

Universidade Federal de Santa Catarina
Centro Tecnológico
Engenharia de Controle e Automação Industrial

**Especificação da Integração
do Sistema de
Gerenciamento de Manutenção da
Fábrica II - EMBRACO**

*Monografia submetida à Universidade Federal de Santa Catarina
como requisito para a aprovação na disciplina:
EEL 5901: Projeto de Fim de Curso*

Elmo Martins

Florianópolis, Dezembro de 1995

Especificação da Integração do Sistema de Gerenciamento de Manutenção da Fábrica II - EMBRACO

Elmo Martins

Esta monografia foi julgada no contexto da disciplina
EEL 5901: Projeto de Fim de Curso e aprovada na sua forma final
pelo curso de **Engenharia de Controle e Automação Industrial**

Banca Examinadora:

Eng. Manuel C. L. F. Braga
Orientador Empresa

Prof. Edison da Rosa
Orientador do Curso

Prof. Augusto Humberto Bruciapaglia
Responsável pela disciplina e coordenador do curso

Prof. João Carlos Espíndola, *Avaliador*

Aluno Henrique Simas, *Debatedor*

Aluno Rodolfo Pinto da Luz, *Debatedor*

Agradecimentos

Agradeço, acima de tudo, a DEUS, por estar sempre presente ao meu lado, nos momentos de alegria e de tristeza. Sem Ele nada disso existiria, e é a Ele que eu dedico este trabalho.

Agradeço aos meus pais, Edésio e Dorcas, pela educação recebida e pelo amor constante em minha vida. Aos meus irmãos, Edson e Elcio, e minha namorada, Chirley, pela confiança e pelo apoio e auxílio nas decisões e nas horas difíceis.

Agradeço à empresa **EMBRACO**, por ter me proporcionado totais condições para realizar este trabalho. Agradeço também aos Engenheiros Manuel Braga e Paulo Henrique, e a todos da **EMBRACO** que direta ou indiretamente me auxiliaram na realização deste trabalho.

Agradeço aos professores Augusto H. Bruciapaglia e Jean Marie, e a todos os demais professores do curso de Engenharia de Controle e Automação, pelo conhecimento e experiência a mim repassados.

A todos o meu sincero *muito obrigado*.

Sumário

AGRADECIMENTOS	ii
SUMÁRIO	iii
RESUMO	vi
ABSTRACT	vii
INTRODUÇÃO	01
CAPÍTULO I <i>A Empresa</i>	03
1.1 Histórico	03
1.2 O Projeto Fábrica II	05
1.3 A Manutenção	07
CAPÍTULO II <i>Conceitos</i>	10
2.1 Base de Dados Relacional	10
2.1.1 Comparação com Outros Modelos	12
2.2 SQL	12
2.2.1 Tipos de Comandos SQL	13
2.3 Redes e Base de Dados	14
2.3.1 O modelo Cliente Servidor	14
2.4 Instâncias de Base de Dados	15
2.5 Processo	16
2.6 Trigger	16
2.7 ODBC	17
CAPÍTULO III <i>O Software MAXIMO</i>	19
3.1 Introdução	19
3.2 Módulo Ordens de Serviço	22
3.3 Módulo Manutenção Preventiva	24
3.4 Módulo Planos de Trabalho	26

3.5 Módulo Equipamento	27
3.6 Módulo Compras	28
3.7 Módulo Inventário	28
3.8 Módulo Recursos.....	29
3.9 Módulo Mão-de-Obra	29
3.10 Módulo Calendários.....	30
3.11 Módulo Notas.....	30
3.12 Módulo Configuração	30
3.13 Módulo Utilidades	32
3.14 MAXIMO Scheduler.....	32
CAPÍTULO IV <i>O Sistema de Informação Industrial - SII</i>	33
CAPÍTULO V <i>O Sistema GDT</i>	38
CAPÍTULO VI <i>Levantamento das Necessidades e Avaliação das Possibilidades</i>	43
6.1 Levantamento das Necessidades	43
6.1.1 Integração MAXIMO & SII	44
6.1.2 Integração MAXIMO & GDT	46
6.2 Avaliação das Possibilidades	47
6.2.1 Integração MAXIMO & SII	47
6.2.2 Integração MAXIMO & GDT	52
CAPÍTULO VII <i>Especificação da Integração</i>	54
7.1 Integração entre MAXIMO e Coletor de Dados	54
7.2 Integração entre MAXIMO e FI2	58
7.2.1 Interface Padrão do FI2	58
7.2.2 Escolha do Tipo de Arquivo Padrão	61
7.2.3 Especificação de Tabelas e Campos de Dados a serem utilizados na Integração.....	64
7.2.4 Especificação dos Processos e Outros Mecanismos	67

CONCLUSÃO.....	70
PERSPECTIVAS.....	71
BIBLIOGRAFIA.....	72
ANEXO I.....	73
ANEXO II.....	74
ANEXO III.....	75

Resumo

Neste trabalho são estudados alguns softwares existentes na **EMBRACO** - Fábrica II, que atualmente funcionam de forma totalmente estanque, com o objetivo de especificar uma integração entre eles.

Inicialmente são descritos os três sistemas que farão parte da integração: O Sistema de Gerenciamento de Manutenção (MAXIMO), o Sistema de Informação Industrial (SII), e o sistema GDT (Gerenciador de Documentação Técnica). Esta primeira descrição aborda os sistemas sob o ponto de vista de usuário, apontando suas características e funcionalidades.

As necessidades que levam à idéia de integrar estes sistemas são enumeradas, e as possibilidades e restrições de implementação desta integração são descritas e avaliadas.

Por fim é especificada a integração entre eles, tratando-a detalhadamente, com possíveis configurações de hardware e software. Também são descritos funcionalmente e através de algoritmos os programas que serão desenvolvidos para implementar a integração.

Abstract

In this work we study some of **EMBRACO's** Plant II softwares, that nowadays perform tasks separated from each other, with the objective of specifying an integration between them.

These softwares are: a Maintenance Management System (**MAXIMO**), an Industrial Information System (**IIS**) and a **TDM** software (Total Documentation Management). Their characteristics and functionalities are first described from the user point of view.

The needs of this integration are highlighted, and the possibilities and restrictions of its implementation are described and evaluated.

At last the integration is specified in a detailed way, showing the possible hardware and software configurations. Also, the programs that will be developed to implement the integration are described, with their functions and algorithms.

Introdução

Com a automação cada vez mais presente dentro das fábricas, cresce a necessidade de agilizar o fluxo intenso de informações existentes. Informações estas que são geradas nos mais diversos setores, desde o chão-de-fábrica, com dados efetivos sobre a produção, passando pelos níveis de programação e ajustes da produção, e chegando a nível gerencial, o qual as recebe, analisa e gera mais informações.

A automação propicia o aumento do número das informações geradas, bem como a rapidez com que elas são disponibilizadas.

Com tudo isso, tornam-se imprescindíveis os sistemas de gerenciamento de banco de dados, bem como todo e qualquer sistema que automatize o processamento e o fluxo de informações, pois já não é mais viável coletar, analisar e armazenar dados manualmente, enquanto as informações são geradas de forma automática.

Os sistemas existentes atualmente, na maioria das vezes, manipulam informações bastante específicas e particulares, como é o caso dos sistemas de gerenciamento de manutenção, os sistemas de gerenciamento de documentos, etc. Geralmente cada setor da fábrica possui um sistema que atenda as suas necessidades, o que é de se esperar.

Em consequência da diversidade dos sistemas existentes nas fábricas, e da necessidade crescente de se agilizar a troca de informações entre setores, a idéia de **integrar** estes sistemas surge como uma solução *eficaz* e *definitiva*. *Eficaz* pelo fato de que efetivamente agiliza o fluxo de informações na fábrica, e *definitiva* pois está associada a uma necessidade indispensável trazida com a automatização das fábricas.

Integrar significa estabelecer um caminho para troca de informações. Integrar sistemas significa implementar programas que funcionem como interface entre eles, que habilitem a comunicação e a troca de dados entre sistemas isolados.

O nível de automação atingido pela **EMBRACO**, bem como a característica de seus sistemas, exige uma integração entre eles, a fim de proporcionar uma maior agilidade na troca de dados, e um controle muito mais eficaz de sua produção, além de outros fatores, como:

- Maior facilidade e rapidez no acesso às informações;
- Aumento da troca de informações entre setores;
- Possibilidade de disponibilizar, no chão-de-fábrica, funcionalidades hoje inexistentes, etc;

A idéia desta integração surgiu da existência de três sistemas que atualmente trabalham isolados: O **MAXIMO**, software de manutenção, o **SII**, Sistema de Informação Industrial (ver capítulos correspondentes aos dois sistemas), e o **GDT** (sistema de gerenciamento de documentação técnica).

O objetivo deste trabalho é avaliar a viabilidade desta integração e especificar como pode ser implementada.

No capítulo I é apresentado um breve histórico sobre a empresa **EMBRACO**.

No capítulo II são descritos os conceitos de alguns assuntos estudados para a realização deste trabalho.

Nos capítulos III, IV e V são descritos os três sistemas que compõem esta integração, tratando-os sob o ponto de vista de usuário.

Em seguida, no capítulo VI são expostos os fatores e propósitos que levam à necessidade de integração e as funcionalidades que podem ser trazidas com ela. As possibilidades de integração são então avaliadas, e feita uma comparação da arquitetura computacional que se tem hoje, e de como ficará o sistema após a implementação da integração.

Por último, no capítulo VII serão especificadas as integrações viáveis, apresentando os mecanismos de software que auxiliarão na integração, e também os programas que devem ser implementados, suas funcionalidades, e um esboço de seus códigos.

Capítulo I - *A Empresa*

1.1 - *Histórico*

A Empresa Brasileira de Compressores S.A. - **EMBRACO**, foi fundada em março de 1971, devido ao grande crescimento da indústria nacional de refrigeração, que até então era obrigada a importar compressores para conseguir manter as exigências de qualidade e quantidade dos produtos. Em novembro do mesmo ano a empresa fechou contrato com a DANFOSS, fabricante dinamarquesa de compressores herméticos, para montar aqui no Brasil, na cidade de Joinville, SC, compressores do tipo PW.

Em 1974, já com a fábrica instalada, teve início a montagem de compressores com a capacidade de produção de 1.000.000 de unidades/ano. Porém, a parte de pesquisa e desenvolvimento ainda era da DANFOSS e os principais componentes eram importados.

Em 1977 a empresa passou ao controle acionário do Grupo BRASMOTOR, que hoje possui empresas como a MULTIBRÁS (detentora das marcas Semer, Cônsul e Brastemp), SABRICO (turismo) entre outras. Em 1978 os componentes mecânicos dos compressores PW, que eram importados, começaram a ser fabricados pela **EMBRACO**, proporcionando investimento em pesquisas e criando laboratórios físico, químico, elétrico e metalúrgico, visando a melhoria dos materiais e processos de fabricação dos componentes de seus produtos. Naquele mesmo ano a empresa teve a qualidade de seus compressores aprovada nos E.U.A..

No ano de 1979 começaram as exportações para os E.U.A. e Canadá. No período entre 1981 e 1983 foi implantada em Joinville a fabricação de compressores do tipo FF, de alta eficiência, desenvolvidos pela DANFOSS. Engenheiros e técnicos foram enviados à Dinamarca para que o desenvolvimento de mão-de-obra especializada tornasse possível a produção de compressores de maior qualidade adaptados às exigências do mercado nacional e internacional. Foram iniciadas parcerias com universidades, centros de pesquisa e foi criado também o centro de pesquisa e

desenvolvimento da própria empresa. A empresa continuou crescendo e investiu em novos projetos como o dos compressores EM, de pequeno porte e baixo custo, com tecnologia totalmente desenvolvida na **EMBRACO**.

Em 1988 entrou em funcionamento sua fundição no distrito de Piraberaba, que fabrica o bloco e o eixo dos compressores, e em 1990 a **EMBRACO** implantou uma fábrica para produção de alguns componentes em Itaiópolis, SC.

No ano de 1993 o governo dos E.U.A., em conjunto com a associação das companhias de fornecimento de energia elétrica daquele país, ofereceu um prêmio para a empresa que apresentasse o projeto do refrigerador mais eficiente dos E.U.A. Foi o prêmio Golden Carrot, e uma das empresas finalistas foi a WHIRLPOOL, uma das maiores fabricantes de refrigeradores do mundo. Esta empresa, vencedora do prêmio, escolheu a **EMBRACO** como fornecedora do compressor, o FGT, que equipou o seu refrigerador. Isto demonstra o elevado nível tecnológico alcançado pela empresa e excelente qualidade dos seus produtos. No âmbito mundial, atualmente os compressores da linha FGT são os mais eficientes na sua faixa de aplicação.

Seus produtos contam com a aprovação de diversos órgãos internacionais de normalização e aprovação de uso, dentre os quais destacam-se a Underwriters Laboratories Inc.(U.L.), dos E.U.A., a Canadian Standard Association (C.S.A.), do Canadá, e o Instituto Italiano del Marchio di Qualità (I.M.Q.). Em 1992 a empresa obteve também o certificado de aprovação da norma ISO 9001, conferido à empresa após uma auditoria realizada pelo B.V.Q.I. (Bureau Veritas Quality International) da Inglaterra.

No ano de 1993 a **EMBRACO** comprou a empresa italiana ASPERA, fabricante de compressores, na cidade de Riva Preso de Chieri. Esta nova unidade, denominada **EMBRACO** EUROPA, possui em torno de 1.800 funcionários e produz cinco milhões de compressores por ano. Em 1994 a **EMBRACO** estabeleceu uma "joint-venture" com a empresa chinesa Beijing Snowflake Electric Appliance Group Corporation. Na nova empresa, denominada Beijing **EMBRACO** Snowflake Compressor Company Limited, a **EMBRACO** participa com 52% do capital social, a Snowflake tem 40% e a americana Whirlpool Corp. com 8%.

Esta unidade emprega 1000 funcionários e inicialmente sua produção será de um milhão de compressores por ano, com previsão para chegar a um milhão e quatrocentos mil.

1.2 - O Projeto Fábrica II

A **EMBRACO** tem como objetivo tornar-se o maior fabricante mundial de compressores herméticos para refrigeração até o ano 2000, sendo que atualmente concorre com a Matsushita, empresa japonesa, pela liderança do mercado.

Devido a isto, em 1992 surgiu a necessidade de expandir a produção de compressores da linha *EM*, e junto a isto, a oportunidade para a **EMBRACO** rever a sua estratégia de manufatura. O objetivo era tornar a **EMBRACO** um *World Class Manufacturer*¹, mas para isso, era necessário saber a relação custo/benefício envolvida nesta mudança. Também havia a idéia de se utilizar um sistema de fabricação voltado para produtos, em contraposição ao que vinha sendo adotado, focalizado em processos.

Para a elaboração do projeto, foi contratada uma empresa americana de consultoria, a Booz-Allen.

A revisão da estratégia de manufatura da **EMBRACO** baseou-se na avaliação de alguns cenários indicativos para a definição do enfoque fabril.

Depois de avaliadas as opções chegou-se à conclusão que a **EMBRACO** deveria mudar o seu layout fabril, aumentando o grau de enfoque em produto a curto e médio prazo, através de uma fábrica piloto. Esta decisão baseia-se na experiência de que uma fábrica focalizada por produto apresenta o melhor desempenho em custo e qualidade, lead-times, flexibilidade e sobretudo simplicidade.

A expansão da capacidade de produção dos compressores *EM* será feita de forma gradual, com o objetivo de se ter, em 1998, uma produção total de 6.54 milhões de unidades/ano, substituindo totalmente os compressores do tipo *PW*.

¹ Estratégia de manufatura que tem como base, entre outras coisas: enfoque no produto/cliente, ênfase na qualidade, reduzido estoque em processo, ênfase na manutenção produtiva total, etc.

Foram adotadas metas de desempenho visando baixar custos de mão de obra, baixar *lead-time*, diminuir o estoque em processo, obter menores custos e elevar a qualidade do produto final.

A fábrica piloto (*Fábrica II*) é constituída de nove células de componentes e duas de montagem. Está instalada no bloco 28, que até então servia para fabricação de alguns componentes de compressores. Estas operações foram transferidas para Itaiópolis, SC.

O layout adotado na *Fábrica II* é totalmente celular, possuindo um fluxo de materiais bastante linear, com distâncias minimizadas. Isto torna a fábrica bastante ágil em comparação com a *Fábrica I* (fábrica mais antiga), que possui um layout baseado em processos, e um fluxo demasiadamente irregular.

Este layout não permite utilizar "redes de segurança", ou estoque em processo. Em consequência, a fábrica se torna bastante sensível a problemas de produção, visto que, no limite, se uma célula parar, a fábrica inteira também irá parar. Neste contexto, é necessário programas robustos de:

- CEP (Controle Estatístico de Processo);
- Manutenção preventiva e preditiva;
- Qualidade de fornecedores;
- Treinamento das equipes operacionais e de apoio.

A fábrica foi dividida em unidades gerenciais para que cada uma tenha mais autonomia, facilitando deste modo as ações e descentralizando as decisões.

A estrutura organizacional da nova fábrica é bastante enxuta, em comparação com a *Fábrica I*, e favorece uma comunicação interna muito mais ágil. Baseia-se entre outras coisas, nos conceitos de equipes auto-gerenciadas, implicando em que cada um garanta a qualidade do produto que faz, reduzindo assim a quantidade de inspeções ao longo dos processos.

Com relação ao Planejamento e Controle da Produção (PCP), este está ligado diretamente ao gerente da fábrica e programa células ao invés de máquinas. Existe autonomia em suas

funções, possuindo interfaces com o restante da empresa, principalmente com suprimentos e vendas.

A *Fábrica II* encontra-se ainda em fase de implantação. Sua capacidade foi recentemente aumentada com a duplicação de todas as suas linhas de fabricação². Houve muitos problemas de ajuste do novo sistema e da implantação dos novos conceitos embutidos neste projeto. Devido a isto, o cronograma está um pouco atrasado. Porém é de se ressaltar a grande experiência adquirida no decorrer de toda a implantação da *Fábrica II*. Uma vez validado, este modelo de manufatura será adotado nas demais plantas da **EMBRACO**.

1.3 - A Manutenção:

Seguindo a estrutura organizacional proposta no projeto *Fábrica II*, a manutenção desta nova unidade se diferencia da *Fábrica I* por ser descentralizada. Na *Fábrica I*, todas as demais áreas requisitam serviços ao setor de manutenção, e este por sua vez "empresta" seus técnicos para a realização dos serviços. Já na *Fábrica II*, cada setor possui seus próprios técnicos de manutenção, não dependendo mais de requisições de serviços e tornando o trabalho muito mais ágil.

Os planejadores de cada área, que elaboram planos de manutenção e coordenam os serviços, permanecem corporativos, bem como a oficina e a manutenção predial. Isto foi feito desta maneira a fim de facilitar a troca de informações entre os planejadores de cada UGB³.

O setor de manutenção da **EMBRACO** trabalha com modernos conceitos de manutenção, como o TPM (**T**otal **P**roductive **M**aintenance). A *Fábrica II* possui seu setor totalmente informatizado utilizando o software **MAXIMO** para o gerenciamento da manutenção (ver capítulo referente ao **MAXIMO**).

² O layout fabril da *Fábrica II* é do tipo celular, baseado em produto. Cada célula é composta por máquinas que estão dispostas em linha. Daí se utilizar o nome "linhas de fabricação".

³ UGB significa Unidade de Gerenciamento Básico e corresponde aos setores da fábrica (Usinagem, Motores, etc).

Pode-se dividir os serviços realizados pela manutenção nos seguintes tipos:

- Manutenção Corretiva/Emergencial:

Este tipo de manutenção diz respeito a alguns serviços emergenciais ou não planejados que devem ser realizados a fim de evitar uma futura pane do equipamento, ou simplesmente corrigir um mau funcionamento da máquina.

- Manutenção Preventiva:

Este tipo de manutenção representa a maioria das ordens de serviço⁴ geradas neste setor. Constitui-se de planos de trabalhos⁵ frequentes definidos pelos planejadores do setor de manutenção, com início e fim definidos.

- Reporte de Eventos:

Representa qualquer evento não esperado (não programado) numa máquina, que pode ou não parar o trabalho (produção), mas não necessariamente requer uma equipe de manutenção para consertá-lo. Por exemplo, se algum equipamento apresenta um ruído, vazamento ou vibração anormal, talvez o próprio operário possa resolver o problema, mas é desejável que o evento seja reportado, para que se mantenha um histórico do equipamento.

- Manutenção Preditiva:

A manutenção preditiva, apesar de ainda em fase inicial de implantação, pretende trabalhar com instrumentação de modo a prever possíveis quebras nos equipamentos. Neste tipo de manutenção, são realizadas análises de ruído, temperatura, vibrações, etc.

O processo de emissão de ordens de serviço funciona da seguinte forma: as ordens de serviço preventivas e de lubrificação

⁴ Ordem de Serviço é um documento gerado, de acordo com a frequência de manutenção das máquinas e outros equipamentos, que informa o que deve ser feito, por quem e onde.

