

MARIANE MOREIRA DE SOUZA

**SISCOMP: UMA PROPOSTA DE COMUNICAÇÃO ENTRE COPOM E
VIATURAS POLICIAIS MILITARES UTILIZANDO COMUNICAÇÃO WIRELESS**

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do Curso de Ciência da Computação para obter o título de Bacharel em Ciência da Computação

Orientador
Prof. André Luiz Zambalde

Lavras
Minas Gerais - Brasil
2005

MARIANE MOREIRA DE SOUZA

**SISCOMP: UMA PROPOSTA DE COMUNICAÇÃO ENTRE COPOM E
VIATURAS POLICIAIS MILITARES UTILIZANDO COMUNICAÇÃO WIRELESS**

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do Curso de Ciência da Computação para obter o título de Bacharel em Ciência da Computação

Aprovada em 17 de Janeiro de 2005

Cap.PM Antônio Claret dos Santos

Prof. André Luiz Zambalde
(Orientador)

Lavras
Minas Gerais - Brasil

Sumário

| | | |
|----------|--|----------|
| 1 | Introdução | 1 |
| 1.1 | Considerações Iniciais | 1 |
| 1.2 | Justificativas e Objetivos | 2 |
| 1.3 | Escopo do Trabalho | 4 |
| 2 | Revisão Bibliográfica | 5 |
| 2.1 | A revolução da Computação Móvel | 5 |
| 2.2 | Visão geral sobre Comunicação <i>Wireless</i> | 8 |
| 2.3 | Tecnologias de Transmissão <i>Wireless</i> | 10 |
| 2.4 | Segurança em Sistemas de Comunicação Militar | 14 |
| 2.5 | Segurança em Redes <i>Wireless</i> | 15 |
| 2.5.1 | Criptografia por Chave Privada | 18 |
| 2.5.2 | Criptografia por Chave Pública | 20 |
| 2.5.3 | Assinatura Digital | 21 |
| 2.5.4 | O protocolo <i>WTLS</i> | 23 |
| 2.5.5 | O Protocolo <i>WEP</i> - A segurança das <i>WLANS</i> do padrão 80211 | 23 |
| 2.6 | Linguagens de Programação para <i>PDA</i> s | 24 |
| 2.7 | A Tecnologia <i>Java</i> | 30 |
| 2.7.1 | A Linguagem de Programação <i>Java</i> | 30 |
| 2.7.2 | A Plataforma de Desenvolvimento <i>Java</i> | 32 |
| 2.8 | A Tecnologia <i>Waba</i> | 33 |
| 2.8.1 | Definição da tecnologia <i>Waba</i> | 34 |
| 2.8.2 | Características do <i>Waba</i> | 34 |
| 2.8.3 | Vantagens da tecnologia <i>Waba</i> | 36 |
| 2.9 | A Tecnologia <i>SuperWaba</i> | 37 |
| 2.9.1 | Como surgiu o <i>SuperWaba</i> | 37 |

| | | |
|----------|---|-----------|
| 2.9.2 | Diferenças entre o <i>Waba</i> e o <i>SuperWaba</i> | 39 |
| 2.9.3 | Diferenças entre o <i>J2ME</i> e o <i>SuperWaba</i> | 39 |
| 3 | Metodologia | 43 |
| 3.1 | Pesquisa-Ação | 43 |
| 3.2 | Ambiente de Desenvolvimento | 43 |
| 3.3 | Rede de Comunicação, Plataformas e Tecnologias | 44 |
| 4 | Resultados e Discussão | 47 |
| 4.1 | Modelagem do Sistema | 47 |
| 4.2 | O Sistema SisComPM | 62 |
| 4.3 | Adoção das tecnologias <i>Java</i> e <i>SuperWaba</i> e do SGBD <i>SQLServer</i> | 78 |
| 4.4 | O Pacote de Comunicação <i>PDA/SGBD</i> com utilização de Cripto- grafia por Chave Privada | 84 |
| 4.5 | Testes e Resultados Aplicados | 86 |
| 5 | Conclusões | 93 |
| 5.1 | Alcance dos Objetivos e Dificuldades | 93 |
| 5.2 | Propostas de trabalhos futuros | 94 |

Lista de Figuras

| | | |
|-----|---|----|
| 1.1 | Troca básica de mensagens entre COPOM e viaturas - Fonte: <i>Projeto SisComPM - Sistema de Comunicação Segura entre 190 e viaturas PM</i> [SISCOMP (2004)] | 3 |
| 1.2 | Funcionamento básico do sistema: acionamento de viaturas e retorno de informações referentes a esse acionamento - Fonte: <i>Projeto SisComPM - Sistema de Comunicação Segura entre 190 e viaturas PM</i> [SISCOMP (2004)] | 3 |
| 2.1 | Exemplos de dispositivos móveis encontrados no mercado - Fonte: <i>Programming Mobile Devices</i> [MOBILE (2004)] | 6 |
| 2.2 | Evolução do mercado de dispositivos móveis - Fonte: <i>Infotech Trends</i> [INFOTECH (2003)] | 7 |
| 2.3 | Visão de aplicações em Computação Móvel - Fonte: <i>Comunicação sem fio e Computação Móvel: Tecnologias, Desafios e Oportunidades</i> [LOUREIRO, et.Al. (2003)] | 7 |
| 2.4 | Exemplo de vários tipos de redes de comunicação <i>wireless</i> - Fonte: <i>Redes sem fio IEEE 802.11</i> [TOSO et.Al. (2004)] | 9 |
| 2.5 | Gráfico do <i>FHSS</i> - Fonte: <i>Adaptado de Bluetooth - Promessas de uma nova tecnologia</i> [MAIA (2003)] | 11 |
| 2.6 | Gráfico do <i>DSSS</i> - Fonte: <i>Bluetooth - Promessas de uma nova tecnologia</i> [MAIA (2003)] | 11 |
| 2.7 | Esquema Bluetooth - Fonte: <i>Bluetooth.org - The Official Bluetooth Membership Site</i> [BLUETOOTH (2004)] | 12 |
| 2.8 | Topologia básica de uma rede local <i>Wi-Fi</i> - Fonte: <i>Redes sem fio IEEE 802.11</i> [TOSO et.Al. (2004)] | 13 |
| 2.9 | Técnicas <i>ECM - Electronic CounterMeasuring</i> - Fonte: <i>Wireless Channel Security Tutorial</i> [RUSSEL (1997)] | 15 |

| | | |
|------|--|----|
| 2.10 | Segurança em vários tipos de redes de informação - Fonte: <i>Wireless Channel Security Tutorial</i> [RUSSEL (1997)] | 16 |
| 2.11 | Encryptando e Decryptando mensagens utilizando o método de criptografia por chave privada - Fonte: <i>Internet and Mobile Business: How To Program, Cap 6 Security</i> [DEITEL (2002)] | 19 |
| 2.12 | Esquema de distribuição de chaves utilizando criptografia por chave privada - Fonte: <i>Internet and Mobile Business: How To Program, Cap 6 Security</i> [DEITEL (2002)] | 20 |
| 2.13 | Encryptando e Decryptando uma mensagem utilizando o método de criptografia por chave pública - Fonte: <i>Internet and Mobile Business: How To Program, Cap 6 Security</i> [DEITEL (2002)] | 21 |
| 2.14 | Autenticação do Transmissor e Receptor utilizando o método de criptografia por chave pública - Fonte: <i>Internet and Mobile Business: How To Program, Cap 6 Security</i> [DEITEL (2002)] | 22 |
| 2.15 | Processo de compilação e interpretação de um programa desenvolvido na linguagem de programação <i>Java</i> - Fonte: <i>Avaliação da tecnologia J2ME no contexto de desenvolvimento de jogos multiplayers para celulares</i> [ASSIS (2003)] | 31 |
| 2.16 | Portabilidade apresentada pela linguagem <i>Java</i> - Fonte: <i>Avaliação da tecnologia J2ME no contexto de desenvolvimento de jogos multiplayers para celulares</i> [ASSIS (2003)] | 32 |
| 2.17 | Versões da linguagem <i>Java</i> e seus respectivos alvos de aplicação - Fonte: <i>Avaliação da tecnologia J2ME no contexto de desenvolvimento de jogos multiplayers para celulares</i> [ASSIS (2003)] | 33 |
| 2.18 | Executando uma aplicação <i>Waba</i> em máquinas virtuais <i>Java</i> - Fonte: <i>The WabaSoft Development Kit</i> [WABA (2003b)] | 35 |
| 2.19 | Executando uma aplicação <i>Waba</i> na <i>WabaVM</i> - Fonte: <i>The WabaSoft Development Kit</i> [WABA (2003b)] | 36 |
| 2.20 | Comparação entre <i>SuperWaba</i> e <i>J2ME</i> - Fonte: <i>SuperWaba - SuperWaba X Outras Plataformas</i> [SW (2004b)] | 41 |
| 4.1 | Diagrama de Caso de Uso - Esquema Geral - Fonte: Documentação do Sistema | 48 |
| 4.2 | Diagrama de Caso de Uso - Efetuar Logon no Sistema do COPOM - Fonte: Documentação do Sistema | 49 |
| 4.3 | Diagrama de Caso de Uso - Efetuar Logon no Sistema da Viatura - Fonte: Documentação do Sistema | 49 |

| | | |
|------|---|----|
| 4.4 | Diagrama de Caso de Uso - Efetuar Logoff no Sistema da Viatura - Fonte: Documentação do Sistema | 50 |
| 4.5 | Diagrama de Caso de Uso - Efetuar Logoff no Sistema do COPOM - Fonte: Documentação do Sistema | 50 |
| 4.6 | Diagrama de Caso de Uso - Acionar Viatura - Fonte: Documentação do Sistema | 51 |
| 4.7 | Diagrama de Caso de Uso - Ler Informações sobre o Acionamento - Fonte: Documentação do Sistema | 51 |
| 4.8 | Diagrama de Caso de Uso - Indicar Partida da Viatura - Fonte: Documentação do Sistema | 52 |
| 4.9 | Diagrama de Caso de Uso - Ler Informações sobre a Partida da Viatura - Fonte: Documentação do Sistema | 53 |
| 4.10 | Diagrama de Caso de Uso - Indicar Chegada da Viatura - Fonte: Documentação do Sistema | 54 |
| 4.11 | Diagrama de Caso de Uso - Ler Informações sobre a Chegada da Viatura - Fonte: Documentação do Sistema | 55 |
| 4.12 | Diagrama de Caso de Uso - Encerrar Ocorrência - Fonte: Documentação do Sistema | 56 |
| 4.13 | Diagrama de Caso de Uso - Ler Informações sobre o Encerramento da Ocorrência - Fonte: Documentação do Sistema | 57 |
| 4.14 | Diagrama de Caso de Uso - Enviar Mensagem para o COPOM - Fonte: Documentação do Sistema | 58 |
| 4.15 | Diagrama de Caso de Uso - Enviar Mensagem para a Viatura - Fonte: Documentação do Sistema | 58 |
| 4.16 | Diagrama de Caso de Uso - Ler Mensagem enviada pela Viatura - Fonte: Documentação do Sistema | 59 |
| 4.17 | Diagrama de Caso de Uso - Ler Mensagem enviada pelo COPOM - Fonte: Documentação do Sistema | 59 |
| 4.18 | Diagrama de Classes do Pacote <i>br.ufla.comp.siscomp.bean</i> - Fonte: Documentação do Sistema | 60 |
| 4.19 | Pacotes do Sistema Servidor - Fonte: Documentação do Sistema | 60 |
| 4.20 | Modelo Relacional - Banco de Dados SisCompM - Fonte: Documentação do Sistema | 61 |
| 4.21 | Sistema do COPOM - Tela de Logon - Fonte: Execução do Sistema | 62 |
| 4.22 | Sistema do COPOM - Tela inicial - Fonte: Execução do Sistema [SISCOMP (2004)] | 63 |

| | | |
|------|---|----|
| 4.23 | Sistema da Viatura - Tela de <i>logon</i> - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] | 64 |
| 4.24 | SisComPM - Viatura Não Acionada, aguardando acionamento - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] . . | 65 |
| 4.25 | SisComPM - Viatura Acionada, aguardando partida para o local da ocorrência - Fonte: Execução do Sistema [SISCOMP (2004)] | 65 |
| 4.26 | SisComPM - Viatura Acionada, partindo para o local da ocorrência - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] . | 66 |
| 4.27 | SisComPM - Viatura Acionada, no local da ocorrência - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] | 66 |
| 4.28 | SisComPM - Informações sobre viaturas conectadas ao sistema - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] . . | 67 |
| 4.29 | SisComPM - Acionamento da Viatura - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] | 68 |
| 4.30 | SisComPM - Informações sobre o acionamento recebidas pelo Sistema da Viatura - Fonte: Execução do Sistema[SISCOMP (2004)] | 68 |
| 4.31 | SisComPM - Informações sobre o local de partida da viatura para o atendimento da ocorrência enviadas pelo Sistema da Viatura - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] . . | 69 |
| 4.32 | SisComPM - Informações sobre o local de partida da viatura para o atendimento da ocorrência recebidas pelo Sistema do COPOM - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] . . | 69 |
| 4.33 | SisComPM - Informações sobre a chegada da viatura ao local da ocorrência enviadas pelo Sistema da Viatura - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] | 70 |
| 4.34 | SisComPM - Informações sobre a chegada da viatura ao local da ocorrência recebidas pelo Sistema do COPOM - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] | 70 |
| 4.35 | SisComPM - Informações sobre as providências tomadas no encerramento da ocorrência enviadas pelo Sistema da Viatura - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] | 71 |
| 4.36 | SisComPM - Informações sobre as providências tomadas no encerramento da ocorrência recebidas pelo Sistema do COPOM - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] . . | 71 |
| 4.37 | SisComPM - Aviso de novo acionamento recebido pelo Sistema da Viatura - Fonte: Execução do Sistema [SISCOMP (2004)] . . | 72 |

| | | |
|------|--|----|
| 4.38 | SisComPM - Mensagem enviada pelo Sistema da Viatura - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] | 74 |
| 4.39 | SisComPM - Aviso de nova mensagem recebida pelo Sistema do COPOM - Fonte: Execução do Sistema [SISCOMP (2004)] . . . | 75 |
| 4.40 | SisComPM - Nova mensagem recebida pelo Sistema do COPOM - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] . | 75 |
| 4.41 | SisComPM - Mensagem enviada pelo Sistema do COPOM - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] | 76 |
| 4.42 | SisComPM - Aviso de nova mensagem recebida pelo Sistema da Viatura - Fonte: Execução do Sistema [SISCOMP (2004)] . . . | 76 |
| 4.43 | Nova Mensagem recebida pelo Sistema da Viatura - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)] | 77 |
| 4.44 | Diagrama de Rede do projeto SisComPM - Fonte: <i>Projeto SisComPM - Sistema de Comunicação Segura entre 190 e viaturas PM</i> [SISCOMP (2004)] | 77 |
| 4.45 | Vantagens do <i>SuperWaba</i> em relação a outras plataformas de desenvolvimento - Fonte: <i>SuperWaba - SuperWaba X Outras Plataformas</i> [SW (2004b)] | 81 |
| 4.46 | Características do <i>SuperWaba</i> com relação à Máquina Virtual - Fonte: <i>SuperWaba - Características da Plataforma</i> [SW (2004a)] | 82 |
| 4.47 | Características do <i>SuperWaba</i> com relação a Bibliotecas de Extensão - Fonte: <i>SuperWaba-Características da Plataforma</i> [SW (2004a)] | 83 |
| 4.48 | Características do <i>SuperWaba</i> com relação a Entrada e Saída - Fonte: <i>SuperWaba - Características da Plataforma</i> [SW (2004a)] | 83 |
| 4.49 | Características do <i>SuperWaba</i> com relação a Interface com o usuário - Fonte: <i>SuperWaba - Características da Plataforma</i> [SW (2004a)] | 83 |
| 4.50 | Transmissão de Informações entre Sistema Servidor e Sistema da Viatura utilizando Criptografia por Chave Privada - Fonte: <i>Projeto SisComPM - Sistema de Comunicação Segura entre 190 e viaturas PM</i> [SISCOMP (2004)] | 86 |
| 4.51 | HP Jornada 560 - Fonte: <i>InfoSync Norge</i> [INFOSYNC (2004)] . . . | 87 |
| 4.52 | Palm Tungsten C - Fonte: <i>ARC Organisers</i> [ARC (2004)] | 87 |
| 4.53 | Ponto de acesso sem fios G - Fonte: <i>Lynksys</i> [LINKSYS (2004)] . . | 91 |

Lista de Tabelas

| | | |
|-----|--|----|
| 2.1 | Elementos de uma comunicação segura - Fonte: <i>Wireless Channel Security Tutorial</i> [RUSSEL (1997)] | 17 |
| 2.2 | Comparação entre o <i>Waba</i> e o <i>SuperWaba</i> - Fonte: <i>SuperWaba - SuperWaba X Outras Plataformas</i> [SW (2004b)] | 40 |
| 4.1 | <i>PDA</i> s compatíveis com a tecnologia <i>SuperWaba</i> e seus preços - Fonte: <i>Pesquisa realizada na internet nos sites oficiais dos vendedores</i> [PALM (2004)], [HP (2004)], [CASIO (2004)] | 90 |

*Aos meus pais Dario e Rita e à minha avó Marli Coimbra de Souza (in
memoriam)*

Agradecimentos

À Deus, por tudo que sempre realizou e realiza em minha vida, aos meus pais Dario e Rita, por todo o apoio, carinho e incentivo, ao meu namorado Humberto por todo o amor, amizade, paciência e ajuda nas horas mais difíceis, aos meus professores por tudo que aprendi, aos meus colegas pela amizade e companhia durante todos esses anos.

SisComPM: Uma proposta de comunicação entre COPOM e viaturas policiais militares utilizando comunicação wireless

Resumo

Há muito tempo a preocupação com a segurança na troca de informações tem despertado a atenção de pesquisadores e profissionais da área de segurança. Essa preocupação é ainda maior com relação à prestação de serviços à comunidade por parte de órgãos que promovem segurança pública, como a Polícia Militar, de modo que o sigilo das informações trocadas entre os membros da equipe é crucial para a garantia da eficiência no momento do atendimento de ocorrências. Baseando-se em registros históricos e em fatos que comprovam a vulnerabilidade do sistema de comunicação via rádio utilizado pela Polícia Militar de Minas Gerais (PMMG), este trabalho tem por objetivo principal apresentar uma proposta de desenvolvimento de um novo sistema de comunicação entre o Centro de Operações Policial Militar (COPOM) e viaturas policiais militares pertencentes ao estado de Minas Gerais, utilizando-se comunicação em redes wireless.

Palavras-Chave

Polícia Militar, Segurança Pública, Tecnologia da Informação, Comunicação *Wireless*, Tecnologia Móvel

SisComPM: a communication proposal between Military Police Operation Center (COPOM) and military police cars, using wireless communication

Abstract

Since a long time the concern about security in information exchange has called the attention of researchers and professionals in security area. This concern is still bigger when it deals with rendering of communitarian services by organizations that promote public security, like Military Police, in way that the exchanged information's secrecy is crucial to guarantee the efficiency in occurrence attendance. Based in historical registers and facts that prove the vulnerability of Radio Communication used by Minas Gerais Military Police, the main purpose of this work is to present a new communication system development proposal between Military Police Operation Center (COPOM) and military police cars from Minas Gerais, using wireless communication.

Keywords

Military Police, Public Security, Information Technology, Wireless Communication, Mobile Technology

Capítulo 1

Introdução

1.1 Considerações Iniciais

Atualmente existe uma grande preocupação com a segurança da informação compartilhada pelos meios de comunicação. Essa preocupação não é recente, e começou a se evidenciar nas épocas da Primeira e Segunda Guerra Mundial. Nesta época especialistas em segurança costumavam trabalhar sobre intensa pressão, tanto para garantir o sigilo da comunicação entre os aliados, como para quebrar o sigilo de informações inimigas, o que muitas vezes acabavam por decidir o desfecho da guerra. Também nessa época, expandia-se a pesquisa e utilização da comunicação via frequências de rádio, e desde então, a vulnerabilidade deste estilo de comunicação começou a ser notada.

Com o surgimento e popularização da Internet, aliados ao grande avanço tecnológico na área da informática, surgem novas alternativas de comunicação bastante eficientes, como a comunicação *wireless* que garante suporte ao paradigma da computação móvel, permitindo a troca de informações entre pessoas em toda e qualquer parte do mundo, de uma maneira mais rápida, independente do lugar onde estejam, com um maior grau de mobilidade.

Porém, à medida que surgem novos estilos de comunicação, surgem também novas preocupações relacionadas ao desenvolvimento de mecanismos de segurança que garantam da melhor forma possível o sigilo de informações compartilhadas, uma vez que é cada vez mais comum que atividades cotidianas sejam intermediadas pelo uso da tecnologia, e portanto, cada vez mais suscetíveis à falta de segurança apresentada pela rede.

1.2 Justificativas e Objetivos

Com base em registros históricos e no atendimento de uma ocorrência na cidade de Poços de Caldas, Minas Gerais, verificou-se uma grande vulnerabilidade apresentada pelo sistema de comunicação via rádio utilizado pela Polícia Militar de Minas Gerais. Nesta ocasião, durante um assalto a um condomínio fechado, a Polícia Militar descobriu que sua rede de rádio estava sendo monitorada pela quadrilha de assaltantes.

Neste sentido, o presente trabalho tem por objetivo descrever uma proposta de desenvolvimento de um novo sistema de comunicação entre o Centro de Operações Policial Militar (COPOM) e viaturas policiais militares que são empregadas diariamente no policiamento ostensivo, utilizando-se comunicação em redes *wireless*, visando uma maior segurança no processo de atendimento a ocorrências.

O sistema proposto será composto por três partes:

- Sistema do COPOM: sistema que o policial militar acessará de um computador *desktop* comum.
- Sistema da Viatura: sistema que o policial militar terá acesso por meio da utilização de um dispositivo móvel (*palmtop*).
- Sistema Servidor: sistema responsável pela comunicação entre o Sistema da Viatura e Servidor de Banco de Dados.

Visando atender requisitos de uma comunicação segura, além da criptografia apresentada pelos protocolos de segurança em redes *wireless*, as informações transmitidas e recebidas pelo Sistema da Viatura também serão codificadas utilizando criptografia antes de serem compartilhadas pela rede. Os métodos de criptografia serão descritos com detalhes no capítulo 2, e o método adotado será especificado na seção 4.4.

As propostas básicas para o sistema têm duas funcionalidades:

- Troca básica de mensagens entre o policial de plantão no COPOM e o policial de plantão na viatura, e entre policiais nas viaturas (figura 1.1).
- Acionamento de viaturas e retorno de informações referentes ao atendimento da ocorrência tais como partida da viatura, chegada da viatura no local e encerramento da ocorrência (figura 1.2).



Figura 1.1: Troca básica de mensagens entre COPOM e viaturas - Fonte: *Projeto SisComPM - Sistema de Comunicação Segura entre 190 e viaturas PM* [SISCOMP (2004)]

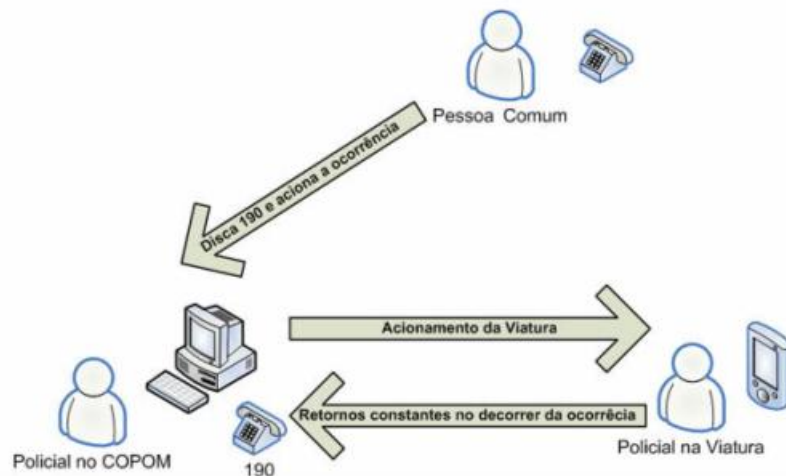


Figura 1.2: Funcionamento básico do sistema: acionamento de viaturas e retorno de informações referentes a esse acionamento - Fonte: *Projeto SisComPM - Sistema de Comunicação Segura entre 190 e viaturas PM* [SISCOMP (2004)]

Todas as informações trocadas pelos policiais serão capturadas pelo sistema, e armazenadas em um banco de dados, possibilitando a geração de informação histórica relacionada ao processo de atendimento de ocorrência, para posterior geração de relatórios via *web*, possibilitando um controle de estatísticas do sistema.

É importante destacar que este novo sistema não pretende substituir a rede de comunicação via rádio *VHF* utilizada pela Polícia Militar, mas ser um comple-

mento à mesma, disponibilizando uma alternativa segura de comunicação, no caso do atendimento de ocorrências de alta complexidade.

1.3 Escopo do Trabalho

No capítulo 2 serão apresentados vários conceitos fundamentais para o entendimento do trabalho: na seção 2.1 serão introduzidos conceitos básicos sobre Computação Móvel; nas seções 2.2 e 2.3 será dada uma visão geral sobre comunicação *wireless*, bem como serão apresentadas suas principais tecnologias de transmissão; na seção 2.4 será apresentada uma análise relativa à segurança em sistemas de comunicação militar; na seção 2.5 serão apresentados vários conceitos referentes à segurança em redes de comunicação *wireless*; nas seções 2.6 a 2.9 serão apresentadas algumas tecnologias pesquisadas para o desenvolvimento do projeto. No capítulo 3 será apresentada a Metodologia utilizada no desenvolvimento do trabalho, o ambiente de desenvolvimento e as tecnologias adotadas. No capítulo 4 serão apresentados a modelagem do sistema, suas principais características e funcionamento, as justificativas relacionadas às tecnologias adotadas, os resultados alcançados e os testes realizados. No capítulo 5 será apresentada uma conclusão sobre o trabalho.

Capítulo 2

Revisão Bibliográfica

O objetivo deste capítulo é apresentar os conceitos necessários ao completo entendimento do presente trabalho. As três primeiras seções introduzem o leitor ao novo ambiente da computação móvel e comunicação *wireless*, adotados no projeto. As duas seções seguintes apresentam uma análise referente à segurança em sistemas de comunicação militar, e em redes de comunicação *wireless*, que é o principal motivo do desenvolvimento deste trabalho. Por fim serão apresentadas as principais tecnologias relacionadas ao desenvolvimento do projeto, mostrando vantagens e desvantagens de cada uma delas.

2.1 A revolução da Computação Móvel

A Computação Móvel representa um novo paradigma computacional que tem como objetivo principal prover ao usuário acesso permanente a uma rede fixa ou móvel independente de sua posição física. É a capacidade de acessar informações em qualquer lugar e a qualquer momento [LOUREIRO, et.AI. (2003)].

Esse novo paradigma surge como uma quarta revolução na computação, antecedida pelos grandes centros de processamento de dados da década de sessenta, o surgimento dos terminais nos anos setenta, e as redes de computadores na década de oitenta [MATEUS, et.AI. (1998)].

Segundo Loureiro [LOUREIRO, et.AI. (2003)], a Computação Móvel está se tornando uma área madura e parece destinada a se tornar uma tecnologia dominante no futuro. O mercado de dispositivos móveis, genericamente chamados de *handhelds*, que englobam telefones celulares, *palms*, PDAs (*Personal Digital Assistants*), etc, está crescendo continuamente, sendo usado em aplicações que en-

volvem negócios, indústrias, escolas, hospitais, lazer, enfim, é uma tecnologia já bastante difundida atualmente. A figura 2.1 ilustra alguns exemplos de dispositivos móveis encontrados no mercado, e a figura 2.2 mostra a evolução do mercado desses dispositivos bem como sua previsão para os próximos anos.

