



# Universidade Norte do Paraná

---

SISTEMA DE ENSINO PRESENCIAL CONECTADO  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E  
DESENVOLVIMENTO DE SISTEMAS

EDINILTON PEREIRA DO NASCIMENTO JÚNIOR

**PROJETO – TCC**

**SISTEMA DE CONTROLE DE FUNCIONÁRIOS DA PMERJ COM  
FORMULÁRIOS DE PARTE E REQUERIMENTO**

---

TERESÓPOLIS  
2011

**EDINILTON PEREIRA DO NASCIMENTO JÚNIOR**

**PROJETO – TCC**

**SISTEMA DE CONTROLE DE FUNCIONÁRIOS DA PMERJ COM  
RELATÓRIOS DE PARTE E REQUERIMENTO**

Trabalho apresentado ao Curso Superior de Tecnologia  
em Análise e Desenvolvimento de Sistemas da UNOPAR  
- Universidade Norte do Paraná, para a disciplina Estágio  
Curricular Obrigatório II - TCC.  
Modulo VI.

Orientador: Cristiane Del Grossi Moreira  
Professor Supervisor: Rodrigo Otávio do Canto Braga

**TERESÓPOLIS**

**2011**

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>3</b>
<b>2 JUSTIFICATIVA.....</b>	<b>4</b>
<b>3 OBJETIVOS.....</b>	<b>4</b>
3.1 OBJETIVO GERAL.....	4
3.2 OBJETIVO ESPECÍFICO.....	4
<b>4 REFERENCIAL TEÓRICO.....</b>	<b>5</b>
4.1 SISTEMAS DE INFORMAÇÃO.....	5
4.2 TECNOLOGIA DA INFORMAÇÃO.....	6
4.3 TECNOLOGIA DA INFORMAÇÃO NO SETOR PÚBLICO.....	7
4.4 CULTURA, POLÍTICA E ESTRUTURA DAS ORGANIZAÇÕES PÚBLICAS.....	8
<b>5 METODOLOGIA.....</b>	<b>9</b>
5.1 REQUISITOS DE HARDWARE E SOFTWARE NECESSÁRIOS.....	10
5.1.1 Hardware.....	10
5.1.2 Software.....	10
<b>6 CARACTERIZAÇÃO DA ORGANIZAÇÃO EM ESTUDO.....</b>	<b>10</b>
<b>7 TECNOLOGIA.....</b>	<b>11</b>
7.1 REQUISITOS DE HARDWARE E SOFTWARE NECESSÁRIOS.....	12
7.1.1 Hardware.....	12
7.1.2 software.....	13
<b>8 LEVANTAMENTO DE REQUISITOS.....</b>	<b>13</b>
<b>9 DIAGRAMA DE CASO DE USO.....</b>	<b>13</b>
<b>10 DIAGRAMA DE CLASSE.....</b>	<b>15</b>
<b>11 DIAGRAMA DE SEQUÊNCIA.....</b>	<b>16</b>
11.1 DIAGRAMA DE LOGIN.....	16
11.2 DIAGRAMA DE PRAÇAS E OFICIAIS.....	17
11.3 DIAGRAMA DO COMANDANTE.....	18
<b>12 DIAGRAMA DE ATIVIDADES.....</b>	<b>18</b>
12.1 DIAGRAMA DE LOGIN.....	19
12.2 DIAGRAMA DE PRAÇAS E OFICIAIS.....	20
12.3 DIAGRAMA DO COMANDANTE.....	21
<b>13 DIAGRAMA DE ENTIDADE E RELACIONAMENTO( DER).....</b>	<b>21</b>
<b>14 TELAS ( PROTOTIPAÇÃO).....</b>	<b>22</b>
14.1 TELA LOGIN.....	22
14.2 TELA PRINCIPAL.....	23
14.3 TELA CADASTRO DE PRAÇAS E OFICIAIS.....	24
14.4 TELA PESQUISA DE PRAÇAS E OFICIAIS.....	25

14.5 TELA CADASTRO DE COMANDANTE.....	26
14.6 TELA DE PESQUISA DE COMANDANTE.....	27
14.7 EXEMPLO DE PARTE.....	28
14.8 EXEMPLO DE REQUERIMENTO.....	29
<b>15 CÓDIGO FONTE.....</b>	<b>30</b>
15.1 CÓDIGO LOGIN (FRMLOGIN).....	30
15.2 CÓDIGO PRINCIPAL (FRMPRINCIPAL).....	31
15.3 CÓDIGO CADASTRO DE PRAÇAS E OFICIAIS (FRMFUNCAIONARIO).....	32
15.4 Cód. PESQ. DE PRAÇAS E OFICIAIS (FRMCONSULTARFUNCIONARIO).....	103
15.5 CÓDIGO CADASTRO DE COMANDANTE (FRMCOMANDANTE).....	107
15.6 Cód. PESQUISA DE COMANDANTE (FRMCONSULTARCOMANDANTE).....	114
<b>16 CRONOGRAMA PROPOSTO PARA O ESTÁGIO CURRICULAR OBRIGATÓRIO II – TCC DO 6º SEMESTRE.....</b>	<b>116</b>
<b>17 CONCLUSÃO.....</b>	<b>117</b>
<b>REFERÊNCIAS.....</b>	<b>118</b>

## 1 INTRODUÇÃO

Na execução do estágio curricular obrigatório na seção de pessoal da Polícia Militar, verifiquei que a seção não possuía nenhum programa para cadastro de funcionários e tinham uma grande dificuldade em organizar o cadastro dos funcionários e também nas questões de requerimentos e partes, pois grande era a procura por esses serviços. Por esses motivos, gerava uma grande perda de tempo e também desgastes dos funcionários.

A seção de pessoal (PM/1) do 34º batalhão da Polícia Militar do Estado do Rio de Janeiro, está localizada em frente à Praça Getúlio Vargas nº 71 no centro de Magé, conta com 4(quatro) policiais militares mais o chefe da seção de pessoal. A PM/1 é responsável por escala de serviço, requerimento, parte, dispensa e outros.

## **2 JUSTIFICATIVA**

Após observar a seção onde estagiei, verifiquei a necessidade de criar um programa para cadastro de funcionário bem como incluir os documentos de partes e requerimentos no projeto, para assim agilizar o serviço na seção, pois em um batalhão o número de policiais é grande. Nesta seção de pessoal onde estagiei trabalham 04(quatro) funcionários para dar conta de mais de 500 policiais militares do batalhão. Este cadastro de funcionário também será útil para organizar a ficha de todos os policiais militares do batalhão, pois observei que os dados cadastrais do policial eram guardados em fichas, gerando perda de tempo em consultas e risco em perdê-las.

## **3 OBJETIVOS**

### **3.1 OBJETIVO GERAL**

Projetar um programa que venha facilitar e agilizar a rotina de trabalho dos funcionários da seção de pessoal com o cadastro do policiais militares e a inclusão das partes e requerimentos no projeto.

### **3.2 OBJETIVO ESPECÍFICO**

- Otimizar a rotina de trabalho dos funcionários;
- Projetar o programa a incluir os documentos de partes e requerimentos;
- Empregar o sistema proposto que no caso será Microsoft Visual C#;
- Empregar o banco de dados proposto que no caso será MICROSOFT OFFICE ACCESS 2007;

## **4 REFERENCIAL TEÓRICO**

A tecnologia de informação é peça fundamental do desenvolvimento tanto das organizações públicas quanto privadas. Para que tenha uma utilização efetiva é preciso que seja vista dentro de um ambiente de integração em todas as áreas onde será utilizada, pois a sua implantação pode desencadear, desde que efetuada de forma criteriosa, uma mudança significativa dos processos de trabalho das organizações. É preciso uma integração entre os vários níveis organizacionais, sendo o planejamento a ferramenta mais adequada para se atingir metas comuns. A época em que vivemos tem características muito específicas, pois estamos passando da Era Industrial para uma Sociedade da Informação. Essa mudança ocasiona uma série de condicionantes que precisam ser analisados e entendidos para que possam ser aproveitados da melhor forma possível.

#### 4.1 SISTEMAS DE INFORMAÇÃO

Segundo Bio (1988) podemos considerar sistema como “um conjunto de elementos interdependentes, ou um todo organizado, ou partes que interagem formando um todo unitário e complexo”. Um sistema pode compor-se sucessivamente de subsistemas, que se relacionam entre si, compondo o sistema maior. Desta forma podemos entender que o sistema de informação é um subsistema. Um conceito de sistemas de informação podemos encontrar em Laudon e Laudon (2000), que diz que um sistema de informação "pode ser definido como um conjunto de componentes inter-relacionados trabalhando juntos para coletar, recuperar, processar, armazenar e distribuir informações com a finalidade de facilitar o planejamento, o controle, a coordenação, a análise e o processo decisório em empresas".

Os sistemas de informação têm como maior objetivo auxiliar os processos de tomada de decisões, quando não se propuserem a atender esse objetivo, sua existência não será significativa para a organização, Bio (1988).

Todo sistema, usando ou não recursos de tecnologia da informação, que armazena, processa e disponibiliza informação pode ser genericamente considerado sistema de informação, Rezende (1999).

Outro conceito de sistema de informação pode ser encontrado em Alter (1996, p. 2): "sistemas de informação são sistemas que usam a tecnologia da informação para capturar, transmitir, manipular, salvar ou exibir informação utilizada em um ou mais processos de negócios". Desta forma, fica evidente a vinculação

entre sistema de informação e tecnologia da informação. Para Davenport (1996, p. 83), “é difícil estabelecer uma separação total entre a informação, a tecnologia e os sistemas de informação”. Esta vinculação entre sistema de informação e TI parece-nos ainda mais clara em Freitas e Lesca (1992, p. 95): “o sistema de informações da empresa é o conjunto interdependente das pessoas, das estruturas da organização, das tecnologias de informação – hardware e software - dos procedimentos e métodos que deveriam permitir à empresa dispor – no tempo desejado – das informações de que necessita – ou necessitará – para seu funcionamento atual e para sua evolução”.

#### 4.2 TECNOLOGIA DA INFORMAÇÃO

Um conceito de TI podemos encontrar em Alter (1996, p. 2): "TI é o conjunto de hardwares e softwares que possibilitam o funcionamento dos sistemas de informações".

Segundo Davenport (1996), desde que ingressaram no ambiente empresarial, os computadores ligaram-se estreitamente à maneira pela qual o trabalho é realizado. Poderíamos até mesmo dizer que a TI começou a modificar radicalmente o trabalho - sua localização, rapidez, qualidade e outras características-chaves. Pelo exposto, percebe-se claramente que o termo "tecnologia de informação" está intimamente ligado à utilização dos computadores.

A grande maioria das organizações não realiza um planejamento de suas atividades na área de TI. As organizações ainda utilizam os recursos desta tecnologia orientados para “dentro” da organização, isto é, para resolver problemas internos de processamento de informações. Este tipo de uso também é necessário, mas é importante que a visão das possibilidades de utilização destas tecnologias seja ampliada e contemple o novo universo que cada vez mais mudará as relações de competitividade em todos os segmentos da economia. Ao lado do uso estratégico, tais tecnologias representam também um papel fundamental como agentes de integração e coesão organizacionais. O papel tradicional do processamento de dados, orientado para as operações básicas da organização, registrando e recuperando transações, processando documentos e dando apoio ao trabalho funcional, continua sendo uma das dimensões importantes no uso adequado das tecnologias de informações.



O uso da TI chega a ser entendido, algumas vezes, como sinônimo de moderno, racional e eficiente, porém nem sempre isso corresponde à realidade. Mesmo assim, sua aplicação nos ambientes de trabalho tem sido apontada como variável estratégica importante, principalmente em ambientes que estão em constante transformação. Seu uso também é muitas vezes apontado como elemento que serve à descentralização da organização, pois tem grande potencial técnico, o que lhe permite distribuir a informação, através do uso das telecomunicações, a lugares variados, ou mesmo em forma de rede.

O eficiente uso desta tecnologia em uma organização compreende vários aspectos; entre os quais, podemos citar a cultura organizacional, qualificação dos recursos humanos, estratégias adequadas ao ambiente da organização, entre outros.

As organizações públicas também se envolvem ou sofrem influências da competitividade. Um exemplo real desse paradigma são os governos que competem entre si para oferecer aos seus cidadãos as melhores condições para satisfação de seus interesses.

#### 4.3 TECNOLOGIA DA INFORMAÇÃO NO SETOR PÚBLICO

Durante o processo de adoção e implantação da informática no Brasil, o setor público teve um papel preponderante, pois foi um dos pioneiros deste processo. A complexidade de seus processos administrativos e o grande volume de informações que estes geram, fizeram destas organizações usuários ideais para a tecnologia que recém surgia. Com o transcorrer do tempo, a informatização avançou e tornou-se uma das mais importantes ferramentas de gestão. As organizações do setor público também seguiram nessa direção, em busca da chamada “informatização”. Atualmente a busca da eficiência, tão almejada por todos, passa pela utilização adequada dos recursos computacionais, cada vez mais complexos.

Com o surgimento de novas tecnologias, maiores exigências legais e maior pressão da sociedade na gestão e transparência dos recursos públicos, esta tecnologia tornou-se uma ferramenta importante para que estes objetivos fossem alcançados. Assim os recursos investidos em TI pelo setor público brasileiro têm crescido significativamente, Cunha (2004). O surgimento da internet modificou radicalmente as formas de comunicação entre os organismos públicos e a

sociedade, quer sejam em serviços prestados a esta como também na relação com fornecedores, Cunha (2004) e Bovo (2006). As exigências legais e de fiscalização de órgãos como os tribunais de contas, fez com que os gestores públicos adotem uma nova postura perante a utilização da TI, especialmente nos softwares de gestão, ou seja, sistemas específicos para controle das atividades fins, sendo uma tendência à terceirização destes sistemas, Rezende (2004), Lobler, Marques e Visentini (2006), Guagliardi e Rezende (2006).

#### 4.4 CULTURA, POLÍTICA E ESTRUTURA DAS ORGANIZAÇÕES PÚBLICAS

As características de cada organização são moldadas conforme suas estratégias, Tavares (1991), nas organizações públicas isto não é diferente. Desta forma podemos entender que os componentes da mesma, devem estar de acordo com estas características.

Desta forma, assim como as organizações privadas são departamentalizadas, geralmente por funções, as públicas também moldam sua hierarquia conforme as funções. Em órgãos públicos, geralmente uma secretaria assume as funções correspondentes a um departamento. Nesta forma de estrutura organizacional o trabalho é distribuído de forma especializada, viabilizando um melhor planejamento, a realização e a avaliação de ações e serviços específicos, em cada secretaria. Essa dinâmica supõe um planejamento geral, acima das secretarias, que articule os objetivos gerais da gestão com as demandas particulares de cada uma delas e as respectivas redes de serviço, Fundasp (2008). Entretanto, infelizmente, em diversas situações condições políticas, culturais e estruturais acabam interferindo nesta relação, Fleury e Fischer (1996), fazendo até mesmo que os interesses internos de cada secretaria se mostrem mais atuantes que o interesse do geral. Nas organizações públicas, muitas vezes isto não acontece pois seus componentes tem ou vem em elementos políticos e estruturais, como por exemplo a estabilidade, recursos que permitem que estes “enfrentem” estas características organizacionais.

Mesmo com todas estas peculiaridades, as organizações do setor público não estão imunes às novas exigências da sociedade atual, para atenderem a estas novas demandas inegavelmente mudanças organizacionais deverão acontecer, Kliksberg (1992). Uma outra particularidade atinge estas organizações,

ou seja, a questão política, pois seus gestores são eleitos e este processo gera continuidade e/ou descontinuidade de comando, nem sempre pautada em pilares de competência.

A TI com suas diversas possibilidades de uso, através de ferramentas tais como: softwares integrados de gestão, softwares de comunicação on-line, softwares de colaboração, maior democratização do acesso às informações, está proporcionando uma nova estrutura para várias atividades produtivas e comunicacionais, Zuboff (1994). Esta tecnologia também tem um importante papel a desempenhar dentro das organizações públicas, pois através de seus recursos pode facilitar, ou ajudar a promover, diversas mudanças organizacionais, em especial no sentido de democratização da informação e agilização de processos.

## **5 METODOLOGIA**

A atuação participativa no treinamento de funcionários da seção de pessoal será uma marca importante da metodologia a ser utilizada.

Antes do desenvolvimento prático dos serviços que o software contempla, serão desenvolvidas ações primordiais para sua utilização como, por exemplo: estudo das necessidades, elaboração, implantação, testes, alterações, implementações, treinamento de usuários, discussão sobre relatórios primordiais, etc.

Para o desenvolvimento deste projeto serão utilizados como referenciais bibliográficos artigos da internet, livros de fundamentos básicos sobre o assunto, módulos do curso ADS – Unopar etc.

