos mostrados na figura seguinte.



Figura 13 – Fixação Inferior de STR - Transfer

Após a fixação da base inferior, a base superior da ferramenta é então presa com parafusos pneumáticos, como mostrado na figura seguinte.

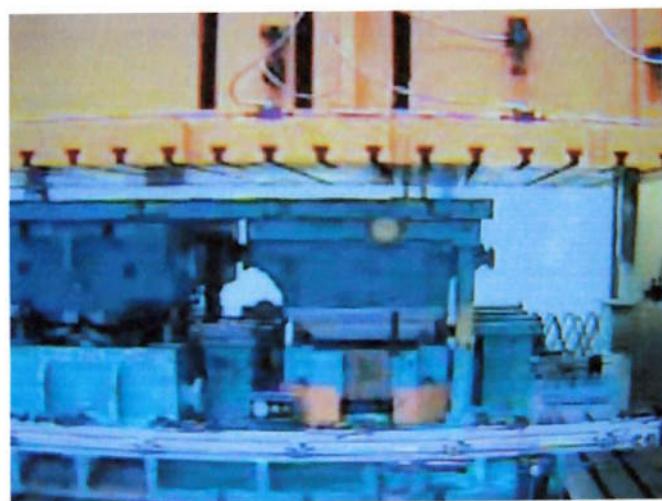


Figura 14 – Sistema STR – Transfer Superior

Mais adiante mostraremos em um estudo de caso, de que forma o setup pode reduzir o tempo de setup de uma máquina, e qual o impacto desta redução na eficiência e utilização das máquinas da estamparia.

6.3 Manutenção Mecânica/Elétrica

As atividades de manutenção existem para evitar a degradação dos equipamentos e instalações, causada pelo seu desgaste natural e pelo uso. Esta degradação se manifesta de diversas formas, desde a aparência externa ruim dos equipamentos, até perdas de desempenho e paradas da produção, fabricação de produtos de má qualidade e poluição ambiental.

Todas estas manifestações têm uma forte influencia na qualidade e produtividade e acabam colocando em risco a sobrevivência da empresa.

6.3.1 - Manutenção Produtiva Total

A Manutenção Produtiva Total tem como objetivo a constituição de uma estrutura empresarial que vise a máxima eficiência do sistema de produção, criando-se mecanismos para prevenir as diversas perdas, atingindo zero acidente, zero defeito e zero quebra/falha, tendo como objetivo o ciclo total de vida útil do sistema de produção, com o envolvimento de todos os departamentos, iniciando-se pela produção e se estendendo aos demais setores, contando-se com a participação desde a alta administração até os colaboradores de primeira linha, atingindo-se a perda zero por meio de atividades sobrepostas de pequenos grupos.

A filosofia da MPT, está baseada em princípios de aproveitamento das pessoas, aproveitamento dos equipamentos e aproveitamento total da organização, ou seja, uma reestruturação da cultura organizacional através de uma total participação das pessoas e do aperfeiçoamento dos equipamentos, com o objetivo de máxima geração de resultados com o menor custo. Para que possamos atingir essa eficiência global é preciso que identifiquemos as perdas existentes. A JIPM (Japan Institute of Plant Maintenance) classifica a ineficiência dentro de 16 grandes perdas:

→ 8 perdas relacionadas a equipamentos, como perda por falhas em equipamentos, perda por troca de serviços, perda por troca de lâminas de corte e gabaritos, perda por acionamento, perdas por pequenas paradas e operação em vazio, perda por velocidade, perda oriunda de defeitos e retrabalhos e perda por desligamento (desacionamento);

→ 4 perdas relacionadas a mão de obra, sendo perdas por controle, perdas por organização inadequada, perdas por movimento, perdas por deficiência logística e perdas por medições e ajustes;

→ 4 perdas relacionadas a material: perdas de rendimento de material, perdas por moldes, ferramentas e gabaritos, perda relacionada a energia e perdas por desperdício de energia

Conforme a aplicação da metodologia da MPT, a eliminação das perdas com o objetivo da maximização do rendimento global, podem obter resultados de produtividade, qualidade, custo, delivery, Segurança, motivação. A MPT se inicia no pilar de manutenção autônoma e é que envolve o maior número de pessoas, é preciso ter uma manutenção estruturada para dar suporte a manutenção autônoma. Essa manutenção estruturada no programa se corresponde ao pilar de manutenção planejada, que tem como tarefa ser eficiente em planejamento, padronização e flexibilização, através de presteza e a versatilidade, aumento do nível de conhecimento técnico da equipe, aumento do nível das técnicas e do pessoal de manutenção e redução de custos, aumentando-se o nível de conhecimento gerencial ao orçar os custos de manutenção;

A implementação da estrutura de manutenção planejada se estabelece em 7 etapas.

- Etapa 1: Avaliar o equipamento e compreender a situação atual
- Etapa 2: Trazer os equipamentos as condições originais
- Etapa 3: Criar um sistema de gestão das informações
- Etapa 4: Estruturação da manutenção periódica
- Etapa 5: Estruturação da manutenção baseada na condição
- Etapa 6: Avaliação do aumento da confiabilidade , manutenibilidade e melhoria na otimização da manutenção x custos.
- Etapa 7: Utilizar o equipamento no limite

7. Resultados Obtidos

Utilizando-se o software que será apresentado mais adiante, pudemos colher as informações que serão apresentadas a seguir.

Para a obtenção dos dados relativos a eficiência, paradas e utilização das máquinas, no nosso estudo de caso de estamparia, definimos uma linha piloto onde centraríamos a nossa atenção, de forma a obter os dados com uma maior confiabilidade.

Um critério utilizado para a escolha de um grupo de máquinas, foi que estas tivessem capacidades semelhantes de trabalho, e que assim pudéssemos analisar os problemas encontrados de uma forma geral no grupo.

Assim, as máquinas analisadas foram 8 prensas Fagores de 250 Tonelas. O período de análise foi de 15 de Setembro à 11 de Novembro do ano de 2004, sendo que o período de funcionamento das máquinas é de 21 horas (24 horas/dia descontadas 3 de refeição) e 7 dias por semana.

Este período totalizou 56 dias de análise (já descontados os feriados existentes), totalizando 4704 horas disponíveis, sendo estas correspondentes às 24 horas/dia subtraídas 3 horas de refeição de cada um dos 3 turnos de trabalho.

7.1 Resultados Gerais

Neste período, o total de paradas foi de 2272,4 horas, resultando em uma utilização de 75,84% das máquinas.

Levando-se em conta o número de peças produzidas, onde entra a eficiência dos funcionários, a eficiência da linha foi de 68,21%.

A proporção em que ocorreu cada uma das paradas é mostrada no gráfico seguinte:

Paradas Gerais versus Motivo

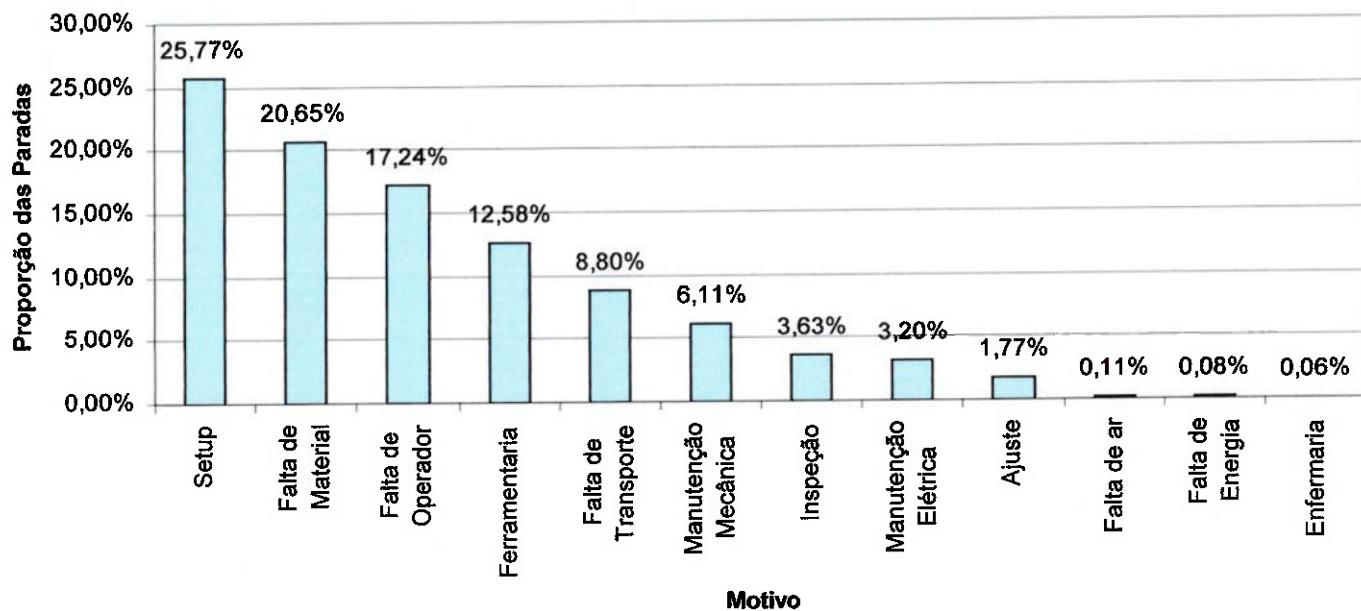


Figura 15 – Paradas Gerais da Estamparia

7.2 Ferramentaria

Pela Proporção das paradas e pelas paradas totais da linha no período analisado, apenas as paradas por ferramentaria, totalizaram 285,92 horas.