Uma diferença importante entre o novo paradigma da Computação Móvel e os anteriores, é a interação entre o mesmo e as diversas áreas da Computação como Sistemas Digitais, Arquitetura de Computadores, Linguagens de Programação, Engenharia de *Software*, Interface Homem- Máquina, Compiladores, Banco de Dados, e outras áreas tais como Psicologia e Sociologia, que possuem o papel importante de definir novas formas de uso da tecnologia de processamento e comunicação de dados [LOUREIRO, et.A1. (2003)].

A figura 2.3 mostra as diversas aplicações da Computação Móvel no contexto do relacionamento com as diversas áreas descritas anteriormente.



Figura 2.1: Exemplos de dispositivos móveis encontrados no mercado - Fonte: *Programming Mobile Devices* [MOBILE (2004)]

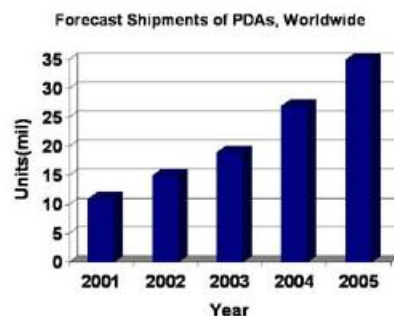


Figura 2.2: Evolução do mercado de dispositivos móveis - Fonte: *Infotech Trends* [INFOTECH (2003)]

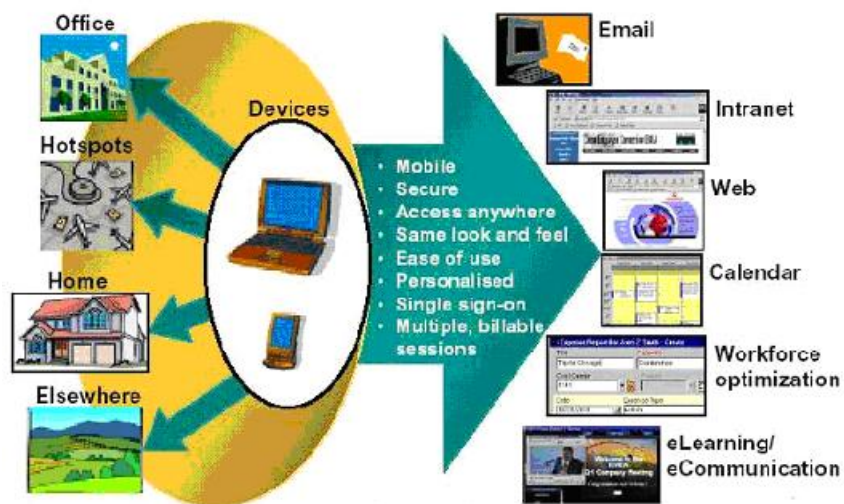


Figura 2.3: Visão de aplicações em Computação Móvel - Fonte: *Comunicação sem fio e Computação Móvel: Tecnologias, Desafios e Oportunidades* [LOUREIRO, et.AI. (2003)]

Como mencionado por Mateus [MATEUS, et.AI. (1998)], a principal característica desse novo paradigma, que é permitir mudanças de localização de seus usuários, ou seja, garantir a mobilidade dos mesmos no momento em que estes

usufruem seus serviços, é possível graças ao suporte oferecido pela comunicação sem fio (*wireless*) que elimina a necessidade do usuário manter-se conectado a uma infra-estrutura fixa e, em geral, estática.

Porém, os principais problemas relacionados à computação móvel se devem principalmente a essa mobilidade oferecida por esse novo paradigma, que introduz restrições inexistentes na computação tradicional formada por computadores estáticos. Alguns dos principais problemas são: localização de estações (antenas), gerência de localização e rastreamento de unidades móveis (celulares, *palms*, etc), interferências na propagação do sinal, alocação de frequências, gerenciamento de energia do aparelho, gerência de dados, segurança, dentre outros [MATEUS, et.AI. (1998)].

2.2 Visão geral sobre Comunicação *Wireless*

A Comunicação *Wireless* pode ser definida simplesmente como todo e qualquer tipo de comunicação que não utiliza fios (ou qualquer outro sistema físico) para se propagar. É o estilo de comunicação que dá suporte à Computação Móvel, permitindo à mesma garantir uma maior mobilidade aos seus usuários.

Segundo a empresa Sun Microsystems [SUN (2003b)], a comunicação *wireless* é um tipo de comunicação que está crescendo rapidamente e seu campo de atuação é enorme, sendo utilizada em transmissões de rádio e televisão, *paggers*, telefones celulares e em comunicações via satélite.

Outro segmento que se encontra em grande crescimento é o segmento das redes locais de comunicação *wireless* (*LANs Wireless*). Segundo Maia [MAIA (2003)], em ambientes corporativos, por exemplo, essas redes funcionam como uma boa opção, permitindo aos usuários o acesso a informações em tempo real de qualquer lugar dentro da organização. Esta mobilidade permite maior produtividade e oportunidades de serviço que não são possíveis em uma rede com fios.

As redes sem fio não dependem das conexões a cabo pois usam sinais eletromagnéticos (rádio e infravermelho) para transmitir os dados, usando o ar como meio de transmissão. Segundo Maia [MAIA (2003)], os dados a serem transmitidos são modulados em frequências de rádio para serem enviadas de um transmissor (antena) para um receptor (outra antena ou qualquer outro dispositivo sem fio). Vários dados podem estar sendo transmitidos e recebidos ao mesmo tempo sem interferirem uns nos outros, desde que suas frequências de modulação sejam diferentes.

Os protocolos utilizados na comunicação *wireless* devem ser aptos à detecção

de colisões e roteamento de dados entre uma rede cabeada, se este for o caso, e a rede sem fio. A pilha de protocolos mais conhecida e utilizada é o *TCP/IP*, sendo este também utilizado amplamente em redes cabeadas.

Outras características importantes a definir seriam a taxa de transferência de dados e o alcance em redes sem fio. Segundo Alves [ALVES (2002)], quanto maior a distância entre os pontos de transmissão, menor a taxa de transferência. Para distâncias de 5.5 Km, por exemplo, tem-se uma taxa de transmissão de 11 *Mbps*, enquanto que para distâncias de 15.8 Km, tem-se uma taxa de 1 *Mbps*. O alcance em redes sem fio é geralmente pequeno, podendo variar desde poucos metros até cerca de 70 Km.

Tratando-se de uma tecnologia avançada em relação a outras, geralmente seus custos também são bem maiores que os custos das redes cabeadas, porém, para Mateus [MATEUS, et.AI. (1998)], os benefícios da comunicação sem fio compensam os maiores gastos atribuídos à mesma.

A figura 2.4 mostra um exemplo da utilização de vários tipos de redes de comunicação *wireless*.

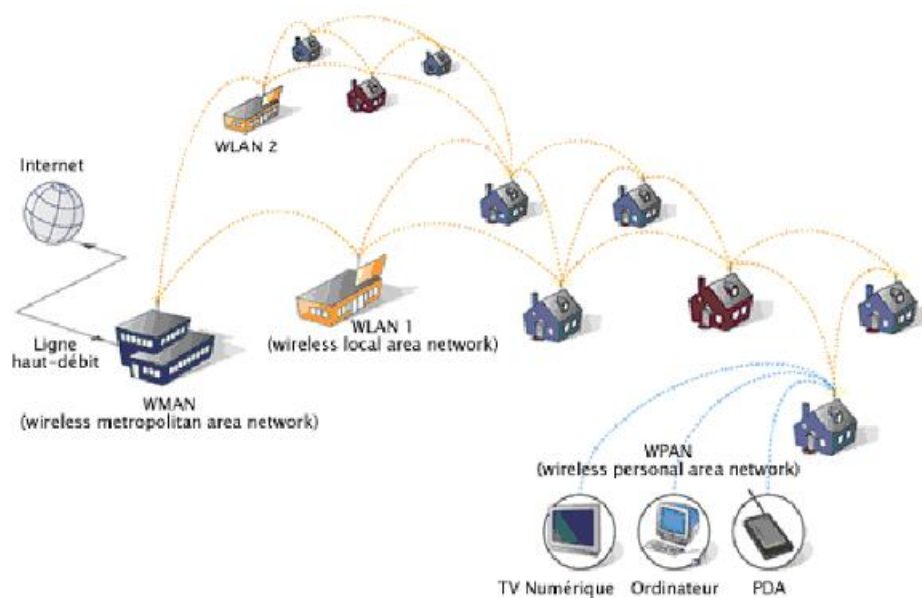


Figura 2.4: Exemplo de vários tipos de redes de comunicação *wireless* - Fonte: *Redes sem fio IEEE 802.11* [TOSO et.AI. (2004)]

2.3 Tecnologias de Transmissão *Wireless*

Em ambientes *wireless* existem vários tipos de tecnologias de transmissão. Em uma camada mais baixa, as mais utilizadas são transmissões por frequências de rádio e transmissões por Infravermelho, enquanto que em camadas mais altas essas técnicas são aplicadas nos padrões *Bluetooth* e *Wi-fi*.

A tecnologia de transmissão por frequências de rádio é a tecnologia mais utilizada, devido ao fato de ser menos sensível a interferências do meio que as demais, além de possuir uma grande largura de banda passante.

Porém, a eficiência de transmissão via rádio depende, dentre outros fatores, da antena utilizada, potência de transmissão, e existência de relevo ou meios interferentes [MATEUS, et.AI. (1998)].

Essa tecnologia de transmissão, também conhecida como *spread spectrum*, exige que o receptor saiba a frequência exata do sinal para poder reconhecê-lo; caso ele não esteja sintonizado na frequência certa, o sinal é escutado como uma interferência (ruído). A tecnologia *spread spectrum* se divide em dois tipos principais [MAIA (2003)]:

- ***FHSS (Frequency Hooping Spread Spectrum) Espectro de Frequência Espalhada em Saltos:*** transmissão de sinal aleatório de frequências de rádio. Os sistemas *FHSS* transmitem sinais em faixas de frequência bem estreitas, porém eles superam interferências potenciais, fazendo a frequência saltar de um valor para outro dentro de uma largura de faixa maior.
- ***DSSS (Direct Sequence Spread Spectrum) Espectro de Frequência Espalhada em Sequência Direta:*** Em uma camada mais alta é conhecida também como *CDMA - Code-Division Multiple Access*, uma tecnologia de transmissão amplamente utilizada em sistemas de telefonia celular. O transmissor *DSSS* converte dados em símbolos que representam, cada um, um conjunto de *bits*. Essa informação é convertida em um sinal que alimenta o transmissor, que também produz um sinal. O receptor utiliza um correlator baseado no código de difusão e remove os *bits*, recuperando os dados. Esta técnica supera ameaças de interferências de sinal ao transmitir os dados através de várias frequências para que os receptores façam a recomposição do sinal. É a técnica mais utilizada em redes de comunicação *wireless*.

As figuras 2.5 e 2.6 ilustram os gráficos representativos de cada uma dessas técnicas.

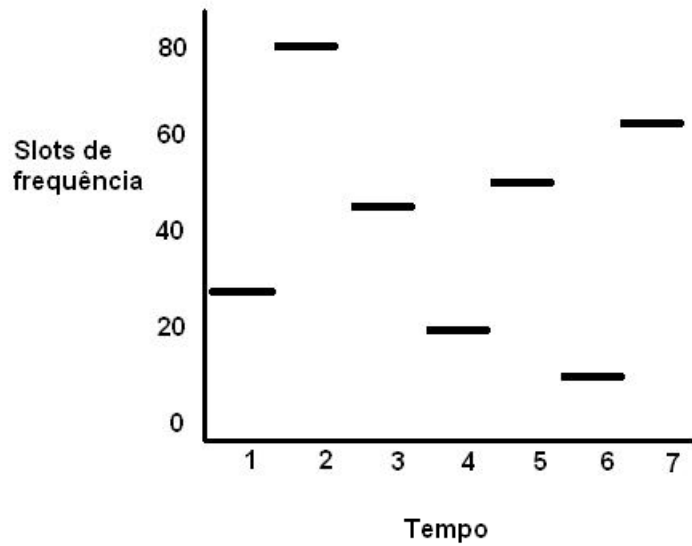


Figura 2.5: Gráfico do FHSS - Fonte: Adaptado de *Bluetooth - Promessas de uma nova tecnologia* [MAIA (2003)]

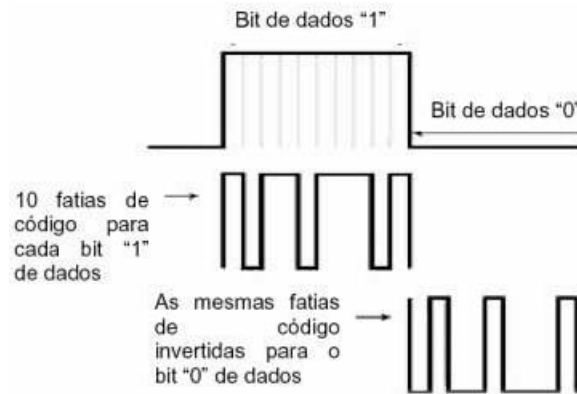


Figura 2.6: Gráfico do DSSS - Fonte: *Bluetooth - Promessas de uma nova tecnologia* [MAIA (2003)]

As redes *wireless* baseadas em *IR (Infra-Red)* ou Infravermelho utilizam a mesma tecnologia apresentada em produtos como controles remotos de aparelhos

de televisão. Segundo Maia [MAIA (2003)], os sinais de *IR* não conseguem penetrar em objetos opacos e podem ser facilmente obstruídos, e por isso sua utilização exige visada direta entre dois pontos a serem conectados, ou a utilização de transmissão por difusão (reflexão). Além disso, a distância da transmissão direcionada é muito pequena (poucos metros) e seu uso é restrito a Redes Pessoais (*PANs*) e algumas aplicações específicas em Redes Locais sem fio (*WLANs*).

Bluetooth é um padrão proposto pelo *Bluetooth SIG (Special Interest Group)*, que é um consórcio das maiores empresas de telecomunicações e computação do mundo. O padrão opera na faixa de $2,4\text{ GHz}$ e tem como princípio propor uma tecnologia de baixo custo, para conectividade sem fio [BLUETOOTH (2004)].

A estrutura básica de comunicação do *Bluetooth* é chamada de *piconet*. *Piconet* é uma rede onde um nodo central (mestre) se comunica em forma de topologia estrela com os demais nodos (escravos) da rede, com no máximo sete elementos [LOUREIRO, et.Al. (2003)].

O pequeno alcance de comunicação dos dispositivos (alcance máximo de cerca de 10 metros) faz com que essas piconets possuam a característica de formarem pequenas redes pessoais conhecidas como *WPAN - Wireless Personal Area Network* [LOUREIRO, et.Al. (2003)].

Em uma camada mais baixa a tecnologia *Bluetooth* utiliza tanto a transmissão por ondas de rádio como por Infravermelho.

A figura 2.7 mostra um esquema da tecnologia *Bluetooth*.



Figura 2.7: Esquema Bluetooth - Fonte: *Bluetooth.org - The Official Bluetooth Membership Site* [BLUETOOTH (2004)]

O *Wi-Fi (Wireless Fidelity)* também conhecido como *WLAN (Wireless Local Area Network)*, é uma tecnologia de redes locais *wireless* que se estendem por

uma pequena área, e que utilizam altas frequências de ondas de rádio, ao invés de fios, para a comunicação entre dispositivos móveis.[DEITEL (2002)]

É uma tecnologia que, apesar de possuir pequenos alcances, permite acesso em banda larga, e está implementada sobre o padrão *IEEE 802.11*, que opera em frequências que variam entre 2,4 GHz e 5 GHz, e que fornece aos seus usuários o acesso a redes públicas e privadas de uma forma simples e cômoda, possibilitando alta mobilidade, flexibilidade e conveniência no acesso à informação, além da facilidade de montagem e manutenção [TOSO et.Al. (2004)].

A tecnologia *Wi-Fi* pode ser constituída de várias formas dependendo do tipo de rede em que ela se situa [TOSO et.Al. (2004)]:

- Rede Local: utilização de *Access-Points*, que são considerados *switches* de uma rede sem fio, e adaptadores *Wi-Fi* simples.
- Rede Metropolitana: utilização de antenas e adaptadores *Wi-Fi* com *pigtails*.
- Rede Ampla: utilização de antenas, amplificadores, adaptadores *Wi-Fi* com *pigtails*.

A figura 2.8 mostra a topologia básica de uma rede local *Wi-Fi*.

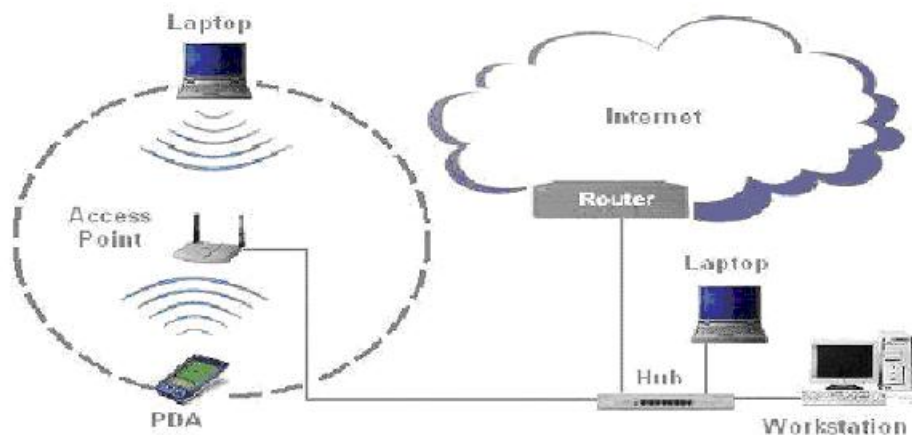


Figura 2.8: Topologia básica de uma rede local *Wi-Fi* - Fonte: *Redes sem fio IEEE 802.11* [TOSO et.Al. (2004)]

No processo de implementação de redes *Wi-Fi*, volta-se atenção para locais estratégicos também chamados de *Hot-Spots*, que são locais frequentados por profissionais em viagem, como por exemplo, hotéis, aeroportos, centros de reuniões,

zonas comerciais, ou estádios de futebol, onde parece ser necessário se ter acesso à *Internet*.

2.4 Segurança em Sistemas de Comunicação Militar

O sistema de comunicação à rádio utilizado pelos militares, e vigente até os dias de hoje, começou a ser utilizado há algum tempo atrás. Como mencionado por Mateus [MATEUS, et.AI. (1998)], em 1939, durante a Segunda Guerra Mundial, foi quando se expandiu intensamente a pesquisa e uso da comunicação via rádio.

Segundo o Departamento Americano do Estado [US.DEP (2003)], a comunicação via rádio consiste na utilização de ondas eletromagnéticas em uma faixa de frequência de rádio, para transmitir ou receber sinais elétricos sem a existência de fios conectando o emissor e receptor.

Um repetidor de rádio recebe o sinal de rádio a uma frequência e retransmite o sinal em uma outra frequência e/ou nível [US.DEP (2003)].

A principal vantagem da utilização da comunicação via rádio, no contexto militar é a de que o contato via rádio é ideal para a disseminação rápida da informação durante situações de emergência. O *broadcast*, envio de mensagens para todos os participantes do sistema, da comunicação à rádio, provém uma imediata e simultânea comunicação com múltiplos participantes do sistema [US.DEP (2003)]. Este é o principal motivo pelo qual esse tipo de comunicação é utilizado até nos dias de hoje não só por militares, policiais, bombeiros e outros agentes do Sistema de Defesa Social, mas também por outros sistemas comerciais, hospitais, sistemas aéreos, frotas de táxi etc [MATEUS, et.AI. (1998)].

Porém, o maior problema encontrado em sistemas de comunicação via rádio, seria a falta de segurança apresentada pelo mesmo. Segundo a companhia Ces Communication, [CES (2003)], é um tipo de comunicação naturalmente inseguro, pois uma vez que se trata de uma tecnologia de *broadcast*, qualquer um dentro da faixa de frequência de recepção, tem acesso ao sinal do transmissor, cabendo ao invasor apenas sintonizar a frequência desejada.

Na época das guerras, em que este sistema foi bastante utilizado, o inimigo empregava várias técnicas eletrônicas para detectar a presença dos sinais de rádio com o intuito de interrompê-los de chegar ao seu destino, ou mesmo explorá-los. Essas técnicas são conhecidas como *ECM (Electronic CounterMeasuring)* [RUSSEL (1997)].

Os sinais podem ser interrompidos por técnicas de *jamming*, que consiste em obstruir o sinal provocando uma interferência deliberativa, impedindo o sucesso

da comunicação. Os sinais podem também serem localizados com o intuito de destruí-los completamente [CES (2003)], [RUSSEL (1997)].

Por outro lado, para explorar o sinal, é necessário utilizar outras técnicas. Uma dessas técnicas é o *spoofing*, que consiste na imitação dos sinais de rádio, com o intuito de se transmitir uma informação falsa. Em sistemas mais complexos, com uma maior segurança, a técnica de *RDF* (*Radio Direction Finding*) ou monitoração de posição pode ser aplicada para encontrar fontes de interferência em qualquer forma de transmissão eletrônica *wireless*, podendo ser utilizada para encontrar a localização do rádio transmissor [RUSSEL (1997)]. A figura 2.9 mostra as principais técnicas *ECM*.

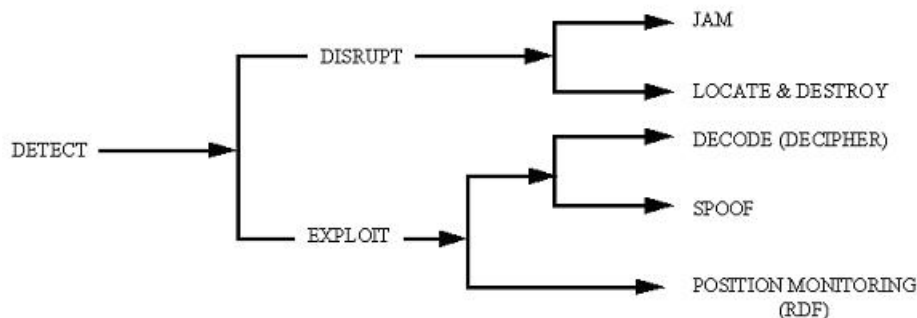


Figura 2.9: Técnicas *ECM* - *Electronic CounterMeasuring* - Fonte: *Wireless Channel Security Tutorial* [RUSSEL (1997)]

Para tentar evitar esses tipos de problemas, algumas técnicas podem ser empregadas, como por exemplo o *scrambling*, que é empregada em sistemas que utilizam transmissões *FHSS* (*Frequency Hopping Spread Spectrum*), já mencionado na seção 2.3, que consiste na manipulação do sinal da fala, geralmente provocando a sua inversão em torno de uma frequência particular, com o intuito de que a mensagem se torne inteligível para o inimigo [CES (2003)].

2.5 Segurança em Redes *Wireless*

Nos dias de hoje, com a ampla popularização da *Internet*, indivíduos normais e grandes organizações têm se preocupado muito com a segurança das suas informações dispostas pela rede.

Com a ampla comercialização de dispositivos móveis e a grande utilização da comunicação *wireless* nos mais diversos segmentos, a preocupação é ainda maior, uma vez que mensagens transmitidas via wireless, pelo ar, são mais fáceis de interceptar do que mensagens transmitidas por meio de cabos. A tecnologia *wireless* está avançando mais rápido do que questões de segurança relacionados à mesma, resultando em potenciais ataques para *hackers* [DEITEL (2002)].

O nível mais alto da rede de informação *wireless* é mostrado na figura 2.10. A rede pública de informação (rede telefônica e *Internet*) e redes privadas como de universidades, são usualmente não seguras, enquanto redes privadas como de indústrias, provedor de serviços *wireless* e redes locais privadas são usualmente seguras. A figura mostra também, a implantação de mecanismos de segurança como *firewalls*, nessas redes consideradas seguras [RUSSEL (1997)]. O mecanismo de segurança denominado *firewall* será explicado na seção 2.5.

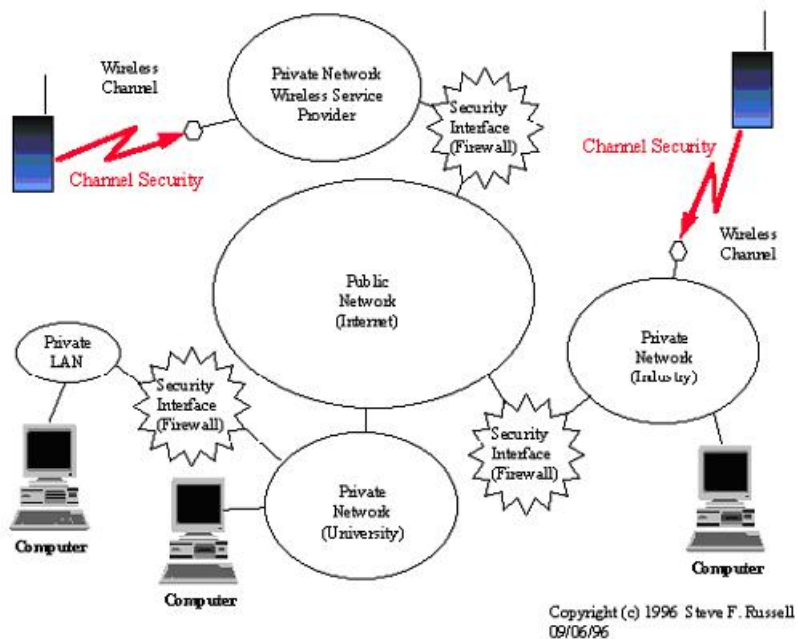


Figura 2.10: Segurança em vários tipos de redes de informação - Fonte: *Wireless Channel Security Tutorial* [RUSSEL (1997)]

Como já mencionado, a tecnologia implementada na comunicação *wireless*, mostrada em vermelho na figura 2.10, não é segura, exceto quando existe utilização de métodos de criptografia de dados, autenticação, e implementações *spread spectrum*, que fornecem segurança a tentativas elementares de *jamming*, *spoofing* e interceptação, definidos na seção 2.4 [RUSSEL (1997)].

Os principais problemas relacionados com a segurança em ambientes *wireless*, bem como seus correspondentes meios de combate, são agrupados em categorias, como mostra a tabela 2.1 [RUSSEL (1997)].

| ECM | Utilização | ECCM |
|------------------------|---|----------------------------|
| Detecção | Determinação de presença e atividade do sinal | Anti-Intercepção |
| Localização | Monitoramento e rastreamento de posições | Anti-Intercepção |
| Danificação de Serviço | Rompimento ou danificação do sinal | Anti-Jam |
| Falsificação | Roubo de serviços | Autenticação Criptografada |
| Decodificação | Obtenção de informações | Criptografia de Dados |
| Spoofing | Fornecimento de informação falsa | Segurança contra Spoofing |

Tabela 2.1: Elementos de uma comunicação segura - Fonte: *Wireless Channel Security Tutorial* [RUSSEL (1997)]

Em redes de comunicação em geral, existem quatro princípios básicos a se analisar para garantir a segurança do sistema [DEITEL (2002)]:

- Privacidade: Como ter certeza de que a informação transmitida ao longo da rede não foi capturada ou passada por um terceiro ponto desconhecido?
- Integridade: Como ter certeza de que a informação transmitida ou recebida não foi alterada?
- Autenticação: Como o emissor e o receptor da mensagem provam suas identidades para cada um?
- Não-repudição: Como provar legalmente que a mensagem foi enviada ou recebida?

Baseando-se nesses princípios, várias maneiras de se prover segurança em comunicações em rede têm sido criadas e utilizadas.

Uma das ferramentas mais básicas de segurança conhecidas é o *firewall*¹. Segundo Spangler [SPANGLER (2000)], utilizar um *firewall*, é provavelmente a maneira mais eficiente e fácil de se garantir segurança a uma pequena rede.

Outro meio bastante utilizado para garantir segurança na rede, é a utilização de sistemas de detecção de intrusão, que trabalham procurando detectar um invasor que conseguiu passar pela segurança do *firewall*, registrando *logs* dessas invasões [DEITEL (2002)].

Atualmente, a utilização de métodos de criptografia é considerada um dos meios mais seguros para garantir segurança na troca de mensagens, além da utilização de protocolos de segurança baseados nesses métodos.