## **6 CARACTERIZAÇÃO DA ORGANIZAÇÃO EM ESTUDO**

Nome da empresa: Seção de Pessoal do 34º BPM do Estado do Rio de Janeiro  
Endereço completo: Praça Getúlio Vargas nº 71, centro Magé, RJ.  
Telefone / Fax: (21) 2633-5488  
Web site; e-mail: [www.policiamilitar.rj.gov.br](http://www.policiamilitar.rj.gov.br); 34bpm@policiamilitar.rj.gov.br  
Nome e cargo da chefia direta: Marco Antônio Oliveira de Andrade

Ramo de atividade e área de atuação: SEGURANÇA PÚBLICA

Números de empregados da seção: 05

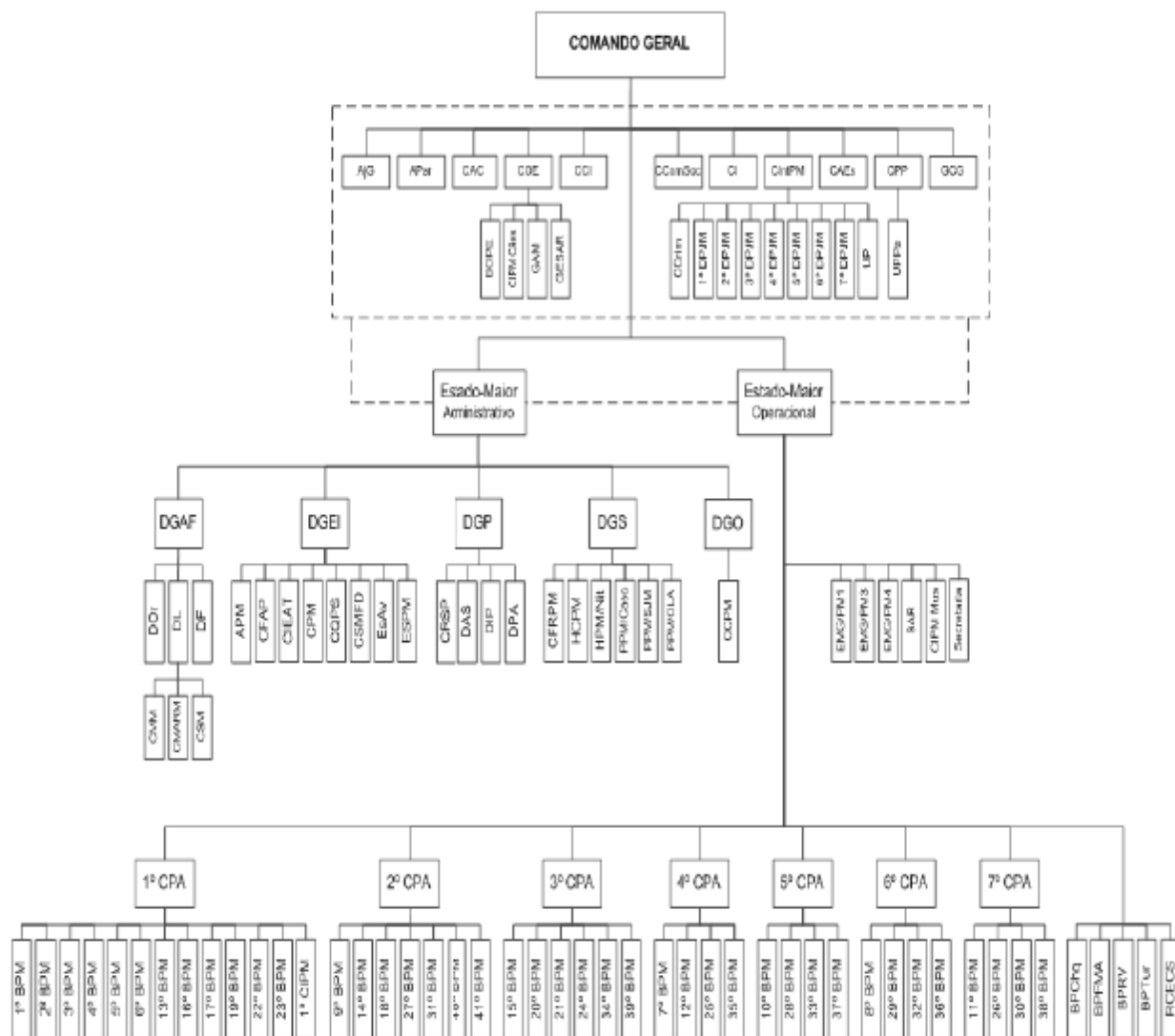
Categorias dos profissionais que atuam na empresa:

Soldado da Polícia Militar = 02

Cabo da Polícia Militar = 02

Chefe da seção = 01

Organograma da empresa:



Produtos e serviços desenvolvidos pela empresa: SEGURANÇA PÚBLICA

### Histórico da Empresa:

Em 21 de Junho de 1977, em solenidade marcante, deu-se a inauguração da então 5ª Companhia destacada, administrativamente pertencente ao 15º Batalhão de Polícia Militar, sediado em Duque de Caxias – RJ, numa tentativa de melhorar as necessidades de um trabalho mais eficaz no combate ao crime no Município de Magé, contando à época, com somente uma viatura operacional e

aproximadamente quarenta Policiais Militares. O dia 05 de fevereiro de 1987, marca a transformação daquela subunidade na então 7ª Companhia Independente de Polícia Militar, renumerada, desde 28 de Fevereiro de 1991 para 3ª Companhia Independente de Polícia Militar e transformada em 34º Batalhão de Polícia Militar através da Resolução nº. 556 de 16 de Agosto de 2002, publicada no Bol. da PM nº. 093 de 23 de agosto de 2002, dotado de instalações mais amplas, modernas e funcionais, atualmente responsável pela segurança dos Municípios de Magé e Guapimirim, com uma área total de atuação de 749 Km<sup>2</sup> e população de aproximadamente 294.700 habitantes.

## **7 TECNOLOGIAS**

Todo sistema de informação deve ter um bom planejamento antes que se comece a produzir o software e este trabalho vai se utilizar de algumas técnicas de construção de software que ajudará a um melhor desempenho na fabricação deste.

No desenvolvimento de modelos na fase de análise do sistema foi utilizada a Linguagem de Modelagem Unificada (UML), que é uma linguagem que serve para especificar, construir, visualizar e documentar os artefatos de um sistema de software. A ênfase da UML é na definição de uma linguagem de modelagem padrão e, independente de linguagens de programação, de ferramentas CASE, bem como dos processos de desenvolvimento. O Objetivo da UML é, dependendo do tipo de projeto, da ferramenta de suporte ou da organização envolvida, poder adotar diferentes processos/metodologias, mantendo-se, contudo, a utilização de uma única linguagem de modelagem. Foi utilizado o Diagrama de caso de uso que vai descrever a funcionalidade proposta para o sistema, o Diagrama de classe que vai descrever a representação da estrutura e relações das classes que servem de modelo para objetos, o Diagrama de sequência que vai representar a sequência de processos (mais especificamente, de mensagens passadas entre objetos) num programa de computador, o Diagrama de Atividades que vai representar os fluxos conduzidos por processamentos e o Diagrama de Entidade e Relacionamento(ER) que vai descrever o modelo de dados de um sistema com alto nível de abstração.

A linguagem de programação utilizada para a codificação do sistema foi o C# 2010. O ambiente de desenvolvimento utilizado foi a Microsoft Visual Studio 2010 que tem recursos novos e avançados. Os Novos recursos trabalham juntos para melhorar a programação do Microsoft Office, aumentam a

capacidade de explorar o código, oferecem suporte para o desenvolvimento controlado por teste e muito mais.

O banco de dados utilizado foi o MICROSOFT OFFICE ACCESS 2007 também conhecido por MSAccess, é um Sistema de gerenciamento de banco de dados da Microsoft, incluído no pacote do Microsoft Office Professional, que combina o Microsoft Jet Database Engine com uma interface gráfica do utilizador (graphical user interface). Ele permite o desenvolvimento rápido de aplicações que envolvem tanto a modelagem e estrutura de dados como também a interface a ser utilizada pelos usuários.

Microsoft Access é capaz de usar dados guardados em Access/Jet, Microsoft SQL Server, Oracle, ou qualquer recipiente de dados compatível com ODBC.

O desenvolvimento da estrutura de dados se dá de forma muito intuitiva, bastando que o desenvolvedor possua conhecimentos básicos em modelagem de dados e lógica de programação.

Programadores relativamente inexperientes e usuários determinados podem usá-lo para construir aplicações simples, sem a necessidade de utilizar ferramentas desconhecidas.

## 7.1 REQUISITOS DE HARDWARE E SOFTWARE NECESSÁRIOS

### 7.1.1 Hardware

- Pentium IV
- Processador 2.0MHz
- Memória: 256 MB
- Monitor
- Mouse
- Teclado

### 7.1.2 Software

- Sistema Operacional Windows XP home
- Linguagem de Programação: Visual C# 2010
- Banco de dados: Microsoft OFFICE ACCESS 2007
- Microsoft Office Word 2003

## **8 LEVANTAMENTO DE REQUISITOS**

O projeto inicialmente foi pensado a solucionar um problema referente a cadastro do policiais militares e a inclusão das partes e requerimentos no projeto, primeiramente foi realizado um estudo sobre o negocio e as necessidades da empresa, foi realizado o levantamento de requisitos baseado no conceito de OO usando a UML e a partir deste ponto o qual obtive informações para fazer a modelagem de projeto aplicando os conceitos:diagrama de caso de uso, diagrama de classe de projeto, diagrama de seqüência, diagrama de atividades e diagrama de entidade e relacionamento(DER) visando geração de documentos de todas as fases do projeto garantido confiança e qualidade.

## **9 DIAGRAMA DE CASO DE USO**

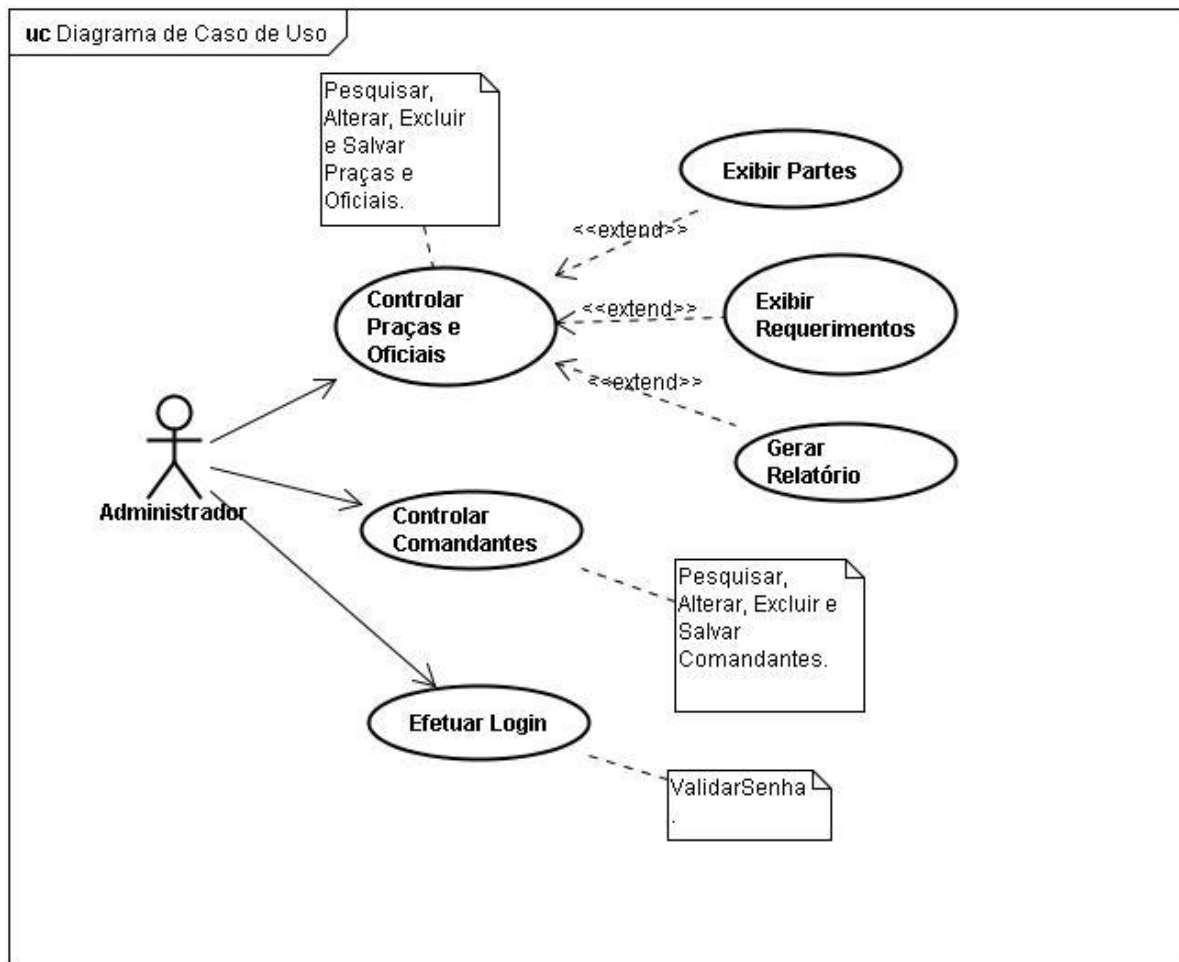
O modelo de Casos de Uso foi proposto por I. Jacobson como um instrumento para descrição das intenções ou requisitos para um sistema computacional. A construção do Modelo de Casos de Uso corresponde a uma das fases iniciais de um projeto de software pois envolve a determinação dos usos que o sistema terá, ou seja, do que ele deverá fornecer como serviços. O modelo de Casos de Uso é diferente da visão funcional utilizada no passado nas abordagens de projeto estruturado. Ao invés de focalizar as funções (atribuições técnicas) do sistema, o modelo de Casos de Uso captura os usos ou aplicações completas do sistema. Este modelo busca responder a questão: Que usos o sistema terá? Ou para que aplicações o sistema será empregado?

Os modelos de Casos de Uso são descritos através de Diagramas de Casos de Uso na UML. De uma forma geral, cada projeto de software conterá um Diagrama de Casos de Uso. Para sistemas mais extensos, é possível decompor o diagrama em um conjunto de subdiagramas.

Uma vez construído o modelo de Casos de Uso, o restante do projeto pode ser guiado baseando-se neste modelo. Dito de outra forma, a partir do modelo de Casos de Uso, o restante do projeto irá se preocupar com a forma de realização dos casos de uso (que classes e objetos são necessários, como e quando cada um atuará etc.).

O modelo de Casos de Uso é um instrumento eficiente para determinação e documentação dos serviços a serem desempenhados pelo sistema.

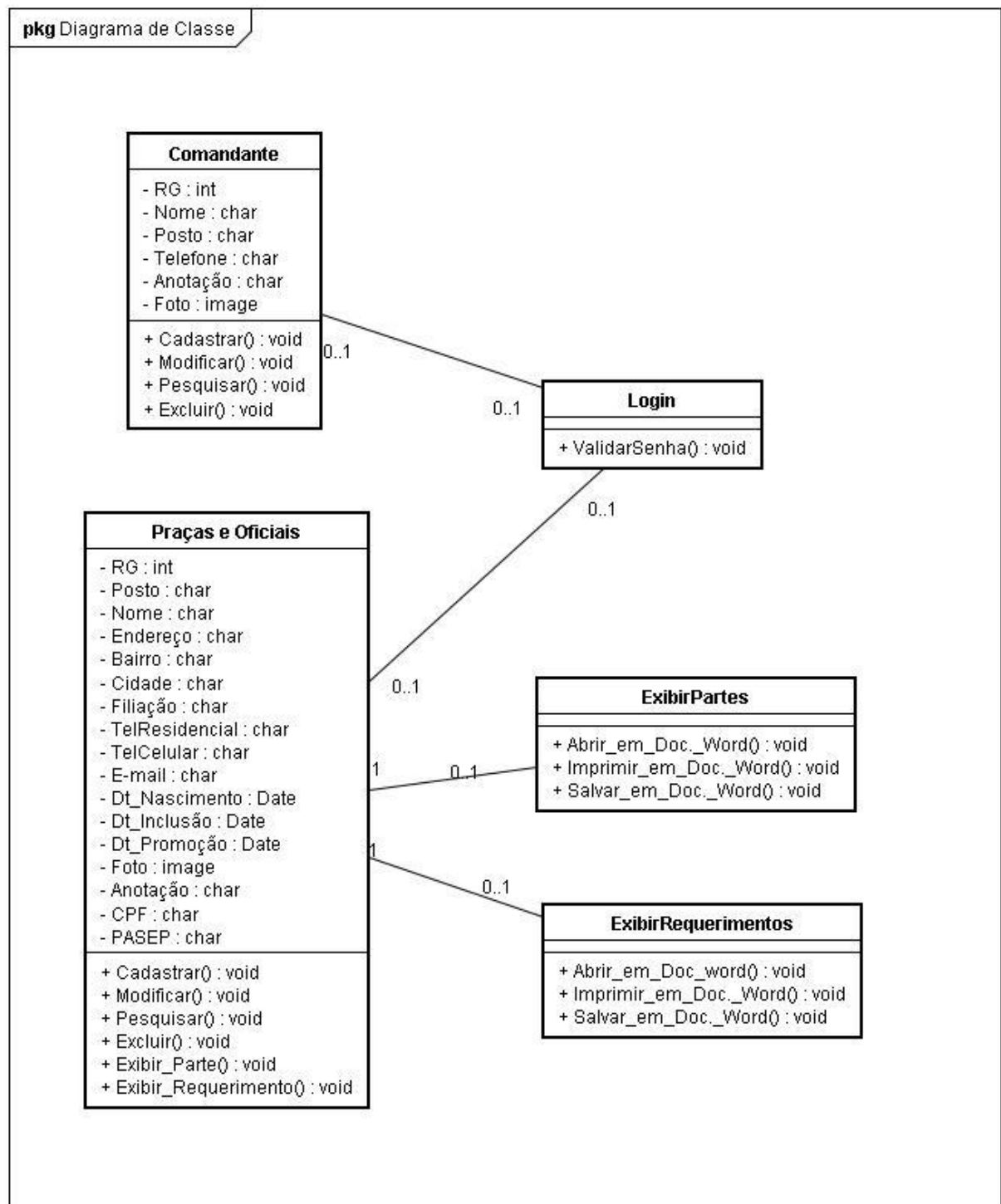
Ele é também um bom meio para comunicação com os clientes no processo de definição dos requisitos do sistema.



## 10 DIAGRAMA DE CLASSE

O diagrama de classes permite a visualização das classes que compõem o sistema com seus respectivos atributos e métodos. Trata-se de uma visão estática de como as classes estão organizadas, preocupando-se em definir a estrutura lógica das mesmas.