Como já citamos anteriormente, as paradas por ferramentaria foram subdivididas nas seguintes paradas:

- Rebarba na Peça
- Puncão Quebrado
- Pino Guia Quebrado
- Peça Fora de Especificação
- Matriz Quebrada
- Mola Quebrada
- Outros

E dentro dos motivos de parada por ferramentaria, as paradas se subdividiram da seguinte forma:

Paradas por ferramentaria versus Motivo

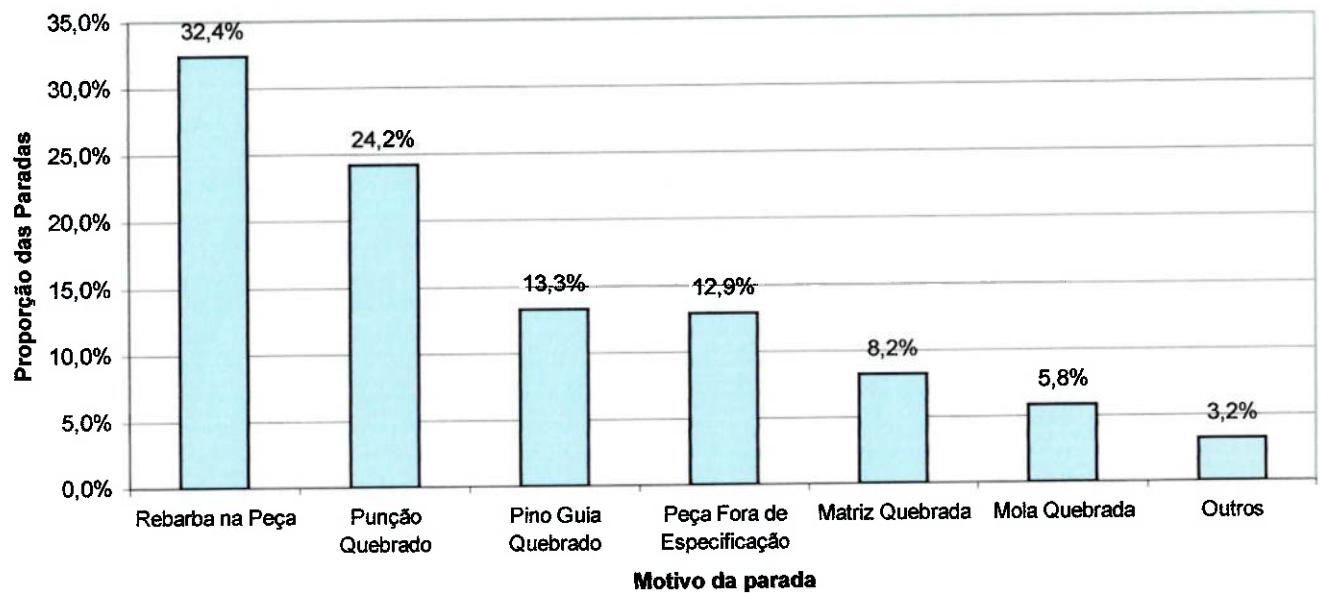


Figura 16 – Paradas de Ferramentaria

Um grande problema nas paradas por ferramentaria é o fato dela modificar de forma drástica o Planejamento e Controle da Produção.

O Gráfico de Gantt seguinte nos mostra o efeito de uma parada por ferramentaria na Programação da Produção.

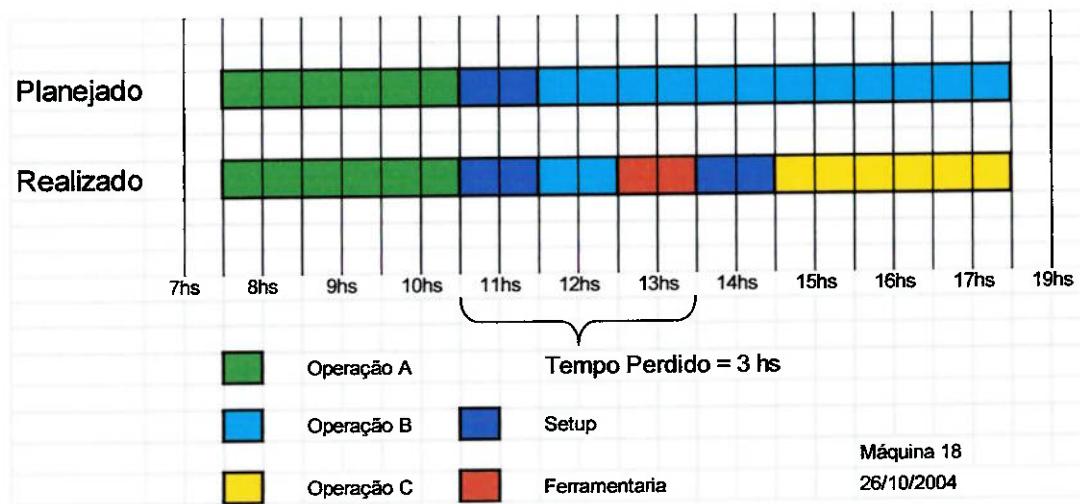


Figura 17 – Gráfico de Gantt - Ferramentaria

Assim, a máquina 18 que estava preparada para realizar uma operação A e a seguir realizar uma operação B, teve de ser reprogramada e receber uma operação C.

Tal fato é responsável não apenas por paradas de ferramentaria, mas por outros que sofrem suas consequências como um efeito cascata. Caso uma máquina estivesse programada para receber o material B, está deveria então de ser reprogramada e assim surgem problemas do tipo Falta de material, que em nosso estudo de caso representou 20,65% das paradas.

Uma forma de tornar uma parada por ferramentaria um pouco menos agressiva à programação da produção é identificar o quanto antes um defeito na peça, e tentar reduzir ao máximo o tempo de setup, assunto este que iremos abordar no próximo item.

7.3 Setup

Na Linha Piloto, onde colhemos os dados apresentados, fizemos também um estudo específico a respeito do tempo de setup das máquinas. Para isso, realizamos um Controle Estatístico de Processo para 50 trocas, onde obtivemos os seguintes resultados.

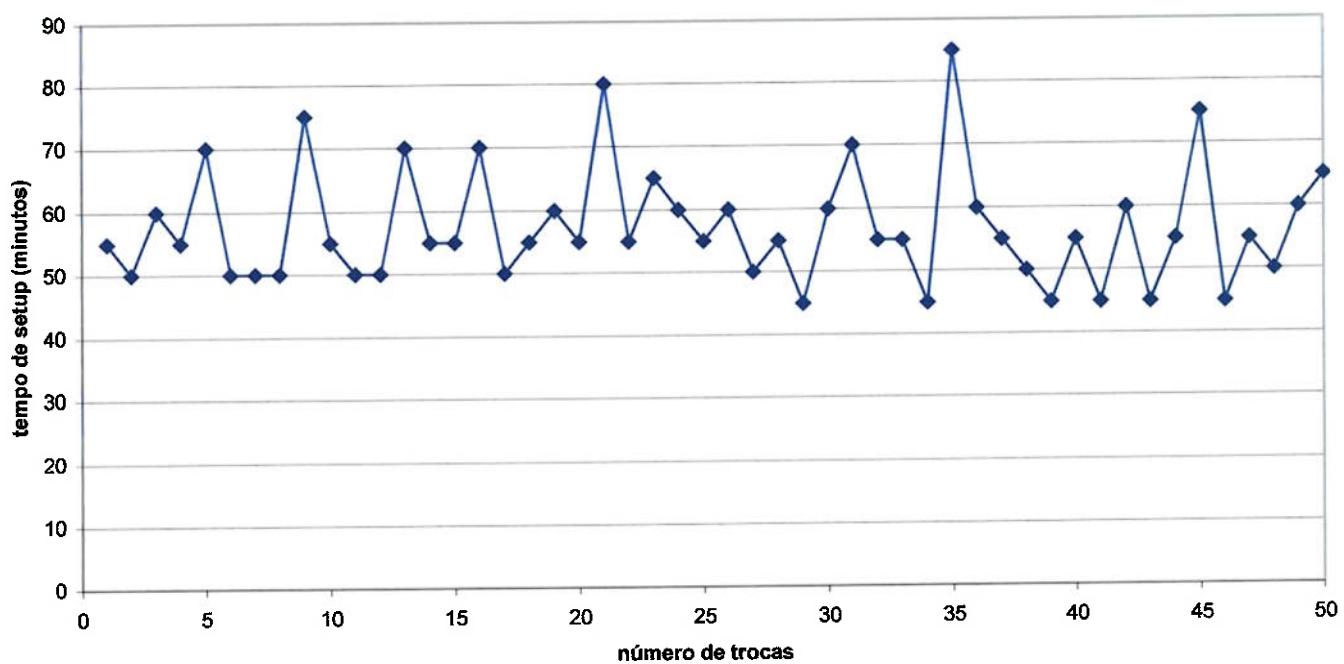


Figura 18 – Controle Estatístico de Processo para Setup

- Tempo médio de Setup: 57,10 minutos
- Desvio Padrão: 9,32 minutos

Sendo o tempo de setup o maior fator de parada das máquinas, desenvolvemos um estudo de qual impacto teríamos na eficiência da linha se adotássemos os chamados Sistemas de Troca Rápida para todas as prensas analisadas.