A criptografia transforma a mensagem utilizando uma cifra ou sistema criptográfico, que encriptam (codificam) a mensagem de modo que a mesma se torne incompreensível. Desta maneira, existe uma chave (*key*), que funciona como uma senha para a cifra, que torna a mensagem incompreensível para todos menos para o emissor e receptor daquela mensagem, que terão acesso a essa chave e poderão decifrar (decodificar) aquela mensagem [DEITEL (2002)].

Para garantir uma maior segurança na execução de transações na rede, existem os protocolos de segurança em transações *wireless*, que fornecem segurança em transações para dispositivos móveis em rede [DEITEL (2002)].

Nas próximas seções serão apresentados os métodos de criptografia por chave privada ou simétrica, criptografia por chave pública, assinatura digital e os protocolos de segurança em transações *wireless* *WTLS* (*Wireless Transport Security Layer*) e *WEP* (*Wired Equivalet Privacy*).

2.5.1 Criptografia por Chave Privada

A criptografia por chave privada² ou simétrica é uma técnica de segurança que funciona da seguinte maneira: o transmissor encripta a mensagem com uma chave simétrica e envia a mensagem criptografada para o receptor, que a decodifica utilizando a mesma chave simétrica [DEITEL (2002)]. Esse esquema é mostrado na figura 2.11.

¹*Firewall* é um sistema colocado entre uma rede privativa e a *Internet* para controlar a entrada e saída de dados, e tem como objetivo básico proteger essa rede privativa (*LAN*) contra intrusos fora desta rede

²Chave privada ou simétrica é uma chave à qual têm acesso somente os participantes (transmissor/receptor) da comunicação, não sendo liberada aos demais usuários da rede

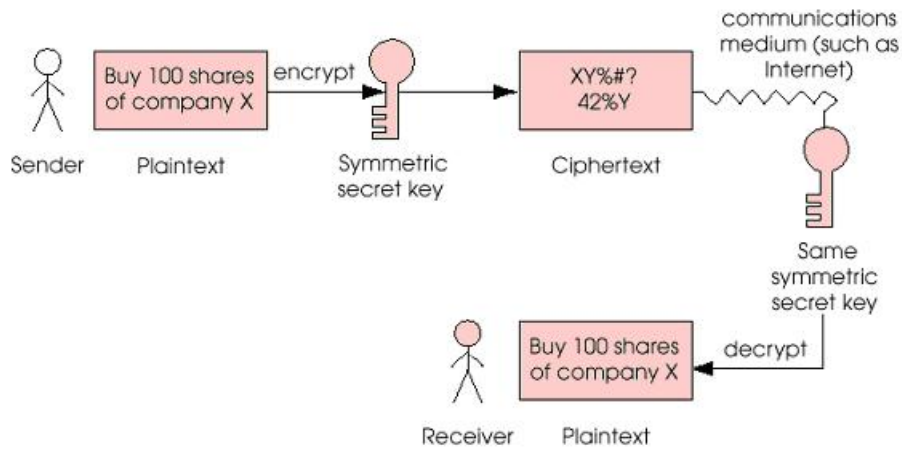


Figura 2.11: Encriptando e Decriptando mensagens utilizando o método de criptografia por chave privada - Fonte: *Internet and Mobile Business: How To Program, Cap 6 Security* [DEITEL (2002)]

É importante destacar que esse método não garante a privacidade e integridade da mensagem se a mesma for interceptada. Além disso como o emissor e receptor utilizam a mesma chave, não existe meio para autenticação. Um outro ponto é que para cada par transmissor-receptor uma nova chave precisa ser criada, tornando inviável a utilização desse método por causa da manutenção de um grande número de chaves simétricas. Uma tentativa para resolver esse problema consiste na criação de um centro de distribuição de chaves (*KDC - Key Distribution Center*), que cria uma chave secreta para uma sessão entre transmissor e receptor. Depois, ele distribui essa chave para o emissor e receptor em questão, encriptada com a chave privada simétrica compartilhada por cada um deles com o *KDC* [DEITEL (2002)]. A figura 2.12 mostra esse esquema.

Como mencionado por Deitel [DEITEL (2002)], os dois algoritmos principais que utilizam criptografia por chave privada é o *DES (Data Encryption Standard)* e o *AES (Advanced Encryption Standard)*.

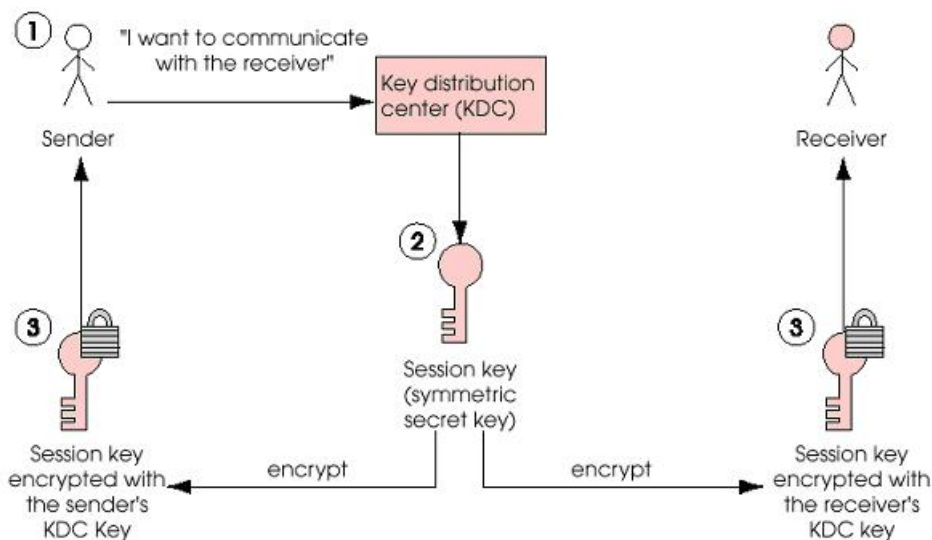


Figura 2.12: Esquema de distribuição de chaves utilizando criptografia por chave privada - Fonte: *Internet and Mobile Business: How To Program, Cap 6 Security* [DEITEL (2002)]

2.5.2 Criptografia por Chave Pública

Segundo Deitel [DEITEL (2002)], um outro método de criptografia é o chamado criptografia por chave pública, que não utiliza chaves simétricas, mas sim dois tipos de chave: uma pública e uma privada relacionadas para cada um dos participantes (transmissor e receptor).

A chave privada é mantida secreta pelo seu dono, enquanto a chave pública, todos da rede têm acesso à mesma. Se a chave pública é utilizada para encriptar a mensagem, somente a sua chave privada correspondente pode funcionar para decriptá-la.

Para transmitir a mensagem de forma segura, o transmissor utiliza a chave pública do receptor, a qual tem acesso, para encriptar a mensagem. O receptor então decripta a mensagem com sua chave privada única [DEITEL (2002)].

Segundo Deitel [DEITEL (2002)], esse sistema garante a privacidade da mensagem, uma vez que não é possível se deduzir uma chave privada a partir de uma chave pública correspondente, impedindo que outro ponto tenha acesso a mensagem encriptada por esse esquema, ilustrado na figura 2.13.

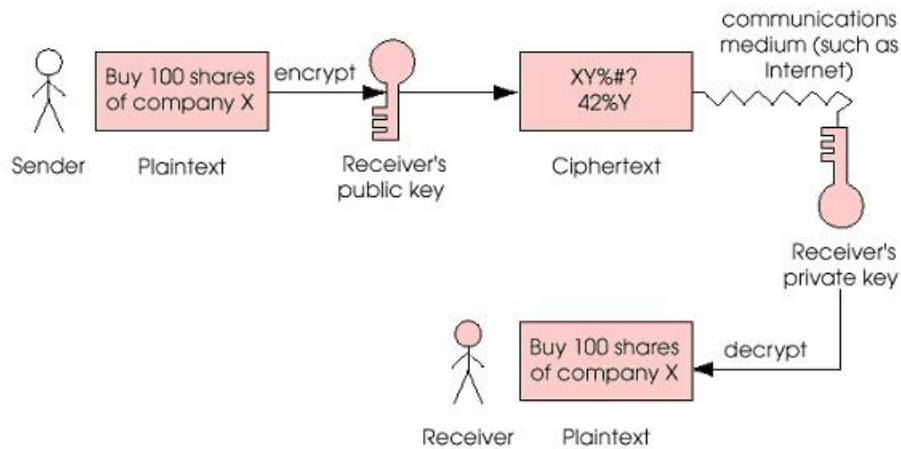


Figura 2.13: Encriptando e Decriptando uma mensagem utilizando o método de criptografia por chave pública - Fonte: *Internet and Mobile Business: How To Program, Cap 6 Security* [DEITEL (2002)]

Esse esquema garante ainda a autenticação tanto do receptor, quando a chave utilizada para decriptar a mensagem é a chave privada do mesmo e para encriptar é a chave pública do mesmo, como do transmissor, quando a chave utilizada para decriptar a mensagem é a chave pública do mesmo e a chave utilizada para encriptar é a chave privada do mesmo. A utilização dos dois métodos em conjunto garante total autenticação, esquema ilustrado na figura 2.14 [DEITEL (2002)].

Como mencionado em DEITEL (2002), os dois algoritmos baseados em criptografia por chave pública, mais comuns são: o *RSA*, que é um algoritmo criado em 1977 por professores do *MIT (Massachusetts Institute Technology)* e o *PGP (Pretty Good Privacy)*, utilizado na criptografia de arquivos e mensagens de email.

2.5.3 Assinatura Digital

Assinatura Digital é outro método utilizado para auxiliar no processo de criptografia por chave pública, resolvendo problemas de autenticação e integridade [DEITEL (2002)].

Primeiramente, o transmissor submete a mensagem a uma função chamada *hash*, que gera um código único correspondente àquela mensagem (*message digest*).

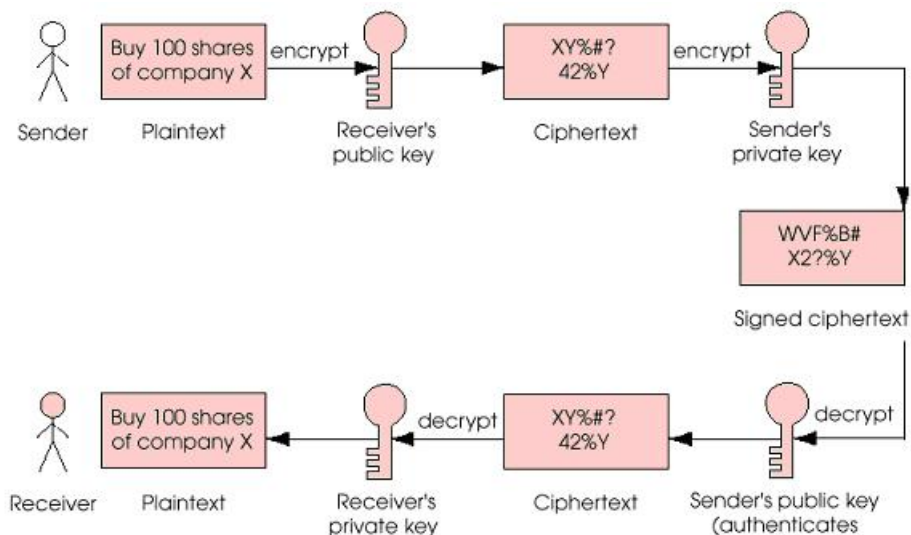


Figura 2.14: Autenticação do Transmissor e Receptor utilizando o método de criptografia por chave pública - Fonte: *Internet and Mobile Business: How To Program, Cap 6 Security* [DEITEL (2002)]

Depois disso, o transmissor utiliza a sua chave privada para encriptar o código gerado (*message digest*). A partir daí está criada a assinatura digital, autenticando o transmissor. A mensagem original, encriptada pela chave pública do receptor, a assinatura digital e a função *hash* são enviadas para o receptor dentro de um envelope digital.

Ao receber o envelope, o receptor utiliza a chave pública do emissor para decifrar a assinatura original, revelando o *message digest*. O receptor utiliza então sua chave privada para decifrar a mensagem original. Finalmente, o receptor aplica a função *hash* à mensagem original. Se o valor for igual ao do *message digest* existe uma integridade de mensagem, indicando que a mesma não foi alterada [DEITEL (2002)].

Para Deitel [DEITEL (2002)], assinaturas digitais, diferentes das assinaturas manuscritas tradicionais, não são únicas, e a cada novo documento gerado, uma nova assinatura é criada, com base neste documento, e atribuída ao mesmo, dificultando ainda mais a sua decodificação por um invasor.

2.5.4 O protocolo WTLS

Wireless Transport Layer Security é um protocolo que garante segurança na conexão entre dispositivos *wireless* e servidores de aplicação. Ele provém uma tecnologia *wireless* com integridade dos dados, privacidade, autenticação e segurança contra *denial-of-service*³ [WAP (2000)].

O protocolo WTLS encripta os dados que são enviados entre o dispositivo móvel com o padrão de comunicação WAP (*Wireless Application Protocol*), e um *gateway* WAP. No *gateway*, as transações são decriptadas do protocolo WTLS e encriptadas em um outro protocolo de segurança chamado SSL (*Secure Socket Layer*) [DEITEL (2002)].

Dispositivos WAP que utilizam o protocolo WTLS só possuem segurança em suas transações antes de chegar no *gateway*, momento em que as mesmas são colocadas na rede cabeada. Apesar disso, ainda não foram encontrados problemas efetivos de ataque devido a esse momento de insegurança, também chamado de *WAP gap*, apresentado no protocolo WTLS [DEITEL (2002)].

Para contornar esse problema, existe uma técnica chamada *WTLS tunneling*, que consiste na existência de um túnel no *gateway*, permitindo que os dados criptografados passem do dispositivo *wireless* através do *gateway*, até chegar no servidor sem serem decriptados. Ao passar do protocolo WTLS para o SSL, o servidor envia uma chave WTLS de 128 *bits* para o dispositivo móvel, encriptada com a chave pública desse dispositivo. A chave WTLS é então utilizada para encriptar dados das transações em todo o resto do processo [DEITEL (2002)].

2.5.5 O Protocolo WEP - A segurança das WLANS do padrão 80211

Wired Equivalent Privacy (WEP) é o protocolo de segurança das redes locais (LANs) *wireless*, também conhecidas como *Wi-Fi*, pertencentes ao padrão IEEE 80211 [DEITEL (2002)].

Segundo Deitel [DEITEL (2002)], os serviços oferecidos pelo protocolo WEP oferecem às redes *Wi-Fi*, procedimentos de checagem de integridade e criptografia dos dados.

Em comunicações seguras via protocolo WEP, o transmissor primeiro aplica uma checagem de integridade à mensagem, garantindo que a mesma não foi alterada durante a transmissão. O resultado dessa checagem é anexado à mensagem. A partir deste ponto, todo o pacote é criptografado com esquema criptográfico de

³*Denial-of-service* são ataques que visam impedir que usuários legítimos de um determinado serviço na rede possam usufruir do mesmo

40 *bits* (inseguro) [NELSON (2001)]. Depois disso, o pacote é enviado ao receptor por meio de um sistema *wireless*.

Segundo Deitel [DEITEL (2002)], por causa da insegurança no processo de transmissão, agravado ainda mais pelo compartilhamento de chaves simétricas na rede, muitas melhoras estão sendo feitas em cima do protocolo *WEP*. O protocolo *WEP2*, sucessor do protocolo *WEP*, possui um esquema criptográfico de 128 *bits* e um vetor de inicialização de chaves diferentes para cada sessão, também em um esquema de 128 *bits*, garantindo uma maior variabilidade e maior tamanho das chaves geradas.

Segundo Garcia [GARCIA (2001)], visando uma maior segurança das redes de seu padrão, a *IEEE* pretende utilizar um sistema de gerenciamento de chaves chamado *Enhanced Security Network (ESN)*, que implementa o *128-bit Advanced Encryption Standard (AES)*, um esquema de criptografia que promove uma maior segurança, sem a utilização de muita memória ou poder de processamento. Adicionalmente, serão utilizados os métodos de *Kerberos*, um forte sistema de autenticação na rede, que lidará com os assuntos de autenticação e acesso à rede.

2.6 Linguagens de Programação para PDAs

Um *PDA (Personal Digital Assistant)* pode ser visto como um computador de mão, com diversas funcionalidades equivalentes às de um computador *desktop*, com a vantagem adicional de possibilitar mobilidade ao usuário. Está inserido na categoria dos dispositivos móveis ou *handhelds* que engloba ainda telefones celulares e *palms*.

Apesar do desenvolvimento de sistemas para dispositivos *PDAs* ser um campo recente, até mesmo pelo fato de estes dispositivos serem recentes, muitas linguagens e ferramentas de programação já estão disponíveis para realizar este trabalho. Esta seção irá apresentar as linguagens mais conhecidas, descrevendo as suas principais características, sistemas operacionais para os quais elas se aplicam, vantagens e desvantagens.

Atualmente, os principais sistemas operacionais existentes para *PDAs* são o *PalmOS* e o *Windows CE* (também distribuído na versão *Pocket PC*). Assim, as ferramentas apresentadas nesta seção são quase todas direcionadas para o desenvolvimento nestas plataformas. Além destes, existem o sistema operacional *Newton* e algumas variações do sistema operacional *Linux*.

Antes de iniciar a apresentação das linguagens e ferramentas mais conhecidas, dois conceitos básicos que ajudarão na avaliação das ferramentas disponíveis,

devem ser apresentados. Estes conceitos são:

- **PDB**: É o banco de dados nativo do sistema operacional *PalmOS*. Também pode ser utilizado em outros sistemas operacionais, uma vez que seu formato é conhecido. A linguagem de programação deve ser capaz de prover acesso aos dados deste banco de dados de maneira fácil e eficiente [BERNSTEIN (2001)].
- **PRC**: *PRC* é a extensão do aplicativo desenvolvido para rodar no sistema operacional *PalmOS*. Em linguagem simplificada, é equivalente ao *EXE* tão famoso no ambiente *Windows* [BERNSTEIN (2001)]. No caso do sistema operacional *Windows CE*, também gera-se arquivos *EXE*, mas estes são diferentes dos gerados para os sistemas operacionais *Windows* versões *desktop*.

Algumas ferramentas simplesmente geram um *PRC* ou um *EXE*, após o processo de compilação, que pode ser instalado diretamente no dispositivo. Outras trabalham com *runtime*, ou seja, em modo interpretado. Geralmente o *runtime* é instalado no aparelho durante o processo de instalação da própria ferramenta. Neste caso, ao se desenvolver aplicações na ferramenta estas não irão funcionar diretamente no aparelho. Para que elas funcionem devem ser executadas sobre o interpretador ou o *runtime*, pois os arquivos gerados pelas ferramentas só são reconhecidos por eles. Ao ser executado, o runtime aciona o aplicativo, interpretando os comandos e executando-o [BERNSTEIN (2001)].

Assim, pode-se começar a avaliar as ferramentas existentes pela eficiência que elas provêm no acesso a banco de dados e também com relação ao formato de arquivos que elas geram, ou seja, arquivos nativos para uma plataforma específica ou arquivos em um formato intermediário que são reconhecidos por um ambiente de *runtime*.

De acordo com Bernstein [BERNSTEIN (2001)], pode-se separar as ferramentas de desenvolvimento em categorias, onde os principais termos de comparação são a flexibilidade contra a facilidade e rapidez de desenvolvimento:

- **NSBasic** [NSBASIC (2003)]

Descrição: De acordo com Alexandroni [ALEXANDRONI (2001a)], o *NSBasic* é uma ferramenta simples e poderosa para o desenvolvimento de aplicações para os *PDAs*. Apesar de ser uma ferramenta baseada na linguagem *Basic*, ela não é obsoleta ou indicada a novatos. Além de muitas funções contidas na linguagem, o *NSBasic* permite a utilização de *Shared Libraries*

escritas em *C*, acesso às rotinas da *API* do *PalmOS* e também que se adicione novas funções à linguagem, inclusive escritas em *C*.

Sistema Operacional: *PalmOS*, *Windows CE* e *Apple Newton*.

Custo: Uma versão demo do produto pode ser adquirida com validade por 30 dias. A versão registrada custa 99.95 dólares.

Características: As aplicações podem ser distribuídas para executar em um *runtime* de aproximadamente 85K, mas estas têm uma performance muito boa e não há custo para distribuição deste *runtime*. É possível também gerar executáveis nativos, onde o *runtime* é adicionado ao executável. Isto é ideal para o caso de se estiver distribuindo uma aplicação comercialmente, mas quando se estiver rodando duas ou mais aplicações desenvolvidas com o *NSBasic* é melhor instalar o *runtime* para que ele seja compartilhado pelas duas aplicações. Outra característica importante é que o desenvolvimento de programas no *NSBasic* é baseado em um ambiente visual.

- ***PDA ToolBox*** [PDA (2003)]

Descrição: O *PDA Toolbox* proporciona uma flexibilidade muito pequena, mas tem o poder de desenvolver aplicativos simples e funcionais.

Sistema Operacional: *PalmOS*.

Custo: É gratuito, mas possui uma versão paga com maiores recursos.

Características: As aplicações desenvolvidas apresentam executáveis extremamente pequenos e rápido. O desenvolvimento é realizado em um ambiente visual, ou seja, não é necessário escrever nenhuma linha de código para obter bons resultados.

- ***PocketStudio*** [POCKET (2003)]

Descrição: Esta ferramenta estava sendo desenvolvida desde 1999 e em 2001 foi finalmente lançada. Tornou-se muito aguardada pelos desenvolvedores para *PalmOS* e também muito utilizada por eles. É semelhante ao ambiente de desenvolvimento da linguagem *Delphi*, compartilhando assim semelhanças também com a linguagem *Pascal*. Construída pela *Pocket-Technologies*, uma empresa localizada no Estado de *Minnesota* nos Estados Unidos e formada por desenvolvedores experientes e muito profissionais, o *PocketStudio* é muito mais que uma simples *IDE*, conta com um compilador poderosíssimo e rápido, que gera aplicações nativas *PalmOS* com velocidade e tamanho comparáveis às geradas por compiladores *C* e sem a necessidade

de *runtime*. Possui suporte aos mais diversos periféricos e equipamentos como impressoras, equipamentos *Symbol* com leitor de códigos de barras e celulares.

Sistema Operacional: *PalmOS*.

Custo: A versão mais completa é vendida por 249.99 dólares, enquanto a mais simples pode ser comprada por 69.99 dólares.

Características: É uma ferramenta poderosa, produtiva e principalmente fácil de utilizar. De acordo com Alexandroni [ALEXANDRONI (2003c)], suas principais características são:

1. Acesso a toda *API* do Sistema Operacional *PalmOS*.
2. Gera executáveis nativos *PalmOS* sem necessidade de *runtime*.
3. Permite instruções *assembler* no meio do código (*inline assembler*).
4. Suporte a *PalmOS* 2.0 a 5.0.
5. Suporte a *HandEra*, *Kyocera*, *HandSpring*, *Symbol*, *Sony*, etc.
6. Acesso Bancos de Dados móveis como o *Oracle Lite* e *IBM DB2*.
7. Suporte a impressoras de qualquer tipo através do acesso aos gerenciadores de impressão mais poderosos do mercado, como o *IRPrint*, *PrintBoy* e *PalmPrint*.
8. *PSLibrary*, interface de acesso à objetos visuais do formulário e também à bancos de dados.
9. *Database Wizard*, cria uma aplicação básica de entrada de dados em segundos, com criação visual do banco de dados.
10. Aceita *plugins* no *IDE* para expansão das características da ferramenta.
11. *Conduit Wizard*, auxilia na criação dos *conduits* para seus bancos de dados. *Conduits* são aplicações que permitem a sincronização de dados dos sistemas *PalmOS* com o computador *desktop*.
12. *Web Update*, atualização da ferramenta diretamente pela *Internet*.
13. Integração com o emulador do *PalmOS*, inclusive para depuração da aplicação.
14. Ambiente similar ao do *Delphi*.
15. Depuração passo-a-passo.
16. Editor de texto que destaca a sintaxe da linguagem.

- ***Satellite Forms MobileApp Designer***[SATELLITE (2003)]

Descrição: Os aplicativos desenvolvidos por esta ferramenta criada pela *Pu-matech* são baseados em *runtime*. Para o desenvolvimento de aplicações para o mercado corporativo, o *Satellite Forms* tem ganhado muito destaque. Este destaque ocorreu devido ao fato de ele ser uma ferramenta de desenvolvimento rápido (*RAD*), com várias extensões que podem ser adquiridas para estender a sua funcionalidade e com suporte aos mais famosos sistemas operacionais para *PDA*s, ou seja, *PalmOS* e *Windows CE*. Os códigos possuem semelhança com o *Visual Basic*.

Sistema operacional: *PalmOS* e *Pocket PC 2002*

Custo: A versão completa pode ser adquirida por 1450 dólares.

Características: De acordo com a empresa Satellite [SATELLITE (2003)], o *Satellite Forms* é uma ferramenta visual, com recursos *drag-and-drop*, de utilização muito fácil e curva de aprendizagem rápida. Possui bibliotecas para a criação de interfaces avançadas.

- **A linguagem C, o *CodeWarrior*, o *PRC Tools***

Descrição: A linguagem C, apesar de parecer complicada e assustadora para muitos, na verdade tem as mesmas estruturas de controle de qualquer outra linguagem de programação. O que a diferencia das outras, e talvez o motivo da sua fama de linguagem de baixo nível, é o uso de ponteiros e algumas estruturas mais complexas. Com ela pode-se ter acesso a todo o poder do sistema operacional *PalmOS* (através do acesso a *API* completa), executáveis menores e mais rápidos, sem *runtimes* e pagamentos de licenças [ALEXANDRONI (2001b)].

De acordo com Alexandroni [ALEXANDRONI (2001b)], para desenvolver em C, existem alguns caminhos que podem ser seguidos: utilizar uma ferramenta comercial, o *CodeWarrior* [CODEWARRIOR (2003)], que tem compilador próprio, ou o *PRC Tools* com a *IDE Falch.net* [FALCH (2003)].

Sistema operacional: *PalmOS*.

Custo: O *CodeWarrior* custa 499 dólares, enquanto o *PRC Tools* é de livre distribuição, mas a *IDE Falch.net* custa 249 dólares.

Características: O *CodeWarrior* é o ambiente de desenvolvimento oficial da *Palm* e é o mais famoso ambiente integrado (*IDE*) para a linguagem C.

O *PRC Tools*, integra várias ferramentas como o *GCC* (compilador), o *GDB* (depurador), *PilRC* (compilador de recursos), entre outros. Porém, ele não disponibilizava uma maneira fácil de utilizar estas ferramentas, era tudo realizado manualmente. Assim, surgiram algumas *IDEs* como o *VFDIDE*, *PilotMag* e o *Falch.net*.

- ***Embedded Visual Tools***

Descrição: Conjunto de ferramentas distribuídas gratuitamente pela *Microsoft* para o desenvolvimento de aplicações completas para dispositivos baseados em *Windows CE*.

Sistema operacional: *Windows CE* e variações.

Custo: Gratuito.

Características: O *Embedded Visual Tools* provê duas ferramentas para o desenvolvimento das aplicações: o *Embedded Visual C++*, baseado no *Visual C++* para *desktops*, e o *Embedded Visual Basic*, baseado no *Visual Basic* para *desktops*. Além destas, também são disponibilizados emuladores para o teste de aplicações.

- **Tecnologias baseadas em *Java***

Descrição: Para o desenvolvimento utilizando as ferramentas da *Sun Microsystems* (criadora do *Java*), temos o *Personal Java* para *Windows CE* e o *J2ME MIDP/CLDC* para *PalmOS*. Outra tecnologia baseada em *Java*, mas não relacionada com a *Sun*, é o *SuperWaba*, que pode ser utilizado tanto no *Windows CE*, quanto no *PalmOS*.

As características e vantagens das tecnologias baseadas em *Java* serão descritas com maiores detalhes nas próximas seções.