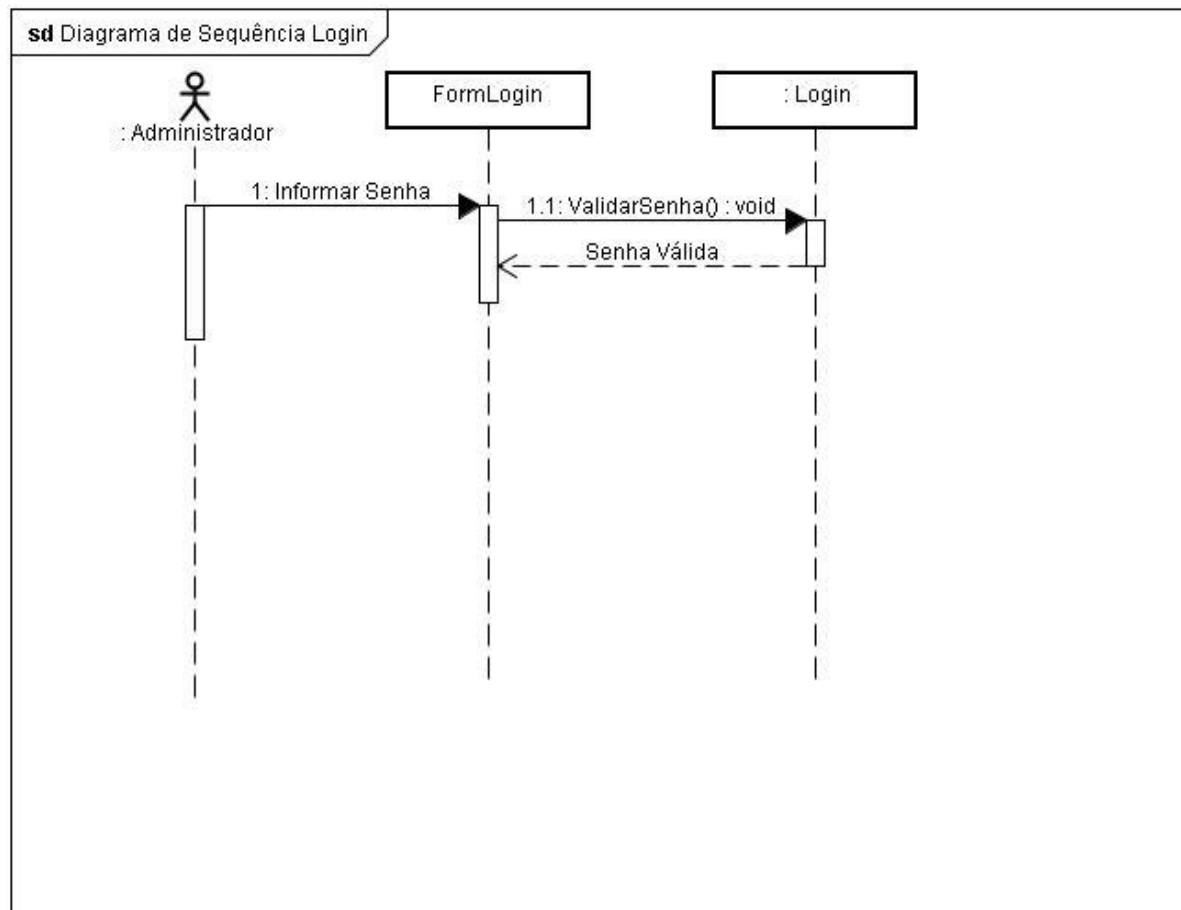


## 11 DIAGRAMA DE SEQUÊNCIA

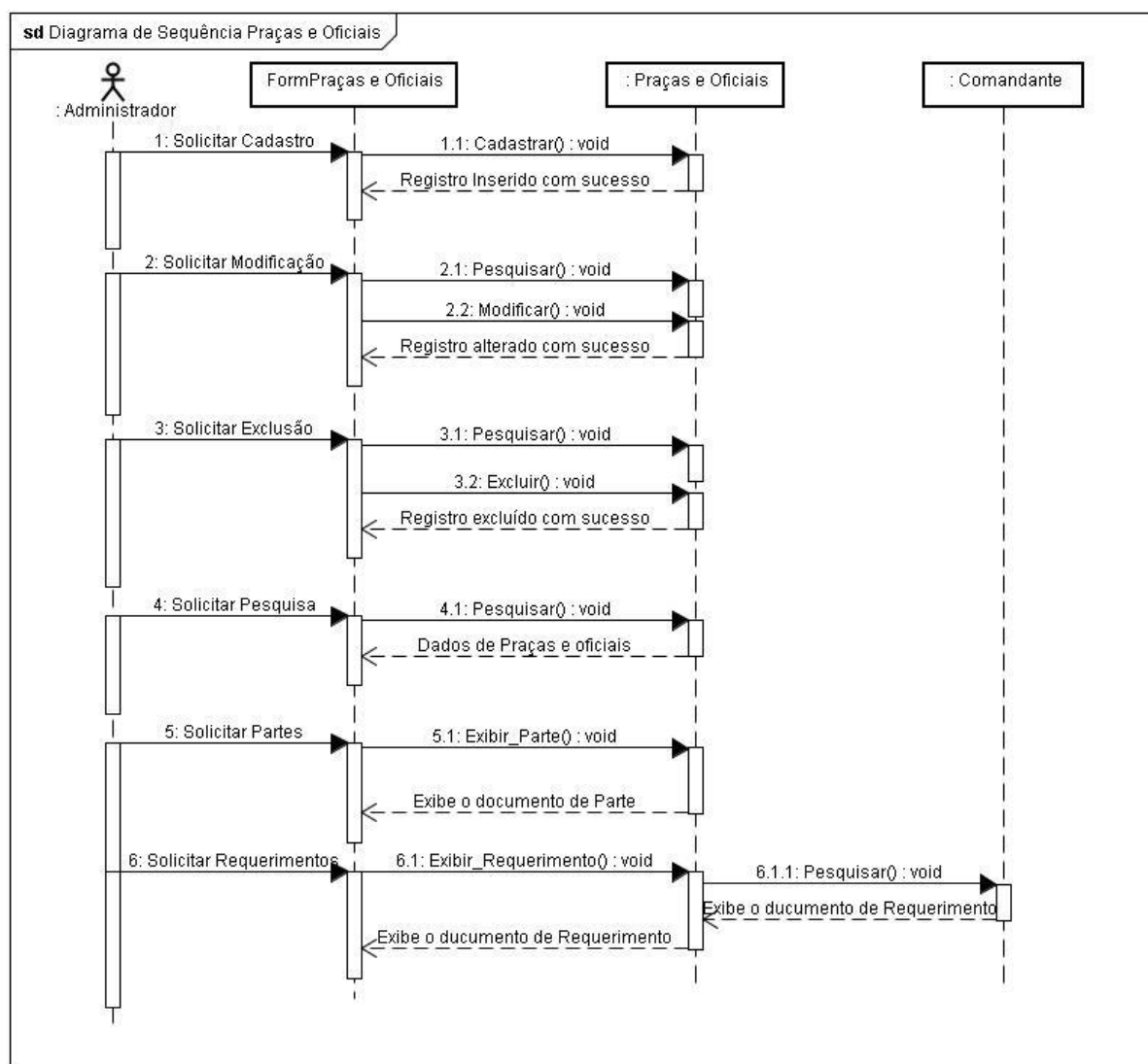
O Diagrama de Sequência é uma das ferramentas UML usadas para representar interações entre objetos de um cenário, realizadas através de operações ou métodos (*procedimentos ou funções*). Este diagrama é construído a partir do

Diagrama de Casos de Usos. Primeiro, se define qual o papel do sistema (Use Cases), depois, é definido como o software realizará seu papel (Sequência de operações).

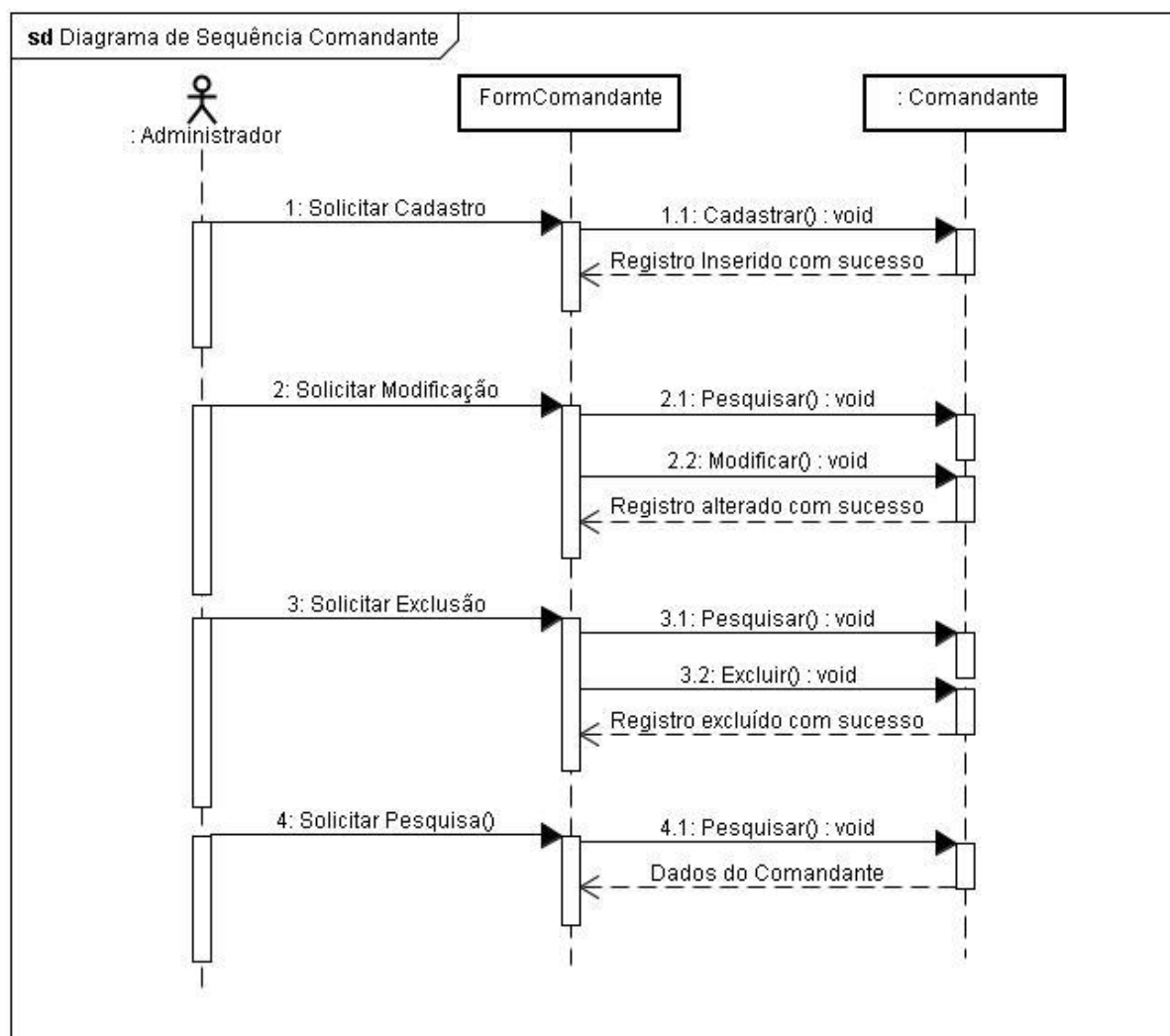
### 11.1 Diagrama de Login



### 11.2 Diagrama de Praças e Oficiais



### 11.3 Diagrama do Comandante

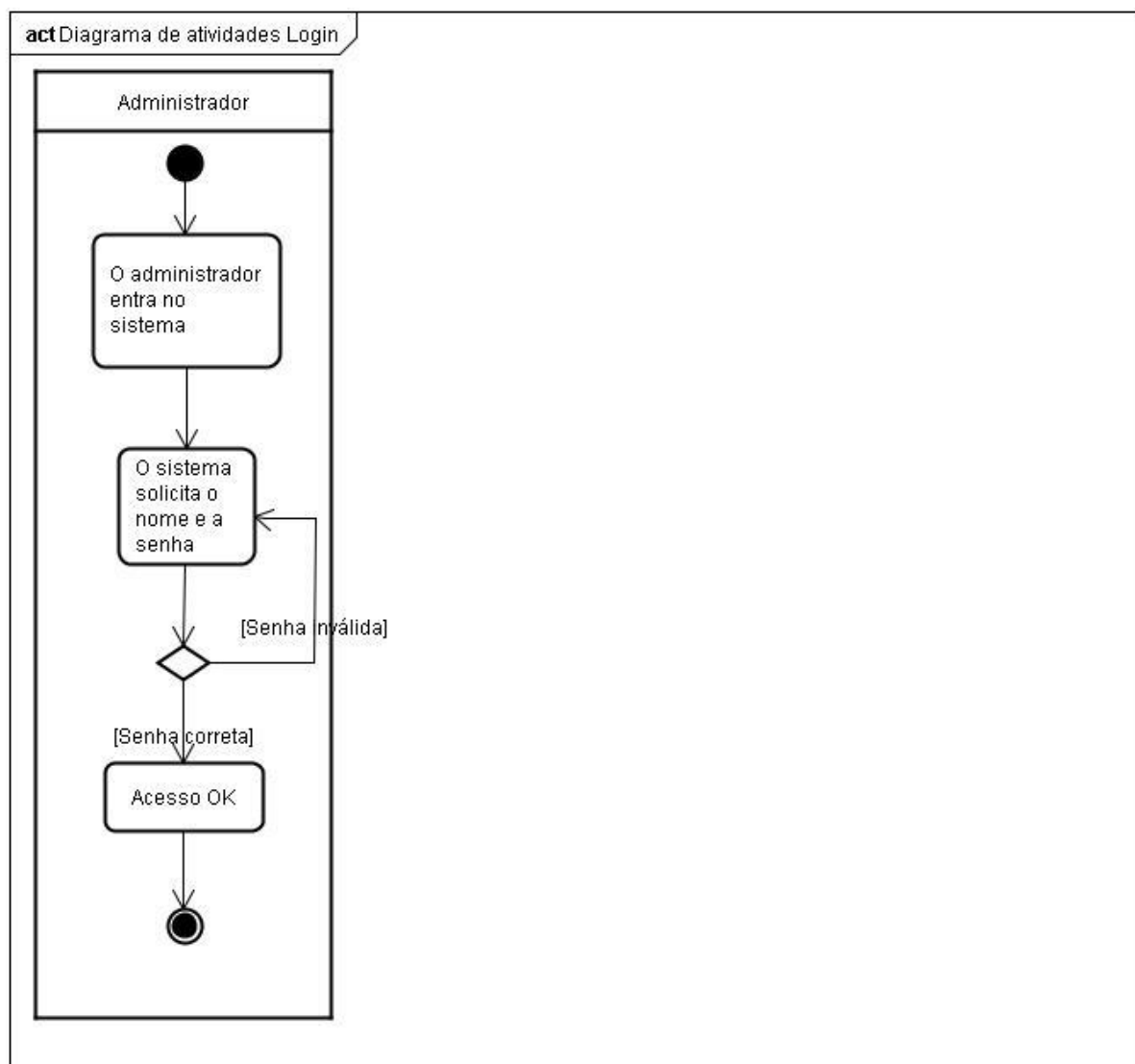


## 12 DIAGRAMA DE ATIVIDADES

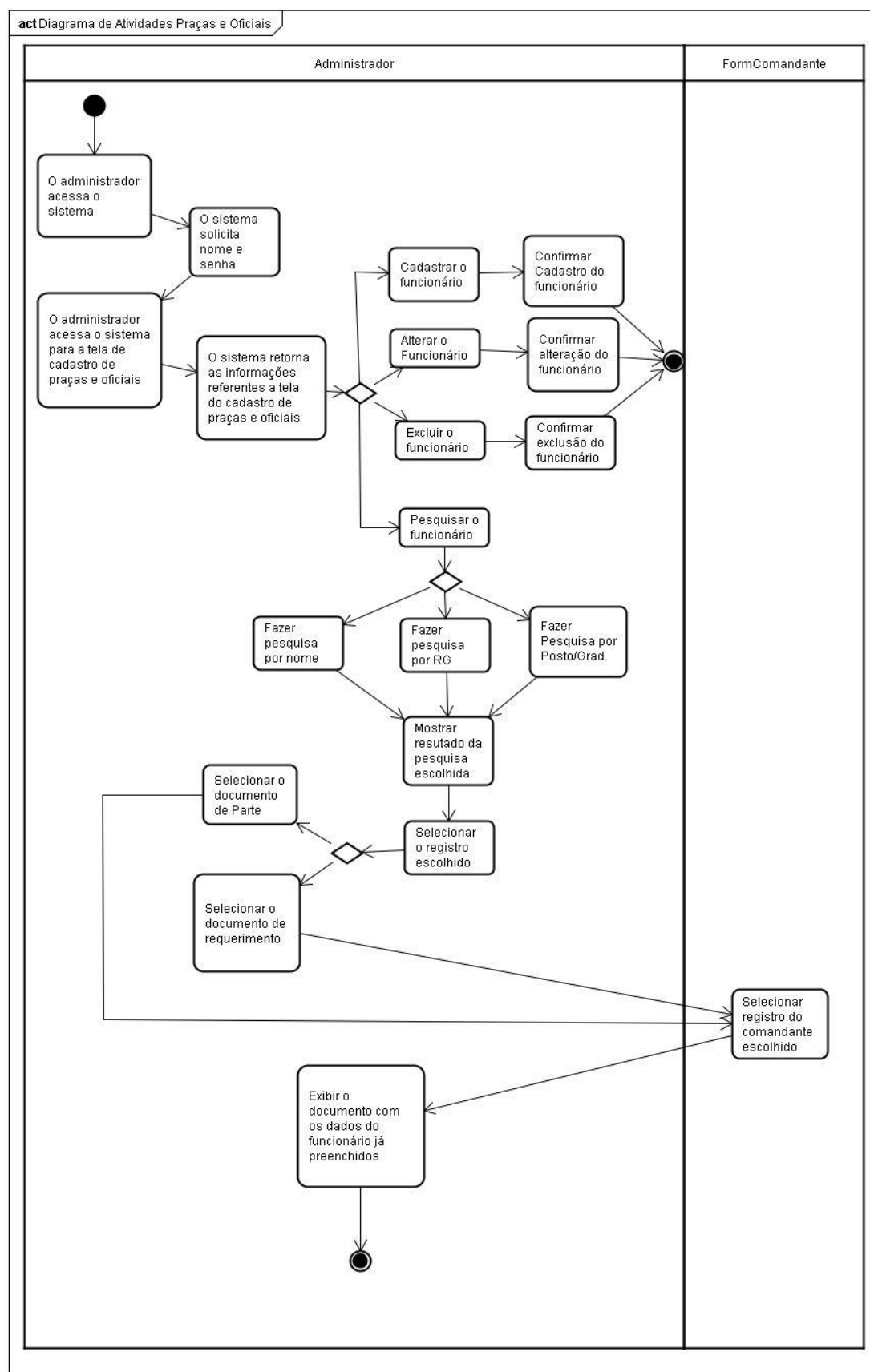
É essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra. Comumente isso envolve a modelagem das etapas seqüenciais em um processo computacional.

Os diagramas de atividade não são importantes somente para a modelagem de aspectos dinâmicos de um sistema ou um fluxograma, mas também para a construção de sistemas executáveis por meio de engenharia de produção reversa.