Para isso, realizamos novamente um Controle Estatístico de Processo só que para uma outra máquina que possui o STR ainda em implementação.

A máquina analisada foi uma Prensa Guttmann de 160 Toneladas, e seu sistema de troca rápida é aquele apresentado anteriormente quando tratamos deste assunto de forma específica.

Estando este sistema ainda em implementação em tal máquina, pudemos fazer a análise de trocas feitas com e sem STR, o que nos forneceu os seguintes dados:

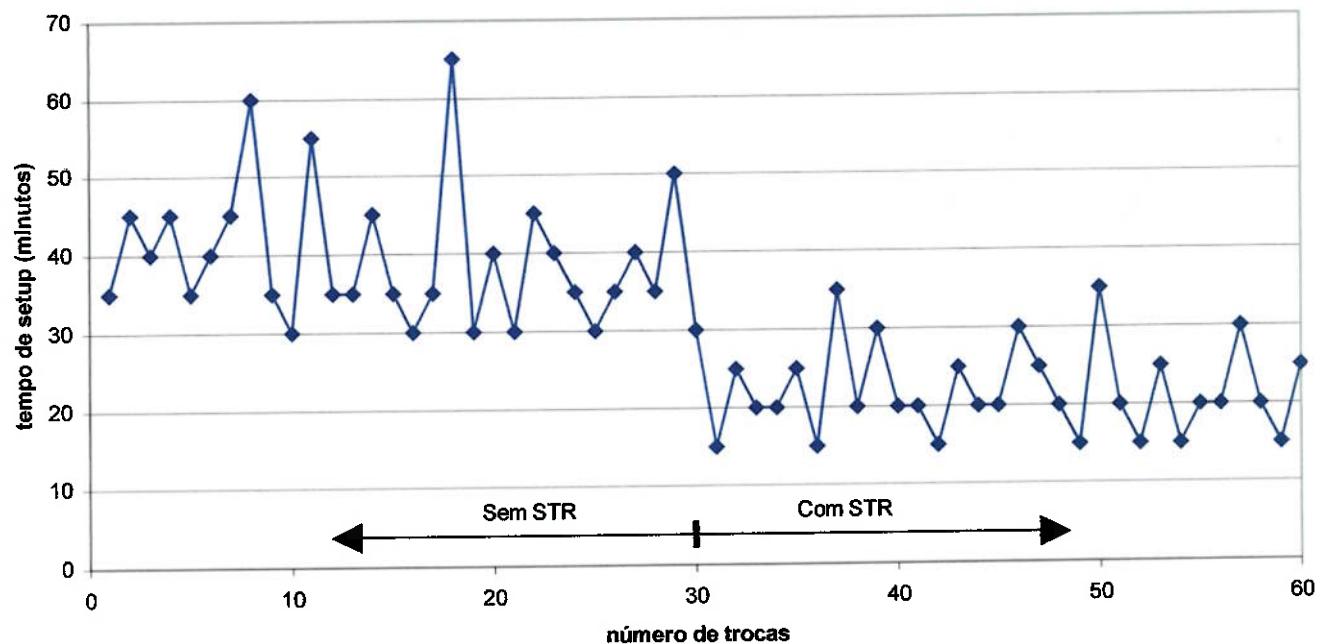


Figura 19 – Controle Estatístico de Processo para STR

	Tempo Médio de Setup (min)	Desvio Padrão (min)
Sem STR	39,50	8,94
Com STR	21,83	5,80

Tabela 4 – Tempo Médio de Setup

Um dado que nos chama a atenção, não é apenas a redução do tempo médio de setup, mas também a redução do desvio padrão do tempo de trocas.

Isso nos indica, que a adoção dos sistemas de troca rápida também auxiliam no sentido de tornar mais precisa a Programação da Produção, por tornar o tempo de setup mais definido e previsível.

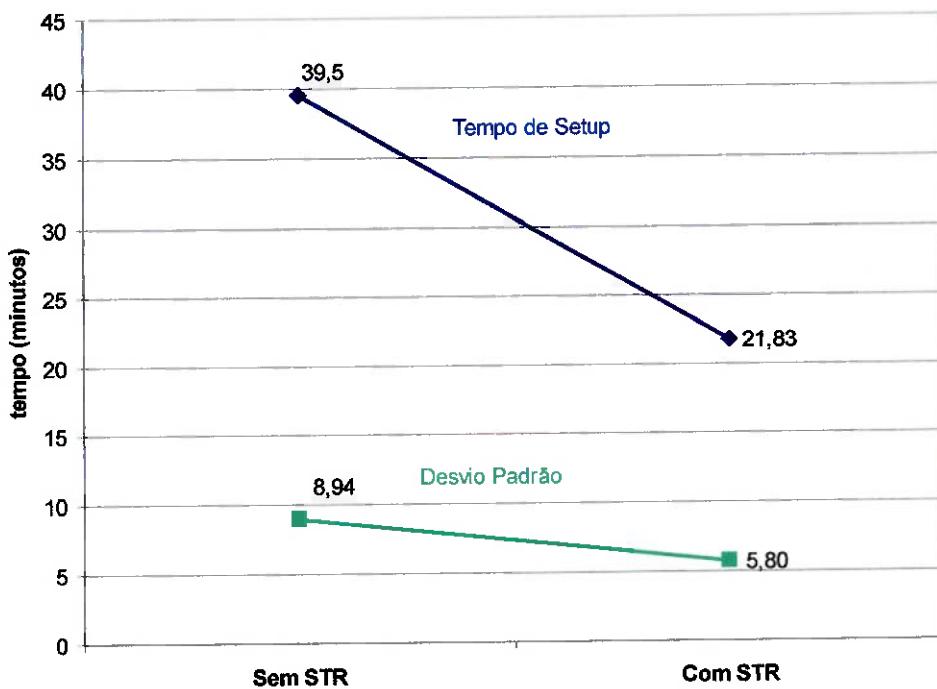


Figura 20 – Comparação de tempo para sistema STR

Apenas como um efeito comparativo, se tivéssemos tal redução no tempo de setup para a Linha Piloto analisada (43,24%), teríamos uma redução no tempo de setup de 585,7 horas para 332,44 horas.

Isso representaria, para esta linha, um aumento na utilização das máquinas de 75,84 para 78,53%.

7.4 Manutenção Mecânica/Elétrica

Como pode ser visto anteriormente, as paradas de manutenção mecânica e elétrica, somadas, resultaram em 9,31% das paradas, totalizando 211,56 horas paradas.

Este valor resulta em uma perda mensal de 105,7 horas de paradas de máquina apenas para manutenção, equivalente à 13 turnos de trabalho.

O fato de tais máquinas pararem na maioria dos casos apenas para sofrerem manutenção corretiva, vem a nos afirmar a importância de se aplicar um plano de Manutenção Preventiva Total (TPM), como explicado anteriormente, para que pudéssemos assim colher resultados a longo prazo, o que não foi possível no desenvolvimento deste trabalho e que fica portanto como uma sugestão para um trabalho futuro.

8. Software de Análise da Eficiência

Devido algumas limitações de recursos do software Excel da Microsoft, tivemos que desenvolver nosso próprio programa para que pudéssemos colher e processar os dados necessários.

Para desenvolvêrmos o software de análise, escolhemos a linguagem de programação Visual Basic, utilizando um banco de dados em access.

A seguir apresentamos as janelas de nosso programa e o código do mesmo em anexo.

The screenshot shows a Windows application window titled 'POLI Eficiência'. On the left, there's a logo with the text 'POLI' and 'Eficiência'. The main area has a title 'Período de Análise:' with fields 'De' and 'à'. Below it are two buttons: 'Calcular' and 'Controle de Produção'. The main table displays data in two columns: 'Paradas' and 'Paradas - Ferramentaria'. The 'Paradas' column lists various reasons for stops, each with its duration in hours and a percentage. The 'Paradas - Ferramentaria' column lists specific tool-related stops with their durations and percentages. At the bottom of the table, there are summary values: 'Total de Paradas', 'Total de Horas Disponíveis', 'Utilização das Máquinas (%)', and 'Eficiência (%)'.

Paradas	(horas)	(%)	Paradas - Ferramentaria	(horas)	(%)
Falta de Material	000	00	Peça Fora de Especificação	000	00
Manutenção Elétrica	000	00	Rebarba na Peça	000	00
Manutenção Mecânica	000	00	Matriz Quebrada	000	00
Falta de Energia	000	00	Punção Quebrado	000	00
Falta de Ar	000	00	Pino Guia Quebrado	000	00
Setup	000	00	Mola Quebrada	000	00
Falta de Transporte	000	00	Outros	000	00
Enfermaria	000	00			
Inspeção	000	00			
Falta de Operador	000	00	Total de Paradas	0000	
Ajuste	000	00	Total de Horas Disponíveis	0000	
Ferramentaria	000	00	Utilização das Máquinas (%)	00,0%	
			Eficiência (%)	00,0%	

Figura 21 – Tela de Entrada do Sistema

Esta é a tela de entrada do programa, onde o usuário indica o período que ele quer fazer a análise dos Controles de Produção apontados. Uma vez pressionado o botão calcular, o usuário obtem as informações de paradas e suas respectivas porcentagens de parada, assim como a utilização e a eficiência da linha.