⁵ Um plano de trabalho é uma descrição detalhada das operações a serem efetuadas num serviço de manutenção.

são geradas no **MAXIMO** e emitidas geralmente no final da semana anterior à que serão executadas. Estas ordens são repassadas aos técnicos, que realizam os serviços e os reportam ao **MAXIMO**, atualizando a base de dados.

As ordens de serviço corretivas são emitidas sempre imediatamente antes de sua execução, caso o defeito não tenha parado o equipamento. Nas situações emergenciais, quando o equipamento pára, o serviço é realizado e só então reportado ao **MAXIMO**.

Desta forma fica contextualizado um dos sistemas que fazem parte deste trabalho: o **MAXIMO**.

Capítulo II - *Conceitos*

Abaixo seguem alguns conceitos a respeito de Base de Dados, Redes de comunicação e alguns outros mecanismos, que servirão para um melhor entendimento do estudo que será feito adiante.

2.1 - Base de Dados Relacional [1], [3]

Existem três modelos de sistemas de gerenciamento de base de dados: *hierárquico*, *rede* e *relacional*, que atualmente é o mais aceito.

O modelo relacional para banco de dados surgiu a partir de estudos sobre teoria de conjuntos. Um conjunto é na verdade, uma série de objetos que possuem uma regra em comum, e que nos permite identificar se determinado objeto faz parte ou não daquele conjunto.

O modelo relacional apresenta aspectos importantes:

- Estruturas: são objetos bem definidos que armazenam os dados da base. Estas estruturas e os dados contidos nelas podem ser manipulados por operações;
- Operações: são ações que permitem aos usuários manipular os dados e as estruturas da base. Estas operações devem se adequar a uma série de regras de integridade pré-definidas;
- Regras de Integridade: são leis que determinam quais operações são permitidas sobre os dados e as estruturas da base. Servem como proteção destes dados e estruturas.

Uma base de dados relacional apresenta uma estrutura de dados num nível bastante simples. Esta estrutura é composta por tabelas bi-dimensionais que contém linhas e colunas de dados. Com

a utilização de tabelas, a modelagem dos dados fica fácil e flexível, além de serem fáceis de visualizar.

Uma coleção de tabelas pode representar relacionamentos complexos entre dados. O relacionamento entre duas tabelas é definido por uma coluna que ambas possuem. Isto torna simples a visualização e o entendimento destas ligações (ver a *figura 2.1*).

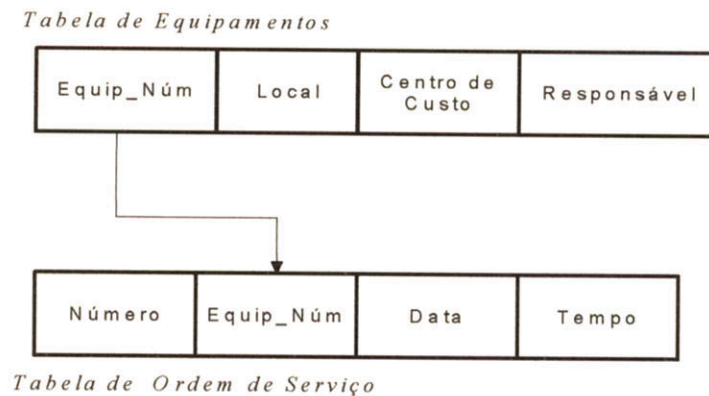


figura 2.1- Exemplo de relacionamento entre tabelas de uma base de dados relacional

Os sistemas de gerenciamento de base de dados relacional oferecem benefícios como:

- independência entre o armazenamento físico dos dados e a estrutura lógica da base de dados;
- grande flexibilidade para elaborar a estrutura da base de dados;
- redução de redundâncias, o que reduz a ocupação de espaço em disco.

Outra vantagem do modelo relacional é que em um comando pode-se recuperar linhas de uma ou várias tabelas para manipulação dos dados. Esta funcionalidade significa que com apenas um comando relacional pode-se além de recuperar dados, atualizar ou deletar múltiplas linhas armazenadas em tabelas.

2.1.1 - Comparação com Outros Modelos

Os modelos *hierárquico* e de *rede* são procedurais. Ou seja, para achar um registro, é necessário navegar, ou achar um caminho até o registro desejado, e fornecer múltiplos comandos procedurais que dizem ao sistema qual caminho tomar, passo-a-passo. É necessário um entendimento profundo do modo como os dados estão armazenados. Uma vez criada e carregada uma base de dados, é difícil alterá-la.

Ao contrário, uma base relacional possibilita uma navegação automática. Não é necessário conhecer como a base representa os dados internamente para obtê-los. Isto facilita o acesso aos dados.

Nos modelos *hierárquico* e de *rede*, alguns dados são armazenados como valores e outros como ponteiros. Para associar dois registros, é necessário a utilização de ponteiros. O usuário deve tomar a decisão a respeito da forma como armazenar o dado, se como valor ou ponteiro. Estes modelos utilizam relações estáticas.

Numa base relacional, pode-se utilizar quaisquer valores para associar uma tabela com outra, e define-se as relações entre valores quando se faz uma procura, e não quando se cria tabelas. Isto aumenta a flexibilidade.

2.2 - SQL [3],[6]

SQL (Structured Query Language) é um conjunto completo de comandos que permitem ao usuário acessar uma base de dados relacional.

É uma interface padrão para a maioria dos bancos de dados relacionais. Possui uma estrutura de comandos simples para definir, acessar e manipular dados. Também é não-procedural. Quando se utiliza SQL, especifica-se apenas o que se deseja, e não como fazer para obter o dado desejado. Quando se acessa algum dado via SQL, é necessário apenas o nome da tabela e as colunas. Não é necessário descrever um método de acesso.

2.2.1 - Tipos de Comandos SQL

Com SQL pode-se:

- Criar tabelas na base de dados;
- Armazenar dados;
- Recuperar dados;
- Modificar dados e estrutura de tabelas;
- Restringir acesso.

Os comandos SQL são divididos basicamente da seguinte forma:

- Comandos de Definição (DDL - Data definition Language): Estes comandos criam e alteram objetos na base de dados como tabelas, etc. Como exemplo tem-se *CREATE*, *ALTER* e *DROP*;
- Comandos de Manipulação (DML - Data manipulation Language): Estes comandos adicionam, atualizam ou deletam dados. Também são usados para realizar procuras na base utilizando, para isso, operadores relacionais (>, <, =, >=, etc). *INSERT*, *UPDATE*, *DELETE* e *SELECT* são exemplos destes comandos;
- Comandos de Controle de Transação: Asseguram a integridade dos dados num ambiente de concorrência quando estes são modificados. Exemplos destes comandos são *COMMIT*, *ROLLBACK* e *SAVEPOINT*.

Como exemplo simples, supomos ter a seguinte procura a realizar:

"Desejo saber o código e a descrição dos equipamentos do setor de montagem"

O código em SQL ficaria assim:

```
SELECT EQUIP_NUM, DESCRIÇÃO
FROM EQUIPAMENTOS
WHERE LOC = 'MONTAGEM'
```

A tradução deste comando, ficaria da seguinte forma:

"Selecione os dados dos campos EQUIP_NUM e DESCRIÇÃO da tabela EQUIPAMENTOS que possuem o campo LOC igual a 'MONTAGEM'".

Outros exemplos de comandos SQL seriam:

```
SELECT NOME FROM OPERÁRIOS;
CREATE TABLE FORNECEDORES;
SELECT * FROM EQUIPAMENTO, OS
    WHERE EQUIPAMENTO.EQUIP_NUM = OS.EQUIP_NUM;
ALTER TABLE EQUIPAMENTO RENAME TABLE EQUIPTO;
DROP TABLE EQUIPTO;
```

2.3 - Redes e Base de Dados [3]

A necessidade de se possuir computadores interligados fez surgir as redes de computadores e a tecnologia de sistemas computacionais distribuídos.

Estes tipos de sistemas permitem que a informação seja disponibilizada em qualquer parte da rede, e que o processamento desta informação seja dividido entre os demais elementos da rede.

Com as redes, os PC's podem se comunicar e compartilhar recursos como, por exemplo, uma base de dados.

2.3.1 - O modelo Cliente-Servidor

Esta arquitetura de rede é uma forma de sistema distribuído onde as atividades computacionais são distribuídas entre os computadores da rede.

Em geral, os clientes são PC's de menor porte (capacidade de processamento e armazenamento) e rodam programas especiais para clientes. Já os servidores são computadores mais poderosos, com grande capacidade de armazenamento e memória RAM, bem como alta velocidade de processamento. Os clientes requisitam serviços ao servidor, que recebe a mensagem e envia o resultado de volta ao cliente. Os recursos específicos de cada cliente não

podem ser compartilhados entre os demais elementos da rede. A *figura 2.2* ilustra esta arquitetura.

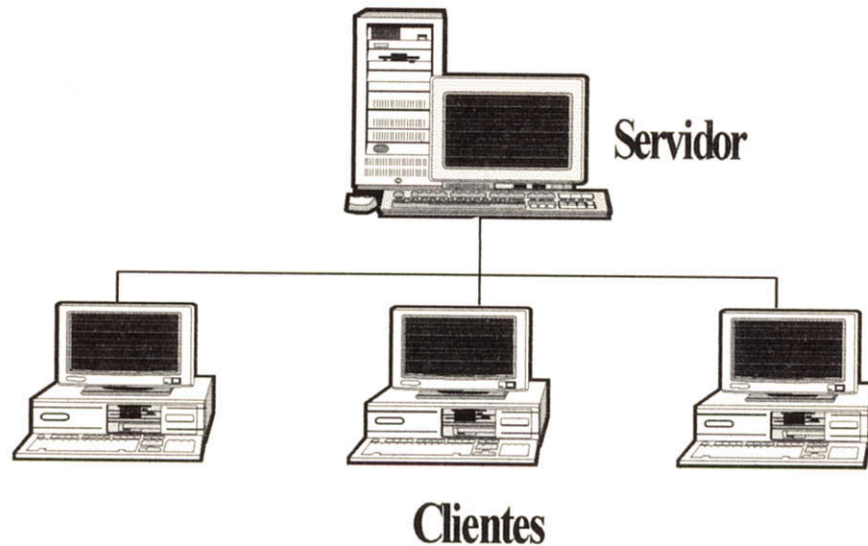


figura 2.2 - A arquitetura Cliente x Servidor

2.4 - Instâncias de Base de Dados [6]

Área de memória compartilhada, que contém dados e informações de controle pertinentes a um determinado sistema. Por exemplo, atualmente, a base de dados do Sistema de Informação Industrial (ver capítulo referente ao **SII**) possui duas instâncias: uma para o sistema de coleta de dados do chão-de-fábrica, e outra para o sistema de planejamento da produção (**FI2**). A título de exemplo, a *figura 2.3* mostra a representação de instâncias de base de dados sob um sistema de gerenciamento de banco de dados ORACLE.

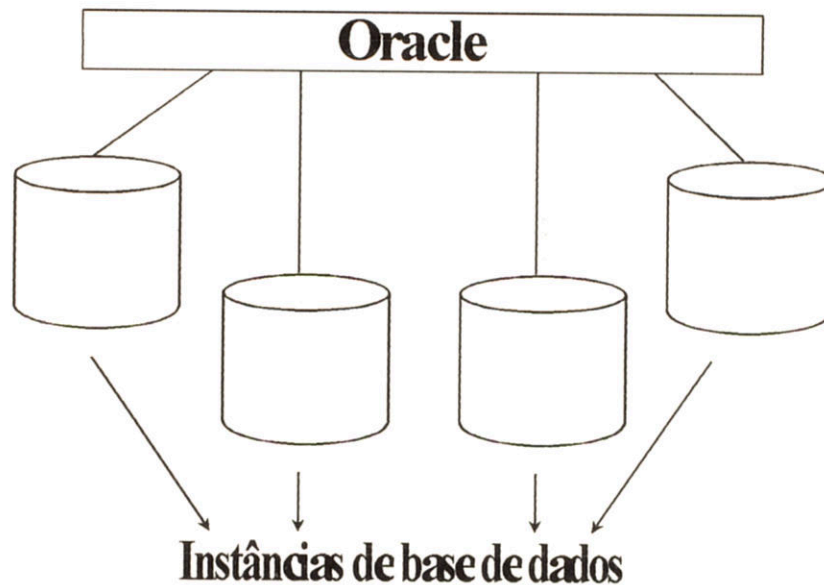


figura 2.3 - Exemplo de instâncias de bases de dados

2.5 - Processo [2] e [6]

Um processo (*task*, tarefa) é um programa em execução, isto é, um conjunto de operações sendo executadas em ordem sequencial.

2.6 - Trigger [6]

Trigger's são procedimentos disparados pelo ORACLE (tipo de sistema de gerenciamento de base de dados), a cada vez que é dado um INSERT, UPDATE ou DELETE sobre uma tabela.

Os *trigger's* são armazenados na base, mas separados da tabela a que estão associados.

Um *trigger* é constituído por comandos PL/SQL, que combinam a flexibilidade dos comandos SQL com a funcionalidade dos comandos de uma linguagem de programação estruturada, como IF...THEN, WHILE, LOOP, etc.

As principais funções dos *trigger's* são:

- atualizar valores em outras tabelas ou em outras colunas;

- prevenir transações inválidas;
- reforçar a segurança no acesso aos dados;
- etc.

Um trigger é composto por três partes:

- Evento de disparo: evento que causa o disparo do *trigger*;
- Restrição de disparo: especifica uma expressão booleana que deve ser verdadeira para que o *trigger* possa ser disparado;
- Ação de disparo: é um bloco que contém comandos SQL e outros comandos a serem executados quando disparado o *trigger*.

Existem basicamente dois tipos de *trigger*:

- Before trigger: a ação de disparo é executada imediatamente antes da execução do evento de disparo;
- After trigger: a ação de disparo é executada imediatamente depois da execução do evento de disparo;

Os *trigger's* podem também estar associados apenas às linhas da tabela que serão alteradas. Ou seja, a ação de disparo só afetará as linhas que forem inseridas, atualizadas ou deletadas.

Um exemplo de código de *trigger* pode ser visto no ANEXO II.

2.7 - ODBC [7]e[9]

ODBC (Open Data Base Connectivity) é uma especificação de uma interface para aplicativos, tais como o Microsoft QUERY¹, que os permite acessar múltiplos sistemas de gerenciamento de base de dados, utilizando SQL.

¹ Software que permite graficamente realizar procuras de dados na base, sem a necessidade de se utilizar os comandos da linguagem SQL, que ficam transparentes ao usuário.

As base de dados podem ser locais ou residir num servidor de rede. Uma vez configurada a fonte de dados no Controlador de ODBC, o acesso fica transparente ao usuário.

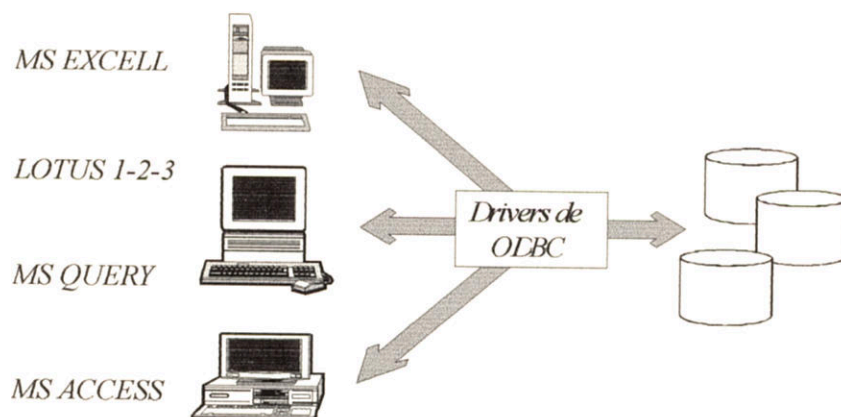


figura 2.4 - Exemplo de algumas aplicações com ODBC

Capítulo III - *O software MAXIMO*

3.1 - Introdução

O **MAXIMO** é um sistema que visa atender as necessidades operacionais e de gerenciamento de serviços de manutenção das fábricas. Sob um outro ponto de vista, o **MAXIMO** pode ser entendido como um gerenciador de banco de dados dedicado a área de manutenção. Foi desenvolvido pela PSDI (Project Software & Development, Inc.), empresa norte-americana, que atualmente é representada no Brasil pela empresa paulista MIPS Sistemas.

Este software automatiza e simplifica o controle da manutenção, facilitando a implantação de técnicas avançadas de gerenciamento como, por exemplo, a *TPM* (Total Productive Maintenance). O **MAXIMO** gerencia todos os aspectos da manutenção, incluindo:

- Gerenciamento de Ordens de Serviço
- Planejamento e Programação de Manutenções
- Gerenciamento de Mão-de-obra
- Compras
- Gerenciamento de Recursos
- Relatórios e Análises

O **MAXIMO** utiliza um sistema de gerenciamento de base de dados relacional para armazenar os dados da manutenção.

Os dados podem ser acessados diretamente via SQL, o que facilita a integração do **MAXIMO** com outros aplicativos.

Trabalha com vários tipos de base de dados, entre elas SQLBASE, ORACLE, DB2, etc. Possui uma arquitetura cliente/servidor, suportando qualquer número de usuários e sendo capaz de rodar sobre qualquer tipo de plataforma de servidor - DOS, OS/2, NLM, UNIX. Na **EMBRACO**, o **MAXIMO** utiliza uma base de dados SQLBASE, da GUPTA Technologies. O **MAXIMO** foi desenvolvido em ambiente WINDOWS, possuindo uma interface gráfica amigável, herdando assim, todas as funcionalidades do próprio ambiente.

Numa configuração cliente/servidor sobre uma LAN (Local Area Network), a melhor distribuição dos componentes do sistema é sugerida da seguinte forma:

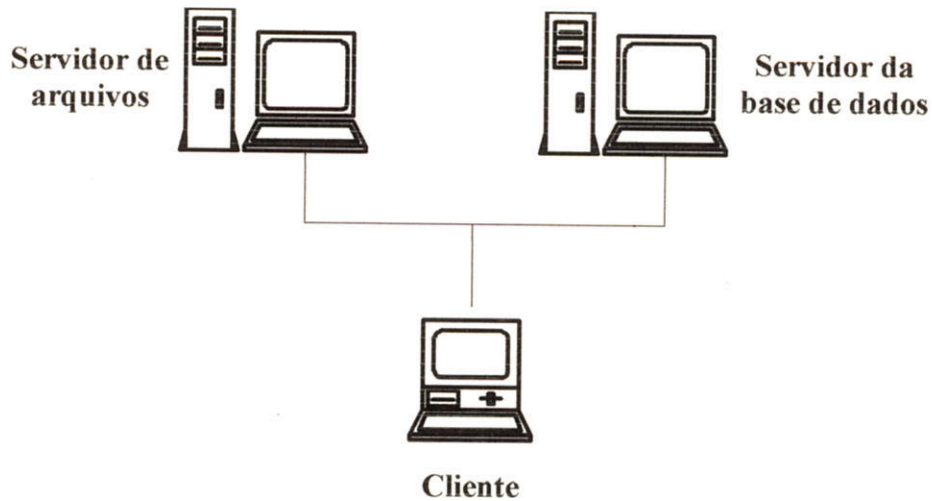


figura 3.1 - Configuração recomendada para arquitetura cliente/servidor

A configuração do sistema atual na **EMBRACO** possui um único servidor, tanto para arquivos quanto para a base de dados (ver *figura 3.2*). Isto prejudica a performance do sistema, pois o servidor não fica dedicado a oferecer serviços exclusivos da base de dados:

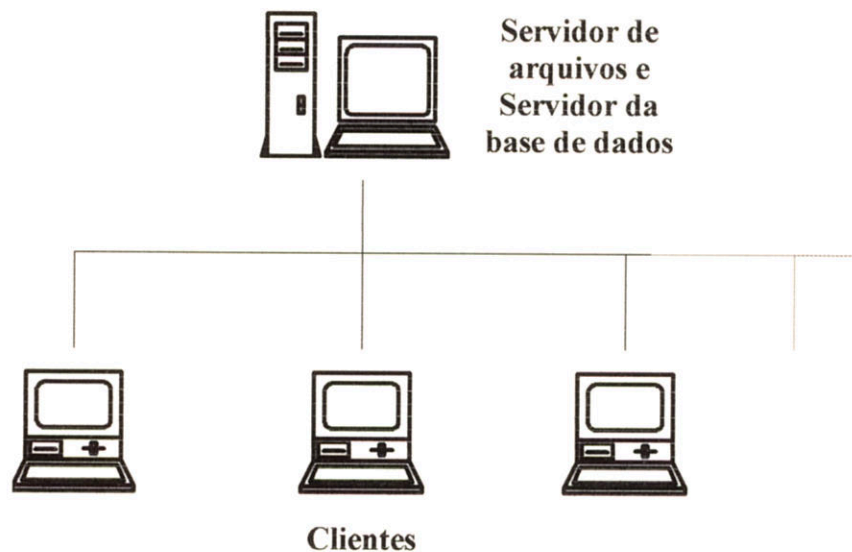


figura 3.2 - Arquitetura do sistema existente na EMBRACO

Os clientes do servidor **MAXIMO** são PC's espalhados pela *Fábrica II* como um todo.

