

Estado de Goiás
Polícia Militar
Academia de Polícia Militar
Curso de Aperfeiçoamento de Oficiais

Sistema de Gerenciamento de Pessoal
Ferramentas para administração da
Informação

Avilmar Santos Ferreira

Goiânia - 1997

POLÍCIA MILITAR
ACADEMIA DE POLÍCIA MILITAR
CURSO DE APERFEIÇOAMENTO DE OFICIAIS

Aperfeiçoando: CAP PM 14601 Avilmar Santos Ferreira

Orientador: Gilberto de Almeida

SISTEMA DE GERENCIAMENTO DE PESSOAL

Tema: Sistema de Gerenciamento de Pessoal

Ferramentas para administração da Informação

Problema Formulado: Como gerenciar com eficiência as informações do pessoal da Polícia Militar do Estado de Goiás através de sistema computacional?

Goiânia, Julho de 1997

SISTEMA DE GERENCIAMENTO DE PESSOAL

Trabalho elaborado em cumprimento à exigência do currículo do **Curso de Aperfeiçoamento de Oficiais, CAO/97**, da Academia de Polícia Militar do Estado de Goiás.

Academia de Polícia Militar do Estado de Goiás

Goiânia, GO, Julho 1997.

Agradecimentos

Agradeço ao pessoal empregado na operação do sistema **de Administração Geral e Integrada, AGI**, da Polícia Militar do Estado de Goiás e aos técnicos da Grupom Informática pelo suporte dispensado durante a elaboração deste projeto.

Dedicatória

A minha mãe, Sra Judite, por ter me mostrado com sabedoria o árduo caminho da nossa existência, à minha esposa Vera Lúcia, meus filhos Michelle e Erik por me dar forças para vencer os desafios da vida.

“Vim, vi e venci”

Caius Julius Cesar - Estadista Romano

ÍNDICE

Introdução

1. Justificativa	1
1.1. Relevância	1
1.2. Viabilidade	1
1.3. Originalidade	2
1.4. Interesse Pessoal	3
1.5. Metodologia	4
2. Objetivos	5
2.1. Objetivo Geral	5
2.2. Objetivos Específicos	5
3. Quadro Teórico	6
4. Formulação das Hipóteses	8
5. Situação Atual	9
6. Análise da Situação Atual	17
7. Situação Proposta	20
Apresentação	21
Abertura	22
Tela Principal	23
Dados Funcionais - Promoção	24
Dados Funcionais - Campanha	25
Dados Funcionais - Agregação	26
Dados Gerais	27
Dados Gerais - Elogios/Punições	28
Dados Gerais - Condenações	29
Dados Gerais - Medalhas	30
Dados Gerais - Condecorações	31
Dados Gerais - Comissões	32
Dados Gerais - Funções	33
Dados Gerais - Empregos	34
Dados Gerais - Afastamentos	35
Movimentação	36
Dado Pessoal	37
Dependentes	38
Cursos	39
Documentação	40
Tempo de serviço - consulta	41
Tempo de serviço - Averbação	42

Tela de saída do sistema	43
Pesquisa - Averbação	44
Pesquisa - Consulta tempo de serviço	45
Pesquisa - Documentação	46
Pesquisas - Cursos	47
Pesquisa - Dependentes	48
Pesquisa - Dados Pessoal	49
Pesquisa - Movimentação	50
Pesquisa - Afastamentos	51
Pesquisa - Emprego	52
Pesquisa - Funções	53
Pesquisa - Comissões	54
Pesquisa - Condecorações	55
Pesquisa - Medalhas	56
Pesquisa - Condenações	57
Pesquisa - Punições	58
Pesquisa - Elogios	59
Pesquisa - Agregação	60
Pesquisa - Unidades	61
Pesquisa - Campanha	62
Pesquisa - Promoção	63
Opções de Menu Manutenção	64
Calculadora	65
Backup do Banco de Dados	66
Controle de Senhas	67
Senha - Cadastro e Exclusão	68
Opções do Menu - Janela	69
Desenvolvimento	70
Sobre o produto	71
Opções do Menu abrir	72
Opções do Menu Pesquisa	73
Opções do Menu Relatório	74
Menu Abrir	75
Menu Pesquisa	76
Menu Pesquisa - Dados Gerais	77
Menu Pesquisa - Dependentes/Cursos	78
Menu Pesquisa - Tempo de Serviço	79
Menu Relatório - Imprimindo	80
Menu Manutenção - 3ª Manutenção	81
Menu Manutenção - 4ª Manutenção	82
Menu Manutenção - 5ª Manutenção	83
Menu Manutenção - 6ª Manutenção	84
Menu Ferramenta	85
Menu Relatório - Informação Pessoal	86
Menu Relatório - Tempo de Serviço	87
Menu Relatório - Tempo de Averbação	88
Menu Relatório - Dependentes	89
Menu Relatório - Documentação	90

8. Códigos de Programação	91
9. Cronograma de Implantação	158
9.1. Levantamento dos equipamentos existentes.	158
9.2. Definição e implantação da rede	158
9.3. Programação	158
9.4. Implantação	158
9.5. Treinamento	159
9.6. Teste de operação	159
9.7. Operações em cópias beta	159
9.8. Manutenção	159
10. Quadro de Desenvolvimento da Proposta	160
11. Do Pessoal	161
12. Consultoria	162
13. Custo da Proposta	163
14. Implantação da Rede	164
14.1. Internet	164
14.2. Linhas Dedicadas	164
15. Propostas para Implantação	165
16. Conclusão	167
17. BIBLIOGRAFIA	168

INTRODUÇÃO

1. JUSTIFICATIVA

1.1. RELEVÂNCIA

As empresas que dispõem de redes locais de computadores buscam o máximo de desempenho, produtividade e segurança no gerenciamento de suas informações, tendo em vista as facilidades hoje oferecidas pelo conjunto hardware e software, associado às relativas facilidades de programação interativa com os bancos de dados existentes no mercado da informática. A Polícia Militar do Estado de Goiás necessita de um sistema de gerenciamento eficaz de informação voltado para a administração de dados de pessoal, considerando que seu efetivo se encontra distribuído em todo o Estado, sendo difícil saber, no presente momento, informações atualizadas deste contingente, por falta de um sistema eficiente de gerenciamento de dados na corporação capaz de administrar a informação instantânea.

1.2. VIABILIDADE

A viabilidade do presente projeto fundamenta-se no fato de se ter disponível o sistema de **Administração Geral e Integrada, AGI**, desenvolvido pela EMCIDEC e operado em diversas unidades da capital e interior, o que em muito contribuirá para a assimilação do sistema proposto, além de já se ter propagado a

cultura da informática no seio da corporação, através da aquisição de microcomputadores pelas unidades da capital e do interior.

1.3. ORIGINALIDADE

O atual sistema de pessoal da Polícia Militar do Estado de Goiás é uma aplicação desenvolvida há mais de dez anos pela EMCIDEC, não tendo sofrido as manutenções necessárias para responder às constantes demandas de informações de administração de pessoal, bem como incorporar as novas alterações dos últimos anos. É um sistema pesado, que roda em grande porte, *mainframe*, plataforma que pouco a pouco está sendo substituída por máquinas menores e potentes, quando se constata que as empresas modernas migram para máquinas menores, através de uma proposta mundialmente conhecida como *dowsizing*, absolvendo, desta maneira, uma forma mais agradável e interativa de se extrair informações de bancos de dados cada vez mais amigáveis e agradáveis de se operar.

Esta nova filosofia de se gerenciar informações dentro das empresas, adotando a plataforma das máquinas desenvolvidas para o ambiente windows, que é uma irreversível tendência mundial, deve estar presente também dentro dos objetivos de gerenciar informações na Polícia Militar. A melhor mercadoria é a informação. Sem a informação não se vai a lugar nenhum. Quem não gerencia sua informação se perde no tempo, perde sua identidade e poderá ser atropelado pelo progresso emergente imposto pela dinâmica de uma sociedade exigente, moderna e competitiva.

O projeto **Sistema de Gerenciamento de Pessoal** é moderno e original porque não existe atualmente na Polícia Militar um software que gerencie globalmente as informações de pessoal em uma única base de dados, o que torna difícil o conhecimento da real situação do pessoal que compõe os diversos quadros da Polícia Militar. A informação existente não é atual. A originalidade do presente projeto é pois a de oferecer a informação instantânea de pessoal, sendo que este sistema será desenvolvido para integrar dados das unidades da Polícia Militar que geram informações referentes à vida funcional de seu pessoal.

1.4. INTERESSE PESSOAL

Desenvolver projetos de sistemas ou orientá-los, como também estudar e analisar modelagem de dados e análise estruturadas de projetos de sistemas estratégicos de informações, tem sido objeto de estudos pelo autor da presente proposta.

Através dos diversos cursos de programação, lógica de programação e análise de sistemas realizados em institutos de ensino e empresas especializadas, além de estágios realizados em CPD de desenvolvimento de sistemas é que me sinto à vontade para contribuir com a formação de conhecimento neste campo para a Polícia Militar do Estado de Goiás, esperando que este projeto desperte o interesse das nossas autoridades superiores ao sentir a importância de se ter as informações do nosso pessoal sob o nosso domínio. Este projeto é uma oportunidade para manifestar o conhecimento e a experiência adquirida no decorrer de anos dedicados ao estudo da computação e esperar que ele possa contribuir para a solução de gerenciamento de informação de pessoal na Polícia Militar do Estado de Goiás.

1.5. METODOLOGIA

A presente proposta teve o seu desenvolvimento através da pesquisa bibliográfica como fonte auxiliar para a implementação do sistema (Software) , sendo que a pesquisa para a sua elaboração se deu através do levantamento e análise da situação atual do sistema **Administração Geral e Integrada, AGI**, da EMCIDEC, além da informatização da Ficha individual da Alterações existente na unidade da Polícia Militar do Estado de Goiás e informações colhidas com os operadores do sistema vigente.

2. OBJETIVOS

2.1. OBJETIVO GERAL

Apresentar um sistema de gerenciamento de pessoal administrado por um banco de dados capaz de receber e atualizar a informação, globalizando todas as alterações funcionais na vida do policial enquanto ele interferir na produção da informação.

2.2. OBJETIVOS ESPECÍFICOS

- a) Permitir que seja efetuada a entrada e o processamento de dados;
 - b) Permitir a emissão de relatórios previamente definidos;
 - c) Permitir que se monte relatórios interativos diversos com base nos registros existentes no banco de dados.
-

3. QUADRO TEÓRICO

Com o progresso das telecomunicações, facilitando os processos de conectividade dos diversos equipamentos de informática, verificamos o crescimento e a difusão dos recursos da informática e o advento das redes locais e/ou remotas revolucionando, assim, o mercado da comunicação através dos diversos processos da teleinformática.

Os grandes e poderosos computadores da década de 80, conhecidos como *mainframes*, que vieram a formar os grandes CPD, hoje estão em desuso devido à miniaturização dos circuitos integrados e poderosos compactadores de dados que atualmente revolucionam os modernos sistemas de gerenciamento de dados.

Redes formadas por um servidor e algumas dezenas de terminais tem o custo desprezível ao analisarmos a importância do gerenciamento desses dados. Quem é dono da informação, sente-se responsável pela sua administração para não ver sua eficiência ser derrotada pela falta de uma política voltada para a informatização da empresa. A segurança, através dos sistemas de gerenciamento de dados tem a função de garantir a integridade, confiabilidade e disponibilidade dos dados ao usuário.

Através da aquisição e distribuição de diversos microcomputadores às unidade da Polícia Militar, o pessoal hoje encontra-se mais preparado para entender a necessidade e a operacionalidade de um sistema integrado de pessoal. Esses

microcomputadores poderão estar ligados via internet ou através de linhas dedicadas, sendo colocados também em disponibilidade para as necessidades diversas da unidade através de seu próprio *hard disk*. O que deverá ser realizado será portanto interligá-los ao servidor e o operador acessar a base de dados, realizando, através desta operação, a atualização do banco de dados de pessoal.

4. FORMULAÇÃO DAS HIPÓTESES

- O sistema de **Administração Geral e Integrada**, AGI, gerencia as informações de pessoal da Polícia Militar do Estado de Goiás;
 - O AGI não recebe a manutenção necessária à prolongação de sua vida útil;
 - Desde a implantação do AGI alguns módulos ficaram a espera de implementação, o que não ocorreu;
 - Gerar relatórios via AGI é uma tarefa cada vez mais impraticável, devido à feitura de códigos para esta geração e o esvaziamento da mão-de-obra técnica na empresa responsável pela manutenção do sistema;
 - O AGI é desprovido de documentação técnica;
 - Algumas unidades da Polícia Militar têm buscado sistemas isolados de administração, não globalizados, por sentirem que o AGI não responde, com eficácia, as questões relativas à administração de dados de seu pessoal;
 - Sem a difusão de uma cultura voltada para a administração de dados de pessoal da organização, pela própria organização, dentro do quadro atual, todo e qualquer sistema de informação está fadado a falhar;
 - Sistemas isolados de informações não globalizam dados.
-

5. SITUAÇÃO ATUAL

O atual sistema de pessoal da Polícia Militar do Estado de Goiás, desenvolvido pela EMCIDEC se encontra dividido em seu menu principal, em quatro opções:

1. Pessoa Física

2. Funcionários

3. Urbanismo

4. Entidades relacionais

A primeira opção, Pessoa Física, quando acessada, oferece as seguintes opções:

- 1.1. Inclusão
 - 1.2. Consulta
 - 1.3. Atualiza endereço
 - 1.4. Casamento
 - 1.5. Situação Judicial / Divórcio
 - 1.6. Óbito
 - 1.7. Consulta óbito
 - 1.8. Atualiza formação profissional
 - 1.9. Consulta formação profissional
 - 1.10. Inclui documentação
 - 1.11. Consulta documentação
 - 1.12. Consulta formula datiloscópica
 - 1.13. Inclui certidão de nascimento
 - 1.14. Atualiza certidão de nascimento
 - 1.15. Atualiza documentação
 - 1.16. Atualiza certidão de casamento
 - 1.17. Atualiza nome de pessoa
 - 1.18. Altera nome de pessoa
 - 1.19. Inclusão de homônimos
-

Quando acessada a opção 1.1, Inclusão, temos as seguintes informações a serem passadas para o sistema:

Nome:

CPF:

Data de nascimento:

Nome da mãe:

Apelido:

Sexo:

Filiação: Pai:

Mãe:

Nascimento:

Local:

Estado civil:

Certificado:

Cartório:

Número registro:

Livro:

Folha:

Dt. Expedição:

Cônjuge:

Casamento local:

Matrícula

Tipo:

Nome:

Número:

Quadra:

Lote:

Complemento:

Bairro:

Município:

U.F.:

Telefone:

Ramal:

Doador (s/n) :

Está primeira opção inclui no sistema informações básicas do funcionário. São informações referentes aos dados pessoal. Nesta opção nota-se uma mistura de informações quando consta no sistema informações de documentação e informação pessoal. O sistema deve ter uma opção para inclusão de informação pessoal e outra para documentação, pois isto facilita a manutenção do sistema e maximiza o seu acesso. Sendo realizada esta alteração, as opções 1.10 e 1.11, Inclui Documentação e Consulta Documentação, poderia ser eliminada do módulo principal. Também poderia ser inserida nesta alteração as opções, 1.13, 1.14, 1.15, 1.16 e 1.18, inclui Certidão de Nascimento, Atualiza Certidão de Nascimento, Atualiza Documentação, Atualiza Certidão de Casamento e Altera Nome de Pessoa respectivamente. Separando estas opções também haveria eliminação na entrada das dados pessoais, condensando-se algumas opções.

Este menu apresentado poderia ser resumido da seguinte maneira:

1. Dados pessoal: Entrada para informações pessoal.
 2. Documentação: Entrada das informações de documentação do funcionário.
-

A opção 1.2, CONSULTA traz para o operador a seguintes informações do sistema:

Nome: CPF:
Apelido: Matrícula:
Sexo: Pis/Pasep:
Nome do pai:
Nome da mãe: Órgão:
Nascimento: Local:
RG Militar: UF: Data: Órgão:
Estado civil: Cônjuge:
Cartório: Número registro: Livro:
Folha: Data de expedição:
Endereço:
Fone:

Nesta opção observa-se a mesma incoerência de informações. Existe uma mistura de informações de pessoal com informações de documentação. Através da adequação do menu principal para duas opções denominadas Dados Pessoal e Documentação, esta questão seria resolvida eliminando, desta maneira, a extensa lista de opções na opção Pessoa Física.

A opção 1.3, ATUALIZA ENDEREÇO, traz dados relativos ao endereço do funcionário. Esta opção também passa informações ao sistema através de uma opção única referente a informações dos dados pessoais.

A opção 1.4, CASAMENTO, traz informações do cartório, número da certidão, livro, folha, data, município, U.F. e nome da esposa. Esta opção oferece condições de inclusão, enquanto que a opção 1.16, ATUALIZA CERTIDÃO DE CASAMENTO, oferece condições de atualização. Estas opções também poderiam ser condensadas em uma única.

A opção 1.6, ÓBITO, traz as seguintes informações:

Cartório:	Número certidão:	
Livro:	Folha:	
Data óbito:	Município:	U.F:
Data da expedição:		
Causa Mortis:		

A opção acima também poderia ter sido alimentada através de uma opção de DOCUMENTAÇÃO, pois refere-se a dados documentais de uma determinada pessoa.

As opções 1.8 e 1.9 ATUALIZA e CONSULTA FORMAÇÃO PROFISSIONAL, são desnecessárias em seu desdobramento, pois o operador

poderá incluir, alterar, excluir e consultar dados funcionais através de uma única tela. No atual sistema, os cursos e profissões são retornados em forma de códigos o que se faz necessário que o operador tenha em mãos sempre uma tabela. O sistema tem que oferecer condições de “display” nominal dos cursos por funcionário ao ser passada para ele o seu registro geral.

Existe uma opção no atual sistema denominado MANUTENÇÃO DE DOCUMENTAÇÃO, a qual também é desnecessária, tendo em vista a adoção de uma opção denominada DOCUMENTAÇÃO, através da qual se farão as operações de inclusão, alteração, consulta e exclusão, após a confirmação do operador ao sistema de gerenciamento das informações do funcionário.

Existem opções para se trabalhar com informações referentes a identificação funcional, dependentes, efetivo e relatórios. Quando da inclusão de funcionários o sistema deve ser alimentado com informações da sua situação, tipo, condição e data de ingresso. Também deve se informar o seu cargo, nível, referência, qualificação e motivo, além do ato que autorizou esta inclusão. Depois vêm os dados relativos a informações de PIS/PASEP, Previdência e outras. Ao oferecer esta opção o operador necessariamente deve trabalhar com uma tabela nas mãos. Alguns cargos, níveis, referências e qualificações devem de ser cadastrados em tabelas definidas pelo sistema, pois isto facilita o trabalho tanto do operador como do consultor do sistema.

A opção para mudança de cargo existe como uma opção. Ele deveria estar subordinado a uma opção principal de dados pessoal com condições de gerar um histórico de cargas por funcionário eliminando, desta forma, uma segunda opção somente para o histórico dos cargos ocupados pelo funcionário.

Na opção para se efetivar movimentação o sistema informa a origem e o destino. Esta opção também poderia estar subordinada às informações gerais do

funcionário e não como opção independente. Esta mesma observação poderia também ser aplicada para a classificação no interesse do serviço, alteração e histórico de funções e alteração de identificação. Esta última opção constante no sistema, ela é tão desnecessária que na própria inclusão de informações gerais do funcionário pode se realizar qualquer alteração referente aos dados funcionais, desde que na sua implementação sejam previstas as opções de Inclusão, Alteração, Consulta e Exclusão.

A opção para inclusão de averbação de tempo de serviço e afastamento devem ser opções independentes tal como constam no sistema. A observação a ser feita é que não há necessidade de recolocar uma opção para registrar o retorno do afastamento, pois, na própria opção AFASTAMENTO, poderia constar um campo para registrar o retorno do funcionário, isto maximiza o retorno, pois se pedirmos um relatório de funcionários de determinada unidade a leitura será feita em um arquivo apenas e não em dois. Se quando da leitura a data do sistema for igual ou menor à data de término de afastamento este funcionário seria listado e caso contrário, se data maior que data do sistema, ele entenderia que o afastamento não foi findado logo ele não seria relacionado na geração do relatório.

A geração do quadro de efetivo é importante na situação não interativa de se extrair informações de um banco de dados. Para quem acessa o sistema pela primeira vez terá dificuldades em listar as informações devido a ausência de um ambiente mais amigável de se operar o sistema, pois o órgão, fixação, exercício, cargo, função e publicação depende de combinações de códigos de tabelas, as quais deverão estar nas mãos do operador, isto traz para o administrador uma dificuldade de se utilizar esta ferramenta para trabalhar o computador.

6. ANÁLISE DA SITUAÇÃO ATUAL

O atual sistema de pessoal da Polícia Militar do Estado de Goiás, **Administração Global e Integrada, AGI**, foi desenvolvido pela EMCIDEC, quando ainda se denominava CODEG, nos idos de 1985.

A empresa responsável pelo seu desenvolvimento no decorrer desses quase 12 anos passou por transformações técnicas, políticas e econômicas que afetaram a performance dos seus objetivos iniciais, além de ser uma empresa que não se dedicou exclusivamente às questões de informática do Estado de Goiás. Com isto, empresas como a CELG, SANEAGO, Secretaria da Fazenda, Tribunal de Justiça e outras que necessitam do gerenciamento eficaz de suas informações para atenderem com eficiência os seus usuários migraram para sistemas próprios, adquirindo equipamentos e contratando pessoal especializado.

Atualmente, o grande usuário da EMCIDEC é o DETRAN - GO e junto com ele a Polícia Militar. É este talvez o único motivo de existência da EMCIDEC. No dia em que o DETRAN - GO começar a gerenciar suas próprias informações a Polícia Militar sentirá dificuldades em manter suas informações nos computadores da EMCIDEC, pois o DETRAN - GO é o órgão que auxilia a EMCIDEC junto à manutenção oferecida pela IBM. Isto será péssimo para a Polícia Militar, pois, além de se ter atualmente um sistema defasado, ela o

acabará perdendo. Daí então a necessidade de se criar, o mais rápido possível, o seu centro de processamento.

Além dessa situação política/financeira, existe ainda a questão da evasão da mão-de-obra técnica desta empresa e o próprio Plano de Demissão Voluntária, PDV, do governo, que esvaziou os poucos técnicos que lá restavam. Isto reflete na manutenção do sistema. Ela já não existe há anos. Se considerarmos que na vida útil de um sistema, 70% se destina à sua manutenção e 30% à sua feitura o nosso atual sistema de pessoal já morreu, faltando apenas ser sepultado.

As grandes e pequenas corporações do mundo moderno há mais ou menos uma década migraram do ambiente DOS para o ambiente Windows, ditando a nova plataforma de gerenciamento e maneira interativa de se criar ou administrar informações, através dos diversos produtos desenvolvidos pelas empresas desenvolvedoras de softwares, seguindo a filosofia da Microsoft. O grande porte vem pouco a pouco sendo substituído através de um processo mundialmente conhecido como “dowsizing” ou seja, a sua substituição por máquinas pequenas, porém mais potentes e já preparadas para o ambiente Windows, tendência irreversível do mundo moderno.

Esta introdução à análise da situação atual serve para sentirmos a quão relegado a planos outros está o nosso sistema e a empresa que é responsável pela sua manutenção e implementação de novos módulos.

O sistema roda atualmente em uma plataforma de grande porte. O ambiente de trabalho não condiz com a atual tendência de se programar e trabalhar com informações de bancos de dados. A questão da conectividade com novas filosofias de gerenciadores de informações também se torna difícil, tendo em vista o ambiente para o qual foi desenvolvido, o DOS.

O mais crítico deste atual sistema talvez não seja tanto a questão de interagir ou não com os microcomputadores que rodam os aplicativos para o ambiente Windows, mas sim a questão da manutenção e implementação de novos módulos necessários ao prolongamento da vida útil do sistema.

O ponto mais crucial da atualização dos dados de pessoal é que a sua implementação é centralizada, o que faz com que um fato gerador de informação na vida funcional de um funcionário, demore mais de três meses para ser passado ao sistema. Isto porque, além de ser desprovido da manutenção pelas questões anteriormente citadas, não existe uma rede de gerenciamento das operações do sistema, que gere a informação instantânea.

Os módulos de entrada e saída de informações do sistema deveriam sofrer uma alteração em seu *layout* para se facilitar a aplicabilidade dessa entrada e saída, pois se uma manutenção fosse feita neste sentido, seguramente os menus diversos que compõem o sistema seriam alterados e adequados a uma interface menos carregada.

Da análise da situação atual conclui-se que a base de dados atualmente existente é rica em informações faltando apenas a aplicação de novas ferramentas de extração das combinações dessas informações e verificar as possibilidades de conectividade do atual banco de dados com bancos de dados mais interativos desenvolvidos para a nova filosofia de trabalho necessária à atualização das necessidades de gerenciamento de informações.

7. SITUAÇÃO PROPOSTA

Nesta etapa do trabalho passo a apresentar a situação proposta, que é o desenvolvimento da programação para a implementação do **Sistema de Gerenciamento de Pessoal**.

Como já foi anteriormente mencionado, fica entendida esta parte do trabalho monográfico como sendo uma apresentação de um software desenvolvido no banco de dados Visual Basic, para o ambiente Windows, o qual se propõe a gerenciar informações de pessoal.

O presente aplicação foi desenvolvida para fins monográficos, para ambiente monousuário devendo, se adotado, ser adaptado para o sistema de rede, preservando toda a sua estrutura básica. Tem este software, como todo os outros, aqui a sua primeira versão para teste em microcomputadores de unidades isoladas. A sua adaptação para a plataforma mais abrangente, adoção da filosofia de redes, sistema aberto e outras inovações ficarão a cargo de versões posteriores.

Este sistema compõe basicamente das apresentadas nas páginas seguintes.

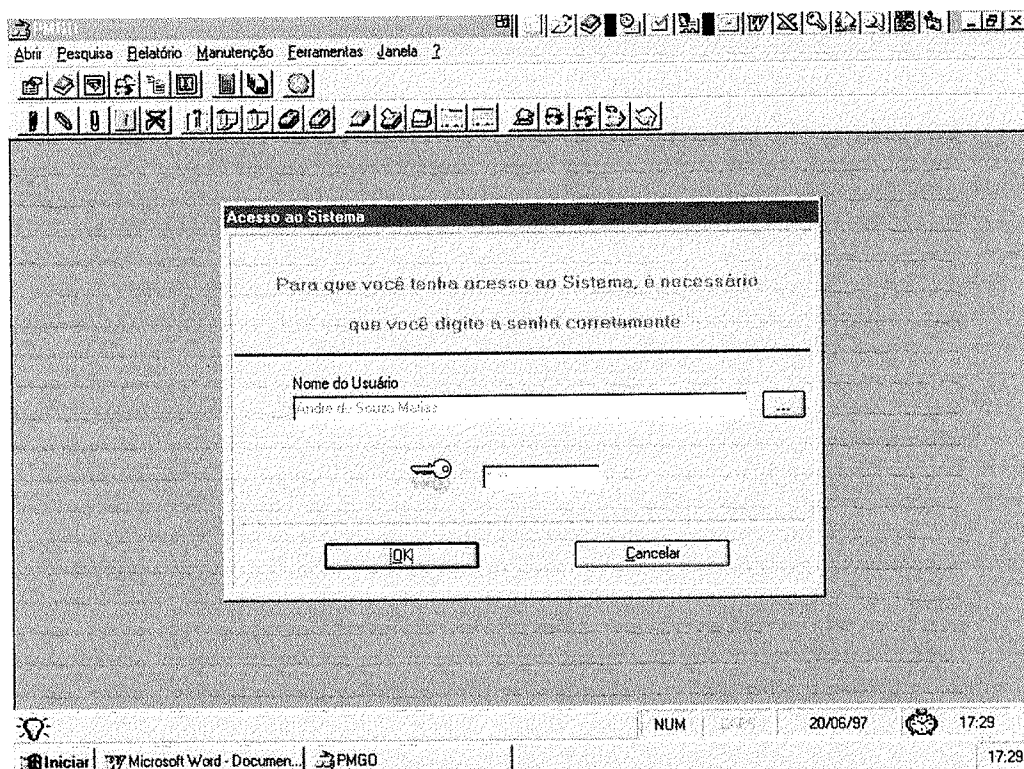
APRESENTAÇÃO



Esta primeira tela é a tela de apresentação do sistema. Pode-se notar a versão do software, versão 1.0, como também a organização para a qual ele foi desenvolvido.

À medida em que se fizer novas alterações no sistema, a versão mudará em função das manutenções às quais quaisquer sistema de computação estão sujeitos.

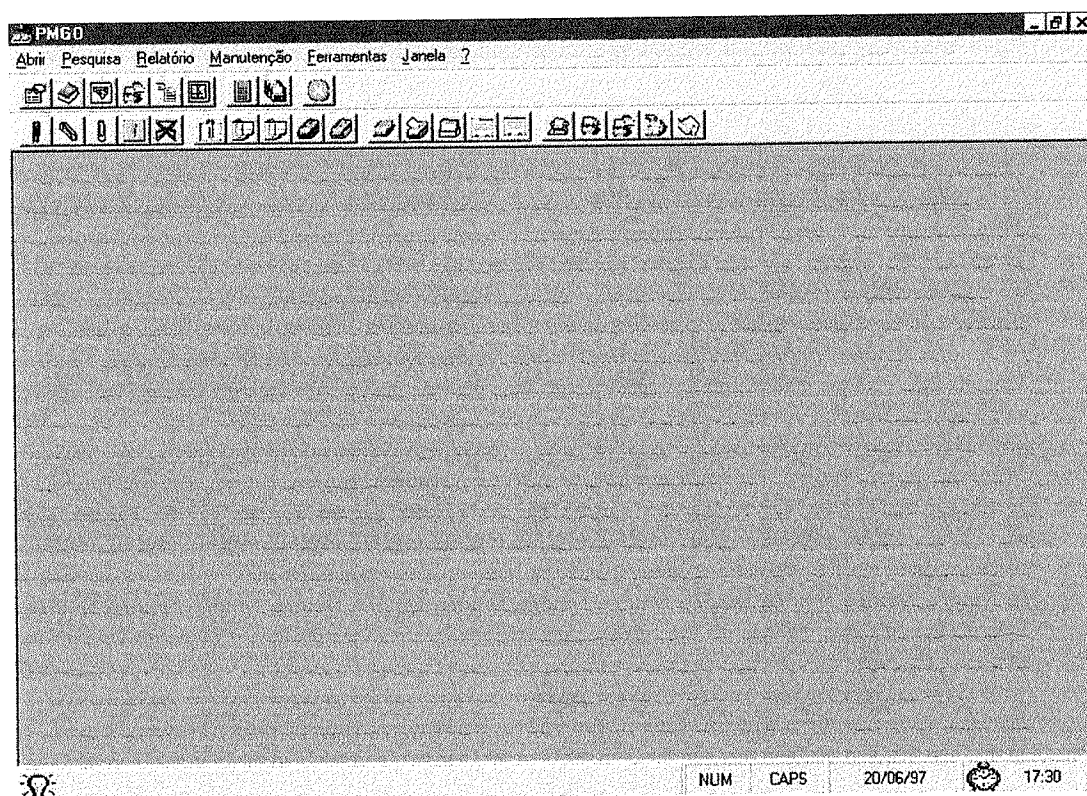
ABERTURA



A tela de abertura do sistema mostra as opções do software. Tem-se a opção Abrir, Relatório, Manutenção, Ferramentas, Janela e Ajuda(?).