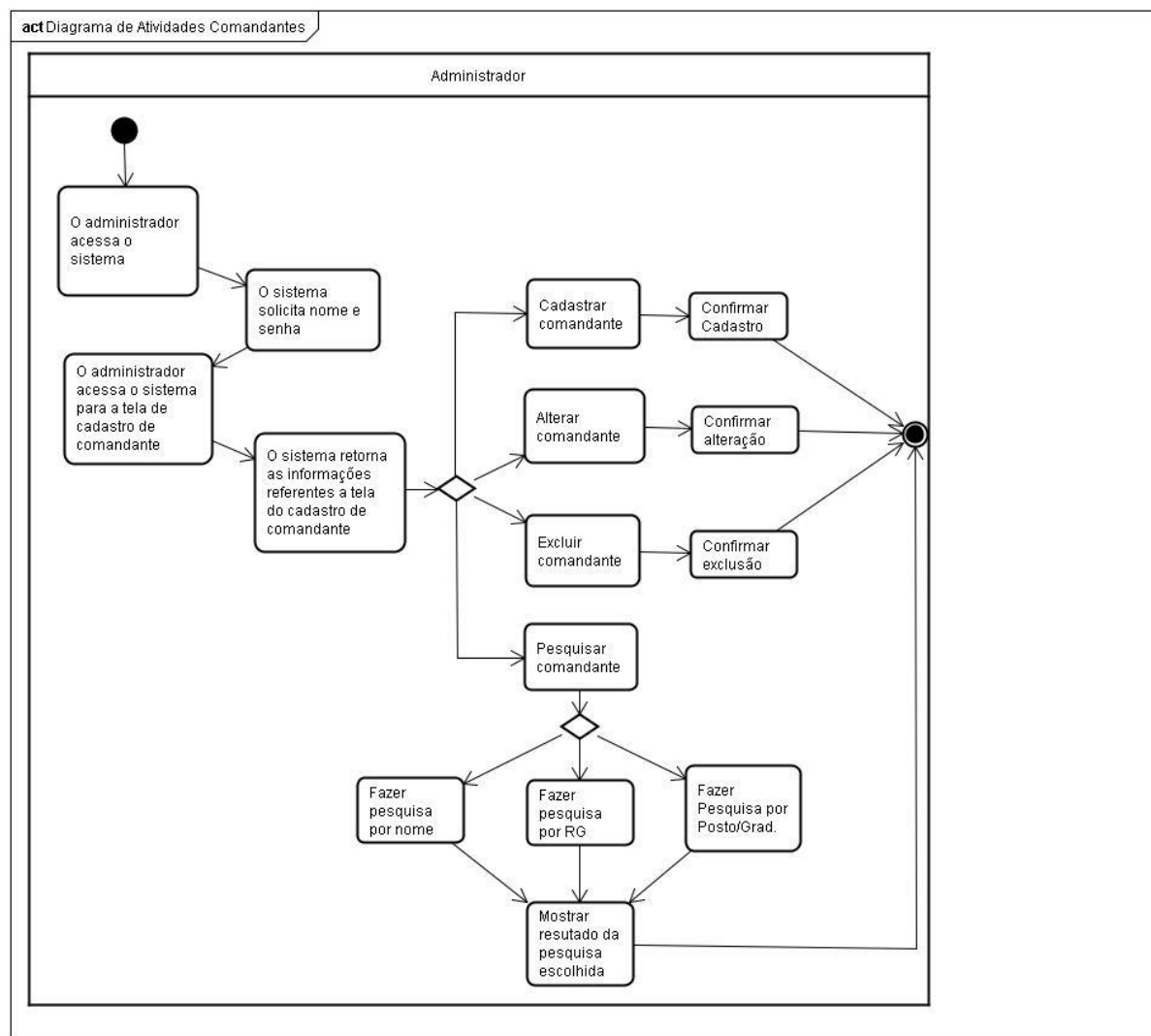
### 12.1 Diagrama de Login



## 12.2 Diagrama de Praças e Oficiais



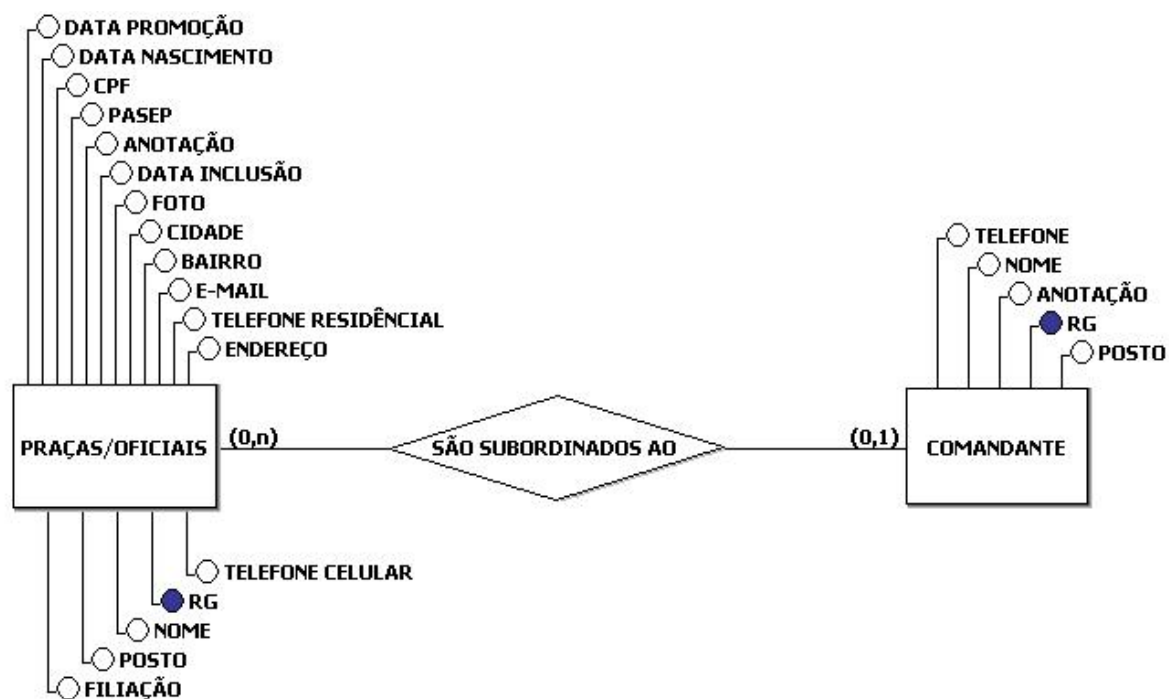
### 12.3 Diagrama do Comandante



### 13 DIAGRAMA DE ENTIDADE E RELACIONAMENTO (DER)

É usado para representar o modelo conceitual do negócio. Não confundir com modelo relacional, que representam as tabelas, atributos e relações materializadas no banco de dados.

- MER: Conjunto de conceitos e elementos de modelagem que o projetista de banco de dados precisa conhecer. O Modelo é de Alto Nível.
- DER: Resultado do processo de modelagem executado pelo projetista de dados que conhece o MER.



## 14 TELAS (PROTOTIPAÇÃO)

### 14.1 Tela Login

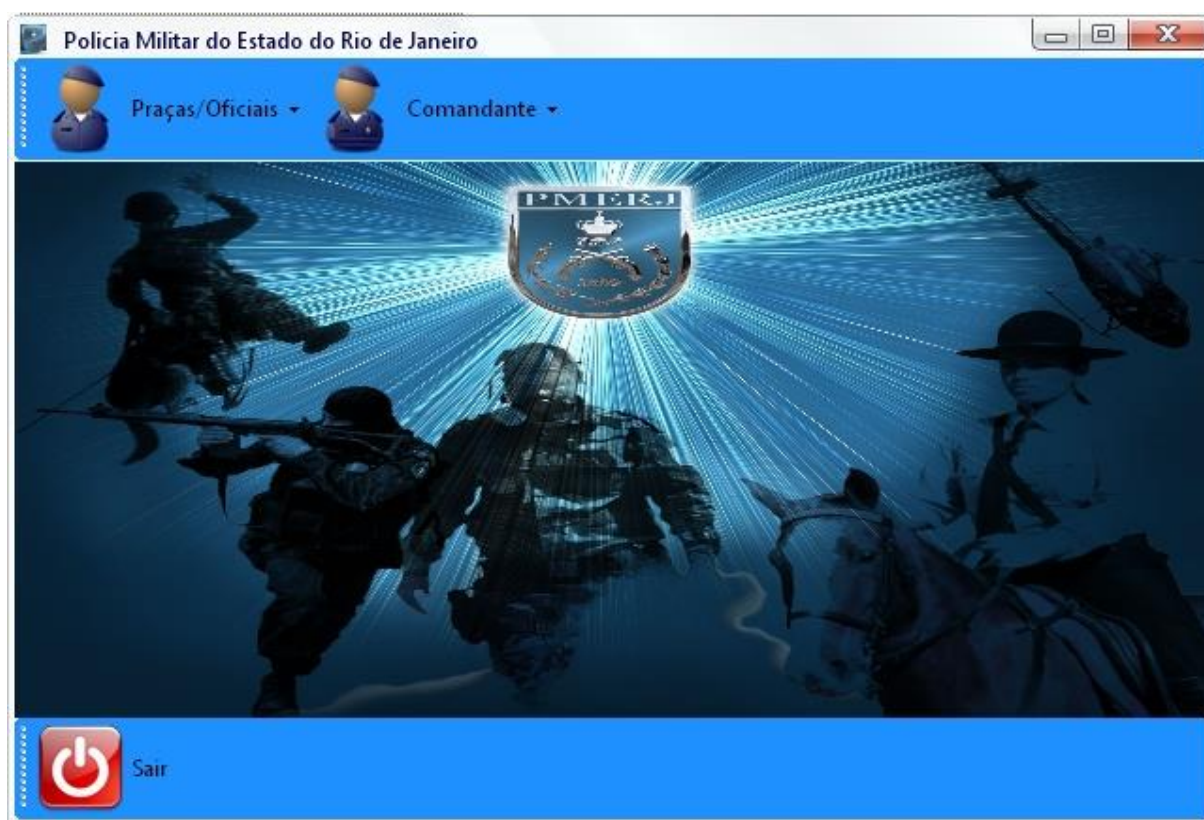
The screenshot shows a window titled "Login" with a standard Windows-style title bar (minimize, maximize, close buttons). On the left side, there is a graphic of a padlock and a folder. To the right of the graphic, there are two input fields:

- Usuário: [text input field]
- Senha: [text input field]

At the bottom of the window, there is a blue bar containing two buttons: "Entrar" (with a small icon of three dots to its left) and "Sair".




## 14.2 Tela Principal



## 14.3 Tela Cadastro de Praças e Oficiais

## 14.4 Tela Pesquisa de Praça e Oficiais

**Pesquisa de Praças e Oficiais**

 Voltar

**Filtro**

☒ Nome ☐ RG ☐ Grad./Posto

Digite o texto a ser pesquisado e clique em pesquisar:





\*Para selecionar um registro é só dar 2 cliques.

RG	Nome	Filiacao	Posto	End

Encontrado(s): "0" registros com a palavra "p"

#### 14.5 Tela Cadastro de Comandante

**Cadastro de Comandantes**


 Voltar  Alterar  Cadastrar  Excluir

0 of {0}





RG:  Posto:

Nome:

Telefone:


 Foto

Anotações:

 Sair  Confirmar Alteração  Confirmar Cadastro  Cancelar

#### 14.6 Tela Pesquisa de Comandante

**Pesquisa de Comandantes**

 Voltar

**Filtro**

☒ Nome ☐ RG ☐ Grad./Posto


Digite o texto a ser consultado e clique em pesquisar:

**Pesquisar**


RG	Nome	Posto	Telefone	And
----	------	-------	----------	-----

Encontrado(s): "0" registros com a palavra "a"

#### 14.7 Exemplo de Parte

POLÍCIA MILITAR DO ESTADO DO RIO DE JANEIRO	
	<div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">CPA CPB</div> <div style="border: 1px solid black; padding: 2px;">CPM 34º BPM</div> </div> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">LOCAL E DATA Quartel em Magé, em 26 de julho de 2011</div> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">DO <div style="background-color: #f0f0f0; height: 20px; width: 100%;"></div></div> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">AO Sr. Comandante da 2ª Cia</div> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">ASSUNTO Participação – FAZ</div>
<div style="border: 1px solid black; padding: 2px; width: 100px; margin: 0 auto;">PARTE S/Nº</div>	<div style="border: 1px solid black; padding: 10px; min-height: 300px; margin-top: 10px;"> <p style="text-align: center; margin-bottom: 20px;">TEXTO</p> <p style="text-align: justify;">Este signatário participa-vos que no dia 29 de Junho de 2009 (Domingo) estará aniversariando, motivo pelo qual solicita a este comando que seja concedida uma dispensa do serviço de 12 horas.</p> <div style="border: 1px solid black; background-color: #f0f0f0; width: 300px; height: 25px; margin: 20px auto;"></div> </div>

14.8 Exemplo de Requerimento



**POLÍCIA MILITAR DO ESTADO DO RIO DE JANEIRO**  
**FOLHA DE REQUERIMENTO (REQ)**

C.P.A.

O.P.M.

Protocolo

ANEXADO(S) O(S) DOCUMENTO(S) COMPROBATORIO(S) REQUER.

<input type="checkbox"/> CARTA DE FIANÇA <input type="checkbox"/> LICENÇA ESPECIAL <input type="checkbox"/> ANULAÇÃO DE CORRETIVO <input type="checkbox"/> CERTIDÃO DE CURRÍCULO <input type="checkbox"/> SUBSTITUIÇÃO DE CARTA DE FIANÇA <input type="checkbox"/> FAZER CONSTAR POSSUIR CURSO	<input checked="" type="checkbox"/> AJUDA DE CUSTO <input type="checkbox"/> AVERBAÇÃO DE TEMPO DE SERVIÇO <input type="checkbox"/> RESTITUIÇÃO DE GARANTIA DE FARDAMENTO <input type="checkbox"/> ADIANTAMENTO PARA CONFEÇÃO DE FARDAMENTO <input type="checkbox"/> INSPEÇÃO DE SAÚDE PARA DEPENDENTES <input type="checkbox"/> ADIANTAMENTO PARA CONFEÇÃO DE FARDAMENTO
---	---

JUSTIFICATIVA

<input checked="" type="checkbox"/> PEDIDO INICIAL	<input type="checkbox"/> RECONSIDERAÇÃO	<input type="checkbox"/> RECURSO
--	---	----------------------------------

AMPARO LEGAL A PRETENSÃO DO REQUERENTE

De acordo com o art.32, item III, e art.33, item II, da lei 279, de 26 de Novembro de 1979.

26/07/11  
DATA

INFORMAÇÕES

FILIAÇÃO

ESTADO CIVIL

<input checked="" type="checkbox"/> SOLTEIRO	<input type="checkbox"/> CASADO	<input type="checkbox"/> DESQUITADO	<input type="checkbox"/> VIÚVO	<input type="checkbox"/> CONCUBINO	<input type="checkbox"/> DIVORCIADO
--	---------------------------------	-------------------------------------	--------------------------------	------------------------------------	-------------------------------------

SITUAÇÃO JUCIARIA

NADA CONSTA ☒

RESPOSTA A:

☐ IPM

☐ IP

☐ I ADM

PROC. MILITAR

PROC. CIVIL

SUBJUDICE

SITUAÇÃO DISCIPLINAR

EXCEPCIONAL

ÓTIMO ☒

INSUFICIENTE

MAU

NOTA DE CULPA

SEM ALTERAÇÃO ☒

SEQUE ANEXA

PARECER DO SECRETÁRIO DA OPM

Convém ENCAMINHAR a DGP/DPA/SCAv

DATA

SECRETÁRIO



TESOURARIA		ORIGEM	
IMPLANTADO	S <input type="checkbox"/>	N <input type="checkbox"/>	C P A <input type="checkbox"/> O P M <input type="checkbox"/>
DESCONTOS NOS 03 ULTIMOS MESES		GARANTIA DE FARDAMENTO	
	R\$	DEPOSITO	SIM <input type="checkbox"/> NAO <input type="checkbox"/>
	R\$	SALDO CONSIGNAVEL	
	R\$		
<div style="display: flex; justify-content: space-between; margin-top: 10px;"> <span>____/____/____ DATA</span> <span>_____ ASS/POSTO/RG</span> </div>			

ALMOXARIFADO	FISCAL ADMINISTRATIVO
DEBITO A FAZENDA ESTADUAL	CONVEM
<input type="checkbox"/> NAO <input type="checkbox"/> SIM <input type="checkbox"/> ANEXO	<input type="checkbox"/> ATENDER <input type="checkbox"/> ENCAMINHAR <input type="checkbox"/> ARQUIVAR <input type="checkbox"/> COMPLEMENTAR <input type="checkbox"/> CONFERIR <input type="checkbox"/> CERTIFICAR
____/____/2006 DATA	____/____/2006 DATA
_____ ASS/POSTO/RG	_____ ASS/POSTO/RG

O PRESENTE REQUERIMENTO PERMANECIU	<input type="checkbox"/>	DIA(S) NOS ÓRGÃOS INFORMANTES
------------------------------------	--------------------------	-------------------------------

CAMPO RESERVADO PARA COMPLEMENTAÇÃO DE INFORMAÇÕES

O signatário solicita Ajuda de Custo em razão de ter sido Transferido do ESPM (Niterói), para o 34º BPM (Mage), conforme Bol PM nº 019, de 26.Jan.07.

Anexos: Cópia do Bol PM citado;  
Cópia do ofício de apresentação

Quartel em Mage, em 26 de julho de 2011.

DECISÃO
<input type="checkbox"/> ATENDER <input type="checkbox"/> INDEFERIR <input type="checkbox"/> PUBLICAR <input type="checkbox"/> ENCAMINHAR À <input type="checkbox"/> ARQUIVAR NA TESOURARIA <input type="checkbox"/> DEVOLVER
____/____/____ DATA

## 15 CÓDIGO FONTE

### 15.1 CÓDIGO LOGIN ( FRMLOGIN)

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;

```



```

using System.Windows.Forms;

namespace SistemadeCadastroPolicial
{
    public partial class frmLogin : Form
    {
        public frmLogin()
        {
            InitializeComponent();
        }

        private void tsbtnEntrarLogin_Click(object sender, EventArgs e)
        {
            this.DialogResult = (ValidarSenha() ? DialogResult.OK :
DialogResult.None);
        }

        private bool ValidarSenha()
        {
            //Faz a verificação de usuário e senha
            if (txtUsuario.Text.Equals("policia") && txtSenha.Text.Equals("1809"))
            {
                return true;
            }

            else
            {
                MessageBox.Show("Ops, ocorreram erros!\n\nUsuário ou Senha inválida!",
                    "Mensagem", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return false;
            }
        }

        private void tsbtnSairLogin_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
    }
}

```

## 15.2 CÓDIGO PRINCIPAL ( FRMPRINCIPAL)

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace SistemadeCadastroPolicial
{
    public partial class frmPrincipal : Form
    {
        public frmPrincipal()
        {
            InitializeComponent();
        }
    }
}

```

```

    }
    #region Botões

    private void tsbtnSair_Click(object sender, EventArgs e)
    {
        {
            if (MessageBox.Show("Você tem certeza que deseja sair do sistema?",
"Mensagem do Sistema",

                                MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.Yes)
            {
                Application.Exit();
            }
        }
    }
    private void tsmPesquisarPraca_Click(object sender, EventArgs e)
    {
        //this.Hide();
        frmConsultarFuncionario objFrmConsulta = new frmConsultarFuncionario();
        objFrmConsulta.ShowDialog();
    }

    private void tsmCadastroPraca_Click(object sender, EventArgs e)
    {
        frmFuncionario objFrmFuncionarios = new frmFuncionario();

        objFrmFuncionarios.ShowDialog();
    }

    private void tsmCadastroComandante_Click(object sender, EventArgs e)
    {
        frmComandante objFrmComandante = new frmComandante();
        objFrmComandante.ShowDialog();
    }

    private void tsmPesquisarComandante_Click(object sender, EventArgs e)
    {
        frmConsultarComandante objFrmConsultarComandante = new
frmConsultarComandante();
        objFrmConsultarComandante.ShowDialog();
    }
    }
}

#endregion

```

### 15.3 CÓDIGO CADASTRO DE PRAÇAS E OFICIAIS ( FRMFUNCIONARIO)

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;
using System.IO;
using SistemadeCadastroPolicial.BDPMDatasetTableAdapters;
using Word = Microsoft.Office.Interop.Word;
using Microsoft.Office.Core;

```

```

namespace SistemadeCadastroPolicial
{
    public partial class frmFuncionario : Form
    {
        public frmFuncionario()
        {
            InitializeComponent();
        }
        public string Nome,Filiacao,Posto, Endereco,Bairro, Cidade,
        TelResidencial,TelCelular, Email,DataNascimento,DataInclusao,DataPromocao, Anotacao,
        CPF, PASEP;
        public string RG;
        // public Image foto_array;
        public byte[] foto_array;

        #region Botões
        private void cadastroPolicialBindingNavigatorSaveItem_Click(object sender,
        EventArgs e)
        {
            this.Validate();
            this.cadastroPolicialBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.bDPMDDataSet);
        }
        private void frmFuncionario_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
            'bDPMDDataSet1.CadastroPolicial' table. You can move, or remove it, as needed.