O **MAXIMO** é totalmente "customizável", ou seja, todas as suas telas podem ser modificadas inteiramente conforme a necessidade do usuário, incluindo ou retirando campos, alterando menus, criando novas telas através de clonagem¹, etc [4]. Pode-se criar botões em suas telas, associando-os a linhas de comandos que acionam outros aplicativos como WORD, CAD, EXCEL, etc.

O **MAXIMO** é composto por 12 módulos (ver *figura 3.3*) interconectados. Estes módulos se relacionam de tal forma que pode-se facilmente navegar através deles a fim de buscar informações em outros lugares. Cada módulo é composto por telas que permitem ao usuário a entrada dos dados. Estas telas possibilitam a aquisição de informações suficientes e necessárias para o completo gerenciamento da manutenção.

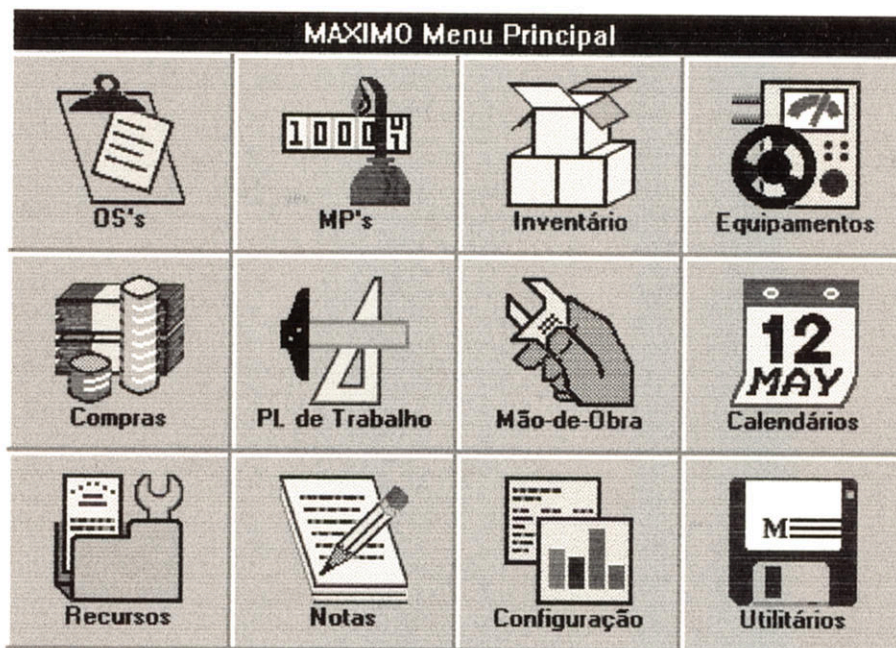


figura 3.3 - Tela principal do MAXIMO com seus 12 módulos

Para um melhor esclarecimento do sistema, os 12 módulos serão brevemente descritos abaixo [5].

¹ Clonagem é um processo que o MAXIMO possui que permite ao usuário duplicar uma tela já existente e também modificá-la da forma como quiser, criando assim uma nova tela no sistema.

3.2 - Módulo Ordens de Serviço

As ordens de serviço são os elementos essenciais da manutenção. No software **MAXIMO** elas servem para rastrear eventos que ocorreram e trabalhos que foram executados, e este módulo possui todas as funcionalidades necessárias para o processamento dos dados referentes aos serviços diários.

Dentro deste módulo existem quatro telas: Acompanhamento de OS (Ordem de Serviço), Requisição de Serviços, Reporte Rápido e Anomalia, sendo que esta última foi criada especialmente para a **EMBRACO**.

A tela de Acompanhamento de OS (*figura 3.4*) permite acesso a todas as informações necessárias para um detalhado planejamento das ordens de serviço, incluindo operações, materiais, mão-de-obra, custos, equipamentos, análise de falhas, etc.

figura 3.4 - Tela de Acompanhamento de Ordens de Serviço

A tela de Requisição de Serviços é uma maneira fácil e rápida de requisitar serviços de manutenção. Pode se limitar a um texto, ou pode trazer maiores informações como o nome do requisitante, código do problema, prioridade, etc.

A tela de Reporte Rápido possibilita facilmente relatar o serviço realizado.

Quando se cria uma ordem de serviço dentro de uma destas telas, dá-se início ao processo de manutenção, criando-se também um registro histórico sobre o trabalho feito.

Cada ordem de serviço, desde sua geração até o seu encerramento, passa por diversos estados, citados a seguir:

- "Waiting for Approval"
A ordem foi criada, porém não aprovada pelo planejador.
- "Approved"
A ordem é finalmente aprovada pelo planejador ou responsável.
- "In Progress"
A ordem é emitida e começa a sua execução.
- "Waiting for Materials"
Parada solicitada devido à falta de material para continuar a manutenção.
- "Complete"
Ordem é completada e espera fechamento.
- "Canceled"
Ordem é cancelada por algum motivo.
- "Closed"
Ordem é finalmente encerrada.

Pode-se colocar estes estados na forma de um grafo, como a seguir:

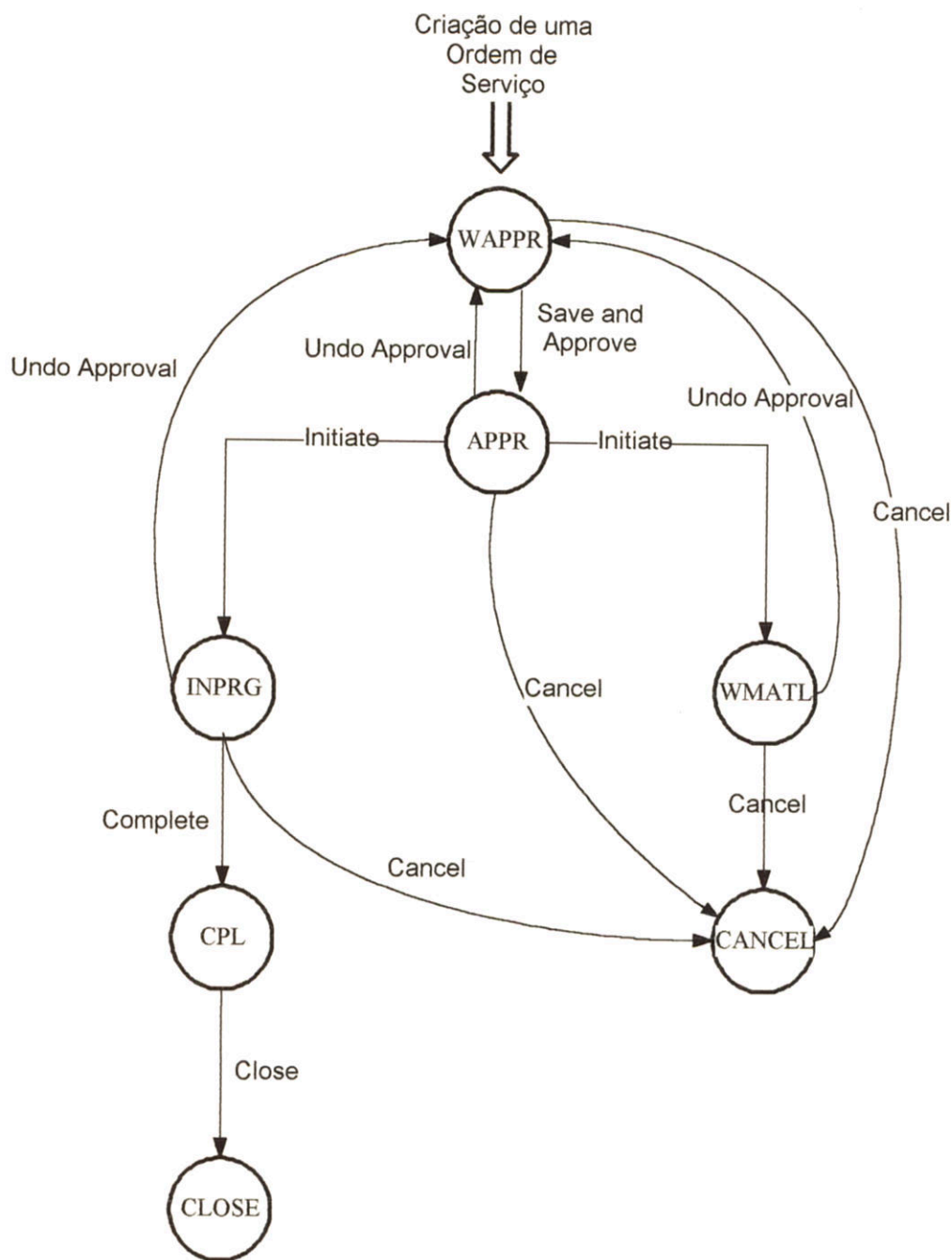


figura 3.5 - Grafo de Estados de Ordens de Serviço

3.3 - Módulo Manutenção Preventiva

Este módulo é usado para gerar ordens de manutenção que devem ocorrer com determinada frequência.

O **MAXIMO** automaticamente gera as ordens de serviço quando a frequência especificada é atingida. Esta frequência é

definida pelo planejador (usuário) e indica o número de dias entre a geração de duas ordens de serviço.

Neste módulo são criados planos mestres de manutenção preventiva. Um plano mestre contém uma sequência de planos de trabalho que devem ser realizados com uma frequência determinada. Cada plano mestre pode ter um ou mais planos de trabalho. Isto permite associar cada plano mestre a vários tipos de serviços - semanal, mensal, semestral, etc. O que se faz é associar um número sequencial a cada plano de trabalho, informando ao sistema qual plano de trabalho utilizar a cada vez que uma ordem de serviço for gerada a partir daquele plano mestre.




Manutenção Preventiva - [Main]			
Arquivo Editar Visualizar Opções Tabela B.Dados Seleção Janela Ajuda			
Salva	Inserir	Limpa	Prévio Próximo
Vis.Ger	Retorna	Ret/Sel	Telas
Menu	Mestres	Seq PT	Ger OS
Ve Seq			
MP Mestre	LB-AM0004	Plano de lubrificação	
Equipamento	AM0004	Alimentador de alumínio (Italpresse)	
Prox.Plano Tr.	EM0004-04	Plano de lubrificação trimestral	
Detalhes			
Supervisor	CLAUDIO	Prioridade	0
Centro Custo	66013	Conta	
Equipe	MOTORES	Requer Parada?	Y
		Tipo Trabalho	LB
		Sub Conta	
		Interrompivel?	N
MPs Baseadas em Tempo		Geracao da Ordem de Serviço	
Frequência(dias)	84	Primeiro dia de Inicio	26/06/95
Prox.Data	04/03/96	Data da Ultima Iniciada	11/12/95
		Ult.Dia Fechada	04/10/95
MPs baseadas em Medidas		Usa o Inicio destino?	Y
Frequência(Unid.Medida)	0.00	Sequenciada?	N
Unid.		Contador	4
Media Medida Unid/Dia	0.00		
Leitura na ult.O.S.	0.00		
Data			
Ultima Leitura	0.00		
Data			
Prox.Leitura Prevista	0.00		
Data			
VERIFICAR		Proximo registro recuperado.	ABC  

figura 3.6 - Tela de Manutenção Preventiva

3.4 - Módulo Planos de Trabalho

Neste módulo se faz o cadastramento dos planos de trabalho, descrevendo, além das operações, mão-de-obra, lista de peças e ferramentas necessárias para a realização do serviço (ver figura abaixo).

Cada plano de trabalho é dividido em uma série de operações, e cada uma destas possui estimativas de peças, mão-de-obra e ferramentas.

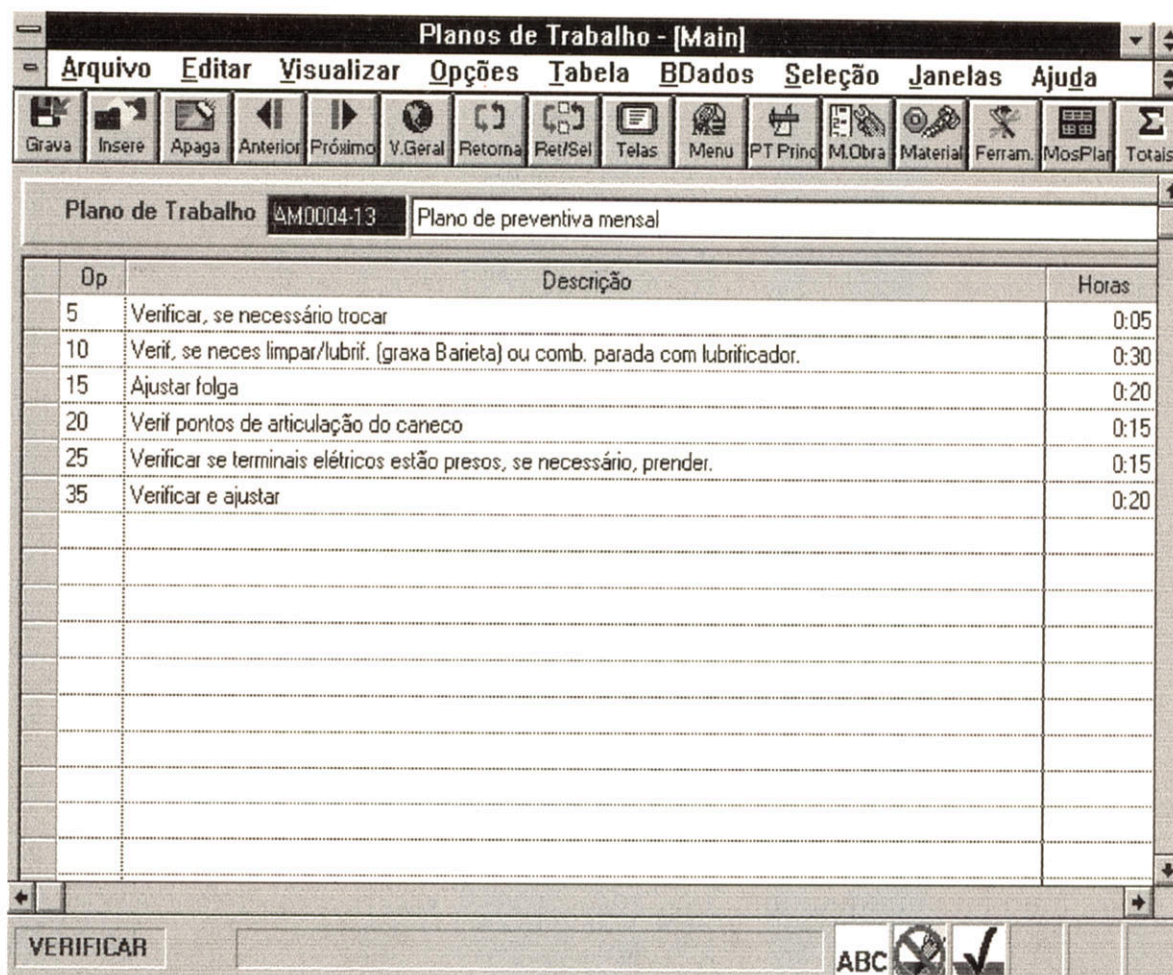


figura 3.7 - Tela de Plano de Trabalho

3.5 - Módulo Equipamento

O módulo de Equipamento permite manter e atualizar dados sobre os equipamentos da fábrica. Possui informações bastante completas, incluindo dados sobre fabricante, garantia, custos e até medições efetuadas no equipamento.

O **MAXIMO** permite que seja construída uma estrutura de equipamentos em forma de árvore, possibilitando assim, que novas peças de equipamentos sejam incluídas na base de dados, bem como suas relações com outros equipamentos.

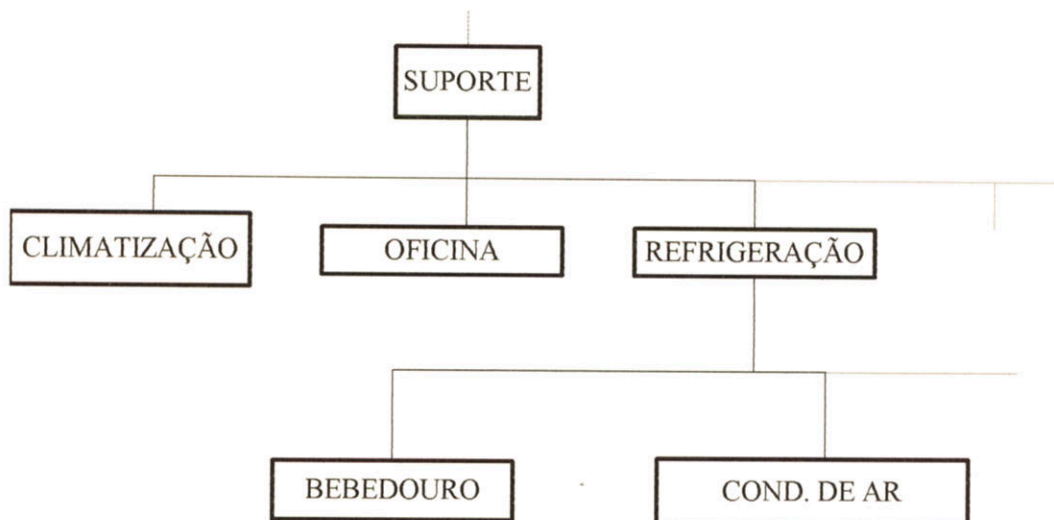


figura 3.8 - Exemplo de estrutura de equipamentos

Além da tela principal onde se inclui os dados pertinentes ao equipamento, há a tela de Monitoramento de Condições que possibilita a entrada de dados sobre medição de determinado equipamento. Podem ser estabelecidos pontos de máximo e mínimo para efeito de emergência. Quando alcançados determinados valores, pode-se gerar automaticamente uma ordem de serviço corretiva.

Outro dado interessante é que se pode criar uma estrutura de falhas. Isto facilita o reporte de quebras de equipamentos, e ajuda em futuras análises do histórico do equipamento.

Equipamentos - [Main]	
Arquivo Editar Visualizar Opções Tabela BDados Seleção Janelas Ajuda	
Salva	Inserir
Limpa	Prévio
Próximo	Vis.Ger.
Retorna	Ret/Sel
Telas	Menu
Eq Print	Submod
MPs	Esp.Eq.
Docs	List Eq
Equipamento	AM0004 Alimentador de alumínio (Italpresse)
Pertence A	TAMPA CILI Célula da tampa do cilindro
Fabricante	ITALPRES
Fornecedor	
Dados Adicionais	
Centro de Custo	66013
Local	BI 28
No. Ativo	
No. Série	
CIPA	
Calendário	
Classe Equipamento	
Classe Falha	
No. Peça	
Prior. Eqto	
Situação Parada	
Ativo?	Y
Data	04/05/95
Total Parada	0:00
Custos	
Total	R\$0.00
No Ano	R\$0.00
Orçado	R\$0.00
N/ Debitado	R\$0.00
Total n/Debitado	R\$0.00
Dados Aquisição	
Data Instalação	
Data Garantia	
Preço Aquisição	R\$0.00
Custo Substituição	R\$0.00
Modificado	
Modificado por	TESTE
Data	29/08/95
VERIFICAR	Primeiro registro recuperado.
ABC	

figura 3.9 - Tela de Equipamentos

3.6 - Módulo Compras

Este módulo possui ferramentas para preparar e imprimir requisições e ordens de compra.

A partir dos módulos de Ordem de Serviço e Inventário, pode-se criar requisições de compra, e destas requisições são emitidas as ordens de compra.

3.7 - Módulo Inventário

O **MAXIMO** mantém dados sobre peças em estoque e peças que não possuem estoque. Indica também quando o estoque desce

abaixo do nível mínimo indicado pelo usuário, e cria Requisições de Compra para regularizar o estoque.

O controle de estoque é feito via curva ABC, onde se definem grupos baseados em valores monetários e em taxas de demandas das peças.

A atualização do estoque é automática. Quando acontece um reporte de peças nas ordens de serviço, isto é contabilizado no estoque. Quando chegam novas peças através de compra, os balanços são acrescidos e os custos recalculados.

É importante lembrar que atualmente, na **EMBRACO**, toda a parte de inventário e compras é feita por um sistema desenvolvido pela empresa, que roda em mainframe IBM.

3.8 - Módulo Recursos

Este módulo possui três telas que são usadas para o gerenciamento os recursos. Cada tela é responsável por um tipo de recurso. São elas:

- Empresas: informações de vendedores, fornecedores, fabricantes, entre outras.
- Contrato de Serviços: dados sobre contratos de serviços com vendedores ou fabricantes.
- Ferramentas: informações sobre o ferramental usado nos serviços de manutenção, assim como custos associados.

3.9 - Módulo Mão-de-Obra

Contém informações sobre toda mão-de-obra disponível para serviços. Neste módulo são cadastrados os funcionários, contendo dados pessoais, bem como um detalhamento das horas trabalhadas (incluindo hora-extra), salários individuais, etc.

Pode-se criar equipes de serviço, com escalas de hora-extra, custo de hora-homem.

A partir destes dados pode ser feita uma análise bastante completa sobre cada mão-de-obra cadastrada.

3.10 - Módulo Calendários

O módulo de Calendários é usado para programar os equipamentos e a mão-de-obra. Um calendário é definido com datas de início e fim, férias, feriados, que servem para a determinação dos dias de trabalho.