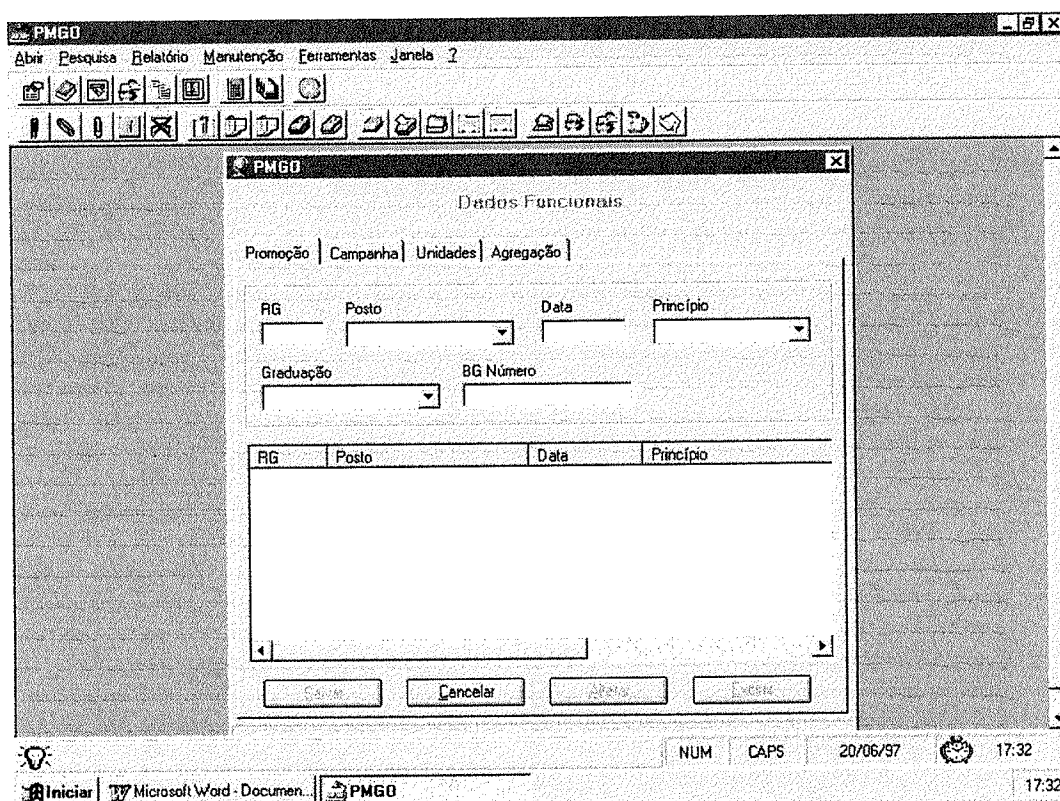
O operador só terá acesso ao sistema se o mesmo for cadastrado. Para tanto, ele deverá entrar com o seu nome o qual será lido no arquivo dos usuários cadastrados além de ter que informar a senha, a qual estará, obviamente, indexada ao seu nome. Quando do cadastro de um usuário no sistema, o administrador do banco de dados definirá as operações que este operador poderá realizar no sistema.

TELA PRINCIPAL



Após informar a senha combinada com o nome ao sistema, o operador cadastrado terá acesso às operações do sistema, com todas as opções definidas.

DADOS FUNCIONAIS - PROMOÇÃO

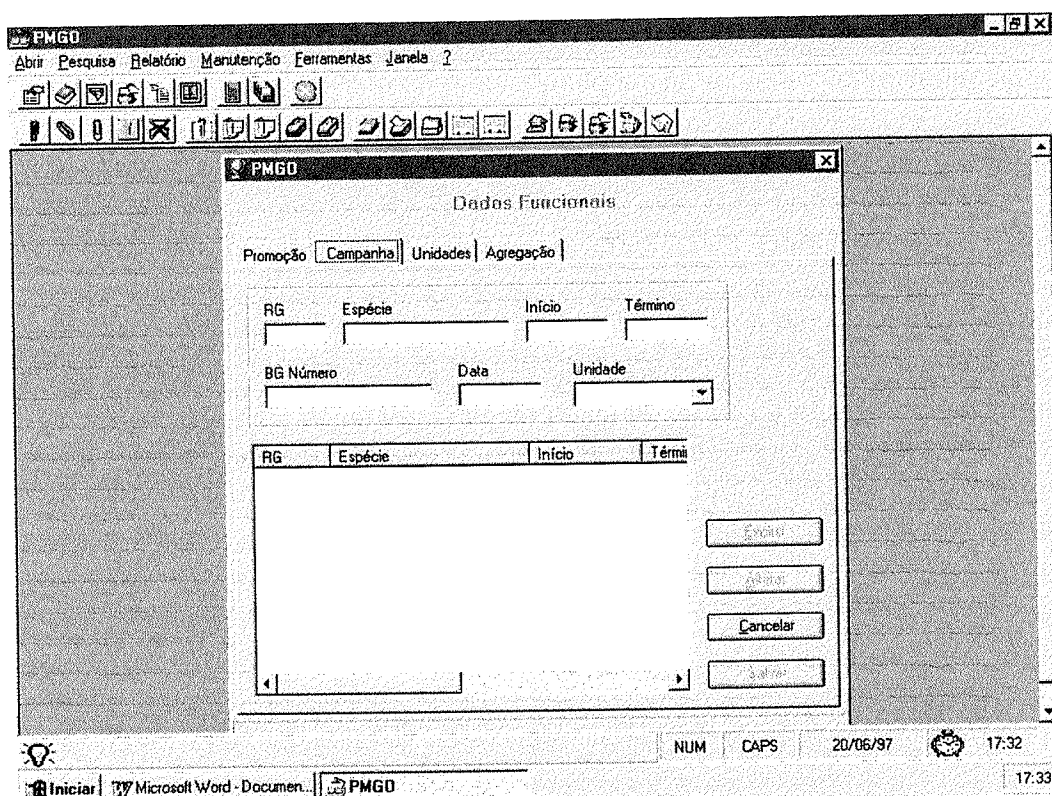


A tela de Dados Funcionais permite a entrada de 4 informações:

- 1- Promoção
- 2- Campanha
- 3- Unidade
- 4- Agregação

Na opção Promoção, após entrar com o RG do Militar, será então gerado um histórico de suas promoções com a data da promoção, princípio e número do RG da alteração do Posto ou Gradação.

DADOS FUNCIONAIS - CAMPANHA



Na opção campanha, será gerado um histórico do tempo de campanha do Policial Militar onde serão registrados a espécie, o início e o término e o Boletim Geral que a publicou, como também a unidade do militar para o qual se está efetuando o registro.

DADOS FUNCIONAIS - AGREGAÇÃO

The screenshot shows a Windows-style application window titled 'PMGO'. The menu bar includes 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela'. The toolbar contains various icons for file operations and editing. The main window displays a dialog box titled 'Dados Funcionais' with a tabbed interface. The 'Agregação' tab is selected. The dialog contains several input fields for data entry:

RG	BG Número	Data	Início	Término
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Below the input fields is a table with a header row and an empty body:

RG	BG Número	Data	Início	Término

At the bottom of the dialog are four buttons: 'Salvar', 'Cancelar', 'Ajuda', and 'Fechar'. The Windows taskbar at the bottom shows the system tray with 'NUM', 'CAPS', the date '20/06/97', and the time '17:32'. The taskbar also shows the 'Iniciar' button and open applications: 'Microsoft Word - Documen...' and 'PMGO'.

Através desta tela se registra o histórico das agregações do militar gerando, desta forma, um histórico das suas agregações. Deve ser informado o Boletim Geral que o agregou, a data, o início e o término quando este retornar, bem como o destino e o tipo de agregação.

DADOS GERAIS

PMGO

Abrir Pesquisa Relatório Manutenção Ferramentas Janela 2

Dados Gerais

Comissões | Funções | Empregos | Afastamentos | Movimentação
Elogios | Punições | Condenações | Medalhas | Condecorações

RG BG Número Data Unidade

BI Número Data Unidade

Texto

RG	BG Número	Data	Unidade

Salvar Cancelar Atualizar Excluir

NUM CAPS 20/06/97 17:34

Iniciar Microsoft Word - Documen... PMGO 17:34

A entrada para dados gerais é a mais abrangente de todas as opções do sistema. Através dela se permite a entrada de informações de comissões, funções, empregos, afastamentos, movimentação, elogios, punições, condenações, medalhas e condecorações.

DADOS GERAIS - ELOGIOS/PUNIÇÕES

The screenshot shows a window titled 'PMGO' with a menu bar containing 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela 2'. Below the menu is a toolbar with various icons. The main area is titled 'Dados Gerais' and contains a menu with options: 'Comissões', 'Funções', 'Empregos', 'Afastamentos', 'Movimentação', 'Elogios', 'Punições', 'Condenações', 'Medalhas', and 'Condecorações'. The form includes several input fields: 'RG', 'BG Número', 'Data', 'BI Número', 'Data', and 'Unidade'. There is a large 'Texto' area for entering details. Below the form is a table with columns 'RG', 'BG Número', 'Data', and 'BI Número'. At the bottom of the form are buttons for 'Cancelar', 'Inserir', and 'Excluir'. The taskbar at the bottom shows 'Iniciar', 'Microsoft Word - Documen...', and 'PMGO'. The system tray displays 'NUM', 'CAPS', '20/06/97', and '17:34'.

Através desta tela permite-se a entrada de dados referentes aos elogios ou às punições do Policial Militar. Após ser informado o Boletim Geral ou o Boletim Interno pode se inserir o texto da referência elogiosa ou a punição. Quando se pedir um relatório desses históricos essas alterações serão lançadas.

DADOS GERAIS - CONDENAÇÕES

The screenshot shows a software window titled 'PMGO' with a menu bar containing 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela 2'. Below the menu is a toolbar with various icons. The main area is titled 'Dados Gerais' and contains a menu with options: 'Comissões', 'Funções', 'Empregos', 'Afastamentos', 'Movimentação', 'Elogios', 'Punições', 'Condenações', 'Medalhas', and 'Condecorações'. The 'Condenações' option is selected. Below the menu are input fields for 'RG', 'BG Número', and 'Data'. Below these are fields for 'BI Número', 'Data', and 'Unidade'. A table with columns 'RG', 'BG Número', and 'Data' is present, but it is empty. To the right of the table are four buttons: 'Incluir', 'Excluir', 'Cancelar', and 'Salvar'. The Windows taskbar at the bottom shows 'Iniciar', 'Microsoft Word - Documen...', 'PMGO', 'NUM', 'CAPS', '20/06/97', and '17:34'.

A exemplo da tela anterior, a presente tela recebe informações de todas as condenações havidas na vida do Policial Militar. A diferença é que nesta opção ficam registrados os números dos documentos que publicaram estas condenações enquanto que na opção anterior, por se tratar de um texto punitivo ou de elogio ele poderá ser transcrito na íntegra.

DADOS GERAIS - MEDALHAS

PMGO

Abrir Pesquisa Relatório Manutenção Ferramentas Janela 2

Comissões | Funções | Empregos | Afastamentos | Movimentação
Elogios | Punições | Condenações | Medalhas | Condecorações

RG BG Número Data Tipo

RG	BG Número	Data	Tipo
----	-----------	------	------

Cancelar Incluir Excluir Atualizar

NUM CAPS 20/06/97 17:34

Iniciar Microsoft Word - Documen... PMGO 17:35

Através desta opção ficam cadastrados os dados referentes às medalhas concedidas ao Policial Militar, sendo gerado, por ordem cronológica, um histórico de suas aquisições.

DADOS GERAIS - CONDECORAÇÕES

The screenshot displays the 'Dados Gerais' window in the PMGD application. The window title is 'PMGD' and the menu bar includes 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela ?'. The toolbar contains various icons for file operations and data management. The main area is titled 'Dados Gerais' and features a tabbed interface with the following tabs: 'Comissões', 'Funções', 'Empregos', 'Afastamentos', 'Movimentação', 'Elogios', 'Punições', 'Condenações', 'Medalhas', and 'Condecorações'. The 'Condecorações' tab is active. The form contains the following fields:

- RG:
- BG Número:
- Data:
- BI Número:
- Data:
- Unidade:
- Tipo:

Below the form is a table with the following columns: RG, BG Número, Data, and BI Número. The table is currently empty. At the bottom of the form are buttons for 'Cancelar', 'OK', and 'Fechar'. The taskbar at the bottom shows the system tray with 'NUM', 'CAPS', the date '20/06/97', and the time '17:34'. The taskbar also shows the 'Iniciar' button and open applications: 'Microsoft Word - Documen...' and 'PMGD'.

Em condecorações, ficam registrados dados do Boletim Geral e Boletim Interno, a data, a unidade e o tipo de condecoração recebida pelo Policial Militar, podendo também ser gerado um relatório pela ordem cronológica dessas condecorações.

DADOS GERAIS - COMISSÕES

The screenshot displays the 'Dados Gerais' window in the PMGO application. The window title is 'PMGO' and the menu bar includes 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela 2'. The main area contains a form with the following fields:

- Elogios** | **Punições** | **Condenações** | **Medalhas** | **Condecorações**
- Comissões.** | **Funções** | **Empregos** | **Afastamentos** | **Movimentação**
- RG** (text field)
- Espécie** (text field)
- Início** (text field)
- Término** (text field)
- BG Número** (text field)
- Data** (text field)
- BI Número** (text field)
- Data** (text field)
- Unidade** (dropdown menu)

Below the form is a table with the following columns: **RG**, **Espécie**, **Início**, **Término**, and **B**. The table is currently empty. To the right of the table are four buttons: **Salvar**, **Atualizar**, **Cancelar**, and **Limpar**.

The taskbar at the bottom shows the system date as 20/06/97 and the time as 17:34. The taskbar also shows the 'Iniciar' button and the 'PMGO' application icon.

As comissões exercidas pelo Policial Militar ficam registradas através destas informações. Deve-se informar a espécie de comissão, o início e o término, e a unidade além dos documentos que a publicaram. Através desta opção o Policial Militar terá registrada todas as comissões exercidas ao longo de sua carreira.

DADOS GERAIS - FUNÇÕES

The screenshot shows a software window titled 'PMGD' with a menu bar (Abrir, Pesquisa, Relatório, Manutenção, Ferramentas, Janela ?) and a toolbar. The main area is titled 'Dados Gerais' and contains several tabs: Elogios, Funções, Condenações, Medalhas, Condecorações, Comissões, Empregos, Afastamentos, and Movimentação. The 'Funções' tab is active.

The form contains the following fields:

- RG: []
- Função: []
- Atividade: []
- Início: []
- Término: []
- BG Número: []
- Data: []
- BI Número: []
- Data: []
- Unidade: []

Below these fields is a table with the following header:

RG	Função	Atividade	Início	Término

At the bottom of the form are buttons for 'OK', 'Cancelar', 'Ajuda', and 'Emissor'. The Windows taskbar at the bottom shows the date 20/06/97 and time 17:34, along with open applications like 'Microsoft Word - Documen...' and 'PMGD'.

As funções exercidas pelo Policial Militar ficam registradas através de informações da função, da atividade, o seu início e término, e a unidade, bem como os documentos legais que a publicou. Para cada função existirá um registro, o que permitirá gerar um relatório do histórico das funções exercidas.

DADOS GERAIS - EMPREGOS

Dados Gerais

Elogios | Punições | Condenações | Medalhas | Condecorações
Comissões | Funções | **Empregos** | Afastamentos | Movimentação

RG Emprego Atividade
 Início Término BG Número Data
 BI Número Data Unidade

RG	Emprego	Atividade	Início

NUM CAPS 20/06/97 17:34

Iniciar Microsoft Word - Documen... PMGD 17:36

Os empregos ficarão registrados através do local de empregos, a atividade, início e término e unidade, como também o número e data do Boletim Geral ou Boletim Interno que determinou o emprego.

DADOS GERAIS - AFASTAMENTOS

PMGO

Abrir Pesquisa Relatório Manutenção Ferramentas Janela 2

Dados Gerais

Elogios | Punições | Condenações | Medalhas | Condecorações
Comissões | Funções | Empregos | Afastamentos | Movimentação

RG Tipo Início Término

BG Número Data

BI Número Data Unidade

RG	Tipo	Início	Término

Salvar Cancelar Adicionar Excluir

NUM CAPS 20/06/97 17:34

Iniciar Microsoft Word - Documen... PMGO 17:36

Para todos os afastamentos do Policial Militar ficam registrados, através da geração de seu relatório quando se passa ao sistema informações de sua unidade, Boletim Geral ou interno que o publicou, e a data de sua apresentação. Na parte inferior da tela, ficarão disponíveis informações sobre esses atos que concretizaram essas movimentações.

MOVIMENTAÇÃO

PMGO

Arquivo Pesquisa Relatório Manutenção Ferramentas Janela 2

Elogios Punições Condenações Medalhas Condecorações
Comissões Funções Empregos Afastamentos **Movimentação**

RG Unidade
BG Número Data Apresentação
BI Número Data Unidade Apresentação
Início Término

RG	Unidade	BG Número
----	---------	-----------

Excluir
Adicionar
Cancelar
Salvar

NUM CAPS 20/06/97 17:34

Iniciar Microsoft Word - Documen... PMGO 17:37

As movimentações do Policial Militar ficam registradas através da geração de seu histórico quando se passa ao sistema informações de sua unidade, Boletim Geral ou interno que a publicou, e a data de sua apresentação. Na parte inferior da tela, ficarão disponíveis informações sobre esses atos que concretizaram essas movimentações.

DADO PESSOAL

The screenshot displays a software application window titled "PMGO" with a menu bar containing "Abrir", "Pesquisa", "Relatório", "Manutenção", "Ferramentas", and "Janela 2". Below the menu is a toolbar with various icons. The main window area is titled "Dados Pessoais" and contains the following fields:

- RG:
- Nome:
- Nome do Pai:
- Nome da Mãe:
- Estado Civil:
- Data de Nascimento:
- Estado:
- Município:
- Distrito:
- Altura:
- Cor:
- Cabelo:
- Olhos:
- Endereço:
- Barrio:
- Mobilizado:
- Engajado:
- Reengajado:
- Exclusão:
- Motivo:
- Posto:
- Graduação:

On the right side of the form, there is a placeholder box for a photograph and a button labeled "Visualizar". The Windows taskbar at the bottom shows the system tray with the date "20/06/97", the time "17:46", and the taskbar itself shows "Iniciar", "Microsoft Word - TELAS.doc", and "PMGO".

Os dados pessoal do Policial Militar constam todos nesta tela, inclusive a sua fotografia, bastando para isso clicar com o cursor do mouse na opção visualizar e relacionar o RG no arquivo de imagens associado à ficha individual. Esta opção traz informações de filiação, endereço, estado civil, data de nascimento, função atual, inclusão, dentre outras.

DEPENDENTES

PMGO

Abir Pesquisa Relatório Manutenção Ferramentas Janela 2

PMGO

Dependentes e Cursos

Dependentes | Cursos

RG Nome Sexo

Data de Nascimento Cidade Estado

Certidão de Nascimento Livro Folha Data Comarca

RG	Nome	Sexo	Data de Nascimento
----	------	------	--------------------

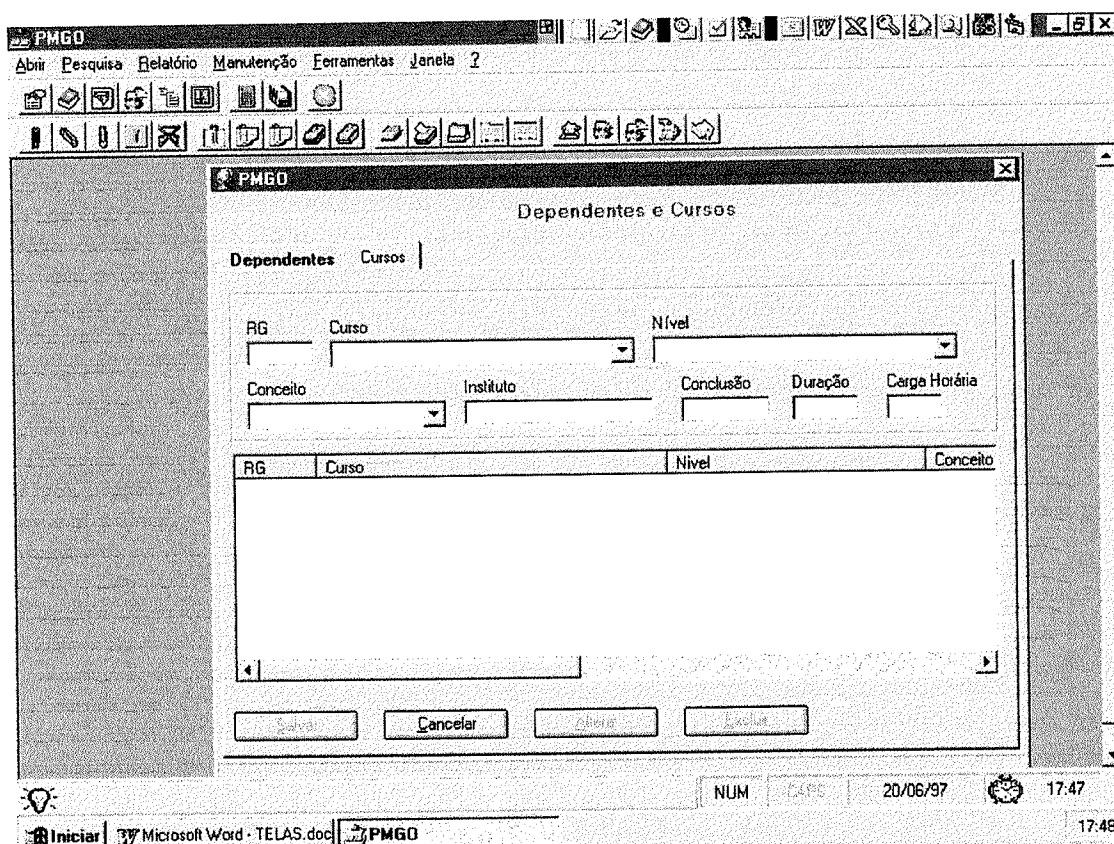
Salvar Cancelar Adicionar Excluir

NUM DATA 20/06/97 17:47

Iniciar Microsoft Word - TELAS.doc PMGO 17:48

Os dados dos dependentes entrarão no sistema através da presente tela. Deve informar o seu nome, sexo, grau de parentesco, data de nascimento, cidade e informações gerais do documento oficial que comprove a informação.

CURSOS



Os cursos que o Policial Militar possuir serão anotados através desta opção. Nele se informará o nome do curso, o nível (se secundário, técnico, superior, etc) o conceito, o nome do instituto de realização, além do ano de conclusão, a duração e a sua carga horária.

DOCUMENTAÇÃO

Documentação

RG Orgão Estado

RG Civil Orgão Estado CPF

Título Eleitoral Cidade Zona Seção

Habilitação Categoria Registro Validade Reservista Força

Categoria Passaporte Validade IPASGO

Certidão de Casamento Livro Folha Data Comarca

Certidão de Nascimento Livro Folha Data Comarca

RG	Orgão	Estado	RG Civil	Orgão	Estado	CPF

NUM 20/06/97 17:47

Iniciar Microsoft Word - TELAS.doc PMGO 17:48

Através desta opção o sistema recebe todas as informações referentes à documentação do Policial Militar. A consulta poderá ser feita através de uma opção geral, onde se deverá relacionar no campo RG, o RG que se quer ter informações de sua documentação.

TEMPO DE SERVIÇO - CONSULTA

PMGD

Abrir Pesquisa Relatório Manutenção Ferramentas Janela 2

Tempo de Serviço

Consulta | Averbação

RG Ativo Afastamento Averbado
Dias Dias

Agregado Total
Dias Dias

RG	Ativo	Afastamento	Averbado	Agregado
----	-------	-------------	----------	----------

Ok Cancelar Fechar

NUM CAPS 20/06/97 17:49

Iniciar Microsoft Word - TELAS.doc PMGD 17:49

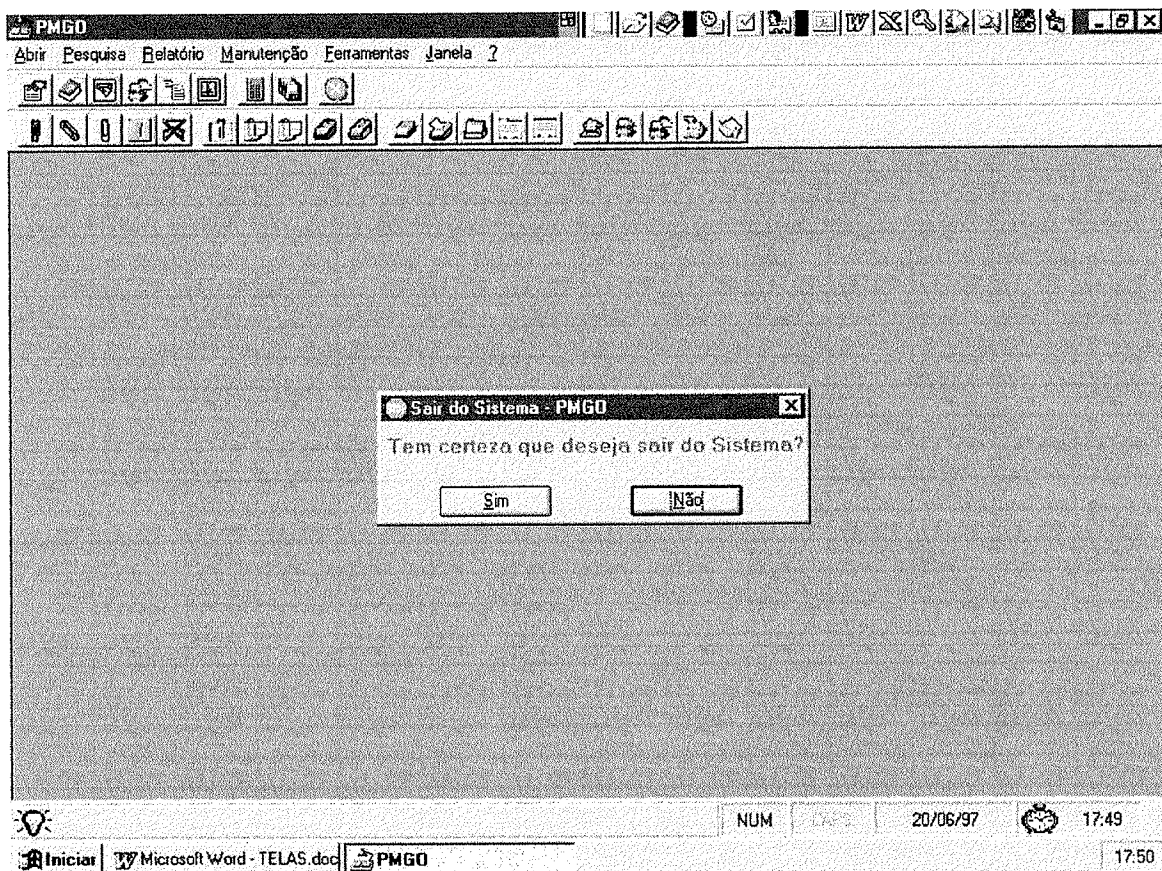
Através da data de inclusão em Dados Pessoal e as alterações efetivadas através dos lançamentos de afastamentos, agregações e tempo averbado, pode se consultar o tempo de serviço do Policial Militar, bastando para isto informar o seu RG ou relacioná-lo a partir da lista de RG no porte inferior da tabela.

TEMPO DE SERVIÇO - AVERBAÇÃO

The screenshot displays the 'Tempo de Serviço' (Service Time) entry form within the PMGO application. The form is titled 'Tempo de Serviço' and has a 'Consulta' dropdown menu set to 'Averbação'. The form contains several input fields: 'RG', 'Origem', 'Início', 'Término', 'BG Número', 'Data', and 'Tempo' (with sub-fields for 'Dias', 'Mês', and 'Anos'). Below the form is a table with the following columns: 'RG', 'Origem', 'Início', 'Término', and 'BGNúmero'. The table is currently empty. At the bottom of the form are buttons for 'Salvar', 'Cancelar', 'Limpar', and 'Ler'. The application's menu bar includes 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', and 'Ferramentas'. The taskbar at the bottom shows the date '20/06/97' and the time '17:49'.

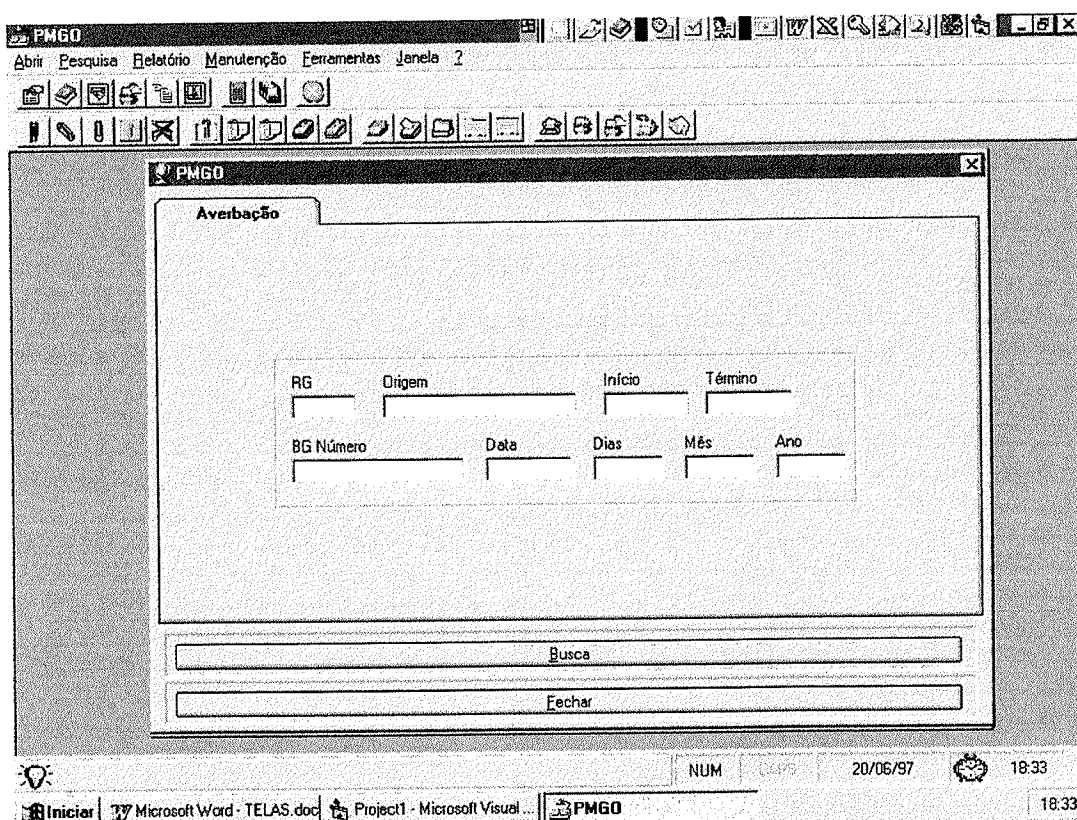
A averbação de tempo de serviço é realizada através desta opção, onde se insere a origem, o início e o término da atividade, o documento que a publicou e a data e em seguida o sistema calcula o total de anos, meses e dias averbados para cada averbação incluída.