Além das ferramentas e linguagens de programação citadas anteriormente, existem várias outras menos famosas e que não chegaram a ser avaliadas profundamente. Para o desenvolvimento em *PalmOS*, por exemplo, a empresa *Palm* [PALMOS (2003)] apresenta todas as ferramentas existentes. Para o desenvolvimento em dispositivos baseados em *Windows CE* e suas variações, as principais ferramentas já foram citadas, sendo a principal delas o *Embedded Visual Tools*.

As próximas seções discutem as tecnologias realmente utilizadas no projeto, começando pela tecnologia *Java*, utilizada no desenvolvimento de quase todos os sistemas criados.

2.7 A Tecnologia *Java*

Esta seção apresenta uma das mais importantes tecnologias escolhidas para se utilizar neste projeto, a tecnologia *Java* da *Sun Microsystems*

A tecnologia *Java* surgiu como um produto indireto de um pequeno projeto, conhecido por *Green Project*, iniciado por *Patrick Naughton*, *Mike Sheridan* e *James Gosling* da *Sun Microsystems* em 1991 [BYOUS (2002)],[SUN (2003b)].

É curioso o fato de as pessoas se referirem à tecnologia *Java*, como se a mesma se tratasse apenas de uma linguagem de programação. Neste sentido, é importante destacar que *Java* não se trata apenas de uma linguagem de programação, mas também de uma plataforma poderosa de desenvolvimento. Essa característica será evidenciada ao longo das próximas seções.

O objetivo desta seção é apresentar os conceitos básicos da tecnologia *Java*, os quais serão necessários para a compreensão dos sistemas desenvolvidos no projeto. A tecnologia *SuperWaba* será apresentada na seção 2.9.

2.7.1 A Linguagem de Programação *Java*

A linguagem de programação *Java* é uma linguagem de alto nível que pode ser caracterizada por ser uma linguagem: simples, segura, dinâmica, robusta, portátil, orientada a objetos e de alta performance [GOSLING (1996)].

Para executar os programas desenvolvidos em *Java*, o processo é feito em duas etapas:

- Na primeira etapa, o programa é compilado, traduzindo-se o código fonte em uma linguagem intermediária (*bytecodes*).
- Na segunda etapa, o *bytecodes*, que é independente da máquina em que foi gerado, é passado para o interpretador da máquina

Esse processo é ilustrado na figura 2.15.

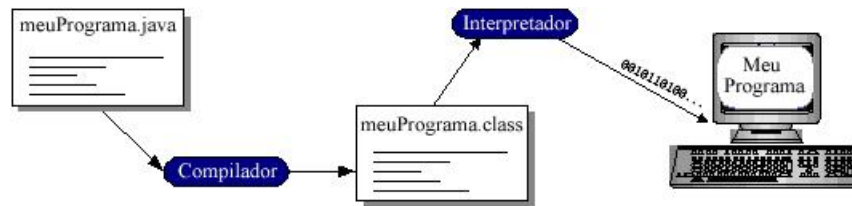


Figura 2.15: Processo de compilação e interpretação de um programa desenvolvido na linguagem de programação *Java* - Fonte: *Avaliação da tecnologia J2ME no contexto de desenvolvimento de jogos multiplayer para celulares* [ASSIS (2003)]

Pode-se perceber que a alta portabilidade da linguagem *Java* se deve a esse processo de compilação e interpretação dos programas. Porém, este processo tem também uma desvantagem que é uma certa lentidão quando comparado com linguagens de programação que possuem compiladores que geram códigos nativos, isto é, códigos para uma plataforma específica. Porém, isso tem sido melhorado com a geração de interpretadores através de tecnologias *just-in-time compilers*⁴, que melhoram sua performance, sem perder a portabilidade, que é uma de suas características principais, como mostra a figura 2.16.

⁴*just-in-time compilers* são tecnologias empregadas durante a implementação de interpretadores que visam melhorar a performance dos mesmos. No caso da plataforma *Java*, essas técnicas são capazes de traduzir em tempo de execução o *bytecode Java*, para código de máquina de uma determinada CPU



Figura 2.16: Portabilidade apresentada pela linguagem *Java* - Fonte: *Avaliação da tecnologia J2ME no contexto de desenvolvimento de jogos multiplayer para celulares* [ASSIS (2003)]

2.7.2 A Plataforma de Desenvolvimento *Java*

Como mencionado no início desta seção, a tecnologia *Java*, mais do que uma linguagem de programação, se trata de uma grande plataforma de desenvolvimento.

A plataforma *Java*, diferente das plataformas comuns dispostas em *hardware* e *software*, é uma plataforma unicamente de *software*, que inclui duas partes principais [CAMPIONE (2002)]:

- *Java Virtual Machine (JVM)* : responsável pela portabilidade da tecnologia, é ela quem realiza a tradução do código criado pelo programador num código compatível com a arquitetura em que o programa será executado. Tem funções de gerência de memória, segurança, alternância de várias linhas de processamento em execução, dentre outras.
- *Java API (Java Application Programming Interface)*: grande coleção de componentes de *software*, que provê várias funcionalidades ao programador como interface gráfica, criptografia, conexões, etc.

Dentre as várias versões desenvolvidas desta plataforma, destaca-se a versão *Java 2*, a partir da qual se criou uma divisão da plataforma em três diferentes edições:

- *Java 2 Platform, Standard Edition (J2SE)* [J2SE (2002)], voltada para o desenvolvimento de aplicações para computadores *desktop* convencionais;

- *Java 2 Platform, Enterprise Edition (J2EE)* [J2EE (2002)], com ênfase no desenvolvimento de aplicações servidoras para empresas com necessidade de servir seus consumidores, fornecedores e empregados com soluções sólidas e completas de negócios. Ela é um superconjunto da *J2SE* e adiciona *APIs* para a computação do lado do servidor.
- *Java 2 Platform, Micro Edition (J2ME)* [J2ME (2002)], é um conjunto de tecnologias e especificações desenvolvidas para pequenos dispositivos como *smart cards*, *paggers* e telefones celulares. Ela usa um subconjunto dos componentes *J2SE*, tais como subconjuntos das máquinas virtuais e *APIs* mais fracas. Representa um retorno às origens do *Java*.

A Figura 2.17 ilustra a relação entre essas edições bem como o alvo de aplicação para cada uma delas.

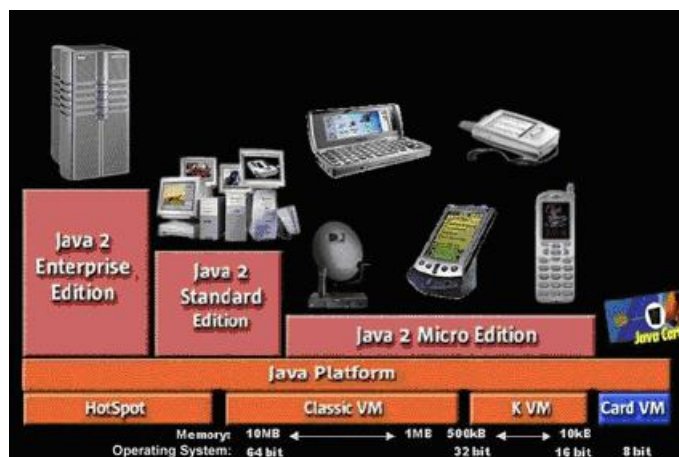


Figura 2.17: Versões da linguagem *Java* e seus respectivos alvos de aplicação - Fonte: *Avaliação da tecnologia J2ME no contexto de desenvolvimento de jogos multiplayer para celulares* [ASSIS (2003)]

2.8 A Tecnologia Waba

O objetivo desta seção é apresentar a tecnologia *Waba*, que tem como principal variação a tecnologia *SuperWaba* quer será analisada na próxima seção.

Esta seção proporciona um conhecimento mais profundo da tecnologia *Waba*, definindo suas principais características, vantagens e desvantagens.

2.8.1 Definição da tecnologia *Waba*

A tecnologia *Waba* [WABA (2003a)], desenvolvida inicialmente por *Rick Wild* da empresa *Wabasoft*, é voltada para o desenvolvimento de sistemas para pequenos dispositivos. Ela é constituída de uma linguagem de programação e uma plataforma de programação, baseadas na linguagem e plataforma definidas pela tecnologia *Java*. Assim, o formato de escrita de códigos é semelhante ao de *Java*, sendo um subconjunto desta linguagem, e também se utiliza a idéia de máquina virtual para ganhar portabilidade entre os mais diversos tipos de *PDA*s. O formato dos *bytecodes Waba* também está relacionado com o formato utilizado pelo *Java*. *Waba* é freqüentemente comparada à plataforma *J2ME* por possuírem quase o mesmo escopo de campo de atuação.

Por causa da maneira como o *Waba* foi projetado, desenvolvedores podem usar ferramentas de desenvolvimento *Java* para desenvolver programas *Waba*. Entretanto, *Waba* não é derivado de *Java* e não possui nenhuma ligação com a *Sun Microsystems*, proprietária da marca *Java*.

Assim como na plataforma *J2ME*, a linguagem de programação, a máquina virtual e a *API Waba* foram otimizadas para o uso em pequenos dispositivos. Algumas classes que foram incluídas na *API Waba* permitem que programas escritos nesta plataforma possam executar em qualquer máquina virtual *Java*, como *applets* ou aplicações normais. Com uma máquina virtual *Waba* nativa, o mesmo programa pode executar em um pequeno dispositivo, como um *PalmPilot* ou *PocketPC*.

2.8.2 Características do *Waba*

Esta subseção faz um resumo das principais características da tecnologia *Waba* e também apresenta as suas especificações.

As características principais do *Waba* são:

- Similaridade com a tecnologia *Java*: A estrutura da tecnologia *Waba* é similar a de *Java*, mas *Waba* não é *Java*. Cada uma possui uma máquina virtual, um conjunto de *APIs* e uma linguagem de programação próprios.
- Execução em máquinas virtuais *Java*: Um programa feito em *Waba* pode ser executado em qualquer máquina virtual *Java*, inclusive como *applets*. Isto é feito através de classes que servem como pontes entre as *APIs Java e Waba*. Esta característica permite o desenvolvimento de programas no *desktop* com posterior instalação no dispositivo móvel.

- Custo baixo: as máquinas virtuais *Waba* são gratuitas, bem como o *kit* de desenvolvimento de sistemas. Todo este *software* é gerenciado pela licença *GNU*, a mesma do sistema operacional *Linux*.
- Flexibilidade: como o código fonte da máquina virtual *Waba* é livre, pode-se adicionar qualquer característica a ela. Uma das possibilidades é a adição de suporte a funções nativas dos dispositivos desejados.

As especificações *Waba* são:

- *Waba Virtual Machine*.
- *Waba Class Library*.
- *Waba Software Development Kit*.

A *Waba Virtual Machine* (em português, Máquina Virtual *Waba*) ou simplesmente *WabaVM* foi projetada de maneira a executar um subconjunto das instruções *bytecodes* definidas pela especificação da máquina virtual *Java*, que está definida em [LINDHOLM (1999)], por Lindholm.

A *Waba Class Library* (em português, Biblioteca de Classes *Waba*) foi projetada de maneira a possibilitar uma programação simples, mas isso não significa que não seja possível construir programas avançados. Foram incluídas classes básicas responsáveis por *I/O*, rede, interface com o usuário e manipulação de gráficos.

O *Waba Software Development Kit* (em português, Conjunto de Ferramentas para o Desenvolvimento em *Waba*) ou simplesmente *WabaSDK*, consiste em uma documentação de referência, um conjunto de classes *Java* e algumas ferramentas que são utilizadas para a compilação do código criado.

As classes *Java* que vem em conjunto com o *WabaSDK* são chamadas de pontes pelo fato delas implementarem as classes da *API Waba* de modo que elas possam ser executadas em uma máquina virtual *Java*. A Figura 2.18 ilustra de uma maneira mais adequada esta relação.

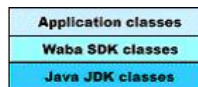


Figura 2.18: Executando uma aplicação *Waba* em máquinas virtuais *Java* - Fonte: *The WabaSoft Development Kit* [WABA (2003b)]

Assim, as classes que vêm em conjunto com a *WabaVM* e o *WabaSDK* são diferentes. As classes na *WabaVM* chamam métodos nativos do dispositivo, enquanto que as do *WabaSDK* são direcionadas para o desenvolvimento em computadores com máquinas virtuais *Java*. A figura 2.19 ilustra a estrutura de classes quando se está executando uma aplicação *Waba* na *WabaVM*.

As classes *Waba* nativas e as do *WabaSDK* possuem a mesma *API*. Assim, pode-se escrever um programa no *desktop* e executá-lo através do *WabaSDK* para, em seguida, colocá-lo em funcionamento em um dispositivo com a *WabaVM* instalada sem nenhuma modificação.

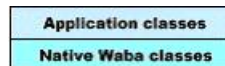


Figura 2.19: Executando uma aplicação *Waba* na *WabaVM* - Fonte: *The WabaSoft Development Kit* [WABA (2003b)]

2.8.3 Vantagens da tecnologia *Waba*

Segundo informações pesquisadas no site da própria empresa *Wabasoft*, alguns tópicos que podem levar a escolha do *Waba* como uma ferramenta para o desenvolvimento de aplicações para dispositivos móveis são:

- **Mobilidade:** *Waba* foi projetado para dispositivos pequenos, geralmente móveis. A máquina virtual *Waba* possui menos de 64KB de memória (incluindo as *APIs*) e executa programas em menos de 10KB de memória.
- **Funcionalidade:** *Waba* permite o desenvolvimento rápido de programas para qualquer plataforma através de ferramentas de desenvolvimento baratas, familiares e fáceis de usar. A linguagem *Waba* é orientada a objetos e inclui algumas características importantes, como o coletor de lixo automático.
- **Confiança:** como o *Waba* utiliza coletor de lixo automático, programas escritos nesta linguagem raramente ocasionam o travamento do dispositivo.
- **Portabilidade:** com o *Waba* pode-se escrever um programa e executar depois no *PalmPilot*, *Windows CE* ou qualquer outra plataforma que suporte *Java*.

2.9 A Tecnologia *SuperWaba*

A tecnologia *SuperWaba* é uma variação da tecnologia *Waba*. Esta seção será responsável por apresentar todas as suas características e um pouco de sua história.

2.9.1 Como surgiu o *SuperWaba*

No início de 2000 existiam duas linguagens, ambas em versão *beta*, para o desenvolvimento de aplicações em sistemas *PalmOS*. Estas linguagens eram o *KVM*, da *Sun Microsystems* e participante da plataforma *J2ME*, e o *Waba*. Nesta mesma época, um brasileiro chamado Guilherme Campos Hazan desejava desenvolver aplicações para seu *handheld*. Guilherme experimentou então o *KVM*, mas pouco tempo depois se deparou com o *Waba*, que segundo ele, era muito simples de trabalhar, afirmando que ao final de uma tarde, já tinha seu primeiro programa *Palm* rodando em seu *PalmProfessional*. [HAZAN (2002)]. Como Guilherme Hazan havia gostado do *Waba*, propôs a si mesmo o desenvolvimento de um programa financeiro utilizando-o. O resultado foi que não era possível realizar tal tarefa devido a simplicidade exagerada da tecnologia: Não havia, por exemplo, como inserir um registro no meio do banco de dados, mas somente no final [HAZAN (2002)]. Utilizando seus conhecimentos avançados de computação e aproveitando do fato de o código do *Waba* ser livre, Guilherme começou a adicionar algumas funções à versão original e, querendo dividir estas melhorias com outros usuários, criou o *SuperWaba*. Guilherme desejava, na verdade, que *Rick Wild*, criador do *Waba*, fundisse as duas versões, mas isso nunca ocorreu [HAZAN (2002)].

O *SuperWaba* começou a ser aprimorado para o *PalmOS*, pois esta era a plataforma que Guilherme desejava dominar. Segundo ele, até a versão 1.21, o *SuperWaba* era apenas uma versão melhorada do *Waba*, com uns poucos métodos nativos a mais e algumas classes, como a classe *Window* (no *Waba* original não haviam janelas *popup*). O *SuperWaba* foi portado para o *Windows CE* através da ajuda de uma pessoa ligada a comunidade criada por Guilherme [HAZAN (2002)].

Durante oito meses de trabalho, Guilherme foi capaz de incluir várias melhorias no *SuperWaba* com relação a versão original, algumas delas de grande importância [HAZAN (2002)]. Estas melhorias foram:

- Suporte a tons de 4 cores de cinza.
- Suporte a inúmeras funções gráficas que não existiam no *PalmOS* original.
- Adição de efeitos *3d* aos controles originais do *PalmOS*.

- Suporte a precisão de 64 *bits* (tipos *double* e *long*).
- Suporte a bibliotecas em *C* e em *Java* (até então, a única forma de se adicionar bibliotecas nativas era mudando a máquina virtual e, conseqüentemente, tornando-a incompatível com a versão oficial).
- Fontes customizáveis.
- Diterização automática de imagens coloridas.
- Suporte a *sprite* e detecção de colisão (para jogos).
- Janelas *popup* e arrastáveis.
- Suporte a *threads*.
- Velocidade de execução dobrada.
- Melhorias propostas e enviadas por membros da comunidade, como suporte a *USB* e *IR*, e aos modelos de *Scanner* da *Symbol Technologies*.

Em Outubro de 2001, Guilherme tomou a importante decisão de se dedicar exclusivamente ao *SuperWaba*, deixando o seu emprego atual na época. Isto provocou uma melhora incrível à tecnologia. Um novo gerenciador de memória com coletor de lixo disponibilizava aos programas toda a memória do *Palm* (Segundo Guilherme Hazan [HAZAN (2002)], o *Waba* original tinha 16kb de memória disponíveis e o *SuperWaba*, antes do novo gerenciador, 50kb). Foi também adicionado suporte a *Exceptions*.

Segundo Guilherme, o objetivo do *SuperWaba* é, após sua concretização no mercado de *handhelds*, atingir as plataformas de telefonia celular.

Em abril de 2002, o *SuperWaba* estava com mais de 3000 usuários cadastrados, em mais de 100 países. Estudantes, empresas de pequeno e grande porte são os maiores utilizadores do *SuperWaba*. O interessante, dentre estes clientes, é que nem sempre a empresa é uma empresa de informática, mas sim de serviços ou alimentação, que criaram pequenas equipes de informática para tentar resolver seus problemas, e com o *SuperWaba* conseguiram [HAZAN (2002)].

Vale lembrar que o *SuperWaba* é distribuído sob a licença *Lesser General Public License* (LGPL) e, sendo assim, vem com o código fonte junto e é *ROYALTEE FREE*, ou seja, a empresa não paga nada para distribuir produtos feitos sob a plataforma [HAZAN (2002)]. Na verdade, a partir de Dezembro de 2004, com a versão 5.0, o *SuperWaba* passa a ter dois pacotes de desenvolvimento:

- Pacote Comunitário: Distribuído gratuitamente para a criação de protótipos de aplicativos mais complexos, implementação de soluções abertas ou pesquisas de universidades, com algumas limitações de bibliotecas da ferramenta.
- Pacote Profissional: Versão completa da ferramenta, distribuída por meio de uma licença de 1350 reais a ser paga anualmente para desenvolvimento de aplicativos proprietários.

2.9.2 Diferenças entre o *Waba* e o *SuperWaba*

Desde que a versão *Waba1.0b* foi lançada, em 14 de fevereiro de 2000, ele nunca mais foi modificado por *Rick Wild*, seu criador. Existem algumas pessoas que melhoraram um pouco a versão 1.0b, através de um projeto no *sourceforge* com o codinome *waba*, mas segundo informações do site do *Waba* [WABA (2003c)], eles também não a modificam desde maio de 2001. O *SuperWaba*, por sua vez, desde a versão 1.0G até a versão atual (versão 5.0) sofreu mais de 31 lançamentos. Da versão 1.0G, de julho de 2002, até a versão atual foram mais de um lançamento por mês.

Com relação aos programas *Warp* (*Waba Resource Packager*) e *Exegen*⁵, existentes na tecnologia *Waba*, estes se tornaram programas *Java* normais, ou seja, necessitam do interpretador *java.exe* instalado na máquina para serem executados. Esta mudança torna o desenvolvimento em *SuperWaba* multiplataforma. Outra mudança está no fato de não existir mais os arquivos *WRP* (espécie de arquivo zip com todos os elementos utilizados na aplicação) para *Windows CE* existente na plataforma *waba*, deve-se usar o mesmo arquivo *PDB* utilizado pelo *PalmOS*.

Para terminar, a tabela 2.2 apresenta uma comparação entre o *SuperWaba* e o *Waba* em relação a diversas categorias.

2.9.3 Diferenças entre o *J2ME* e o *SuperWaba*

O objetivo do *SuperWaba* é prover uma plataforma para o desenvolvimento de aplicações em *PDA*s, que são dispositivos com grandes restrições de recursos. O objetivo do *J2ME* também é proporcionar uma plataforma de desenvolvimento

⁵*Warp* e *Exegen* são ferramentas utilizadas para geração de arquivos de um determinado programa para serem executados no *PDA*. O *Warp* captura, a partir da classe principal, todas as classes e imagens que são utilizadas por ela e se encontram na definição do caminho da classe (*classpath*) e os armazena no arquivo *PDB*. O *Exegen* gera o arquivo *PRC*, que guardará o ícone e algumas outras informações, como proteção contra cópia e versão do seu produto

| Descrição | Waba | SuperWaba |
|--|---|--|
| Número de pacotes (exceto o java.lang, waba.applet e pacotes de extensão). | 36 | 69 |
| Tamanho (classes + VM). | 72.3Kb | 293Kb |
| Comparação de tempo de execução(executando o programa Bench). | 390860ms | 162900ms |
| Memória disponível para programas em PalmOS. | 32kb | Ilimitado |
| Suporte a exceções. | Não | Sim |
| Suporte a cores. | Não(presente na conversão do sourceforge) | Tons de cinza e cores em todos os dispositivos |
| Suporte a tipos double/long (precisão de 64 bits) | Não | Sim |
| Interface com o usuário. | Extremamente simples | Muito completo |

Tabela 2.2: Comparação entre o *Waba* e o *SuperWaba* - Fonte: *SuperWaba - SuperWaba X Outras Plataformas* [SW (2004b)]

para estes dispositivos, mas de uma maneira diferente. O *J2ME* não define uma especificação única, pois isto é impossível no seu mercado alvo. São várias as especificações definidas, cada uma voltada para um certo grupo de dispositivos que compartilham características em comum.

Para resumir, o *J2ME* define configurações e *profiles*. Uma configuração define uma *API* básica e uma máquina virtual para um grupo grande de dispositivos. Um *profile* é um perfil construído sobre uma configuração e visa limitar o grupo de dispositivos em que ela pode atuar através do acréscimo de certas características particulares a um grupo menor de dispositivos. Assim, para desenvolver em *J2ME*, basta escolher a configuração e o *profile* cujas características melhor se adequam ao aparelho alvo.

Para o desenvolvimento em *PDA*s, deve-se utilizar a configuração *CLDC*, que implementa a *KVM* como máquina virtual, e o *profile MIDP 1.0*. Mas este *profile* não foi desenvolvido visando o mercado de *PDA*s, mas sim o de celulares. Assim, como os celulares possuem restrições de recursos bem maiores que os *PDA*s, ao

se desenvolver para um *PDA* não se utiliza todo o potencial possível do aparelho. A Figura 2.20 compara as duas plataformas.

| SuperWaba X J2ME | | | |
|---|--|--|---|
| Características | CLDC / MIDP 2.0 | CDC / PP | SuperWaba |
| Royalty Free | Depende | Depende | Sim |
| Plataforma | Palm OS | Pocket PC (Não disponível para Palm OS) | Ambos |
| compatível com J2ME | Sim | Sim | Não |
| Entrada e Saída | tcp/ip, datagram, file | tcp/ip, datagram, file | tcp/ip, file, SD/ Compactflash cards |
| Interface com o usuário (novos em <i>italic</i>) | alert, gauge, list, textbox, textfield, stringitem, <i>ticker</i> | dialog, scrollbar, choice, textarea, textfield, label, checkbox, <i>list, menu,</i> <i>button</i> | messagebox, scrollbar, combobox, multiedit, edit, label, checkbox, listbox, menubar, button |
| Pacote útil | calendar, date, hashtable, random, stack, timer, vector | arraylist, arrays, bitset, dictionary, hashtable, stack, timer, vector | date, hashtable, random, vector, <i>inhtable, intvector</i> |
| Porta Serial | Sim | Sim | Sim |
| Ponto Flutuante | Não | Sim | Sim |
| Scanners da Symbol | Não | Não | Sim |
| Memória ocupada (J9/IBM) | 901kb | 1500kb | 340kb |

Figura 2.20: Comparação entre *SuperWaba* e *J2ME* - Fonte: *SuperWaba - SuperWaba X Outras Plataformas* [SW (2004b)]

Capítulo 3

Metodologia

Neste capítulo será apresentada a metodologia empregada no desenvolvimento deste trabalho, bem como o ambiente de desenvolvimento, a rede de comunicação em que o sistema está inserido e as tecnologias e plataformas adotadas.

3.1 Pesquisa-Ação

O presente trabalho trata-se de um desenvolvimento experimental de um *software*, baseando-se nas linhas de estudo e pesquisa de tecnologia da informação aplicada à Segurança Pública.

O método utilizado baseia-se na pesquisa-ação, com abordagem qualitativa, com o objetivo de aumentar a compreensão do caso em estudo.

Segundo Thiollent [THIOLLENT (1997)], na pesquisa-ação aplicada em sistemas de informação, o pesquisador participa da implementação de um sistema e, simultaneamente, realiza intervenções de ordem técnica. No contexto da informação a pesquisa-ação tem sido pensada como instrumento adaptado ao estudo, em situação real, das mudanças organizacionais que acompanham a introdução de novas tecnologias, principalmente as baseadas na informática. Com ela pretende-se facilitar a implementação e a assimilação das novas técnicas.

3.2 Ambiente de Desenvolvimento

O presente trabalho foi desenvolvido no Núcleo de Desenvolvimento de Projetos da Sexta Região da Polícia Militar, que tem sede na cidade de Lavras/MG e abrange uma totalidade de 141 cidades e 9 Distritos. A Sexta Região de Polícia

Militar é dividida em seis comandos de Unidades Operacionais, sendo estes: o Oitavo Batalhão com sede em Lavras, o Vigésimo Batalhão com sede em Pouso Alegre, o Vigésimo Quarto Batalhão com sede em Varginha, o Vigésimo Nono Batalhão com sede em Poços de Caldas, a Quinta Companhia Independente com sede em Itajubá e a Décima Quarta Companhia Independente com sede em São Lourenço.

Para o desenvolvimento do trabalho, contou-se com a seguinte infraestrutura:

- Computadores *desktop* para o desenvolvimento de todas as partes do sistema (COPOM, servidor e viatura).
- PDAs para testes relacionados com o sistema das viaturas.
- Acesso à rede *wireless* UFLA.
- Disponibilização de viaturas por parte da PMMG, para realização de testes relativos ao sistema.
- Apoio teórico dos professores do Departamento de Ciência da Computação (DCC) da Universidade Federal de Lavras (UFLA) e do Chefe do Núcleo de Desenvolvimento de Projetos da Sexta Região de Polícia Militar.

3.3 Rede de Comunicação, Plataformas e Tecnologias

O ambiente de comunicação do sistema é o da rede *wireless* do provedor *UFLANet* com o padrão de tecnologia *Wi-Fi (Wireless Fidelity)*, que são redes locais pertencente ao padrão *IEEE 80211* de redes sem fio, que utilizam transmissão por *FHSS e DSSS*, definidos com detalhes na seção 2.3. A rede *wireless* da *UFLANet* é composta pelos seguintes elementos:

- Para a transmissão de sinais: Duas antenas omni-direcionais, que são antenas capazes de transmitir sinais de um único ponto para vários outros pontos simultaneamente. Essas antenas estão localizadas na Serra da Bocaina e Zona Norte da cidade de Lavras, Minas Gerais.
- Para a conexão de clientes: uma antena direcional, que concentra o sinal em uma única direção e um adaptador *Wi-Fi* com *pigtail*. No caso específico da utilização de PDAs pode-se utilizar tanto o adaptador *Wi-Fi* com *pigtail* como cartões *wireless* embutidos no aparelho que realizam conexão direta, além de um ponto de acesso sem fio (*bridge* ou *access-point*).