            this.cadastroPolicialTableAdapter1.Fill(this.bDPMDDataSet1.CadastroPolicial);
            PreencherConsulta();
        }
        private void tsbtnVoltar_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void tsbtnSair_Click(object sender, EventArgs e)
        {
            if (MessageBox.Show("Você tem certeza que deseja sair do sistema?",
            "Mensagem do Sistema",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                Application.Exit();
            }
        }

        private void tsbtnConfirmarCad_Click(object sender, EventArgs e)
        {
            Gravar();
        }

        private void tsbtnConfirmarAlt_Click(object sender, EventArgs e)
        {
            Alterar();
        }

        private void tsbtnCancelar_Click(object sender, EventArgs e)
        {

```

```

        if (MessageBox.Show("Você tem certeza que deseja voltar e cancelar o
cadastro?", "Mensagem do Sistema",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            this.Dispose();
        }
    }

    private void btnFoto_Click(object sender, EventArgs e)
    {
        OpenFileDialog oFd1 = new OpenFileDialog();
        oFd1.Filter = "jpeg|*.jpg|bmp|*.bmp|all files|*.*";
        DialogResult res = oFd1.ShowDialog();

        if (res == DialogResult.OK)
        {
            picFoto.Image = Image.FromFile(oFd1.FileName);
        }
    }

    private void tsbtnAlterar_Click(object sender, EventArgs e)
    {
        DesabilitaBotoesAlterar();
        HabilitaBotoesAlterar();
        txtNome.Focus();
    }

    private void tsbtnCadastrar_Click(object sender, EventArgs e)
    {
        LimparCampos();
        DesabilitaBotoesCadastrar();
        HabilitaBotoesCadastrar();
        txtRG.Focus();
    }

    private void tsbtnExcluir_Click(object sender, EventArgs e)
    {
        Excluir();
    }

    private void ajudaDeCustoToolStripMenuItem_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqAjudadeCusto();
    }

    private void tsreqCartãoFUSPOM_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqCartaoFuspom();
    }

    private void tsreqconfecçãoDeAuxilioFardamento3ºSGTE2ºSGT_Click(object sender,
    EventArgs e)
    {
        PreencherDocWordReqConfAuxilioFardamento3ºSGTe2ºSGT();
    }

    private void tsreqarmasDeFogo_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqArmasdeFogo();
    }

    private void tsreqcALCULODEATRASADO_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqCalculodeAtrasado();
    }

    private void tsreqidentidade_Click(object sender, EventArgs e)
    {

```

```

        PreencherDocWordReqIdentidade();
    }
    private void tsreqidentidadeParaDependentes_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqIdentidadeDependente();
    }
    private void tsreqinclusãoDeTempoDeEnsinoAlunoAprendiz_Click(object sender,
EventArgs e)
    {
        PreencherDocWordReqInclusaoAlunoAprendiz();
    }
    private void tsreqinscriçãoCAS_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqInscricaoCas();
    }
    private void tsreqlicençaEspecialLE1ª_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqLicencaEspecial();
    }
    private void tsreqlicençaEspecialLTIPIInterrupção_Click(object sender,
EventArgs e)
    {
        PreencherDocWordReqLicencaEspecialLTIPIinterrupcao();
    }
    private void tsreqaquisiçãoDeArmasDeFogo_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqAquisicaodeArmadeFogo();
    }
    private void tsreqaquisiçãoDaPistolaCal40_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqAquisicaodaPistola40();
    }
    private void tsreqmudançaDeEstadoCivil_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqMudancadeEstadocivil();
    }
    private void tsreqtransferirArmasDeFogo_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqTransferirArmadeFogo();
    }
    private void tsreqinscriçãoPOEPP_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqInscricaoPOEPP();
    }
    private void tsreqsolicitaçãoUsoDeCelular_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqLicitacaoUsodeCelular();
    }
    private void tsreqinclusãoDeTempoINSS_Click(object sender, EventArgs e)
    {
        PreencherDocWordReqInclusaodeTempoINSS();
    }
    private void tsreqpermissãoParaPrestarConcurso_Click(object sender, EventArgs
e)
    {
        PreencherDocWordReqPermisaoparaPrestarConcurso();
    }
    private void tspartetermoDeTransferênciaArmaDeFogo_Click(object sender,
EventArgs e)
    {
        PreencherDocWordParteTermodeTransferenciaArmadeFogo();
    }
}

```

```

        private void tsparteInclusãoDeDependenteCartaoFUSPOM_Click(object sender,
EventArgs e)
        {
            PreencherDocWordParteInclusaodeDependenteCartãoFUSPOM();
        }
        private void tsparteaniversário_Click(object sender, EventArgs e)
        {
            PreencherDocWordParteAniversario();
        }
        private void tspartemudançaDeEndereço_Click(object sender, EventArgs e)
        {
            PreencherDocWordParteMudancadeEndereco();
        }

#endregion
#region Métodos

private void Gravar()
{
    bool camposValidados = true;

    try
    {
        OleDbConnection objConexao = new
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\\BDPM.accdb");

        string strConn = "INSERT INTO CadastroPolicial
(RG, Nome, Filiacao, Posto, Endereco, Bairro, " +
"Cidade, TelResidencial, TelCelular, Email, DataNascimento, DataInclusao, DataPromocao, Foto,
Anotacao, CPF, PASEP)" +
        "Values
(@RG, @Nome, @Filiacao, @Posto, @Endereco, @Bairro, @Cidade, @TelRes, @TelCel, @Email, @DtNasc, @
DtIncl, @DtProm, @Foto, @Anotacao, @CPF, @PASEP)";
        OleDbCommand objCommand = new OleDbCommand(strConn, objConexao);

        //RG
        if (!String.IsNullOrEmpty(txtRG.Text))
        {
            objCommand.Parameters.AddWithValue("@RG", txtRG.Text);

            camposValidados = true;
        }

        else
        {
            epErro.SetError(txtRG, "O campo RG é obrigatório!");
            camposValidados = false;
        }

        //Nome
        if (!String.IsNullOrEmpty(txtNome.Text))
        {
            objCommand.Parameters.AddWithValue("@Nome", txtNome.Text);

            camposValidados = true;
        }

        else
        {

```

```

        epErro.SetError(txtNome, "O campo Nome é obrigatório!");
        camposValidados = false;
    }

    //Filiação
    if (!String.IsNullOrEmpty(txtFiliacao.Text))
    {
        objCommand.Parameters.AddWithValue("@Filiacao", txtFiliacao.Text);

        camposValidados = true;
    }

    else
    {
        epErro.SetError(txtFiliacao, "O campo Filiação é obrigatório!");
        camposValidados = false;
    }

    //Posto
    if (!String.IsNullOrEmpty(cmbPosto.Text))
    {
        objCommand.Parameters.AddWithValue("@Posto", cmbPosto.Text);

        camposValidados = true;
    }

    else
    {
        epErro.SetError(cmbPosto, "O campo Posto é obrigatório!");
        camposValidados = false;
    }

    //Endereço
    {
        objCommand.Parameters.AddWithValue("@Endereco", txtEndereco.Text);

        camposValidados = true;
    }

    //Bairro
    {
        objCommand.Parameters.AddWithValue("@Bairro", txtBairro.Text);

        camposValidados = true;
    }

    //Cidade
    {
        objCommand.Parameters.AddWithValue("@Cidade", txtCidade.Text);

        camposValidados = true;
    }

    //Telefone Residencial
    {
        objCommand.Parameters.AddWithValue("@TelRes",
mskTelResidencial.Text);
    }

```

```

        camposValidados = true;
    }

    //Telefone Celular
    {
        objCommand.Parameters.AddWithValue("@TelCel", mskTelCelular.Text);

        camposValidados = true;
    }

    //Email
    {
        objCommand.Parameters.AddWithValue("@Email", txtEmail.Text);

        camposValidados = true;
    }

    //Data Nascimento
    if (!String.IsNullOrEmpty(mskDtNascimento.Text))
    {
        objCommand.Parameters.AddWithValue("@DtNasc",
mskDtNascimento.Text);
        camposValidados = true;
    }
    else
    {
        epErro.SetError(mskDtNascimento, "O campo Data de Nascimento é
obrigatório!");
        camposValidados = false;
    }

    //Data Inclusão
    if (!String.IsNullOrEmpty(mskDtInclusao.Text))
    {
        objCommand.Parameters.AddWithValue("@DtIncl", mskDtInclusao.Text);
        camposValidados = true;
    }
    else
    {
        epErro.SetError(mskDtInclusao, "O campo Data de Inclusão é
obrigatório!");
        camposValidados = false;
    }

    //Data Promoção
    {
        objCommand.Parameters.AddWithValue("@DtProm", mskDtPromocao.Text);
        camposValidados = true;
    }

    //Foto
    MemoryStream ms = new MemoryStream();
    //convertendo a foto para dados binários
    if (picFoto.Image != null)
    {
        ms = new MemoryStream();
        picFoto.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg);
        //byte[] foto_array = ms.ToArray();
    }

```



```

        byte[] foto_array = new byte[ms.Length];
        ms.Position = 0;
        ms.Read(foto_array, 0, foto_array.Length);
        objCommand.Parameters.AddWithValue("@Foto", foto_array);
        //foto_array = ms.ToArray();
        picFoto.Image = null;
    }
    else
    {
        epErro.SetError(picFoto, "O campo Foto é obrigatório!");
        camposValidados = false;
    }

    //Anotação
    {
        objCommand.Parameters.AddWithValue("@Anotacao", rtbAnotacao.Text);
        camposValidados = true;
    }

    //CPF
    {
        objCommand.Parameters.AddWithValue("@CPF", mskCPF.Text);
        camposValidados = true;
    }

    //PASEP
    {
        objCommand.Parameters.AddWithValue("@PASEP", mskPASEP.Text);
        camposValidados = true;
    }

    //Verifico se o retorno de minha variável camposValidados é true
    if (camposValidados)
    {
        objConexao.Open();
        //Uso o método ExecuteNonQuery para executar os comandos e
        realizar o Insert no banco

        objCommand.ExecuteNonQuery();

        //Fecho a conexão
        objConexao.Close();

        //Exibo a mensagem ao usuário de confirmação da inserção no banco
        MessageBox.Show("Registro inserido com sucesso!", "Mensagem",
        MessageBoxButtons.OK, MessageBoxIcon.Information);

        //Chamo o método para limpar os campos e dou o foco ao rgTextBox
        LimparCampos();
        txtRG.Focus();
    }
    else
    {
        //Exibo a mensagem ao usuário de erro

```

```

        camposValidados = false;
        MessageBox.Show("Ops, ocorreram erros!\n\nPreencha os campos e
tente novamente",
        "Mensagem", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
catch (Exception ex)
{
    MessageBox.Show("Ops, ocorreram erros!\n" + ex.Message);
}
}

private void Alterar()
{
    bool camposValidados = false;

    try
    {
        OleDbConnection objConexao = new
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\\BDPM.accdb");

        string strConn = "UPDATE CadastroPolicial SET
Nome=@Nome,Filiacao=@Filiacao,Posto=@Posto,Endereco=@Endereco,Bairro=@Bairro," +
"Cidade=@Cidade,TelResidencial=@TelRes,TelCelular=@TelCel,Email=@Email,DataNascimento=
@DtNasc,DataInclusao=@DtIncl,DataPromocao=@DtProm," +
"Foto=@Foto,Anotacao=@Anotacao, CPF=@CPF, PASEP=@PASEP where RG="
+ txtRG.Text;

        OleDbCommand objCommand = new OleDbCommand(strConn, objConexao);

        //Nome
        if (!String.IsNullOrEmpty(txtNome.Text))
        {
            objCommand.Parameters.AddWithValue("@Nome", txtNome.Text);
            camposValidados = true;
        }
        else
        {
            epErro.SetError(txtNome, "O campo Nome é obrigatório!");
            camposValidados = false;
        }

        //Filiação
        if (!String.IsNullOrEmpty(txtFiliacao.Text))
        {
            objCommand.Parameters.AddWithValue("@Filiacao", txtFiliacao.Text);
            camposValidados = true;
        }
        else
        {
            epErro.SetError(txtFiliacao, "O campo Filiação é obrigatório!");
            camposValidados = false;
        }

        //Posto
        if (!String.IsNullOrEmpty(cmbPosto.Text))

```

```

{
    objCommand.Parameters.AddWithValue("@Posto", cmbPosto.Text);
    camposValidados = true;
}

else
{
    epErro.SetError(cmbPosto, "O campo Posto é obrigatório!");
    camposValidados = false;
}

//Endereço

{
    objCommand.Parameters.AddWithValue("@Endereco", txtEndereco.Text);
    camposValidados = true;
}

//Bairro

{
    objCommand.Parameters.AddWithValue("@Bairro", txtBairro.Text);
    camposValidados = true;
}

//Cidade

{
    objCommand.Parameters.AddWithValue("@Cidade", txtCidade.Text);
    camposValidados = true;
}

//Telefone Residencial

{
    objCommand.Parameters.AddWithValue("@TelRes",
mskTelResidencial.Text);
    camposValidados = true;
}

//Telefone Celular

{
    objCommand.Parameters.AddWithValue("@TelCel", mskTelCelular.Text);
    camposValidados = true;
}

//Email

{
    objCommand.Parameters.AddWithValue("@Email", txtEmail.Text);
    camposValidados = true;
}

//Data Nascimento

if (!String.IsNullOrEmpty(mskDtNascimento.Text))
{
    objCommand.Parameters.AddWithValue("@DtNasc",
mskDtNascimento.Text);
    camposValidados = true;
}

```

```

else
{
    epErro.SetError(mskDtNascimento, "O campo Data de Nascimento é
obrigatório!");
    camposValidados = false;
}

//Data Inclusão

if (!String.IsNullOrEmpty(mskDtInclusao.Text))
{
    objCommand.Parameters.AddWithValue("@DtIncl", mskDtInclusao.Text);
    camposValidados = false;
}

else
{
    epErro.SetError(mskDtInclusao, "O campo Data de Inclusão é
obrigatório!");
    camposValidados = false;
}

//Data Promoção

{
    objCommand.Parameters.AddWithValue("@DtProm", mskDtPromocao.Text);
    camposValidados = true;
}

//Foto

MemoryStream ms = new MemoryStream();
//convertendo a foto para dados binários
if (picFoto.Image != null)
{
    ms = new MemoryStream();
    picFoto.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg);
    //byte[] foto_array = ms.ToArray();
    byte[] foto_array = new byte[ms.Length];
    ms.Position = 0;
    ms.Read(foto_array, 0, foto_array.Length);
    objCommand.Parameters.AddWithValue("@Foto", foto_array);
    //foto_array = ms.ToArray();
    picFoto.Image = null;
}
else
{
    epErro.SetError(picFoto, "O campo Foto é obrigatório!");
}

//Anotações

{
    objCommand.Parameters.AddWithValue("@Anotacao", rtbAnotacao.Text);
    camposValidados = true;
}

//CPF

{
    objCommand.Parameters.AddWithValue("@CPF", mskCPF.Text);
    camposValidados = true;
}

```

```

    }

    //PASEP

    {
        objCommand.Parameters.AddWithValue("@PASEP", mskPASEP.Text);
        camposValidados = true;
    }

    //Verifico se o retorno de minha variável camposValidados é true
    if (camposValidados)
    {
        objConexao.Open();
        //Uso o método ExecuteNonQuery para executar os comandos e
        realizar o Insert no banco

        objCommand.ExecuteNonQuery();

        //Fecho a conexão

        objConexao.Close();

        //Exibo a mensagem ao usuário de confirmação da inserção no banco

        MessageBox.Show("Registro alterado com sucesso!", "Mensagem",
        MessageBoxButtons.OK, MessageBoxIcon.Information);

        txtNome.Focus();
    }

    else
    {
        //Exibo a mensagem ao usuário de erro

        MessageBox.Show("Ops, ocorreram erros!\n\nPreencha os campos e
        tente novamente",

        "Mensagem", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
catch
{
    MessageBox.Show("Ops, ocorreram erros!\n\nPreencha os campos e tente
    novamente",

    "Mensagem", MessageBoxButtons.OK, MessageBoxIcon.Information);
}

private void Excluir()
{
    //Se o usuário confirmar a exclusão, crio a conexão com o banco e excluo o
    respectivo registro
    if (MessageBox.Show("Tem certeza que deseja excluir o registro?",
    "Mensagem do Sistema",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        try
        {
            OleDbConnection objConexao = new
            OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\\BDPM.accdb");
            string strConn = "DELETE FROM CadastroPolicial WHERE RG = @RG";

```

```

        OleDbCommand objCommand = new OleDbCommand(strConn, objConexao);
        OleDbDataAdapter ad = new OleDbDataAdapter(objCommand);
        BDPMDDataSet ds = new BDPMDDataSet();
        objCommand.Parameters.AddWithValue("@RG", txtRG.Text);
        objConexao.Open();
        objCommand.ExecuteNonQuery();
        objConexao.Close();
        MessageBox.Show("Registro excluído com sucesso!", "Mensagem",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void DesabilitaBotoesCadastrar()
{
    tsbtnAlterar.Enabled = false;
    tsbtnExcluir.Enabled = false;
    tsbtnSair.Enabled = false;
    tsbtnVoltar.Enabled = false;
    tsbtnDocumentos.Enabled = false;
}

private void HabilitaBotoesAlterar()
{
    txtNome.Enabled = true;
    cmbPosto.Enabled = true;
    txtFiliacao.Enabled = true;
    txtEndereco.Enabled = true;
    txtEmail.Enabled = true;
    txtCidade.Enabled = true;
    txtBairro.Enabled = true;
    mskTelResidencial.Enabled = true;
    mskTelCelular.Enabled = true;
    mskDtNascimento.Enabled = true;
    mskDtInclusao.Enabled = true;
    mskDtPromocao.Enabled = true;
    mskCPF.Enabled = true;
    mskPASEP.Enabled = true;
    btnFoto.Enabled = true;
    rtbAnotacao.Enabled = true;
    tsbtnConfirmarAlt.Enabled = true;
    tsbtnCancelar.Enabled = true;
}

private void DesabilitaBotoesAlterar()
{
    tsbtnExcluir.Enabled = false;
    tsbtnSair.Enabled = false;
    tsbtnVoltar.Enabled = false;
    tsbtnCadastrar.Enabled = false;
    tsbtnDocumentos.Enabled = false;
}

private void HabilitaBotoesCadastrar()
{
    txtRG.Enabled = true;
    txtNome.Enabled = true;

```

```

        cmbPosto.Enabled = true;
        txtFiliacao.Enabled = true;
        txtEndereco.Enabled = true;
        txtEmail.Enabled = true;
        txtCidade.Enabled = true;
        txtBairro.Enabled = true;
        mskTelResidencial.Enabled = true;
        mskTelCelular.Enabled = true;
        mskDtNascimento.Enabled = true;
        mskDtInclusao.Enabled = true;
        mskDtPromocao.Enabled = true;
        mskCPF.Enabled = true;
        mskPASEP.Enabled = true;
        btnFoto.Enabled = true;
        rtbAnotacao.Enabled = true;
        tsbtnConfirmarCad.Enabled = true;
        tsbtnCancelar.Enabled = true;
    }

    private void LimparCampos()
    {
        txtRG.Clear();
        txtNome.Clear();
        txtFiliacao.Clear();
        txtEndereco.Clear();
        txtBairro.Clear();
        txtCidade.Clear();
        mskTelResidencial.Clear();
        mskTelCelular.Clear();
        txtEmail.Clear();
        mskDtNascimento.Clear();
        mskDtInclusao.Clear();
        mskDtPromocao.Clear();
        mskCPF.Clear();
        mskPASEP.Clear();
        rtbAnotacao.Clear();
        epErro.Clear();
    }

    private void PreencherConsulta()
    {
        txtRG.Text = RG;
        txtNome.Text = Nome;
        txtFiliacao.Text = Filiacao;
        cmbPosto.Text = Posto;
        txtEndereco.Text = Endereco;
        txtBairro.Text = Bairro;
        txtCidade.Text = Cidade;
        mskTelResidencial.Text = TelResidencial;
        mskTelCelular.Text = TelCelular;
        txtEmail.Text = Email;
        mskDtNascimento.Text = DataNascimento;
        mskDtInclusao.Text = DataInclusao;
        mskDtPromocao.Text = DataPromocao;
        MemoryStream ms = new MemoryStream();
        //ms = new MemoryStream();
        picFoto.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg);
        //byte[] foto_array = ms.ToArray();
        foto_array = new byte[ms.Length];
        ms.Position = 0;
        ms.Read(foto_array, 0, foto_array.Length);
        foto_array = ms.ToArray();
    }

```

```

        picFoto.Image = null;
        picFoto.Image = Image.FromStream(ms);
        rtbAnotacao.Text = Anotacao;
        mskCPF.Text = CPF;
        mskPASEP.Text = PASEP;
    }

//Requerimentos

    private void PreencherDocWordReqAjudadeCusto()
    {
        frmComandante objFormComandante = new frmComandante();
        objFormComandante.ShowDialog();
        //Objeto a ser usado nos parâmetros opcionais

        object missing = System.Reflection.Missing.Value;

        //Abre a aplicação Word e faz uma cópia do documento mapeado

        Word.Application oApp = new Word.Application();

        object template = @"C:\\Partes e Requerimentos\\Requerimentos Prontos\\Ajuda de
Custo.doc";

        Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref missing,
ref missing);