TELA DE SAÍDA DO SISTEMA



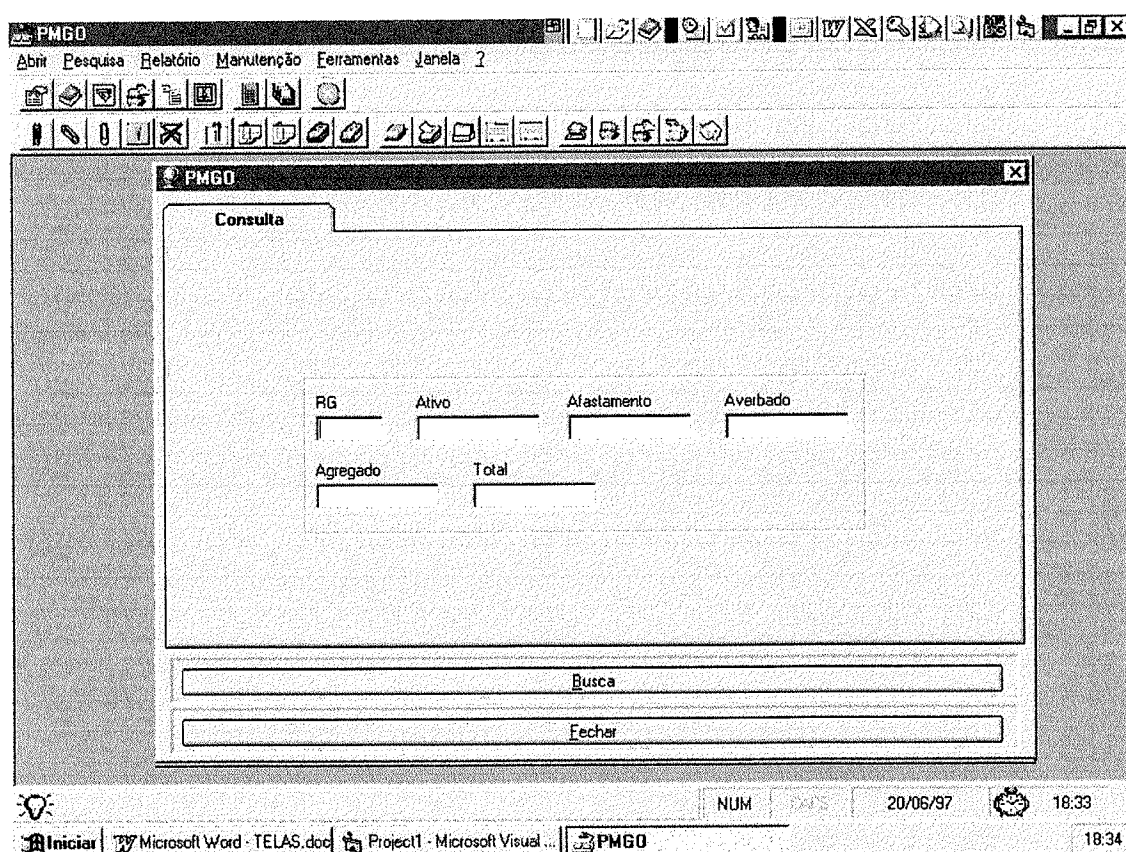
Esta tela como ela mesma se explica, é uma subopção da opção abrir, para se sair do sistema.

PESQUISA - AVERBAÇÃO



Esta opção de averbação, é uma subopção da opção que se encontra no menu pesquisa. Ela oferece, após se informar o RG, informações de tempos averbados. Se o operador clicar com o ponteiro do mouse a opção Busca, ele então trará os RG registrados no sistema, onde se poderá relacionar qualquer um deles.

PESQUISA - CONSULTA TEMPO DE SERVIÇO



Esta opção funciona com as mesmas características da anterior. Enquanto aquela informava as averbações registradas, esta informa o tempo ativo, afastado, averbado e agregado e traz o tempo total de serviço. Pode se informar um RG ou buscar um específico através da opção de busca.

PESQUISA - DOCUMENTAÇÃO

The screenshot displays the PMGO software interface. At the top, there is a menu bar with options: 'Abriu Pesquisa Relatório Manutenção Ferramentas Janela 2'. Below the menu is a toolbar with various icons. The main window is titled 'PMGO' and contains a 'Documentação' section. This section has a search form with the following fields:

- RG
- Orgão
- Estado
- RG Civil
- Orgão
- Estado
- CPF
- Título Eleitoral
- Cidade
- Zona
- Seção
- Habilitação
- Categoria
- Registro
- Validade
- Reservista
- Força
- Categoria
- Passaporte
- Validade
- IPASGD

At the bottom of the form, there are two buttons: 'Busca' and 'Fecha'. The Windows taskbar at the bottom shows the date '20/06/97' and the time '18:33'. The taskbar also displays several open applications: 'Iniciar', 'Microsoft Word - TELAS.doc', 'Project1 - Microsoft Visual ...', and 'PMGO'.

Através desta opção pode se informar o RG e o sistema trará os cursos um a um, referentes ao RG. Pode-se também relacionar o RG através da opção Busca. A diferença consiste que se efetuar a pesquisa pela Busca, o operador terá todos os RG existentes na base de dados dispostos por uma ordem crescente.

PESQUISAS - CURSOS

The screenshot displays the PMGO software interface. At the top, there is a menu bar with options: 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela 2'. Below the menu is a toolbar with various icons. The main window is titled 'PMGO' and contains a search form for 'Cursos'. The form has the following fields:

RG	Curso	Nível
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

Below the form are two buttons: 'Busca' and 'Fechar'. The taskbar at the bottom shows the system tray with the date '20/06/97' and time '18:33'. The taskbar also displays the 'Iniciar' button and several open applications: 'Microsoft Word - TELAS.doc', 'Project1 - Microsoft Visual...', and 'PMGO'.

Os cursos do Policial Militar ou dos Policiais Militares poderão ser pesquisados através desta opção. Após pedida as informações o sistema trará o nome do curso, o nível e outras informações básicas.

PESQUISA - DEPENDENTES

The screenshot displays the PMGO application window with a menu bar (Abrir, Pesquisa, Relatório, Manutenção, Ferramentas, Janela ?) and a toolbar. The main window is titled 'PMGO' and contains a 'Dependentes' section. This section features a search form with the following fields:

RG	Nome	Sexo		
Data de Nascimento	Cidade	Estado		
Certidão de Nascimento	Livro	Folha	Data	Comarca

Below the form are two buttons: 'Busca' and 'Fechar'. The Windows taskbar at the bottom shows the system tray with 'NUM', '20/06/97', and '18:33', and the taskbar with 'Iniciar', 'Microsoft Word - TELAS.doc', 'Project1 - Microsoft Visual ...', and 'PMGO'.

Os dados dos dependentes também poderão ser apresentados através desta opção. Pode se informar o RG do Policial Militar ou efetuar esta pesquisa através da busca onde se tem todas as informações referentes ao dependente.

PESQUISA - DADOS PESSOAL

The screenshot displays a software application window titled "PMGO" with a menu bar containing "Abrir", "Pesquisa", "Relatório", "Manutenção", "Ferramentas", and "Janela 2". Below the menu bar is a toolbar with various icons. The main area of the window contains a form titled "Dados Pessoais" with the following fields:

- RG
- Nome
- Nome do Pai
- Nome da Mãe
- Estado Civil (dropdown menu)
- Data de Nascimento
- Estado (dropdown menu)
- Município
- Distrito
- Endereço
- Bairro

At the bottom of the form are two buttons: "Busca" and "Fechar". The Windows taskbar at the bottom shows the system tray with the date "20/06/97" and time "18:33". The taskbar also displays several open applications: "Iniciar", "Microsoft Word - TELAS.doc", "Project1 - Microsoft Visual ...", and "PMGO".

Uma pesquisa importante é a pesquisa de dados pessoais. As informações aqui apresentadas é uma síntese das informações cadastradas em dados pessoal, mais completa. Aqui temos informações de filiação, RG, nome, endereço, estado civil, data e local de nascimento. Também pode se pesquisar todas os Policiais Militares cadastrados na base.

PESQUISA - MOVIMENTAÇÃO

The screenshot displays the 'PMGO' application window with a menu bar containing 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela 2'. Below the menu is a toolbar with various icons. The main window area is titled 'Movimentação' and contains a search form with the following fields:

- RG
- Unidade
- RG Número
- Data
- Apresentação
- BI Número
- Data
- Unidade
- Apresentação
- Início
- Término

At the bottom of the form are two buttons: 'Busca' and 'Fechar'. The Windows taskbar at the bottom shows the system tray with the date '20/06/97', time '18:33', and the taskbar itself with 'Iniciar', 'Microsoft Word - TELAS.doc', 'Project1 - Microsoft Visual...', and 'PMGO'.

A movimentação do Policial Militar, através do menu pesquisa, traz um histórico de sua movimentação apresentando as unidades pelas quais ele passou, além das apresentações referentes aos boletins que as publicou, o seu início e término. Pode se pesquisar também através da opção busca.

PESQUISA - AFASTAMENTOS

The screenshot displays the PMGO software interface. At the top, there is a menu bar with options: 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela 2'. Below the menu is a toolbar with various icons. The main window is titled 'PMGO' and contains a sub-window titled 'Afastamentos'. This sub-window has a search form with the following fields:

RG	Tipo	Início	Término
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
BG Número <input type="text"/>		Data <input type="text"/>	
BI Número <input type="text"/>		Data <input type="text"/>	Unidade <input type="text"/>

Below the form are two buttons: 'Busca' and 'Fechar'. The taskbar at the bottom shows the system tray with the date '20/06/97' and time '18:33'. The taskbar also displays several open applications: 'Iniciar', 'Microsoft Word - TELAS.doc', 'Project1 - Microsoft Visual ...', and 'PMGO'.

Os afastamentos também poderão ser pesquisados através do mesmo menu. Esta opção traz os documentos que registraram o afastamento, o início e o término e a unidade à época de cada afastamento.

PESQUISA - EMPREGO

The screenshot shows a graphical user interface for a software application named PMGO. The main window is titled "Empregos" and contains a search form with the following fields:

RG	Emprego	Atividade	Início	Término
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Below the main table, there are additional search criteria:

BG Número	Data	
<input type="text"/>	<input type="text"/>	
BI Número	Data	Unidade
<input type="text"/>	<input type="text"/>	<input type="text"/>

At the bottom of the form, there are two buttons: "Busca" and "Fechar". The taskbar at the bottom of the screen shows the date 20/06/97 and the time 18:33. The taskbar also displays several open applications: "Iniciar", "Microsoft Word - TELAS.doc", "Project1 - Microsoft Visual ...", and "PMGO".

Através desta opção pode se pesquisar o emprego do Policial Militar. Ela registra, durante o decorrer de sua permanência na Polícia Militar, os seus empregos, a atividade, início e término, além de outras informações referentes à situação.

PESQUISA - FUNÇÕES

The screenshot displays the PMGO software interface. At the top, there is a menu bar with options: 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela 2'. Below the menu is a toolbar with various icons. The main window is titled 'PMGO' and contains a sub-window titled 'Funções'. Inside the 'Funções' window, there is a search form with the following fields:

RG	Função	Atividade	Início	Término
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
BG Número		Data		
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
BI Número		Data		Unidade
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Below the search form, there are two buttons: 'Busca' and 'Fechar'. The taskbar at the bottom shows the system tray with the date '20/06/97' and time '18:33'. The taskbar also displays several open applications: 'Iniciar', 'Microsoft Word - TELAS.doc', 'Project1 - Microsoft Visual...', and 'PMGO'.

A exemplo da opção anterior, esta opção realiza uma pesquisa nas funções de Policial Militar trazendo informações básicas desses registros tais como a própria função exercida, a atividade, início e término, além de outros.

PESQUISA - COMISSÕES

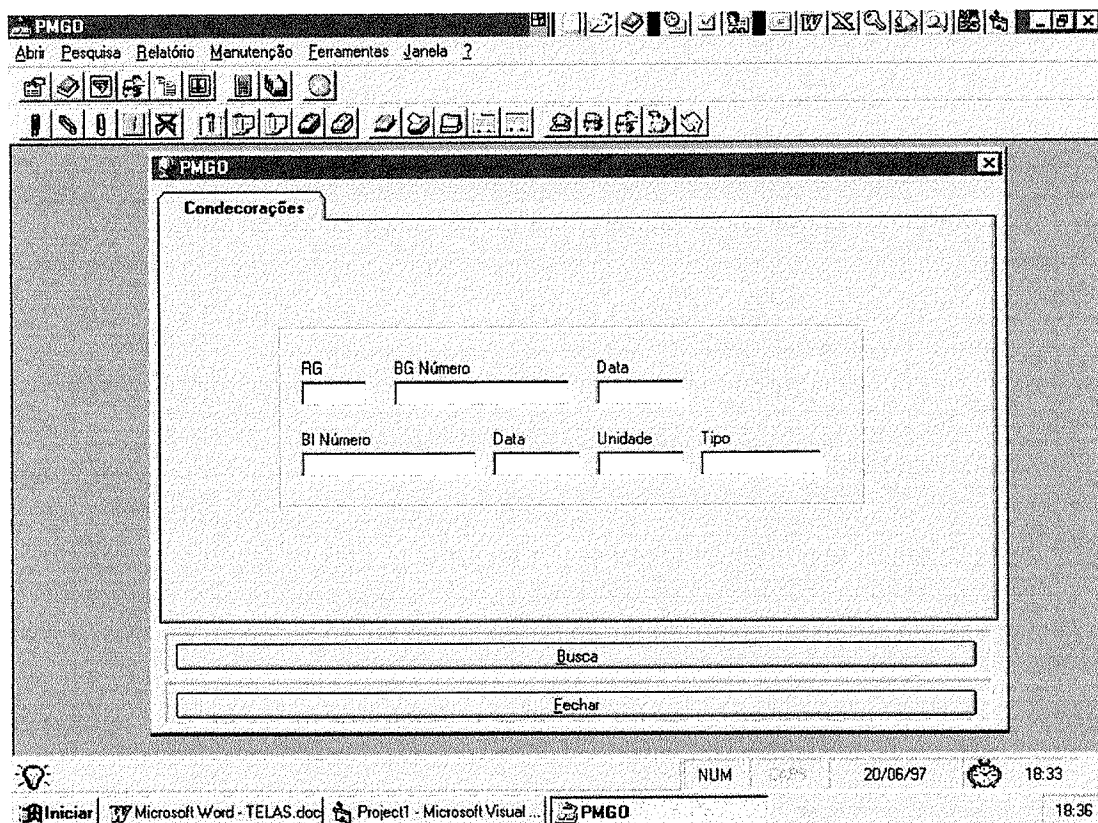
The screenshot displays the PMGO software interface. At the top, there is a menu bar with options: 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela 2'. Below the menu is a toolbar with various icons. The main window is titled 'PMGO' and contains a sub-window titled 'Comissões'. Inside the 'Comissões' window, there is a search form with the following fields:

RG	Espécie	Início	Término
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
BG Número		Data	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
BI Número		Unidade	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Below the search form, there are two buttons: 'Busca' and 'Fechar'. The Windows taskbar at the bottom shows the system tray with the date '20/06/97' and time '18:33'. The taskbar also displays several open applications: 'Iniciar', 'Microsoft Word - TELAS.doc', 'Project1 - Microsoft Visual ...', and 'PMGO'.

Esta opção traz informações das comissões exercidas pelo Policial Militar. Registra a espécie, o seu início e término, unidade, além do Boletim Geral ou interno que o designou.

PESQUISA - CONDECORAÇÕES



As condecorações poderão ser pesquisadas através desta opção. Ela traz informações do Boletim Geral ou interno, a data, a unidade e o tipo da condecoração. Para cada condecoração será gerado um registro.

PESQUISA - MEDALHAS

PMGO

Abrir Pesquisa Relatório Manutenção Ferramentas Janela 2

Medalhas

RG	BG Número	Data	Tipo
----	-----------	------	------

Busca

Fechar

NUM 20/06/97 18:33

Iniciar Microsoft Word - TELAS.doc Project1 - Microsoft Visual ... PMGO 18:36

Esta opção de pesquisa medalhas concedidas apresenta o Boletim Geral que publicou esta concessão, a data e o tipo de medalha.

PESQUISA - CONDENAÇÕES

The screenshot displays the PMGO software interface. At the top, there is a menu bar with options: 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela 1'. Below the menu is a toolbar with various icons. The main window is titled 'PMGO' and contains a sub-window titled 'Condenacoes'. Inside this sub-window, there is a search form with the following fields:

RG	BG Número	Data
BI Número	Data	Unidade

Below the search fields, there are two buttons: 'Busca' and 'Fechar'. The taskbar at the bottom shows the system tray with the date '20/06/97' and time '18:33', and the taskbar with open applications: 'Iniciar', 'Microsoft Word - TELAS.doc', 'Project1 - Microsoft Visual ...', and 'PMGO'.

A opção para pesquisar as condenações imposta ao Policial Militar traz informações do Boletim Geral ou Interno, a data e a unidade do Policial Militar.

PESQUISA - PUNIÇÕES

The screenshot displays the PMGO software interface. At the top, there is a menu bar with options: 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela ?'. Below the menu is a toolbar with various icons. The main window is titled 'PMGO' and contains a sub-window titled 'Punições'. Inside the 'Punições' window, there is a search form with the following fields:

RG	BG Número	Data	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
BI Número	Data	Unidade	Texto
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Below the search form, there are two buttons: 'Busca' and 'Fechar'. The taskbar at the bottom shows the system tray with the date '20/06/97' and time '18:33'. The taskbar also displays several open applications: 'Iniciar', 'Microsoft Word - TELAS.doc', 'Project1 - Microsoft Visual ...', and 'PMGO'.

Através desta pesquisa em punições sofridas pelo Policial Militar podemos obter informações do documento que a publicou, além de sua unidade e acesso ao texto.

PESQUISA - ELOGIOS

PMGO

Abrir Pesquisa Relatório Manutenção Ferramentas Janela ?

PMGO

Elogios

RG BG Número Data Unidade

BI Número Data Unidade Texto

Busca

Fechar

NUM CAP 20/06/97 18:33

Iniciar Microsoft Word - TELAS.doc Project1 - Microsoft Visual ... PMGO 18:37

A opção para se pesquisar elogios tem as mesmas informações da opção anterior. Cada elogio registrado gera um registro para o Policial Militar. Nesta opção, como na anterior, pode se ter o acesso ao texto publicado.

PESQUISA - AGREGAÇÃO

The screenshot displays the PMGD software interface. At the top, there is a menu bar with options: 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela 2'. Below the menu is a toolbar with various icons. The main window is titled 'PMGD' and contains a form titled 'Agregação'. The form has six input fields arranged in two rows: 'RG', 'BG Número', 'Data', and 'Início' in the first row; 'Término', 'Destino', and 'Tipo' in the second row. Below the form are two buttons: 'Busca' and 'Fechar'. The Windows taskbar at the bottom shows the 'Iniciar' button, open applications including 'Microsoft Word - TELAS.doc' and 'Project1 - Microsoft Visual ...', and the system tray with the date '20/06/97' and time '18:33'.

Esta opção traz registro das agregações verificadas na vida funcional do Policial Militar, através das informações do documento que as publicou, a data, início e término da agregação, o destino e o tipo da agregação.

PESQUISA - UNIDADES

The screenshot displays the PMGO software interface. At the top, there is a menu bar with options: 'Abrir', 'Pesquisa', 'Relatório', 'Manutenção', 'Ferramentas', and 'Janela ?'. Below the menu is a toolbar with various icons. The main window is titled 'Unidades' and contains a search form with the following fields:

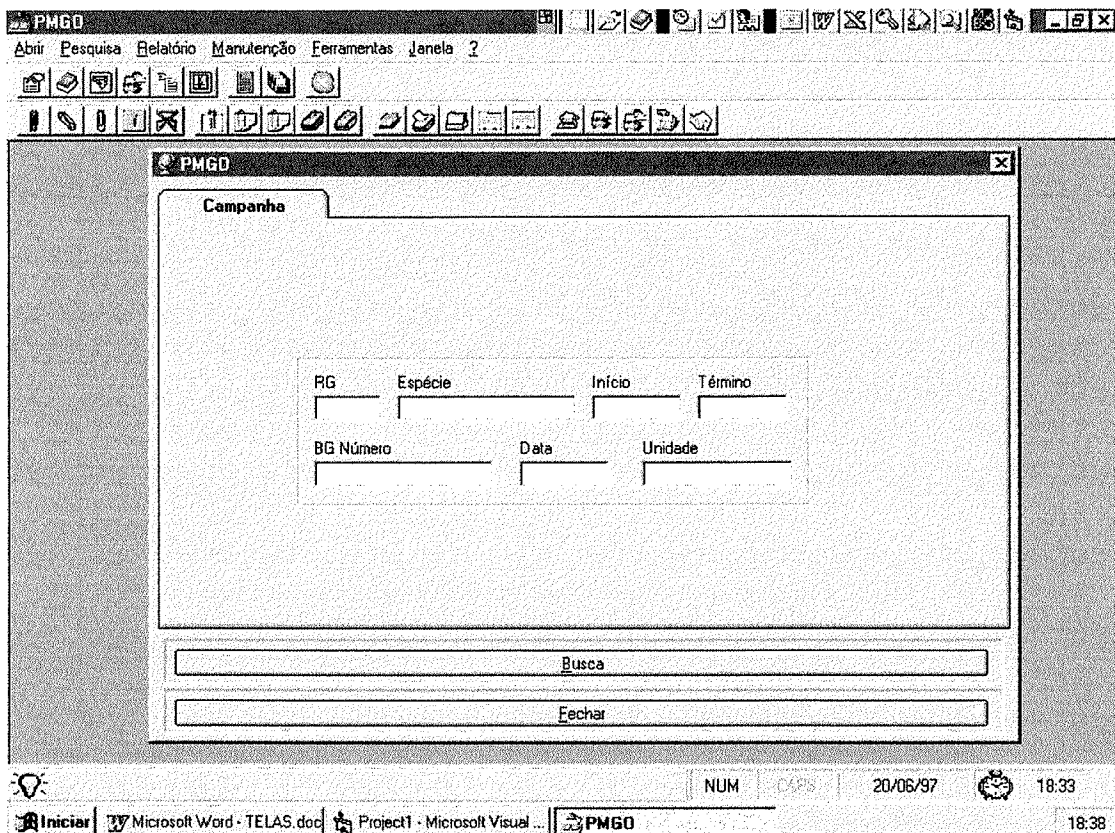
RG	Unidade	Apresentação
<input type="text"/>	<input type="text"/>	<input type="text"/>
BG Número	Data	BI Número
<input type="text"/>	<input type="text"/>	<input type="text"/>

Transferência	BG Número	Data
<input type="text"/>	<input type="text"/>	<input type="text"/>
BI Número	Data	
<input type="text"/>	<input type="text"/>	

Below the form are two buttons: 'Busca' and 'Fechar'. The taskbar at the bottom shows the system tray with 'NUM', 'CAPS', '20/06/97', and '18:33'. The taskbar also displays several open applications: 'Iniciar', 'Microsoft Word - TELAS.doc', 'Project1 - Microsoft Visual ...', and 'PMGO'.

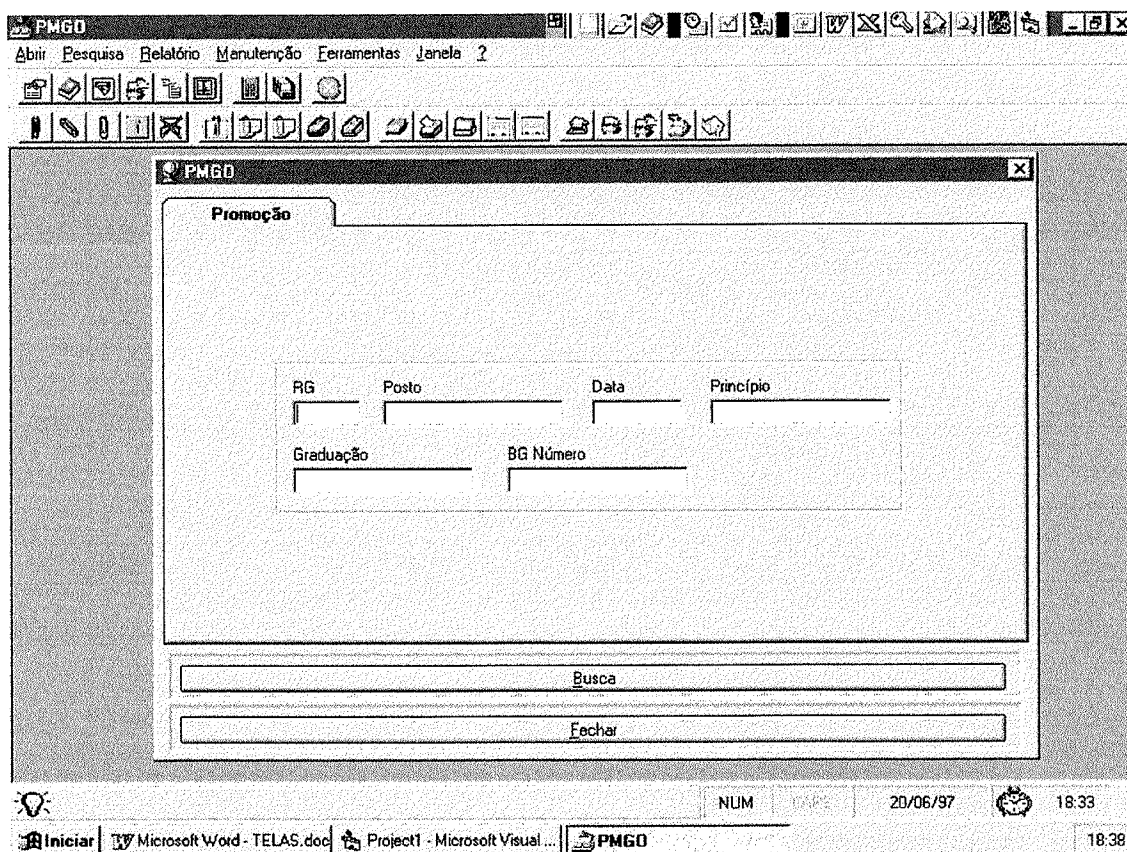
As classificações nas diversas unidades poderão ser pesquisadas através desta opção. Acessando esta opção teremos informações da unidade, a apresentação ou transferência além de dados dos documentos que efetivaram essas movimentações.

PESQUISA - CAMPANHA



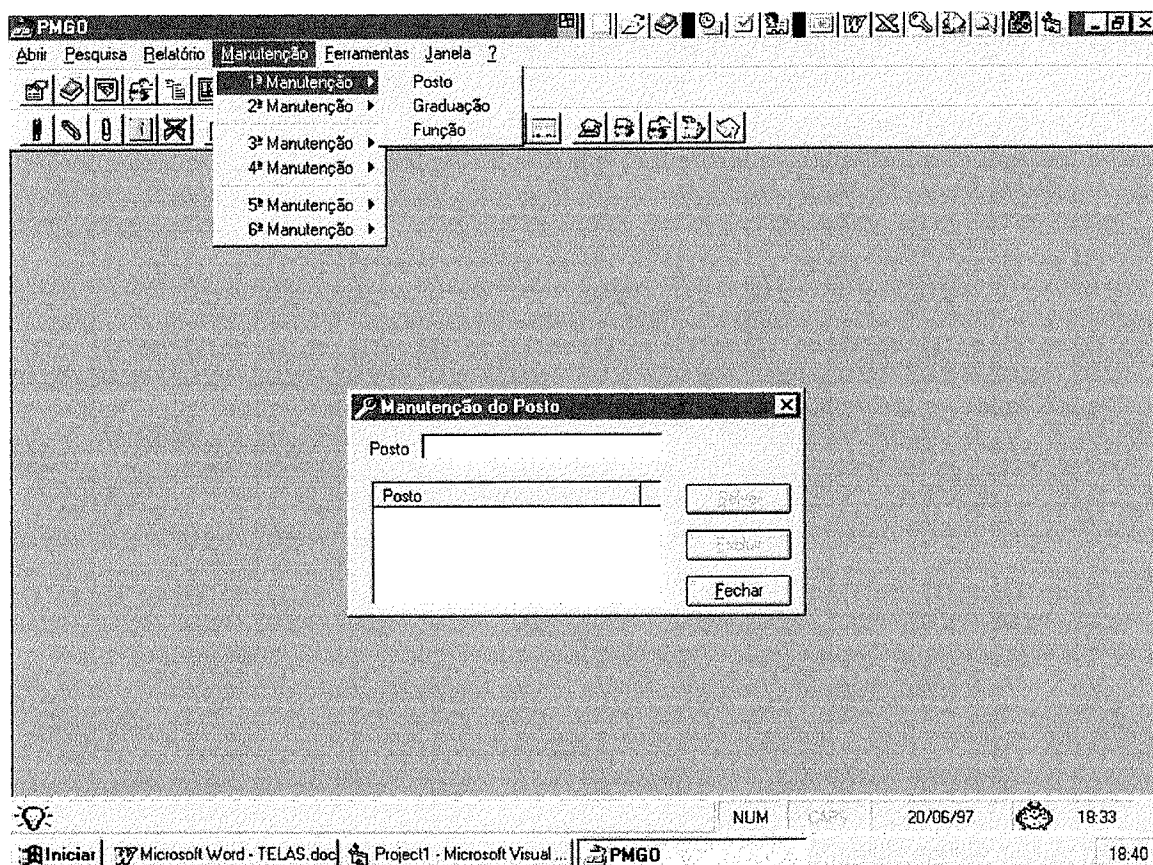
O tempo de campanha poderá ser pesquisado através desta opção, trazendo o início e término desta situação, além do documento que a registrou.