Para que o usuário insira uma nova folha de Controle de Produção, basta pressionar a botão Controle de Produção para que apareça a tela seguinte.

CONTROLE DE PRODUÇÃO					
Número da Máquina:	15	Data:		Códigos de Parada	
Ferramenta	Período	Peças/hora (OP)	Produção	Parada/Código (1)	Parada/Código (2)
	05:50 às 07:00				
	07:00 às 08:00				
	08:00 às 09:00				
	09:00 às 10:00				
	10:00 às 11:00				
	11:00 às 12:00				
	12:00 às 13:00				
	13:00 às 14:10				
	14:10 às 15:00				
	15:00 às 16:00				
	16:00 às 17:00				
	17:00 às 18:00				
	18:00 às 19:00				
	19:00 às 20:00				
	20:00 às 21:00				
	21:00 às 22:26				
	22:26 às 23:00				
	23:00 às 00:00				
	00:00 à 01:00				
	01:00 às 02:00				
	02:00 às 03:00				
	03:00 às 04:00				
	04:00 às 05:00				
	05:00 às 05:46				

Figura 22 – Controle de Produção do Sistema

Nesta tela o usuário pode verificar os códigos de parada, adicionar ou remover um registro. Feito isso, pode-se voltar para a outra janela e obter os resultados desejados.

9. Conclusões

A Eficiência nos serve de forma significativa para detectarmos a qualidade de um dado processo de fabricação, entretanto, a forma como ela é passada nem sempre nos possibilita tomar decisões acerca da identificação de um problema.

Assim, com base em um apontamento de Controle de Produção, podemos detectar algumas das falhas do sistema e com base nelas propor melhorias.

Foram abordadas as principais razões de parada de uma estamparia, e para algumas delas propomos melhorias com base nos resultados encontrados. Para o caso de paradas por ferramentaria, subdividimos esta de forma a tornar a sua análise mais específica.

Para as paradas por setup, verificamos a importância da adoção dos Sistemas de Troca Rápida, que aumentam significativamente a utilização das máquinas com a queda do tempo de setup, além de melhor definir o tempo de troca de ferramentas melhorando a programação da produção.

Também verificamos a importância da adoção de um sistema de Manutenção Preventiva Total (TPM), de forma a minimizarmos os tempos decorrentes de manutenção mecânica ou elétrica das máquinas.

Anexo 1 – Controle de Produção

CONTROLE DE PRODUÇÃO

N.º da Maquina _____ TON. _____

Data ____ / ____ / ____

Produto	Ferram.	Pçs/ Bat.	Hora	RE	Pçs/hora (OP)	Produção	Parada	Cód. Parada	Acumulado
			05:50 - 07:00						
			07:00 - 08:00						
			08:00 - 09:00						
			09:00 - 10:00						
			10:00 - 11:00						
			11:00 - 12:00						
			12:00 - 13:00						
			13:00 - 14:10						

Produto	Ferram.	Pçs/ Bat.	Hora	RE	Pçs/hora (OP)	Produção	Parada	Cód. Parada	Acumulado
			14:10 - 15:00						
			15:00 - 16:00						
			16:00 - 17:00						
			17:00 - 18:00						
			18:00 - 19:00						
			19:00 - 20:00						
			20:00 - 21:00						
			21:00 - 22:26						

Produto	Ferram.	Pçs/ Bat.	Hora	RE	Pçs/hora (OP)	Produção	Parada	Cód. Parada	Acumulado
			22:26 - 23:00						
			23:00 - 00:00						
			00:00 - 01:00						
			01:00 - 02:00						
			02:00 - 03:00						
			03:00 - 04:00						
			04:00 - 05:00						
			05:00 - 05:46						

- Códigos de Parada:**
- | | | |
|-----------------------|----------------------|---------------------|
| 0 – Falta de Material | 5 – Falta ar | 10 - Falta Operador |
| 1 – Ferramentaria | 6 – Setup | 12 - Ajuste |
| 2 – Manut. Elétrica | 7 - Falta Transporte | 14 - Refeição |
| 3 – Manut. Mecânica | 8 – Enfermaria | 18 – Reunião |
| 4 – Falta Energia | 9 – Inspeção | 19 – Try Out |

Ocorrências (verso)

Anexo 2 – Código do Programa / Forms

1. Form 1

Option Explicit

```
' barra de rolagem
Private Sub ArrangeScrollBars()
Dim have_wid As Single
Dim have_hgt As Single
Dim need_wid As Single
Dim need_hgt As Single
Dim need_hbar As Boolean
Dim need_vbar As Boolean

IfWindowState = vbMinimized Then Exit Sub

need_wid = picInner.Width + (picOuter.Width - picOuter.ScaleWidth)
need_hgt = picInner.Height + (picOuter.Height - picOuter.ScaleHeight)
have_wid = ScaleWidth
have_hgt = ScaleHeight

need_hbar = (need_wid > have_wid)
If need_hbar Then have_hgt = have_hgt - HBar.Height

need_vbar = (need_hgt > have_hgt)
If need_vbar Then

    have_wid = have_wid - VBar.Width
    If Not need_hbar Then
        need_hbar = (need_wid > have_wid)
        If need_hbar Then have_hgt = have_hgt - HBar.Height
    End If
End If

picOuter.Move 0, 0, have_wid, have_hgt

If need_hbar Then
    HBar.Move 0, have_hgt, have_wid
    HBar.Min = 0
    HBar.Max = picOuter.ScaleWidth - picInner.Width
    HBar.LargeChange = picOuter.ScaleWidth
    HBar.SmallChange = picOuter.ScaleWidth / 5
    HBar.Visible = True
Else
    HBar.Visible = False
```

```
End If

If need_vbar Then
    VBar.Move have_wid, 0, VBar.Width, have_hgt
    VBar.Min = 0
    VBar.Max = picOuter.ScaleHeight - picInner.Height
    VBar.LargeChange = picOuter.ScaleHeight
    VBar.SmallChange = picOuter.ScaleHeight / 5
    VBar.Visible = True
Else
    VBar.Visible = False
End If
End Sub
```

```
Private Sub Command1_Click()
Dim pesq As String, busc As String
pesq = InputBox("Informe o número do máquina ", "nome")
Data1.Recordset.FindFirst busc
If Data1.Recordset.NoMatch Then
    MsgBox ("Nenhum dado registrado para a máquina " & pesq & "!")
End If
End Sub
```

```
Private Sub Command2_Click()
Dim numquina As Single
numquina = InputBox("Informe o número da máquina", "maquina")
Data1.Recordset.AddNew
Text1.Text = numquina

End Sub
```

```
Private Sub Command3_Click()
Data1.Recordset.Delete
End Sub
```

```
Private Sub Command4_Click()
Form3.Show

End Sub
```

```
Private Sub Form_Resize()
    ArrangeScrollBars

End Sub
Private Sub HBar_Change()
    picInner.Left = HBar.Value

```

```
End Sub
```

```
Private Sub HBar_Scroll()  
    picInner.Left = HBar.Value  
End Sub
```

```
Private Sub VBar_Change()  
    picInner.Top = VBar.Value  
End Sub
```

```
Private Sub VBar_Scroll()  
    picInner.Top = VBar.Value  
End Sub
```

2. Form 2

```
Public cp_dados As Database
Public cp_tabela As Recordset
```

```
Private Sub Command1_Click()
```

```
Dim db As Database
```

```
Dim rs As Recordset
```

```
Dim cont As String
```

```
cont = 0
```

```
Do While cont < 38
```

```
l1d0 = "Select SUM(l1d) AS cond11d FROM dados where l1e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l2d0 = "Select SUM(l2d) AS cond12d FROM dados where l2e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l3d0 = "Select SUM(l3d) AS cond13d FROM dados where l3e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l4d0 = "Select SUM(l4d) AS cond14d FROM dados where l4e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l5d0 = "Select SUM(l5d) AS cond15d FROM dados where l5e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l6d0 = "Select SUM(l6d) AS cond16d FROM dados where l6e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l7d0 = "Select SUM(l7d) AS cond17d FROM dados where l7e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l8d0 = "Select SUM(l8d) AS cond18d FROM dados where l8e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l9d0 = "Select SUM(l9d) AS cond19d FROM dados where l9e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l10d0 = "Select SUM(l10d) AS cond110d FROM dados where l10e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l11d0 = "Select SUM(l11d) AS cond111d FROM dados where l11e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l12d0 = "Select SUM(l12d) AS cond112d FROM dados where l12e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l13d0 = "Select SUM(l13d) AS cond113d FROM dados where l13e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l14d0 = "Select SUM(l14d) AS cond114d FROM dados where l14e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l15d0 = "Select SUM(l15d) AS cond115d FROM dados where l15e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l16d0 = "Select SUM(l16d) AS cond116d FROM dados where l16e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