A partir dos calendários cadastrados, pode-se visualizar as horas de trabalho disponíveis para qualquer dia do ano, ou entre duas datas especificadas.

Pode-se também gerar relatórios e comparar a mão-de-obra requerida e a mão-de-obra disponível para possíveis ajustes.

3.11 - Módulo Notas

O módulo de Notas permite que o usuário crie suas próprias tabelas e telas a fim de complementar as telas do **MAXIMO**. Estas tabelas podem ser acessadas a partir de qualquer módulo ou dentro do próprio módulo de Notas.

Pode-se criar tabelas de propósito geral (que contém qualquer tipo de informação), tabelas associadas especificamente com um módulo do **MAXIMO**, e tabelas de especificações de equipamento, contendo informações pertinentes a uma classe específica de equipamento (motores, bombas, válvulas etc.).

3.12 - Módulo Configuração

Este módulo é usado para configurar a base de dados, definir níveis de segurança de usuários, e configurar aplicações que serão acionadas a partir dos módulos (telas) do **MAXIMO**. Possui seis telas:

- Configuração do Banco de Dados: Permite a customização da base de dados, incluindo ajustes como tamanho de campos e criação de novos tipos de dados;
- Documentos: Cadastrar e ligar informações, como desenhos ou fontes de programas, a equipamentos e peças de inventário;
- Segurança de Usuários: Pode-se limitar o direito de acesso dos usuários a módulos, aplicações, telas e opções apenas organizando grupos de usuários com hierarquias diferentes.
- Carregamento de Aplicações: Permite conectar aplicações do Windows a campos ou botões em telas desejadas.
- Configuração de Aplicações: Nesta tela se faz a alteração do menu principal do **MAXIMO**, além da clonagem de telas para customização.
- Relatórios e Outras Aplicações: Cadastrar relatórios e outras aplicações para uso com o **MAXIMO**.

A maioria dos relatórios, como a impressão das ordens de serviço, é confeccionada no software SQR (Structured Query Report). Existe também o software QUEST, que é um produto que faz parte do pacote **MAXIMO** e que permite fazer consultas na base de dados do **MAXIMO** para a confecção de relatórios. O problema é que estes dois softwares são pouco conhecidos e também bastante limitados em termos de funções oferecidas, sem falar da linguagem pobre e da péssima qualidade visual apresentada principalmente pelo SQR.

A base de dados SQLBASE também suporta controladores de ODBC (ver capítulo II). Isto traz a possibilidade de se gerar relatórios no EXCEL apertando apenas um botão dentro de uma planilha. A base de dados é consultada via Query e ODBC, e os dados são trazidos para dentro do EXCEL. Tudo isso de forma automática e transparente para o usuário. Alguns relatórios já gerados desta forma estão descritos no ANEXO III. Uma próxima etapa é, então, migrar todos os relatórios que hoje são feitos em SQR e QUEST para EXCEL, visto que o último é um software muito mais poderoso e conhecido.

3.13 - Módulo Utilidades

O módulo de utilidades contém funções que permitem ao usuário customizar o sistema e administrar a base de dados. Três telas são oferecidas neste módulo:

- SQL Interativo: Permite o acesso direto à base de dados através de comandos SQL;
- Editor de Telas: Através desta opção, o usuário do **MAXIMO** pode redesenhar as telas da forma como quiser;
- Arquivar dados do MAXIMO: Permite remover certos registros do banco de dados que já não são mais utilizados. Isto permite "enxugar" o banco de dados de tempos em tempos. Como curiosidade, o banco de dados da **EMBRACO** ocupa hoje um total de 40 MB, o que deixa o sistema bastante carregado, acarretando em perda de performance. Como este volume cresce dia a dia, arquivar dados é essencial.

3.14 - MAXIMO Scheduler

O software Scheduler é um produto que auxilia os planejadores da manutenção na programação dos serviços. Utilizando técnicas avançadas de comparação entre recursos requeridos e recursos disponíveis, o Scheduler ajuda na elaboração de planos mais eficientes e na otimização da mão-de-obra. Pode também ser usado para prever o impacto da manipulação de ordens de serviço e mão-de-obra no tempo.

As ordens de serviço são geradas no **MAXIMO**, e depois passadas ao Scheduler, a fim de serem programadas.

A programação é mostrada na tela na forma de um gráfico de Gantt, juntamente com um histograma que mostra a comparação entre mão-de-obra disponível e a mão-de-obra requerida nas ordens de serviço programadas. Toda a programação pode ser visualizada através de relatórios pré-definidos.

Capítulo IV - O Sistema de Informação Industrial - SII

A proposta de trabalhar com um nível de estoque na *Fábrica II* (~US\$ 110.000) bem inferior ao da *Fábrica I* (~US\$ 1.200.000), motivou a aquisição de um sistema de informação para a *Fábrica II*, a fim de fazer a sincronização da produção.

A *figura 4.1* compara o fluxo de dados na *Fábrica I* e na *Fábrica II*. Enquanto na *Fábrica I* todos os dados são transportados sobre papel, na *Fábrica II* os dados trafegam sobre uma rede de computadores.

O Sistema de Informação Industrial (**SII**) é um sistema de integração de informação que engloba as atividades de Planejamento, Coleta de Dados e Geração de Relatórios de Acompanhamento da Produção em um sistema único, visando agilizar a disponibilização das informações por toda a Fábrica.

O **SII** é um sistema com arquitetura Cliente x Servidor, composto por diversas estações conectadas em rede. O Servidor é uma máquina RISC com sistema operacional UNIX, onde executam um software de auxílio ao planejamento fino da produção (**FI2**), um sistema de gerenciamento de base de dados relacional Oracle, e diversos processos clientes dessa base de dados. Conectados a este servidor, via TCP/IP, estão coletores industriais de dados e PC's de escritório. Estão ainda conectados a este servidor, via barramento serial RS485, painéis eletrônicos que se encontram espalhados pelo chão-de-fábrica, juntamente com os coletores de dados.

Os coletores de dados são os dispositivos de entrada de dados de produção do sistema. Estes coletores consistem de PC's industriais especiais dedicados à coleta de dados. Através deles, são realizados os apontamentos de quaisquer eventos relevantes, relacionados à produção que ocorram no chão-de-fábrica. Alguns eventos são apontados automaticamente (contagem de peças na entrada e na saída da linha), através de entradas digitais disponíveis nos coletores; outros têm de ser apontados manualmente, via teclado (paradas de máquinas, início e fim de setup ou de produção de uma ordem etc).

manualmente, via teclado (paradas de máquinas, início e fim de setup ou de produção de uma ordem etc).

Nos PC's de escritório executam aplicativos que auxiliam na manutenção do sistema, bem como aplicativos que geram os gráficos e relatórios de acompanhamento da produção citados anteriormente. Estes computadores são utilizados também para o acompanhamento da utilização do sistema. Executa também em PC's de escritório um software de supervisão do sistema, que permite o acompanhamento do estado da produção em toda a fábrica, bem como o fácil acesso aos relatórios de produção.

Os painéis eletrônicos têm a função de repassar aos operadores das máquinas informações a respeito do estado corrente da produção de cada linha, quais os próximos componentes cuja produção está planejada e ainda mostrar mensagens de caráter geral, que podem ser digitadas nos coletores de dados pelos próprios operadores, ou nos PC's de escritório, pelas chefias, gerência ou administração do sistema.

Um esquema simplificado do hardware do **SII** está mostrado na *figura 4.2*, e do software, na *figura 4.3*.

É importante ressaltar que o SII apenas automatiza o fluxo e o processamento de dados de chão-de-fábrica na *Fábrica II*. Os dados a serem coletados e processados são basicamente os mesmos que são coletados e processados manualmente na *Fábrica I*. O processo que acontece hoje com o fluxo de dados de uma fábrica é o mesmo que aconteceu com suas máquinas. Antigamente, os movimentos das partes de uma máquina eram realizados manualmente pelos operadores. Com a automação, os movimentos passaram a ser realizados por dispositivos elétricos, hidráulicos e pneumáticos, que foram incorporados às máquinas. Do mesmo modo, o **SII** automatiza um processo que era antes totalmente manual, com o objetivo de tornar a *Fábrica II* mais ágil e flexível.

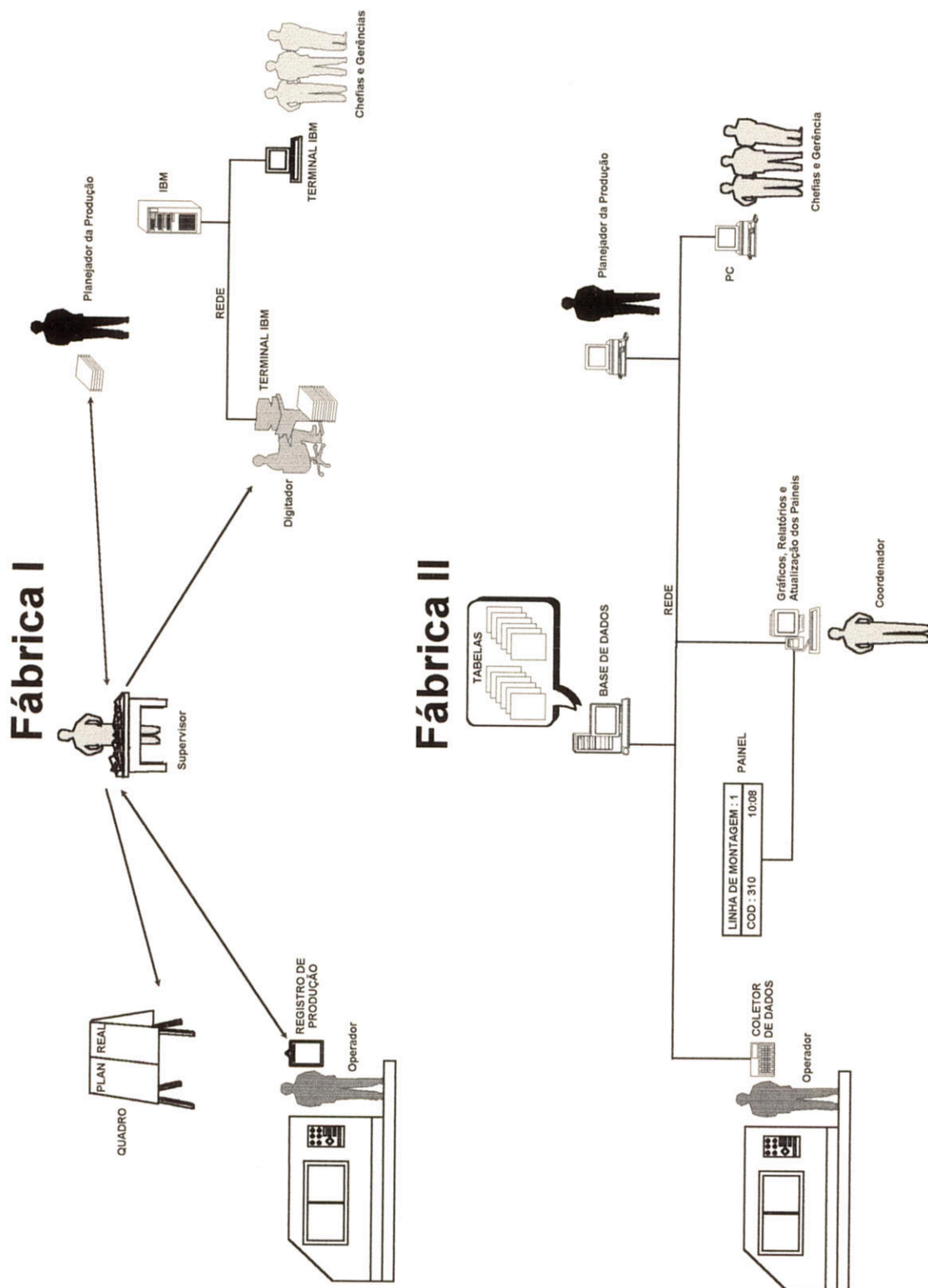


figura 4.1 - Comparação do fluxo de dados nas Fábricas I e II

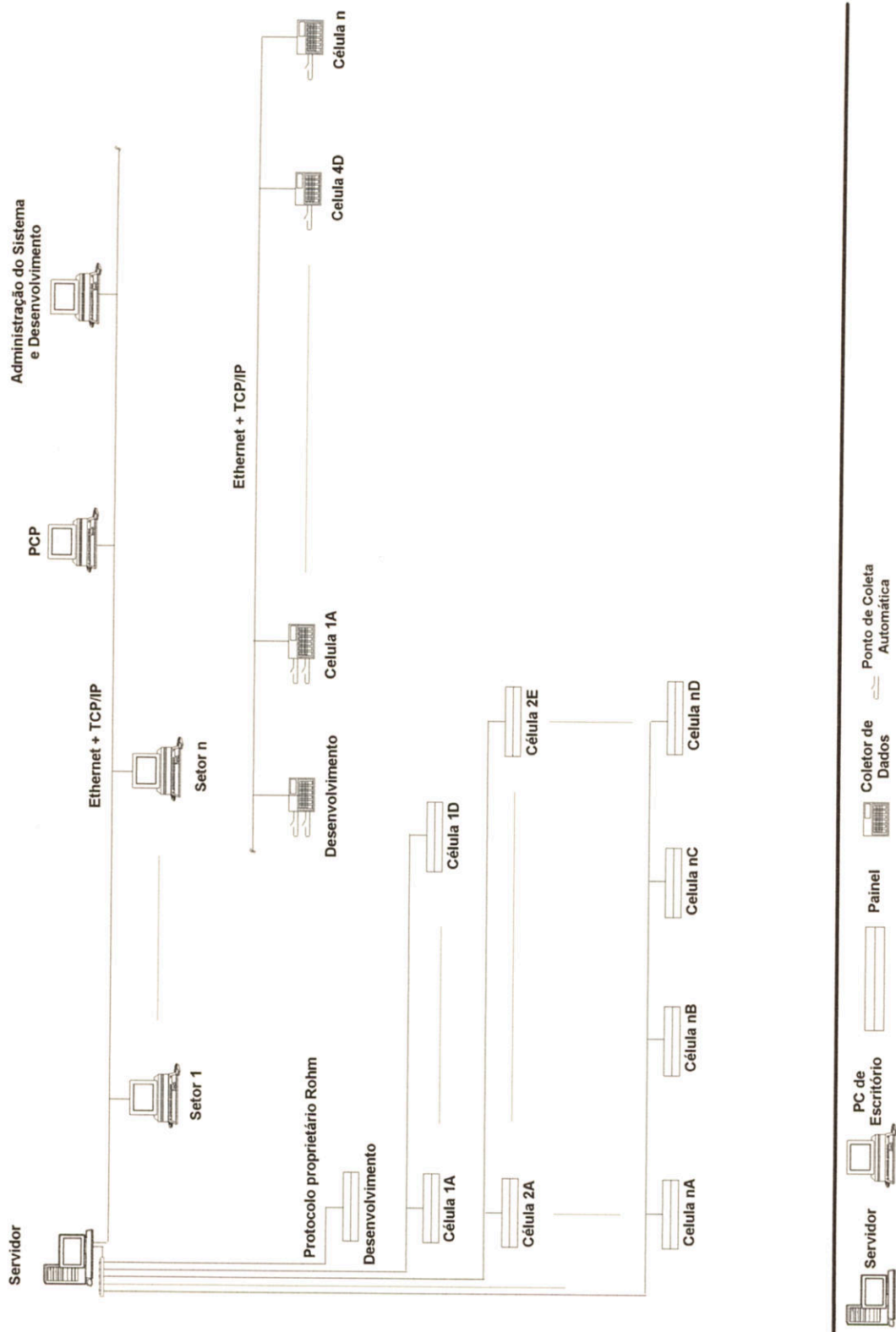


figura 4.2 - Esquema simplificado do hardware do SII

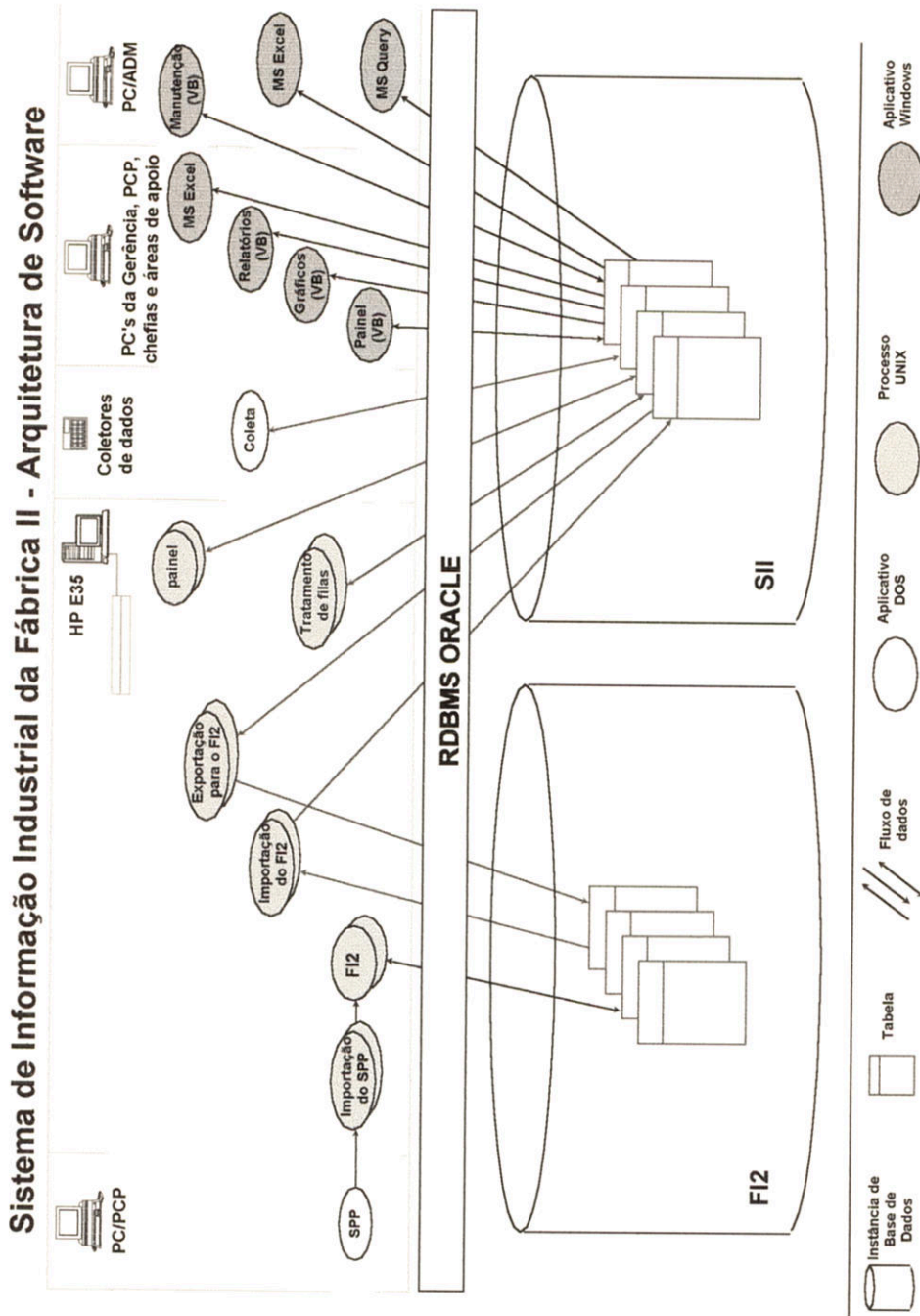


figura 4.3 - Esquema simplificado de software do SII

Capítulo V - *O sistema GDT*

O **GDT** (Gerenciador de Documentação Técnica) ou, em inglês, TDM (Total Data Management), é um administrador de informações que dispõe de aplicações e ferramentas que suportam várias plataformas e inúmeros serviços de redes e protocolos. É um produto desenvolvido pela FORMTEK Inc., capaz de administrar diferentes tipos de documentos (desenhos em CAD, fontes de programas de CLP's, textos em geral, normas técnicas, etc.).

Este software é modular e composto de quatro elementos funcionais:

- gerenciamento de dados (documentos)
- automação de serviços
- gerenciamento de estruturas de produto
- gerenciamento de configuração do produto

Na maioria das empresas que criam e fabricam produtos, as informações necessárias para projetar, analisar, produzir, montar, testar, validar, e dar suporte, estão espalhadas por toda a empresa, em ambientes diferentes e envolvendo diferentes tipos de dados. A arquitetura do **GDT** tem a capacidade de armazenar e gerenciar todos os tipos de dados, de forma transparente para o usuário.

Ao invés de gerenciar arquivos, o **GDT** gerencia informação, e entrega esta informação ao usuário no ambiente em que ele estiver trabalhando. O **GDT** possui uma heterogeneidade de hardware e software, ou seja, ele é capaz de disponibilizar o documento ao usuário no software em que o usuário deseja, desde que o documento seja suportado por este software.

O **GDT** consegue representar o mesmo objeto (documento) em diferentes tipos de dados. Por exemplo, um desenho pode ser representado como um objeto complexo, que não pode ser executado ou editado em qualquer plataforma, ou como um tipo de dado simples, que serve apenas para visualização e distribuição.