O alcance da rede é de cerca de mais ou menos 40 Km, provendo acesso rápido com links que, por meio de testes realizados, já chegaram até 11 Mbps. Como mecanismo de segurança, a rede implementa o protocolo de segurança *Wired Equivalent Privacy (WEP)*, definido na seção 2.5.5.

O desenvolvimento do sistema foi feito em computadores *desktop* com a plataforma *Windows XP*, e os *PDA*s utilizados nos testes durante o desenvolvimento foram os da plataforma *Windows CE* e *PalmOS 5*.

As tecnologias adotadas no desenvolvimento dos sistemas foram as seguintes:

- Sistema do COPOM: tecnologia *Java*, (*J2SE - Java 2 Platform, Standard Edition*).
- Sistema das viaturas: tecnologia *SuperWaba*, padrão de desenvolvimento de aplicações para dispositivos móveis, baseado em *Java*.
- Sistema Servidor : tecnologia *Java* (*J2EE - Java 2 Platform, Enterprise Edition*).
- SGBD para armazenamento e gerenciamento de informações compartilhadas pelos sistemas: *SGBD Microsoft SQLServer 2000*.

Para a implementação dos três sistemas utilizou-se a ferramenta *NetBeans* da empresa *Sun Microsystems* [NETBEANS (2004)], própria para o desenvolvimento de aplicativos da plataforma *Java*.

Inicialmente, o sistema foi documentado na linguagem *Unified Modeling Language (UML)* [UML (2003)] utilizando-se a ferramenta *Rational Rose Enterprise Edition*. A modelagem da interface do sistema foi feita utilizando-se a ferramenta *Microsoft Visio* [VISIO (2003)], e a modelagem do banco de dados foi feita utilizando-se a ferramenta *ERwin* [ERWIN (2004)].

Capítulo 4

Resultados e Discussão

Este capítulo tem por objetivo descrever maiores detalhes sobre o projeto. A seção 4.1 apresenta a modelagem do sistema, criada e utilizada durante o desenvolvimento do mesmo. A seção 4.2, descreve todas as características e funcionalidades do sistema. As seções 4.3 e 4.4 apresentam uma discussão sobre as tecnologias adotadas no desenvolvimento do sistema. A seção 4.5 apresenta os testes realizados e os resultados obtidos.

4.1 Modelagem do Sistema

Para que o desenvolvimento do projeto fosse feito de maneira simples e com maior qualidade optou-se pela documentação do mesmo antes do início de sua implementação propriamente dita.

Na primeira fase da documentação, com o objetivo de conhecer melhor o escopo do projeto, foi criado o Documento de Requisitos SisComPM, identificando em alto nível suas principais funcionalidades, divididas em requisitos do sistema. Esses requisitos descrevem aquilo que o sistema deve fazer e em que condições. O documento de requisitos pode ser encontrado no site oficial do sistema [SISCOMPM (2004)].

Depois de aprovado, o documento de requisitos foi utilizado como base para a criação da modelagem do sistema, com o objetivo principal de facilitar a posterior manutenção do mesmo, prevendo, antes do início da implementação, os principais problemas que poderiam ocorrer, buscando soluções para os mesmos. A modelagem foi feita utilizando-se a linguagem *Unified Modeling Language (UML)* [UML (2003)], que é uma linguagem padrão para modelagem de sistemas. Foram

criados diagramas de caso de uso, que indicam em um nível mais alto o comportamento do sistema, descrevendo determinada funcionalidade. Geralmente existe uma correspondência de 1 para 1, entre os diagramas de caso de uso e os requisitos do projeto constantes no Documento de Requisitos. Foram criados também os diagramas de classe que indicam em um nível mais baixo, a organização do sistema em pacotes e classes, segundo o paradigma de Orientação a Objetos.

Os diagramas de caso de uso do sistema estão ilustrados nas figuras 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17.

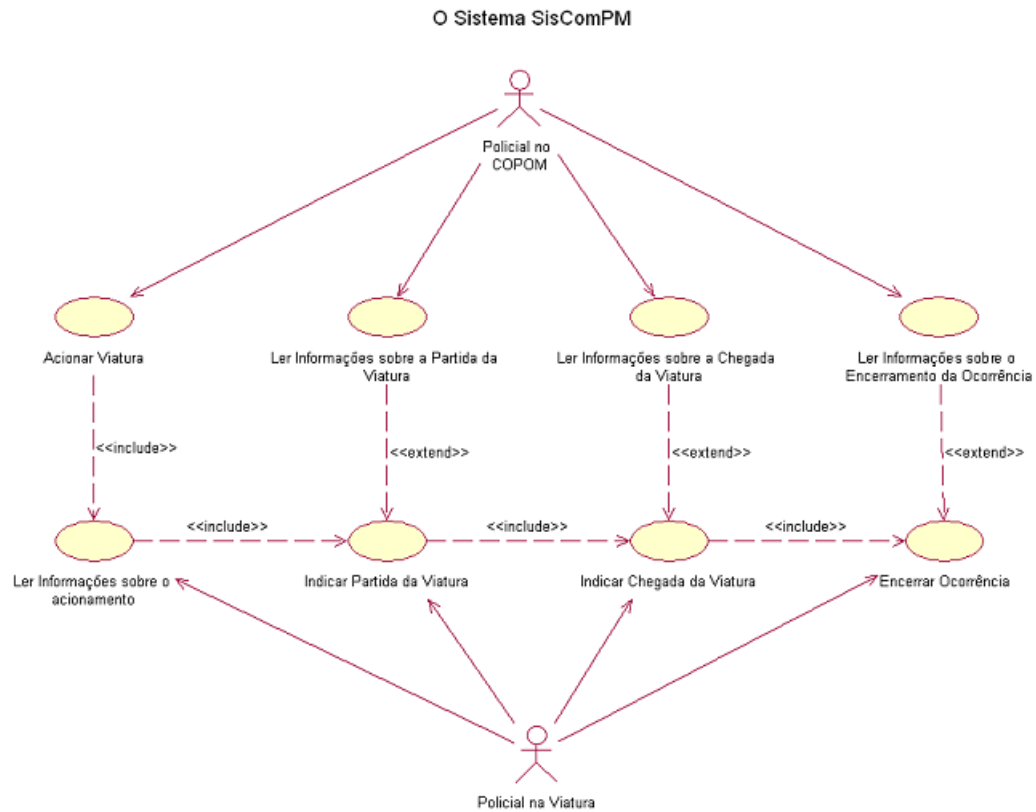


Figura 4.1: Diagrama de Caso de Uso - Esquema Geral - Fonte: Documentação do Sistema

RF001 - Efetuar Logon no COPOM

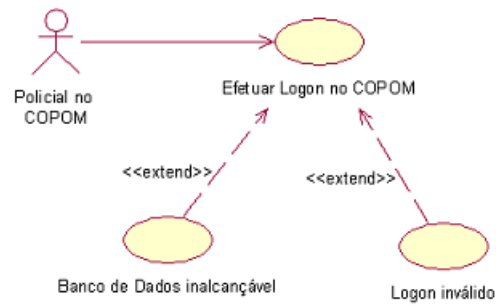


Figura 4.2: Diagrama de Caso de Uso - Efetuar Logon no Sistema do COPOM -
Fonte: Documentação do Sistema

RF003 - Efetuar Logon na Viatura

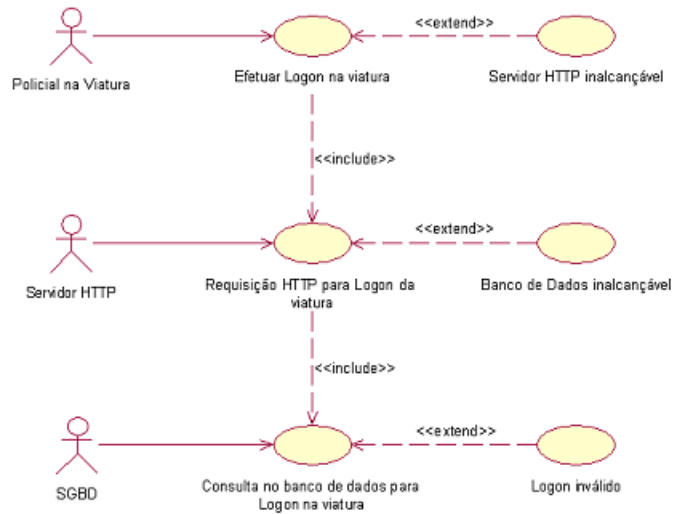


Figura 4.3: Diagrama de Caso de Uso - Efetuar Logon no Sistema da Viatura -
Fonte: Documentação do Sistema

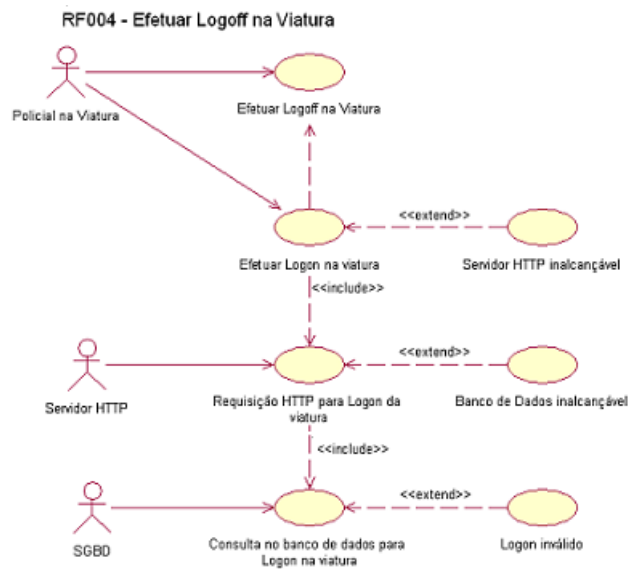


Figura 4.4: Diagrama de Caso de Uso - Efetuar Logoff no Sistema da Viatura -
 Fonte: Documentação do Sistema

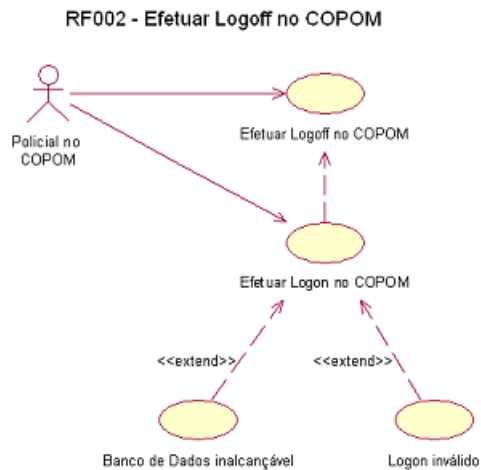


Figura 4.5: Diagrama de Caso de Uso - Efetuar Logoff no Sistema do COPOM -
 Fonte: Documentação do Sistema

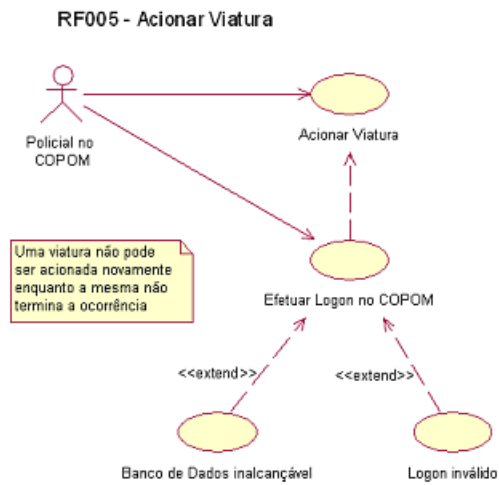


Figura 4.6: Diagrama de Caso de Uso - Acionar Viatura - Fonte: Documentação do Sistema

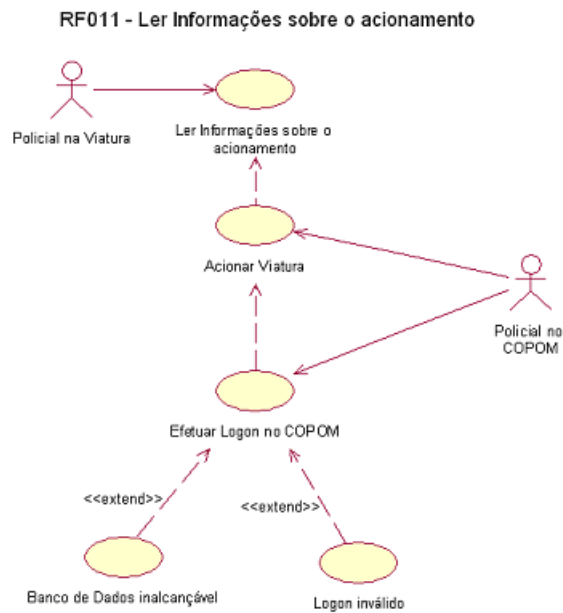


Figura 4.7: Diagrama de Caso de Uso - Ler Informações sobre o Acionamento - Fonte: Documentação do Sistema

RF006 - Indicar Partida da Viatura

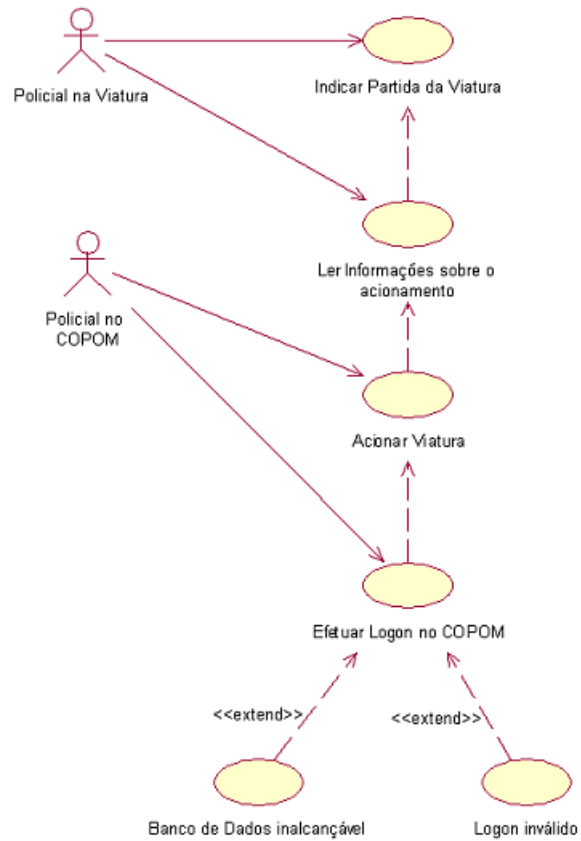


Figura 4.8: Diagrama de Caso de Uso - Indicar Partida da Viatura - Fonte: Documentação do Sistema

RF012 - Ler Informações sobre a partida da Viatura

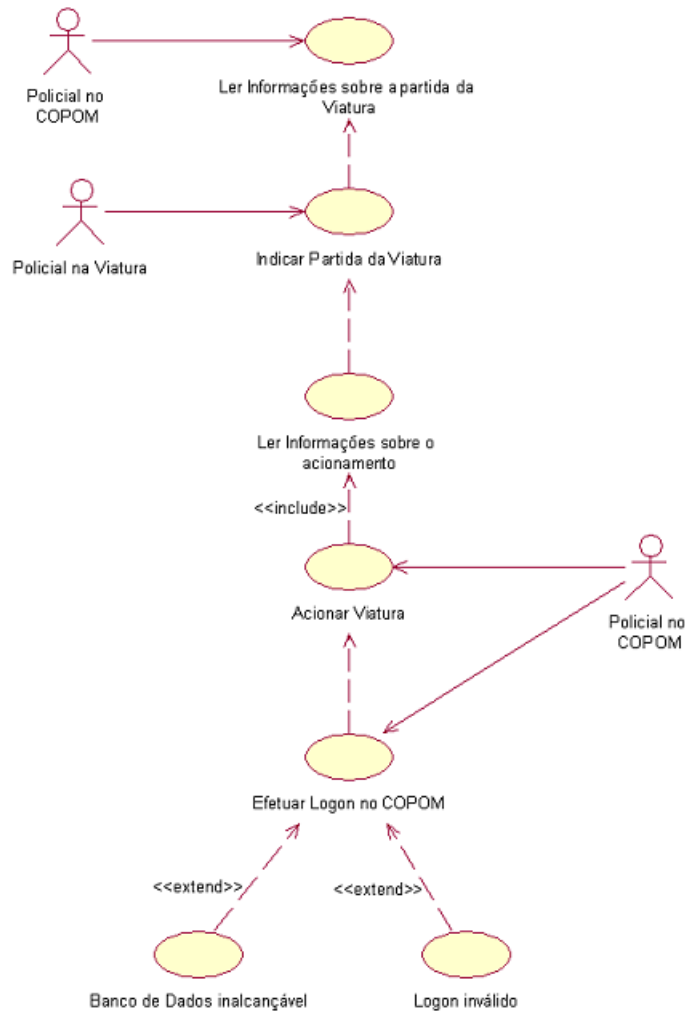


Figura 4.9: Diagrama de Caso de Uso - Ler Informações sobre a Partida da Viatura
- Fonte: Documentação do Sistema

RF007 - Indicar Chegada da Viatura

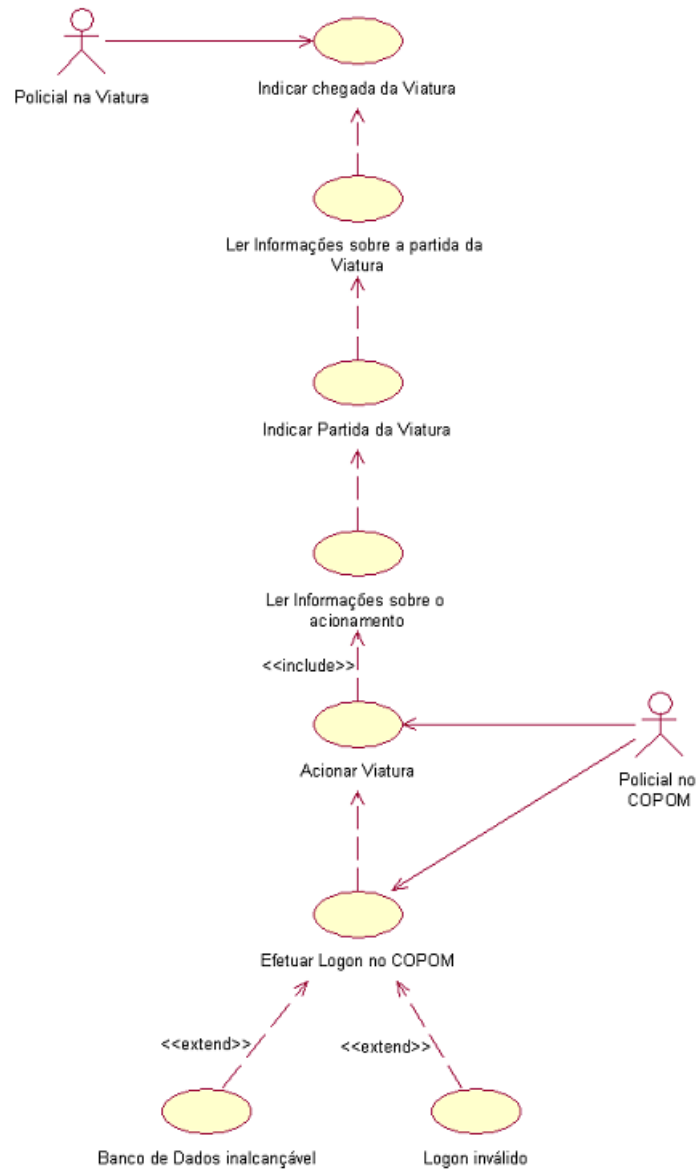


Figura 4.10: Diagrama de Caso de Uso - Indicar Chegada da Viatura - Fonte: Documentação do Sistema

RF013 - Ler Informações sobre a chegada da Viatura

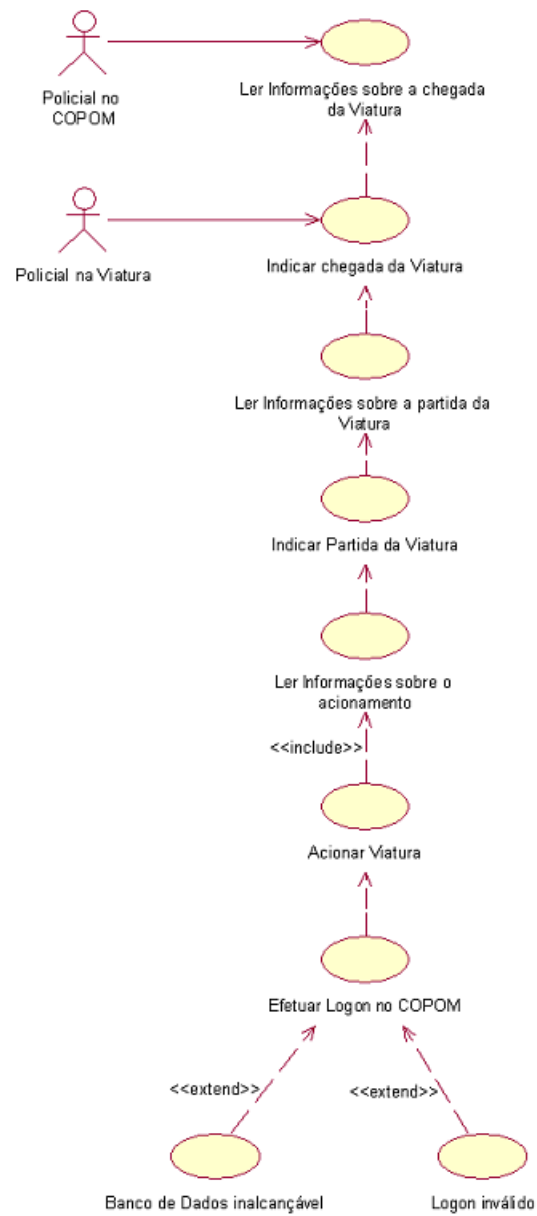


Figura 4.11: Diagrama de Caso de Uso - Ler Informações sobre a Chegada da Viatura - Fonte: Documentação do Sistema

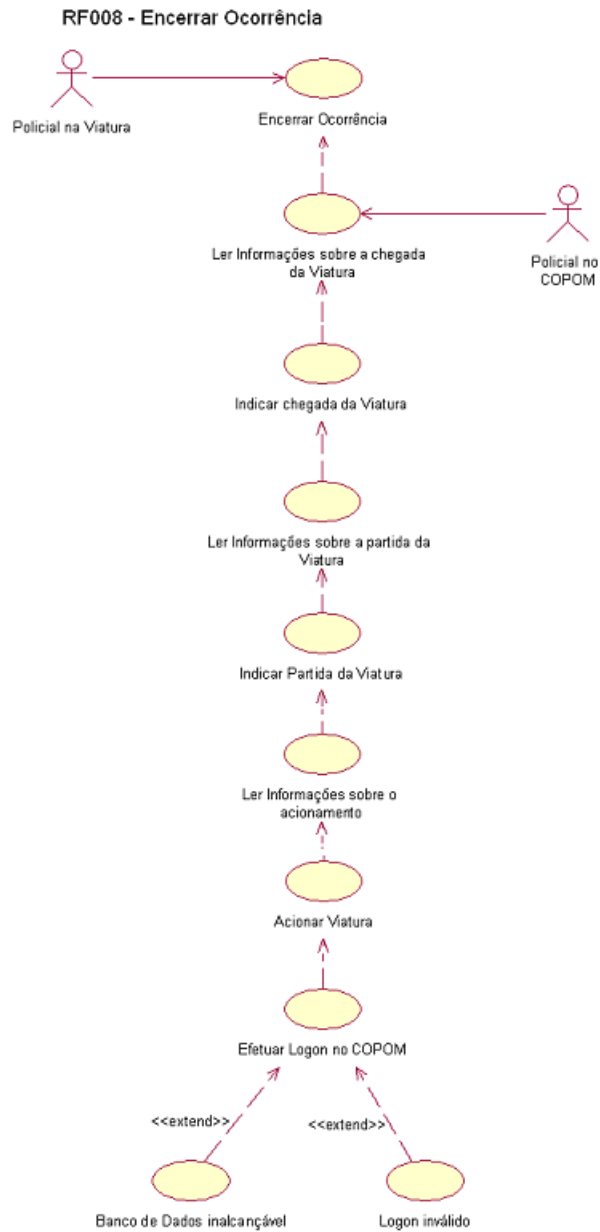


Figura 4.12: Diagrama de Caso de Uso - Encerrar Ocorrência - Fonte: Documentação do Sistema

RF014 - Ler Informações sobre o Encerramento da Ocorrência

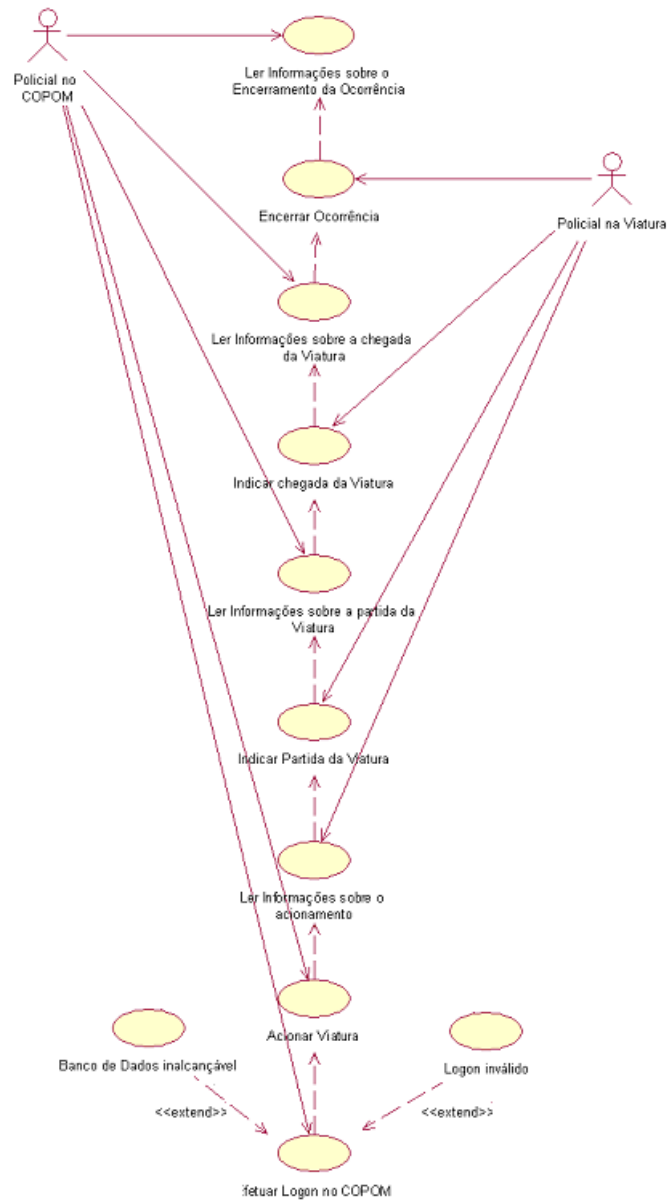


Figura 4.13: Diagrama de Caso de Uso - Ler Informações sobre o Encerramento da Ocorrência - Fonte: Documentação do Sistema

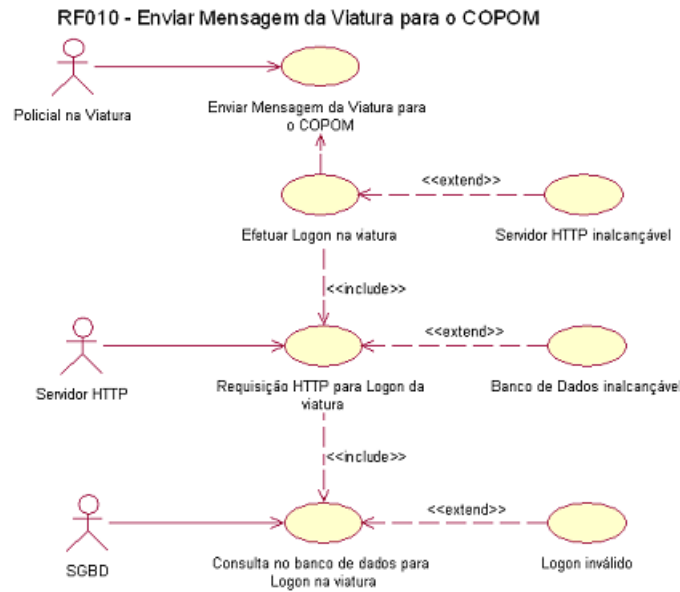


Figura 4.14: Diagrama de Caso de Uso - Enviar Mensagem para o COPOM - Fonte: Documentação do Sistema