        //Troca o conteúdo de alguns tags

        Word.Range oRng = oDoc.Range(ref missing, ref missing);

        object FindText = "[rg]";

        object ReplaceWith = txtRG.Text;

        object MatchWholeWord = true;

        object Forward = true;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = txtRG.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }

        FindText = "[nome]";

        ReplaceWith = txtNome.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = txtNome.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[Filiacao]";
    }

```



```

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtFiliacao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = cmbPosto.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[datanasc]";

ReplaceWith = mskDtNascimento.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtNascimento.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataincl]";

ReplaceWith = mskDtInclusao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtInclusao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataprom]";

ReplaceWith = mskDtPromocao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtPromocao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[NomeCmt]";

ReplaceWith = objFormComandante.txtNomeCmt.Text;

```

```

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = objFormComandante.txtNomeCmt.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[RGCmt]";

        ReplaceWith = objFormComandante.txtRGCmt;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = objFormComandante.txtRGCmt.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[PostoCmt]";

        ReplaceWith = objFormComandante.cmbPostoCmt.Top;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = objFormComandante.cmbPostoCmt.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        oApp.Visible = true;
    }

    private void PreencherDocWordReqCartaoFuspom()
    {
        frmComandante objFormComandante = new frmComandante();
        objFormComandante.ShowDialog();
        //Objeto a ser usado nos parâmetros opcionais

        object missing = System.Reflection.Missing.Value;

        //Abre a aplicação Word e faz uma cópia do documento mapeado

        Word.Application oApp = new Word.Application();

        object template = @"C:\\Partes e Requerimentos\\Requerimentos Prontos\\Cartão
FUSPOM.doc";

        Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref missing,
ref missing);

        //Troca o conteúdo de alguns tags

        Word.Range oRng = oDoc.Range(ref missing, ref missing);

        object FindText = "[rg]";

        object ReplaceWith = txtRG.Text;

        object MatchWholeWord = true;

```

```

object Forward = true;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtRG.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[nome]";

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtNome.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[Filiacao]";

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtFiliacao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = cmbPosto.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[datanasc]";

ReplaceWith = mskDtNascimento.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtNascimento.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataincl]";

```

```

ReplaceWith = mskDtInclusao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtInclusao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataprom]";

ReplaceWith = mskDtPromocao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtPromocao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[NomeCmt]";

ReplaceWith = objFormComandante.txtNomeCmt.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = objFormComandante.txtNomeCmt.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[RGCmt]";

ReplaceWith = objFormComandante.txtRGCmt;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = objFormComandante.txtRGCmt.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[PostoCmt]";

ReplaceWith = objFormComandante.cmbPostoCmt.Top;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = objFormComandante.cmbPostoCmt.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
oApp.Visible = true;
}

private void PreencherDocWordReqConfAuxilioFardamento3°SGTe2°SGT()
{

```

```

frmComandante objFormComandante = new frmComandante();
objFormComandante.ShowDialog();
//Objeto a ser usado nos parâmetros opcionais

object missing = System.Reflection.Missing.Value;

//Abre a aplicação Word e faz uma cópia do documento mapeado

Word.Application oApp = new Word.Application();

object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Confeção de auxilio fardamento (3º SGT e 2º SGT).doc";

Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

//Troca o conteúdo de alguns tags

Word.Range oRng = oDoc.Range(ref missing, ref missing);

object FindText = "[rg]";

object ReplaceWith = txtRG.Text;

object MatchWholeWord = true;

object Forward = true;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtRG.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[nome]";

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtNome.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[Filiacao]";

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtFiliacao.Text;

```

```

        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[posto]";

    ReplaceWith = cmbPosto.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = cmbPosto.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[datanasc]";

    ReplaceWith = mskDtNascimento.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtNascimento.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[dataincl]";

    ReplaceWith = mskDtInclusao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtInclusao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[dataprom]";

    ReplaceWith = mskDtPromocao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtPromocao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[NomeCmt]";

    ReplaceWith = objFormComandante.txtNomeCmt.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,

```

```

        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtNomeCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[RGCmt]";

    ReplaceWith = objFormComandante.txtRGCmt;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtRGCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[PostoCmt]";

    ReplaceWith = objFormComandante.cmbPostoCmt.Top;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.cmbPostoCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    oApp.Visible = true;
}
private void PreencherDocWordReqArmasdeFogo()
{
    frmComandante objFormComandante = new frmComandante();
    objFormComandante.ShowDialog();
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Armas de Fogo.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

    //Troca o conteúdo de alguns tags

    Word.Range oRng = oDoc.Range(ref missing, ref missing);

    object FindText = "[rg]";

    object ReplaceWith = txtRG.Text;

    object MatchWholeWord = true;

    object Forward = true;

```

```

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtRG.Text;
oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[nome]";

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtNome.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[Filiacao]";

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtFiliacao.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = cmbPosto.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[datanasc]";

ReplaceWith = mskDtNascimento.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskDtNascimento.Text;
oRng = oDoc.Range(ref missing, ref missing);
}

```



```

    }
    FindText = "[dataincl]";

    ReplaceWith = mskDtInclusao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtInclusao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[dataprom]";

    ReplaceWith = mskDtPromocao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtPromocao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[NomeCmt]";

    ReplaceWith = objFormComandante.txtNomeCmt.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtNomeCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[RGCmt]";

    ReplaceWith = objFormComandante.txtRGCmt;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtRGCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[PostoCmt]";

    ReplaceWith = objFormComandante.cmbPostoCmt.Top;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)

```

```

    {
        oRng.Text = objFormComandante.cmbPostoCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    oApp.Visible = true;
}
private void PreencherDocWordReqCalculodeAtrasado()
{
    frmComandante objFormComandante = new frmComandante();
    objFormComandante.ShowDialog();
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Cálculo de Atrasado.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

    //Troca o conteúdo de alguns tags

    Word.Range oRng = oDoc.Range(ref missing, ref missing);

    object FindText = "[rg]";

    object ReplaceWith = txtRG.Text;

    object MatchWholeWord = true;

    object Forward = true;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtRG.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }

    FindText = "[nome]";

    ReplaceWith = txtNome.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtNome.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[Filiacao]";

    ReplaceWith = txtFiliacao.Text;

```

```

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = txtFiliacao.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[posto]";

        ReplaceWith = cmbPosto.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = cmbPosto.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[datanasc]";

        ReplaceWith = mskDtNascimento.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = mskDtNascimento.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[dataincl]";

        ReplaceWith = mskDtInclusao.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = mskDtInclusao.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[dataprom]";

        ReplaceWith = mskDtPromocao.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = mskDtPromocao.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }

```

```

FindText = "[NomeCmt]";

ReplaceWith = objFormComandante.txtNomeCmt.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = objFormComandante.txtNomeCmt.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[RGCmt]";

ReplaceWith = objFormComandante.txtRGCmt;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = objFormComandante.txtRGCmt.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[PostoCmt]";

ReplaceWith = objFormComandante.cmbPostoCmt.Top;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = objFormComandante.cmbPostoCmt.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
oApp.Visible = true;
}
private void PreencherDocWordReqIdentidade()
{
frmComandante objFormComandante = new frmComandante();
objFormComandante.ShowDialog();
//Objeto a ser usado nos parâmetros opcionais

object missing = System.Reflection.Missing.Value;

//Abre a aplicação Word e faz uma cópia do documento mapeado

Word.Application oApp = new Word.Application();

object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Identidade.doc";

Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

//Troca o conteúdo de alguns tags

Word.Range oRng = oDoc.Range(ref missing, ref missing);

```

```

object FindText = "[rg]";

object ReplaceWith = txtRG.Text;

object MatchWholeWord = true;

object Forward = true;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtRG.Text;
oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[nome]";

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtNome.Text;
oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[Filiacao]";

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtFiliacao.Text;
oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = cmbPosto.Text;
oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[end]";

ReplaceWith = txtEndereco.Text;

```

```

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
            ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = txtEndereco.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[bairro]";

        ReplaceWith = txtBairro.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
            ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = txtBairro.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[munic]";

        ReplaceWith = txtCidade.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
            ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = txtCidade.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[telres]";

        ReplaceWith = mskTelResidencial.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
            ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = mskTelResidencial.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[telcel]";

        ReplaceWith = mskTelCelular.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
            ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = mskTelCelular.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[cpf]";

```

```

ReplaceWith = mskCPF.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskCPF.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[pasep]";

ReplaceWith = mskPASEP.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskPASEP.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[datanasc]";

ReplaceWith = mskDtNascimento.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskDtNascimento.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataincl]";

ReplaceWith = mskDtInclusao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskDtInclusao.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataprom]";

ReplaceWith = mskDtPromocao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskDtPromocao.Text;

```

```

        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[NomeCmt]";

    ReplaceWith = objFormComandante.txtNomeCmt.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtNomeCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[RGCmt]";

    ReplaceWith = objFormComandante.txtRGCmt;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtRGCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[PostoCmt]";

    ReplaceWith = objFormComandante.cmbPostoCmt.Top;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.cmbPostoCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    oApp.Visible = true;
}
private void PreencherDocWordReqIdentidadeDependente()
{
    frmComandante objFormComandante = new frmComandante();
    objFormComandante.ShowDialog();
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Identidade para Dependentes.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

    //Troca o conteúdo de alguns tags

```



```

Word.Range oRng = oDoc.Range(ref missing, ref missing);

object FindText = "[rg]";

object ReplaceWith = txtRG.Text;

object MatchWholeWord = true;

object Forward = true;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtRG.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[nome]";

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtNome.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[Filiacao]";

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtFiliacao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = cmbPosto.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[end]";

```

```

ReplaceWith = txtEndereco.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtEndereco.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[bairro]";

ReplaceWith = txtBairro.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtBairro.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[munic]";

ReplaceWith = txtCidade.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtCidade.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[telres]";

ReplaceWith = mskTelResidencial.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskTelResidencial.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[telcel]";

ReplaceWith = mskTelCelular.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskTelCelular.Text;
oRng = oDoc.Range(ref missing, ref missing);
}

```

```

    }
    FindText = "[cpf]";

    ReplaceWith = mskCPF.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskCPF.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[pasep]";

    ReplaceWith = mskPASEP.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskPASEP.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[datanasc]";

    ReplaceWith = mskDtNascimento.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtNascimento.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[dataincl]";

    ReplaceWith = mskDtInclusao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtInclusao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[dataprom]";

    ReplaceWith = mskDtPromocao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)

```

```

    {
        oRng.Text = mskDtPromocao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[NomeCmt]";

    ReplaceWith = objFormComandante.txtNomeCmt.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtNomeCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[RGCmt]";

    ReplaceWith = objFormComandante.txtRGCmt;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtRGCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[PostoCmt]";

    ReplaceWith = objFormComandante.cmbPostoCmt.Top;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.cmbPostoCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    oApp.Visible = true;
}
private void PreencherDocWordReqInclusaoAlunoAprendiz()
{
    frmComandante objFormComandante = new frmComandante();
    objFormComandante.ShowDialog();
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Inclusão de Tempo de Ensino-Aluno Aprendiz.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

```

```

//Troca o conteúdo de alguns tags

Word.Range oRng = oDoc.Range(ref missing, ref missing);

object FindText = "[rg]";

object ReplaceWith = txtRG.Text;

object MatchWholeWord = true;

object Forward = true;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtRG.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[nome]";

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtNome.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[Filiacao]";

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtFiliacao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = cmbPosto.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

```

```
FindText = "[datanasc]";

ReplaceWith = mskDtNascimento.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtNascimento.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataincl]";

ReplaceWith = mskDtInclusao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtInclusao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataprom]";

ReplaceWith = mskDtPromocao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtPromocao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[NomeCmt]";

ReplaceWith = objFormComandante.txtNomeCmt.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = objFormComandante.txtNomeCmt.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[RGCMt]";

ReplaceWith = objFormComandante.txtRGCMt;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
```

```

        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtRGCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[PostoCmt]";

    ReplaceWith = objFormComandante.cmbPostoCmt.Top;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.cmbPostoCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    oApp.Visible = true;
}
private void PreencherDocWordReqInscricaoCas()
{
    frmComandante objFormComandante = new frmComandante();
    objFormComandante.ShowDialog();
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Inscrição CAS.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

    //Troca o conteúdo de alguns tags

    Word.Range oRng = oDoc.Range(ref missing, ref missing);

    object FindText = "[rg]";

    object ReplaceWith = txtRG.Text;

    object MatchWholeWord = true;

    object Forward = true;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtRG.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }

    FindText = "[nome]";

```

```

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtNome.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[Filiacao]";

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtFiliacao.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = cmbPosto.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[datanasc]";

ReplaceWith = mskDtNascimento.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskDtNascimento.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataincl]";

ReplaceWith = mskDtInclusao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskDtInclusao.Text;
oRng = oDoc.Range(ref missing, ref missing);
}

```



```

    }
    FindText = "[dataprom]";

    ReplaceWith = mskDtPromocao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtPromocao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[NomeCmt]";

    ReplaceWith = objFormComandante.txtNomeCmt.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtNomeCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[RGCmt]";

    ReplaceWith = objFormComandante.txtRGCmt;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtRGCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[PostoCmt]";

    ReplaceWith = objFormComandante.cmbPostoCmt.Top;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.cmbPostoCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    oApp.Visible = true;
}
private void PreencherDocWordReqLicencaEspecial()
{
    frmComandante objFormComandante = new frmComandante();
    objFormComandante.ShowDialog();
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

```

```

//Abre a aplicação Word e faz uma cópia do documento mapeado

Word.Application oApp = new Word.Application();

object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Licença Especial (LE) 1ª.doc";

Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

//Troca o conteúdo de alguns tags

Word.Range oRng = oDoc.Range(ref missing, ref missing);

object FindText = "[rg]";

object ReplaceWith = txtRG.Text;

object MatchWholeWord = true;

object Forward = true;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtRG.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[nome]";

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtNome.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[Filiacao]";

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtFiliacao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

```

```

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = cmbPosto.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[datanasc]";

        ReplaceWith = mskDtNascimento.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = mskDtNascimento.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[dataincl]";

        ReplaceWith = mskDtInclusao.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = mskDtInclusao.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[dataprom]";

        ReplaceWith = mskDtPromocao.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = mskDtPromocao.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[NomeCmt]";

        ReplaceWith = objFormComandante.txtNomeCmt.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = objFormComandante.txtNomeCmt.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }

```

```

FindText = "[RGCmt]";

ReplaceWith = objFormComandante.txtRGCmt;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = objFormComandante.txtRGCmt.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[PostoCmt]";

ReplaceWith = objFormComandante.cmbPostoCmt.Top;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = objFormComandante.cmbPostoCmt.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
oApp.Visible = true;
}
private void PreencherDocWordReqLicencaEspecialLTIPIInterrupcao()
{
    frmComandante objFormComandante = new frmComandante();
    objFormComandante.ShowDialog();
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Licença Especial (LTIP) INTERRUPÇÃO.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

    //Troca o conteúdo de alguns tags

    Word.Range oRng = oDoc.Range(ref missing, ref missing);

    object FindText = "[rg]";

    object ReplaceWith = txtRG.Text;

    object MatchWholeWord = true;

    object Forward = true;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,

```

```

        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtRG.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }

    FindText = "[nome]";

    ReplaceWith = txtNome.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtNome.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[Filiacao]";

    ReplaceWith = txtFiliacao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtFiliacao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[posto]";

    ReplaceWith = cmbPosto.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = cmbPosto.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[datanasc]";

    ReplaceWith = mskDtNascimento.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtNascimento.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[dataincl]";

    ReplaceWith = mskDtInclusao.Text;

```

```

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
            ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = mskDtInclusao.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[dataprom]";

        ReplaceWith = mskDtPromocao.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
            ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = mskDtPromocao.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[NomeCmt]";

        ReplaceWith = objFormComandante.txtNomeCmt.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
            ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = objFormComandante.txtNomeCmt.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[RGCmt]";

        ReplaceWith = objFormComandante.txtRGCmt;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
            ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = objFormComandante.txtRGCmt.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[PostoCmt]";

        ReplaceWith = objFormComandante.cmbPostoCmt.Top;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
            ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = objFormComandante.cmbPostoCmt.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        oApp.Visible = true;

```

```

}
private void PreencherDocWordReqAquisicaodeArmadoFogo()
{
    frmComandante objFormComandante = new frmComandante();
    objFormComandante.ShowDialog();
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Aquisição de Armas de Fogo.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

    //Troca o conteúdo de alguns tags

    Word.Range oRng = oDoc.Range(ref missing, ref missing);

    object FindText = "[rg]";

    object ReplaceWith = txtRG.Text;

    object MatchWholeWord = true;

    object Forward = true;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtRG.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }

    FindText = "[nome]";

    ReplaceWith = txtNome.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtNome.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[Filiacao]";

    ReplaceWith = txtFiliacao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,

```

```

        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtFiliacao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[posto]";

    ReplaceWith = cmbPosto.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = cmbPosto.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[datanasc]";

    ReplaceWith = mskDtNascimento.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtNascimento.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[dataincl]";

    ReplaceWith = mskDtInclusao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtInclusao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[dataprom]";

    ReplaceWith = mskDtPromocao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtPromocao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[NomeCmt]";

    ReplaceWith = objFormComandante.txtNomeCmt.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref

```



```

MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
                    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
                    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = objFormComandante.txtNomeCmt.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[RGCmt]";

ReplaceWith = objFormComandante.txtRGCmt;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
                    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
                    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = objFormComandante.txtRGCmt.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[PostoCmt]";

ReplaceWith = objFormComandante.cmbPostoCmt.Top;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
                    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
                    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = objFormComandante.cmbPostoCmt.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
oApp.Visible = true;
}
private void PreencherDocWordReqAquisicaodaPistola40()
{
    frmComandante objFormComandante = new frmComandante();
    objFormComandante.ShowDialog();
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Aquisição da Pistola 40.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

    //Troca o conteúdo de alguns tags

    Word.Range oRng = oDoc.Range(ref missing, ref missing);

    object FindText = "[rg]";

    object ReplaceWith = txtRG.Text;

```

```

object MatchWholeWord = true;

object Forward = true;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtRG.Text;
oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[nome]";

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtNome.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[Filiacao]";

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtFiliacao.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = cmbPosto.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[datanasc]";

ReplaceWith = mskDtNascimento.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)

```

```

{
    oRng.Text = mskDtNascimento.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataincl]";

ReplaceWith = mskDtInclusao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtInclusao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataprom]";

ReplaceWith = mskDtPromocao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtPromocao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[NomeCmt]";

ReplaceWith = objFormComandante.txtNomeCmt.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = objFormComandante.txtNomeCmt.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[RGCmt]";

ReplaceWith = objFormComandante.txtRGCmt;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = objFormComandante.txtRGCmt.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[PostoCmt]";

ReplaceWith = objFormComandante.cmbPostoCmt.Top;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,

```

```

missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.cmbPostoCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    oApp.Visible = true;
}
private void PreencherDocWordReqMudancadeEstadocivil()
{
    frmComandante objFormComandante = new frmComandante();
    objFormComandante.ShowDialog();
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Requerimentos
    Prontos\\Mudança de estado civil.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

    //Troca o conteúdo de alguns tags

    Word.Range oRng = oDoc.Range(ref missing, ref missing);

    object FindText = "[rg]";

    object ReplaceWith = txtRG.Text;

    object MatchWholeWord = true;

    object Forward = true;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtRG.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }

    FindText = "[nome]";

    ReplaceWith = txtNome.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtNome.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
}

```

```

FindText = "[Filiacao]";

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtFiliacao.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = cmbPosto.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[datanasc]";

ReplaceWith = mskDtNascimento.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskDtNascimento.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataincl]";