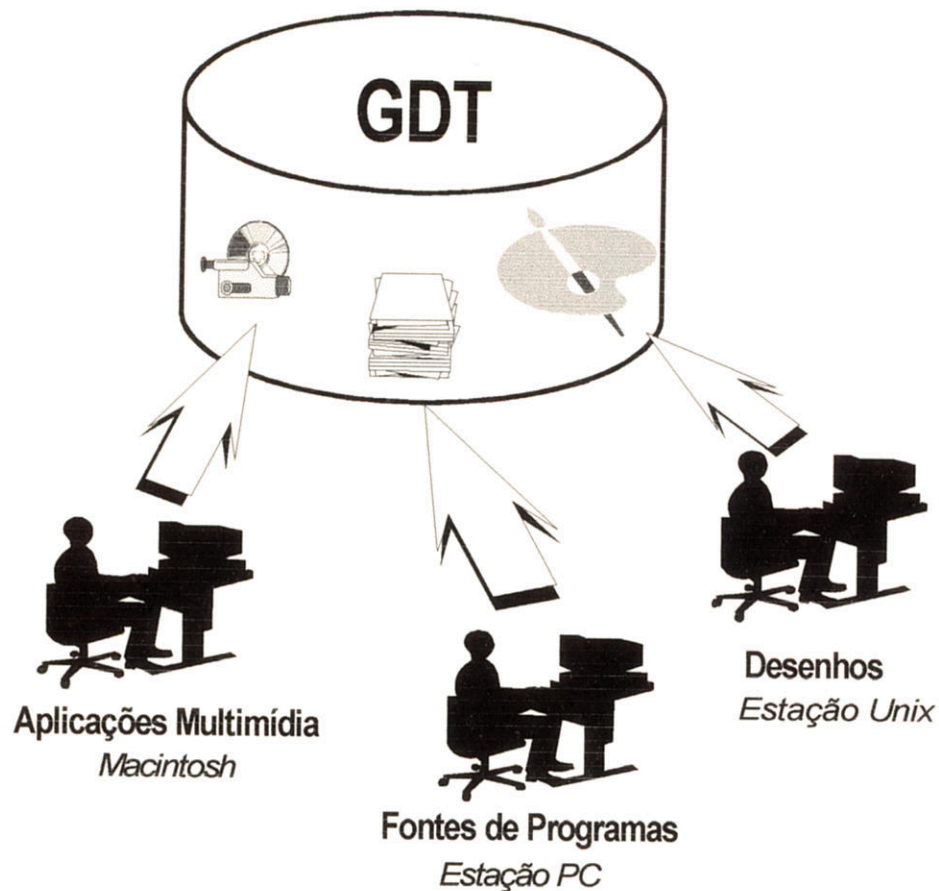


figura 5.1 - Aplicações utilizando GDT

Cada objeto contém um arquivo fonte, representações em formatos mais comuns, e *links* com os aplicativos necessários para criar ou visualizar cada uma destas representações. Todas estas informações encontram-se "encapsuladas" no objeto. Quando o usuário procura abrir um determinado documento, o **GDT** sabe qual o tipo de dado a ser aberto e qual a aplicação carregar, baseado na plataforma em que o usuário se encontra, e a função que será realizada (edição, visualização, etc). Estes *links* entre aplicações e documentos são configurados pelo administrador do sistema. Cada documento pode ser armazenado num número ilimitado de formatos, para qualquer número de aplicações.

A arquitetura do **GDT** é do tipo cliente-servidor, suporta várias plataformas e possui uma interface bastante completa e amigável para com o usuário. O software do **GDT** que roda no cliente é disponível em Windows, Macintosh System 7, Sun

Solaris, IBM RS/6000 AIX, HP/UX, e estações Silicon Graphics IRIS. O servidor roda em Sun Solaris, IBM RS/6000 AIX, HP/UX, e estações Silicon Graphics IRIS. Suporta vários sistemas de base de dados, inclusive Oracle 7 (utilizado pela **EMBRACO**).

A procura por documentos é realizada através de formulários. O **GDT** vem com diversos tipos de formulários padrão, que podem ser *customizados*, ou seja, alterados graficamente para cada tipo de aplicação. As consultas podem ser feitas através de "procuras-por-exemplo", ou seja, com alguns comandos simples denominados *wildcards* (*%*, *-*) pode-se procurar, por exemplo, todos as normas brasileiras cadastradas, utilizando a linha de comando *NBR%*. Isto não só previne erros de digitação, mas facilita o processo de busca de dados para novos usuários.

Feita a consulta, é retornada uma lista de itens com vários tipos de objetos. Depois de escolher um destes itens e uma ação a ser tomada (visualizar, imprimir, exportar, etc), a aplicação apropriada é carregada diretamente pelo **GDT**.

Outra característica essencial é a habilidade de tratar de forma diferente, objetos de diferentes origens. Por exemplo, documentos de um determinado departamento podem ser identificados usando uma estrutura de nomes diferente da estrutura utilizada em um outro departamento. Também as informações sobre determinado objeto podem ser diferenciadas (exemplificado na *figura 5.4*) através de diferentes índices. As classes de objetos são definidas pelo administrador do sistema. Quando um objeto é cadastrado no sistema, ele passa a fazer parte de uma classe. As convenções de nomes, índices e formulários para aquela classe são automaticamente herdadas pelo objeto.

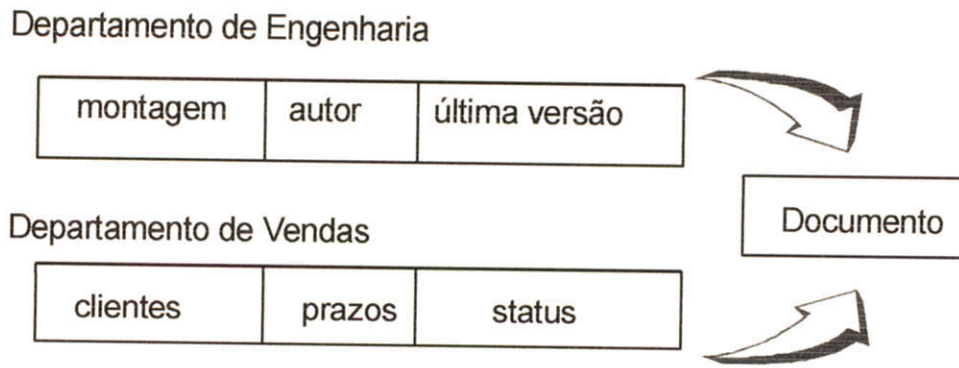


figura 5.4 - Exemplo que mostra o mesmo documento sob o ponto de vista de dois departamentos diferentes de uma empresa

Pode ser criada uma estrutura hierárquica de objetos. Isto permite organizar de forma mais racional toda a documentação existente. Por exemplo, ao criarmos uma classe de objetos "Linha de Montagem de Compressores", podemos criar uma estrutura da seguinte forma:

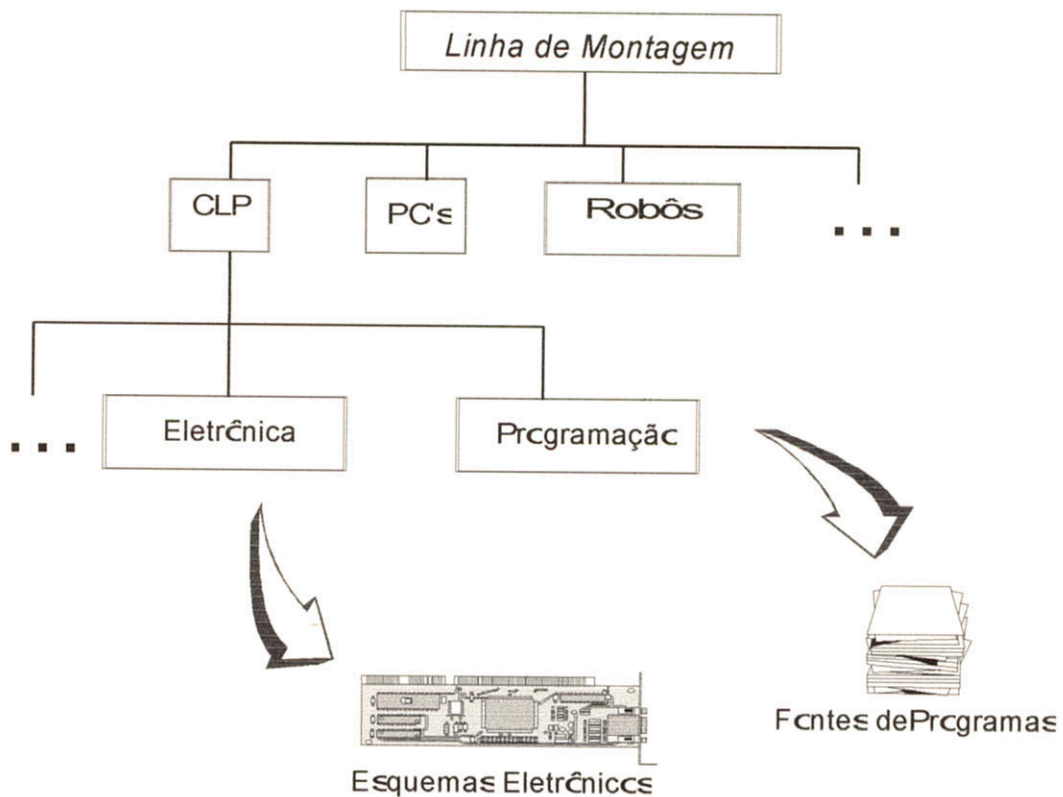


figura 5.5 - Exemplo de estrutura hierárquica de objetos

O **GDT** realiza um controle de versões de documentos. Isto permite estabelecer um histórico do que já foi feito e das etapas que levaram até a última versão. Quando se deseja atualizar um documento, um parâmetro é anexado ao documento, dizendo que ele está momentaneamente "separado" para atualização de versão. Qualquer usuário poderá visualizar o documento, mas não editá-lo. Quando a atualização estiver terminada, a nova versão é disponibilizada para os usuários.

O **GDT** possibilita total controle do acesso aos documentos. Diferentes direitos podem ser atribuídos a cada usuário, e podem ser criados grupos, os quais possuem os mesmos direitos de acesso. Além de restrições ao acesso de documentos, pode-se definir quais comandos serão disponibilizados para cada usuário ou grupo de usuários.

A implantação do **GDT** na **EMBRACO** está ainda em andamento e os usuários ainda encontram-se em fase de treinamento. O objetivo é que ele seja utilizado em todos os níveis da empresa.

Outras informações sobre este software encontram-se na referência [10].

Capítulo VI - *Levantamento das Necessidades e Avaliação das Possibilidades*

Antes de efetivamente especificar uma possível integração dos sistemas citados anteriormente (**SII**, **MAXIMO** e **GDT**), é necessário que se saiba claramente, quais os objetivos a serem alcançados com esta integração, as vantagens que esta integração irá proporcionar, e se é viável a sua implementação. Ou seja, num primeiro momento, deve ser realizado um *levantamento das necessidades* que motivam a idéia de integração destes sistemas e, em seguida, deve-se realizar uma *avaliação* de todas estas necessidades, a fim de verificar a viabilidade delas, e até mesmo sugerir outras possibilidades ou funcionalidades que poderiam surgir a partir desta integração.

6.1 - Levantamento das Necessidades

Para o levantamento das necessidades de integração, foi necessário reunir as pessoas diretamente ligadas aos sistemas envolvidos. Estas pessoas são as seguintes:

- Manuel Braga: Engenheiro especialista da *Fábrica II* e responsável pela implantação do Sistema de Informação Industrial (**SII**);
- Paulo Henrique: Engenheiro, coordenador e planejador da Manutenção na área de Suporte. Responsável pela implantação do **MAXIMO** na *Fábrica II*;
- Naudir R. de Alcântara: Planejador da Manutenção (*Fábrica II*) na área de Usinagem. Usuário do sistema **MAXIMO**;
- Marcelo Pires: Responsável pelo Planejamento e Controle da Produção (PCP) da *Fábrica II*. Usuário do Sistema de Informação Industrial (**SII**);

- Aguinaldo R. de Campos: Especialista do CIM (setor da **EMBRACO**), responsável pela implantação do **GDT**.

Com cada uma destas pessoas foi realizado um "brainstorming" sobre as necessidades, visando a obtenção das informações necessárias para a correta especificação da integração. Nestas reuniões, procurou-se saber a respeito de quais informações devem ser trocadas entre os sistemas, segundo a expectativa dos usuários dos sistemas. Hoje, para que a troca de informações aconteça, são necessárias reuniões entre estas pessoas, ou o envio de papéis de um lado para o outro, o que torna o trabalho demorado e, muitas vezes, impraticável. Foram discutidas também outras funcionalidades que poderiam tornar o ambiente de trabalho melhor e mais ágil.

Abaixo, são apresentados os dados obtidos a partir do levantamento das necessidades.

6.1.1 - Integração MAXIMO & SII

"Que tipo de informações o planejador do PCP gostaria de saber a respeito da Manutenção, para que haja uma efetiva integração entre as partes?"

A *Fábrica II* foi projetada para trabalhar com estoque zero. Desta forma, a Manutenção e o PCP devem estar em profunda sincronia, visto que, nos serviços de manutenção mais pesados, que exigem parada de equipamento, a produção pode ser prejudicada. Hoje, a troca de informações entre PCP e Manutenção dificilmente acontece. É comum a hora extra para compensar a produção comprometida por parada de máquina. Isto acontece principalmente nos finais de semana, quando a maior parte das manutenções acontece. Neste sentido, seria interessante, por parte do PCP, conhecer a programação da manutenção com antecedência. Isto propiciaria ao responsável pelo PCP, alocar a produção de forma tal a não coincidir com os horários de manutenção. Esta visualização poderia ser feita no **FI2**, já que a programação da produção é feita com auxílio deste software (ver *figura 6.1*).

Desta forma, a programação das manutenções em cada linha apareceria em forma de gráfico de Gantt, sendo facilmente visualizada pelo planejador do PCP (ver figura abaixo). Mais precisamente, as informações desejadas pelo PCP são: dia, hora, tempo de parada, e as Células em que ocorrerão as manutenções com parada de máquina.

ÍTEM	Semana	ANO DE 1996									
		01	02	03	04	05	06	07	08	09	10
BLOCO1											
PISTÃO											
CORPO											
TAMPA											
MONTAGEM											

figura 6.1 - Desenho esquemático da tela do FI2

"Que funções do MAXIMO poderiam ser disponibilizadas no coletor de dados?"

O **MAXIMO** atualmente encontra-se instalado apenas no suporte da manutenção e nas áreas de suporte das próprias UGB's. Surge porém, a necessidade de que o **MAXIMO** seja levado para o chão-de-fábrica, ou seja, para perto do operador do equipamento. Isto traz grandes vantagens, no sentido em que não é necessário, por exemplo, o operador sair do seu posto de trabalho para relatar ao técnico um mau funcionamento do equipamento. Outra vantagem, é que o técnico, ao terminar o serviço pode, ali mesmo junto ao equipamento, reportar o serviço realizado.

Isto poderia ser feito com a utilização dos coletores de dados que, como visto, são em grande número, e espalhados por toda a fábrica. Poderiam ser criados, na tela do coletor, menus que propiciariam ao operador e ao técnico, algumas funções do **MAXIMO**. Estas funções seriam as seguintes, em ordem de prioridade de implementação:

- **Abrir Requisição de Serviços** - realizado pelo operador ou técnico;
- **Abrir Ordem de Serviço** - realizado pelo operador ou técnico;

- **Reporte Rápido de Ordem de Serviço** - realizado pelo técnico;
- **Fechar Ordem de Serviço** - realizado por um técnico;
- **Verificar histórico de ordens** - realizado por operador e por técnico;
- **Chamar um técnico via coletor** - a chamada poderia ser feita via painel.

6.1.2 - Integração MAXIMO & GDT

O **GDT**, como visto anteriormente, gerencia qualquer tipo de informação. Para a manutenção, os documentos necessários são principalmente desenhos de equipamentos e esquemas que mostram como deve ser feita a manutenção em determinada máquina. A partir do **MAXIMO**, pode-se disparar aplicações como CAD, por exemplo, aonde estes desenhos poderiam ser visualizados e/ou editados. Porém, como a **EMBRACO** possui o **GDT**, que é um software poderoso e que será amplamente difundido na empresa, surgiu a necessidade de integrá-lo com o **MAXIMO**. Esta integração se daria da seguinte forma: dentro do **MAXIMO**, existiriam botões ou um menu de opções associados a um ou mais desenhos esquemáticos. Ao acionar um destes botões, o desenho seria trazido à tela, mas todo o controle de busca do arquivo, como também o acionamento do software gráfico seria feito pelo **GDT**, de forma transparente para o usuário (ver *figura 6.2*). Isto adicionaria vantagens aos usuários do **MAXIMO**, tais como, controle de acesso dos usuários aos documentos, controle de versões, etc.

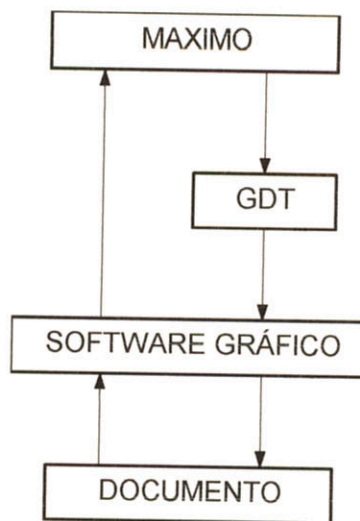


figura 6.2 Arquitetura *desejada* com a integração entre *MAXIMO* e *GDT*

6.2 - Avaliação das Possibilidades

Estas necessidades, tornam-se, nesta fase de avaliação, objetivos a serem alcançados com a integração dos sistemas. Elas definem também, o grau de integração desejado.

Resta saber, se todas estas necessidades são realmente viáveis de serem implementadas com a integração. A avaliação destas necessidades é também a avaliação da viabilidade da integração como um todo, já que, se nenhuma destas necessidades forem viáveis, a própria integração não poderá ser implementada.

Os estudos feitos para a avaliação da viabilidade desta integração serão descritos a seguir.

6.2.1 - Integração MAXIMO & SII

A configuração atual dos dois sistemas encontra-se da seguinte forma:

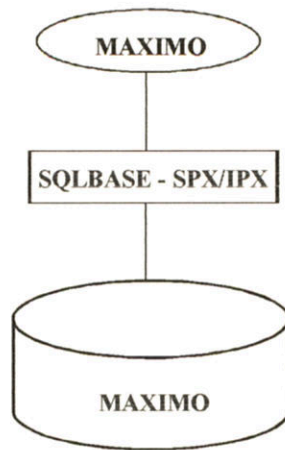


figura 6.3 - Configuração atual do sistema MAXIMO

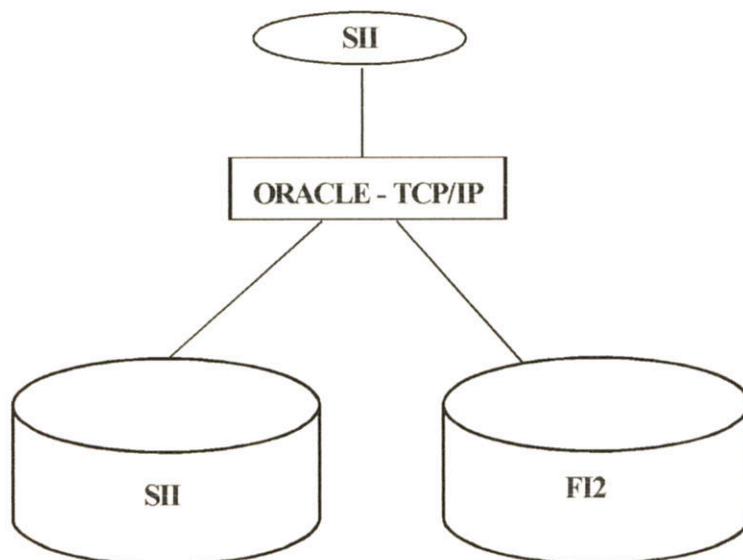


figura 6.4 - Configuração atual do sistema SII

O **MAXIMO** tem sua base de dados em SQLBASE num servidor NetWare¹ que é acessado via protocolo SPX/IPX. Já o Sistema de Informação Industrial (**SII**) tem sua base de dados em ORACLE num servidor UNIX que é acessado via TCP/IP. O **SII** possui duas instâncias de base de dados: uma para a coleta de dados do chão de fábrica (**SII**) e outra para os dados do **FI2**.

¹ NetWare é o nome associado aos vários componentes da rede NOVELL [6]. Corresponde aos drivers de *link* entre dados, software cliente, e o sistema operacional do servidor NetWare. Roda em Token Ring, Ethernet e ARCNET.