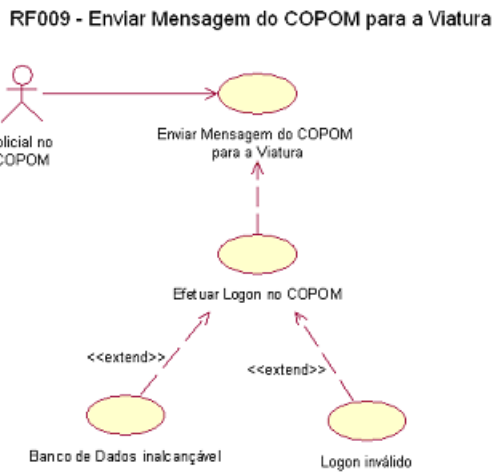


Figura 4.15: Diagrama de Caso de Uso - Enviar Mensagem para a Viatura - Fonte: Documentação do Sistema

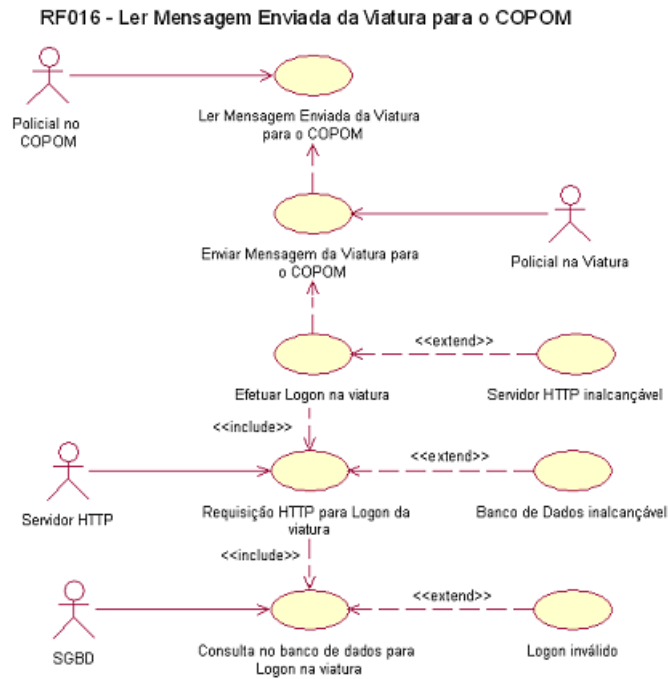


Figura 4.16: Diagrama de Caso de Uso - Ler Mensagem enviada pela Viatura -
 Fonte: Documentação do Sistema

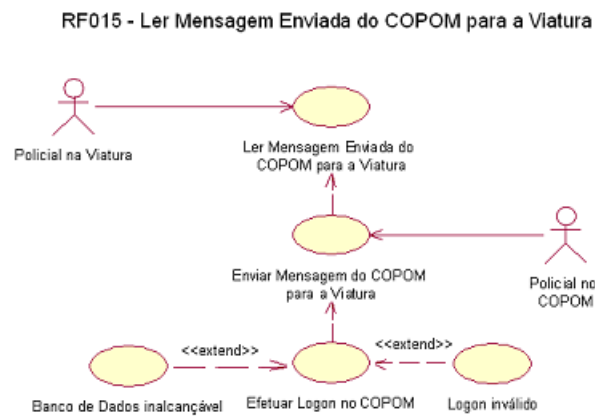


Figura 4.17: Diagrama de Caso de Uso - Ler Mensagem enviada pelo COPOM -
 Fonte: Documentação do Sistema

Parte dos diagramas de classe do Sistema está ilustrada na figura 4.18. A figura 4.19 ilustra os pacotes que formam o Sistema Servidor e sua dependência em relação aos pacotes do Sistema da Viatura.

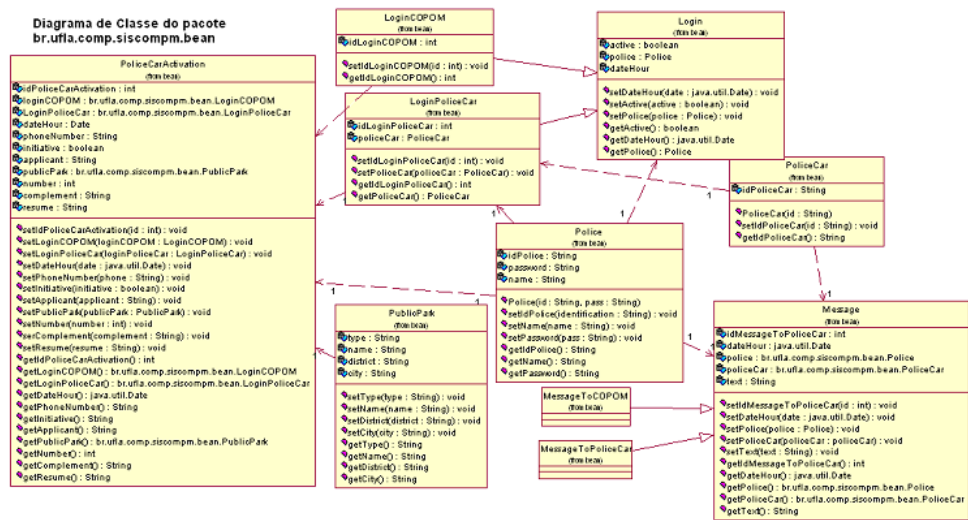


Figura 4.18: Diagrama de Classes do Pacote `br.ufpa.comp.siscompm.bean` - Fonte: Documentação do Sistema

Dependencia entre os pacotes do servidor

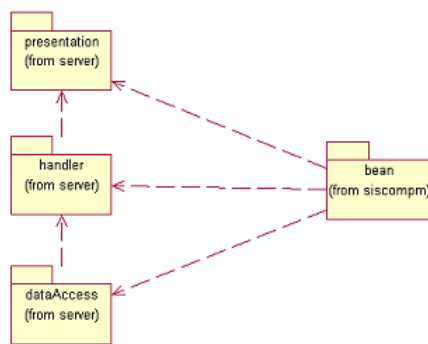


Figura 4.19: Pacotes do Sistema Servidor - Fonte: Documentação do Sistema

A base de dados do sistema também foi modelada, e a partir dessa modelagem foi possível a criação automática das tabelas que compõem atualmente o banco de dados do sistema. O modelo relacional do banco de dados SisComPM, com suas respectivas tabelas e relacionamentos está ilustrado na figura 4.20.

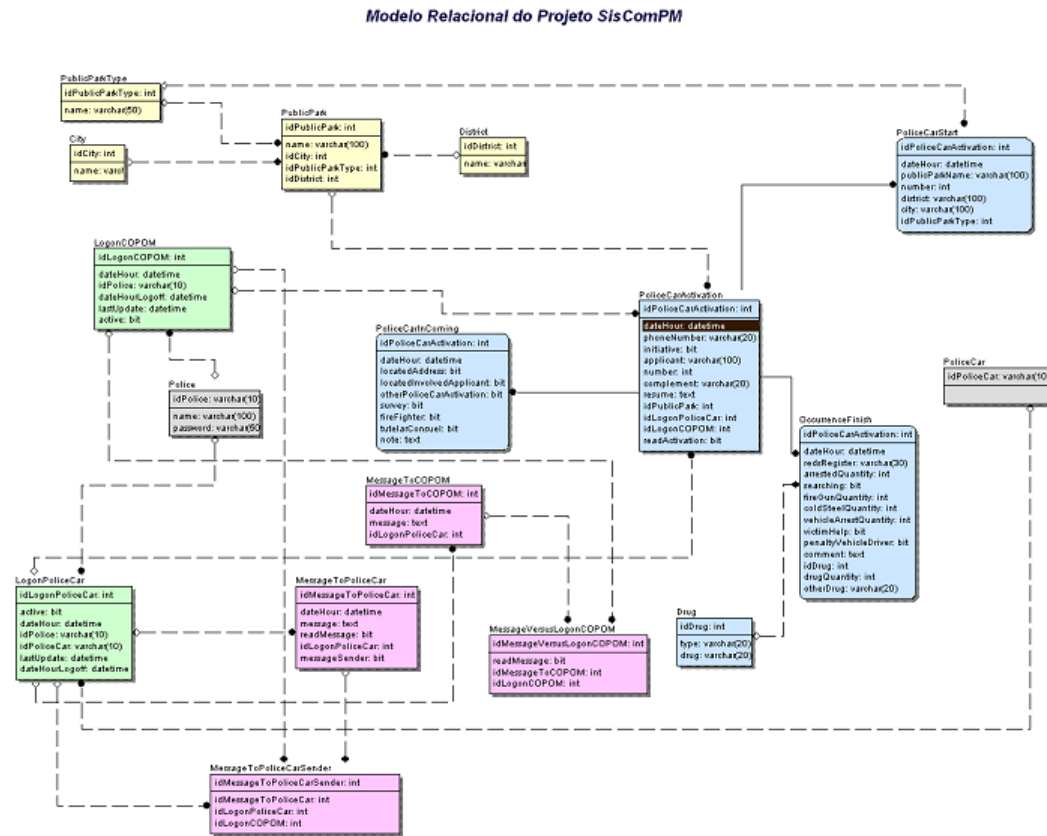


Figura 4.20: Modelo Relacional - Banco de Dados SisCompM - Fonte: Documentação do Sistema

4.2 O Sistema SisComPM

O SisComPM é um sistema de comunicação em redes *wireless*, que tem como objetivo principal prover uma maior segurança na comunicação entre o Centro de Operações Policial Militar (COPOM) e viaturas policiais militares no momento do atendimento de ocorrências.

A comunicação ocorre entre dois sistemas:

- Sistema do COPOM: sistema acessado através de um computador *desktop* pelo policial de plantão no COPOM, responsável pelo acionamento de viaturas para o atendimento de ocorrências. As figuras 4.21 e 4.22 ilustram respectivamente a tela de *logon* e a tela inicial do Sistema do COPOM.
- Sistema da Viatura: sistema acessado através de um dispositivo móvel (*PDA*) pelo policial na viatura, responsável pelo atendimento da ocorrência. A figura 4.23 representa a tela de *logon* do Sistema da Viatura.

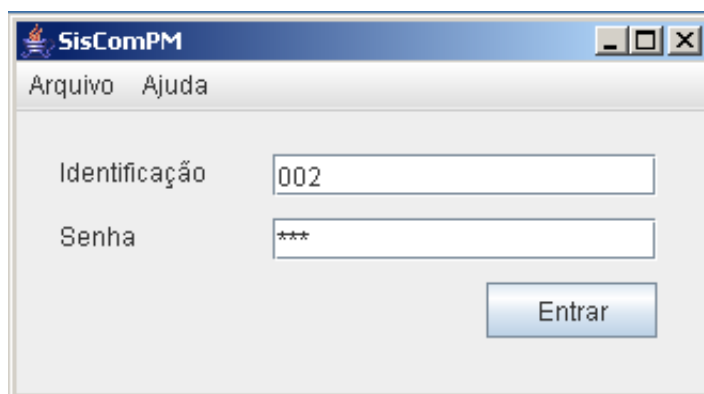


Figura 4.21: Sistema do COPOM - Tela de Logon - Fonte: Execução do Sistema

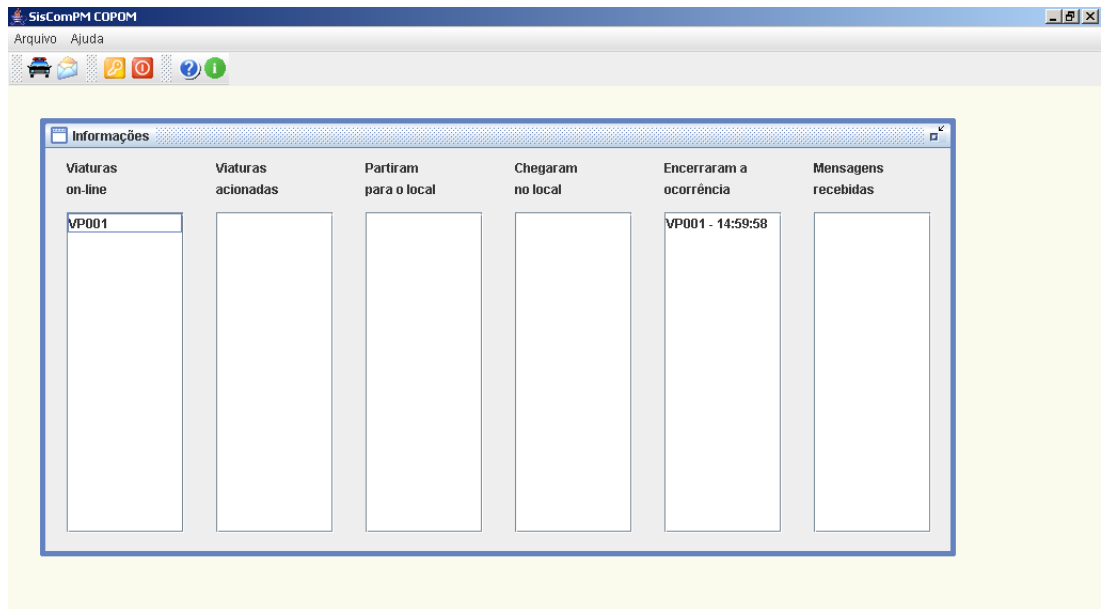


Figura 4.22: Sistema do COPOM - Tela inicial - Fonte: Execução do Sistema [SISCOMPM (2004)]

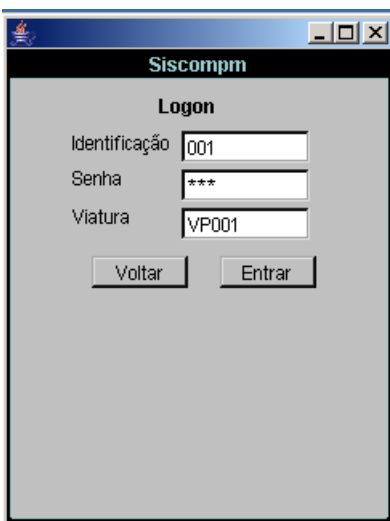


Figura 4.23: Sistema da Viatura - Tela de *logon* - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)]

No sistema, uma viatura pode apresentar diferentes estados:

- Viatura Não Acionada, aguardando acionamento (figura 4.24).
- Viatura Acionada, aguardando partida para o local da ocorrência (figura 4.25).
- Viatura Acionada, partindo para o local da ocorrência (figura 4.26)
- Viatura Acionada, no local da ocorrência (figura 4.27).

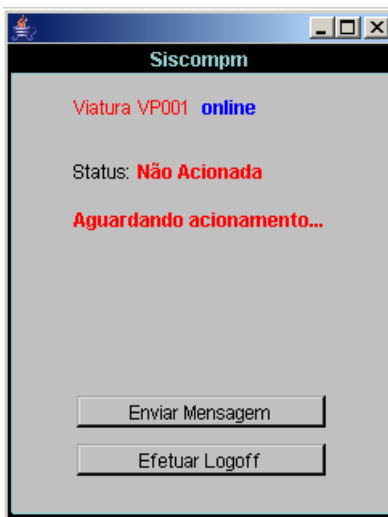


Figura 4.24: SisComPM - Viatura Não Acionada, aguardando acionamento -
Fonte: Execução do Sistema SisComPM [SISCOMPM (2004)]

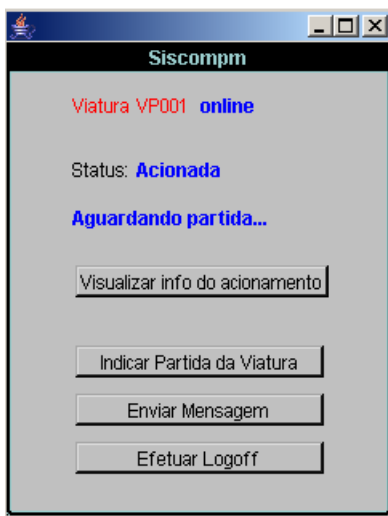


Figura 4.25: SisComPM - Viatura Acionada, aguardando partida para o local da
ocorrência - Fonte: Execução do Sistema [SISCOMPM (2004)]

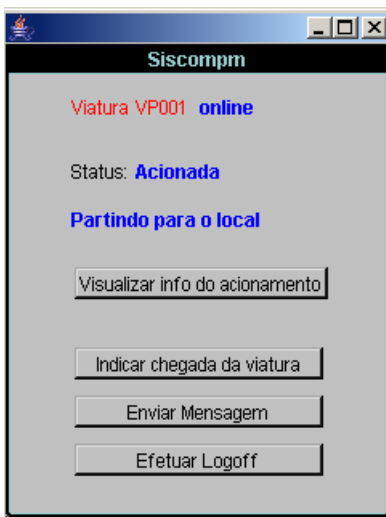


Figura 4.26: SisComPM - Viatura Acionada, partindo para o local da ocorrência - Fonte: Execução do Sistema SisComPM [SISCOMPM (2004)]

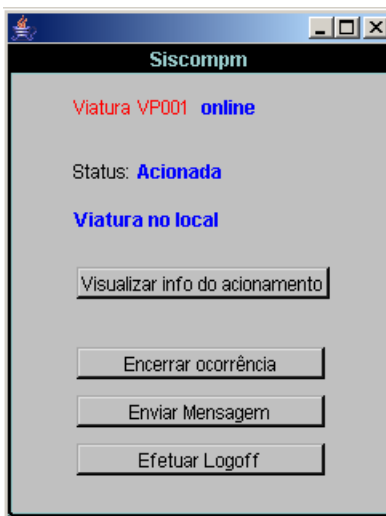


Figura 4.27: SisComPM - Viatura Acionada, no local da ocorrência - Fonte: Execução do Sistema SisComPM [SISCOMPM (2004)]

Durante a transição de um estado para o próximo, o Sistema da Viatura envia

para o Sistema do COPOM as informações obtidas durante o estado anterior, que são armazenadas no banco de dados. As principais informações compartilhadas pelos dois sistemas são:

- Informações sobre as viaturas conectadas naquele momento (figura 4.28).
- Informações sobre o acionamento enviadas pelo COPOM à viatura tais como nome do solicitante, local da ocorrência, etc (figuras 4.30 e 4.29).
- Informações sobre o local de onde a viatura partiu para o atendimento da ocorrência. Essa informação pode ser útil para prever o tempo que a viatura poderá levar para chegar ao local da ocorrência, no caso de situações de emergência (figuras 4.31 e 4.32).
- Informações sobre a chegada da viatura no local da ocorrência tal como localização do solicitante e do local da ocorrência (figuras 4.33 e 4.34).
- Informações sobre as providências tomadas no encerramento da ocorrência tais como apreensão de armas, drogas, etc (figuras 4.35 e 4.36).

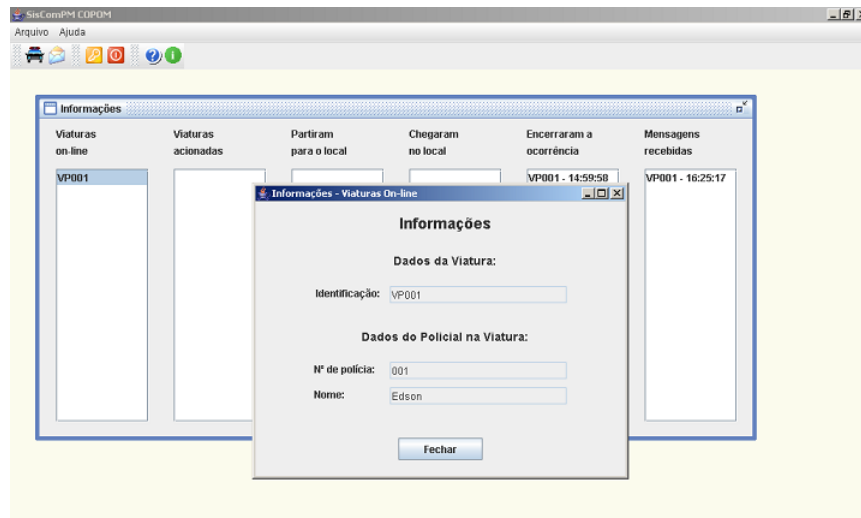


Figura 4.28: SisComPM - Informações sobre viaturas conectadas ao sistema - Fonte: Execução do Sistema SisComPM [SISCOMPM (2004)]

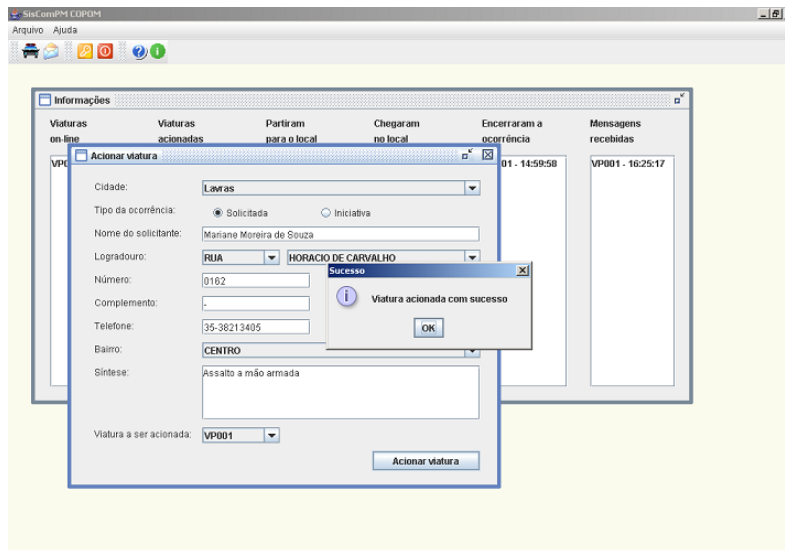


Figura 4.29: SisComPM - Acionamento da Viatura - Fonte: Execução do Sistema SisComPM [SISCOMPM (2004)]

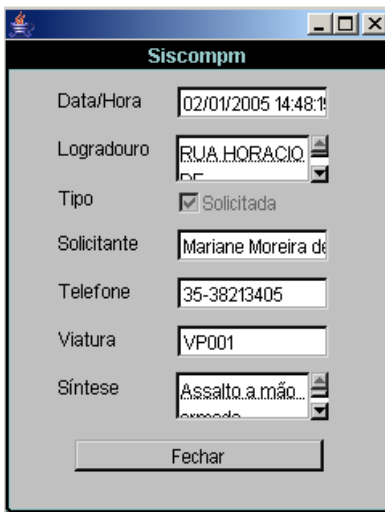


Figura 4.30: SisComPM - Informações sobre o acionamento recebidas pelo Sistema da Viatura - Fonte: Execução do Sistema[SISCOMPM (2004)]

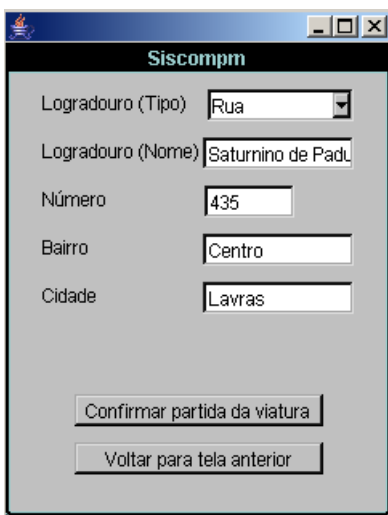


Figura 4.31: SisComPM - Informações sobre o local de partida da viatura para o atendimento da ocorrência enviadas pelo Sistema da Viatura - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)]

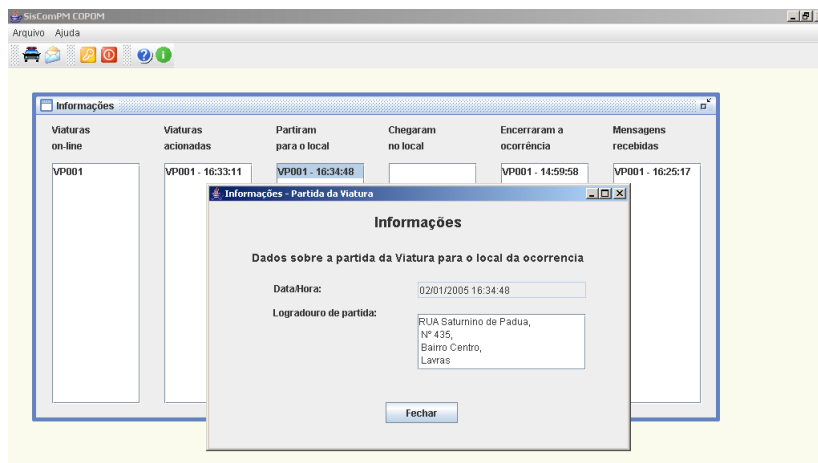


Figura 4.32: SisComPM - Informações sobre o local de partida da viatura para o atendimento da ocorrência recebidas pelo Sistema do COPOM - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)]

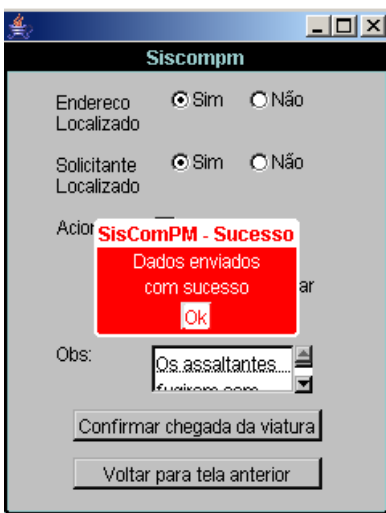


Figura 4.33: SisComPM - Informações sobre a chegada da viatura ao local da ocorrência enviadas pelo Sistema da Viatura - Fonte: Execução do Sistema SisComPM [SISCOMPM (2004)]

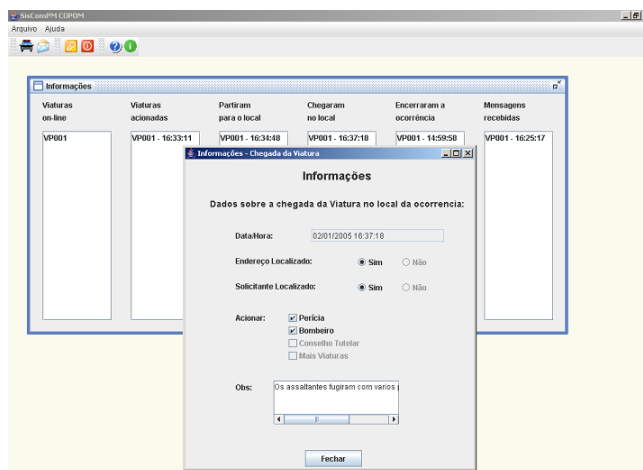


Figura 4.34: SisComPM - Informações sobre a chegada da viatura ao local da ocorrência recebidas pelo Sistema do COPOM - Fonte: Execução do Sistema SisComPM [SISCOMPM (2004)]

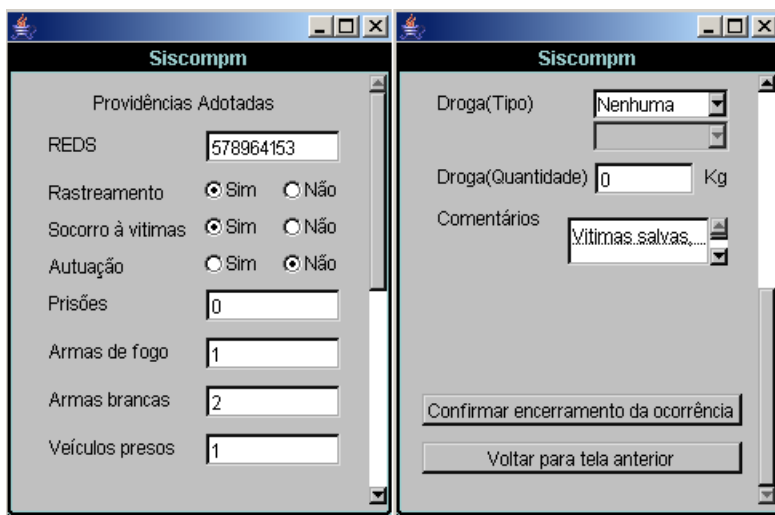


Figura 4.35: SisComPM - Informações sobre as providências tomadas no encerramento da ocorrência enviadas pelo Sistema da Viatura - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)]

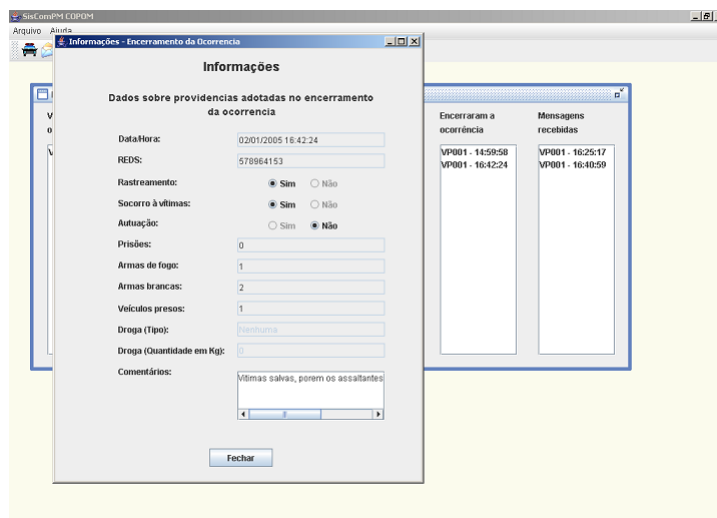


Figura 4.36: SisComPM - Informações sobre as providências tomadas no encerramento da ocorrência recebidas pelo Sistema do COPOM - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)]

Desta maneira, é possível obter um histórico de todas as ocorrências atendidas pelas viaturas em um determinado período, tornando possível, por exemplo, a identificação de áreas com grandes índices de criminalidade na cidade, a partir da análise de dados estatísticos do sistema.