```
l17d0 = "Select SUM(l17d) AS cond117d FROM dados where l17e=" + cont + " and data  
between #" & Text1.Text & "# and #" & Text2.Text & "#"
```

l18d0 = "Select SUM(l18d) AS cond18d FROM dados where l18e=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l19d0 = "Select SUM(l19d) AS cond19d FROM dados where l19e=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l20d0 = "Select SUM(l20d) AS cond20d FROM dados where l20e=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l21d0 = "Select SUM(l21d) AS cond21d FROM dados where l21e=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l22d0 = "Select SUM(l22d) AS cond22d FROM dados where l22e=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l23d0 = "Select SUM(l23d) AS cond23d FROM dados where l23e=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l24d0 = "Select SUM(l24d) AS cond24d FROM dados where l24e=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"

l1f0 = "Select SUM(l1f) AS cond1f FROM dados where l1g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l2f0 = "Select SUM(l2f) AS cond2f FROM dados where l2g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l3f0 = "Select SUM(l3f) AS cond3f FROM dados where l3g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l4f0 = "Select SUM(l4f) AS cond4f FROM dados where l4g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l5f0 = "Select SUM(l5f) AS cond5f FROM dados where l5g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l6f0 = "Select SUM(l6f) AS cond6f FROM dados where l6g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l7f0 = "Select SUM(l7f) AS cond7f FROM dados where l7g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l8f0 = "Select SUM(l8f) AS cond8f FROM dados where l8g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l9f0 = "Select SUM(l9f) AS cond9f FROM dados where l9g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l10f0 = "Select SUM(l10f) AS cond10f FROM dados where l10g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l11f0 = "Select SUM(l11f) AS cond11f FROM dados where l11g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l12f0 = "Select SUM(l12f) AS cond12f FROM dados where l12g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l13f0 = "Select SUM(l13f) AS cond13f FROM dados where l13g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l14f0 = "Select SUM(l14f) AS cond14f FROM dados where l14g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l15f0 = "Select SUM(l15f) AS cond15f FROM dados where l15g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"
l16f0 = "Select SUM(l16f) AS cond16f FROM dados where l16g=" + cont + " and data between #" & Text1.Text & "# and #" & Text2.Text & "#"

```

l17f0 = "Select SUM(l17f) AS cond17f FROM dados where l17g=" + cont + " and data
between #" & Text1.Text & "# and #" & Text2.Text & "#"
l18f0 = "Select SUM(l18f) AS cond18f FROM dados where l18g=" + cont + " and data
between #" & Text1.Text & "# and #" & Text2.Text & "#"
l19f0 = "Select SUM(l19f) AS cond19f FROM dados where l19g=" + cont + " and data
between #" & Text1.Text & "# and #" & Text2.Text & "#"
l20f0 = "Select SUM(l20f) AS cond20f FROM dados where l20g=" + cont + " and data
between #" & Text1.Text & "# and #" & Text2.Text & "#"
l21f0 = "Select SUM(l21f) AS cond21f FROM dados where l21g=" + cont + " and data
between #" & Text1.Text & "# and #" & Text2.Text & "#"
l22f0 = "Select SUM(l22f) AS cond22f FROM dados where l22g=" + cont + " and data
between #" & Text1.Text & "# and #" & Text2.Text & "#"
l23f0 = "Select SUM(l23f) AS cond23f FROM dados where l23g=" + cont + " and data
between #" & Text1.Text & "# and #" & Text2.Text & "#"
l24f0 = "Select SUM(l24f) AS cond24f FROM dados where l24g=" + cont + " and data
between #" & Text1.Text & "# and #" & Text2.Text & "#"

```

```

Set db = OpenDatabase("C:\Documents and Settings\Rafael\Desktop\POLI-
eficiencia\cp.mdb")

```

```

Set rs = db.OpenRecordset(l1d0, dbOpenSnapshot)
If rs.Fields("cond1d") > 0 Then a1 = rs.Fields("cond1d") Else a1 = 0
Set rs = db.OpenRecordset(l2d0, dbOpenSnapshot)
If rs.Fields("cond2d") > 0 Then a2 = rs.Fields("cond2d") Else a2 = 0
Set rs = db.OpenRecordset(l3d0, dbOpenSnapshot)
If rs.Fields("cond3d") > 0 Then a3 = rs.Fields("cond3d") Else a3 = 0
Set rs = db.OpenRecordset(l4d0, dbOpenSnapshot)
If rs.Fields("cond4d") > 0 Then a4 = rs.Fields("cond4d") Else a4 = 0
Set rs = db.OpenRecordset(l5d0, dbOpenSnapshot)
If rs.Fields("cond5d") > 0 Then a5 = rs.Fields("cond5d") Else a5 = 0
Set rs = db.OpenRecordset(l6d0, dbOpenSnapshot)
If rs.Fields("cond6d") > 0 Then a6 = rs.Fields("cond6d") Else a6 = 0
Set rs = db.OpenRecordset(l7d0, dbOpenSnapshot)
If rs.Fields("cond7d") > 0 Then a7 = rs.Fields("cond7d") Else a7 = 0
Set rs = db.OpenRecordset(l8d0, dbOpenSnapshot)
If rs.Fields("cond8d") > 0 Then a8 = rs.Fields("cond8d") Else a8 = 0
Set rs = db.OpenRecordset(l9d0, dbOpenSnapshot)
If rs.Fields("cond9d") > 0 Then a9 = rs.Fields("cond9d") Else a9 = 0
Set rs = db.OpenRecordset(l10d0, dbOpenSnapshot)
If rs.Fields("cond10d") > 0 Then a10 = rs.Fields("cond10d") Else a10 = 0
Set rs = db.OpenRecordset(l11d0, dbOpenSnapshot)
If rs.Fields("cond11d") > 0 Then a11 = rs.Fields("cond11d") Else a11 = 0
Set rs = db.OpenRecordset(l12d0, dbOpenSnapshot)
If rs.Fields("cond12d") > 0 Then a12 = rs.Fields("cond12d") Else a12 = 0
Set rs = db.OpenRecordset(l13d0, dbOpenSnapshot)
If rs.Fields("cond13d") > 0 Then a13 = rs.Fields("cond13d") Else a13 = 0
Set rs = db.OpenRecordset(l14d0, dbOpenSnapshot)

```

```
If rs.Fields("condl14d") > 0 Then a14 = rs.Fields("condl14d") Else a14 = 0
Set rs = db.OpenRecordset(l15d0, dbOpenSnapshot)
If rs.Fields("condl15d") > 0 Then a15 = rs.Fields("condl15d") Else a15 = 0
Set rs = db.OpenRecordset(l16d0, dbOpenSnapshot)
If rs.Fields("condl16d") > 0 Then a16 = rs.Fields("condl16d") Else a16 = 0
Set rs = db.OpenRecordset(l17d0, dbOpenSnapshot)
If rs.Fields("condl17d") > 0 Then a17 = rs.Fields("condl17d") Else a17 = 0
Set rs = db.OpenRecordset(l18d0, dbOpenSnapshot)
If rs.Fields("condl18d") > 0 Then a18 = rs.Fields("condl18d") Else a18 = 0
Set rs = db.OpenRecordset(l19d0, dbOpenSnapshot)
If rs.Fields("condl19d") > 0 Then a19 = rs.Fields("condl19d") Else a19 = 0
Set rs = db.OpenRecordset(l20d0, dbOpenSnapshot)
If rs.Fields("condl20d") > 0 Then a20 = rs.Fields("condl20d") Else a20 = 0
Set rs = db.OpenRecordset(l21d0, dbOpenSnapshot)
If rs.Fields("condl21d") > 0 Then a21 = rs.Fields("condl21d") Else a21 = 0
Set rs = db.OpenRecordset(l22d0, dbOpenSnapshot)
If rs.Fields("condl22d") > 0 Then a22 = rs.Fields("condl22d") Else a22 = 0
Set rs = db.OpenRecordset(l23d0, dbOpenSnapshot)
If rs.Fields("condl23d") > 0 Then a23 = rs.Fields("condl23d") Else a23 = 0
Set rs = db.OpenRecordset(l24d0, dbOpenSnapshot)
If rs.Fields("condl24d") > 0 Then a24 = rs.Fields("condl24d") Else a24 = 0
Set rs = db.OpenRecordset(l1f0, dbOpenSnapshot)
If rs.Fields("condl1f") > 0 Then a25 = rs.Fields("condl1f") Else a25 = 0
Set rs = db.OpenRecordset(l2f0, dbOpenSnapshot)
If rs.Fields("condl2f") > 0 Then a26 = rs.Fields("condl2f") Else a26 = 0
Set rs = db.OpenRecordset(l3f0, dbOpenSnapshot)
If rs.Fields("condl3f") > 0 Then a27 = rs.Fields("condl3f") Else a27 = 0
Set rs = db.OpenRecordset(l4f0, dbOpenSnapshot)
If rs.Fields("condl4f") > 0 Then a28 = rs.Fields("condl4f") Else a28 = 0
Set rs = db.OpenRecordset(l5f0, dbOpenSnapshot)
If rs.Fields("condl5f") > 0 Then a29 = rs.Fields("condl5f") Else a29 = 0
Set rs = db.OpenRecordset(l6f0, dbOpenSnapshot)
If rs.Fields("condl6f") > 0 Then a30 = rs.Fields("condl6f") Else a30 = 0
Set rs = db.OpenRecordset(l7f0, dbOpenSnapshot)
If rs.Fields("condl7f") > 0 Then a31 = rs.Fields("condl7f") Else a31 = 0
Set rs = db.OpenRecordset(l8f0, dbOpenSnapshot)
If rs.Fields("condl8f") > 0 Then a32 = rs.Fields("condl8f") Else a32 = 0
Set rs = db.OpenRecordset(l9f0, dbOpenSnapshot)
If rs.Fields("condl9f") > 0 Then a33 = rs.Fields("condl9f") Else a33 = 0
Set rs = db.OpenRecordset(l10f0, dbOpenSnapshot)
If rs.Fields("condl10f") > 0 Then a34 = rs.Fields("condl10f") Else a34 = 0
Set rs = db.OpenRecordset(l11f0, dbOpenSnapshot)
If rs.Fields("condl11f") > 0 Then a35 = rs.Fields("condl11f") Else a35 = 0
Set rs = db.OpenRecordset(l12f0, dbOpenSnapshot)
If rs.Fields("condl12f") > 0 Then a36 = rs.Fields("condl12f") Else a36 = 0
Set rs = db.OpenRecordset(l13f0, dbOpenSnapshot)
If rs.Fields("condl13f") > 0 Then a37 = rs.Fields("condl13f") Else a37 = 0
```