PESQUISA - PROMOÇÃO



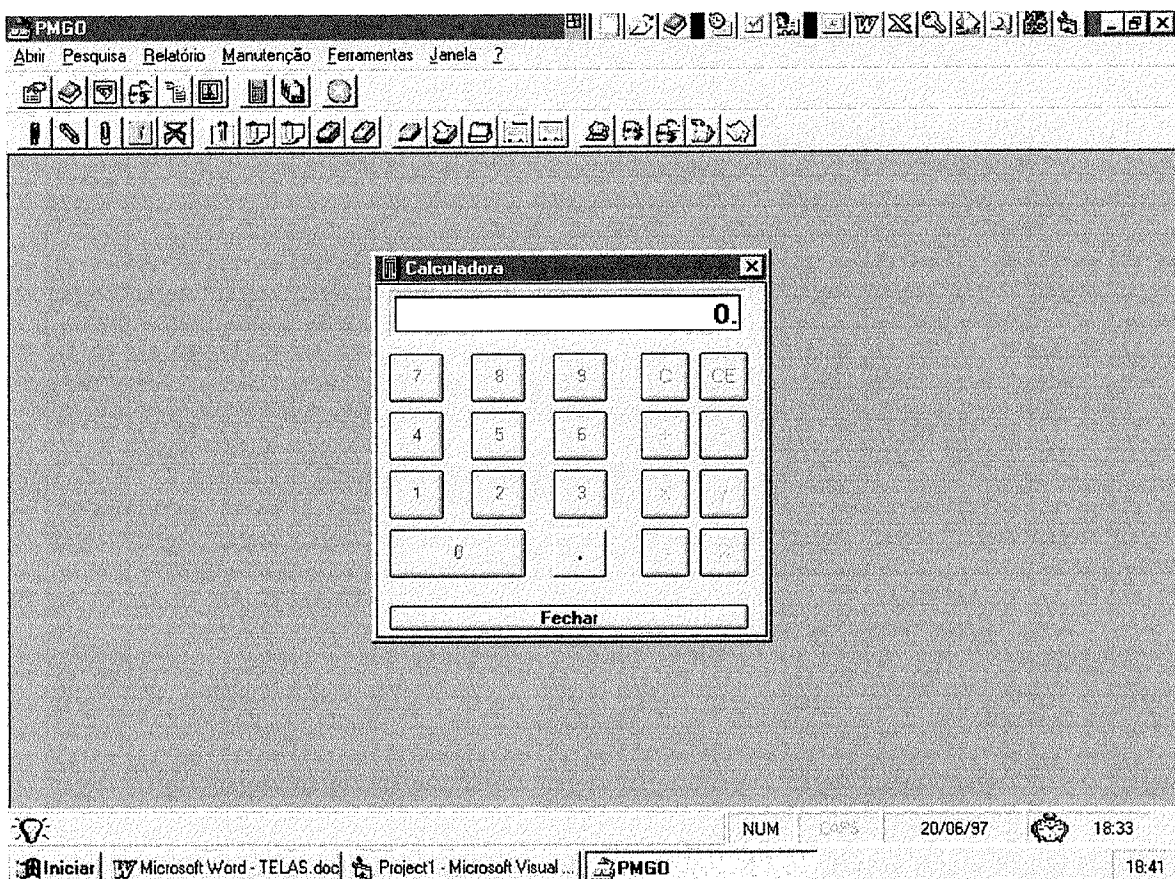
As promoções havidas durante a vida do Policial Militar poderão ser pesquisadas através da opção Promoção, no menu Pesquisa. Após acessada, esta opção trará informações do último posto, a data de promoção e o principio desta promoção no posto ou na graduação.

OPÇÕES DE MENU MANUTENÇÃO



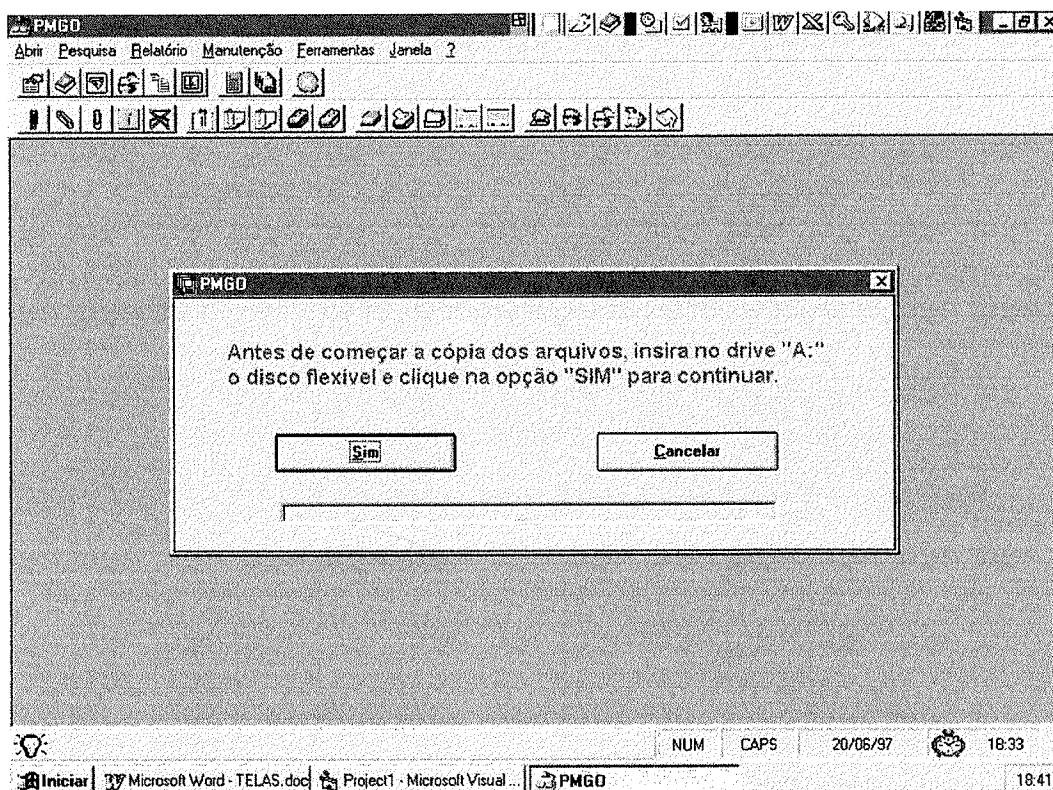
Através da opção Manutenção pode se fazer a entrada para as diversas tabelas do sistema, tendo sido esta dividida em seis manutenções. A exemplo de ilustração temos na primeira opção de manutenção o cadastro do Posto, Graduação e Função, onde a pessoa poderá inserir, ou excluir dados das tabelas.

CALCULADORA



O sistema de gerenciamento de pessoal traz também embutido em sua barra de ferramentas uma calculadora a qual poderá ser aberta a qualquer momento pelo operador do sistema.

BACKUP DO BANCO DE DADOS



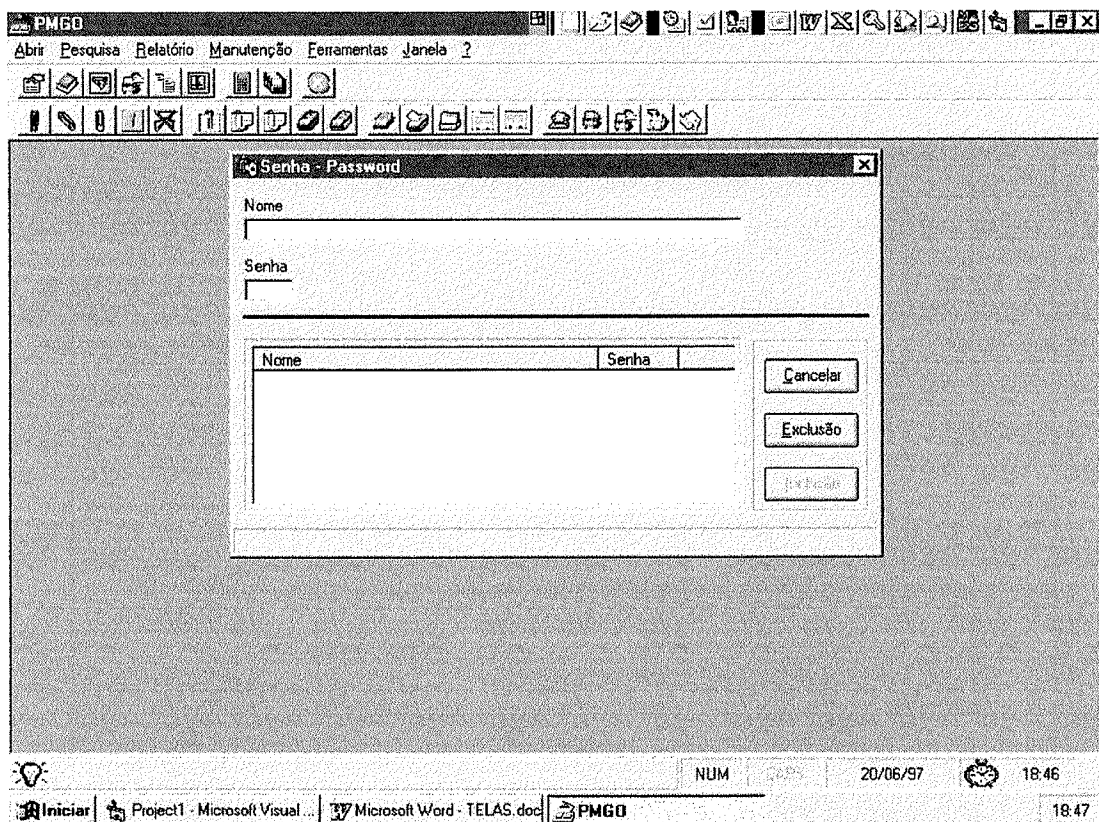
Uma opção importante do sistema é a condição de se fazer cópias de segurança, pois dados magnéticos estão sujeitos a danos diversos. Este backup poderá se feito em disquete ou, se o servidor possuir drives específicos, em fitas stremmer, as quais armazenam mais informações, podendo se fazer o backup até de um Winchester inteiro.

CONTROLE DE SENHAS



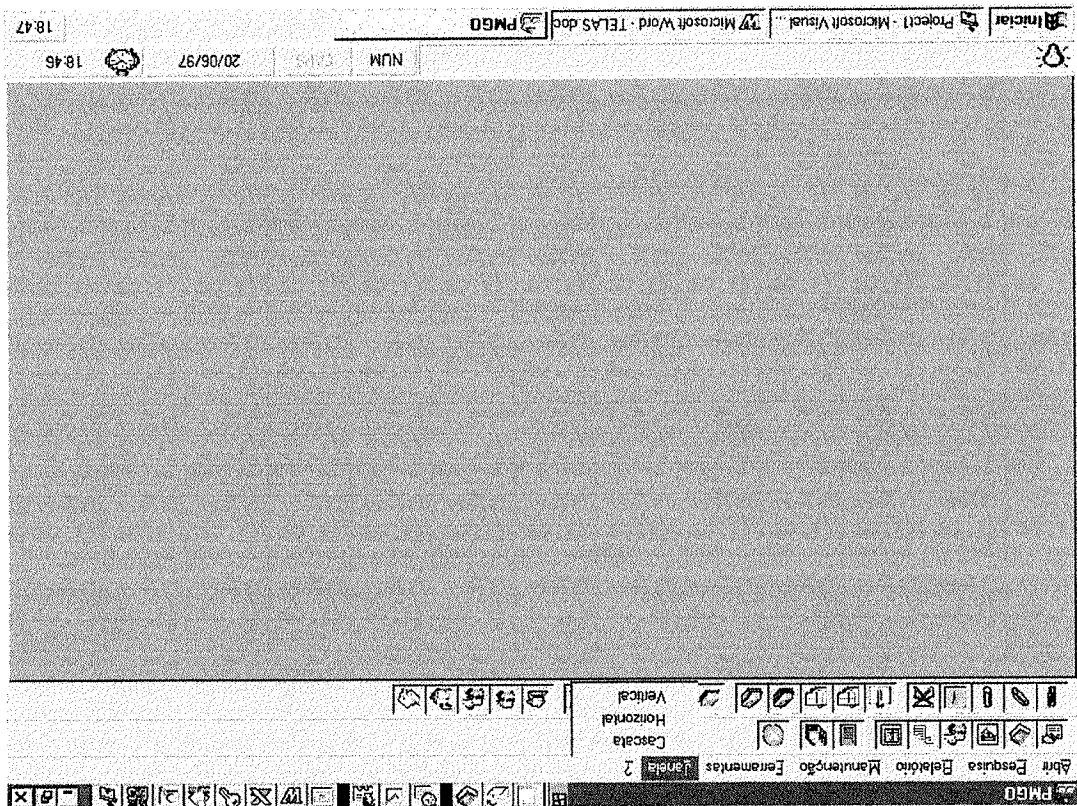
Através desta opção, o operador principal do sistema poderá realizar o cadastro de novos usuários.

SENHA - CADASTRO E EXCLUSÃO



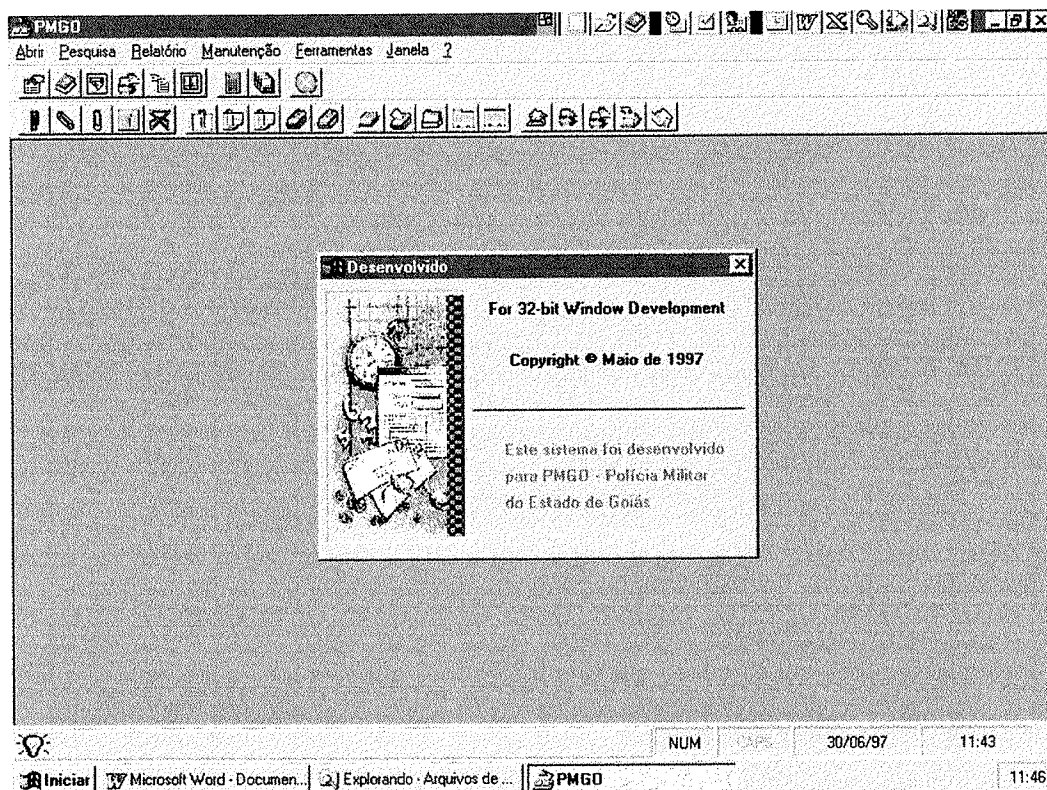
Nesta opção, o operador principal do sistema cadastra ou inclui novos operadores, como também se tem um controle sobre todas as senhas utilizadas no sistema, podendo controlar todos os acessos realizados no sistema através das senhas distribuídas.

Quando o operador estiver operando mais de uma opção do sistema ao mesmo tempo, um benefício da interatividade dos softwares para o ambiente Windows, ele poderá dispor estas telas na posição cascata, horizontal ou vertical, através do menu janela, no menu geral do sistema.



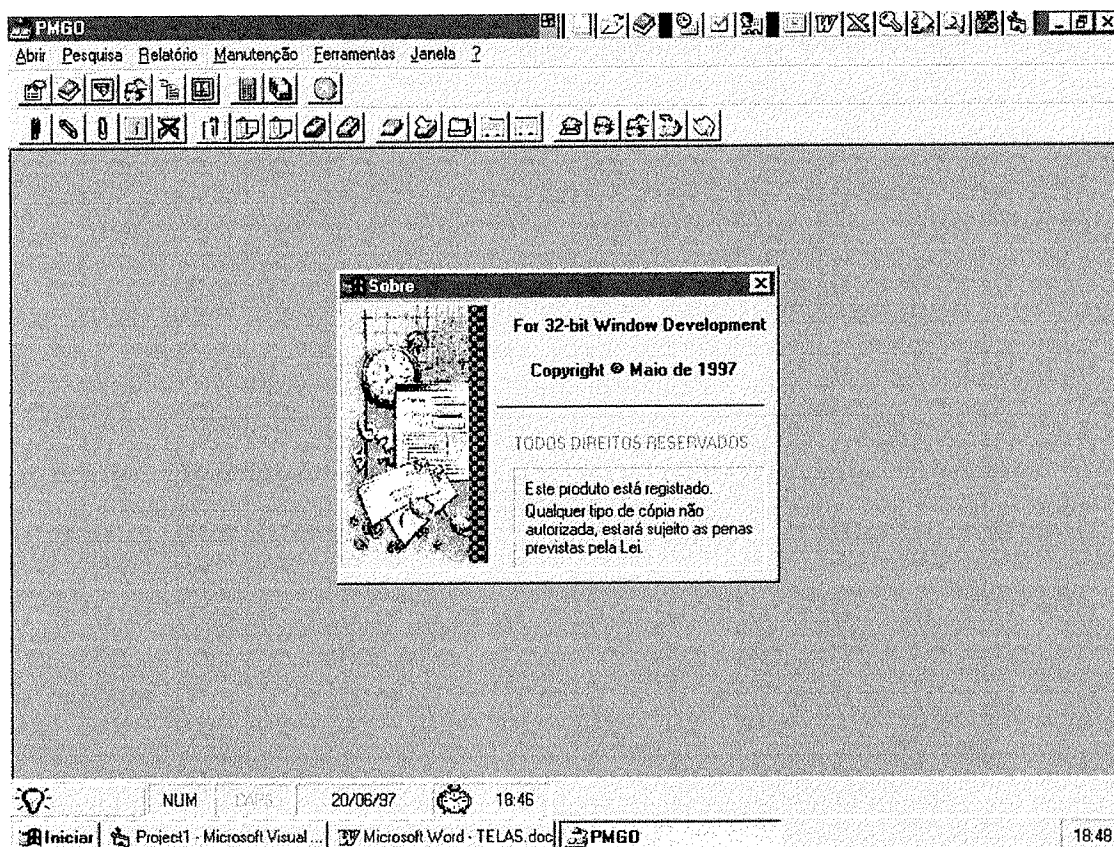
OPÇÕES DO MENU - JANELA

DESENVOLVIMENTO



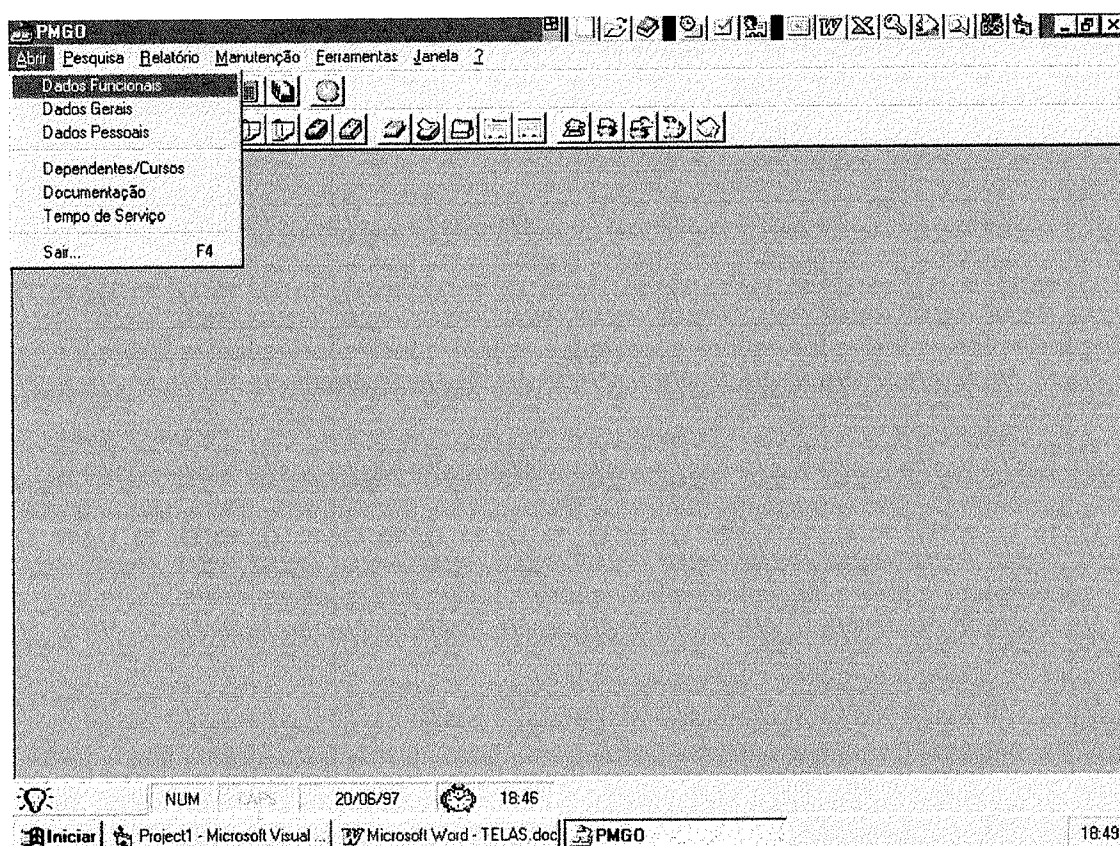
Esta é uma opção que demonstra a autoria do desenvolvimento do sistema.

SOBRE O PRODUTO



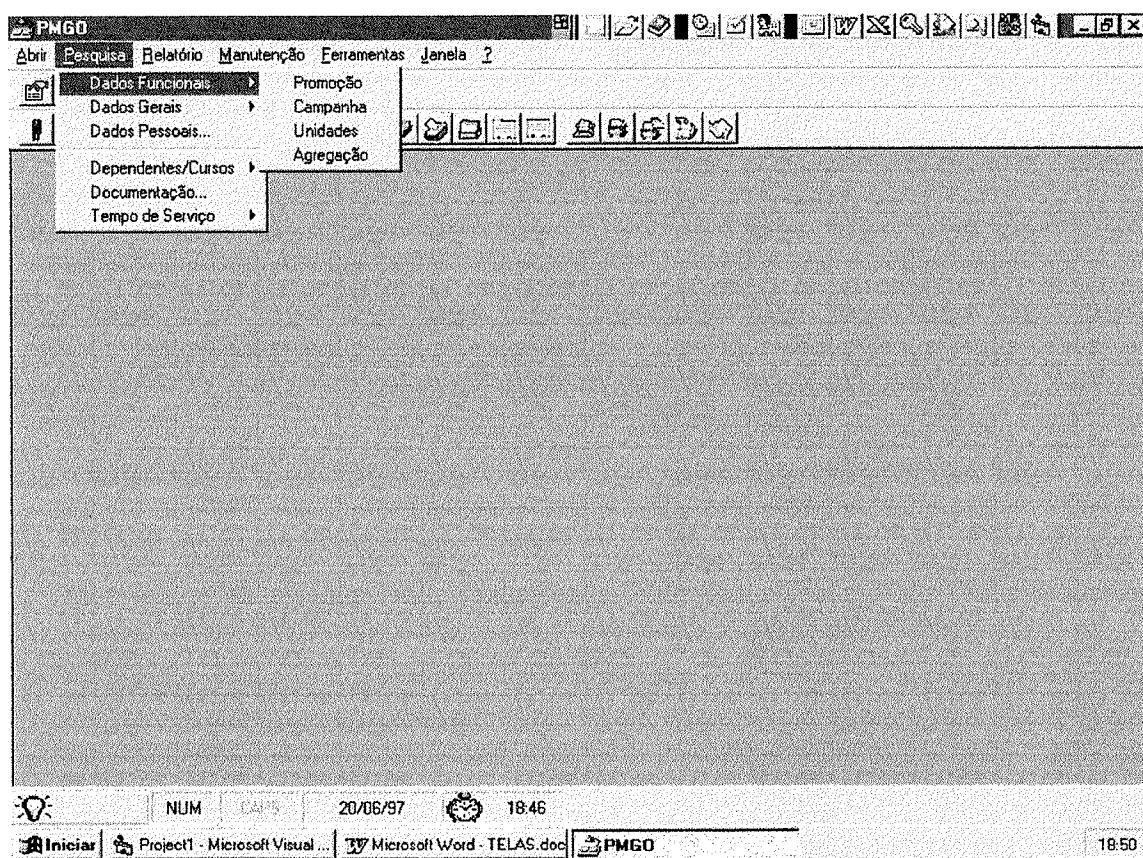
Através desta tela, uma opção denominada Sobre, vê-se o ambiente para o qual o produto foi desenvolvido, informa o tipo de máquina na qual ele deve rodar, o mês de implementação além de informar sobre o registro e os direitos do autor sobre o software.

OPÇÕES DO MENU ABRIR



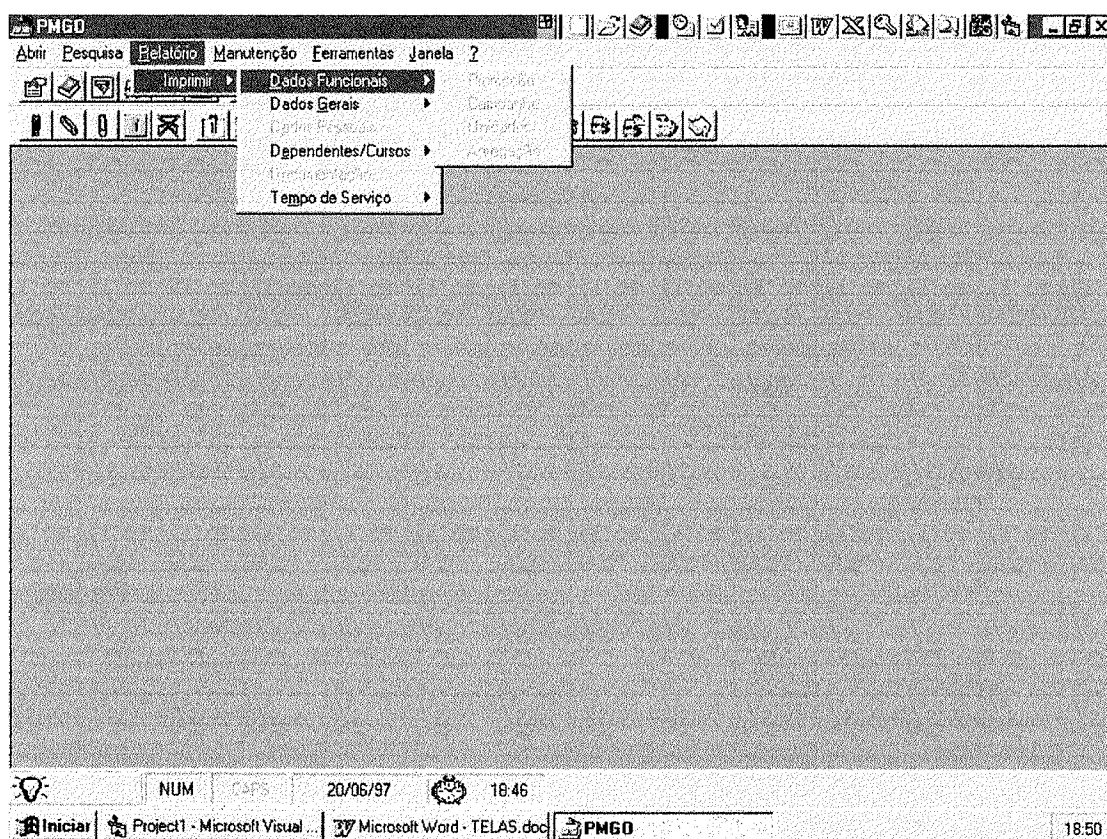
Esta opção mostra o desdobramento do Menu abrir que, ao ser acessado, oferece opções de se abrir dados funcionais, gerais e pessoais, dependentes e cursos, documentação e tempo de serviço, além de oferecer a opção para se sair do sistema.

OPÇÕES DO MENU PESQUISA



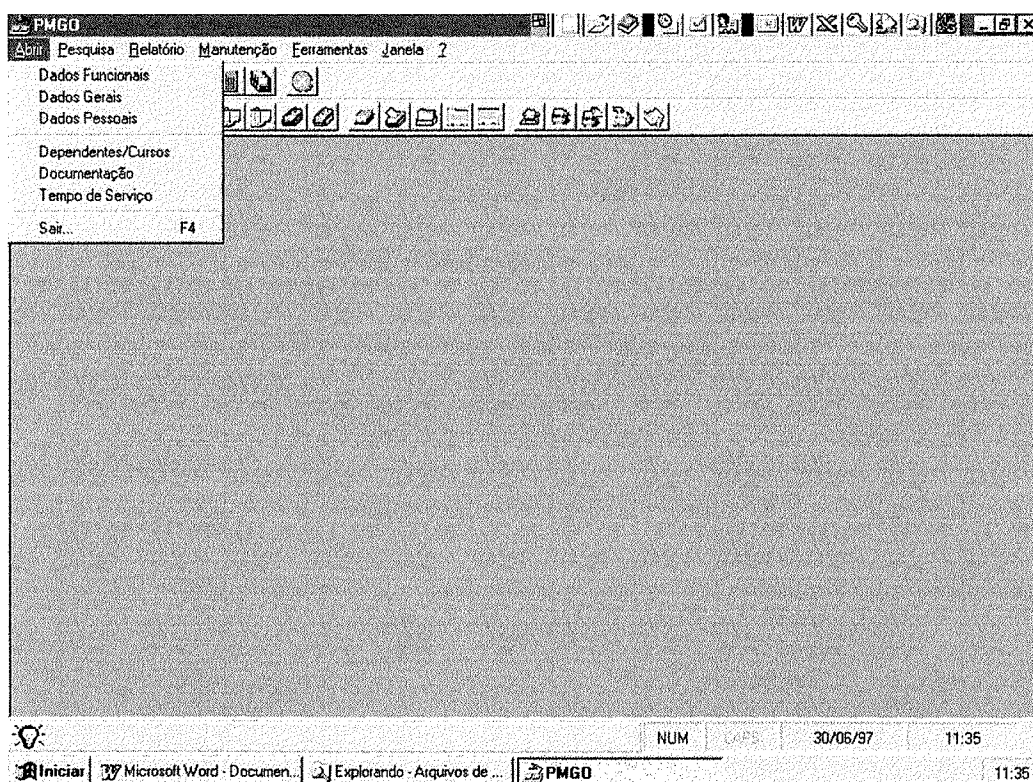
Através desta opção, Pesquisa, tem-se o acesso à informações cadastradas através da opção abrir. As opções são as mesmas relacionadas anteriormente, sendo que, na opção anterior, se cadastram os dados e nesta se pesquisa e altera estes dados.