O sistema funciona da seguinte maneira:

- Uma pessoa realiza uma chamada por telefone ao COPOM, especificando a ocorrência.
- O policial de plantão no COPOM, uma vez conectado ao sistema, visualiza as viaturas conectadas ao mesmo, escolhe uma das viaturas que estejam disponíveis e realiza o acionamento (figura 4.29).
- O policial na viatura recebe um aviso de que sua viatura foi acionada, fornece uma resposta e parte para o atendimento da ocorrência (figura 4.37).



Figura 4.37: SisComPM - Aviso de novo acionamento recebido pelo Sistema da Viatura - Fonte: Execução do Sistema [SISCOMPM (2004)]

Tanto durante o atendimento de uma ocorrência quanto durante o período em que a viatura aguarda um acionamento, é possível que mensagens sejam trocadas pelos operantes dos sistemas do COPOM e da Viatura. As figuras 4.38, 4.39, 4.40, 4.41, 4.42 e 4.43 ilustram uma troca de mensagem entre os sistemas.

A rede de comunicação em que o sistema está inserido é ilustrada na figura 4.44.

O Sistema do COPOM e o Sistema da Viatura não se comunicam diretamente. A comunicação entre os mesmos é intermediada pelo banco de dados, por meio da utilização de *Stored Procedures*¹. Essa arquitetura permite que as informações trocadas pelo sistema estejam centralizadas, facilitando sua posterior recuperação.

A tecnologia utilizada no Sistema do COPOM, especificada na seção 2.7, permite que o mesmo tenha acesso direto ao banco de dados por meio da rede privativa da Polícia Militar. Já a tecnologia utilizada no Sistema da Viatura, especificada na seção 2.9, não permite o acesso direto ao banco de dados, de maneira que foi necessário a criação de um terceiro sistema (Sistema Servidor ou ponte *HTTP (Hyper Text Transfer Protocol)*²) responsável por esse acesso. Maiores informações sobre este sistema são encontradas na seção 4.4.

As informações trocadas entre o Sistema da Viatura e o Sistema Servidor, em ambos os sentidos, serão criptografadas. O método de criptografia utilizado está definido na seção 4.4. A utilização da criptografia dos dados entre o Sistema da Viatura e o Sistema Servidor é necessária, pois existe a possibilidade de um simples navegador *WEB* interagir com o sistema, capturando dados do banco de dados através do Sistema Servidor. Com a utilização da criptografia essa interação se torna impraticável.

As principais vantagens desse novo sistema de comunicação sobre o sistema de comunicação via rádio, vigente atualmente, é a possibilidade de transmissão de informação de uma maneira mais segura e a possibilidade de armazenamento das informações trocadas para posterior geração de relatórios. Apesar disso, é importante dizer que o novo sistema proposto não foi feito para substituir totalmente o sistema de comunicação à rádio utilizado atualmente, mesmo porque em momentos em que é necessário uma maior agilidade e rapidez na transmissão, o sistema via rádio é mais vantajoso. O novo sistema de comunicação deve ser utilizado juntamente com o sistema atual quando existir a necessidade de se transmitir informações confidenciais que não deveriam ser captadas por nenhum tipo de invasor.

Além das funcionalidades básicas do sistema, existem ainda algumas tarefas relacionadas à implementação de sistemas celulares, que são executadas implicitamente pelos sistemas. Essas tarefas são:

¹*Stored Procedures* ou Procedimentos Armazenados são conjuntos de comandos *SQL*, que realizam uma determinada operação. É um programa na linguagem *SQL* compilado, que pode ser executado manualmente ou por outros programas

²*HTTP* é um protocolo utilizado para transmissão de dados na rede

- Verificação de novos acionamentos e novas mensagens.
- Verificação e atualização de usuários inativos nos sistemas aos quais não foi possível informar ao sistema a desconexão.
- Gerenciamento de *logons* (acesso ao sistema) de mesmo usuário.

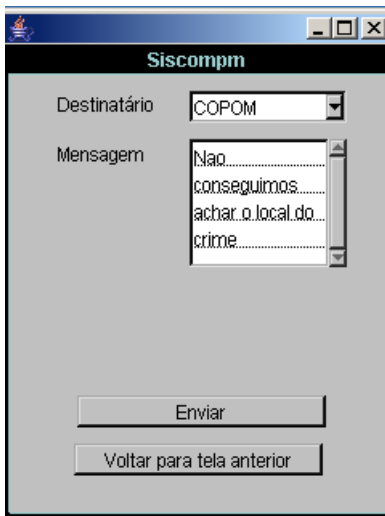


Figura 4.38: SisComPM - Mensagem enviada pelo Sistema da Viatura - Fonte: Execução do Sistema SisComPM [SISCOMP (2004)]

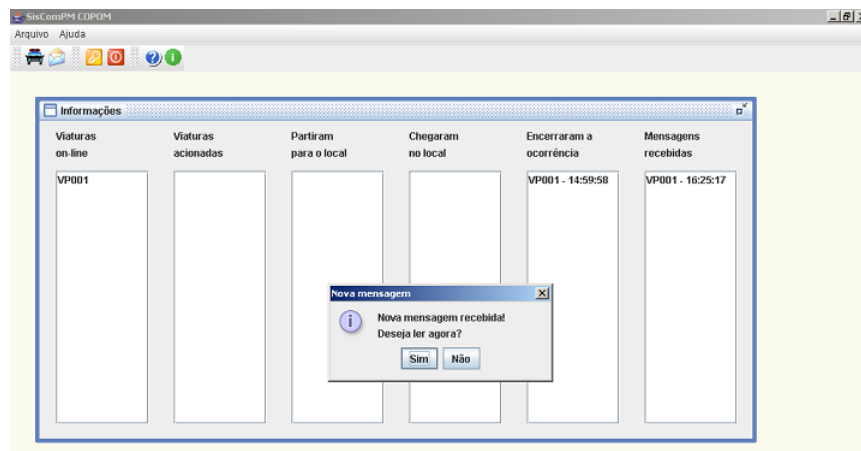


Figura 4.39: SisComPM - Aviso de nova mensagem recebida pelo Sistema do CO-POM - Fonte: Execução do Sistema [SISCOMPM (2004)]

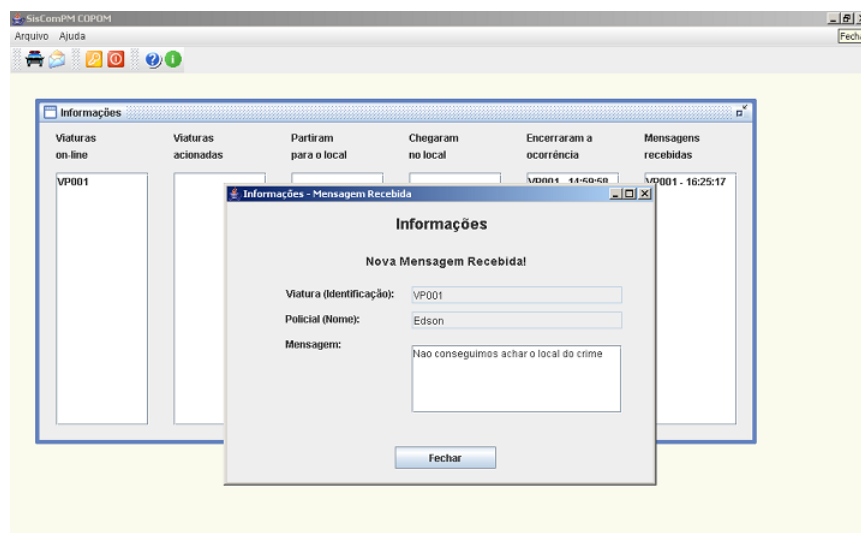


Figura 4.40: SisComPM - Nova mensagem recebida pelo Sistema do COPOM - Fonte: Execução do Sistema SisComPM [SISCOMPM (2004)]

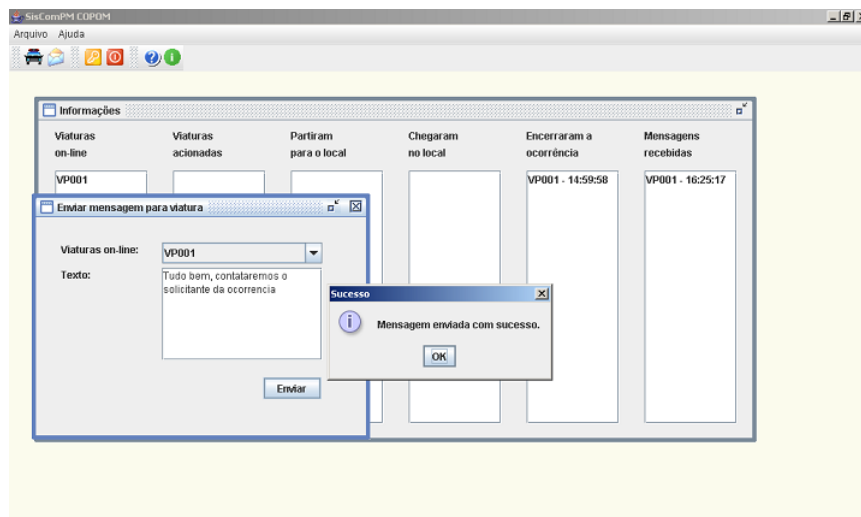


Figura 4.41: SisComPM - Mensagem enviada pelo Sistema do COPOM - Fonte: Execução do Sistema SisComPM [SISCOMPM (2004)]



Figura 4.42: SisComPM - Aviso de nova mensagem recebida pelo Sistema da Viatura - Fonte: Execução do Sistema [SISCOMPM (2004)]

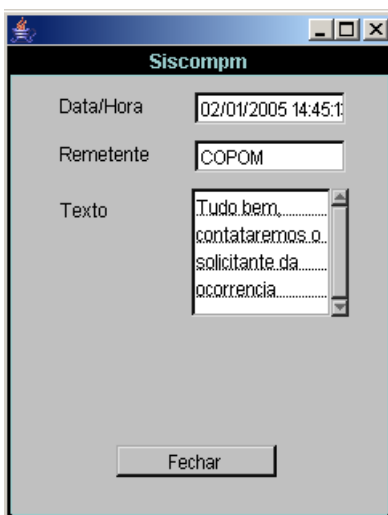


Figura 4.43: Nova Mensagem recebida pelo Sistema da Viatura - Fonte: Execução do Sistema SisComPM [SISCOMPM (2004)]

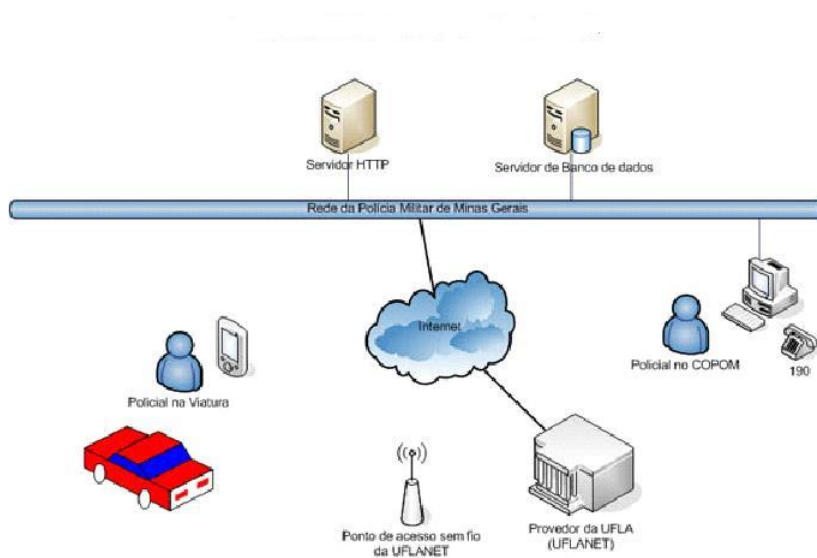


Figura 4.44: Diagrama de Rede do projeto SisComPM - Fonte: *Projeto SisComPM - Sistema de Comunicação Segura entre 190 e viaturas PM* [SISCOMPM (2004)]

4.3 Adoção das tecnologias *Java* e *SuperWaba* e do SGBD SQLServer

A tecnologia *Java* foi a tecnologia escolhida para ser utilizada no Sistema do CO-POM (*J2SE*), no Sistema Servidor (*J2EE*) e indiretamente no Sistema da Viatura, com a adoção da tecnologia *SuperWaba*. Essa tecnologia foi adotada devido a sua alta portabilidade (vide seção 2.7), permitindo a utilização do sistema em diversas plataformas.

Outro motivo foi a facilidade de programação oferecida pela linguagem, devido à total Orientação a Objetos³ apresentada pela mesma e por se tratar não apenas de uma linguagem de programação, mas de um padrão de tecnologia que se encontra em constante melhoria, com um número cada vez maior de desenvolvedores.

Outro fator importante foi o fato de ser uma plataforma livre, ou seja, não é necessário o pagamento de nenhum tipo de licença para utilizá-la, de maneira que este fato colocou a plataforma *Java* como uma das principais candidatas à adoção no projeto, permanecendo acima de outras linguagens como *Delphi*, *Visual Basic*, tecnologias *.NET*, dentre outras, que são ferramentas de ótima qualidade, porém pagas.

Para a escolha da tecnologia a ser utilizada no sistema dos *PDA*s nas viaturas, diversas outras foram analisadas. Cada uma dessas tecnologias está definida na seção 2.6.

O formato de arquivos gerados pela ferramenta foi fundamental para a sua escolha ou eliminação do projeto. Como exemplo, imagine uma ferramenta que gere apenas aplicações em um formato nativo para um sistema operacional como o *PalmOS*, mas os aparelhos disponíveis para o projeto são baseados apenas em *Windows CE*, esta ferramenta seria eliminada, por mais poderosa que fosse. Agora, imagine que para o projeto estão disponíveis *PDA*s com três tipos de sistemas operacionais. Uma ferramenta que gere arquivos em um formato intermediário e que possua o *runtime* (vide seção 2.6) para os três sistemas operacionais seria a mais adequada.

Antes de escolher, Bernstein [BERNSTEIN (2001)] propõe que a análise do projeto como um todo deve ser realizada pensando em todos os parâmetros possí-

³Orientação a Objetos é um paradigma de programação que pressupõe uma organização de *software* em termos de coleção de objetos discretos incorporando estrutura e comportamento próprios. Este paradigma garante uma maior independência entre diferentes partes do sistema, permitindo uma maior reusabilidade de suas funcionalidades

veis. Alguns destes parâmetros, são:

- **Tamanho e performance do aplicativo X Facilidade e rapidez de desenvolvimento:** aplicativos desenvolvidos na linguagem *C* terão um tamanho bem menor do que os demais. Quando bem escrito, sua performance poderá ser bem diferenciada. Entretanto, o desenvolvimento em *C* requer tempo e habilidade. Caso se preencha estes dois pré-requisitos, o resultado final com certeza será bem melhor. Se o programador conseguir montar uma biblioteca de funções, o problema do tempo será reduzido. A flexibilidade da linguagem *C* proporciona um poder muito grande, como o acesso a todas as funções nativas do sistema operacional, no caso do *PalmOS*. O desenvolvedor deve estar bem consciente nessa escolha, pois o nível de dificuldade é bem maior comparado aos das ferramentas *RAD* (*Rapid Application Development*), o que implica no fator tempo, que é tão importante na hora de dimensionar o prazo de um projeto para um cliente.

Assim, deve-se verificar com o cliente o tempo disponível para o desenvolvimento do projeto e então definir se a ferramenta a utilizar será mais flexível, mas de desenvolvimento lento ou uma ferramenta *RAD*. Caso se exija performance a escolha mais adequada é por ferramentas flexíveis, mas o fator tempo ainda deve ser considerado.

- **Runtime:** avaliar o espaço ocupado e se a empresa cobra um custo por *runtime*, ou seja, se ao desenvolver um projeto ainda deverá se pagar um custo de distribuição por *PDA* que executará o programa. Deve-se avaliar a real necessidade de *runtime*.
- **Preço das ferramentas:** avaliar o preço da ferramenta a ser utilizada, tomando como parâmetros o número de desenvolvedores (instalações) na empresa e, caso tenha *runtime*, o custo de cada um instalado nos *PDAs* de seus clientes. Ficar também atento à licença de uso por tempo determinado, além da política de *upgrades*.

Analisando-se as ferramentas propostas na seção 2.6 de acordo com os critérios definidos acima, estão listadas abaixo cada uma das ferramentas, bem como os motivos pelos quais as mesmas não foram adotadas no projeto.

- **NSBasic** [NSBASIC (2003)]: apesar de ser uma ferramenta poderosa e com características mais que adequadas para o projeto, inclusive por seu caráter multiplataforma, não pode ser utilizada devido o seu custo, considerado alto para o projeto.

- ***PDA ToolBox*** [PDA (2003)]: a sua utilização foi descartada pelo fato de apresentar suporte apenas ao *PalmOS*.
- ***PocketStudio*** [POCKET (2003)]: sua utilização no projeto não foi aprovada devido ao fato de apresentar suporte apenas ao *PalmOS* e pelo seu alto custo.
- ***Satellite Forms MobileApp Designer***[SATELLITE (2003)]: o custo foi o que tornou a escolha desta ferramenta inadequada para a utilização no projeto.
- **A linguagem C, o CodeWarrior, o PRC Tools**: o custo e o caráter de plataforma única destas ferramentas impediu o seu uso no projeto.
- ***Embedded Visual Tools***: o fato de estar disponível apenas para o desenvolvimento em *Windows CE* não permitiu a sua escolha.
- ***Tecnologias baseadas em Java (J2ME)***: o fato de não existir uma máquina virtual gratuita para o sistema operacional *Windows CE* não permitiu sua escolha, uma vez que no início do projeto os *PDA*s disponíveis tinham apenas esse sistema operacional.

A tecnologia *Superwaba* (vide seção 2.9) foi escolhida para ser utilizada neste projeto, na implementação do sistema para os *PDA*s nas viaturas, devido ao fato de ser totalmente gratuita, muito poderosa, com uma interface gráfica completa e multiplataforma.

A figura 4.45 mostra as vantagens e desvantagens do *SuperWaba* em relação a outras plataformas de desenvolvimento.

| SuperWaba X Outros | | | | |
|---------------------------------|--------------|--------------|------------------|---|
| Ferramenta / Plataforma | Linguagem | Velocidade | Custo da Licença | Implementações |
| Satellite Forms (Pumatech) | Script | lento | US\$995.00 | Palm OS, PocketPC |
| NS Basic | Basic | lento | US\$445.00 | Palm OS, PocketPC, Newton |
| Mobile VB (AppForge) | Visual Basic | lento | US\$900.00 | Palm OS, Symbian |
| SuperWaba | Java | médio | Nenhum | PalmOS, PocketPC, Browser, Win2k |
| Pocket Studio | Pascal | rápido | US\$249.99 | Palm OS |
| Code Warrior (MetroWerks) | C / C++ | rápido | US\$500.00 | Palm OS |
| eMbedded Visual C++ (MicroSoft) | C / C++ | rápido | Nenhum | PocketPC |
| Developer Studio (Falch.net) | C / C++ | rápido | US\$229.00 | Palm OS |

Figura 4.45: Vantagens do *SuperWaba* em relação a outras plataformas de desenvolvimento - Fonte: *SuperWaba - SuperWaba X Outras Plataformas* [SW (2004b)]

Outras características que levaram à escolha do *SuperWaba* como plataforma de desenvolvimento do sistema dos *PDA*s nas viaturas, foram [SW (2004c)]:

- O *SuperWaba* é uma comunidade crescente de usuários: mais de 1300 novos usuários por mês.
- Em 2003, o *SuperWaba SDK* teve cerca de 25000 *downloads* e o site recebeu mais de 250.000 visitas.
- Uma comunidade que contribui gratuitamente: por volta de 60 por cento do *SDK* foram contribuições de seus membros.
- Várias novas versões por ano e inúmeras melhorias e correções.
- A plataforma é disponibilizada sob a licença *GNU Lesser General Public License (LGPL)*, que permite o desenvolvimento de aplicações comerciais, ao mesmo tempo que protege a plataforma de ser fechada.
- Portabilidade: aplicações desenvolvidas são executadas em todos os *PDA*s suportados nas várias plataformas sem requererem quaisquer modificações ou especializações.

- Bibliotecas poderosas e fáceis de usar, focadas no desenvolvimento rápido de aplicações, com um baixo uso de memória.
- Baixo custo de propriedade, livre de licenças (*royalty free*) até a versão 4 e derivadas.
- Acesso ao código fonte, permite customizações para necessidades específicas, além de permitir a verificação, por questões de segurança ou estratégicas.
- Baixo risco de descontinuidade: O projeto *SuperWaba* pode ser mantido pela própria comunidade e não está nas mãos de apenas uma companhia.

As figuras 4.46, 4.47, 4.48 e 4.49, mostram outras características importantes dessa plataforma, com relação a quatro categorias: Máquina Virtual, Bibliotecas de Extensão, Entrada e Saída e Interface com o usuário.

| Máquina Virtual |
|--|
| Diversas plataformas suportadas |
| Permite a criação de bibliotecas nativas em C |
| Suporte completo a Unicode |
| Suporte a escala de cinza (Palm OS 2.0 ou superior), PDAs coloridos e também alta-resolução em todos os PDAs |
| Suporte a exceptions, double e long |
| Utilização de toda a memória disponível para suas aplicações: SuperWaba VM amplia a capacidade do PDA. |

Figura 4.46: Características do *SuperWaba* com relação à Máquina Virtual - Fonte: *SuperWaba - Características da Plataforma* [SW (2004a)]

| Bibliotecas de Extensão |
|--|
| Container de HTML robusto |
| Tokenizador de Xml |
| Classes para exibição de informações de GPS, baseados no protocolo Garmin. |
| Suporte para scanners da Symbol® para Pocket PC e Palm OS. |
| Suporte a leitura de arquivos PalmDoc e aos formatos de arquivo PalmZip. |
| Suporte a algoritmos de criptografia: Blowfish, MD5, SHA1, TEA |
| Personal Information Abstract Layer : permite acessar dados pessoais nas plataformas Pocket PC and Palm OS |
| API de Jogos, suportando sprites, botões animados, etc. |

Figura 4.47: Características do *SuperWaba* com relação a Bibliotecas de Extensão - Fonte: *SuperWaba- Características da Plataforma* [SW (2004a)]

| Entrada e Saída |
|---|
| Suporte a TCP/IP, porta serial, USB, Infra-vermelho, Bluetooth. |
| Manipulação de arquivos em cartões Secure Digital e Memory Stick. |
| Utiliza arquivos PDBs também no Pocket PC, tornando o banco de dados específico da Palm multi-plataforma. |
| Especificação de Conectividade para Banco de Dados(WDBC): com drivers for IBM Db2e e o PDB SQL Driver |
| Capacidade de impressão. |

Figura 4.48: Características do *SuperWaba* com relação a Entrada e Saída - Fonte: *SuperWaba - Características da Plataforma* [SW (2004a)]

| Interface com o Usuário |
|--|
| Dois estilos diferentes para todos os controles: Palm OS and Windows CE. |
| Em ambos os estilos, todos os controles têm um estado desabilitado visível |
| Janelas popup arrastáveis. |
| Posicionamento relativo faz com que o programa se adapte a qualquer resolução. |

Figura 4.49: Características do *SuperWaba* com relação a Interface com o usuário - Fonte: *SuperWaba - Características da Plataforma* [SW (2004a)]

O banco de dados escolhido para armazenamento e gerenciamento dos dados relacionados ao sistema de comunicação foi o *Microsoft SQLServer 2000*.

O Microsoft *SQLServer* 2000 é uma ferramenta proprietária poderosa de armazenamento e gerenciamento de dados, criada pela Microsoft.

Este *SGBD* (Sistema Gerenciador de Banco de Dados) foi escolhido para o projeto pelos seguintes motivos [SQL (2000)]:

- Alta capacidade de armazenamento de informações.
- Maior segurança dos dados.
- Ferramentas gráficas que facilitam a manipulação de registros e tabelas.
- Possibilidade de utilização de procedimentos armazenados (*Stored Procedures*).

No desenvolvimento do projeto optou-se pela utilização de *Stored Procedures*. Essa escolha foi devido à independência da parte lógica do sistema em relação ao banco de dados fornecida, permitindo que alterações relacionadas a este sejam totalmente independentes do código do sistema, não sendo necessário recompilá-lo novamente, garantindo assim, uma maior portabilidade do sistema desenvolvido.

4.4 O Pacote de Comunicação *PDA/SGBD* com utilização de Criptografia por Chave Privada

Durante o desenvolvimento do sistema foi descoberto um problema grave: A tecnologia adotada para implementação do sistema dos *PDA*s nas viaturas não possuía uma *API* (*Applicativo Interface Protocol*)⁴ nativa de conexão com um *SGBD* qualquer. Para o sistema a ser desenvolvido isso era essencial, uma vez que um dos objetivos do mesmo era armazenar todas as informações trocadas pelo sistema de maneira centralizada de modo a facilitar sua recuperação para elaboração de relatórios. Sem o acesso a um *SGBD* essa funcionalidade estaria prejudicada, e os dados seriam armazenados localmente no banco de dados *PDB* do *PDA* (vide seção 2.6). Para resolver este problema, foi desenvolvido um pacote genérico de comunicação entre um *PDA* e um *SGBD* quaisquer. Este pacote é formado pelos seguinte elementos:

- Um Sistema Servidor que representa a ponte *HTTP*, responsável pelo acesso do Sistema da Viatura ao *SGBD*.

⁴*API* é um conjunto normalizado de rotinas e chamadas de *software* que podem ser referenciadas por um programa aplicativo para acessar serviços essenciais de uma rede

- Uma classe de formatação de texto denominada *MyResultSet*, que organiza todas as informações na forma de um vetor de registros a serem armazenados no banco de dados (no caso de informações transmitidas no sentido *PDA/SGBD*) ou registros retornados de uma consulta ao banco de dados (no sentido *SGBD/PDA*).