ReplaceWith = mskDtInclusao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskDtInclusao.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataprom]";

ReplaceWith = mskDtPromocao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{

```

```

        oRng.Text = mskDtPromocao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[NomeCmt]";

    ReplaceWith = objFormComandante.txtNomeCmt.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtNomeCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[RGCmt]";

    ReplaceWith = objFormComandante.txtRGCmt;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtRGCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[PostoCmt]";

    ReplaceWith = objFormComandante.cmbPostoCmt.Top;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.cmbPostoCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    oApp.Visible = true;
}
private void PreencherDocWordReqTransferirArmadoFogo()
{
    frmComandante objFormComandante = new frmComandante();
    objFormComandante.ShowDialog();
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Transferir Armas de Fogo.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

```

```

//Troca o conteúdo de alguns tags

Word.Range oRng = oDoc.Range(ref missing, ref missing);

object FindText = "[rg]";

object ReplaceWith = txtRG.Text;

object MatchWholeWord = true;

object Forward = true;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtRG.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[nome]";

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtNome.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[Filiacao]";

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtFiliacao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = cmbPosto.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[datanasc]";

```

```

ReplaceWith = mskDtNascimento.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskDtNascimento.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataincl]";

ReplaceWith = mskDtInclusao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskDtInclusao.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataprom]";

ReplaceWith = mskDtPromocao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskDtPromocao.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[NomeCmt]";

ReplaceWith = objFormComandante.txtNomeCmt.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = objFormComandante.txtNomeCmt.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[RGCmt]";

ReplaceWith = objFormComandante.txtRGCmt;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = objFormComandante.txtRGCmt.Text;

```



```

        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[PostoCmt]";

    ReplaceWith = objFormComandante.cmbPostoCmt.Top;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.cmbPostoCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    oApp.Visible = true;
}
private void PreencherDocWordReqInscricaoPOEPP()
{
    frmComandante objFormComandante = new frmComandante();
    objFormComandante.ShowDialog();
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado
    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Inscrição POEPP.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

    //Troca o conteúdo de alguns tags

    Word.Range oRng = oDoc.Range(ref missing, ref missing);

    object FindText = "[rg]";

    object ReplaceWith = txtRG.Text;

    object MatchWholeWord = true;

    object Forward = true;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtRG.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }

    FindText = "[nome]";

    ReplaceWith = txtNome.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref

```

```

MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
                    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
                    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtNome.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[Filiacao]";

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
                    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
                    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtFiliacao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
                    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
                    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = cmbPosto.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[datanasc]";

ReplaceWith = mskDtNascimento.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
                    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
                    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtNascimento.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataincl]";

ReplaceWith = mskDtInclusao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
                    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
                    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtInclusao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataprom]";

```

```

ReplaceWith = mskDtPromocao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = mskDtPromocao.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[NomeCmt]";

ReplaceWith = objFormComandante.txtNomeCmt.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = objFormComandante.txtNomeCmt.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[RGCmt]";

ReplaceWith = objFormComandante.txtRGCmt;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = objFormComandante.txtRGCmt.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[PostoCmt]";

ReplaceWith = objFormComandante.cmbPostoCmt.Top;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = objFormComandante.cmbPostoCmt.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
oApp.Visible = true;
}
private void PreencherDocWordReqLicitacaoUsodeCelular()
{
frmComandante objFormComandante = new frmComandante();
objFormComandante.ShowDialog();
//Objeto a ser usado nos parâmetros opcionais

object missing = System.Reflection.Missing.Value;

//Abre a aplicação Word e faz uma cópia do documento mapeado

```

```

Word.Application oApp = new Word.Application();

object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\solicitação uso de celular.doc";

Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

//Troca o conteúdo de alguns tags

Word.Range oRng = oDoc.Range(ref missing, ref missing);

object FindText = "[rg]";

object ReplaceWith = txtRG.Text;

object MatchWholeWord = true;

object Forward = true;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtRG.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[nome]";

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtNome.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[Filiacao]";

ReplaceWith = txtFiliacao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtFiliacao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,

```

```

missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = cmbPosto.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[datanasc]";

ReplaceWith = mskDtNascimento.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtNascimento.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataincl]";

ReplaceWith = mskDtInclusao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtInclusao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[dataprom]";

ReplaceWith = mskDtPromocao.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = mskDtPromocao.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[NomeCmt]";

ReplaceWith = objFormComandante.txtNomeCmt.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = objFormComandante.txtNomeCmt.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[RGCmt]";

ReplaceWith = objFormComandante.txtRGCmt;

```

```

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = objFormComandante.txtRGCmt.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[PostoCmt]";

        ReplaceWith = objFormComandante.cmbPostoCmt.Top;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = objFormComandante.cmbPostoCmt.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        oApp.Visible = true;
    }
    private void PreencherDocWordReqInclusaodeTempoINSS()
    {
        frmComandante objFormComandante = new frmComandante();
        objFormComandante.ShowDialog();
        //Objeto a ser usado nos parâmetros opcionais

        object missing = System.Reflection.Missing.Value;

        //Abre a aplicação Word e faz uma cópia do documento mapeado

        Word.Application oApp = new Word.Application();

        object template = @"C:\\Partes e Requerimentos\\Requerimentos
Prontos\\Inclusão de Tempo-INSS.doc";

        Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

        //Troca o conteúdo de alguns tags

        Word.Range oRng = oDoc.Range(ref missing, ref missing);

        object FindText = "[rg]";

        object ReplaceWith = txtRG.Text;

        object MatchWholeWord = true;

        object Forward = true;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = txtRG.Text;

```

```

        oRng = oDoc.Range(ref missing, ref missing);
    }

    FindText = "[nome]";

    ReplaceWith = txtNome.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtNome.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[Filiacao]";

    ReplaceWith = txtFiliacao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtFiliacao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[posto]";

    ReplaceWith = cmbPosto.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = cmbPosto.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[datanasc]";

    ReplaceWith = mskDtNascimento.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtNascimento.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[dataincl]";

    ReplaceWith = mskDtInclusao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,

```

```

missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtInclusao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[dataprom]";

    ReplaceWith = mskDtPromocao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtPromocao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[NomeCmt]";

    ReplaceWith = objFormComandante.txtNomeCmt.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtNomeCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[RGCmt]";

    ReplaceWith = objFormComandante.txtRGCmt;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtRGCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[PostoCmt]";

    ReplaceWith = objFormComandante.cmbPostoCmt.Top;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.cmbPostoCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    oApp.Visible = true;
}
private void PreencherDocWordReqPermisaoparaPrestarConcurso()

```



```

{
    frmComandante objFormComandante = new frmComandante();
    objFormComandante.ShowDialog();
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Requerimentos
    Prontos\\Permissão para Prestar Concurso.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
    missing, ref missing);

    //Troca o conteúdo de alguns tags

    Word.Range oRng = oDoc.Range(ref missing, ref missing);

    object FindText = "[rg]";

    object ReplaceWith = txtRG.Text;

    object MatchWholeWord = true;

    object Forward = true;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
    MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
    missing,
    ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtRG.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }

    FindText = "[nome]";

    ReplaceWith = txtNome.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
    MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
    missing,
    ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtNome.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }

    FindText = "[Filiacao]";

    ReplaceWith = txtFiliacao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
    MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
    ref missing, ref missing, ref ReplaceWith, ref missing, ref
    missing,
    ref missing, ref missing, ref missing) == true)
    {

```

```

        oRng.Text = txtFiliacao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[posto]";

    ReplaceWith = cmbPosto.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = cmbPosto.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[datanasc]";

    ReplaceWith = mskDtNascimento.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtNascimento.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[dataincl]";

    ReplaceWith = mskDtInclusao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtInclusao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[dataprom]";

    ReplaceWith = mskDtPromocao.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskDtPromocao.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[NomeCmt]";

    ReplaceWith = objFormComandante.txtNomeCmt.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,

```

```

        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtNomeCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[RGCmt]";

    ReplaceWith = objFormComandante.txtRGCmt;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.txtRGCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[PostoCmt]";

    ReplaceWith = objFormComandante.cmbPostoCmt.Top;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = objFormComandante.cmbPostoCmt.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    oApp.Visible = true;
}

//Partes

private void PreencherDocWordParteTermodeTransferenciaArmadoFogo()
{
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Partes Prontas\\Termo de
Transferência- Arma de Fogo.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

    //Troca o conteúdo de alguns tags

    Word.Range oRng = oDoc.Range(ref missing, ref missing);

    object FindText = "[rg]";

    object ReplaceWith = txtRG.Text;

```

```

object MatchWholeWord = true;

object Forward = true;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtRG.Text;
oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[nome]";

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtNome.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = cmbPosto.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[end]";

ReplaceWith = txtEndereco.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
oRng.Text = txtEndereco.Text;
oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[bairro]";

ReplaceWith = txtBairro.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)

```

```

    {
        oRng.Text = txtBairro.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    FindText = "[munic]";

    ReplaceWith = txtCidade.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = txtCidade.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }

    FindText = "[cpf]";

    ReplaceWith = mskCPF.Text;

    while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
    {
        oRng.Text = mskCPF.Text;
        oRng = oDoc.Range(ref missing, ref missing);
    }
    oApp.Visible = true;
}
private void PreencherDocWordParteInclusaodeDependenteCartãoFUSPOM()
{
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Partes Prontas\\Parte
Inclusão de Dependente Cartao FUSPOM.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

    //Troca o conteúdo de alguns tags

    Word.Range oRng = oDoc.Range(ref missing, ref missing);

    object FindText = "[rg]";

    object ReplaceWith = txtRG.Text;

    object MatchWholeWord = true;

    object Forward = true;

    while (oRng.Find.Execute(ref FindText, ref missing, ref

```

```

MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
                    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
                    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtRG.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[nome]";

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
                    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
                    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtNome.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
                    ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
                    ref missing, ref missing, ref missing) == true)
{
    oRng.Text = cmbPosto.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

oApp.Visible = true;
}
private void PreencherDocWordParteAniversario()
{
    //Objeto a ser usado nos parâmetros opcionais

    object missing = System.Reflection.Missing.Value;

    //Abre a aplicação Word e faz uma cópia do documento mapeado

    Word.Application oApp = new Word.Application();

    object template = @"C:\\Partes e Requerimentos\\Partes
Prontas\\Aniversário.doc";

    Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

    //Troca o conteúdo de alguns tags

    Word.Range oRng = oDoc.Range(ref missing, ref missing);

    object FindText = "[rg]";

    object ReplaceWith = txtRG.Text;

```

```

        object MatchWholeWord = true;

        object Forward = true;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = txtRG.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }

        FindText = "[nome]";

        ReplaceWith = txtNome.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = txtNome.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }

        FindText = "[posto]";

        ReplaceWith = cmbPosto.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = cmbPosto.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }

        oApp.Visible = true;
    }
    private void PreencherDocWordParteMudancadeEndereco()
    {
        //Objeto a ser usado nos parâmetros opcionais

        object missing = System.Reflection.Missing.Value;

        //Abre a aplicação Word e faz uma cópia do documento mapeado

        Word.Application oApp = new Word.Application();

        object template = @"C:\\Partes e Requerimentos\\Partes Prontas\\Mudança de
Endereço.doc";

        Word.Document oDoc = oApp.Documents.Add(ref template, ref missing, ref
missing, ref missing);

        //Troca o conteúdo de alguns tags

```

```

Word.Range oRng = oDoc.Range(ref missing, ref missing);

object FindText = "[rg]";

object ReplaceWith = txtRG.Text;

object MatchWholeWord = true;

object Forward = true;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtRG.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[nome]";

ReplaceWith = txtNome.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtNome.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}

FindText = "[posto]";

ReplaceWith = cmbPosto.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = cmbPosto.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[end]";

ReplaceWith = txtEndereco.Text;

while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
ref missing, ref missing, ref missing) == true)
{
    oRng.Text = txtEndereco.Text;
    oRng = oDoc.Range(ref missing, ref missing);
}
FindText = "[bairro]";

```



```

        ReplaceWith = txtBairro.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = txtBairro.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }
        FindText = "[munic]";

        ReplaceWith = txtCidade.Text;

        while (oRng.Find.Execute(ref FindText, ref missing, ref
MatchWholeWord, ref missing, ref missing, ref Forward, ref missing,
        ref missing, ref missing, ref ReplaceWith, ref missing, ref
missing,
        ref missing, ref missing, ref missing) == true)
        {
            oRng.Text = txtCidade.Text;
            oRng = oDoc.Range(ref missing, ref missing);
        }

        oApp.Visible = true;
    }
}

#endregion

```

## 15.4 PESQUISA DE PRAÇAS E OFICIAIS (FRMCONSULTARFUNCIONARIO)

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;
using SistemadeCadastroPolicial.BDPMDataset1TableAdapters;
using System.IO;
namespace SistemadeCadastroPolicial
{
    public partial class frmConsultarFuncionario : Form
    {
        private string RG, Nome, Filiacao, Posto, Endereco, Bairro, Cidade,
        TelResidencial, TelCelular, Email, DataNascimento, DataInclusao, DataPromocao,
        Anotacao, CPF, PASEP;
        //private Image foto_array;
        byte[] foto_array;
        public frmConsultarFuncionario()
        {
            InitializeComponent();
        }
    }
}

```

```

public int linhaAtual = 0;
#region Botões

private void tsbtnVoltar_Click(object sender, EventArgs e)
{
    this.Close();
}
private void btnConsulta_Click(object sender, EventArgs e)
{
    Consultar();
}
private void rbtNome_CheckedChanged(object sender, EventArgs e)
{
    if (rbtNome.Checked)
    {
        txtConsulta.Text = string.Empty;
        txtConsulta.Focus();
    }
}
private void rbtRG_CheckedChanged(object sender, EventArgs e)
{
    if (rbtRG.Checked)
    {
        txtConsulta.Text = string.Empty;
        txtConsulta.Focus();
    }
}
private void rbtPosto_CheckedChanged(object sender, EventArgs e)
{
    if (rbtPosto.Checked)
    {
        txtConsulta.Text = string.Empty;
        txtConsulta.Focus();
    }
}
private void frmConsultarFuncionario_Load(object sender, EventArgs e)
{
    // TODO: This line of code loads data into the
    'BDPMDDataSet1.CadastroPolicial' table. You can move, or remove it, as needed.
    this.cadastroPolicialTableAdapter1.Fill(this.BDPMDDataSet1.CadastroPolicial);
}
private void tsbtnVoltarPesquisa_Click(object sender, EventArgs e)
{
    this.Close();
}
private void dgvDados_CellDoubleClick(object sender, DataGridViewCellEventArgs
e)
{
    BDPMDDataSet1 ds = new BDPMDDataSet1();
    linhaAtual = int.Parse(e.RowIndex.ToString());
    TransferirDados();
}

#endregion
#region Métodos
private void Consultar()
{
    try
    {

```

```

        OleDbConnection objConexao = new
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\\BDPM.accdb");

        if (rbtNome.Checked && !String.IsNullOrEmpty(txtConsulta.Text))
        {
            string strConn = ("select * from CadastroPolicial where Nome LIKE
(@Nome)");

            OleDbDataAdapter Da = new OleDbDataAdapter(strConn, objConexao);
            //DataTable dt = new DataTable();
            Da.SelectCommand.Parameters.AddWithValue("@Nome", txtConsulta.Text
+ "%");

            DataSet ds = new DataSet();
            Da.Fill(ds, "Nome");
            dgvDados.DataSource = ds.Tables["Nome"];
        }

        else if (rbtRG.Checked && !String.IsNullOrEmpty(txtConsulta.Text))
        {
            string strConn = ("select * from CadastroPolicial where RG LIKE
(@RG)");

            OleDbDataAdapter Da = new OleDbDataAdapter(strConn, objConexao);
            Da.SelectCommand.Parameters.AddWithValue("@RG", txtConsulta.Text +
"%");

            DataSet ds = new DataSet();
            Da.Fill(ds, "RG");
            dgvDados.DataSource = ds.Tables["RG"];
        }
        else
        {
            string strConn = ("select * from CadastroPolicial where Posto LIKE
(@Posto)");

            OleDbDataAdapter Da = new OleDbDataAdapter(strConn, objConexao);
            Da.SelectCommand.Parameters.AddWithValue("Posto", txtConsulta.Text
+ "%");

            DataSet ds = new DataSet();
            Da.Fill(ds, "Posto");
            dgvDados.DataSource = ds.Tables["Posto"];
        }
        toolStripStatusLabel2.Text = "Encontrado(s): \"\" + dgvDados.RowCount +
        "\" registros com a palavra \"\" + txtConsulta.Text + "\"\"; ;
    }

    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }

    //this.BDPoliciaMilitarTableAdapter.Fill(this.BDPoliciaMilitarDataSet.Nomes);
    }

    private void obterDadosGrid()
    {
        BDPMDDataSet1 ds = new BDPMDDataSet1();
        frmFuncionario ff = new frmFuncionario();
        //obtem os dados do datagridview da linha selecionada usando as posições
das colunas

        //a primeira coluna , a coluna 0 a segunda , a coluna 1 , e , assim por
diante

        Nome = dgvDados[1, linhaAtual].Value.ToString();
        Filiacao = dgvDados[2, linhaAtual].Value.ToString();
    }

```

```

        Posto = dgvDados[3, linhaAtual].Value.ToString();
        Endereco = dgvDados[4, linhaAtual].Value.ToString();
        Bairro = dgvDados[5, linhaAtual].Value.ToString();
        Cidade = dgvDados[6, linhaAtual].Value.ToString();
        TelResidencial = dgvDados[7, linhaAtual].Value.ToString();
        TelCelular = dgvDados[8, linhaAtual].Value.ToString();
        Email = dgvDados[9, linhaAtual].Value.ToString();
        DataNascimento = dgvDados[10, linhaAtual].Value.ToString();
        DataInclusao = dgvDados[11, linhaAtual].Value.ToString();
        DataPromocao = dgvDados[12, linhaAtual].Value.ToString();
        //foto_array =(Image)dgvDados[13,linhaAtual].Value;
        foto_array = (byte[])dgvDados[13,linhaAtual].Value;
        Anotacao = dgvDados[14, linhaAtual].Value.ToString();
        CPF = dgvDados[15, linhaAtual].Value.ToString();
        PASEP = dgvDados[16, linhaAtual].Value.ToString();
    }

    private void TransferirDados()
    {
        try
        {
            //obtem o RG do cliente a partir da linha selecionada no datagridview
            RG = dgvDados[0, linhaAtual].Value.ToString();

            if (linhaAtual >= 0 )
            {
                //obtem dados do datagridview e atribui as variaveis definidas no
                formulario ff

                obterDadosGrid();

                frmFuncionario ff = new frmFuncionario();

                foto_array = null;
                ff.RG = RG;
                ff.Nome = Nome;
                ff.Filiacao = Filiacao;
                ff.Posto = Posto;
                ff.Endereco = Endereco;
                ff.Bairro = Bairro;
                ff.Cidade = Cidade;
                ff.TelResidencial = TelResidencial;
                ff.TelCelular = TelCelular;
                ff.Email = Email;
                ff.DataNascimento = DataNascimento;
                ff.DataInclusao = DataInclusao;
                ff.DataPromocao = DataPromocao;
                ff.foto_array = foto_array ;
                ff.Anotacao = Anotacao;
                ff.CPF = CPF;
                ff.PASEP = PASEP;
                //exibe o formulario para altera#o

                ff.ShowDialog();

                //atualiza o grid e reexibe os dados

                dgvDados.Update();
            }
        }
    }
}

```

```

        catch (Exception ex)
        {
            MessageBox.Show("Erro..." + ex.Message);
        }
    }
}

#endregion