Observando este total isolamento entre os dois sistemas, o que dificultaria ou mesmo inviabilizaria qualquer intenção de integrá-los, surge a necessidade de unificação de suas plataformas (base de dados, rede e protocolo de comunicação), devido aos seguintes fatores:

- Padronização: Toda e qualquer padronização é útil ao bom funcionamento de qualquer organização. Neste sentido, a **EMBRACO** tem incentivado a padronização em todas as suas áreas, e isso inclui também o setor de informática. Dentro da **EMBRACO** existem grupos de estudo e padronização de softwares.
- Custos de Manutenção e Treinamento: O custo para se manter duas bases de dados diferentes é muito grande. Duas bases de dados significam duas empresas de consultoria diferentes prestando serviços. Outra questão é com relação ao treinamento, visto que é necessário existir dentro da **EMBRACO**, pessoal de suporte com bom conhecimento técnico nos dois tipos de banco de dados. Isto também envolve um custo bastante elevado.
- Técnico: o software que comanda o coletor de dados não consegue "enxergar" a base SQLBASE, devido a forma como está implementado.

A escolha será, provavelmente, a de migrar o software **MAXIMO** de SQLBASE para ORACLE. Esta migração consiste em comprar uma versão do **MAXIMO** que rode sobre ORACLE, e transferir os dados atuais para este novo sistema de gerenciamento de base de dados. Esta solução foi escolhida devido aos seguintes fatores:

- O ORACLE foi homologado por um grupo de trabalho da Informática. Esta homologação é válida a nível de grupo BRASMOTOR;
- O banco de dados ORACLE é mais robusto e bastante conhecido. Haverá a possibilidade de se obter suporte do próprio pessoal da **EMBRACO**, não

necessitando auxílio externo (diminuição de custos de manutenção).

- A base de dados SQLBASE não demonstrou ser confiável no *upgrade* de versões feito com o software **MAXIMO**. Uma série de inconsistências² nas tabelas da base de dados apareceu no momento do *upgrade*. Inconsistências estas que deveriam ter sido detectadas pelo banco de dados muito antes, ou seja, na hora em que as alterações indevidas foram realizadas.

A configuração do sistema após a integração será a seguinte:

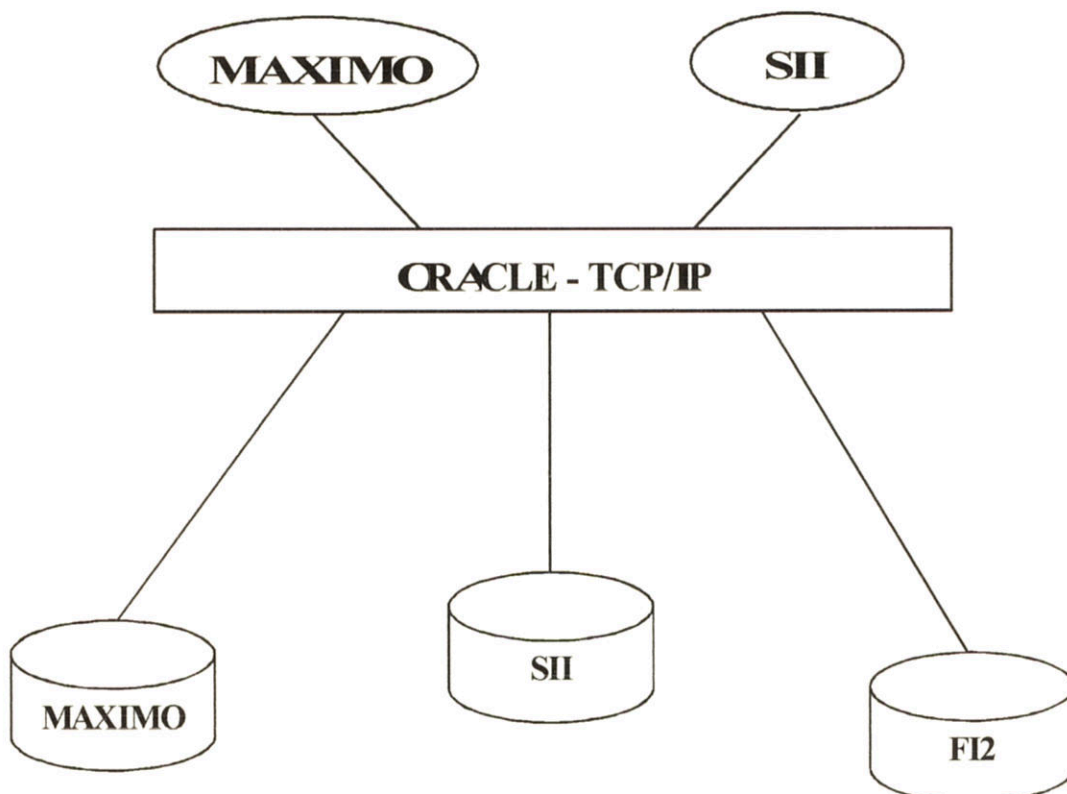


figura 6.4 - Configuração prevista para integração

² Uma inconsistência significa que uma informação foi armazenada num formato diferente do utilizado para a mesma informação em outras tabelas [1].

Esta nova configuração possui, agora, três instâncias de base de dados: **MAXIMO**, **SII** e **FI2**. Estas três instâncias estão sob um mesmo sistema de gerenciamento de base de dados (ORACLE) que é acessado via protocolo TCP/IP.

Com esta configuração, a troca de informações entre os sistemas fica mais fácil e mais rápida, visto que os seus dados estão em um mesmo banco de dados (apesar de estarem em instâncias diferentes), num mesmo servidor, e o protocolo de comunicação é único.

Desta forma, resolve-se o problema do acesso ao **MAXIMO** via coletor de dados, pois assim o software que roda no coletor consegue buscar ou armazenar dados também na base do **MAXIMO**. Isto viabiliza a inclusão de algumas funções de um cliente do **MAXIMO** no coletor de dados.

Porém, segundo estudos feitos nas tabelas da base de dados do **MAXIMO**, nem todas as funções desejadas poderão ser incluídas, pois existem restrições técnicas. As funções de Abrir Ordens de Serviço, Reporte Rápido de Ordem de Serviço e Fechar Ordem de Serviço tornam-se inviáveis de implementar no momento, pois elas atualizam várias tabelas, e estas atualizações não são totalmente conhecidas. A implementação destas funções pode ocasionar inconsistências na base de dados. É necessário, então, realizar um estudo profundo deste caso, e conhecer todas as tabelas envolvidas em cada uma destas ações. Outro fato, é que se desejarmos implementar um número muito grande de funções deste tipo, estaremos quase que reescrevendo o software cliente do **MAXIMO** para o coletor.

Já as funções de Abrir Requisição de Serviços e Verificar Histórico de Ordens são viáveis, pois envolvem apenas uma tabela (WO - "Work Order"). Na verdade, a segunda função acima apenas realiza uma consulta na base de dados, diferentemente da primeira que realiza uma atualização numa tabela da base.

A função que prevê Chamar um Técnico Via Coletor fará parte da integração, porém é totalmente viável, pois não envolve tabela alguma de nenhuma base de dados. Esta função pode ser feita manualmente, acionando alguma função na tela do coletor, ou automaticamente, quando é feita uma requisição de serviços.

Com relação à troca de informações entre **MAXIMO** e **FI2**, esta é totalmente viável, visto que ambos os sistemas possuem mecanismos para tal. Tais mecanismos serão melhor detalhados no próximo capítulo.

6.2.2 - Integração MAXIMO & GDT

A integração entre **MAXIMO** e **GDT**, da forma como foi desejada, não é possível de ser implementada. Isto acontece devido à restrições no software **GDT**.

O **GDT** é um software cuja arquitetura é fechada. Não se conhece nada a respeito de suas tabelas e sua forma de organizar dos dados. O que se sabe, é que, ao inserir um novo documento, o **GDT** grava este objeto em um caminho desconhecido, ou seja, em um diretório que somente o **GDT** conhece.

Sendo assim, nenhum outro aplicativo consegue acessar estes documentos, senão através do próprio **GDT**. O procedimento usado pelo **GDT** é o seguinte: quando o usuário tenta visualizar um documento, o **GDT** cria um diretório temporário e faz uma cópia do documento verdadeiro para dentro deste diretório. Quando terminada a visualização, o diretório é apagado automaticamente. Já se o usuário deseja editar um documento (se possuir direitos para tal), o **GDT** cria também um diretório temporário, e transfere o documento do diretório no qual ele se encontra (desconhecido) para este diretório temporário. Terminada a edição, o documento é novamente transferido para o seu diretório de origem, e o diretório temporário é apagado [10].

Outro problema é que o **GDT**, na versão atual, não possui uma linha de comando que permita a chamada de arquivos juntamente com o disparo do próprio **GDT**. Quase todos os outros aplicativos em WINDOWS possuem este tipo de mecanismo. Como exemplo, a linha de comando:

```
WORD.EXE ARQUIVO.DOC
```

acionaria o software WORD já abrindo o documento arquivo.doc.

Logo, a partir do **MAXIMO** não se conseguiria, através de botões ou menu de opções, chamar diretamente um documento da forma como foi desejado anteriormente. Atualmente, a única forma de abrir um documento que está cadastrado no **GDT** é, primeiramente acionar o **GDT** (a partir de um botão), e, estando dentro do **GDT**, procurar e trazer à tela o documento desejado. Esta fica sendo a solução viável até o presente momento.

É sabido que, provavelmente, a próxima versão do **GDT** virá com a possibilidade de incluir o documento desejado já na linha de comando. Se isso realmente acontecer, esta integração poderá ser implementada.

Assim ficam conhecidas quais as possibilidades viáveis de serem implementadas na integração dos sistemas.

No próximo capítulo será mostrado como esta integração pode ser implementada.

Capítulo VII - *Especificação da Integração*

Como visto anteriormente, das duas integrações propostas (entre **MAXIMO** e **SII**, e entre **MAXIMO** e **GDT**), a única viável de ser implementada no momento é a integração entre o **MAXIMO** e o Sistema de Informação Industrial (**SII**), já que a segunda apresenta algumas restrições técnicas (mostradas no capítulo anterior) que inviabilizam sua implementação.

Portanto, neste capítulo, será apresentada a especificação da intergração entre **MAXIMO** e **SII**. Esta integração, por sua vez, será dividida em duas partes: a primeira, que busca disponibilizar funções do **MAXIMO** na tela do coletor de dados, e a segunda, que visa integrar o software **FI2** com o **MAXIMO**.

7.1 - Integração entre **MAXIMO** e Coletor de Dados

A idéia aqui, como colocado anteriormente, é disponibilizar ao operador de máquina e ao técnico de manutenção, um meio pelo qual se possa utilizar algumas funções simples do **MAXIMO**. Estas funções se concentram basicamente em: requisitar serviços de manutenção e realizar consultas simples na base do **MAXIMO**.

Estas funções serão disponibilizadas no coletor, pois existem vários destes aparelhos espalhados pelo chão-de-fábrica, e também porque atualmente não são utilizados em tempo integral. A inclusão destas funções no coletor trará também o benefício de uma maior utilização deste, agregando mais valor a estes aparelhos.

Atualmente, o coletor trabalha da seguinte forma: No coletor, roda um aplicativo em DOS. Todos os menus que aparecem na tela, a contagem automática de dados, e a comunicação com o servidor UNIX via rede, tudo isto é comandado por este aplicativo.

Existem dois tipos de coleta: coleta manual e coleta automática. A primeira é feita pelo usuário através dos teclados e diz respeito aos apontamentos de *setup*, início e fim de produção, etc. Os dados destes apontamentos são enviados, via TCP/IP ao

servidor e atualizados em tabelas da base do **SII**. A coleta automática é realizada por sensores que, ao detectar a passagem de peças, enviam um sinal para o coletor de dados. O aplicativo do coletor faz uma varredura sobre as entradas digitais do coletor e insere um registro numa tabela do ORACLE com o apontamento de quantidade, quando a passagem de peças é detectada.

No servidor existem processos UNIX em execução que lêem as tabelas atualizadas pelo coletor e alteram as demais tabelas da base de dados. Tudo isto pode ser verificado na figura abaixo.

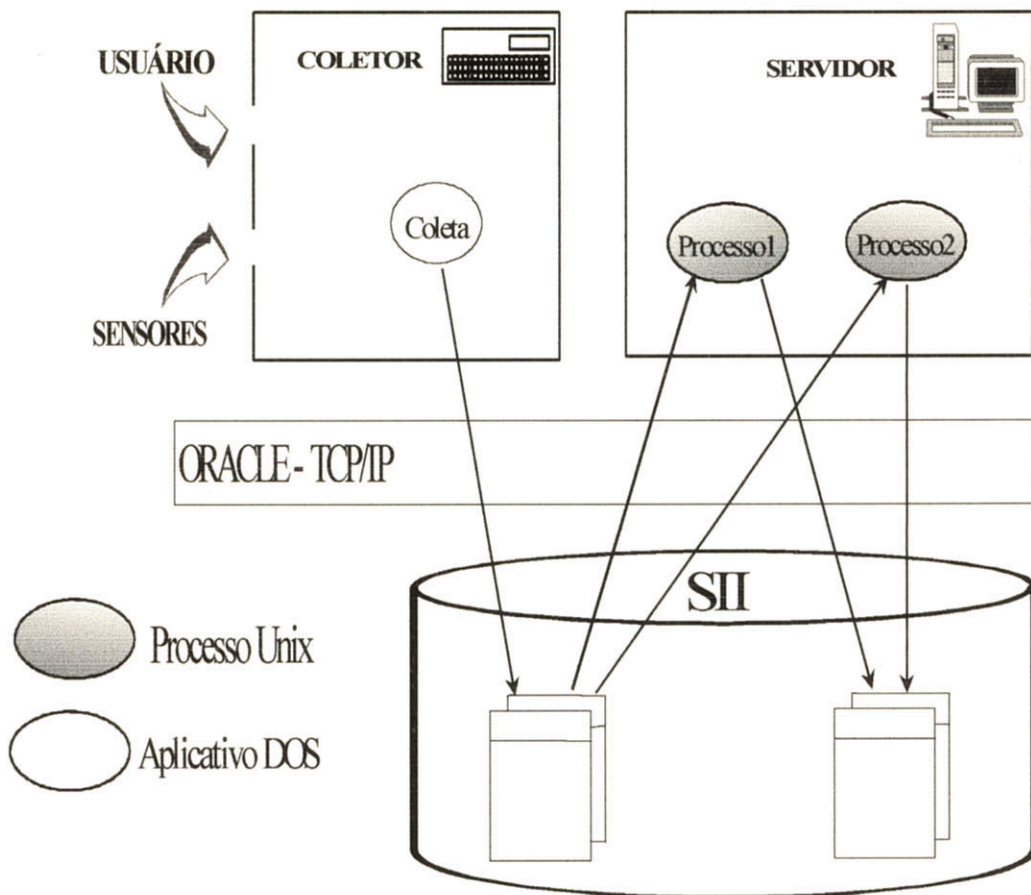


figura 7.1 - Configuração atual da coleta de dados

A integração proposta pretende utilizar a mesma arquitetura acima, a fim de manter o padrão atual já definido para o sistema.

Sendo assim, a integração se dará da seguinte forma: no coletor de dados existirão novas funções que disponibilizarão na

tela, novas opções de menu de entrada. Nestas opções o operador poderá selecionar, a partir de uma lista, o equipamento com problema, e requisitar um serviço ao técnico.

Primeiramente o aplicativo no coletor fará uma leitura na tabela WORKORDER (MAXIMO) a fim de saber o último número sequencial de ordem de serviço registrado. Em seguida, será realizada a inserção de uma nova ordem de serviço nesta mesma tabela. Esta ordem de serviço registrada possui poucas informações - apenas o número do equipamento, a data atual, o nome do operador, o centro de custo, e o número da ordem de serviço. As demais informações (código de falhas, reporte de mão-de-obra, peças etc.) devem ser registradas pelo técnico após a realização dos serviços. Isto deve ser feito nos PC's das salas de suporte das UGB's ou nos PC's da própria Manutenção.

Uma outra idéia, surgida a partir do levantamento das necessidades (capítulo VI), propõe que apareça uma mensagem de alarme na tela do painel da linha onde se encontra o operador. O técnico mais próximo atenderá a chamada.

As novas opções de menu permitirão realizar também consultas simples à base do **MAXIMO**. Um operador ou mesmo um técnico, pode, por exemplo, desejar saber se já foi feito um serviço de manutenção, ou qual a hora do próximo serviço naquele equipamento. Pode-se também incluir consultas de histórico de máquinas. Quando um técnico estiver resolvendo um problema, ele pode querer saber quais os últimos problemas acontecidos naquela máquina.

Todas estas consultas são bastante simples de serem efetuadas, e obviamente, não significam risco algum à integridade da base de dados. Pode-se, inclusive, criar novas *views*¹, as quais possuirão dados para algumas consultas especiais, como as citadas acima.

Estas funções adicionais no coletor de dados, serão escritas em "ProC". O pré-compilador "ProC" da ORACLE permite a

¹Uma *view* assemelha-se a uma tabela (e pode-se pensar como tal), e é formada a partir de colunas de outras tabelas da base de dados. Uma *view* é, implicitamente, uma linha de comandos de consulta em SQL. Quando se consulta uma *view* (da mesma forma como se consulta qualquer tabela), executa-se implicitamente estes comandos, e o resultado é apresentado na forma de uma tabela. Uma *view* não contém dados, apenas recupera-os das tabelas as quais está associada [3],[6].

inclusão de comandos em SQL no código do programa [6]. Ou seja, escreve-se o programa em "C" , e nele inclui-se alguns comandos em SQL. Depois este programa é pré-compilado, e em seguida compilado e "linkado" normalmente, como qualquer fonte em "C".

A figura abaixo representa esquematicamente esta integração, supondo que as novas funções foram colocadas em dois módulos separados, dentro do aplicativo Coleta.

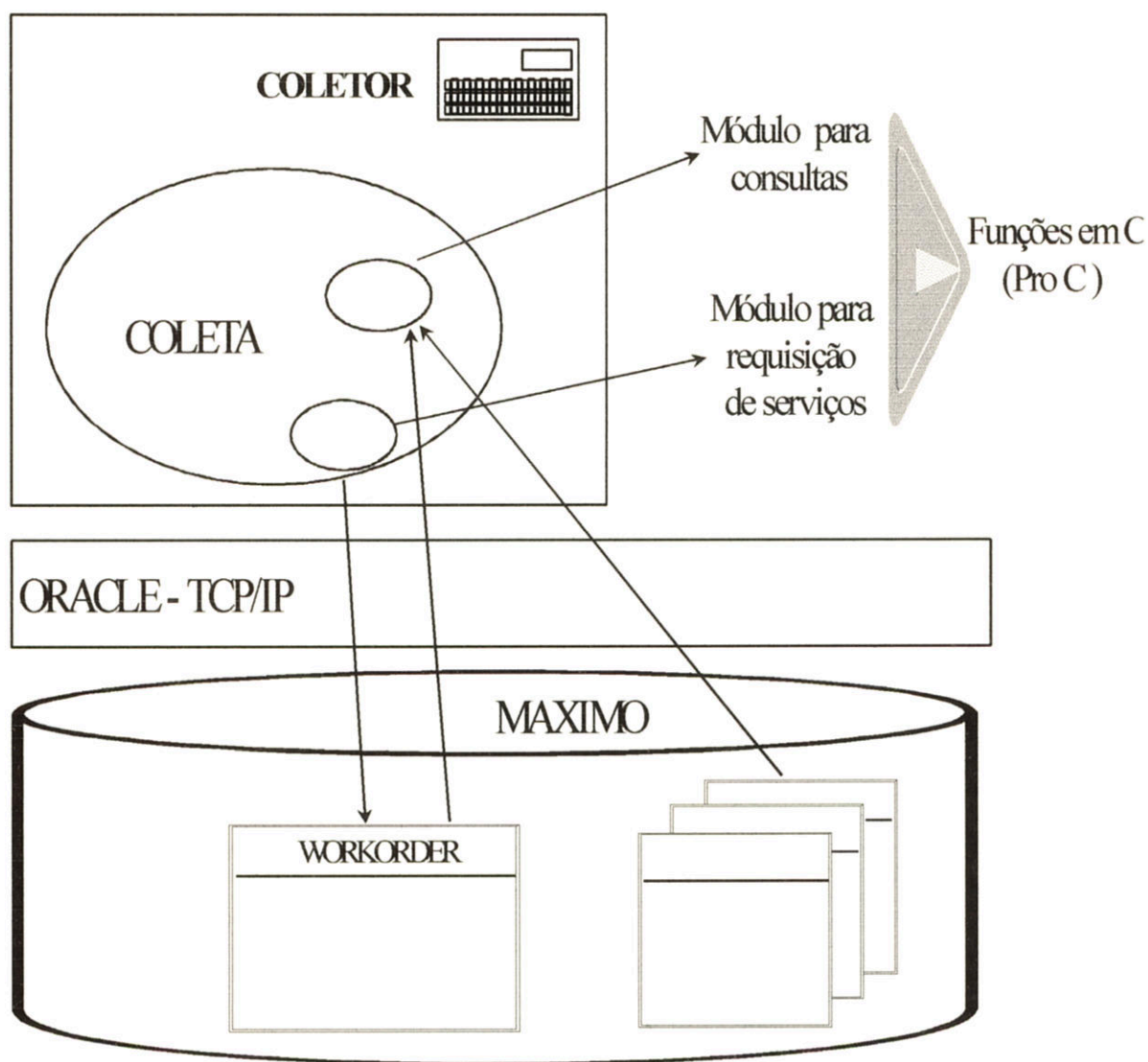


figura 7.2 - Especificação da Integração entre MAXIMO e Coletor de Dados

A função que chama um técnico através do painel não será ilustrada aqui, por não fazer parte diretamente da integração

entre **MAXIMO** e **SII**, pois significa apenas uma função a mais nos aplicativos do coletor.