OPÇÕES DO MENU RELATÓRIO



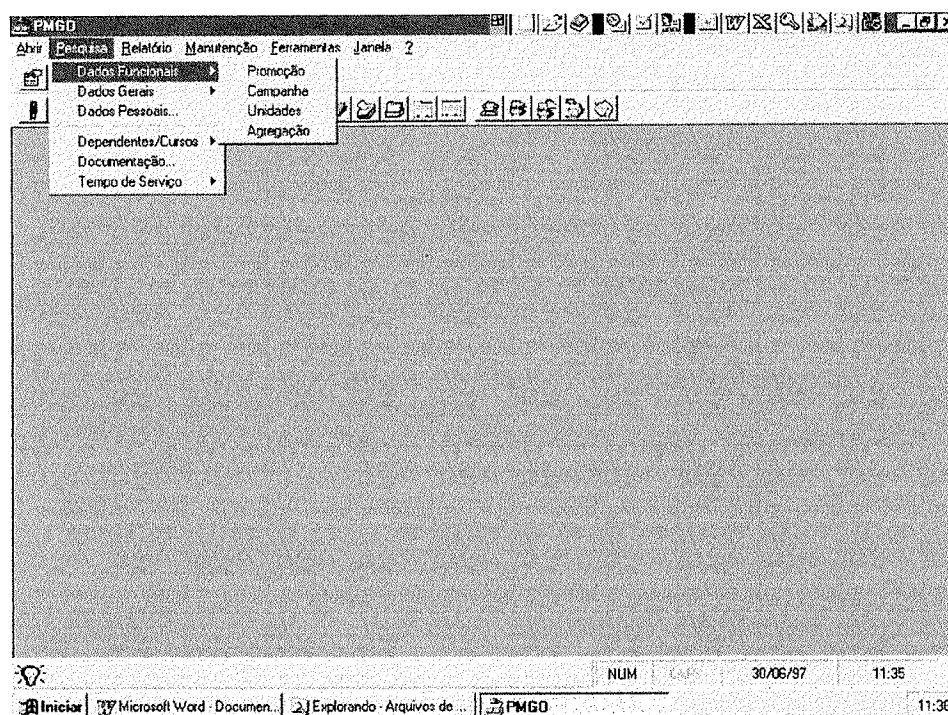
Uma das principais, se não a principal opção de um banco de dados, é a opção de geração de relatórios. É através dela que se extraem as informações contidas no banco de dados. Elas podem ser várias. As opções aqui relacionadas estão colocadas para se entender a sua utilidade.

MENU ABRIR



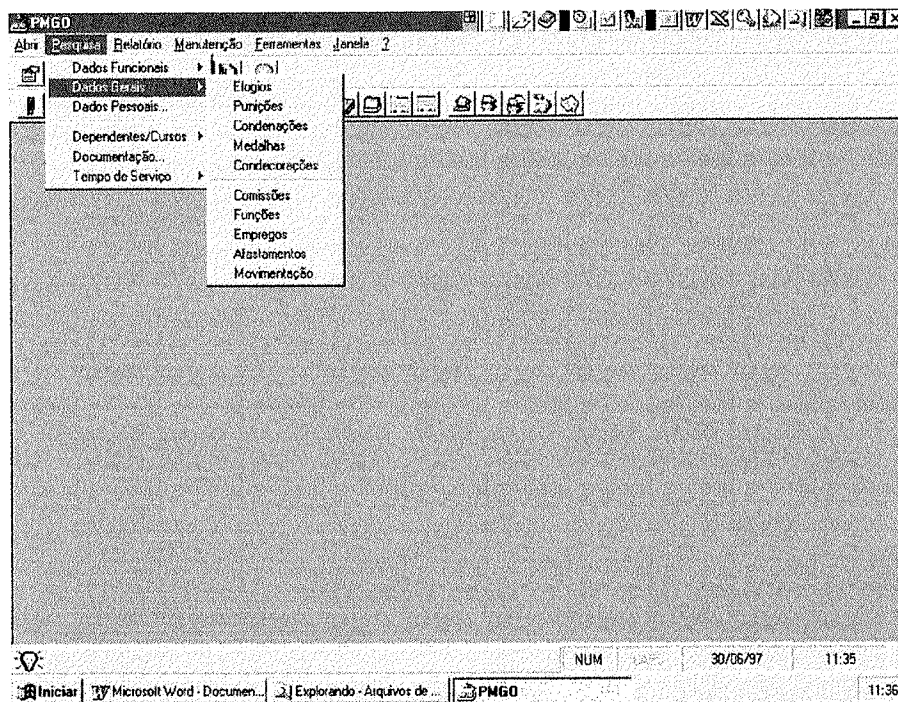
Esta opção mostra as opções do Menu Abrir. Através desta opção podemos ter acesso ao cadastro de Dados Funcionais, Dados Gerais, Dados Pessoais, Dependentes e Cursos, Documentação, Tempo de Serviço e como última opção, sair do sistema.

MENU PESQUISA



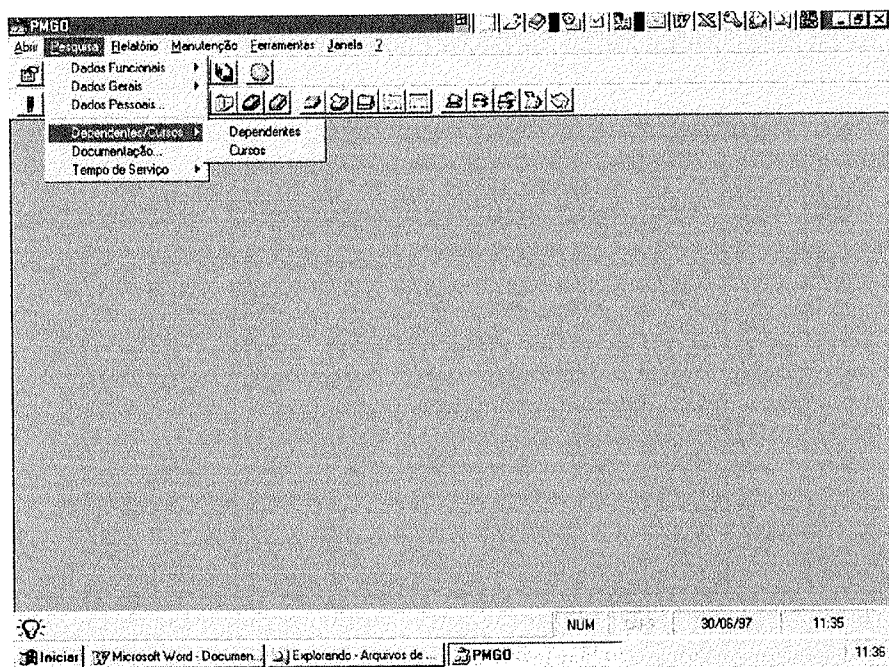
Através da opção pesquisa tem a condição de efetuar uma pesquisa nos dados cadastrados em Abrir. Neste exemplo temos a opção de se pesquisar, Promoção, Campanha, Unidades e Agregação, através da opção Dados Funcionais.

MENU PESQUISA - DADOS GERAIS



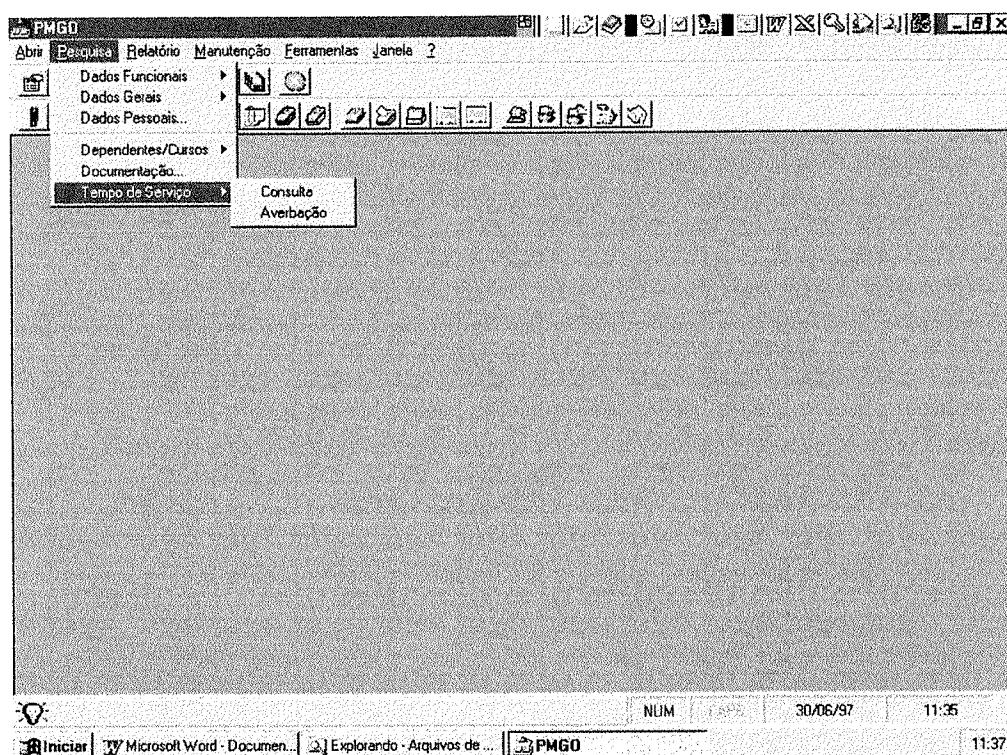
Através da opção de pesquisa Dados Gerais, temos a opção de pesquisa para Elogios, Punição, Condenação, Medalhas, Condenações, Comissões, Funções, Empregos e Movimentação.

MENU PESQUISA - DEPENDENTES/CURSOS



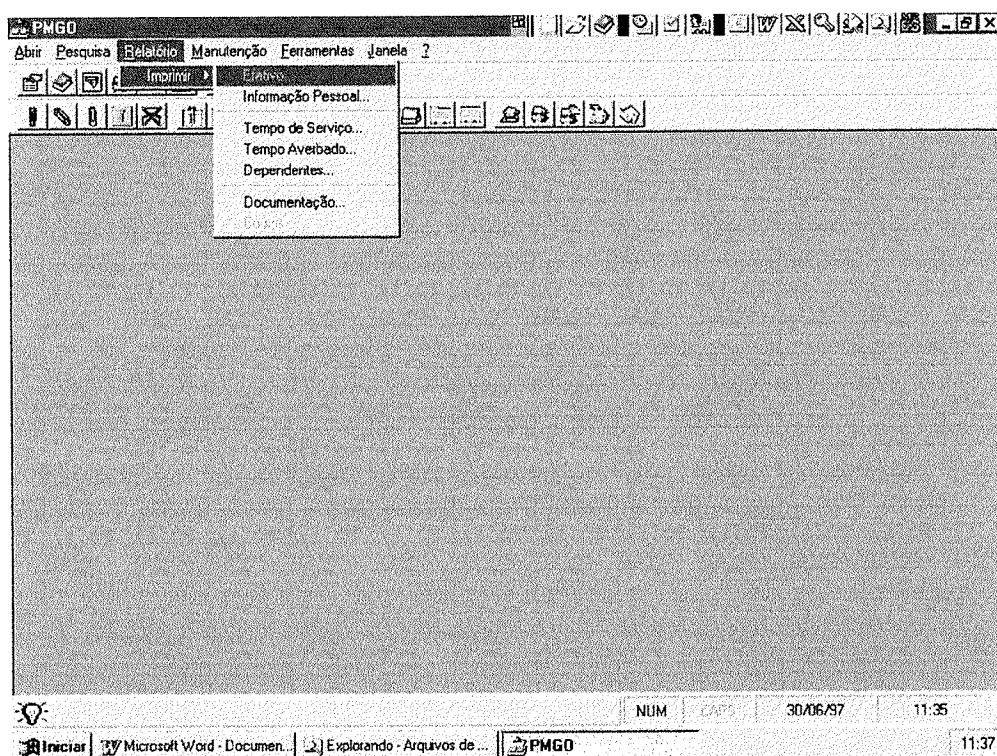
Em pesquisa Dependentes/Cursos pode-se pesquisar individualmente a dependentes ou os curso do Policial Militar cadastrado no sistema.

MENU PESQUISA - TEMPO DE SERVIÇO



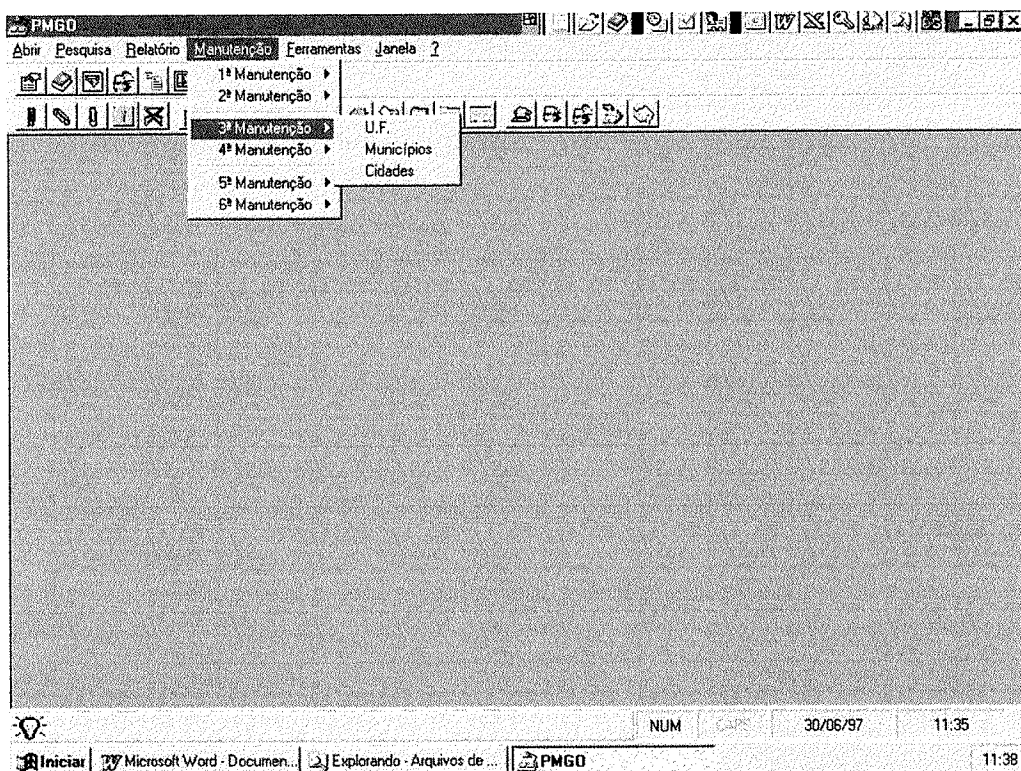
A opção de pesquisa, em relação a Tempo de Serviço, permite-se uma consulta normal ou consulta em tempo averbado.

MENU RELATÓRIO - IMPRIMINDO



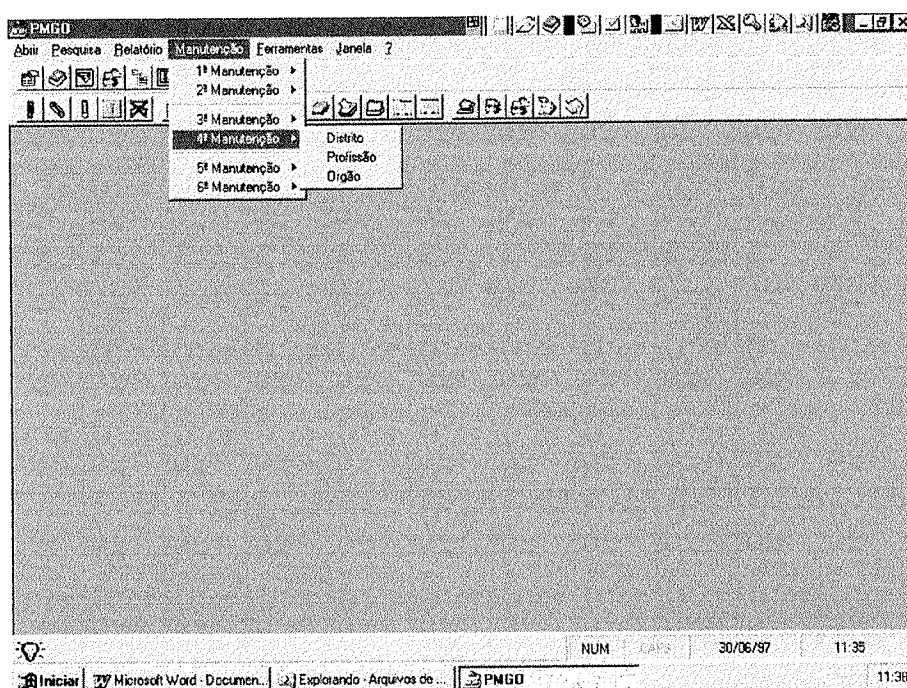
A opção de Relatório permite se imprimir relatório de efetivo, informação pessoal, tempo de serviço, tempo averbado, dependentes e documentação.

MENU MANUTENÇÃO - 3ª MANUTENÇÃO



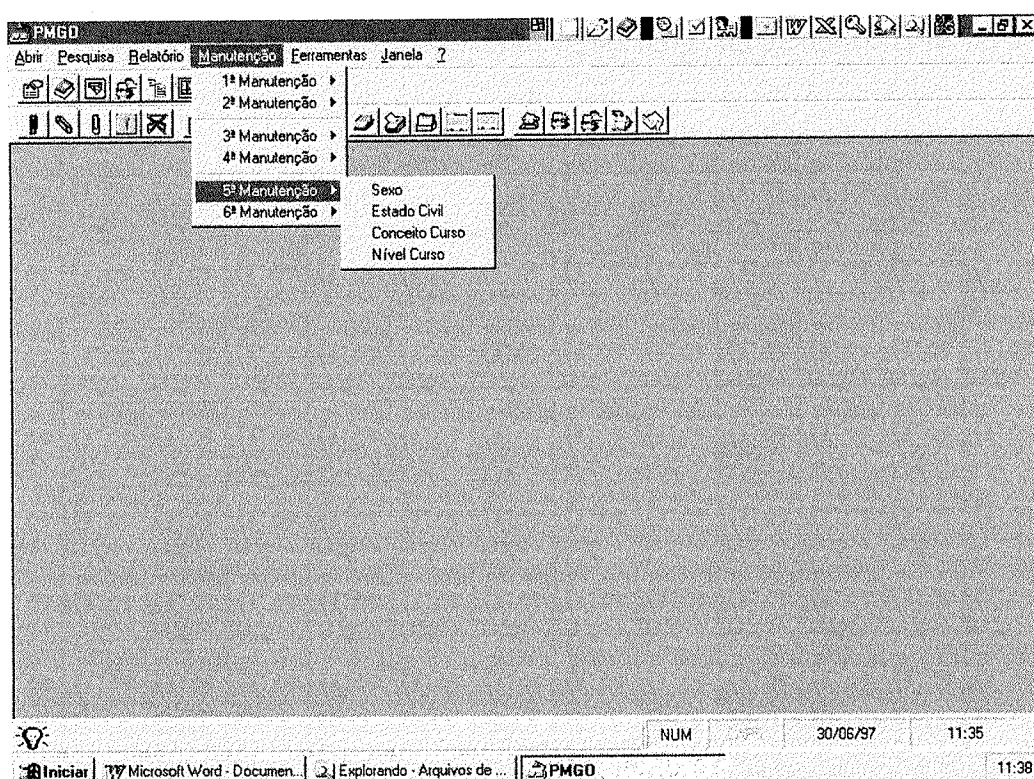
Na terceira opção do menu manutenção temos a opção para se cadastrar dados para UF, Municípios e Cidades.

MENU MANUTENÇÃO - 4ª MANUTENÇÃO



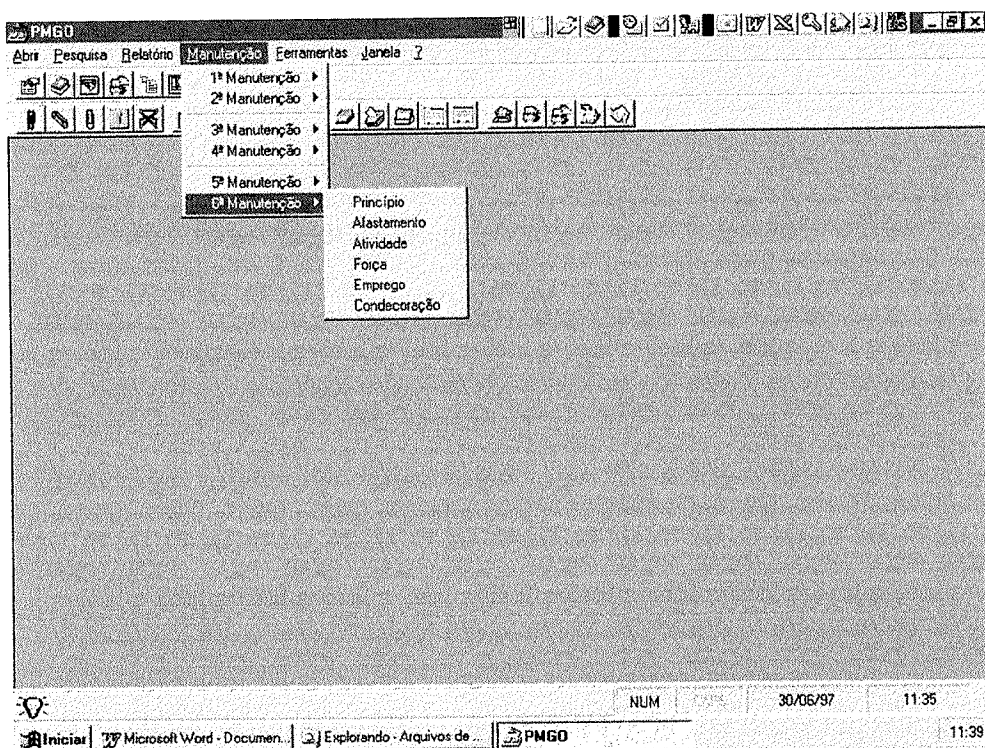
Na quarta opção do Menu Manutenção temos a opção para se cadastrar dados para Distritos, Profissão e Órgão.

MENU MANUTENÇÃO - 5ª MANUTENÇÃO



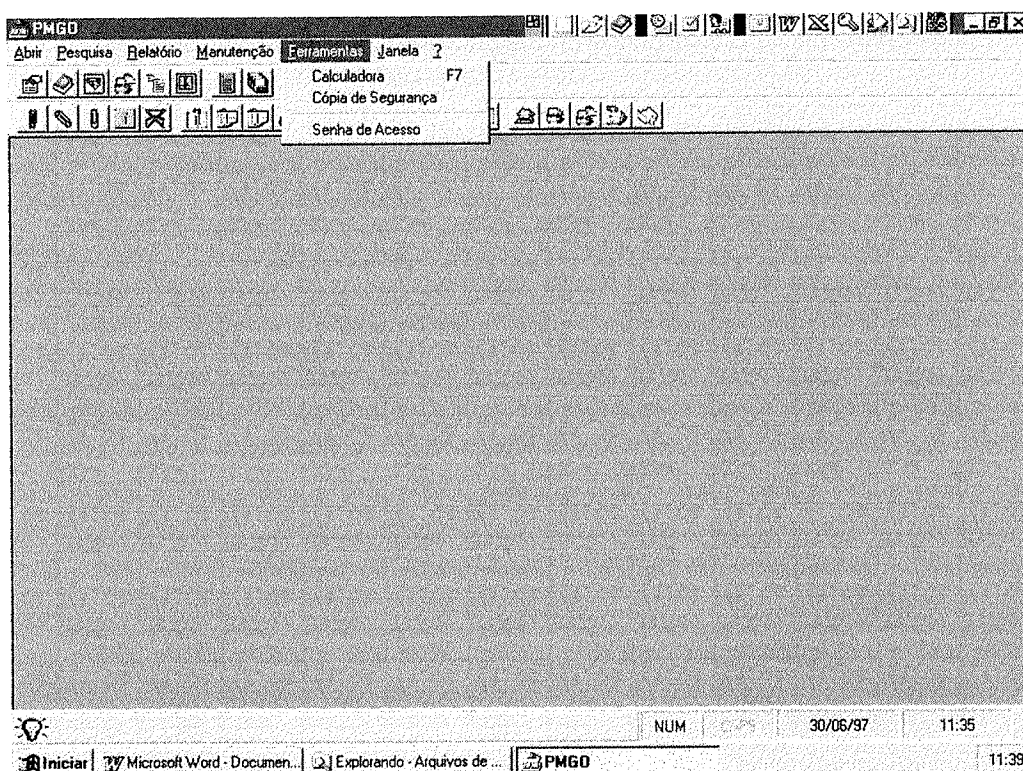
Na quinta opção do Menu Manutenção temos a opção para se cadastrar dados para Sexo, Estado Civil, Conceito Curso e Nível de Curso.

MENU MANUTENÇÃO - 6ª MANUTENÇÃO



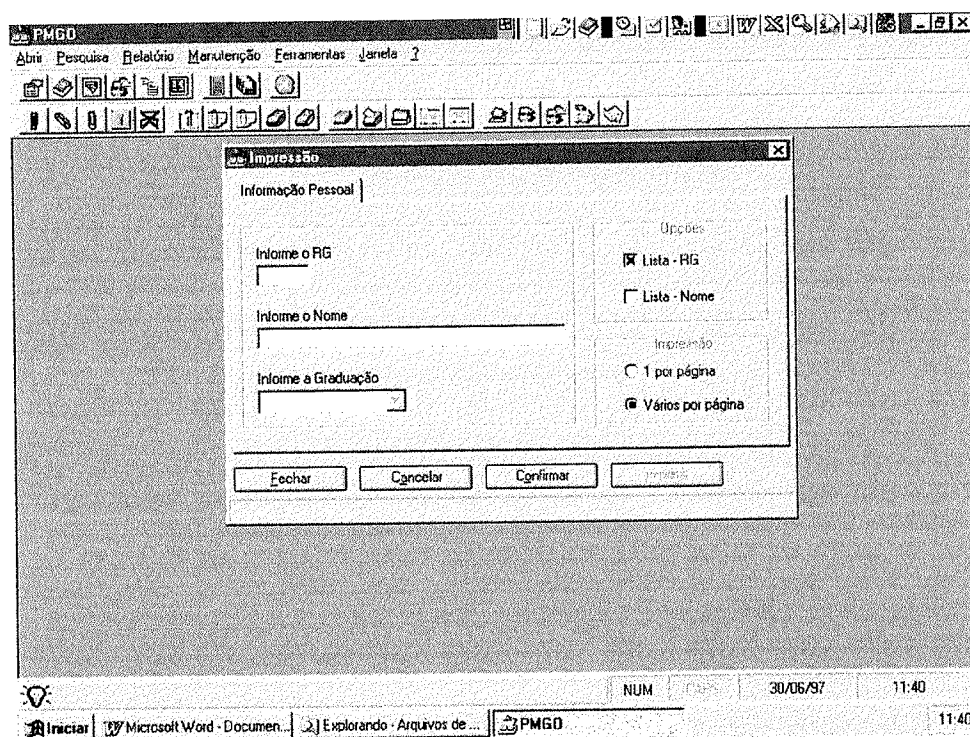
Na sexta opção do Menu Manutenção temos a opção para se cadastrar dados para Princípio, Afastamento, Atividade, Força, Emprego, Condecoração.

MENU FERRAMENTA



Através da opção Ferramentas, temos opção para se abrir a calculadora incorporada ao sistema, efetuar cópias de segurança e se realizar uma verificação na senha de acesso ao sistema.

MENU RELATÓRIO - INFORMAÇÃO PESSOAL



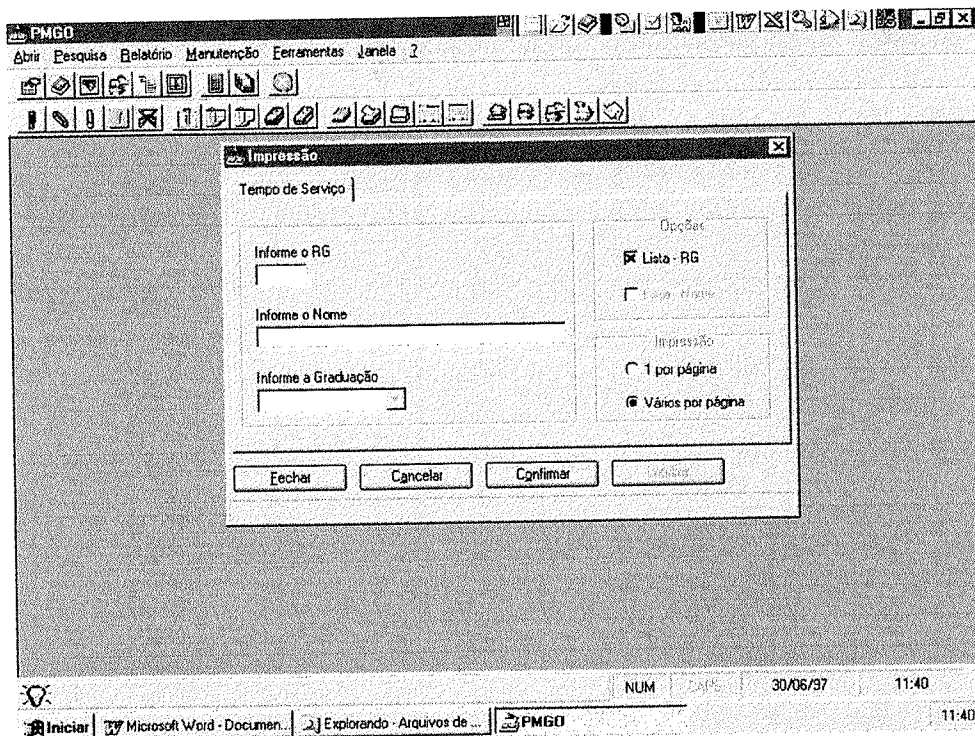
Nas opções 15 a 19 temos as opções de imprimir relatório do sistema.