```

Set rs = db.OpenRecordset(l14f0, dbOpenSnapshot)
If rs.Fields("condl14f") > 0 Then a38 = rs.Fields("condl14f") Else a38 = 0
Set rs = db.OpenRecordset(l15f0, dbOpenSnapshot)
If rs.Fields("condl15f") > 0 Then a39 = rs.Fields("condl15f") Else a39 = 0
Set rs = db.OpenRecordset(l16f0, dbOpenSnapshot)
If rs.Fields("condl16f") > 0 Then a40 = rs.Fields("condl16f") Else a40 = 0
Set rs = db.OpenRecordset(l17f0, dbOpenSnapshot)
If rs.Fields("condl17f") > 0 Then a41 = rs.Fields("condl17f") Else a41 = 0
Set rs = db.OpenRecordset(l18f0, dbOpenSnapshot)
If rs.Fields("condl18f") > 0 Then a42 = rs.Fields("condl18f") Else a42 = 0
Set rs = db.OpenRecordset(l19f0, dbOpenSnapshot)
If rs.Fields("condl19f") > 0 Then a43 = rs.Fields("condl19f") Else a43 = 0
Set rs = db.OpenRecordset(l20f0, dbOpenSnapshot)
If rs.Fields("condl20f") > 0 Then a44 = rs.Fields("condl20f") Else a44 = 0
Set rs = db.OpenRecordset(l21f0, dbOpenSnapshot)
If rs.Fields("condl21f") > 0 Then a45 = rs.Fields("condl21f") Else a45 = 0
Set rs = db.OpenRecordset(l22f0, dbOpenSnapshot)
If rs.Fields("condl22f") > 0 Then a46 = rs.Fields("condl22f") Else a46 = 0
Set rs = db.OpenRecordset(l23f0, dbOpenSnapshot)
If rs.Fields("condl23f") > 0 Then a47 = rs.Fields("condl23f") Else a47 = 0
Set rs = db.OpenRecordset(l24f0, dbOpenSnapshot)
If rs.Fields("condl24f") > 0 Then a48 = rs.Fields("condl24f") Else a48 = 0

If cont = 0 Then material = a1 + a2 + a3 + a4 + a5 + a6 + a7 + a8 + a9 + a10 + a11 + a12 +
a13 + a14 + a15 + a16 + a17 + a18 + a19 + a20 + a21 + a22 + a23 + a24 + a25 + a26 + a27 +
a28 + a29 + a30 + a31 + a32 + a33 + a34 + a35 + a36 + a37 + a38 + a39 + a40 + a41 +
a42 + a43 + a44 + a45 + a46 + a47 + a48
If cont = 2 Then eletrica = a1 + a2 + a3 + a4 + a5 + a6 + a7 + a8 + a9 + a10 + a11 + a12 +
a13 + a14 + a15 + a16 + a17 + a18 + a19 + a20 + a21 + a22 + a23 + a24 + a25 + a26 + a27 +
a28 + a29 + a30 + a31 + a32 + a33 + a34 + a35 + a36 + a37 + a38 + a39 + a40 + a41 +
a42 + a43 + a44 + a45 + a46 + a47 + a48
If cont = 3 Then mecanica = a1 + a2 + a3 + a4 + a5 + a6 + a7 + a8 + a9 + a10 + a11 + a12 +
a13 + a14 + a15 + a16 + a17 + a18 + a19 + a20 + a21 + a22 + a23 + a24 + a25 + a26 + a27 +
a28 + a29 + a30 + a31 + a32 + a33 + a34 + a35 + a36 + a37 + a38 + a39 + a40 + a41 +
a42 + a43 + a44 + a45 + a46 + a47 + a48
If cont = 4 Then energia = a1 + a2 + a3 + a4 + a5 + a6 + a7 + a8 + a9 + a10 + a11 + a12 +
a13 + a14 + a15 + a16 + a17 + a18 + a19 + a20 + a21 + a22 + a23 + a24 + a25 + a26 + a27 +
a28 + a29 + a30 + a31 + a32 + a33 + a34 + a35 + a36 + a37 + a38 + a39 + a40 + a41 +
a42 + a43 + a44 + a45 + a46 + a47 + a48
If cont = 5 Then ar = a1 + a2 + a3 + a4 + a5 + a6 + a7 + a8 + a9 + a10 + a11 + a12 + a13 +
a14 + a15 + a16 + a17 + a18 + a19 + a20 + a21 + a22 + a23 + a24 + a25 + a26 + a27 + a28 +
a29 + a30 + a31 + a32 + a33 + a34 + a35 + a36 + a37 + a38 + a39 + a40 + a41 + a42 +
a43 + a44 + a45 + a46 + a47 + a48
If cont = 6 Then setup = a1 + a2 + a3 + a4 + a5 + a6 + a7 + a8 + a9 + a10 + a11 + a12 +
a13 + a14 + a15 + a16 + a17 + a18 + a19 + a20 + a21 + a22 + a23 + a24 + a25 + a26 + a27

```



```
If cont = 37 Then outros = a1 + a2 + a3 + a4 + a5 + a6 + a7 + a8 + a9 + a10 + a11 + a12 +
a13 + a14 + a15 + a16 + a17 + a18 + a19 + a20 + a21 + a22 + a23 + a24 + a25 + a26 + a27
+ a28 + a29 + a30 + a31 + a32 + a33 + a34 + a35 + a36 + a37 + a38 + a39 + a40 + a41 +
a42 + a43 + a44 + a45 + a46 + a47 + a48
```

```
cont = cont + 1
```

```
If cont = 13 Then cont = cont + 18
```

```
If cont = 1 Then cont = cont + 1
```

```
If cont = 13 Then cont = cont + 1
```

```
Loop
```

```
l1d0 = "Select SUM(l1b) AS cond11d FROM dados where data between #" & Text1.Text &
"#" and #" & Text2.Text & "#"
```

```
l2d0 = "Select SUM(l2b) AS cond12d FROM dados where data between #" & Text1.Text &
"#" and #" & Text2.Text & "#"
```

```
l3d0 = "Select SUM(l3b) AS cond13d FROM dados where data between #" & Text1.Text &
"#" and #" & Text2.Text & "#"
```

```
l4d0 = "Select SUM(l4b) AS cond14d FROM dados where data between #" & Text1.Text &
"#" and #" & Text2.Text & "#"
```

```
l5d0 = "Select SUM(l5b) AS cond15d FROM dados where data between #" & Text1.Text &
"#" and #" & Text2.Text & "#"
```

```
l6d0 = "Select SUM(l6b) AS cond16d FROM dados where data between #" & Text1.Text &
"#" and #" & Text2.Text & "#"
```

```
l7d0 = "Select SUM(l7b) AS cond17d FROM dados where data between #" & Text1.Text &
"#" and #" & Text2.Text & "#"
```

```
l8d0 = "Select SUM(l8b) AS cond18d FROM dados where data between #" & Text1.Text &
"#" and #" & Text2.Text & "#"
```

```
l9d0 = "Select SUM(l9b) AS cond19d FROM dados where data between #" & Text1.Text &
"#" and #" & Text2.Text & "#"
```

```
l10d0 = "Select SUM(l10b) AS cond10d FROM dados where data between #" &
Text1.Text & " #" and #" & Text2.Text & "#"
```

```
l11d0 = "Select SUM(l11b) AS cond11d FROM dados where data between #" &
Text1.Text & " #" and #" & Text2.Text & "#"
```

```
l12d0 = "Select SUM(l12b) AS cond12d FROM dados where data between #" &
Text1.Text & " #" and #" & Text2.Text & "#"
```

```
l13d0 = "Select SUM(l13b) AS cond13d FROM dados where data between #" &
Text1.Text & " #" and #" & Text2.Text & "#"
```

```
l14d0 = "Select SUM(l14b) AS cond14d FROM dados where data between #" &
Text1.Text & " #" and #" & Text2.Text & "#"
```

```
l15d0 = "Select SUM(l15b) AS cond15d FROM dados where data between #" &
Text1.Text & " #" and #" & Text2.Text & "#"
```