7.2 - Integração entre **MAXIMO** e **FI2**

Antes de detalhar a especificação desta integração, serão discutidos alguns aspectos relacionados com o software **FI2**. Isto é necessário para o bom entendimento da especificação feita.

7.2.1 - Interface Padrão do **FI2** [11]

O software **FI2** possui uma interface padrão para a troca de dados com outros sistemas. Esta interface se constitui de arquivos para troca de dados, com formatos pré-definidos, e de programas específicos do **FI2** que efetuam a leitura e a escrita destes arquivos.

Três tipos de sistemas podem trocar dados com o **FI2**: sistemas de coleta de dados, sistemas de planejamento da produção (MRPII - Manufacturing Resources Planning) e sistemas de gerenciamento de recursos. Para cada um destes existem programas específicos que realizam a exportação e a importação de dados para o **FI2**.

Cada um destes programas forma um caminho de transmissão de dados. Os seis programas são citados abaixo, com a descrição de suas funções:

- vonpps: recebe os dados do MRPII e os envia para a base do **FI2**;
- anpps: transfere os dados do **FI2** para o MRPII;
- vonbde: transfere dados da coleta para o **FI2**;
- anbde: transfere dados do **FI2** para a coleta;
- vonrss: transfere dados do sistema de gerenciamento de recursos para o **FI2**;
- anrss: transfere dados do **FI2** para o sistema de gerenciamento de recursos.

Uma melhor visualização pode ser feita através da *figura 7.3*, onde são mostrados os três tipos de sistemas com os quais a troca de dados é possível, bem como os programas que interagem com cada um deles.

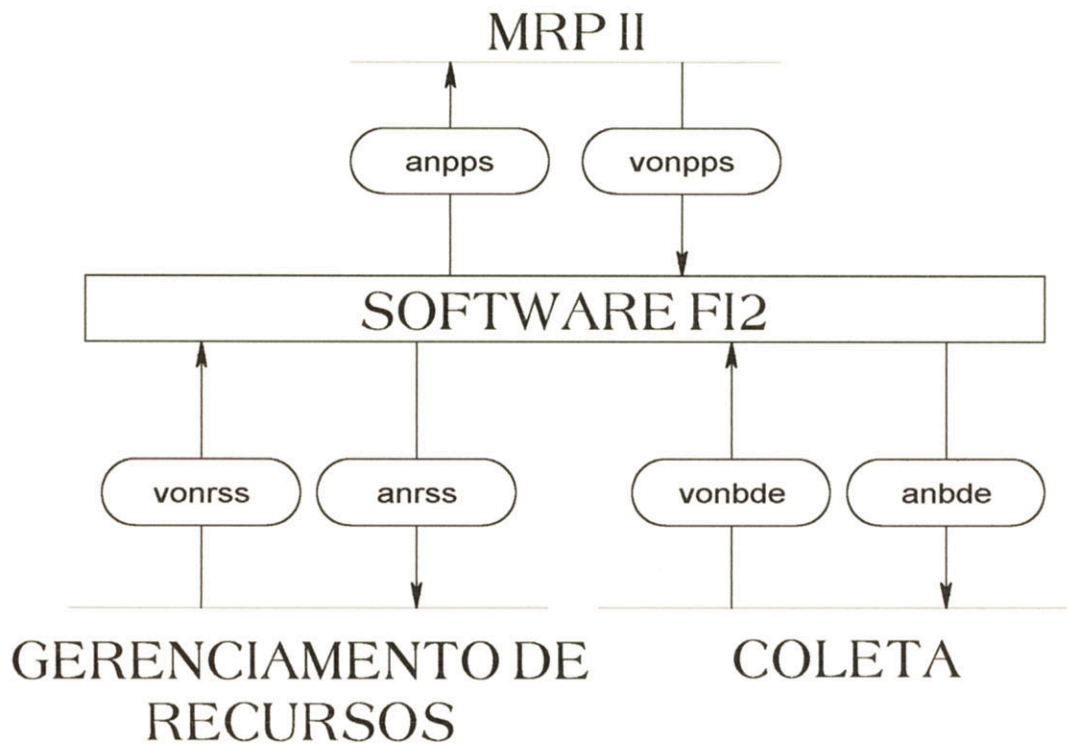


figura 7.3 - Interface padrão do FI2

Estes programas escrevem ou lêem dados de arquivos que possuem uma formatação padrão, definida pelo **FI2**. Os programas que exportam dados do **FI2**, escrevem dados nestes arquivos, enquanto que os programas que importam dados para o **FI2**, lêem os dados contidos nestes arquivos.

Para exemplificar, a *figura 7.4* ilustra a troca de informações entre o **FI2** e um sistema de coleta de dados qualquer: para importar dados para o **FI2**, o processo "importar" busca os dados na base da coleta, e os escreve no arquivo "arqimp", utilizando o formato padrão exigido pelo **FI2**. O programa "vonbde" lê este arquivo e envia os dados para a base do **FI2**.

Já para se exportar dados do **FI2**, o programa "anbde" busca os dados na base do **FI2** e os escreve no arquivo "arqexp", também utilizando o padrão do **FI2**. O processo "exportar" lê este arquivo e atualiza a base da coleta.

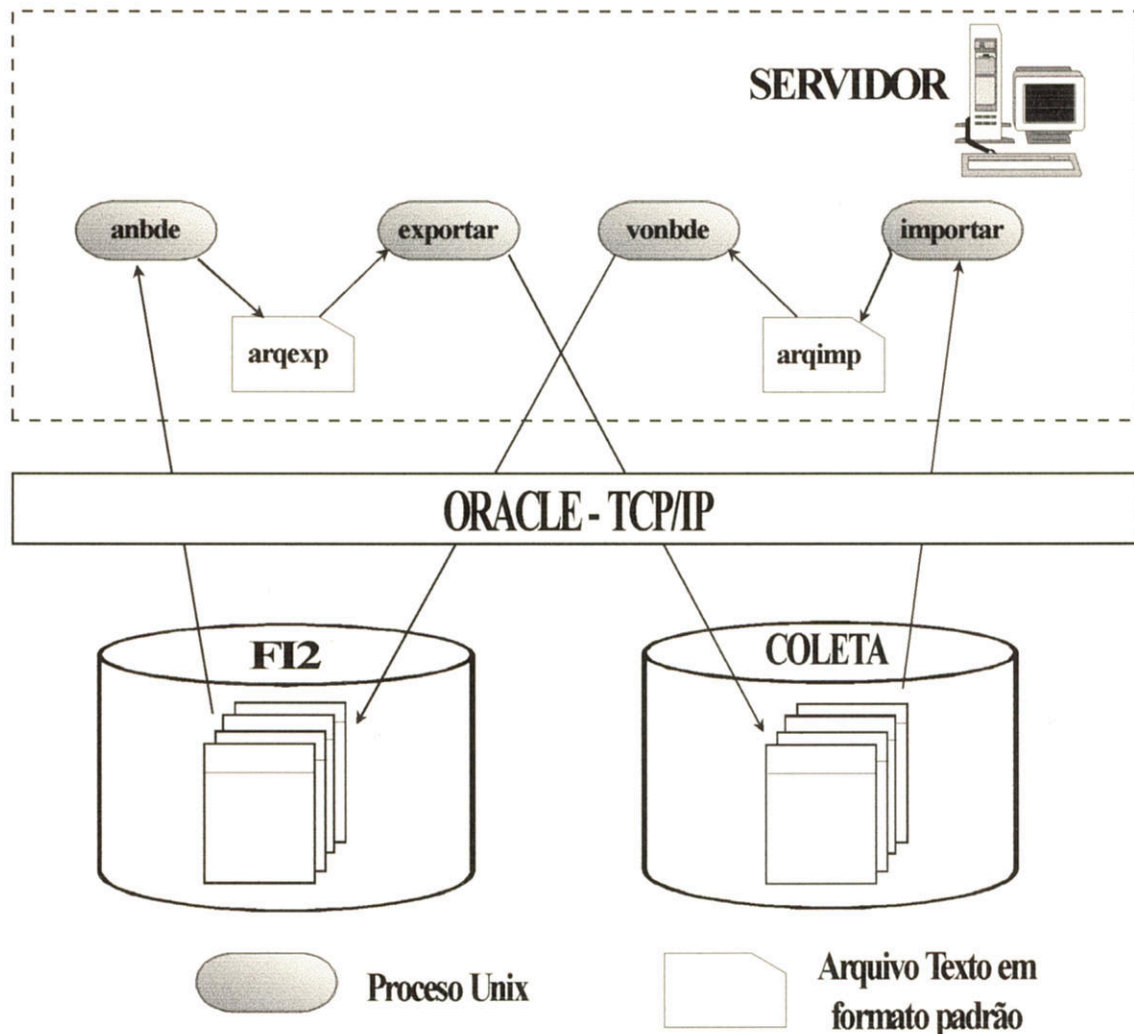


figura 7.4 - Exemplo de troca de dados entre **FI2** e Coleta de Dados

Os arquivos de troca de dados são compostos por campos de tamanho e tipo definidos. Existem diversos padrões definidos para estes arquivos. Cada um deles define um tipo de informação a ser trocada, ou seja, os campos dos arquivos carregam diferentes informações. Existem padrões de arquivos que contém informação sobre ordens de fabricação, outros que contém informação sobre

recursos², etc. Um dos tipos de arquivos que possui informações sobre a coleta de dados, está descrito no *ANEXO I*.

No início de cada arquivo existe um cabeçalho. Nele, além de outras coisas, deve ser identificada uma lógica de operação. Estas lógicas são códigos numéricos de operação definidos dentro do próprio **FI2**. Elas servem para informar ao **FI2**, qual a operação que deve ser realizada com os dados contidos no arquivo. Um mesmo arquivo, com os mesmos dados, pode servir tanto para inserir na base do **FI2** as informações contidas nele, quanto para apagar da base aquelas informações. Isto sendo feito apenas com uma troca da lógica de operação.

Portanto, os campos de dados dos arquivos definem o tipo de informação que se está tratando, enquanto que a lógica define o que fazer com estas informações.

É esta interface padrão que se pretende utilizar para integrar as bases do **MAXIMO** e do **FI2**.

7.2.2 - Escolha do Tipo de Arquivo Padrão

Tendo entendido como funciona a troca de dados com o **FI2**, precisamos, para o restante desta especificação, supor que o **MAXIMO** apresenta sua base de dados em ORACLE. O porquê desta suposição está explicado no capítulo anterior a este.

De toda a discussão sobre a interface padrão do **FI2**, surge ainda uma dúvida: qual tipo de arquivo utilizar, já que o **MAXIMO** não faz parte de nenhum dos três tipos de sistemas que trocam dados com o **FI2**?

É claro que, para integrar o **MAXIMO** com o **FI2**, um dos tipos de arquivo padrão deve ser utilizado, já que, do contrário, o **FI2** não entenderá as informações vindas do **MAXIMO**.

Outro fato, é que a troca de informações entre os dois sistemas será, a princípio, unidirecional, ou seja, do **MAXIMO** para o **FI2**. O sentido inverso não é viável, pois não seria possível a visualização, no **MAXIMO**, da programação da produção feita no **FI2**.

²Recurso, para o FI2, significa qualquer tipo de produto fabricado externamente ou matéria prima necessária para a produção.

O que pode ser implementado, de alguma forma, é o envio de uma mensagem ao **MAXIMO**, por parte do PCP, de que determinada manutenção em determinado dia irá atrapalhar o andamento da produção. Porém isto não é tão crítico, e pode ser feito facilmente, sem a necessidade de se utilizar um computador.

Através do estudo do manual do **FI2**, buscou-se então, dentre os vários tipos de arquivos de dados existentes, encontrar um que atendesse os seguintes requisitos:

- Apresentar, na tela do **FI2**, os horários de manutenção existentes;
- A apresentação deve ser feita através de uma barra, diferente da que é utilizada para a programação das ordens de fabricação;
- De preferência, a linha na qual será feita a manutenção, deve ser bloqueada durante o período;
- É desejável que se possa enviar algum comentário, como por exemplo, se determinada manutenção é extremamente necessária ou não, etc. Este comentário deve ser visualizado também na tela do **FI2**;
- O desbloqueio da linha deve poder ser feito facilmente, de dentro do **FI2**, pelo programador da produção.

A partir de estudos e testes realizados, verificou-se que o único tipo de arquivo que permite claramente a entrada de horários de manutenção, não permite, por sua vez, bloquear a linha durante aquele tempo, e também não mostra na tela qualquer tipo de gráfico indicando a presença de manutenção naquele período.

A solução encontrada foi através de uma função do **FI2** que permite programar uma interrupção de linha por algum tempo. Esta interrupção pode ser, a princípio, por motivos diversos, apesar de que, em se tratando de programação de um tempo futuro, tende a ser por causa de manutenções, ou férias coletivas, já que não é muito óbvio conseguir programar uma quebra de máquina.

Este arquivo é o de número 500 e está descrito em detalhes no *ANEXO I*. Ele é bastante simples, pois contém poucos campos a serem preenchidos. Os campos são os seguintes:

- Header: Cabeçalho comum a todos os outros tipos de arquivo;
- WCNo: Identificação da linha na qual a parada ocorrerá;
- Start: Data e hora de início da interrupção;
- End: Data e hora de fim da interrupção;
- Disruption Reason: Código que define a razão da parada;
- Comment: Comentário que pode ser inserido junto com a interrupção.

A barra que aparece na tela do **FI2**, quando programada uma interrupção via este arquivo, é fina e de cor vermelha, diferindo-se bastante das barras de programação das ordens de fabricação.

O bloqueio da linha acontece da seguinte forma: se desejarmos programar uma ordem de fabricação e sobrepô-la a uma interrupção, isto não será possível, e a barra de programação de ordem aparecerá depois da barra de interrupção. Porém se já houver uma ordem programada, e em seguida for incluída uma interrupção que sobrepõe esta ordem, não haverá problema algum, e a barra de interrupção aparece por cima da barra de programação da OF (ordem de fabricação). Isto é muito útil, pois mostra claramente os casos em que acontecem choques de horários entre manutenção e produção.

Para desbloquear a linha basta deletar, numa tela que mostra todas as interrupções programadas, a interrupção desejada.

Os comentários inseridos no arquivo podem ser visualizados na tela do **FI2**, bastando-se obter os detalhes sobre a interrupção.

Com isto, todos os requisitos citados acima foram atendidos.

O cabeçalho do arquivo padrão terá, então o seguinte aspecto:



figura 7.5 - Cabeçalho do arquivo padrão tipo 500 com lógica para inserir uma interrupção no FI2



figura 7.6 - Cabeçalho do arquivo padrão tipo 500 com lógica para apagar uma interrupção no FI2

O programa que lerá este arquivo e importará os dados para dentro do **FI2**, será o "vonbde". Isto significa que o **MAXIMO** funcionará como se fosse um sistema de coleta de dados.

Com a interface para troca de dados entre os dois sistemas bem definida, serão especificadas as tabelas da base de dados do **MAXIMO**, que serão utilizadas por esta interface.

7.2.3 - Especificação de Tabelas e Campos de Dados Utilizados na Integração

O procedimento para o usuário após a implementação da integração será o seguinte: uma ordem de serviço é programada pelo planejador da manutenção através do **MAXIMO**. Esta ordem é atualizada na base do **FI2** como sendo uma interrupção³ de linha. Quando o programador da produção abrir o **FI2**, ele enxergará,

³É importante lembrar que só serão enviadas ao FI2, as manutenções que exigirem parada de máquina.

juntamente com a programação das ordens de fabricação, barras finas em vermelho, correspondendo a períodos de interrupção de linha por causa de manutenção. Caso estas manutenções não atrapalhem a produção, ele pode ignorá-las. Caso contrário, ele comunica o fato ao planejador da manutenção, o qual irá, através do **MAXIMO**, reprogramar a ordem de serviço.

Quando o planejador da manutenção alterar a hora ou o dia de um serviço, através do **MAXIMO**, a ordem de serviço antiga deve ser apagada da base do **FI2**, e inserida a nova ordem.

O desbloqueio da linha, para viabilizar a programação das ordens de fabricação, pode ser feito de duas maneiras: automaticamente (quando for alterada a hora ou o dia de uma ordem de serviço), ou manualmente pelo próprio programador da produção.

Todos os dados necessários para viabilizar o processo descrito acima, estão armazenados numa única tabela da base de dados do **MAXIMO**: a **WORKORDER**. As colunas desta tabela que serão utilizados são as seguintes:

- **EQNUM**: Indica o número do equipamento no qual será feita a manutenção;
- **STATUS**: Indica o estado da OS (ordem de Serviço). Serão utilizadas apenas as OS que possuírem **INPRG** (em progresso) ou **WAPR** (esperando aprovação);
- **WORKTYPE**: Indica o tipo de manutenção. Serão utilizadas apenas as manutenções preventivas: tipos **PM** ou **LB**;
- **PRIORITY**: Indica prioridade da OS. Será enviada ao **FI2** como comentário;
- **TARGSTARTDATE**: Data e hora de início da manutenção;
- **TARGCOMPDATE**: Data e hora de fim da manutenção;
- **DOWNTIME**: Indica se é ou não necessária a parada de máquina. Importante na hora de buscar os dados na base.

No procedimento de troca de dados é necessário, como visto acima, que sejam inseridas novas interrupções e apagadas as

antigas. Estas últimas dizem respeito às interrupções que foram alteradas, e também àquelas com data inferior à data atual.

Serão criadas, dentro da base de dados do **MAXIMO**, duas tabelas, as quais conterão apenas os dados necessários para serem enviados ao **FI2**. Uma tabela, que será denominada "TABMARCARFI2", conterá os dados que deverão ser escritos na base do **FI2**. A outra tabela, denominada de "TABAPAGARFI2", conterá os dados que serão apagados da base do **FI2**.

Os campos da tabela "TABMARCARFI2" serão os seguintes:

- DtHrInicio: Campo que contém a data e a hora do início da manutenção;
- DtHrFim: Campo que contém a data e a hora do fim da manutenção;
- EQNUM: Campo que contém o número do equipamento no qual será feita a manutenção;
- WONUM: Campo que contém o número da OS. Isto será necessário para facilitar a identificação das interrupções que chocam horários com as OF's.
- PRIORITY: Campo que servirá para informar ao planejador da produção a prioridade da OS;

Os campos da tabela "TABAPAGARFI2" serão os seguintes:

- DtHrInicio: Campo que contém a data e a hora do início da manutenção;
- DtHrFim: Campo que contém a data e a hora do fim da manutenção;
- EQNUM: Campo que contém o número do equipamento no qual será feita a manutenção;

Desta forma ficam definidas, integralmente, as tabelas necessárias para a troca de dados entre **MAXIMO** e **FI2**. Resta-nos definir os processos e outros mecanismos também necessários para a troca de dados.

7.2.4 - Especificação dos Processos e Outros Mecanismos

É necessário que, a cada vez que for criada uma ordem de serviço na base de dados do **MAXIMO**, esta seja enviada à base do **FI2**. Isto será feito, é claro, desde que atenda aos requisitos necessários, os quais são:

- Ser do tipo PM ou LB - manutenção preventiva;
- Necessitar de parada de máquina.

O mecanismo *trigger*, do ORACLE (explicado no capítulo II), permite executar um procedimento (definido pelo usuário), a cada vez que for realizada um operação sobre determinada tabela. Este mecanismo será utilizado para atualizar as tabelas "TABMARCARFI2" e "TABAPAGARFI2", dentro da base do **MAXIMO**.

Serão criados dois *triggers* (denominaremos provisoriamente de TRIGGER1 e TRIGGER2): um que atualizará a tabela "TABMARCARFI2", e que será "disparado" imediatamente após cada inserção ou atualização na tabela WORKORDER (TRIGGER2). O outro *trigger* (TRIGGER1) atualizará a tabela "TABAPAGARFI2", e será "disparado" imediatamente antes de cada deleção ou atualização na tabela WORKORDER.

A atualização da tabela WORKORDER é definida para os dois *triggers*, pois há a necessidade de escrever na tabela "TABAPAGARFI2" os dados antigos, e em seguida, escrever na tabela "TABMARCARFI2", os dados atualizados ou novos.

Depois de atualizadas as tabelas "TABMARCARFI2" e "TABAPAGARFI2", é necessário um processo em Unix, que leia estas tabelas, e escreva os dados no arquivo padrão reconhecido pelo **FI2**. Este processo, que será denominado de "pAtualizaManutFI2", conforme padrão utilizado, e terá as seguintes funções:

- Transferir para o arquivo de troca de dados, as interrupções encontradas em "TABAPAGARFI2", e retirá-las desta tabela;
- Transferir para o arquivo de troca de dados, as interrupções encontradas em "TABMARCARFI2", e retirá-las desta tabela;

Depois de escrito o arquivo padrão, que para a coleta de dados possui o nome "bdeanfi2", deve ser rodado o programa "vonbde", o qual atualizará as tabelas do **FI2**, disponibilizando na tela as interrupções de linha programadas.