Temos pela ordem as seguintes opções de impressão:

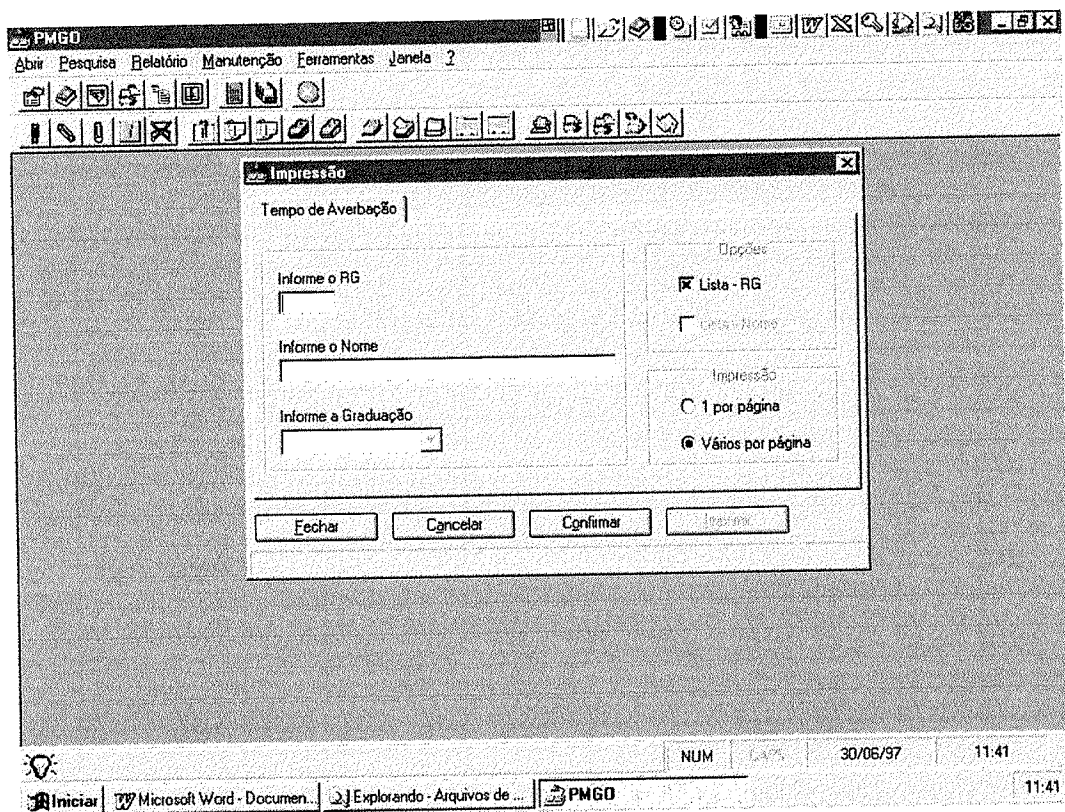
- 1 - Impressão de Dados Pessoal.
- 2 - Impressão de Tempo de Serviço.
- 3 - Impressão de Tempo Averbado.
- 4 - Impressão de Dependentes.
- 5 - Impressão de Documentação.

Outros relatórios, atendendo necessidades, poderão ser gerados através das informações gravadas no bando de dados.

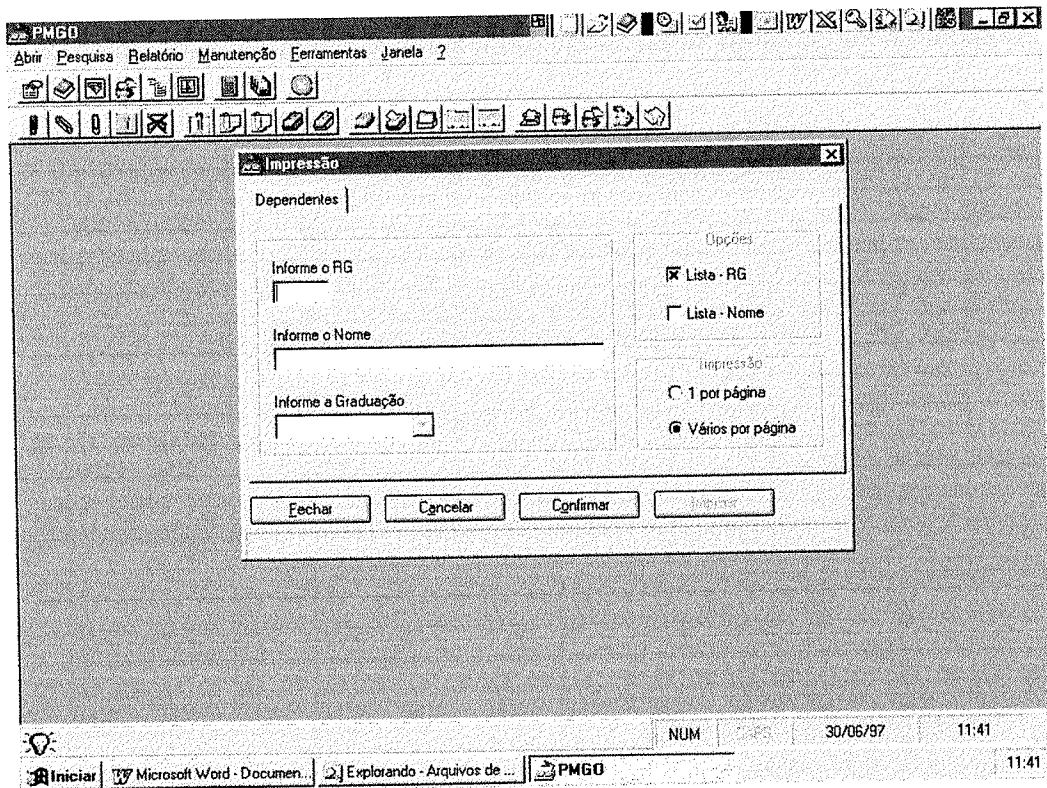
MENU RELATÓRIO - TEMPO DE SERVIÇO



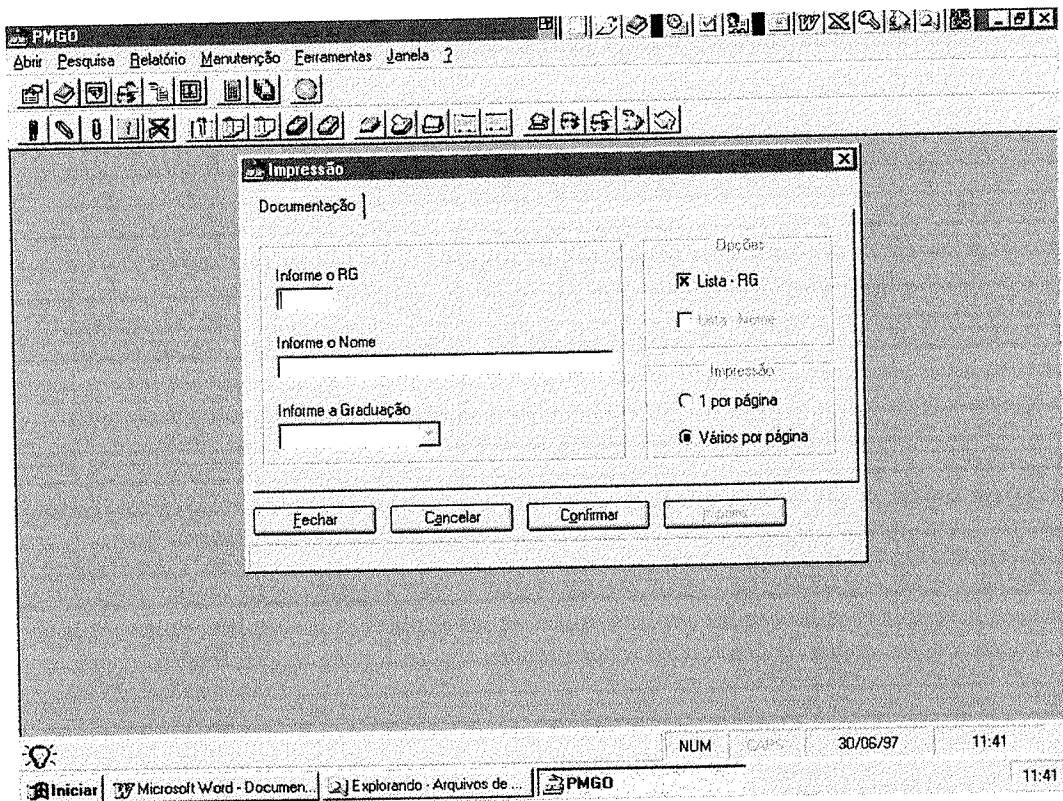
MENU RELATÓRIO - TEMPO DE AVERBAÇÃO



MENU RELATÓRIO - DEPENDENTES



MENU RELATÓRIO - DOCUMENTAÇÃO



8. CÓDIGOS DE PROGRAMAÇÃO

SISTEMA DE GERENCIAMENTO DE PESSOAL

Autor do Sistema: **CAP PM 14601 Avilmar Santos Ferreira**

Local: Goiânia- Goiás

Banco de dados: **Visual Basic, Versão 4.0 - Microsoft**

```
Declare Function FlashWindow Lib "user32" (ByVal hwnd As Long, ByVal  
bInvert As Long) As Long 'Coloca a janela piscando  
Dim Bdados As Database
```

```
Dim TabTempoServico As Recordset 'Consulta  
Dim TabTempoServico2 As Recordset 'Averbação  
Dim TabDadosPessoais As Recordset 'Busca Dados Pessoais  
Dim TabAfastamentos As Recordset 'Busca Tabela Afastamentos  
Dim TabAgregacao As Recordset 'Busca pela Tabela Agregacao
```

```
Private Sub BtnAverbacao_Click(Index As Integer)  
    Select Case Index
```

Case 0 'Salvar

If Not TxtAverbacao(0) <> "" Then

MsgBox "O Registro Geral não pode conter valor nulo...", 0 +
64, "Atenção..."

LimpaCampos2

TxtAverbacao(0).SetFocus

Exit Sub

Else

TabTempoServico2.AddNew

TabTempoServico2!RG = TxtAverbacao(0)

TabTempoServico2!Origem = TxtAverbacao(1)

TabTempoServico2!Inicio = TxtAv1

TabTempoServico2!Termino = TxtAv2

TabTempoServico2!BGNumero = TxtAverbacao(4)

TabTempoServico2!Data = TxtAverbacao(5)

TabTempoServico2!Dias = TxtAv3

End If

Case 1 'Cancelar

BtnAverbacao(0).Enabled = False

BtnAverbacao(2).Enabled = False

BtnAverbacao(3).Enabled = False

LimpaCampos2

Case 2 'Alterar

If TabTempoServico2!Origem = TxtAverbacao(1)

TabTempoServico2!Inicio = TxtAv1

TabTempoServico2!Termino = TxtAv2

TabTempoServico2!BGNumero = TxtAverbacao(4)

TabTempoServico2!Data = TxtAverbacao(5)

TabTempoServico2!Dias = TxtAv3

TabTempoServico2!Mes = TxtAv4

TabTempoServico2!Ano = TxtAv5

TabTempoServico2.Update

BtnAverbacao(0).Enabled = False

BtnAverbacao(2).Enabled = False

BtnAverbacao(3).Enabled = False

LimpaCampos2

```

    TxtAverbacao(0).SetFocus
End If

```

```

Case 0 'Salvar

```

```

    StatusBar.SimpleText = "Salva as informações..."

```

```

Case 1 'Cancelar

```

```

    StatusBar.SimpleText = "Cancela a entrada das informações..."

```

```

Case 2 'Alterar

```

```

    StatusBar.SimpleText = "Altera o registro..."

```

```

Set TabAfastamentos = Bdados.OpenRecordset("TabAfastamentos")

```

```

Set TabAgregacao = Bdados.OpenRecordset("TabAgregacao")

```

```

TabAgregacao.Index = "INDRG"

```

```

TabDadosPessoais.Index = "IND1"

```

```

TabTempoServico2.Index = "IND1"

```

```

TabTempoServico.Index = "INDRG"

```

```

TabAfastamentos.Index = "INDRG"

```

```

TxtConsulta(1) = Format(TabTempoServico(1), "&") 'Ativo

```

```

TxtConsulta(2) = Format(TabTempoServico(2), "&") 'Afastamento

```

```

TxtConsulta(3) = Format(TabTempoServico(3), "&") 'Averbado

```

```

TxtConsulta(4) = Format(TabTempoServico(4), "&") 'Agregado

```

```

Txt = Format(TabTempoServico(5), "&") 'Total

```

```

BtnConsulta(0).Enabled = False

```

```

BtnConsulta(3).Enabled = True

```

```

Exit Sub

```

```

Else

```

```

    LimpaCampos

```

```

        If Not TabTempoServico2.NoMatch Then

```

```

            MsgBox "Este Registro já está cadastrado !!!",
vbExclamation + vbOKOnly, "Alerta"

```

```

            BtnAverbacao(0).Enabled = False

```

```

            BtnAverbacao(2).Enabled = False

```

```

            BtnAverbacao(3).Enabled = False

```

```

            TxtAverbacao(0) = ""

```

```

        TxtAverbacao(0).SetFocus
    Exit Sub
End If

' If Mac evaluates as true, do the statements following the #If.
#If Mac Then
    ' Place exclusively Mac statements here.
    '
    '
' Otherwise, if it is a 32-bit Windows program, do this:
#ElseIf Win32 Then
    ' Place exclusively 32-bit Windows statements here.
    '
    '
' Otherwise, if it is neither, do this:
#Else
    ' Place other platform statements here.
    '
    '
#End If

#Const DebugVersion = 1 ' Will evaluate true in #If block.

Dim MyValue
MyValue = 2 * 2 ' Returns 4.
MyValue = 459.35 * 334.90 ' Returns 153836.315.

Dim MyStr
MyStr = "Hello" & " World" ' Returns "Hello World".
MyStr = "Check " & 123 & " Check" ' Returns "Check 123 Check".

Private Sub Form_Load ()
    Picture1.ScaleHeight = 100 ' Set scale to 100.
    Picture1.ScaleWidth = 100
    Picture1.AutoRedraw = True ' Turn on AutoRedraw.
    Picture1.ForeColor = 0 ' Set ForeColor.
    Picture1.FillColor = QBColor(9) ' Set FillColor.
    Picture1.FillStyle = 0 ' Set FillStyle.

```

```

Picture1.Circle (50, 50), 30    ' Draw a circle.
Picture1.AutoRedraw = False    ' Turn off AutoRedraw.
End Sub

```

```

Private Sub Form_Click ()
    ' Define bitmap formats.
    Dim ClpFmt, Msg ' Declare variables.
    On Error Resume Next ' Set up error handling.
    If Clipboard.GetFormat(vbCFText) Then ClpFmt = ClpFmt + 1
    If Clipboard.GetFormat(vbCFBitmap) Then ClpFmt = ClpFmt + 2
    If Clipboard.GetFormat(vbCFDIB) Then ClpFmt = ClpFmt + 4
    If Clipboard.GetFormat(vbCFRTF) Then ClpFmt = ClpFmt + 8
    Select Case ClpFmt
        Case 1
            Msg = "The Clipboard contains only text."
        Case 2, 4, 6
            Msg = "The Clipboard contains only a bitmap."
        Case 3, 5, 7
            Msg = "The Clipboard contains text and a bitmap."
        Case 8, 9
            Msg = "The Clipboard contains only rich text."
        Case Else
            Msg = "There is nothing on the Clipboard."
    End Select
    MsgBox Msg ' Display message.
End Sub

```

```

Private Sub Form_Load ()
    ' Load the icons.
    Picture1.Picture = LoadPicture("ICONS\COMPUTER\TRASH02A.ICO")
    Picture2.Picture = LoadPicture("ICONS\COMPUTER\TRASH02B.ICO")
End Sub

```

```

Private Sub Form_Click ()
    ' Switch the icons.
    Picture3.Picture = Picture1.Picture
    Picture1.Picture = Picture2.Picture

```

```
Picture2.Picture = Picture3.Picture
' Clear the third picture (not necessary if not visible).
```

```
Picture3.Picture = LoadPicture()
End Sub
```

```
Dim MyValue
MyValue = 10 / 4      ' Returns 2.5.
MyValue = 10 / 3     ' Returns 3.333333.
```

```
Dim MyCheck
MyCheck = "aBBBa" Like "a*a"   ' Returns True.
MyCheck = "F" Like "[A-Z]"     ' Returns True.
MyCheck = "F" Like "[!A-Z]"    ' Returns False.
MyCheck = "a2a" Like "a#a"     ' Returns True.
MyCheck = "aM5b" Like "a[L-P]#[!c-e]" ' Returns True.
MyCheck = "BAT123khg" Like "B?T*" ' Returns True.
MyCheck = "CAT123khg" Like "B?T*" ' Returns False.
```

```
Dim MyNumber
MyNumber = Abs(50.3)    ' Returns 50.3.
MyNumber = Abs(-50.3)  ' Returns 50.3.
```

```
Sub AbsolutePositionX()
```

```
    Dim dbsNorthwind As Database
    Dim rstEmployees As Recordset
    Dim strMessage As String
```

```
    Set dbsNorthwind = OpenDatabase("Northwind.mdb")
    ' AbsolutePosition only works with dynasets or snapshots.
    Set rstEmployees = _
        dbsNorthwind.OpenRecordset("Employees", _
            dbOpenSnapshot)
```

```
    With rstEmployees
        ' Populate Recordset.
        .MoveLast
```

```

.MoveFirst

' Enumerate Recordset.
Do While Not .EOF
    ' Display current record information. Add 1 to
    ' AbsolutePosition value because it is zero-based.
    strMessage = "Employee: " & !LastName & vbCr & _
        "(record " & (.AbsolutePosition + 1) & _
        " of " & .RecordCount & ")"
    If MsgBox(strMessage, vbOKCancel) = vbCancel _
        Then Exit Do
    .MoveNext
Loop

.Close
End With

dbsNorthwind.Close

```

End Sub

Sub PercentPositionX()

```

Dim dbsNorthwind As Database
Dim rstProducts As Recordset
Dim strFind As String
Dim strMessage As String

Set dbsNorthwind = OpenDatabase("Northwind.mdb")
' PercentPosition only works with dynasets or snapshots.
Set rstProducts = dbsNorthwind.OpenRecordset( _
    "SELECT ProductName FROM Products " & _
    "ORDER BY ProductName", dbOpenSnapshot)

With rstProducts
    ' Populate the Recordset.
    .MoveLast

```

.MoveFirst

Do While True

```
' Show current record information and ask user
' for input.
strMessage = "Product: " & !ProductName & vbCrLf & _
    "The record pointer is " & _
    Format(.PercentPosition, "##0.0") & _
    "% from the " & vbCrLf & _
    "beginning of the Recordset." & vbCrLf & _
    "Please enter a character search string " & _
    "for a product name."
strFind = Trim(InputBox(strMessage))
If strFind = "" Then Exit Do
```

```
' Try to find a record matching the search string.
.FindFirst "ProductName >= " & strFind & ""
If .NoMatch Then .MoveLast
```

Loop

.Close

End With

dbsNorthwind.Close

End Sub

Sub RecordCountX()

Dim dbsNorthwind As Database

Dim rstEmployees As Recordset

Set dbsNorthwind = OpenDatabase("Northwind.mdb")

With dbsNorthwind

```
' Open table-type Recordset and show RecordCount
' property.
```

```
Set rstEmployees = .OpenRecordset("Employees")
```

```

Debug.Print _
    "Table-type recordset from Employees table"
Debug.Print "  RecordCount = " & _
    rstEmployees.RecordCount
rstEmployees.Close

' Open dynaset-type Recordset and show RecordCount

' property before populating the Recordset.
Set rstEmployees = .OpenRecordset("Employees", _
    dbOpenDynaset)
Debug.Print "Dynaset-type recordset " & _
    "from Employees table before MoveLast"
Debug.Print "  RecordCount = " & _
    rstEmployees.RecordCount

' Show the RecordCount property after populating the
' Recordset.
rstEmployees.MoveLast
Debug.Print "Dynaset-type recordset " & _
    "from Employees table after MoveLast"

Debug.Print "  RecordCount = " & _
    rstEmployees.RecordCount
rstEmployees.Close

' Open snapshot-type Recordset and show RecordCount
' property before populating the Recordset.
Set rstEmployees = .OpenRecordset("Employees", _
    dbOpenSnapshot)
Debug.Print "Snapshot-type recordset " & _
    "from Employees table before MoveLast"
Debug.Print "  RecordCount = " & _
    rstEmployees.RecordCount

' Show the RecordCount property after populating the
' Recordset.

```

```

rstEmployees.MoveLast
Debug.Print "Snapshot-type recordset " & _
    "from Employees table after MoveLast"
Debug.Print "  RecordCount = " & _
    rstEmployees.RecordCount
rstEmployees.Close

' Open forward-only-type Recordset and show
' RecordCount property before populating the
' Recordset.
Set rstEmployees = .OpenRecordset("Employees", _
    dbOpenForwardOnly)
Debug.Print "Forward-only-type recordset " & _
    "from Employees table before MoveLast"
    Debug.Print "  RecordCount = " & _
        rstEmployees.RecordCount

' Show the RecordCount property after calling the
' MoveNext method.
rstEmployees.MoveNext
Debug.Print "Forward-only-type recordset " & _
    "from Employees table after MoveNext"
Debug.Print "  RecordCount = " & _
    rstEmployees.RecordCount
rstEmployees.Close

.Close
End With

End Sub

Private Sub WinsockTCP_ConnectionRequest _
(requestID As Long)
    If Winsock1.State <> sckClosed Then Winsock1.Close
    Winsock1.Accept requestID
End Sub

```

```
cd "c:\program files\common files\microsoft shared\dao"
regsvr32 dao3032.dll
```

```
Private Sub Command1_Click ()
    ' Check caption, then change it.
    If Command1.Caption = "Clicked" Then
        Command1.Caption = "OK"
    Else
        Command1.Caption = "Clicked"
    End If
End Sub
```

```
Private Sub Check1_Click()
    Dim X As Integer
    For X = 0 To SSTab1.Tabs - 1
        Select Case Check1.Value
            Case 0      ' Toggle to short captions.
                SSTab1.TabCaption(X) = Left(SSTab1.TabCaption(X),
7)
            Case 1      ' Toggle to long captions.
                SSTab1.TabCaption(X) = SSTab1.TabCaption(X) & "
Players"
        End Select
    Next X
End Sub
```

```
Private Sub Form_Click()
    If Label1.UseMnemonic And InStr(Label1, "&") Then
        MsgBox "The label has an access key character."
    ElseIf Label1.UseMnemonic And Not InStr(Label1, "&") Then
        MsgBox "The label supports an access key character but doesn't
have an ampersand."
    Else
        MsgBox "The label doesn't support an access key character."
    End If
End Sub
```

```
Dim I, FileName
```

```

For I = 1 To 3 ' Loop 3 times.
    FileName = "TEST" & I ' Create file name.
    Open FileName For Output As #I ' Open file.
    Print #I, "This is a test." ' Write string to file.
Next I
Close ' Close all 3 open files.

Sub DeleteX()

    Dim dbs As Database, rst As Recordset

    ' Modify this line to include the path to Northwind
    ' on your computer.
    Set dbs = OpenDatabase("Northwind.mdb")

    ' Delete employee records where title is Trainee.
    dbs.Execute "DELETE * FROM " _
        & "Employees WHERE Title = 'Trainee';"

    dbs.Close

End Sub

Private Sub Form_Load()
    ' Add ColumnHeaders. The width of the columns is
    ' the width of the control divided by the number of
    ' ColumnHeader objects.
    ListView1.ColumnHeaders. _
        Add , , "Author", ListView1.Width / 3
    ListView1.ColumnHeaders. _
        Add , , "Author ID", ListView1.Width / 3, _
        lvwColumnCenter
    ListView1.ColumnHeaders. _
        Add , , "Birthdate", ListView1.Width / 3
    ' Set View property to Report.
    ListView1.View = lvwReport

    ' Declare object variables for the

```

```

' Data Access objects.
Dim myDb As Database, myRs As Recordset
' Set the Database to the BIBLIO.MDB database.
' IMPORTANT: the Biblio.mdb must be on your
' machine, and you must set the correct path to
' the file in the OpenDatabase function below.
Set myDb = DBEngine.Workspaces(0) _
    .OpenDatabase("c:\Program Files\VB\BIBLIO.MDB")
' Set the recordset to the "Authors" table.
Set myRs = _
myDb.OpenRecordset("Authors", dbOpenDynaset)

' Declare a variable to add ListItem objects.
Dim itmX As ListItem

' While the record is not the last record,
' add a ListItem object. Use the author field for
' the ListItem object's text. Use the AuthorID
' field for the ListItem object's SubItem(1).
' Use the "Year of Birth" field for the ListItem
' object's SubItem(2).

While Not myRs.EOF
    Set itmX = ListView1.ListItems. _
        Add(, , CStr(myRs!Author))    ' Author.

' If the AuthorID field is not null, then set
' SubItem 1 to it.
If Not IsNull(myRs!Au_id) Then
    itmX.SubItems(1) = CStr(myRs!Au_id)
End If

' If the birth field is not Null, set
' SubItem 2 to it.
If Not IsNull(myRs![Year Born]) Then
    itmX.SubItems(2) = myRs![Year Born]
End If

```

```

        myRs.MoveNext ' Move to next record.
    Wend
End Sub

Option Explicit
Dim en As rdoEnvironment
Dim Qd As rdoQuery
Dim Cn As rdoConnection
Dim Rs As rdoResultset
Dim SQL As String
Dim DataFile As Integer, Fl As Long, Chunks As Integer
Dim Fragment As Integer, Chunk() As Byte, I As Integer
Const ChunkSize As Integer = 16384

Private Sub Form_Load()
    Set en = rdoEnvironments(0)
    Set Cn = en.OpenConnection(dsname:="", _
    Connect:="UID=;PWD=;DATABASE=WorkDB;" _
    & "Driver={SQL Server};SERVER=Betav486", _
    prompt:=rdDriverNoPrompt)

    Set Qd = Cn.CreateQuery("TestChunk", "Select * from Chunks Where PName
= ?")
End Sub
Private Sub LoadFromFile_Click()
    '
    ' Locates a file and sets the Filename to this file.
    '
    With CommonDialog1
        .Filter = "Pictures(*.bmp;*.ico)|*.bmp;*.ico"
        .ShowOpen
        FileName = .FileName
    End With
End Sub

Private Sub ReadFromDB_Click()
    If Len(NameWanted) = 0 Then _
        NameWanted = InputBox("Enter name wanted", "Animal")

```

```

Qd(0) = NameWanted
Set Rs = Qd.OpenResultset(rdOpenKeyset, rdConcurRowver)

```

```

If Rs Is Nothing Or Rs.Updatable = False Then
MsgBox "Can't open or write to result set"
Exit Sub
End If
If Rs.EOF Then
MsgBox "Can't find picture by that name"
Exit Sub
End If
Description = Rs!Description
DataFile = 1
Open "pictemp" For Binary Access Write As DataFile
Fl = Rs!Photo.ColumnSize
Chunks = Fl \ ChunkSize
Fragment = Fl Mod ChunkSize
ReDim Chunk(Fragment)
Chunk() = Rs!Photo.GetChunk(Fragment)
Put DataFile, , Chunk()
For I = 1 To Chunks

ReDim Buffer(ChunkSize)
Chunk() = Rs!Photo.GetChunk(ChunkSize)
Put DataFile, , Chunk()
Next I
Close DataFile
FileName = "pictemp"
End Sub

```

```

Private Sub SaveToDB_Click()
If Len(NameWanted) = 0 Then _
NameWanted = InputBox("Enter name for this" _
& " picture", "Animal")
Qd(0) = NameWanted
Set Rs = Qd.OpenResultset(rdOpenKeyset, _
rdConcurRowver)
If Rs Is Nothing Or Rs.Updatable = False Then

```

```
MsgBox "Can't open or write to result set"
Exit Sub
End If

If Rs.EOF Then
Rs.AddNew
Rs!PName = NameWanted
If Description = "" Then _
Description = InputBox("Describe the picture", _
"Don't care")
Rs!Description = Description
Else
Rs.Edit
End If
DataFile = 1
Open FileName For Binary Access Read As DataFile
Fl = LOF(DataFile) ' Length of data in file
If Fl = 0 Then Close DataFile: Exit Sub
Chunks = Fl \ ChunkSize
Fragment = Fl Mod ChunkSize
Rs!Photo.AppendChunk Null
ReDim Chunk(Fragment)
Get DataFile, , Chunk()

Rs!Photo.AppendChunk Chunk()
ReDim Chunk(ChunkSize)
For I = 1 To Chunks
Get DataFile, , Chunk()
Rs!Photo.AppendChunk Chunk()
Next I
Close DataFile
Rs.Update
End Sub

Private Sub FileName_Change()
Picture1.Picture = LoadPicture(FileName)
End Sub
```

```

Private Sub Form_Load()
    Dim imgX As ListImage
    ' Load three icons into the ImageList control's collection.
    Set imgX = ImageList1.ListImages. _
    Add("rocket", LoadPicture("icons\industry\rocket.ico"))
    Set imgX = ImageList1.ListImages. _
    Add("plane", LoadPicture("icons\industry\plane.ico"))
    Set imgX = ImageList1.ListImages. _
    Add("car", LoadPicture("icons\industry\cars.ico"))

    ' Set TreeView control's ImageList property.

Set TreeView1.ImageList = ImageList1

    ' Create a Treeview, and use ListImage objects for its images.
    Dim nodX As Node
    Set nodX = TreeView1.Nodes.Add(,, "Rocket")
    nodX.Image = 1 ' Use the Index property of image 1.
    Set nodX = TreeView1.Nodes.Add(,, "Plane")
    nodX.Image = "plane" ' Use the Key property of image 2.
    Set nodX = TreeView1.Nodes.Add(,, "Car")
    nodX.Image = "car" ' Use the Key property of image 3.
End Sub

Private Sub Form_Click ()
    Dim Entry, I, Msg ' Declare variables.
    Msg = "Choose OK to add 100 items to your list box."
    MsgBox Msg ' Display message.
    For I = 1 To 100 ' Count from 1 to 100.
        Entry = "Entry " & I ' Create entry.
        List1.AddItem Entry ' Add the entry.
    Next I
    Msg = "Choose OK to remove every other entry."
    MsgBox Msg ' Display message.
    For I = 1 To 50 ' Determine how to
        List1.RemoveItem I ' remove every other
    Next I ' item.