Uma vez desenvolvido o pacote de comunicação, outro problema, desta vez relacionado à segurança do sistema foi observado: Devido à transmissão de informações entre o Sistema da Viatura e o Sistema Servidor ocorrer via protocolo *HTTP*, existe a possibilidade de uma interação com o sistema por outras vias que não seja o próprio sistema, como um navegador *WEB* comum, no qual um intruso teria acesso facilmente a informações confidenciais da polícia.

Neste sentido, observou-se a necessidade de utilização de métodos de criptografia para codificar informações que fossem trocadas entre o Sistema da Viatura e o Sistema Servidor antes que as mesmas fossem enviadas via protocolo *HTTP*. Sendo assim, um possível invasor poderia acessar a informação por meio de um navegador *WEB*, porém, ele teria acesso à mensagem codificada, sem nenhum tipo de significado associado, e esta seria decodificada ao alcançar seu destinatário.

O método de criptografia escolhido para implementação no sistema foi o método de Criptografia por Chave Privada (vide seção 2.5.1). Esse método foi escolhido pela maior facilidade de implementação e maior rapidez no processo de encriptar e decriptar mensagens, apesar da necessidade de criação de uma chave privada ou simétrica para cada par transmissor/receptor. Uma idéia proposta para futuros trabalhos seria a implementação de um Centro de Distribuição de Chaves, especificado na seção 2.5.1, que ameniza o problema da privacidade e distribuição da chave simétrica ou ainda a implementação do método de Criptografia por Chave Pública (vide seção 2.5.2), visando aumentar ainda mais a segurança do sistema.

De acordo com a figura 4.50, a transmissão de dados entre o Sistema da Viatura e o Sistema Servidor utilizando o método de Criptografia por Chave Privada será feita da seguinte maneira:

- Inicialmente têm-se uma mensagem (informação) a ser transmitida por qualquer um dos sistemas (Viatura ou Servidor)
- A mensagem é criptografada pelo sistema com uma determinada chave simétrica e enviada via ponte *HTTP* pela *Internet*.
- Ao chegar ao sistema destinatário, a mensagem é decriptografada utilizando-se a mesma chave simétrica do transmissor, que também é conhecida pelo receptor.

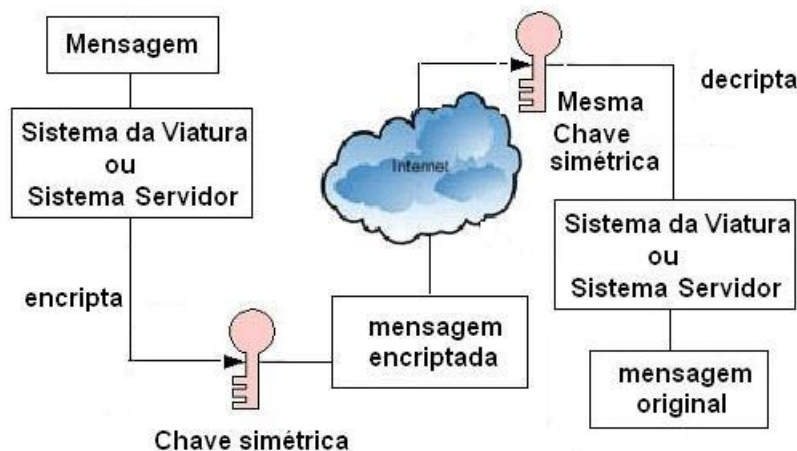


Figura 4.50: Transmissão de Informações entre Sistema Servidor e Sistema da Viatura utilizando Criptografia por Chave Privada - Fonte: *Projeto SisComPM - Sistema de Comunicação Segura entre 190 e viaturas PM* [SISCOMP (2004)]

4.5 Testes e Resultados Aplicados

Durante o desenvolvimento do sistema foram realizados vários testes. Os testes relacionados ao Sistema do COPOM e Sistema Servidor foram realizados em computadores *desktop*, e os testes relacionados com o Sistema da Viatura foram realizados tanto no simulador do *PDA* presente nos computadores *desktop*, como no próprio *PDA*, o que facilitou bastante a visualização de detalhes de interface.

A configuração dos equipamentos utilizados na realização dos testes de desenvolvimento é a seguinte:

- Computadores *desktop* para testes com o Sistema do COPOM e Sistema Servidor: *Pentium 4*, 1.2 GHz, 512 MB de memória, na plataforma *Microsoft Windows XP*.
- PDAs para testes com o Sistema da Viatura: *Pocket PC* com processador ARM, 16 MB de memória, na plataforma *Pocket PC(Windows CE)* (figura 4.51); *Palm Tungsten C* com processador ARM, 32 MB de memória na plataforma *PalmOS 5.2.1*, com teclado aclopado (figura 4.52).



Figura 4.51: HP Jornada 560 - Fonte: *InfoSync Norge* [INFOSYNC (2004)]



Figura 4.52: Palm Tungsten C - Fonte: *ARC Organisers* [ARC (2004)]

É importante destacar que como a tecnologia utilizada no desenvolvimento do Sistema da Viatura é multiplataforma (vide seção 2.9), existem outros PDAs, de custos mais acessíveis, que também podem ser utilizados na implantação do sistema. A tecnologia *SuperWaba* funciona na maioria dos PDAs com o sistema operacional *PalmOS* e *Windows CE/PocketPC/Windows Mobile* para modelos com *Windows CE 2.11* ou superior. De acordo com informações do site do *SuperWaba*, os seguintes aparelhos suportam a plataforma:

- *Aparelhos com sistema operacional PalmOS:*
 - *AlphaSmart: Dana, Dana.wireless*
 - *Handspring:*
 - * *Treo 180, 270, 600*
 - * *Visor Pro (PalmOS 3.5)*
 - * *Visor Platinum*
 - * *Visor Prism*
 - *Palm:*
 - * *Professional, III, IIIx, IIIc, IIIe, V, Vx*

- * M105, M500, M505, M515, M125, M130
- * Zire, Zire 31, Zire 71, Zire 21, Zire 72
- * Tungsten C, Tungsten W, Tungsten T, Tungsten T2, Tungsten T3, Tungsten E
- Samsung: SPH-I330, SPH-i500
- Kyocera: 6035, 7135
- Sony CLIE: S300, T-615, S360, TG50, N770C/E, NX70V, NX60, SJ30, SL10, SJ33, SJ20, TJ27
- Symbol: SPT 1500, 1550 e 1700
- *Aparelhos com sistema operacional Windows CE/PocketPC:*
 - Acer: n10
 - Casio: Cassiopeia
 - Compaq: iPaq 3670, iPaq H3970, iPaq 3900, iPaq 2210, iPaq 1910, iPaq 1915, iPaq 1930, Pocket PC Aero 1550, iPaq h1940/1945
 - Dell: Axim A5, Axim X3i, Axim X30
 - Fujitsu-Siemens: LOOX 720
 - HHP: Dolphin 9550 (ARM XScale PXA255) Pocket PC 4.20.0
 - HP: Jornada 540 (SH3), Jornada 680/690
 - HTC: Falcon, Harrier
 - O2: XDA II
 - Samsung: Nexio Wireless Hand PC (S160)
 - Siemens: SIMpad SL4
 - Symbol: PDT8100 (Pocket PC 3.0), PDT8146 (Pocket PC 2002)
 - Toshiba: e350 Intel PXA
 - Vandem: Clio (HPC 2.11)
 - ViewSonic: V35, V37
 - Yakumo: 300 GPS
 - Pision: WorkAbout Pro
- *Plataforma Windows 32 bits que funcionam os emuladores:*
 - Windows 98

- *Windows 2000*
- *Windows XP*
- *Plataforma Linux que funcionam os emuladores:*
 - *WINE - Windows Emulator*
- *Plataforma Symbian que funcionam os emuladores:*
 - *P800*

Ainda segundo informações do site do *SuperWaba*, até a versão 4.1 da plataforma, os seguintes PDAs não a suportam:

- *Visor com PalmOS 3.1H1.*
- *Sony NR70 Japanese Version*

A tabela 4.1 apresenta alguns dos *PDAs* compatíveis com o sistema desenvolvido na plataforma *SuperWaba*, com seus respectivos preços.

Os testes realizados durante o desenvolvimento dos sistemas foram feitos de forma separada para cada um dos mesmos, procurando identificar possíveis falhas de implementação. Para isso os principais testes realizados foram:

- Testes de conexão com banco de dados (Sistema do COPOM e Sistema da Viatura).
- Testes para verificar a corretude de comandos SQL implementados na forma de *Stored Procedures*, incluindo consultas, inserções e deleções.
- Testes de Lógica, procurando certificar se a lógica do sistema estava sendo obedecida. Ex: No Sistema do COPOM, ao realizar o acionamento, certificar que as viaturas disponíveis estão conectadas ao sistema, impedindo a ocorrência de acionamentos inválidos;
- Testes de Interface, procurando identificar a melhor forma de interação com o usuário, facilitando o uso da ferramenta. Esses testes foram importantes principalmente na implementação da interface do Sistema da Viatura, pela existência de limitações do próprio *PDA*.

| PDA | Sistema Operacional | Preço |
|---|--|---------------|
| <i>Palm Tungsten C (Wi-Fi integrado)</i> | <i>PalmOS 5.2.1</i> | 1864 reais |
| <i>HP Jornada 560</i> | <i>Microsoft Pocket PC</i> | 1800 reais |
| <i>Casio E-125</i> | <i>Microsoft Pocket PC</i> | 1399 reais |
| <i>Compaq Ipaq H3650</i> | <i>Microsoft Pocket PC</i> | 1499 reais |
| <i>Compaq Ipaq H3670</i> | <i>Microsoft Pocket PC</i> | 1799 reais |
| <i>Palm Tungsten T5</i> | <i>PalmOS 5.4</i> | 1736 reais |
| <i>HP IPAQ rx3715 Mobile Media (Wi-Fi integrado)</i> | <i>Microsoft Windows Mobile 2003</i> | 1931 reais |
| <i>HP IPAQ rx3115 Mobile Media (Wi-Fi integrado)</i> | <i>Microsoft Windows Mobile 2003</i> | 1499 reais |
| <i>HP IPAQ hx2415 Pocket PC (Wi-Fi integrado)</i> | <i>Microsoft Windows Mobile 2003</i> | 1349,97 reais |
| <i>HP IPAQ h4355 Pocket PC (Wi-Fi integrado + teclado acoplado)</i> | <i>Microsoft Windows Mobile 2003 Premium</i> | 1800,97 reais |

Tabela 4.1: PDAs compatíveis com a tecnologia *SuperWaba* e seus preços - Fonte: Pesquisa realizada na internet nos sites oficiais dos vendedores [PALM (2004)], [HP (2004)], [CASIO (2004)]

Após o desenvolvimento do sistema foram realizados outros testes visando certificar seu real funcionamento em uma rede *wireless*. Esses testes foram realizados no *CIN-UFLA* (Centro de Informática da Universidade Federal de Lavras), em uma ambiente de rede *Wi-Fi* do padrão *IEEE 802.11* de redes *wireless*. Para a realização destes testes foram utilizados os seguintes equipamentos:

- Um computador *Notebook* para o funcionamento do Sistema do COPOM.
- Um *PDA* para o funcionamento do Sistema da Viatura. Neste caso foi utilizado o *Palm Tungsten C*.
- Um ponto de acesso sem fios (*Access Point*) utilizado para permitir a conexão do *PDA* à rede *wireless* eficientemente (figura 4.53). Neste caso, não foi necessário a utilização de um adaptador *Wi-Fi*, pois o *PDA* utilizado possui um cartão para conexão *wireless* embutido.
- Um computador *desktop* localizado no Núcleo de Desenvolvimento de Pro-

jetos da 6 RPM, configurado na rede da *UFLANet* para o funcionamento do Sistema Servidor.



Figura 4.53: Ponto de acesso sem fios G - Fonte: *Linksys* [LINKSYS (2004)]

Para a implantação do sistema será necessário a utilização dos equipamentos definidos acima, além de uma antena direcional em cada viatura para aumentar o alcance do sinal, visando uma transmissão de dados mais eficiente.

O ponto de acesso sem fios a ser utilizado possui duas antenas rosqueadas para permitir a conexão do mesmo à antena direcional da viatura. A instalação do ponto de acesso sem fios é feita diretamente na viatura, utilizando-se um conector que fornece ao aparelho a energia necessária para seu funcionamento, a partir da bateria do carro policial.

Com a realização dos testes posteriores ao desenvolvimento, que incluíram testes de funcionamento do sistema integrado, simulando o acionamento de viaturas e uma rápida troca de mensagens entre os sistemas, foram obtidos bons resultados mostrando uma grande eficiência de comunicação entre os mesmos, porém a velocidade de execução foi prejudicada devido ao grande número de operações realizadas implicitamente durante a execução do programa e as limitações de recursos de memória e processamento do dispositivo móvel.

Capítulo 5

Conclusões

Este capítulo apresenta as principais conclusões do trabalho. A seção 5.1 apresenta as principais dificuldades encontradas e os principais conhecimentos adquiridos durante o desenvolvimento do projeto. A seção 5.2 apresenta algumas propostas para trabalhos futuros.

5.1 Alcance dos Objetivos e Dificuldades

O desenvolvimento de aplicativos para dispositivos móveis foi considerada uma tarefa árdua e complexa, uma vez que neste tipo de desenvolvimento existem muitas restrições impostas pelas limitações do próprio aparelho e pelo ambiente de redes móveis em que o mesmo se encontra. No caso do sistema desenvolvido, algumas restrições da plataforma adotada também dificultaram o desenvolvimento. As principais dificuldades encontradas foram:

- Uso restrito de recursos de memória e processamento, prejudicando a implementação da execução concorrente de algumas tarefas do sistema de maneira mais eficiente. Neste caso, a inexistência de *threads* preemptivas¹ na plataforma *SuperWaba* contribuiu para a perda de velocidade e eficiência no processamento do sistema do *PDA*.
- Falta de uma *API* para acesso a um *SGBD* por parte da plataforma *SuperWaba*, resultando na implementação de um terceiro sistema (Sistema

¹*Threads* são linhas de processamento criadas para diferentes tarefas que devem ser executadas de maneira concorrente em um sistema. *Threads* preemptivas realizam alternância entre tarefas concorrentes, simulando o paralelismo na execução das mesmas

Servidor), que realizasse essa tarefa.

- As limitações referentes à dimensão da tela do dispositivo móvel dificultaram a criação de uma interface amigável para o sistema do *PDA*.

Apesar das dificuldades, muitas foram as vantagens oferecidas pelas plataformas adotadas para o desenvolvimento do projeto:

- Pelo fato dos sistemas terem sido desenvolvidos em plataformas livres (*Java* e *SuperWaba*), muitas dúvidas puderam ser resolvidas através de listas de discussão com outros desenvolvedores.
- O desenvolvimento também foi facilitado pela extensa documentação oferecida pelas plataformas *Java* e *SuperWaba*, que foram muito úteis durante toda a implementação dos sistemas, especialmente na implementação de estruturas gráficas.
- Pela característica multiplataforma da tecnologia *SuperWaba*, foi possível a utilização de um *PDA* com um determinado sistema operacional durante todo o desenvolvimento do sistema e a utilização deste em outros *PDA*s com sistemas operacionais diferentes sem a necessidade de mudanças bruscas.
- A implementação de métodos de criptografia foi facilitada pela existência de estruturas próprias para essa aplicação nas plataformas adotadas (*javax.crypto*, *javax.security*).
- A existência de um padrão para desenvolvimento de aplicações servidoras (*J2EE*) na plataforma *Java* possibilitou a criação do Sistema Servidor também nessa plataforma, padronizando assim a utilização da tecnologia *Java* em todo o projeto.

5.2 Propostas de trabalhos futuros

O objetivo básico deste projeto foi a criação de um sistema de comunicação de maior segurança entre COPOM e viaturas policiais militares, possibilitando a troca e o registro de informações referentes ao atendimento de ocorrências, tornando possível a geração de relatórios. Com o objetivo de conferir uma melhoria ao sistema desenvolvido, algumas propostas para trabalhos futuros são sugeridas:

- Implementação do método de criptografia por Chave Pública, visando uma maior segurança na transmissão de dados no sistema.
- Integração do sistema SisComPM com o sistema *Web* da Polícia Militar (Portal da Sexta Região de Polícia Militar), possibilitando a visualização de relatórios detalhados sobre as ocorrências atendidas.
- Integração do sistema SisComPM com o projeto de GeoProcessamento possibilitando um monitoramento das viaturas em tempo real. Esta integração faria com que os policiais deixassem de cadastrar o local de partida da viatura, simplificando o sistema SisComPM. Além da integração com o sistema implementado neste trabalho, isso possibilitaria uma indicação do melhor caminho a ser percorrido pela viatura, até o local de atendimento, através do algoritmo do caminho mínimo em grafos de Dijkstra [CARVALHO (2003)].
- Implantação de Câmeras de Vídeo, visando a captura de imagens pelo policial militar de serviço na viatura, e o envio dessas imagens a um banco de dados de pessoas presas, para análise, comparação e pesquisa.

Referências Bibliográficas

- [ALEXANDRONI (2001a)] Alexandroni, M., *Introdução ao NSBasic.*, URL: <http://www.clubepalm.com.br>, Artigo pesquisado em 25 de Abril de 2001.
- [ALEXANDRONI (2001b)] Alexandroni, M., *Tutorial de Programação em C, Parte 1, Introdução.*, URL: <http://www.clubepalm.com.br>, Tutorial pesquisado em 16 de Outubro de 2001.
- [ALEXANDRONI (2003c)] Alexandroni, M., *PocketStudio: Poder, Produtividade e Facilidade de Uso.*, URL: <http://www.clubepalm.com.br>, Artigo pesquisado em 15 de Março de 2003.
- [ALVES (2002)] Alves, S.T.; *Sistema para configuração e análise DE Redes wireless para clientes em ambientes Linux*, Monografia de Conclusão do Curso de Bacharelado em Ciência da Computação - Universidade Federal de Lavras, 2002.
- [ARC (2004)] ARC Organisers *Palm Tungsten C.*, URL: <http://www.arc.be/docs/fiches/organisers/nl/palm/palmtungstenc.htm>
- [ASSIS (2003)] Assis, W.M. *Avaliação da tecnologia J2ME no contexto de desenvolvimento de jogos multiplayer para celulares* Monografia de Conclusão do Curso de Bacharelado em Ciência da Computação - Universidade Federal de Lavras, 2002.
- [BERNSTEIN (2001)] Bernstein, Ricardo Gruber., *Iniciando o Desenvolvimento na Plataforma PalmOS.*, URL: <http://www.clubepalm.com.br>, Artigo pesquisado em 20 de Janeiro de 2001.

- [BLUETOOTH (2004)] *Bluetooth.org - The Official Bluetooth Membership Site.*, URL: <http://www.bluetooth.org>, Pesquisado em 05 de Junho de 2004.
- [BYOUS (2002)] Byous, Jon., *Java Technology: an early history.*, URL: <http://java.sun.com/features/1998/05/birthday.html>, Artigo pesquisado em 07 de Junho de 2002.
- [CAMPIONE (2002)] Campione, Mary; Walrath, Kathy., *About the Java Technology.*, URL: <http://developer.java.sun.com/developer/onlineTraining/new2java/overview.html>, Pesquisado 13 de Maio de 2002.
- [CARVALHO (2003)] Carvalho, Bruno M. P. S. *Algoritmo de Dijkstra*, Departamento de Engenharia Informática Universidade de Coimbra, Portugal, 2003.
- [CASIO (2004)] Casio Net, URL: www.casionet.com, Pesquisado em 21 de Dezembro de 2004.
- [CES (2003)] CES Communications LTDA., *Security Risks of Radio Communications.*, URL: <http://www.cescomm.co.nz/about/radiosecurity.html>, Pesquisado em 21 de Março de 2004.
- [CODEWARRIOR (2003)] CodeWarrior, Metrowerks., URL: <http://www.metrowerks.com/>, Pesquisado em 05 de Maio de 2003.
- [DEITEL (2002)] Deitel e Deitel., *Internet and Mobile Business: How To Program, Cap 6 Security.*, 2002.
- [ERWIN (2004)] *AllFusion ERwin Data Modeler* , URL: <http://www.ca.com/>, Pesquisado em 20 de Dezembro de 2004.
- [FALCH (2003)] Falch.net., URL: <http://www.falch.net/>, Pesquisado em 05 de Maio de 2003.
- [GARCIA (2001)] Garcia, A., *Revista eweek*, 26 de Março de 2001., *Wep remains vulnerable.*, 2001.

- [GOSLING (1996)] Gosling, James; McGilton, Henry., *The Java Language Environment. A White Paper.*, URL: <http://java.sun.com/docs/white/langenv/index.html>, Artigo pesquisado em Maio de 2002.
- [HAZAN (2002)] Hazan, Guilherme C., *SuperWaba , Uma máquina Virtual brasileira.*, URL: <http://www.clubepalm.com.br>, Artigo pesquisado em 23 de Abril de 2002.
- [HP (2004)] HP Online Shopping, URL: www.hpshopping.com, Pesquisado em 21 de Dezembro de 2004.
- [INFOTECH (2003)] Infotech trends *Worldwide sales of PDAs to reach 35 million units by 2005* URL: www.infotechrends.com/pdaanalysis.htm, Pesquisado em 05 de Março de 2003.
- [INFOSYNC (2004)] Infosync., URL: <http://www.infosync.no/no/news/n/260.html>, 20 de Dezembro de 2004.
- [J2EE (2002)] Java 2 Enterprise Edition., URL: <http://java.sun.com/j2ee/>, 05 de Março de 2002.
- [J2ME (2002)] Java 2 Micro Edition., URL: <http://java.sun.com/j2me/>, 05 de Março de 2002.
- [J2SE (2002)] Java 2 Standard Edition., URL: <http://java.sun.com/j2se/>, 05 de Março de 2002.
- [LINDHOLM (1999)] Lindholm, Tim; Yellin, Frank., *The Java Virtual Machine Specification. Second Edition.*, 1999.
- [LINKSYS (2004)] Linksys **WAP54G**. URL: <http://http://www.linksys.com>, 20 de Dezembro de 2004.
- [LOUREIRO, et.Al. (2003)] Loureiro, A.A.F.; Sadok D.F.H.; Mateus G.R., Nogueira J.M.S.; Kelner J. *Comunicação sem fio e Computação Móvel: Tecnologias, Desafios e Oportunidades.*, Artigo publicado no Congresso da Sociedade Brasileira da Computação (SBC), em Campinas - SP, 2003.

- [MAIA (2003)] Maia, R.M.F.; **Bluetooth - Promessas de uma nova tecnologia**, *Monografia de Conclusão do Curso de Bacharelado em Sistemas De Informação - Faculdade Integrada do Recife*, 2003.
- [MATEUS, et.Al. (1998)] Mateus G.R; Loureiro, A.A.F., **Introdução à Computação Móvel**; *Livro publicado pelo Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro - URFJ*, 1998.
- [MOBILE (2004)] **Programming Mobile Devices**. URL: <http://daveandal.net/alshed/mobiledevices/mobiledevices.jpg>
Pesquisado em 20 de Dezembro de 2004
- [NELSON (2001)] Nelson, M., *Revista Information Week*, 05 de Fevereiro de 2001., **Untethered doesn't mean insecure.**, URL: <http://www.informationweek.com>.
- [NETBEANS (2004)] NETBeans., URL: <http://www.netbeans.org>, Site oficial da ferramenta - Pesquisado em 20 de Dezembro de 2004.
- [NSBASIC (2003)] NSBasic., URL: <http://www.nsbasic.com>, Site oficial da ferramenta - Pesquisado em 05 de Maio de 2003.
- [PALM (2004)] PalmOne, URL: www.palmone.com, Pesquisado em 21 de Dezembro de 2004.
- [PALMOS (2003)] PalmOS, **PalmSource, Palm OS Tools e Downloads.**, URL: <http://www.palmos.com/dev/tools>, Pesquisado em 05 de Maio de 2003.
- [PDA (2003)] PDA Toolbox., URL: www.pdatoolbox.com, Pesquisado em 05 de Maio de 2003.
- [POCKET (2003)] PocketStudio, Pocket-Technologies., URL: <http://www.pocket-technologies.com/>, Pesquisado em 05 de Maio de 2003.
- [RUSSEL (1997)] Russel, S. F., **Wireless Channel Security Tutorial.**, URL: <http://www.public.iastate.edu/~sfr/wireless/wtut1.html>, Pesquisado em 22 de Março de 2003.

- [SASSE (2001)] Sasse, Erick., *Satellite Forms.*, URL: <http://www.clubepalm.com.br>; Pesquisado em 09 de Fevereiro de 2002.
- [SATELLITE (2003)] Satellite Forms., URL: <http://www.pumatech.com>, Pesquisado em 05 de Maio de 2003.
- [SISCOMP (2004)] *Projeto SisComp - Sistema de Comunicação Segura entre 190 e viaturas PM.*, URL: <http://www.comp.ufla.br/humberto/projetos/siscomp/index.htm>
- [SPANGLER (2000)] Sprangler, T., *Revista Interactive Week*, 10 de Abril de 2000, pags. 28 a 34., *Home is Where the Hack is.*, 2000.
- [SQL (2000)] *The SQL Server Home .*, URL: <http://www.microsoft.com/sql>, Pesquisado em 20 de Dezembro de 2004.
- [SUN (2002a)] Sun Microsystems., *A Brief History of the Green Project.*, URL: <http://java.sun.com/people/jag/green/>, Artigo pesquisado em 28 de Novembro de 2002.
- [SUN (2003b)] Sun Microsystems *Introduction to Wireless Java R Technology*, URL: <http://wireless.java.sun.com/getstart/>, Pesquisado em 04 de Fevereiro de 2003.
- [SW (2004a)] *SuperWaba - Características da Plataforma*, URL: www.superwaba.com.br/pt/vantagens.asp, Pesquisado em 16 de Maio de 2004.
- [SW (2004b)] *SuperWaba - SuperWaba X Outras Plataformas*, URL: <http://www.superwaba.com.br/pt/swxj2me.asp>, Pesquisado em 16 de Maio de 2004.
- [SW (2004c)] *SuperWaba - Develop portable handheld applications* URL: <http://www.superwaba.org/etc/SuperWabaFolderEn.pdf>, Pesquisado em 20 de Maio de 2004.
- [THIOLLENT (1997)] M. São Paulo, Atlas *Pesquisa-Ação nas organizações*, 1997.

- [TOSO et.AI. (2004)] Toso, R. F.; Andrade, C.E.; Nogueira, F.L.B., **Redes sem fio IEEE 802.11.**, URL: <http://www.comp.ufla.br/rtoso/docs/Wireless.pdf>, Pesquisado em 02 de Junho de 2004.
- [US.DEP (2003)] US. Department of State Foreign Affairs., **Hand-Book Volume 5 - HandBook 2 - Telecomunicações - Managing Radio Networks, Communications.**, URL: <http://foia.state.gov/masterdocs/05fah02/tel0710.pdf>, Artigo pesquisado em 14 de Novembro de 2003.
- [UML (2003)] **UML Resource Page** URL: <http://www.uml.org.>, Pesquisado em 20 de Dezembro de 2004.
- [VISIO (2003)] **Microsoft Office Online - Visio Home Page** URL: <http://office.microsoft.com/pt-br/FX010857981046.aspx>, Pesquisado em 20 de Dezembro de 2004.
- [WABA (2003a)] WabaSoft., URL: <http://www.wabasoft.com.>, Pesquisado em 20 de Julho de 2002.
- [WABA (2003b)] **The WabaSoft Development Kit** ., URL: <http://www.wabasoft.com/specsdk.shtml.>, Pesquisado em 20 de Julho de 2002.
- [WABA (2003c)] Sourceforge projeto Waba., URL: <http://waba.sourceforge.net>, Pesquisado em 02 de Maio de 2003.
- [WAP (2000)] Wap Forum., **Wireless Application Protocol White Paper.**,2000.