```
l16d0 = "Select SUM(l16b) AS cond16d FROM dados where data between #" &
Text1.Text & " #" and #" & Text2.Text & "#"
```

```
l17d0 = "Select SUM(l17b) AS cond17d FROM dados where data between #" &
Text1.Text & " #" and #" & Text2.Text & "#"
```

I18d0 = "Select SUM(I18b) AS condI18d FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I19d0 = "Select SUM(I19b) AS condI19d FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I20d0 = "Select SUM(I20b) AS condI20d FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I21d0 = "Select SUM(I21b) AS condI21d FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I22d0 = "Select SUM(I22b) AS condI22d FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I23d0 = "Select SUM(I23b) AS condI23d FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I24d0 = "Select SUM(I24b) AS condI24d FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"

I1f0 = "Select SUM(I1c) AS condI1f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I2f0 = "Select SUM(I2c) AS condI2f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I3f0 = "Select SUM(I3c) AS condI3f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I4f0 = "Select SUM(I4c) AS condI4f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I5f0 = "Select SUM(I5c) AS condI5f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I6f0 = "Select SUM(I6c) AS condI6f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I7f0 = "Select SUM(I7c) AS condI7f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I8f0 = "Select SUM(I8c) AS condI8f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I9f0 = "Select SUM(I9c) AS condI9f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I10f0 = "Select SUM(I10c) AS condI10f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I11f0 = "Select SUM(I11c) AS condI11f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I12f0 = "Select SUM(I12c) AS condI12f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I13f0 = "Select SUM(I13c) AS condI13f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I14f0 = "Select SUM(I14c) AS condI14f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I15f0 = "Select SUM(I15c) AS condI15f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"
I16f0 = "Select SUM(I16c) AS condI16f FROM dados where data between #" & Text1.Text & "# and #" & Text2.Text & "#"

```

l17f0 = "Select SUM(l17c) AS cond17f FROM dados where data between #" &
Text1.Text & "# and #" & Text2.Text & "#"
l18f0 = "Select SUM(l18c) AS cond18f FROM dados where data between #" &
Text1.Text & "# and #" & Text2.Text & "#"
l19f0 = "Select SUM(l19c) AS cond19f FROM dados where data between #" &
Text1.Text & "# and #" & Text2.Text & "#"
l20f0 = "Select SUM(l20c) AS cond20f FROM dados where data between #" &
Text1.Text & "# and #" & Text2.Text & "#"
l21f0 = "Select SUM(l21c) AS cond21f FROM dados where data between #" &
Text1.Text & "# and #" & Text2.Text & "#"
l22f0 = "Select SUM(l22c) AS cond22f FROM dados where data between #" &
Text1.Text & "# and #" & Text2.Text & "#"
l23f0 = "Select SUM(l23c) AS cond23f FROM dados where data between #" &
Text1.Text & "# and #" & Text2.Text & "#"
l24f0 = "Select SUM(l24c) AS cond24f FROM dados where data between #" &
Text1.Text & "# and #" & Text2.Text & "#"

total3 = "Select count(*) AS Numerodias FROM dados where data between #" &
Text1.Text & "# and #" & Text2.Text & "#"

Set rs = db.OpenRecordset(total3, dbOpenSnapshot)
If rs.Fields("Numerodias") > 0 Then num = rs.Fields("Numerodias") Else num = 0

Set rs = db.OpenRecordset(l1d0, dbOpenSnapshot)
If rs.Fields("cond11d") > 0 Then b1 = rs.Fields("cond11d") Else a1 = 0
Set rs = db.OpenRecordset(l2d0, dbOpenSnapshot)
If rs.Fields("cond12d") > 0 Then b2 = rs.Fields("cond12d") Else a2 = 0
Set rs = db.OpenRecordset(l3d0, dbOpenSnapshot)
If rs.Fields("cond13d") > 0 Then b3 = rs.Fields("cond13d") Else a3 = 0
Set rs = db.OpenRecordset(l4d0, dbOpenSnapshot)
If rs.Fields("cond14d") > 0 Then b4 = rs.Fields("cond14d") Else a4 = 0
Set rs = db.OpenRecordset(l5d0, dbOpenSnapshot)
If rs.Fields("cond15d") > 0 Then b5 = rs.Fields("cond15d") Else a5 = 0
Set rs = db.OpenRecordset(l6d0, dbOpenSnapshot)
If rs.Fields("cond16d") > 0 Then b6 = rs.Fields("cond16d") Else a6 = 0
Set rs = db.OpenRecordset(l7d0, dbOpenSnapshot)
If rs.Fields("cond17d") > 0 Then b7 = rs.Fields("cond17d") Else a7 = 0
Set rs = db.OpenRecordset(l8d0, dbOpenSnapshot)
If rs.Fields("cond18d") > 0 Then b8 = rs.Fields("cond18d") Else a8 = 0
Set rs = db.OpenRecordset(l9d0, dbOpenSnapshot)
If rs.Fields("cond19d") > 0 Then b9 = rs.Fields("cond19d") Else a9 = 0
Set rs = db.OpenRecordset(l10d0, dbOpenSnapshot)
If rs.Fields("cond110d") > 0 Then b10 = rs.Fields("cond110d") Else a10 = 0
Set rs = db.OpenRecordset(l11d0, dbOpenSnapshot)
If rs.Fields("cond111d") > 0 Then b11 = rs.Fields("cond111d") Else a11 = 0
Set rs = db.OpenRecordset(l12d0, dbOpenSnapshot)
If rs.Fields("cond112d") > 0 Then b12 = rs.Fields("cond112d") Else a12 = 0

```

```
Set rs = db.OpenRecordset(l13d0, dbOpenSnapshot)
If rs.Fields("condl13d") > 0 Then b13 = rs.Fields("condl13d") Else a13 = 0
Set rs = db.OpenRecordset(l14d0, dbOpenSnapshot)
If rs.Fields("condl14d") > 0 Then b14 = rs.Fields("condl14d") Else a14 = 0
Set rs = db.OpenRecordset(l15d0, dbOpenSnapshot)
If rs.Fields("condl15d") > 0 Then b15 = rs.Fields("condl15d") Else a15 = 0
Set rs = db.OpenRecordset(l16d0, dbOpenSnapshot)
If rs.Fields("condl16d") > 0 Then b16 = rs.Fields("condl16d") Else a16 = 0
Set rs = db.OpenRecordset(l17d0, dbOpenSnapshot)
If rs.Fields("condl17d") > 0 Then b17 = rs.Fields("condl17d") Else a17 = 0
Set rs = db.OpenRecordset(l18d0, dbOpenSnapshot)
If rs.Fields("condl18d") > 0 Then b18 = rs.Fields("condl18d") Else a18 = 0
Set rs = db.OpenRecordset(l19d0, dbOpenSnapshot)
If rs.Fields("condl19d") > 0 Then b19 = rs.Fields("condl19d") Else a19 = 0
Set rs = db.OpenRecordset(l20d0, dbOpenSnapshot)
If rs.Fields("condl20d") > 0 Then b20 = rs.Fields("condl20d") Else a20 = 0
Set rs = db.OpenRecordset(l21d0, dbOpenSnapshot)
If rs.Fields("condl21d") > 0 Then b21 = rs.Fields("condl21d") Else a21 = 0
Set rs = db.OpenRecordset(l22d0, dbOpenSnapshot)
If rs.Fields("condl22d") > 0 Then b22 = rs.Fields("condl22d") Else a22 = 0
Set rs = db.OpenRecordset(l23d0, dbOpenSnapshot)
If rs.Fields("condl23d") > 0 Then b23 = rs.Fields("condl23d") Else a23 = 0
Set rs = db.OpenRecordset(l24d0, dbOpenSnapshot)
If rs.Fields("condl24d") > 0 Then b24 = rs.Fields("condl24d") Else a24 = 0
Set rs = db.OpenRecordset(l1f0, dbOpenSnapshot)
If rs.Fields("condl1f") > 0 Then b25 = rs.Fields("condl1f") Else a25 = 0
Set rs = db.OpenRecordset(l2f0, dbOpenSnapshot)
If rs.Fields("condl2f") > 0 Then b26 = rs.Fields("condl2f") Else a26 = 0
Set rs = db.OpenRecordset(l3f0, dbOpenSnapshot)
If rs.Fields("condl3f") > 0 Then b27 = rs.Fields("condl3f") Else a27 = 0
Set rs = db.OpenRecordset(l4f0, dbOpenSnapshot)
If rs.Fields("condl4f") > 0 Then b28 = rs.Fields("condl4f") Else a28 = 0
Set rs = db.OpenRecordset(l5f0, dbOpenSnapshot)
If rs.Fields("condl5f") > 0 Then b29 = rs.Fields("condl5f") Else a29 = 0
Set rs = db.OpenRecordset(l6f0, dbOpenSnapshot)
If rs.Fields("condl6f") > 0 Then b30 = rs.Fields("condl6f") Else a30 = 0
Set rs = db.OpenRecordset(l7f0, dbOpenSnapshot)
If rs.Fields("condl7f") > 0 Then b31 = rs.Fields("condl7f") Else a31 = 0
Set rs = db.OpenRecordset(l8f0, dbOpenSnapshot)
If rs.Fields("condl8f") > 0 Then b32 = rs.Fields("condl8f") Else a32 = 0
Set rs = db.OpenRecordset(l9f0, dbOpenSnapshot)
If rs.Fields("condl9f") > 0 Then b33 = rs.Fields("condl9f") Else a33 = 0
Set rs = db.OpenRecordset(l10f0, dbOpenSnapshot)
If rs.Fields("condl10f") > 0 Then b34 = rs.Fields("condl10f") Else a34 = 0
Set rs = db.OpenRecordset(l11f0, dbOpenSnapshot)
If rs.Fields("condl11f") > 0 Then b35 = rs.Fields("condl11f") Else a35 = 0
Set rs = db.OpenRecordset(l12f0, dbOpenSnapshot)
```