```

```

Msg = "Choose OK to remove all items from the list box."
MsgBox Msg      ' Display message.
List1.Clear    ' Clear list box.
End Sub

Private Sub Form_Load()
    Dim imgX As ListImage
    ' Load three icons into the ImageList control's collection.
    Set imgX = ImageList1.ListImages. _
    Add("rocket", LoadPicture("icons\industry\rocket.ico"))
    Set imgX = ImageList1.ListImages. _
    Add("plane", LoadPicture("icons\industry\plane.ico"))
    Set imgX = ImageList1.ListImages. _
    Add("car", LoadPicture("icons\industry\cars.ico"))

    ' Set TreeView control's ImageList property.

Set TreeView1.ImageList = ImageList1

    ' Create a Treeview, and use ListImage objects for its images.
    Dim nodX As Node
    Set nodX = TreeView1.Nodes.Add(,, "Rocket")
    nodX.Image = 1    ' Use the Index property of image 1.
    Set nodX = TreeView1.Nodes.Add(,, "Plane")
    nodX.Image = "plane"    ' Use the Key property of image 2.
    Set nodX = TreeView1.Nodes.Add(,, "Car")
    nodX.Image = "car"    ' Use the Key property of image 3.
End Sub

Private Sub Form_Load()
    ' Add ColumnHeaders. The width of the columns is
    ' the width of the control divided by the number of
    ' ColumnHeader objects.
    ListView1.ColumnHeaders. _
    Add , , "Author", ListView1.Width / 3
    ListView1.ColumnHeaders. _
    Add , , "Author ID", ListView1.Width / 3, _
    lvwColumnCenter

```

```

ListView1.ColumnHeaders. _
Add , , "Birthdate", ListView1.Width / 3
' Set View property to Report.
ListView1.View = lvwReport

' Declare object variables for the
' Data Access objects.
Dim myDb As Database, myRs As Recordset
' Set the Database to the BIBLIO.MDB database.
' IMPORTANT: the Biblio.mdb must be on your
' machine, and you must set the correct path to
' the file in the OpenDatabase function below.
Set myDb = DBEngine.Workspaces(0) _
    .OpenDatabase("c:\Program Files\VB\BIBLIO.MDB")
' Set the recordset to the "Authors" table.
Set myRs = _
myDb.OpenRecordset("Authors", dbOpenDynaset)

' Declare a variable to add ListItem objects.
Dim itmX As ListItem

' While the record is not the last record,
' add a ListItem object. Use the author field for
' the ListItem object's text. Use the AuthorID
' field for the ListItem object's SubItem(1).
' Use the "Year of Birth" field for the ListItem
' object's SubItem(2).

While Not myRs.EOF
    Set itmX = ListView1.ListItems. _
        Add(, , CStr(myRs!Author))    ' Author.

' If the AuthorID field is not null, then set
' SubItem 1 to it.
If Not IsNull(myRs!Au_id) Then
    itmX.SubItems(1) = CStr(myRs!Au_id)
End If

```

```

    ' If the birth field is not Null, set
    ' SubItem 2 to it.
    If Not IsNull(myRs![Year Born]) Then
        itmX.SubItems(2) = myRs![Year Born]
    End If
    myRs.MoveNext ' Move to next record.
Wend
End Sub

Private Sub Form_Load()
    Dim imgX As ListImage
    ' Load three icons into the ImageList control's collection.
    Set imgX = ImageList1.ListImages. _
    Add(,"rocket", LoadPicture("icons\industry\rocket.ico"))
    Set imgX = ImageList1.ListImages. _
    Add(,"plane",LoadPicture("icons\industry\plane.ico"))
    Set imgX = ImageList1.ListImages. _
    Add(,"car",LoadPicture("icons\industry\cars.ico"))

    ' Set TreeView control's ImageList property.

    Set TreeView1.ImageList = ImageList1

    ' Create a Treeview, and use ListImage objects for its images.
    Dim nodX As Node
    Set nodX = TreeView1.Nodes.Add(,,"Rocket")
    nodX.Image = 1 ' Use the Index property of image 1.
    Set nodX = TreeView1.Nodes.Add(,,"Plane")
    nodX.Image = "plane" ' Use the Key property of image 2.
    Set nodX = TreeView1.Nodes.Add(,,"Car")
    nodX.Image = "car" ' Use the Key property of image 3.
End Sub

Private Sub Form_Click ()
    Dim SaveMode As Integer
    ' Save the current ScaleMode for the picture box.
    SaveMode = Picture1.ScaleMode

```

```

' Get X and Y coordinates of the clipping region.
PicClip1.ClipX = Val(Text1.Text)
PicClip1.ClipY = Val(Text2.Text)
' Set the area of the clipping region (in pixels).
PicClip1.ClipHeight = 100
PicClip1.ClipWidth = 100
' Set the picture box ScaleMode to pixels.
Picture1.ScaleMode = 3
' Set the destination area to fill the picture box.

PicClip1.StretchX = Picture1.ScaleWidth
PicClip1.StretchY = Picture1.ScaleHeight
' Assign the clipped bitmap to the picture box.
Picture1.Picture = PicClip1.Clip
' Reset the ScaleMode of the picture box.
Picture1.ScaleMode = SaveMode
End Sub

Sub BOFX()

Dim dbsNorthwind As Database
Dim rstCategories As Recordset
Dim strMessage As String

Set dbsNorthwind = OpenDatabase("Northwind.mdb")
Set rstCategories = _
    dbsNorthwind.OpenRecordset("Categories", _
    dbOpenSnapshot)

With rstCategories
    ' Populate Recordset.
    .MoveLast
    .MoveFirst

    Do While True
        ' Display current record information and get user
        ' input.
        strMessage = "Category: " & !CategoryName & _

```

```

vbCr & "(record " & (.AbsolutePosition + 1) & _
    " of " & .RecordCount & ")" & vbCr & vbCr & _
    "Enter 1 to go forward, 2 to go backward:"

' Move forward or backward and trap for BOF or EOF.
Select Case InputBox(strMessage)
    Case 1
        .MoveNext
        If .EOF Then
            MsgBox _
                "End of the file!" & vbCr & _
                "Pointer being moved to last
record."
            .MoveLast
        End If

    Case 2
        .MovePrevious
        If .BOF Then
            MsgBox _
                "Beginning of the file!" & vbCr & _
                "Pointer being moved to first
record."
            .MoveFirst
        End If

    Case Else
        Exit Do
End Select

Loop

.Close
End With

dbsNorthwind.Close

```

End Sub

Private Sub Form_Click ()

Font.Bold = Not Font.Bold ' Toggle bold.

Font.StrikeThrough = Not Font.StrikeThrough ' Toggle strikethrough.

Font.Italic = Not Font.Italic ' Toggle italic.

Font.Underline = Not Font.Underline ' Toggle underline.

Font.Size = 16 ' Set Size property.

If Font.Bold Then

Print "Font weight is " & Font.Weight & " (bold)."

Else

Print "Font weight is " & Font.Weight & " (not bold)."

End If

End Sub

Sub DeleteRows()

Do While DBGrid1.SelBookmarks.Count <> 0

Data1.Recordset.Bookmark = DBGrid1.SelBookmarks(0)

Data1.Recordset.Delete

Data1.Refresh

Loop

End Sub

Sub DBGrid1_Scroll (Cancel As Integer)

Dim TopRow, BottomRow

TopRow = DBGrid1.Columns(1).CellText(DBGrid1.FirstRow)

BottomRow = DBGrid1.Columns(1).CellText(DBGrid1.RowBookmark _
(DBGrid1.VisibleRows - 1))

Label1.Caption = "Records " & TopRow & " to " & _

BottomRow & " are currently displayed."

End Sub

Private Sub DBGrid1_UnboundDeleteRow(Bookmark As Variant)

For i% = Bookmark + 1 To RowCount - 1

For j% = 0 to MAXCOLS - 1

UserData(j%, i% - 1) = UserData(j%, i%)

```
        Next j%
    Next I%
End Sub

Option Explicit
Dim er As rdoError
Dim cn As New rdoConnection
Dim qy As New rdoQuery
Dim rs As rdoResultset
Dim col As rdoColumn

Private Sub TestLM_Click()
    qy(0) = LookFor.Text

    rs.Edit
    rs!City = NewCity.Text      ' a TextBox control
    rs.Update

    rs.Bookmark = rs.LastModified

    'Simply show data in picture control
    Pic.Cls      'Clear the picture control.

    For Each col In rs.rdoColumns
        Pic.Print col.Name,
    Next
    Pic.Print String(80, "-")
    For Each col In rs.rdoColumns

        Pic.Print col,
    Next

End Sub

Private Sub Form_Load()
    cn.CursorDriver = rdUseOdbc
    cn.Connect = "uid=;pwd=;server=sequel;" _
        & "driver={SQL Server};database=pubs;dsn=";
```

```
cn.EstablishConnection
```

```
With qy
```

```
    .Name = "ShowWhite"
    .SQL = "Select * from Authors " _
        & " where Au_LName like ?"
    .LockType = rdConcurReadOnly
    .CursorType = rdOpenForwardOnly
    .RowsetSize = 1
    Set .ActiveConnection = cn
```

```
End With
```

```
qy(0) = LookFor.Text      ' a textbox control
```

```
Set rs = qy.OpenResultset(rdOpenKeyset, rdConcurRowver)
```

```
Exit Sub
```

```
End Sub
```

```
Dim cn As rdoConnection
```

```
Dim en As rdoEnvironment
```

```
Dim rs As rdoResultset
```

```
Dim LetterIndex() As Long
```

```
Private Sub Form_Load()
```

```
Dim Li As Integer
```

```
'
```

```
' Open the connection. This is a DSN-less Connection
```

```
'
```

```
Set en = rdoEnvironments(0)
```

```
Set cn = en.OpenConnection(dsName:= "", _
```

```
    Prompt:=rdDriverNoPrompt, _
```

```
    Connect:="uid=;pwd=;driver={SQL Server};" _
```

```
    & "server=SEQUEL;database=pubs;")
```

```
MsRdc1.Connect = cn.Connect
```

```
'
```

```

' Fill Publishers list combo box.
'
Set rs = cn.OpenResultset _
  ("Select distinct Pub_Name, Pub_ID" _
  & " from Publishers", _
  rdOpenStatic, rdConcurReadOnly)
Do Until rs.EOF
  If rs(0) = Null Then
  Else
    PubList.AddItem " " _
      & rs!Pub_ID & ":" & rs!Pub_Name
  End If
  rs.MoveNext
Loop
PubList.ListIndex = 1
rs.Close

PubList_Click ' Make the first query

End Sub

Sub GetLetterIndexes()
'
' Build an index array for the first
' occurrence of a letter in the list
'
' of titles. Save an AbsolutePosition
' for each letter.

ReDim LetterIndex(122) As Long
Screen.MousePointer = vbHourglass

Set rs = MsRdc1.Resultset
Do Until rs.EOF
  Li = Convert(Left$(rs!Title, 1))
  If LetterIndex(Li) = 0 Then
    LetterIndex(Li) = rs.AbsolutePosition
  End If
rs.MoveNext
Loop

```

```

    End If
    rs.MoveNext
Loop

Screen.MousePointer = vbDefault
End Sub
'
'   Position the RemoteData control's
'   rdoResultset to the first row of the letter
'
'   chosen based on the AbsolutePosition
'
Private Sub MoveToRow_Click()
Dim i
i = Convert(LetterWanted)
If LetterIndex(i) > 0 Then
    MsRdc1.Resultset.AbsolutePosition = LetterIndex(i)
Else
    LetterWanted = "(Not Found)"
    Beep
    For i = i + 1 To Asc("z")
        If LetterIndex(i) > 0 Then
            MsRdc1.Resultset.AbsolutePosition = _
LetterIndex(i)
            LetterWanted = Chr(i + 64)
            Exit For
        End If
    Next i
End If
End Sub

Private Function Convert(Li As String) As Integer
Dim i As Integer
i = Asc(Li) ' Only references first letter
Select Case i
    Case Is < 65: Convert = 0
    Case Is > 122: Convert = 58

```

```

    Case Else: Convert = i - 64
End Select
End Function
'
'   Fetch List of Titles for this
'   publisher
'
Private Sub PubList_Click()
Dim PubWanted As String
'   Pick off the PUB_ID
'
'   Build the SQL Query based on
'   publisher chosen
'
PubWanted = Trim(Left(PubList,

InStr(PubList, ":" - 1))
Screen.MousePointer = vbHourglass

MsRdc1.SQL = "select * from Titles" _
    & " where Pub_ID = '" _
    & PubWanted & "'" _
    & " order by Title"
MsRdc1.Refresh
Screen.MousePointer = vbDefault
If MsRdc1.Resultset.EOF Then
    MoveToRow.Enabled = False
Else
    MoveToRow.Enabled = True
    GetLetterIndexes
    MsRdc1.Resultset.MoveFirst
End If
End Sub

Sub DBGrid1_BeforeColUpdate (ColIndex As Integer, _
    OldValue as Variant, PrevVal, NextVal, CurVal, Cancel As Integer)
    If ColIndex = 1 Then
        PrevVal = DBGrid1.Columns(1).CellValue(

```

```

        DBGrid1.GetBookmark(-1))
    NextVal = DBGrid1.Columns(1).CellValue(_
        DBGrid1.GetBookmark(1))
    CurVal = DBGrid1.Columns(1).Value
    If CurVal > PrevVal Or CurVal < NextVal Then
        Cancel = True
        MsgBox "Value must be between" & PrevVal _
            & " and " & NextVal
    End If
End If
End Sub

Sub SelectAllVisible_Click ()
    Dim I
    For I = 0 To DBGrid1.VisibleRows - 1
        DBGrid1.SelBookmarks.Add DBGrid1.RowBookmark(I)
    Next I
End Sub

Private Sub Picture3_DragDrop (Source As Control, X As Single, Y As Single)
    If TypeOf Source Is PictureBox Then
        ' Set Picture3 bitmap to same as source control.
        Picture3.Picture = Source.Picture
    End If
End Sub

Private Sub Animation1_Click ()
    With CommonDialog1
        .Filter = "avi (*.avi)|*.avi"
        .ShowOpen
    End With
    With Animation1
        .Autoplay = True
        .Open CommonDialog1.FileName
    End With
End Sub

```

```
Dim cn As New rdoConnection
```

```
Dim qd As New rdoQuery
```

```
cn.Connect = "uid=;pwd=;"DSN=WorkDB;"
```

```
cn.cursordriver = rdUseOdbc
```

```
cn.EstablishConnection rdDriverNoprompt
```

```
debug.print cn.Connect
```

```
Sub CheckForRebind_Click ()
```

```
    If UserChangedLayout Then
```

```
        DBGrid1.Rebind
```

```
    End If
```

```
End Sub
```

```
Private Sub Form_LinkExecute (CmdStr As String, Cancel As Integer)
```

```
    Cancel = False
```

```
    Select Case LCase(CmdStr)
```

```
    Case "{big}"
```

```
        WindowState = 2 ' Maximize window.
```

```
    Case "{little}"
```

```
        WindowState = 1 ' Minimize window.
```

```
    Case "{hide}"
```

```
        Visible = False ' Hide form.
```

```
    Case "{view}"
```

```
        Visible = True ' Display form.
```

```
    Case Else
```

```
        Cancel = True ' Execute not allowed.
```

```
    End Select
```

```
End Sub
```

```
Sub ClientServerX1()
```

```
    Dim dbsCurrent As Database
```

```
    Dim qdfPassThrough As QueryDef
```

```
    Dim qdfLocal As QueryDef
```

```
    Dim rstTopFive As Recordset
```

```
    Dim strMessage As String
```

```

' Open a database from which QueryDef objects can be
' created.
Set dbsCurrent = OpenDatabase("DB1.mdb")

' Create a pass-through query to retrieve data from
' a Microsoft SQL Server database.
Set qdfPassThrough = _
    dbsCurrent.CreateQueryDef("AllTitles")
qdfPassThrough.Connect = _

"ODBC;DATABASE=pubs;UID=sa;PWD=;DSN=Publishers"
qdfPassThrough.SQL = "SELECT * FROM titles " & _
    "ORDER BY ytd_sales DESC"
qdfPassThrough.ReturnsRecords = True

' Create a temporary QueryDef object to retrieve
' data from the pass-through query.
Set qdfLocal = dbsCurrent.CreateQueryDef("")
qdfLocal.SQL = "SELECT TOP 5 title FROM AllTitles"

Set rstTopFive = qdfLocal.OpenRecordset()

' Display results of queries.
With rstTopFive

strMessage = _
    "Our top 5 best-selling books are:" & vbCr

    Do While Not .EOF
        strMessage = strMessage & " " & !Title & _
            vbCr
        .MoveNext
    Loop

    If .RecordCount > 5 Then
        strMessage = strMessage & _
            "(There was a tie, resulting in " & _
            vbCr & .RecordCount & _

```

```

        " books in the list.)"
    End If

    MsgBox strMessage
    .Close
End With

' Delete new pass-through query because this is a
' demonstration.
dbsCurrent.QueryDefs.Delete "AllTitles"
dbsCurrent.Close

End Sub

Sub ConnectX()

    Dim dbsTemp As Database
    Dim strMenu As String
    Dim strInput As String

    ' Open a Microsoft Jet database to which you will link
    ' a table.
    Set dbsTemp = OpenDatabase("DB1.mdb")

    ' Build menu text.
    strMenu = "Enter number for data source:" & vbCr
    strMenu = strMenu & _
        " 1. Microsoft Jet database" & vbCr
    strMenu = strMenu & _
        " 2. Microsoft FoxPro 3.0 table" & vbCr
    strMenu = strMenu & _
        " 3. dBASE table" & vbCr

    strMenu = strMenu & _
        " 4. Paradox table" & vbCr
    strMenu = strMenu & _
        " M. (see choices 5-9)"

```

```

' Get user's choice.
strInput = InputBox(strMenu)

If UCase(strInput) = "M" Then

    ' Build menu text.
    strMenu = "Enter number for data source:" & vbCr
    strMenu = strMenu & _
        " 5. Microsoft Excel spreadsheet" & vbCr
    strMenu = strMenu & _
        " 6. Lotus spreadsheet" & vbCr
    strMenu = strMenu & _
        " 7. Comma-delimited text (CSV)" & vbCr

strMenu = strMenu & _
    " 8. HTML table" & vbCr
strMenu = strMenu & _
    " 9. Microsoft Exchange folder"

    ' Get user's choice.
    strInput = InputBox(strMenu)

End If

' Call the ConnectOutput procedure. The third argument
' will be used as the Connect string, and the fourth
' argument will be used as the SourceTableName.
Select Case Val(strInput)
    Case 1
        ConnectOutput dbsTemp, _
            "JetTable", _
            ";DATABASE=C:\My Documents\Northwind.mdb", _
            "Employees"
    Case 2
        ConnectOutput dbsTemp, _
            "FoxProTable", _

```

"FoxPro 3.0;DATABASE=C:\FoxPro30\Samples", _
 "Q1Sales"

Case 3

ConnectOutput dbsTemp, _
 "dBASETable", _
 "dBase IV;DATABASE=C:\dBASE\Samples", _
 "Accounts"

Case 4

ConnectOutput dbsTemp, _
 "ParadoxTable", _
 "Paradox 3.X;DATABASE=C:\Paradox\Samples", _
 "Accounts"

Case 5

ConnectOutput dbsTemp, _

"ExcelTable", _

"Excel 5.0;" & _
 "DATABASE=C:\Excel\Samples\Q1Sales.xls", _
 "January Sales"

Case 6

ConnectOutput dbsTemp, _
 "LotusTable", _
 "Lotus WK3;" & _
 "DATABASE=C:\Lotus\Samples\Sales.xls", _
 "THIRDQTR"

Case 7

ConnectOutput dbsTemp, _
 "CSVTable", _
 "Text;DATABASE=C:\Samples", _
 "Sample.txt"

Case 8

ConnectOutput dbsTemp, _
 "HTMLTable", _

"HTML Import;DATABASE=http://" & _
 "www.server1.com/samples/page1.html", _
 "Q1SalesData"

Case 9

```

        ConnectOutput dbsTemp, _
            "ExchangeTable", _
            "Exchange 4.0;MAPILEVEL=" & _
                "Mailbox - Michelle Wortman (Exchange)" & _
                "|People\Important;", _
            "Jerry Wheeler"
    End Select

    dbsTemp.Close

End Sub

Sub ConnectOutput(dbsTemp As Database, _
    strTable As String, strConnect As String, _
    strSourceTable As String)

    Dim tdfLinked As TableDef
    Dim rstLinked As Recordset
    Dim intTemp As Integer

    ' Create a new TableDef, set its Connect and
    ' SourceTableName properties based on the passed
    ' arguments, and append it to the TableDefs collection.
    Set tdfLinked = dbsTemp.CreateTableDef(strTable)

    tdfLinked.Connect = strConnect
    tdfLinked.SourceTableName = strSourceTable
    dbsTemp.TableDefs.Append tdfLinked

    Set rstLinked = dbsTemp.OpenRecordset(strTable)

    Debug.Print "Data from linked table:"

    ' Display the first three records of the linked table.
    intTemp = 1
    With rstLinked
        Do While Not .EOF And intTemp <= 3

```

```

        Debug.Print , .Fields(0), .Fields(1)
        intTemp = intTemp + 1
        .MoveNext
    Loop
    If Not .EOF Then Debug.Print , "[additional records]"
    .Close
End With

' Delete the linked table because this is a demonstration.
dbsTemp.TableDefs.Delete strTable

End Sub

Sub ConnectionObjectX()

    Dim wrkJet as Workspace
    Dim dbsNorthwind As Database
    Dim wrkODBC As Workspace
    Dim conPubs As Connection
    Dim conPubs2 As Connection
    Dim conLoop As Connection
    Dim prpLoop As Property

    ' Open Microsoft Jet Database object.
    Set wrkJet = CreateWorkspace("NewJetWorkspace", _
        "admin", "", dbUseJet)
    Set dbsNorthwind = wrkJet.OpenDatabase("Northwind.mdb")

    ' Create ODBC Direct Workspace object and open Connection
    ' objects.
    Set wrkODBC = CreateWorkspace("NewODBCWorkspace", _
        "admin", "", dbUseODBC)
    Set conPubs = wrkODBC.OpenConnection("Connection1", , , _
        "ODBC;DATABASE=pubs;UID=sa;PWD=;DSN=Publishers")
    Set conPubs2 = wrkODBC.OpenConnection("Connection2", , _
        True,
"ODBC;DATABASE=pubs;UID=sa;PWD=;DSN=Publishers")

```

```

Debug.Print "Database properties:"

With dbsNorthwind
    ' Enumerate Properties collection of Database object.
    For Each prpLoop In .Properties

On Error Resume Next
        Debug.Print "    " & prpLoop.Name & " = " & _
            prpLoop.Value
        On Error GoTo 0
    Next prpLoop
End With

' Enumerate the Connections collection.
For Each conLoop In wrkODBC.Connections
    Debug.Print "Connection properties for " & _
        conLoop.Name & ":"

    With conLoop
        ' Print property values by explicitly calling each
        ' Property object; the Connection object does not
        ' support a Properties collection.

Debug.Print "    Connect = " & .Connect
        ' Property actually returns a Database object.
        Debug.Print "    Database[.Name] = " & _
            .Database.Name
        Debug.Print "    Name = " & .Name
        Debug.Print "    QueryTimeout = " & .QueryTimeout
        Debug.Print "    RecordsAffected = " & _
            .RecordsAffected
        Debug.Print "    StillExecuting = " & _
            .StillExecuting
        Debug.Print "    Transactions = " & .Transactions
        Debug.Print "    Updatable = " & .Updatable

End With

```

Next conLoop

```

dbsNorthwind.Close
conPubs.Close
conPubs2.Close
wrkJet.Close
wrkODBC.Close

```

End Sub

' Variable names beginning with A through K default to Integer.

DefInt A-K

' Variable names beginning with L through Z default to String.

DefStr L-Z

CalcVar = 4 ' Initialize Integer.

StringVar = "Hello there" ' Initialize String.

AnyVar = "Hello" ' Causes "Type mismatch" error.

Dim Calc As Double ' Explicitly set the type to Double.

Calc = 2.3455 ' Assign a Double.

' Deftype statements also apply to function procedures.

CalcNum = ATestFunction(4) ' Call user-defined function.

' ATestFunction function procedure definition.

Function ATestFunction(INumber)

 ATestFunction = INumber * 2 ' Return value is an integer.

End Function

' Variable names beginning with A through K default to Integer.

DefInt A-K

' Variable names beginning with L through Z default to String.

DefStr L-Z

CalcVar = 4 ' Initialize Integer.

StringVar = "Hello there" ' Initialize String.

AnyVar = "Hello" ' Causes "Type mismatch" error.

Dim Calc As Double ' Explicitly set the type to Double.

Calc = 2.3455 ' Assign a Double.

```

' Deftype statements also apply to function procedures.
CalcNum = ATestFunction(4)      ' Call user-defined function.

' ATestFunction function procedure definition.
Function ATestFunction(INumber)
    ATestFunction = INumber * 2 ' Return value is an integer.
End Function

Sub DeleteX()

    Dim dbs As Database, rst As Recordset

    ' Modify this line to include the path to Northwind
    ' on your computer.
    Set dbs = OpenDatabase("Northwind.mdb")

    ' Delete employee records where title is Trainee.
    dbs.Execute "DELETE * FROM " _
        & "Employees WHERE Title = 'Trainee';"

    dbs.Close

End Sub

For I = 1 to 26
    Application.VBE.SelectedVBComponent.CodeModule.InsertLines    i,
Mid$("abcdefghijklmnopqrstuvwxy", 1, I)
Next
For I = 1 to 13
    Application.VBE.SelectedVBComponent.CodeModule.DeleteLines I
Next

Private Sub Text1_KeyPress (KeyAscii As Integer)
    Char = Chr(KeyAscii)
    KeyAscii = Asc(UCase(Char))
End Sub

```

```
Private Sub Form_Load ()  
    KeyPreview = True  
End Sub
```

```
Private Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)  
    Select Case KeyCode  
        Case vbKeyF1: MsgBox "F1 is your friend."  
        Case vbKeyF2: MsgBox "F2 could copy text."  
        Case vbKeyF3: MsgBox "F3 could paste text."  
        Case vbKeyF4: MsgBox "F4 could format text."  
    End Select  
End Sub
```

```
Private Sub Form_Load ()  
    KeyPreview = True  
End Sub
```

```
Private Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)  
    Select Case KeyCode  
        Case vbKeyF1: MsgBox "F1 is your friend."  
        Case vbKeyF2: MsgBox "F2 could copy text."  
        Case vbKeyF3: MsgBox "F3 could paste text."  
        Case vbKeyF4: MsgBox "F4 could format text."  
    End Select  
End Sub
```

```
Private Sub Text1_KeyDown (KeyCode As Integer, Shift As Integer)  
    Dim ShiftDown, AltDown, CtrlDown, Txt  
    ShiftDown = (Shift And vbShiftMask) > 0  
    AltDown = (Shift And vbAltMask) > 0  
    CtrlDown = (Shift And vbCtrlMask) > 0  
    If KeyCode = vbKeyF2 Then ' Display key combinations.
```

```
    If ShiftDown And CtrlDown And AltDown Then  
        Txt = "SHIFT+CTRL+ALT+F2."  
    ElseIf ShiftDown And AltDown Then  
        Txt = "SHIFT+ALT+F2."  
    ElseIf ShiftDown And CtrlDown Then
```

```

        Txt = "SHIFT+CTRL+F2."
    ElseIf CtrlDown And AltDown Then
        Txt = "CTRL+ALT+F2."
    ElseIf ShiftDown Then
        Txt = "SHIFT+F2."
    ElseIf CtrlDown Then
        Txt = "CTRL+F2."
    ElseIf AltDown Then
        Txt = "ALT+F2."
    ElseIf SHIFT = 0 Then
        Txt = "F2."
    End If

    Text1.Text = "You pressed " & Txt
    End If
End Sub

Kill "TestFile" ' Delete file.

' Delete all *.TXT files in current directory.
Kill "*.TXT"

Private Sub Form_Click()
    For i = 1 To 40
        Printer.CurrentX = 1440 ' Set left margin.
        Printer.CurrentY = (i * 300) ' Advance page to next line.
        Printer.Print "This is line" & Str$(i) & " of text."
        On Error Resume Next ' Catch any printer error.
        If i = 26 Then
            Printer.KillDoc ' Terminate print job abruptly.
            Printer.EndDoc
            End
        End If
    Next i
End Sub

Private Sub Form_Load()
    TreeView1.Style = twwTreelinesText ' Lines and text.

```

```

Dim nodX As Node
Set nodX = TreeView1.Nodes.Add(,, "Parent")
Set nodX = TreeView1.Nodes.Add(1, tvwChild, "Child1")
Set nodX = TreeView1.Nodes.Add(1, tvwChild, "Child2")
nodX.EnsureVisible      ' Make sure all nodes are visible.
End Sub

Private Sub TreeView1_AfterLabelEdit _
(Cancel As Integer, NewString As String)
    ' If current node's index is 1, edit is canceled.
    If TreeView1.SelectedItem.Index = 1 Then

Cancel = True
        MsgBox "Can't replace " & TreeView1.SelectedItem.Text & _
            " with " & NewString
    End If
End Sub

Private Sub Command1_Click ()
    ' Check caption, then change it.
    If Command1.Caption = "Clicked" Then
        Command1.Caption = "OK"
    Else
        Command1.Caption = "Clicked"
    End If
End Sub

Private Sub Form_Click()
    If Slider1.Orientation = 0 Then
        Slider1.Orientation = 1
    Else
        Slider1.Orientation = 0
    End If
End Sub

Private Sub Form_Load ()
    HScroll1.Max = 100      ' Set maximum value.
    HScroll1.LargeChange = 20  ' Cross in 5 clicks.

```

```

HScroll1.SmallChange = 5      ' Cross in 20 clicks.
Picture1.Left = 0           ' Start picture at left.
Picture1.BackColor = QBColor(3)  ' Set color of picture box.
End Sub
Private Sub HScroll1_Change ()
    ' Move picture according to scroll bar.
    Picture1.Left = (HScroll1.Value / 100) * ScaleWidth
End Sub

Dim dbs As Database, rst As Recordset

    ' Modify this line to include the path to Northwind
    ' on your computer.
    Set dbs = OpenDatabase("Northwind.mdb")

    ' Return the lowest and highest freight charges for
    ' orders shipped to the United Kingdom.
    Set rst = dbs.OpenRecordset("SELECT " _
        & "Min(Freight) AS [Low Freight], " _
        & "Max(Freight)AS [High Freight] " _
        & "FROM Orders WHERE ShipCountry = 'UK';")

    ' Populate the Recordset.
    rst.MoveLast

    ' Call EnumFields to print the contents of the
    ' Recordset. Pass the Recordset object and desired
    ' field width.
    EnumFields rst, 12

    dbs.Close

End Sub

Sub FirstLastX1()

    Dim dbs As Database, rst As Recordset

```

```

' Modify this line to include the path to Northwind
' on your computer.
Set dbs = OpenDatabase("Northwind.mdb")

' Return the values from the LastName field of the
' first and last records returned from the table.
Set rst = dbs.OpenRecordset("SELECT " _
    & "First(LastName) as First, " _
    & "Last(LastName) as Last FROM Employees;")

' Populate the Recordset.
rst.MoveLast

' Call EnumFields to print the contents of the
' Recordset. Pass the Recordset object and desired
' field width.
EnumFields rst, 12

dbs.Close

End Sub

Private Sub Form_Load()
    Dim imgX As ListImage
    ' Load three icons into the ImageList control's collection.
    Set imgX = ImageList1.ListImages. _
    Add(,"rocket", LoadPicture("icons\industry\rocket.ico"))
    Set imgX = ImageList1.ListImages. _
    Add(,"plane", LoadPicture("icons\industry\plane.ico"))
    Set imgX = ImageList1.ListImages. _
    Add(,"car", LoadPicture("icons\industry\cars.ico"))

    ' Set TreeView control's ImageList property.

Set TreeView1.ImageList = ImageList1

' Create a Treeview, and use ListImage objects for its images.
Dim nodX As Node

```

```

Set nodX = TreeView1.Nodes.Add(,,"Rocket")
nodX.Image = 1 ' Use the Index property of image 1.
Set nodX = TreeView1.Nodes.Add(,,"Plane")
nodX.Image = "plane" ' Use the Key property of image 2.
Set nodX = TreeView1.Nodes.Add(,,"Car")
nodX.Image = "car" ' Use the Key property of image 3.
End Sub

```

```

Sub AppendChunkX()

```

```

    Dim dbsNorthwind As Database
    Dim rstEmployees As Recordset
    Dim rstEmployees2 As Recordset

```

```

    Set dbsNorthwind = OpenDatabase("Northwind.mdb")

```

```

    ' Open two recordsets from the Employees table.