```

## 15.5 CÓDIGO CADASTRO DE COMANDANTE (FRMCOMANDANTE)

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Data.OleDb;

namespace SistemadeCadastroPolicial
{
    public partial class frmComandante : Form
    {
        public frmComandante()
        {
            InitializeComponent();
        }

        private void comandanteBindingNavigatorSaveItem_Click(object sender, EventArgs
e)
        {
            this.Validate();
            this.comandanteBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.bDPMDDataSet1);
        }

        private void frmComandante_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'bDPMDDataSet1.Comandante'
            table. You can move, or remove it, as needed.
            this.comandanteTableAdapter.Fill(this.bDPMDDataSet1.Comandante);
        }

        #region Botões
        private void btnFotocmt_Click(object sender, EventArgs e)
        {
            OpenFileDialog oFd1 = new OpenFileDialog();
            oFd1.Filter = "jpeg|*.jpg|bmp|*.bmp|all files|*.*";
            DialogResult res = oFd1.ShowDialog();

            if (res == DialogResult.OK)
            {
                picFotoCmt.Image = Image.FromFile(oFd1.FileName);
            }
        }
    }
}

```

```

}
private void tsbtnVoltar_Click(object sender, EventArgs e)
{
    this.Close();
}
private void tsbtnCadastroCmt_Click(object sender, EventArgs e)
{
    DesabilitaBotõesCadastrar();
    HabilitaBotoesCadastrar();
    LimparCampos();
}
private void tsbtnSair_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Você tem certeza que deseja sair do sistema?",
"Mensagem do Sistema",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        Application.Exit();
    }
}
private void tsbtnCancelar_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Você tem certeza que deseja voltar e cancelar o
cadastro?", "Mensagem do Sistema",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        this.Dispose();
    }
}
private void tsbtnAlterarCmt_Click(object sender, EventArgs e)
{
    DesabilitaBotoesAlterar();
    HabilitaBotoesAlterar();
}
private void tsbtnExcluiRCmt_Click(object sender, EventArgs e)
{
    Excluir();
}
private void tsbtnConfirmarCadastroCmt_Click(object sender, EventArgs e)
{
    Gravar();
}
private void tsbtnConfirmarAlteracaoCmt_Click(object sender, EventArgs e)
{
    Alterar();
}
#endregion

#region Métodos

private void DesabilitaBotõesCadastrar()
{
    tsbtnAlterarCmt.Enabled = false;
    tsbtnExcluiRCmt.Enabled = false;
    tsbtnSair.Enabled = false;
    tsbtnVoltar.Enabled = false;
}
private void HabilitaBotoesCadastrar()
{
    txtRGCMt.Enabled = true;
    txtNomeCmt.Enabled = true;

```

```

        cmbPostoCmt.Enabled = true;
        mskTelefoneCmt.Enabled = true;
        btnFotocmt.Enabled = true;
        rtbAnotacaoCmt.Enabled = true;
        tsbtnConfirmarCadastroCmt.Enabled = true;
        tsbtnCancelar.Enabled = true;
    }
    private void DesabilitaBotoesAlterar()
    {
        tsbtnExcluirCmt.Enabled = false;
        tsbtnSair.Enabled = false;
        tsbtnVoltar.Enabled = false;
        tsbtnCadastroCmt.Enabled = false;
    }
    private void HabilitaBotoesAlterar()
    {
        txtNomeCmt.Enabled = true;
        cmbPostoCmt.Enabled = true;
        btnFotocmt.Enabled = true;
        mskTelefoneCmt.Enabled = true;
        rtbAnotacaoCmt.Enabled = true;
        tsbtnConfirmarAlteracaoCmt.Enabled = true;
        tsbtnCancelar.Enabled = true;
    }
    private void LimparCampos()
    {
        txtRGCmt.Clear();
        txtNomeCmt.Clear();
        mskTelefoneCmt.Clear();
        rtbAnotacaoCmt.Clear();
        epErro.Clear();
    }
    private void Excluir()
    {
        //Se o usuário confirmar a exclusão, crio a conexão com o banco e excluo o
        //respectivo registro
        if (MessageBox.Show("Tem certeza que deseja excluir o registro?",
            "Mensagem do Sistema",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            try
            {
                OleDbConnection objConexao = new
                OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\\BDPM.accdb");
                string strConn = "DELETE FROM Comandante WHERE RG = @RG";

                OleDbCommand objCommand = new OleDbCommand(strConn, objConexao);
                OleDbDataAdapter ad = new OleDbDataAdapter(objCommand);
                BDPMDataset ds = new BDPMDataset();
                objCommand.Parameters.AddWithValue("@RG", txtRGCmt.Text);
                objConexao.Open();
                objCommand.ExecuteNonQuery();
                objConexao.Close();
                MessageBox.Show("Registro excluído com sucesso!", "Mensagem",
                MessageBoxButtons.OK, MessageBoxIcon.Information);

                //this.cadastroPolicialTableAdapter.Fill(bDPMDataset.CadastroPolicial);

                //this.cadastroPolicialTableAdapter.Update(this.bDPMDataset.CadastroPolicial);
                //ad.Fill(ds.CadastroPolicial);
            }
        }
    }

```

```

        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}
private void Gravar()
{
    bool camposValidados = true;

    try
    {
        OleDbConnection objConexao = new
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\\BDPM.accdb");

        string strConn = "INSERT INTO Comandante
(RG,Nome,Posto,Telefone,Anotacao,Foto) values
(@RG,@Nome,@Posto,@Telefone,@Anotacao,@Foto)";

        OleDbCommand objCommand = new OleDbCommand(strConn, objConexao);

        //RG
        if (!String.IsNullOrEmpty(txtRGCmt.Text))
        {
            objCommand.Parameters.AddWithValue("@RG", txtRGCmt.Text);
            camposValidados = true;
        }
        else
        {
            epErro.SetError(txtRGCmt, "O campo RG é obrigatório!");
            camposValidados = false;
        }

        //Nome
        if (!String.IsNullOrEmpty(txtNomeCmt.Text))
        {
            objCommand.Parameters.AddWithValue("@Nome", txtNomeCmt.Text);
            camposValidados = true;
        }
        else
        {
            epErro.SetError(txtNomeCmt, "O campo Nome é obrigatório!");
            camposValidados = false;
        }

        //Posto
        if (!String.IsNullOrEmpty(cmbPostoCmt.Text))
        {
            objCommand.Parameters.AddWithValue("@Posto", cmbPostoCmt.Text);
            camposValidados = true;
        }
        else
        {
            epErro.SetError(cmbPostoCmt, "O campo Posto é obrigatório!");

```



```

        camposValidados = false;
    }

    //Telefone

    {
        objCommand.Parameters.AddWithValue("@Telefone",
mskTelefoneCmt.Text);
        camposValidados = true;
    }

    //Anotação

    {
        objCommand.Parameters.AddWithValue("@Anotacao",
rtbAnotacaoCmt.Text);
        camposValidados = true;
    }

    //Foto

    MemoryStream ms = new MemoryStream();
    //convertendo a foto para dados binários
    if (picFotoCmt.Image != null)
    {
        ms = new MemoryStream();
        picFotoCmt.Image.Save(ms,
System.Drawing.Imaging.ImageFormat.Jpeg);
        //byte[] foto_array = ms.ToArray();
        byte[] foto_array = new byte[ms.Length];
        ms.Position = 0;
        ms.Read(foto_array, 0, foto_array.Length);
        objCommand.Parameters.AddWithValue("@Foto", foto_array);
        //foto_array = ms.ToArray();
        picFotoCmt.Image = null;
    }
    else
    {
        epErro.SetError(picFotoCmt, "O campo Foto é obrigatório!");
        camposValidados = false;
    }

    //Verifico se o retorno de minha variável camposValidados é true

    if (camposValidados)
    {
        objConexao.Open();
        //Uso o método ExecuteNonQuery para executar os comandos e
realizar o Insert no banco

        objCommand.ExecuteNonQuery();

        //Fecho a conexão

        objConexao.Close();

        //Exibo a mensagem ao usuário de confirmação da inserção no banco

        MessageBox.Show("Registro inserido com sucesso!", "Mensagem",
MessageBoxButtons.OK, MessageBoxIcon.Information);

        //Chamo o método para limpar os campos e dou o foco ao rGTextBox

```

```

        LimparCampos();
        txtRGCmt.Focus();
    }

    else
    {
        //Exibo a mensagem ao usuário de erro
        camposValidados = false;
        MessageBox.Show("Ops, ocorreram erros!\n\nPreencha os campos e
tente novamente",
            "Mensagem", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
catch (Exception ex)
{
    MessageBox.Show("Ops, ocorreram erros!\n" + ex.Message);
}
}

private void Alterar()
{
    bool camposValidados = true;

    try
    {
        OleDbConnection objConexao = new
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\\BDPM.accdb");

        string strConn = "UPDATE Comandante SET
Nome=@Nome,Posto=@Posto,Telefone=@Telefone,Anotacao=@Anotacao,Foto=@Foto where RG=" +
txtRGCmt.Text;

        OleDbCommand objCommand = new OleDbCommand(strConn, objConexao);

        //Nome

        if (!String.IsNullOrEmpty(txtNomeCmt.Text))
        {
            objCommand.Parameters.AddWithValue("@Nome", txtNomeCmt.Text);
            camposValidados = true;
        }

        else
        {
            epErro.SetError(txtNomeCmt, "O campo Nome é obrigatório!");
            //MessageBox.Show("O campo Nome é obrigatório", "Mensagem",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            camposValidados = false;
        }

        //Posto

        if (!String.IsNullOrEmpty(cmbPostoCmt.Text))
        {
            objCommand.Parameters.AddWithValue("@Posto", cmbPostoCmt.Text);
            camposValidados = true;
        }

        else
        {
            epErro.SetError(cmbPostoCmt, "O campo Posto é obrigatório!");

```

```

        camposValidados = false;
    }

    //Telefone

    {
        objCommand.Parameters.AddWithValue("@Telefone",
mskTelefoneCmt.Text);
        camposValidados = true;
    }

    //Anotações

    {
        objCommand.Parameters.AddWithValue("@Anotacao",
rtbAnotacaoCmt.Text);
        camposValidados = true;
    }

    //Foto

    MemoryStream ms = new MemoryStream();
    //convertendo a foto para dados binários
    if (picFotoCmt.Image != null)
    {
        ms = new MemoryStream();
        picFotoCmt.Image.Save(ms,
System.Drawing.Imaging.ImageFormat.Jpeg);
        //byte[] foto_array = ms.ToArray();
        byte[] foto_array = new byte[ms.Length];
        ms.Position = 0;
        ms.Read(foto_array, 0, foto_array.Length);
        objCommand.Parameters.AddWithValue("@Foto", foto_array);
        //foto_array = ms.ToArray();
        picFotoCmt.Image = null;
    }
    else
    {
        epErro.SetError(picFotoCmt, "O campo Foto é obrigatório!");
    }

    //Verifico se o retorno de minha variável camposValidados é true

    if (camposValidados)
    {
        objConexao.Open();
        //Uso o método ExecuteNonQuery para executar os comandos e
realizar o Insert no banco

        objCommand.ExecuteNonQuery();

        //Fecho a conexão

        objConexao.Close();

        //Exibo a mensagem ao usuário de confirmação da inserção no banco

        MessageBox.Show("Registro alterado com sucesso!", "Mensagem",
MessageBoxButtons.OK, MessageBoxIcon.Information);

        txtNomeCmt.Focus();
    }

```

```

        else
        {
            //Exibo a mensagem ao usuário de erro

            MessageBox.Show("Ops, ocorreram erros!\n\nPreencha os campos e
tente novamente",
                "Mensagem", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    catch
    {
        MessageBox.Show("Ops, ocorreram erros!\n\nPreencha os campos e tente
novamente",
            "Mensagem", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
}

#endregion

```

## 15.6 PESQUISA DE COMANDANTE (FRMCONSULTARCOMANDANTE)

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;

namespace SistemadeCadastroPolicia
{
    public partial class frmConsultarComandante : Form
    {
        public frmConsultarComandante()
        {
            InitializeComponent();
        }
        #region Botões

        private void btnConsultarCmt_Click(object sender, EventArgs e)
        {
            Consultar();
        }
        private void rbtNome_CheckedChanged(object sender, EventArgs e)
        {
            if (rbtNome.Checked)
            {
                txtConsultaCmt.Text = string.Empty;
                txtConsultaCmt.Focus();
            }
        }
        private void rbtRG_CheckedChanged(object sender, EventArgs e)
        {
            if (rbtRG.Checked)
            {
                txtConsultaCmt.Text = string.Empty;
            }
        }
    }
}

```

```

        txtConsultaCmt.Focus();
    }
}
private void rbtPosto_CheckedChanged(object sender, EventArgs e)
{
    if (rbtPosto.Checked)
    {
        txtConsultaCmt.Text = string.Empty;
        txtConsultaCmt.Focus();
    }
}
private void toolStripButton1_Click(object sender, EventArgs e)
{
    this.Close();
}
#endregion

#region Métodos

private void Consultar()
{
    try
    {
        OleDbConnection objConexao = new
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\\BDPM.accdb");

        if (rbtNome.Checked && !String.IsNullOrEmpty(txtConsultaCmt.Text))
        {
            string strConn = ("select * from Comandante where Nome LIKE
(@Nome)");

            OleDbDataAdapter Da = new OleDbDataAdapter(strConn, objConexao);
            //DataTable dt = new DataTable();
            Da.SelectCommand.Parameters.AddWithValue("@Nome",
txtConsultaCmt.Text + "%");
            DataSet ds = new DataSet();
            Da.Fill(ds, "Nome");
            dgvDadosCmt.DataSource = ds.Tables["Nome"];
        }

        else if (rbtRG.Checked && !String.IsNullOrEmpty(txtConsultaCmt.Text))
        {
            string strConn = ("select * from Comandante where RG LIKE (@RG)");
            OleDbDataAdapter Da = new OleDbDataAdapter(strConn, objConexao);
            Da.SelectCommand.Parameters.AddWithValue("@RG",
txtConsultaCmt.Text + "%");
            DataSet ds = new DataSet();
            Da.Fill(ds, "RG");
            dgvDadosCmt.DataSource = ds.Tables["RG"];
        }
        else
        {
            string strConn = ("select * from Comandante where Posto LIKE
(@Posto)");

            OleDbDataAdapter Da = new OleDbDataAdapter(strConn, objConexao);
            Da.SelectCommand.Parameters.AddWithValue("@Posto",
txtConsultaCmt.Text + "%");
            DataSet ds = new DataSet();
            Da.Fill(ds, "Posto");
            dgvDadosCmt.DataSource = ds.Tables["Posto"];
        }
    }
}

```

```

        toolStripStatusLabel1.Text = "Encontrado(s): \"" +
dgvDadosCmt.RowCount +
        "\" registros com a palavra \"" + txtConsultaCmt.Text + "\""; ;
    }

    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }

    //this.BDPoliciaMilitarTableAdapter.Fill(this.BDPoliciaMilitarDataSet.Nomes);
    }
}
}

#endregion

```

## 16 CRONOGRAMA PROPOSTO PARA O ESTÁGIO CURRICULAR OBRIGATÓRIO II – TCC DO 6º SEMESTRE

Atividades	Abr.	Mai.	Jun.	Jul.	Ago.	Set.	Out.	Nov.
Contato com a empresa								
Realização do estágio e redação de relatório								
Inserção dos relatórios								
Elaboração do Relatório								
Inserção do relatório no portfólio								
Inserção do pré-projeto no portfólio								
Elaboração do Estágio Curricular Obrigatório II - TCC								
Inserção do Estágio Curricular Obrigatório II - TCC								
Apresentação do Estágio Curricular Obrigatório II - TCC								

## 11 CONCLUSÃO

Através da experiência adquirida no programa de estágio, ficou evidente que ele é vital e indispensável para aquele que deseja se tornar um profissional capacitado e preparado para enfrentar e vencer todos os grandes desafios que a carreira lhe trará.

Toda a rotina e procedimentos que foram observados no estágio são de grande necessidade para todas as atividades que devem vir a ser realizadas. Onde pude acompanhar e entender de uma forma prática, todas as informações conceituais e teóricas que são abrangidos na faculdade.

Fica, portanto, a afirmação que para se tornar um grande profissional e possuir uma carreira de sucesso, devem-se aliar conceitos e estudos, que são aprendidos em faculdades, cursos, etc., junto ao conhecimento prático, do “dia-a-dia” da profissão.

## REFERÊNCIAS

ALTER, S. **Information Systems: a management perspective**. Menlo Park CA: Benjamin e Cummings, 2 ed. 1996.

BIO, S. R. **Sistemas de Informação: Um enfoque Gerencial**. São Paulo: Editora Atlas, 1988.

BOVO, J. M., FONSECA, S. A. & RIBOLDY, R. **A tecnologia da Informação no Processo de Modernização Promovido pelo PMAT-BNDES no Município de São José do Rio Preto**. II ENAPG 2006, São Paulo, Novembro, Anais do Evento.

CUNHA, M. A., MARQUES, E. V. & MEIRELLES, F. S. **Modelos de Gestão de Tecnologia de Informação no Setor Público Brasileiro**. I ENAPG 2004, Salvador, Setembro, Anais do Evento

DAVENPORT, T. H. **Reengenharia de Processos, como inovar na empresa através da tecnologia da informação**. São Paulo. 6 Edição. Editora Campus. 1996.

FLEURY, Maria Tereza Leme, FISCHER, Rosa Maria. **Cultura e poder nas organizações**. Rio de Janeiro: Atlas, 1996.

FREITAS, H., LESCA, H. **Competitividade Empresarial na Era da Informação**. Revista da administração da USP, São Paulo, Jul/Set. 1992.

FUNDASP. Fundação Desenvolvimento Administrativo. Estado de São Paulo. <http://www.fundap.sp.gov.br/publicacoes/TextosTecnicos/textec1.htm>. Acesso em março de 2008.

GUAGLIARDI, J. A. REZENDE, D. A. **Estratégias, Sistemas de Informação, Tecnologia da Informação, Planos e Planejamentos Municipais: Alinhamento e Contribuições em 2 Prefeituras de Capitais Brasileiras**. Balneário Camboriú. SLADE-Encontro Luso-Brasileiro Estratégia. 2006.

KLIKSBERG, Bernardo. **Como transformar o Estado, para além dos mitos e dogmas**. Brasília: ENAP, 1992 .

LAUDON, K. C. & LAUDON J. P. **Sistemas de Informação**. São Paulo. Editora LTC, 2000.

LOBLER, M., MARQUES, C. S. & VISENTINI, M. S. **Dimensionamento da Tecnologia da Informação no Setor Público: Em Estudo Exploratório nos Municípios do Estado do RS**. II ENAPG/2006, São Paulo, Novembro, Anais do Evento.

REZENDE, D. A. **Engenharia de software e sistemas de informação**. Rio de Janeiro: Brasport, 1999.

REZENDE, D. A. **Alinhamento da Tecnologia da Informação ao Planejamento Municipal: Análise da Prática de Gestão de uma Prefeitura**. ENANPAD 2004, Curitiba, Setembro, Anais do Evento.



TAVARES, Maria das Graças de Pinho. **Cultura organizacional: uma abordagem antropológica da mudança**. São Paulo: Qualitymark, 1991.

ZUBOFF, S. **Automatizar/Informatizar: As duas faces de tecnologia inteligente**. RAE – Revista Administração Empresas. São Paulo. Nov/Dez. 1994.

Web Aula - Sem 5 - Unidade 1 e 2 - Estágio Curricular Obrigatório I e II- TCC.