```

If rs.Fields("condl12f") > 0 Then b36 = rs.Fields("condl12f") Else a36 = 0
Set rs = db.OpenRecordset(l13f0, dbOpenSnapshot)
If rs.Fields("condl13f") > 0 Then b37 = rs.Fields("condl13f") Else a37 = 0
Set rs = db.OpenRecordset(l14f0, dbOpenSnapshot)
If rs.Fields("condl14f") > 0 Then b38 = rs.Fields("condl14f") Else a38 = 0
Set rs = db.OpenRecordset(l15f0, dbOpenSnapshot)
If rs.Fields("condl15f") > 0 Then b39 = rs.Fields("condl15f") Else a39 = 0
Set rs = db.OpenRecordset(l16f0, dbOpenSnapshot)
If rs.Fields("condl16f") > 0 Then b40 = rs.Fields("condl16f") Else a40 = 0
Set rs = db.OpenRecordset(l17f0, dbOpenSnapshot)
If rs.Fields("condl17f") > 0 Then b41 = rs.Fields("condl17f") Else a41 = 0
Set rs = db.OpenRecordset(l18f0, dbOpenSnapshot)
If rs.Fields("condl18f") > 0 Then b42 = rs.Fields("condl18f") Else a42 = 0
Set rs = db.OpenRecordset(l19f0, dbOpenSnapshot)
If rs.Fields("condl19f") > 0 Then b43 = rs.Fields("condl19f") Else a43 = 0
Set rs = db.OpenRecordset(l20f0, dbOpenSnapshot)
If rs.Fields("condl20f") > 0 Then b44 = rs.Fields("condl20f") Else a44 = 0
Set rs = db.OpenRecordset(l21f0, dbOpenSnapshot)
If rs.Fields("condl21f") > 0 Then b45 = rs.Fields("condl21f") Else a45 = 0
Set rs = db.OpenRecordset(l22f0, dbOpenSnapshot)
If rs.Fields("condl22f") > 0 Then b46 = rs.Fields("condl22f") Else a46 = 0
Set rs = db.OpenRecordset(l23f0, dbOpenSnapshot)
If rs.Fields("condl23f") > 0 Then b47 = rs.Fields("condl23f") Else a47 = 0
Set rs = db.OpenRecordset(l24f0, dbOpenSnapshot)
If rs.Fields("condl24f") > 0 Then b48 = rs.Fields("condl24f") Else a48 = 0

```

era = b1 + b2 + b3 + b4 + b5 + b6 + b7 + b8 + b9 + b10 + b11 + b12 + b13 + b14 + b15 +
b16 + b17 + b18 + b19 + b20 + b21 + b22 + b23 + b24

foi = b25 + b26 + b27 + b28 + b29 + b30 + b31 + b32 + b33 + b34 + b35 + b36 + b37 +
b38 + b39 + b40 + b41 + b42 + b43 + b44 + b45 + b46 + b47 + b48

ferramentaria = peca + rebarba + matriz + puncao + pino + mola + outros

'LABELS

```

Label25.Caption = FormatNumber(material / 60, 2)
Label26.Caption = FormatNumber(eletrica / 60, 2)
Label27.Caption = FormatNumber(mecanica / 60, 2)
Label28.Caption = FormatNumber(energia / 60, 2)
Label29.Caption = FormatNumber(ar / 60, 2)
Label30.Caption = FormatNumber(setup / 60, 2)
Label31.Caption = FormatNumber(transporte / 60, 2)
Label32.Caption = FormatNumber(enfermaria / 60, 2)
Label33.Caption = FormatNumber(inspecao / 60, 2)
Label34.Caption = FormatNumber(operator / 60, 2)
Label35.Caption = FormatNumber(ajuste / 60, 2)
Label36.Caption = FormatNumber(ferramentaria / 60, 2)

```

```
Label37.Caption = FormatNumber(peca / 60, 2)
Label38.Caption = FormatNumber(rebarba / 60, 2)
Label39.Caption = FormatNumber(matriz / 60, 2)
Label40.Caption = FormatNumber(puncao / 60, 2)
Label41.Caption = FormatNumber(pino / 60, 2)
Label42.Caption = FormatNumber(mola / 60, 2)
Label43.Caption = FormatNumber(outros / 60, 2)
```

```
paradas = (material + mecanica + eletrica + energia + ar + setup + transporte + enfermaria
+ inspecao + operador + ajuste + ferramentaria)
```

```
Label48.Caption = FormatNumber(paradas / 60, 2)
```

```
disponiveis = num * 21
```

```
Label49.Caption = FormatNumber(disponiveis, 2)
```

```
utilizacao = ((disponiveis - (paradas / 60)) / disponiveis) * 100
```

```
eficiencia = (foi / era) * 100
```

```
Label44.Caption = FormatNumber(utilizacao, 2)
```

```
Label45.Caption = FormatNumber(eficiencia, 2)
```

```
'porcentagem
```

```
Label56.Caption = FormatNumber((material / paradas) * 100, 2)
Label57.Caption = FormatNumber((eletrica / paradas) * 100, 2)
Label58.Caption = FormatNumber((mecanica / paradas) * 100, 2)
Label59.Caption = FormatNumber((energia / paradas) * 100, 2)
Label60.Caption = FormatNumber((ar / paradas) * 100, 2)
Label61.Caption = FormatNumber((setup / paradas) * 100, 2)
Label62.Caption = FormatNumber((transporte / paradas) * 100, 2)
Label63.Caption = FormatNumber((enfermaria / paradas) * 100, 2)
Label64.Caption = FormatNumber((inspecao / paradas) * 100, 2)
Label65.Caption = FormatNumber((operador / paradas) * 100, 2)
Label66.Caption = FormatNumber((ajuste / paradas) * 100, 2)
Label67.Caption = FormatNumber((ferramentaria / paradas) * 100, 2)
```

```
Label68.Caption = FormatNumber((peca / ferramentaria) * 100, 2)
Label69.Caption = FormatNumber((rebarba / ferramentaria) * 100, 2)
Label70.Caption = FormatNumber((matriz / ferramentaria) * 100, 2)
Label71.Caption = FormatNumber((puncao / ferramentaria) * 100, 2)
Label72.Caption = FormatNumber((pino / ferramentaria) * 100, 2)
Label73.Caption = FormatNumber((mola / ferramentaria) * 100, 2)
Label74.Caption = FormatNumber((outros / ferramentaria) * 100, 2)
```

```
End Sub
```

```
Private Sub Command2_Click()
Form1.Show
End Sub

Private Sub Form_Load()
Set cp_dados = OpenDatabase("C:\Documents and Settings\Rafael\Desktop\POLI-
eficiência\cp.mdb", False)
Set cp_tabela = cp_dados.OpenRecordset("dados", dbOpenDynaset)

End Sub
```

10. Bibliografia

1. BIDÁ, A.G.; SILVA, I.B.; **Melhoria de Vida de Ferramenta de Forjamento a Quente: Metodologia e Análise de Causas**, 24^a Senafor, Porto Alegre, 2003.
2. BONFATTI, Danilo B., **Estimativa do Tempo de Produção em uma Indústria Intermittente sob Encomenda**, Trabalho de Formatura apresentado à EPUSP, 2003.
3. CORTES, Pedro L. ; **Conhecendo e Trabalhando com o Visual Basic**, Ed Érica, 7^a Ed, 2004
4. PORTER M.E.; **Vantagem Competitiva: criando e sustentando um desempenho superior**, ed. Campus, Rio de Janeiro, 1992.
5. WERKEMA M.C.C.; **As Ferramentas da Qualidade no Gerenciamento de Processos**, Editora Sografe, Belo Horizonte, 1995.
6. DOYLE, L.E.; **Processos de Fabricação e Materiais para Engenheiros**, 3.a Edicao, Editora Edgard Blucher, 1962
7. SCHALCH, André; Canovas C, Dílson; **Informatização e Gerenciamento da Fabricação Mecânica utilizando a Sistemografia. Estudo de caso: Processo de Retifica e Lapidação**, Trabalho de Formatura apresentado à EPUSP, 2000.
8. SLACK, N.; **Administração da Produção**, 1.a edição, Ed Atlas, São Paulo, 1996.
9. MOREIRA, D.A.; **Administração da Produção e operações**, 2^a. ed, Editora Pioneira, São Paulo, 1996.
10. VOLLMANN, T.E., **Manufacturing planning and control systems**, 4th Edition, McGraw, Boston, 1997.

11. DEGARMO, E.P., **Materials and Processes in Manufacturing**, 7th Edition,
McMilhan Publishing Company, New York, 1988.

12. ZACCARELLI, S. B. **Programação e Controle da Produção**. São Paulo:
Pioneira, 1973.