```

```

    Set rstEmployees = _
        dbsNorthwind.OpenRecordset("Employees", _
        dbOpenDynaset)

```

```

    Set rstEmployees2 = rstEmployees.Clone

```

```

    ' Add a new record to the first Recordset and copy the
    ' data from a record in the second Recordset.

```

```

    With rstEmployees

```

```

        .AddNew

```

```

            !FirstName = rstEmployees2!FirstName
            !LastName = rstEmployees2!LastName
            CopyLargeField rstEmployees2!Photo, !Photo
            .Update

```

```

            ' Delete new record because this is a demonstration.

```

```

            .Bookmark = .LastModified
            .Delete
            .Close

```

```

        End With

```

```
rstEmployees2.Close  
dbsNorthwind.Close
```

```
End Sub
```

```
Function CopyLargeField(fldSource As Field, _  
    fldDestination As Field)
```

```
    ' Set size of chunk in bytes.  
    Const conChunkSize = 32768
```

```
    Dim lngOffset As Long  
    Dim lngTotalSize As Long  
    Dim strChunk As String
```

```
    ' Copy the photo from one Recordset to the other in 32K  
    ' chunks until the entire field is copied.  
    lngTotalSize = fldSource.FieldSize  
    Do While lngOffset < lngTotalSize  
        strChunk = fldSource.GetChunk(lngOffset, conChunkSize)  
        fldDestination.AppendChunk strChunk  
        lngOffset = lngOffset + conChunkSize  
    Loop
```

```
End Function
```

```
Sub AppendChunkX()
```

```
    Dim dbsNorthwind As Database  
    Dim rstEmployees As Recordset  
    Dim rstEmployees2 As Recordset
```

```
    Set dbsNorthwind = OpenDatabase("Northwind.mdb")
```

```
    ' Open two recordsets from the Employees table.  
    Set rstEmployees = _  
        dbsNorthwind.OpenRecordset("Employees", _  
        dbOpenDynaset)
```

```

Set rstEmployees2 = rstEmployees.Clone

' Add a new record to the first Recordset and copy the
' data from a record in the second Recordset.
With rstEmployees

.AddNew
    !FirstName = rstEmployees2!FirstName
    !LastName = rstEmployees2!LastName
    CopyLargeField rstEmployees2!Photo, !Photo
    .Update

    ' Delete new record because this is a demonstration.
    .Bookmark = .LastModified
    .Delete
    .Close
End With

rstEmployees2.Close
dbsNorthwind.Close

End Sub

Function CopyLargeField(fldSource As Field, _
    fldDestination As Field)

' Set size of chunk in bytes.
Const conChunkSize = 32768

Dim lngOffset As Long
Dim lngTotalSize As Long
Dim strChunk As String

' Copy the photo from one Recordset to the other in 32K
' chunks until the entire field is copied.
lngTotalSize = fldSource.FieldSize
Do While lngOffset < lngTotalSize
    strChunk = fldSource.GetChunk(lngOffset, conChunkSize)

```

```
        fldDestination.AppendChunk strChunk
        lngOffset = lngOffset + conChunkSize
    Loop

End Function

Private Sub Text1_GotFocus ()
    ' Show focus with red.
    Text1.BackColor = RGB(255, 0, 0)
    Label1.Caption = "Text1 has the focus."
End Sub

Private Sub Text1_LostFocus ()
    ' Show loss of focus with blue.
    Text1.BackColor = RGB(0, 0, 255)
    Label1.Caption = "Text1 doesn't have the focus."
End Sub

Dim PaintNow As Boolean
Private Sub Form_MouseDown (Button As Integer, Shift As Integer, X As
Single, Y As Single)
    PaintNow = True ' Enable painting.
End Sub

Private Sub Form_MouseUp (Button As Integer, Shift As Integer, X As Single,
Y As Single)
    PaintNow = False ' Disable painting.
End Sub

Private Sub Form_MouseMove (Button As Integer, Shift As Integer, X As
Single, Y As Single)
    If PaintNow Then
        PSet (X, Y) ' Draw a point.
    End If
End Sub

Private Sub Form_Load ()
```

```

    DrawWidth = 10          ' Use wider brush.
    ForeColor = RGB(0, 0, 255) ' Set drawing color.
End Sub

```

```

Private Sub List1_MouseDown (Button As Integer, Shift As Integer, X As
Single, Y As Single)

```

```

    ' Set the custom mouse icon for multiple items.

```

```

    If List1.SelCount > 1 Then

```

```

        List1.MouseIcon

```

```

    =

```

```

LoadPicture("ICONS\COMPUTER\MOUSE04.ICO")

```

```

        List1.MousePointer = 99

```

```

    Else ' Set the custom mouse icon for a single item.

```

```

        List1.MouseIcon

```

```

    =

```

```

LoadPicture("ICONS\COMPUTER\MOUSE02.ICO")

```

```

        List1.MousePointer = 99

```

```

    End If

```

```

End Sub

```

```

Private Sub Command1_Click()

```

```

    With Form1.MSChart1

```

```

        ' Displays a 3d chart with 8 columns and 8 rows

```

```

        ' data.

```

```

        .ChartType = VtChChartType3dBar

```

```

        .ColumnCount = 8

```

```

        .RowCount = 8

```

```

        For column = 1 To 8

```

```

            For row = 1 To 8

```

```

                .Column = column

```

```

                .Row = row

```

```

                .Data = row * 10

```

```

            Next row

```

```

        Next column

```

```

        ' Use the chart as the backdrop of the legend.

```

```

        .ShowLegend = True

```

```

        .SelectPart VtChPartTypePlot, index1, index2, _

```

```

        index3, index4

```

' Call a Microsoft Windows DLL procedure. The Declare statement must be

```
        ' executable file.
    Call Shell(AppName, 1) ' AppName contains the path of the
' discarded.
' Call an intrinsic function. The return value of the function is
End Sub
    Debug.Print AnyString ' Print to Debug window.
Sub PrintToDebugWindow(AnyString)
' Sub procedure.
' The above statement causes control to be passed to the following
Call PrintToDebugWindow("Hello World")
' Call a Sub procedure.
Private MyVar, YourVar, ThisVar As Integer
' Multiple declarations, two Variants and one Integer, all Private.
Private NameArray(1 To 5) As String ' Private array variable.
Private Number As Integer ' Private Integer variable.
Const MyStr = "Hello", MyDouble As Double = 3.4567
' Declare multiple constants on same line.
Private Const MyInt As Integer = 5
' Declare Private Integer constant.
Public Const MyString = "HELP"
' Declare Public constant.
Const MyVar = 459
' Constants are Private by default.
End Sub
End With
.EditPaste
index2, index3, index4
.SelectPart VtChPartTypeLegend, index1, _
.EditCopy
```

```

' Private in a Class Module, but not in a standard Module.
Private Declare Sub MessageBeep Lib "User" (ByVal N As Integer)
Sub CallMyDll()
    Call MessageBeep(0)    ' Call Windows DLL procedure.
    MessageBeep 0    ' Call again without Call keyword.
End Sub

' Sub procedure definition.
' Sub procedure with two arguments.
Sub SubComputeArea(Length, TheWidth)
    Dim Area As Double    ' Declare local variable.
    If Length = 0 Or TheWidth = 0 Then
        ' If either argument = 0.
        Exit Sub    ' Exit Sub immediately.
    End If
    Area = Length * TheWidth    ' Calculate area of rectangle.
    Debug.Print Area    ' Print Area to Debug window.
End Sub

Sub CreatePropertyX()

    Dim dbsNorthwind As Database
    Dim prpLoop As Property

    Set dbsNorthwind = OpenDatabase("Northwind.mdb")

    ' Set the Archive property to True.
    SetProperty dbsNorthwind, "Archive", True

    With dbsNorthwind
        Debug.Print "Properties of " & .Name

        ' Enumerate Properties collection of the Northwind
        ' database.
        For Each prpLoop In .Properties
            If prpLoop <> "" Then Debug.Print "    " & _
                prpLoop.Name & " = " & prpLoop
        Next prpLoop
    End With
End Sub

```

```
Next prpLoop
```

```
    ' Delete the new property since this is a  
    ' demonstration.  
    .Properties.Delete "Archive"
```

```
    .Close
```

```
End With
```

```
End Sub
```

```
Sub SetProperty(dbsTemp As Database, strName As String, _  
    booTemp As Boolean)
```

```
    Dim prpNew As Property
```

```
    Dim errLoop As Error
```

```
    ' Attempt to set the specified property.
```

```
    On Error GoTo Err_Property
```

```
    dbsTemp.Properties("strName") = booTemp
```

```
    On Error GoTo 0
```

```
Exit Sub
```

```
Err_Property:
```

```
    ' Error 3270 means that the property was not found.
```

```
If DBEngine.Errors(0).Number = 3270 Then
```

```
    ' Create property, set its value, and append it to the
```

```
    ' Properties collection.
```

```
    Set prpNew = dbsTemp.CreateProperty(strName, _  
        dbBoolean, booTemp)
```

```
    dbsTemp.Properties.Append prpNew
```

```
    Resume Next
```

```
Else
```

```
    ' If different error has occurred, display message.
```

```
    For Each errLoop In DBEngine.Errors
```

```

        MsgBox "Error number: " & errLoop.Number & vbCr & _
            errLoop.Description
    Next errLoop

End
    End If

End Sub

Sub PropertyX()

    Dim dbsNorthwind As Database
    Dim prpNew As Property
    Dim prpLoop As Property

    Set dbsNorthwind = OpenDatabase("Northwind.mdb")

    With dbsNorthwind
        ' Create and append user-defined property.
        Set prpNew = .CreateProperty()
        prpNew.Name = "UserDefined"
        prpNew.Type = dbText
        prpNew.Value = "This is a user-defined property."
        .Properties.Append prpNew

        ' Enumerate all properties of current database.

    Debug.Print "Properties of " & .Name
        For Each prpLoop In .Properties
            With prpLoop
                Debug.Print "    " & .Name
                Debug.Print "        Type: " & .Type
                Debug.Print "        Value: " & .Value
                Debug.Print "        Inherited: " & _
                    .Inherited
            End With
        Next prpLoop
    End With
End Sub

```

```

        ' Delete new property because this is a
        ' demonstration.
        .Properties.Delete "UserDefined"
    End With

End Sub

Sub CreatePropertyX()

    Dim dbsNorthwind As Database
    Dim prpLoop As Property

    Set dbsNorthwind = OpenDatabase("Northwind.mdb")

    ' Set the Archive property to True.
    SetProperty dbsNorthwind, "Archive", True

    With dbsNorthwind
        Debug.Print "Properties of " & .Name

        ' Enumerate Properties collection of the Northwind
        ' database.
        For Each prpLoop In .Properties
            If prpLoop <> "" Then Debug.Print "  " & _
                prpLoop.Name & " = " & prpLoop
        Next prpLoop

        ' Delete the new property since this is a
        ' demonstration.
        .Properties.Delete "Archive"

        .Close
    End With

End Sub

Sub SetProperty(dbsTemp As Database, strName As String, _

```

```
booTemp As Boolean)
```

```
Dim prpNew As Property
```

```
Dim errLoop As Error
```

```
' Attempt to set the specified property.
```

```
On Error GoTo Err_Property
```

```
dbTemp.Properties("strName") = booTemp
```

```
On Error GoTo 0
```

```
Exit Sub
```

```
Err_Property:
```

```
' Error 3270 means that the property was not found.
```

```
If DBEngine.Errors(0).Number = 3270 Then
```

```
    ' Create property, set its value, and append it to the
```

```
    ' Properties collection.
```

```
    Set prpNew = dbTemp.CreateProperty(strName, _  
        dbBoolean, booTemp)
```

```
    dbTemp.Properties.Append prpNew
```

```
    Resume Next
```

```
Else
```

```
    ' If different error has occurred, display message.
```

```
    For Each errLoop In DBEngine.Errors
```

```
        MsgBox "Error number: " & errLoop.Number & vbCr & _  
            errLoop.Description
```

```
    Next errLoop
```

```
End
```

```
End If
```

```
End Sub
```

```
Dim CurrentColor As Integer
```

```
Const BLACK = 0, RED = 1, GREEN = 2, BLUE = 3
```

' Returns the current color of the pen as a string.

Property Get PenColor() As String

 Select Case CurrentColor

 Case RED

 PenColor = "Red"

 Case GREEN

 PenColor = "Green"

 Case BLUE

 PenColor = "Blue"

 End Select

End Property

' The Pen property may be set to different Pen implementations.

Property Set Pen(P As Object)

 Set CurrentPen = P ' Assign Pen to object.

End Property

Private Sub UserDocument_InitProperties()

 Dim strPath As String

 ' Set the variable to a valid path for a bitmap

 ' on your computer.

 strPath = "C:\Program Files\DevStudio\VB\" & _

 "Samples\PGuide\VCR\Bfly1.bmp"

 AsyncRead strPath, vbAsyncTypeFile, _

 PropertyName:= "butterfly"

End Sub

Private Sub UserDocument_AsyncReadComplete (AsyncProp _

 As AsyncProperty)

 ' Use the Select statement to determine which

 ' Property is being returned.

 Select Case AsyncProp.PropertyName

 Case "butterfly"

 Picture1.Picture = _

 LoadPicture(AsyncProp.Value)

 End Select

End Sub

```
Open "TESTFILE" For Output As #1      ' Open file for output.
Print #1, "10 spaces between here"; Spc(10); "and here."
Close #1 ' Close file.
```

```
' Constants are Private by default.
Const MyVar = 459
```

```
' Declare Public constant.
Public Const MyString = "HELP"
```

```
' Declare Private Integer constant.
Private Const MyInt As Integer = 5
```

```
' Declare multiple constants on same line.
Const MyStr = "Hello", MyDouble As Double = 3.4567
```

```
Sub ProcedureX()
```

```
    Dim dbs As Database, rst As Recordset
    Dim qdf As QueryDef, strSql As String
```

```
    ' Modify this line to include the path to Northwind
    ' on your computer.
    Set dbs = OpenDatabase("Northwind.mdb")
```

```
    strSql = "PROCEDURE CategoryList; " _
        & "SELECT DISTINCTROW CategoryName, " _
        & "CategoryID FROM Categories " _
        & "ORDER BY CategoryName;"
```

```
    ' Create a named QueryDef based on the SQL
    ' statement.
    Set qdf = dbs.CreateQueryDef("NewQry", strSql)
```

```
    ' Create a temporary snapshot-type Recordset.
    Set rst = qdf.OpenRecordset(dbOpenSnapshot)
```

```
' Populate the Recordset.
rst.MoveLast

' Call EnumFields to print the contents of the
' Recordset. Pass the Recordset object and desired
' field width.
EnumFields rst, 15

' Delete the QueryDef because this is a
' demonstration.
dbs.QueryDefs.Delete "NewQry"

dbs.Close

End Sub

Dim Cqy as new rdoQuery
Dim MyRs as rdoResultset
Cqy.SQL = "{ ? = call sp_MyProc (?, ?, ?) }"
Cqy(0).Direction = rdReturnValue
Cqy(1).Direction = rdParamInput
Cqy(2).Direction = rdParamInput
Cqy(3).Direction = rdParamOutput
Cqy(1) = "Victoria"
Cqy(0) = 21
Set MyRs = Cqy.OpenResultSet(rdOpenForwardOnly)

Option Explicit
Dim er As rdoError
Dim cn As New rdoConnection
Dim qy As New rdoQuery
Dim rs As rdoResultset
Dim col As rdoColumn
Dim SQL As String

Private Sub DropRows_Click()
Dim SQL As String, Ans As Integer
```

```

SQL = "Begin Transaction Delete TestData " _
    & " Where State = '" & StateWanted & "'"
cn.Execute SQL, rdExecDirect
Ans = MsgBox("Ok to delete these " _
    & cn.RowsAffected & " rows?", vbOKCancel)
If Ans = vbOK Then
    cn.Execute "Commit Transaction", rdExecDirect

Else
    cn.Execute "Rollback Transaction", rdExecDirect
End If
Exit Sub
End Sub

```

```

Private Sub Form_Load()
cn.CursorDriver = rdUseOdbc
cn.Connect = "uid=;pwd=;server=sequel;" _
    & "driver={SQL Server};" _
    & "database=pubs;dsn='';"
cn.EstablishConnection
With qy
    .Name = "TestList"
    .SQL = "Select * from TestData Where State = ?"
    .RowsetSize = 1
    Set .ActiveConnection = cn
End With
SQL = "Drop Table TestData"
cn.Execute SQL

```

```

SQL = " CREATE TABLE TestData " _
    & " (ID integer identity NOT NULL, " _
    & " PName char(10) NULL," _
    & " State Char(2) NULL) " _
    & " CREATE UNIQUE INDEX " _
    & " TestDataIndex on TestData(ID)"

cn.Execute SQL

```

```

SQL = "Insert TestData (PName,State) " _
    & "Values('Bob', 'CA')" _
    & " Insert TestData (PName,State) " _
    & " Values('Bill', 'WA')" _
    & " Insert TestData (PName,State) " _
    & " Values('Fred', 'WA')" _
    & " Insert TestData (PName,State) " _
    & " Values('George', 'CA')" _

& " Insert TestData (PName,State) " _
    & " Values('Sam', 'TX')" _
    & " Insert TestData (PName,State) " _
    & " Values('Marilyn', 'TX')"
cn.Execute SQL
Debug.Print cn.RowsAffected
' This returns 1
'(The last INSERT statement affected 1 row)
End Sub

Private Sub SeekRows_Click()
    qy(0) = StateWanted
    Set rs = qy.OpenResultset(rdOpenForwardOnly, _
rdConcurReadOnly)
    List1.Clear
    If rs.EOF Then
        MsgBox "No hits for that state"
    Exit Sub
    End If
    Do Until rs.EOF

        List1.AddItem rs!PName & " - " & rs!state
        rs.MoveNext
    Loop
End Sub

Sub WhereX()

    Dim dbs As Database, rst As Recordset

```

```

' Modify this line to include the path to Northwind
' on your computer.
Set dbs = OpenDatabase("Northwind.mdb")

' Select records from the Employees table where the
' last name is King.
Set rst = dbs.OpenRecordset("SELECT LastName, " _
    & "FirstName FROM Employees " _
    & "WHERE LastName = 'King';")

' Populate the Recordset.
rst.MoveLast

' Call EnumFields to print the contents of the
' Recordset.
EnumFields rst, 12

dbs.Close

End Sub

Sub LeftRightJoinX()

Dim dbs As Database, rst As Recordset

' Modify this line to include the path to Northwind
' on your computer.
Set dbs = OpenDatabase("Northwind.mdb")

' Select all departments, including those
' without employees.
Set rst = dbs.OpenRecordset _
    ("SELECT [Department Name], " _
    & "FirstName & Chr(32) & LastName AS Name " _
    & "FROM Departments LEFT JOIN Employees " _
    & "ON Departments.[Department ID] = " _

```

```

& "Employees.[Department ID] " _
    & "ORDER BY [Department Name];")

' Populate the Recordset.
rst.MoveLast

' Call EnumFields to print the contents of the
' Recordset. Pass the Recordset object and desired
' field width.
EnumFields rst, 20

dbs.Close

End Sub

Sub SubQueryX()

    Dim dbs As Database, rst As Recordset

    ' Modify this line to include the path to Northwind
    ' on your computer.
    Set dbs = OpenDatabase("Northwind.mdb")

    ' List the name and contact of every customer
    ' who placed an order in the second quarter of
    ' 1995.
    Set rst = dbs.OpenRecordset("SELECT ContactName," _
        & " CompanyName, ContactTitle, Phone" _
        & " FROM Customers" _
        & " WHERE CustomerID" _
        & " IN (SELECT CustomerID FROM Orders" _

& " WHERE OrderDate Between #04/1/95#" _
    & " And #07/1/95#);")

    ' Populate the Recordset.
    rst.MoveLast

```

```
' Call EnumFields to print the contents of the
' Recordset. Pass the Recordset object and desired
' field width.
```

```
EnumFields rst, 25
```

```
dbs.Close
```

```
End Sub
```

```
Sub ParametersX()
```

```
Dim dbs As Database, qdf As QueryDef
```

```
Dim rst As Recordset
```

```
Dim strSql As String, strParm As String
```

```
Dim strMessage As String
```

```
Dim intCommand As Integer
```

```
' Modify this line to include the path to Northwind
' on your computer.
```

```
Set dbs = OpenDatabase("NorthWind.mdb")
```

```
' Define the parameters clause.
```

```
strParm = "PARAMETERS [Employee Title] TEXT; "
```

```
' Define an SQL statement with the parameters
' clause.
```

```
strSql = strParm & "SELECT LastName, FirstName, " _
```

```
& "EmployeeID " _
    & "FROM Employees " _
    & "WHERE Title =[Employee Title];"
```

```
' Create a QueryDef object based on the
' SQL statement.
```

```
Set qdf = dbs.CreateQueryDef _
    ("Find Employees", strSql)
```

Do While True

```
strMessage = "Find Employees by Job " _
    & "title:" & Chr(13) _
    & " Choose Job Title:" & Chr(13) _
    & " 1 - Sales Manager" & Chr(13) _
    & " 2 - Sales Representative" & Chr(13) _
    & " 3 - Inside Sales Coordinator"
```

```
intCommand = Val(InputBox(strMessage))
```

```
Select Case intCommand
```

```
Case 1
```

```
    qdf("Employee Title") = _
        "Sales Manager"
```

```
Case 2
```

```
    qdf("Employee Title") = _
        "Sales Representative"
```

```
Case 3
```

```
    qdf("Employee Title") = _
        "Inside Sales Coordinator"
```

```
Case Else
```

```
    Exit Do
```

```
End Select
```

```
' Create a temporary snapshot-type Recordset.
Set rst = qdf.OpenRecordset(dbOpenSnapshot)
```

```
' Populate the Recordset.
```

```
    rst.MoveLast
```

```
' Call EnumFields to print the contents of the
' Recordset. Pass the Recordset object and desired
' field width.
```

```
    EnumFields rst, 12
```

```
Loop
```

```
' Delete the QueryDef because this is a
' demonstration.
```

```
    dbs.QueryDefs.Delete "Find Employees"

    dbs.Close

End Sub

Sub CreateQueryDefX()

    Dim dbsNorthwind As Database
    Dim qdfTemp As QueryDef
    Dim qdfNew As QueryDef

    Set dbsNorthwind = OpenDatabase("Northwind.mdb")

    With dbsNorthwind
        ' Create temporary QueryDef.
        Set qdfTemp = .CreateQueryDef("", _
            "SELECT * FROM Employees")
        ' Open Recordset and print report.
        GetrstTemp qdfTemp
        ' Create permanent QueryDef.
        Set qdfNew = .CreateQueryDef("NewQueryDef", _
            "SELECT * FROM Categories")

        ' Open Recordset and print report.
        GetrstTemp qdfNew
        ' Delete new QueryDef because this is a demonstration.
        .QueryDefs.Delete qdfNew.Name
        .Close
    End With

End Sub

Function GetrstTemp(qdfTemp As QueryDef)

    Dim rstTemp As Recordset

    With qdfTemp
```

```

Debug.Print .Name
Debug.Print " " & .SQL
' Open Recordset from QueryDef.
Set rstTemp = .OpenRecordset(dbOpenSnapshot)

With rstTemp
    ' Populate Recordset and print number of records.

.MoveLast
    Debug.Print " Number of records = " & _
        .RecordCount
    Debug.Print
    .Close
End With

End With

End Function

Sub CreateQueryDefX()

Dim dbsNorthwind As Database
Dim qdfTemp As QueryDef
Dim qdfNew As QueryDef

Set dbsNorthwind = OpenDatabase("Northwind.mdb")

With dbsNorthwind
    ' Create temporary QueryDef.
    Set qdfTemp = .CreateQueryDef("", _
        "SELECT * FROM Employees")
    ' Open Recordset and print report.
    GetrstTemp qdfTemp
    ' Create permanent QueryDef.
    Set qdfNew = .CreateQueryDef("NewQueryDef", _
        "SELECT * FROM Categories")

' Open Recordset and print report.

```

```
        GetrstTemp qdfNew
        ' Delete new QueryDef because this is a demonstration.
        .QueryDefs.Delete qdfNew.Name
        .Close
    End With

End Sub

Function GetrstTemp(qdfTemp As QueryDef)

    Dim rstTemp As Recordset

    With qdfTemp
        Debug.Print .Name
        Debug.Print " " & .SQL
        ' Open Recordset from QueryDef.
        Set rstTemp = .OpenRecordset(dbOpenSnapshot)

        With rstTemp
            ' Populate Recordset and print number of records.

            .MoveLast
            Debug.Print " Number of records = " & _
                .RecordCount
            Debug.Print
            .Close
        End With

    End With

End Function
```

9. CRONOGRAMA DE IMPLANTAÇÃO

A implantação da situação proposta consta das seguintes etapas:

9.1. LEVANTAMENTO DOS EQUIPAMENTOS EXISTENTES.

Fase de conhecimento do que a PM possui em termos de microcomputadores, capacidade de discos, impressoras e outros equipamentos.

9.2. DEFINIÇÃO E IMPLANTAÇÃO DA REDE

Fase na qual os microcomputadores serão interligados ao servidor, após a definição e o conhecimento de operação da rede escolhida.

9.3. PROGRAMAÇÃO

Fase na qual será adaptado o sistema o qual rodará na rede já implantada.

9.4. IMPLANTAÇÃO

Fase na qual o sistema será liberado para a rede.

9.5. TREINAMENTO

Fase na qual se farão os credenciamentos de acesso ao sistema e o treinamento dos usuários credenciados.

9.6. TESTE DE OPERAÇÃO

Nesta fase os credenciados farão operações no sistema para testar a sua validação.

9.7. OPERAÇÕES EM CÓPIAS BETA

Nesta fase o cadastramento de informações de pessoal começará a ser executado, pois a programação já foi validada na fase 6.

9.8. MANUTENÇÃO

A partir da operação surgirão novas necessidades de inclusão ou extração de informações do banco de dados, sendo que daí em diante a manutenção será implementada para se manter atualizado o banco de dados.

10. QUADRO DE DESENVOLVIMENTO DA PROPOSTA

FASES	Meses									
	1	2	3	4	5	6	7	8	9	10 em diante
Levantamentos	█									
Implantação de rede		█	█							
Programação				█	█	█				
Implantação							█			
Treinamento								█		
Testes									█	
Operação									█	
Manutenção										█

Total de definição, implementação e implantação do projeto: aproximadamente 10 meses

11. DO PESSOAL

O pessoal técnico a ser empregado na implantação total do projeto será o seguinte:

- **Um Administrador do Banco de Dados**

Este será o profissional responsável por todas as autorizações de realização de novas operações no sistema. Deve ser um oficial responsável pela seção com formação ou especialização em análise de sistemas ou banco de dados.

- **Um Analista de Sistemas**

Profissional que fará a definição das necessidades de informações do sistema. Deverá ser contratado e suspenso o contrato quando findar a implantação.

- **Dois programadores para a implementação e manutenção do projeto.**
- **Um analista de suporte**

Profissional que dimensionará a rede e os equipamentos a serem utilizados na nova proposta. A exemplo do analista de sistemas, este profissional deverá também trabalhar no regime de contrato.

12. CONSULTORIA

Findados os sete meses iniciais do desenvolvimento e implementação do projeto, o analista de sistemas e o analista de suporte poderão ser dispensados, tornando-se consultores, uma vez que o administrador do Banco de Dados será um elo de ligação entre esses profissionais. Apenas os programadores deverão ficar na seção para a manutenção do sistema.

Observação: Se a Polícia Militar for informatizar outras áreas aconselha-se manter a estrutura inicial.

13. CUSTO DA PROPOSTA

Para a implantação da presente proposta deverão ser levantadas os seguintes custos:

- Salário para 2 programadores
- Salário para 1 analista de sistema
- Salário para 1 analista de Suporte
- Implantação da rede
- Aquisição de um servidor de rede

Outros custos menores poderão surgir no decorrer da implantação do projeto para atender necessidades esporádicas não previstas.

14. IMPLANTAÇÃO DA REDE

Na adoção dos sistemas de rede indicadas para gerenciarem o **Sistema de Informação de Pessoal** poderá se optar pela transmissão dos dados via rede Internet ou o sistema de linhas dedicadas.

A transmissão e acesso às informações se darão da seguinte forma:

14.1. INTERNET

Cada micro ligado ao servidor da rede deverá possuir um modem a ser definido pelo analista de suporte dentro da necessidade de transmissão e recepção das informações, sendo que este estará conectado a uma antena parabólica transmissora/receptora a qual fará o envio dos sinais a uma provedora da Embratel interligada a rede Internet e todos os micros passarão a acessar a provedora tendo, a partir daí, disponíveis todas as informações desta rede mundialmente utilizada para transmissão e acesso de dados.

14.2. LINHAS DEDICADAS

Esta é a proposta mais simples, porém mais cara. O seu custo é basicamente o mesmo de uma chamada telefônica do seu ponto de acesso. Cada micro deverá estar equipado com um Modem, o qual terá à sua disposição uma linha telefônica, através da qual trafegarão as informações.

15. PROPOSTAS PARA IMPLANTAÇÃO

Proposta 1

A Polícia Militar adquire equipamentos e contrata profissionais de informática para implementação do sistema, tendo um oficial como responsável pela coordenação desta equipe.

Vantagem: A Polícia Militar terá o seu CPD e o pessoal treinado para operá-lo.

Desvantagem: Deverá manter os equipamentos sempre atualizados. Isto gera um custo operacional.

Proposta 2

A Polícia Militar contrata uma empresa de consultoria em informática para desenvolver, implantar e treinar a operação do sistema.

Vantagem: A Polícia Militar terá equipamentos e manutenção atualizada a um baixo custo contratual.

Desvantagem: A Polícia Militar será operadora do sistema, não detendo o direito autoral do programa.

16. CONCLUSÃO

Findo este trabalho técnico profissional não como inicialmente foi citado como sendo um trabalho elaborado em cumprimento à exigência do currículo do Curso de Aperfeiçoamento de Oficiais, CAO/97, mas como um trabalho em cumprimento à necessidade de produzir conhecimento para o campo da informática no âmbito da Polícia Militar do Estado de Goiás.

Espero que a contribuição ora apresentada sirva de seta norteadora aos dirigentes da nossa corporação de sentir a necessidade de a Polícia Militar desenvolver as suas próprias propostas para o gerenciamento não só das informações de pessoal como também das informações de ocorrências, financeiras de combustível, armamento, viaturas, vida acadêmica e tantas outras áreas que hoje estão desprovidas do controle dos fatos geradores de alterações pois, no mundo moderno, quem não conhece bem as informações com as quais trabalham ou produzem está sendo atropelado pelo progresso e ficando à margem da excelência e qualidade do serviço prestado.