A periodicidade com que serão disparados estes processos ("pAtualizaManutFI2" e "vonbde") deve ser analisada depois de implementada a integração, através de testes. Deverá haver um meio termo entre necessidade do Programador da produção e performance do sistema.

A figura 7.7, mostra a arquitetura da integração proposta.

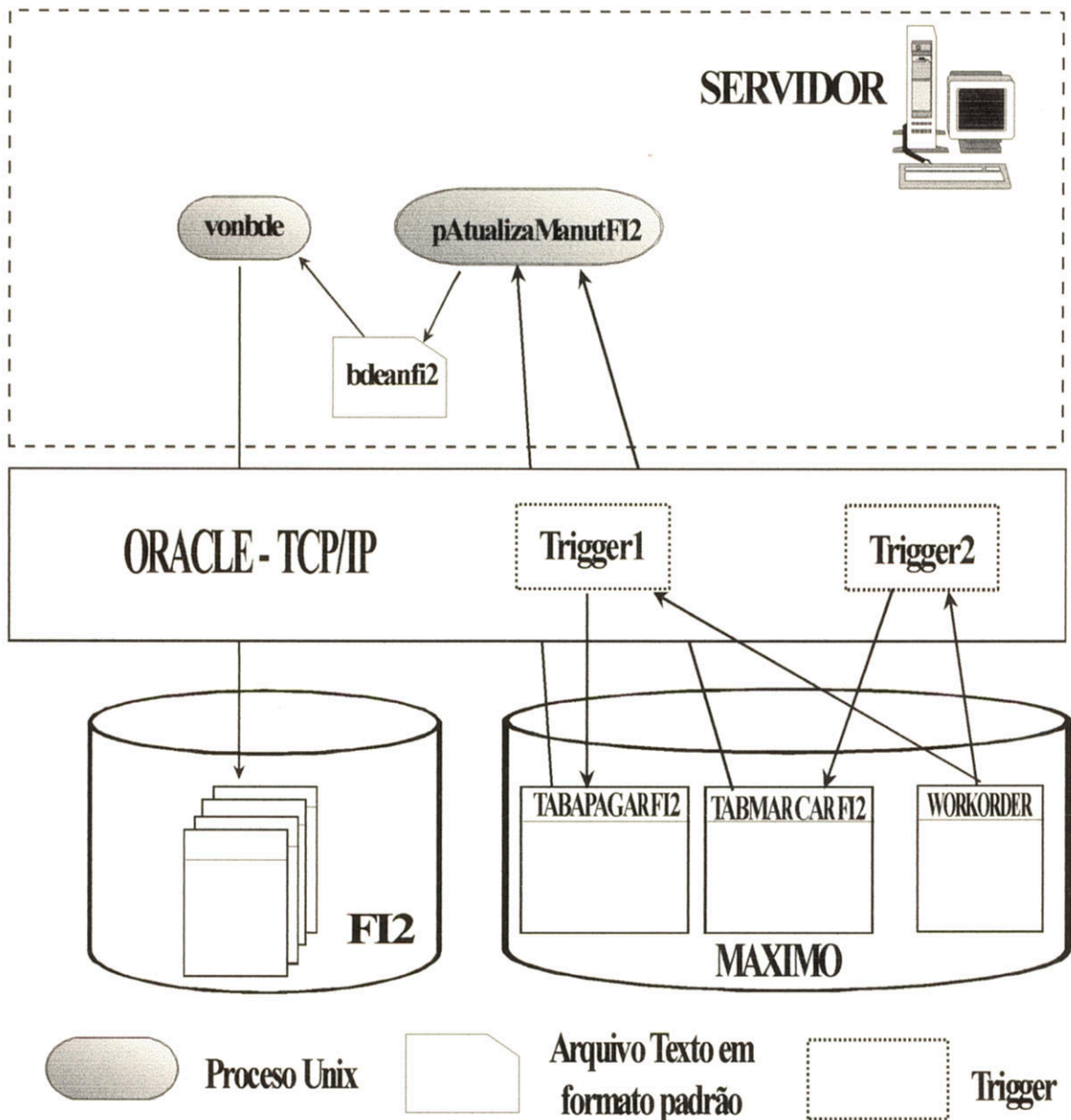


figura 7.7 - Arquitetura da Integração Proposta

Uma dúvida que pode aparecer ao visualizar a figura é a seguinte: *Por que não utilizar o processo "pAtualizaManutFI2" para realizar as procuras diretamente na tabela WORKORDER, e escrever os dados no arquivo padrão? Isto eliminaria duas tabelas e dois triggers.*

Duas razões explicam a utilização das tabelas e *triggers*:

- Na troca de dados que existe atualmente entre a coleta do **SII** e o **FI2** também se utiliza esta configuração (definida pela empresa fornecedora do sistema **SII**). Portanto, é desejável manter um padrão.
- Os processos em UNIX não têm a mesma capacidade que o *trigger* possui, de somente realizar operações quando a tabela for alterada. Isto significa que o processo deveria, periodicamente, fazer uma busca em toda a tabela a fim de recuperar dados. Outro fato, é que todas as linhas da tabela devem ser verificadas, e não somente aquelas que serão alteradas (como faz o *trigger*). Isto, dependendo da periodicidade de disparo do processo, pode perfeitamente, deteriorar a performance do **MAXIMO**.

Quanto aos códigos de programa do processo e dos *triggers*, já se tem uma idéia de como serão. O processo em Unix é bastante simples de ser implementado, e seguirá o mesmo formato do processo já existente na coleta, pois praticamente realiza a mesma função.

Já os códigos dos *triggers* foram esboçados, dado que não existe, no sistema, nenhuma utilização deste mecanismo. Este código está mostrado no *ANEXO II*.

Com isso, a integração fica totalmente especificada e viável de ser implementada.

Conclusão

Como consequência dos estudos realizados neste trabalho, foi especificada a integração entre dois sistemas existentes na **EMBRACO - Fábrica II**: o Sistema de Gerenciamento de Manutenção (**MAXIMO**) e o Sistema de Informação Industrial (**SII**).

Da proposta original do trabalho, que também incluía a especificação da integração entre **MAXIMO** e **GDT** (Gerenciador de Documentação Técnica), concluiu-se que esta última é inviável de ser implementada, devido a própria arquitetura do sistema **GDT**, como explicado no capítulo VI.

Já a integração entre **MAXIMO** e **SII** foi especificada em detalhes, apenas com a restrição de que o **MAXIMO** deve, antes de mais nada, possuir sua base de dados em ORACLE (atualmente está em SQLBASE), devido a alguns fatores explicados no capítulo VI. Não fosse esta restrição, a integração poderia ter sido implementada, mesmo que parcialmente, ou mesmo apenas a nível de testes.

A integração destes sistemas trará melhorias consideráveis na troca de informações entre os setores a que pertencem, principalmente entre o PCP e a manutenção. A disponibilização de funções do **MAXIMO** no coletor de dados também proporcionará uma maior facilidade e rapidez no envio de informações, pertinentes à manutenção, do chão-de-fábrica para o **MAXIMO**.

Os objetivos iniciais deste trabalho foram alcançados, mostrando assim, que a integração entre sistemas que funcionam isolados, quando viável de ser implementada, representa uma solução bastante interessante na automatização do processamento e do fluxo de informações em uma fábrica.

Perspectivas

A implementação da integração especificada neste trabalho depende basicamente de três fatores:

- Maior conhecimento e utilização do software **MAXIMO**, por parte do pessoal da manutenção (técnicos, planejadores etc.);
- Término da fase inicial de implantação do Sistema de Informação Industrial (**SII**);
- Compra da versão do **MAXIMO** para ORACLE e migração dos dados para a nova base;

Tem-se como perspectiva, que os sistemas **MAXIMO** e **SII** estarão em pleno funcionamento ao final de 1996. Sendo assim, a implantação da integração deverá ocorrer no ano de 1997, após a compra da versão do **MAXIMO** para ORACLE.

Existe a possibilidade de inclusão de outras funções do **MAXIMO** no coletor de dados, além daquelas avaliadas como inicialmente viáveis (ver capítulo VI). No entanto, é necessário que primeiramente seja dominado o funcionamento da base de dados do **MAXIMO**. Ou seja há a necessidade de se conhecer quais tabelas são atualizadas em cada operação que se deseja fazer. Isto diminuirá os riscos de se criar inconsistências na base do **MAXIMO**.

Outra perspectiva, é que a integração entre **MAXIMO** e **GDT** possa ocorrer futuramente. É sabido que, provavelmente, a nova versão do software **GDT** (ainda em desenvolvimento) possibilitará esta integração. Por enquanto, o **GDT** pode ser acionado de dentro do **MAXIMO**, como qualquer outro aplicativo. Isto não é o mais desejado, como discutido no capítulo VI.

Bibliografia

- [1] *Furtado, A.L. e C.S. dos Santos. Organização de Banco de Dados. 2ª Edição. Editora Campus, Rio de Janeiro. 1980.*
- [2] *Stemmer, Marcelo. Apostila de Informática Industrial II. 1994.*
- [3] *Wright, Patrícia e Kamali Abelson. SQLBase - Database Administrator's Guide. Versão 5.2. Gupta Technologies, Inc. Maio de 1994.*
- [4] *Maximo Series5 - System Administrator's Guide. Project Software & Development, Inc (PSDI). Versão 2.1. 1994-1995.*
- [5] *Maximo Series5 - User's Manual. Project & Software Development, Inc (PSDI). Versão 1.0.1. 1994.*
- [6] *Bobrowski, Steven. ORACLE7™ Server - Concepts Manual. ORACLE Corporation. Dezembro de 1992.*
- [7] *Intersolv DataDirect ODBC Drivers. Intersolv, Inc. Junho de 1995.*
- [8] *Microsoft Excel - Guia do Usuário do Visual Basic®. Versão 5.0. Microsoft Corporation. 1993-1994.*
- [9] *Microsoft Excel - Guia do Usuário. Versão 5.0. Microsoft Corporation. 1993-1994.*
- [10] *FORMTEK:TDM - A Complete VPSC Solution. FORMTEK, Inc. Dezembro de 1994*
- [11] *Intelligent Leitstand FI2 - System Manual. IDS Prof. Scheer CAM GmbH. Julho de 1994.*

Anexo I

Arquivo padrão para troca de dados com o FI2 - **TIPO 500**

Nº	Nome do Campo	Tipo	Tamanho	mandatório/opcional	Posição
00	Header	char	22	m	001-022
01	WCNo	char	12	m	023-034
02	Start	date	12	o/m	035-046
03	End	date	12	o/m	047-058
04	Disruption Reason	number	05	o	059-063
05	Comment	char	20	o	064-083

Anexo II

Esboço dos programas dos *trigger's*

Trigger 1

```
Before Update or Delete on WorkOrder { Colunas alteradas }
When { critério }
For Each Row

Insert Into TabApagarFI2 DtHrInicio, DtHrFim, EqNum
    Values (:old.TargStarDate, :old.TargCompDate, :old.Eqnum)

End;
```

Trigger 2

```
After Update or Insert on WorkOrder { Colunas alteradas }
When { critério }
For Each Row

Insert Into TabMarcarFI2 DtHrInicio, DtHrFim, EqNum, WONUM
    Values (:new.TargStarDate, :new.TargCompDate, :new.Eqnum,
           :nem.WONUM)

End;
```


Anexo III

A manutenção, assim como todos os outros setores da empresa, necessita demonstrar o seu desempenho nas atividades realizadas. Existem diversas formas de se fazer isso, porém a mais difundida é sem dúvida, através de planilhas e gráficos. Na **EMBRACO**, o software mais utilizado para a criação destas planilhas e gráficos é o MS EXCEL.

No entanto, obter manualmente os dados para a confecção destes relatórios nem sempre é tão simples e rápido, e com isso, determinados relatórios podem demorar horas para serem feitos.

O MS EXCEL possui uma linguagem de programação própria que permite ao usuário implementar quaisquer tipos de funções. Estas funções são denominadas *MACROS*, e escritas em *Visual Basic* próprio do EXCEL [8]. Os comandos desta linguagem permitem configurar páginas, abrir planilhas, criar gráficos, etc, e tudo isso automaticamente. Pode-se também associar estas funções a um botão, colocando-o dentro de qualquer planilha em qualquer pasta de trabalho¹. Quando acionado o botão a função será executada automaticamente.

O MS EXCEL também possui interface com o MS QUERY, através do qual pode-se realizar qualquer tipo de procura em uma base de dados.

Logo, pode-se criar no EXCEL *macros* que automaticamente busquem dados desejados numa base qualquer.

Devido às necessidades de criação de planilhas e gráficos apresentadas acima, surgiu a oportunidade de se trabalhar com estas ferramentas.

Os seguintes relatórios foram desenvolvidos para o setor de manutenção:

- Plano de Inspeção do Operador (PIO): plano de trabalho diário, no qual o operador inspeciona

¹No EXCEL, um arquivo é denominado de pasta de trabalho, e este, por sua vez, pode ser dividido em várias planilhas.

alguns itens do equipamento, dizendo se estão normais ou anormais;

- Calendário de Manutenções Preventivas (CMP):
Relatório útil ao planejador. Permite visualizar como está distribuída a programação das manutenções preventivas no ano (M - Mensais, T - Trimestrais, S - Semestrais, A - Anuais, B - Bianuais).
- Análise de Execução de Manutenções Preventivas:
Existe na forma de dois gráficos - um que mostra o desempenho de cada centro de custo na execução das manutenções corretivas (CM), preventivas (PM) e de lubrificação (LB), e outro gráfico que mostra o desempenho geral (todos os centros de custo) na execução das manutenções preventivas. Este relatório é uma forma de medir a produtividade do setor de manutenção.

Todos estes gráficos são mostrados abaixo.

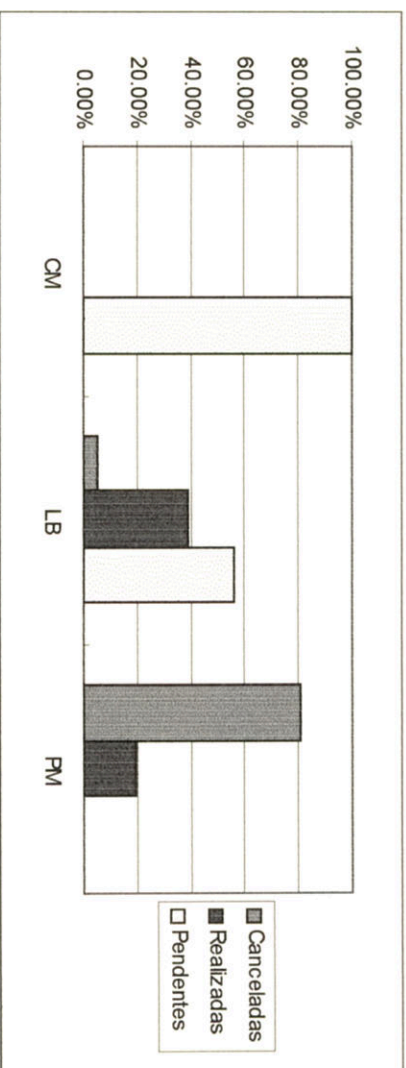
O código de uma *macro*, que realiza a obtenção dos dados para o relatório PIO, é também listada em seguida.



Análise de Execução de Manutenções Preventivas por UGB

Controle Mensal

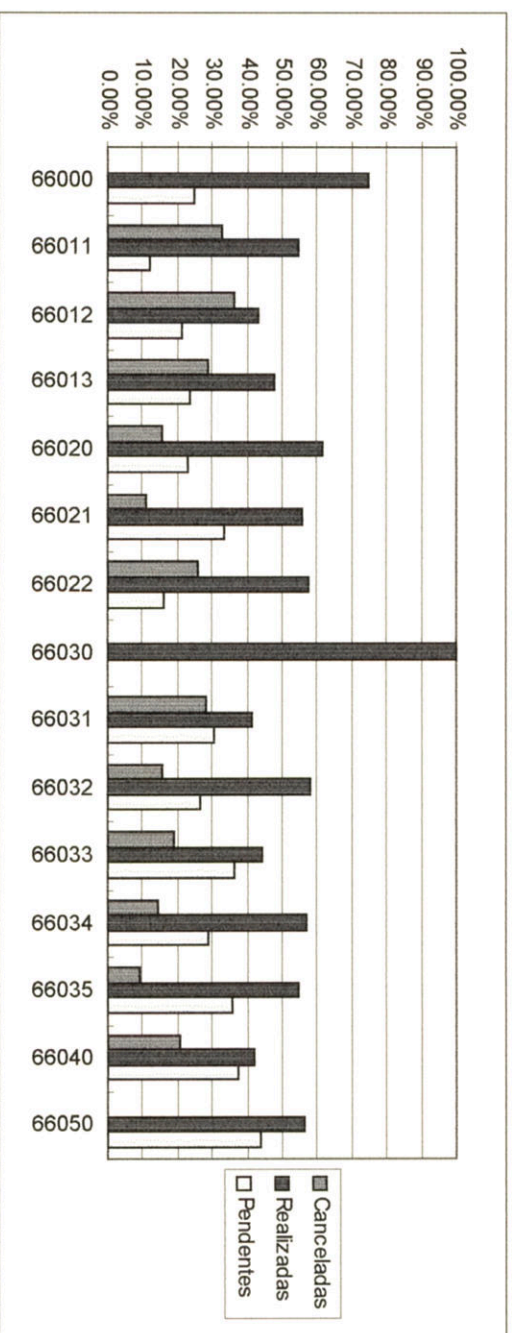
UGB - SUPORTE - FÁB2



Tipo de OS	Canceladas	Realizadas	Pendentes	TOTAL
CM	0	0	26	26
LB	2	16	23	41
PM	74	18	0	92
TOTAL	76	34	49	159

Análise de Execução de Manutenções

Controle Mensal



Centro Custio	Canceladas	Realizadas	Pendientes	TOTAL
66000	0	15	5	20
66011	27	45	10	82
66012	72	86	42	200
66013	6	10	5	21
66020	2	8	3	13
66021	8	42	25	75
66022	32	71	20	123
66030	0	4	0	4
66031	83	122	89	294
66032	24	90	41	155
66033	31	72	59	162
66034	5	20	10	35
66035	5	29	19	53
66040	68	138	124	330
66050	0	39	30	69
TOTAL	363	791	482	1636

Abaixo é mostrado o código de programação de uma *macro* que realiza a busca de dados para a confecção do relatório PIO

```

'
' Macro ObtemDados
' Macro gravada em 04/12/95 por Elmo Martins
'
'
Proc ObtemDadosPIO(m)
  Se TestaMacrosPIO = Verdadeiro Então
    PlanilhasTrabalho("Dados").Ativar
  Senão
    PlanilhasTrabalho.Adicionar
    PlanilhaAtiva.nome = "Dados"
  Fim Se
  Intervalo("A1").Selecionar
  PrimeiroDia = CData("01-" & m & "-" & Ano(Data))
  Str1 = Formato(PrimeiroDia; "aaa-mm-dd")
  Aplicativo.Executar "ConsultaInfoDados"; ""; _
"SELECT JOBOPERATION.DESCRPTION, EQUIPMENT.EQNUM, EQUIPMENT.DESCRPTION,
JOBOPERATION.JO1, EQUIPMENT" _
  ; ; ; ; Falso
  Aplicativo.Executar "ConsultaInfoDados"; ""; _
".COSTCENTER, WORKORDER.WONUM FROM MAXIMO.EQUIPMENT EQUIPMENT,
MAXIMO.JOBOPERATION JOBOPERATION, MAX" _
  ; ; ; ; Falso
  Aplicativo.Executar "ConsultaInfoDados"; ""; _
"IMO.WORKORDER WORKORDER WHERE EQUIPMENT.EQNUM = WORKORDER.EQNUM AND
JOBOPERATION.JPNUM = WORKORDER." _
  ; ; ; ; Falso
  Aplicativo.Executar "ConsultaInfoDados"; ""; _
"JPNUM AND ((WORKORDER.PMNUM Like 'PV%') AND (WORKORDER.STATUS='INPRG') AND
(WORKORDER.STATUSDATE>{ts" _
  ; ; ; ; Falso
  Aplicativo.Executar "ConsultaInfoDados"; ""; _
" " & Str1 & "}))" _
  ; ; ; ; Falso
  Aplicativo.Executar "ConsultaInfoDados"; ""; _
" ORDER BY EQUIPMENT.COSTCENTER, WORKORDER.WONUM" _
  ; ; ; ; Falso
  Aplicativo.Executar "ConsultaInfoDados";
"DSN=MAXIMO;UID=MAXIMO;DB=EMBRACO;SRVR=SERVER1;PWD=mips"; _
""; Verdadeiro; Verdadeiro; Falso; Intervalo("$A$1"); Verdadeiro
  Colunas("B:B").Selecionar
  Seleção.FormatoSúmero = "dd/mm/aa"
  Planilhas("PIO").Selecionar

Fim Proc

```