

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**PROGRAMA DE PÓS-GRADUAÇÃO  
EM CIÊNCIA DA COMPUTAÇÃO**

**LABORATÓRIO DE REDES E GERÊNCIA**

**Mirela Sechi Moretti Annoni Notare**

**CONCEPÇÃO, DESENVOLVIMENTO E ANÁLISE  
DE UM SISTEMA DE GERÊNCIA DE SEGURANÇA  
PARA REDES DE TELECOMUNICAÇÕES**

Tese submetida à  
Universidade Federal de Santa Catarina  
como parte dos requisitos  
para a obtenção do grau de  
Doutor em Ciência da Computação.

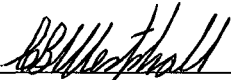
Professor Orientador: Dr. Carlos Becker Westphall

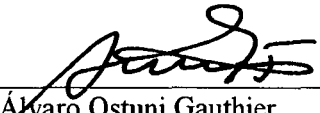
Florianópolis, março de 2000.

# Concepção, Desenvolvimento e Análise de um Sistema de Gerência de Segurança para Redes de Telecomunicações

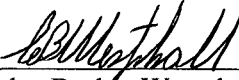
Mirela Sechi Moretti Annoni Notare

Esta Tese foi julgada adequada para a obtenção do Título de Doutor em Ciência da Computação, Área de Concentração *Sistemas de Computação*, e aprovada em sua forma final pelo Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina.

  
Prof. Dr. Carlos Becker Westphall  
Orientador

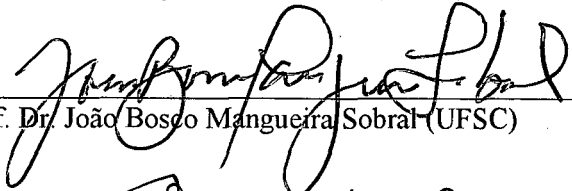
  
Prof. Dr. Fernando Alvaro Ostuni Gauthier  
Coordenador do Programa de Pós-Graduação em Ciência da Computação da UFSC

Banca Examinadora:

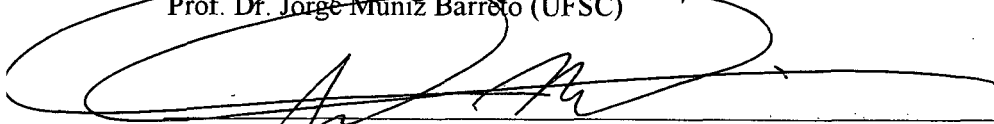
  
Prof. Dr. Carlos Becker Westphall (UFSC) Presidente

  
Prof. Dr. João Bosco da Mota Alves (UFSC) Co-Orientador

  
Prof. Dr. Bernardo Gonçalves Riso (UFSC) Membro

  
Prof. Dr. João Bosco Manguiera Sobral (UFSC) Membro

  
Prof. Dr. Jorge Muniz Barreto (UFSC) Membro

  
Prof. Dr. Roberto Sérgio Barbosa Martins (UFPB) Membro

  
Prof. Dr. Otto Carlos Muniz Bandeira Duarte (UFRJ) Membro

A Deus,  
que em cada momento  
cuida de mim  
com sua infinita sabedoria.

## Agradecimentos

Reconheço a importância de algumas contribuições de pessoas às quais sou muito grata:

Azzedine Boukerche, da UNT (*University North of Texas*), EUA – pela disponibilidade na revisão técnica e na revisão da Língua Inglesa; e pela co-autoria em capítulo de livro;

Alexander Nikolaevich Firsov, da Mercedes Telecom, Moscou – pelas informações sobre clonagem na telefonia móvel;

Hubert Garavel, do INRIA (*Institut National de Recherche en Informatique et en Automatique*), França – pela parceria bilateral internacional; pelas referências em Relatórios de Atividades e em Casos de Estudo do INRIA; e pela receptividade na visita ao INRIA/Grenoble;

Luís Fernando Kormann, da UFSC/LRG – pela parceria nas atividades organizacionais e de pesquisa em projetos e eventos científicos; pelas traduções da Língua Inglesa para a Língua Portuguesa; e pelas instalações Unix;

José Renato de Faria, da Ponto Final Revisões – pela dedicação e profissionalismo na revisão da Língua Portuguesa;

Kathia Jucá, da UFSC/NPD – pelas instalações e configurações em ambiente Solaris;

José Leomar Todesco, da UFSC/EPS – pelas discussões matemáticas;

Ricardo Costa Rodrigues, da UFMG/DCC – pela apresentação de trabalho internacional;

Arno Holz e Alexandro Castyro Jesus, da Telefônica Celular/RS – pela parceria e disponibilidade de base de dados;

Hermida, Walter, Spíndola, Fernanda, Karla, Castello, Milioli, Borges, Désiré, Laura, Alexandre e Cruz, da UFSC/LRG – pela escolha do SSTCC como objeto das pesquisas;

Westphall e Alves – pela orientação e co-orientação;

Riso, Sobral, Barreto, Joberto, Otto e Gauthier – pela participação na seção de defesa desta tese, como membros da banca examinadora e como moderador; e

Itamar Annoni Notare – pelo incentivo e coragem transmitida para a realização dessa tese.

# Sumário

LISTA DE FIGURAS.....	VII
LISTA DE TABELAS.....	IX
LISTA DE ABREVIATURAS .....	X
RESUMO .....	XIII
ABSTRACT .....	XIV
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1 FRAUDES DE CLONAGEM E DE HABILITAÇÃO EM TELEFONIA MÓVEL.....	5
1.2 TRABALHOS RELACIONADOS.....	6
1.3 ESCOPO DA TESE – PROPOSTA E PRINCIPAIS CONTRIBUIÇÕES .....	9
1.4 ORGANIZAÇÃO DESTE TRABALHO .....	13
<b>2 ESPECIFICAÇÃO E VALIDAÇÃO FORMAL (ISO 8807).....</b>	<b>14</b>
2.1 A TÉCNICA DE DESCRIÇÃO FORMAL LOTOS.....	15
2.1.1 <i>Processos LOTOS stop, exit e Infinitos</i> .....	17
2.1.2 <i>Operadores LOTOS</i> .....	17
2.1.3 <i>Modelo Semântico</i> .....	19
2.2 ESPECIFICAÇÃO DE SERVIÇO DO SSTCC.....	20
2.3 ESPECIFICAÇÕES DE PROTOCOLO DO SSTCC .....	22
2.3.1 <i>Especificação do Processo SsccClone</i> .....	23
2.3.2 <i>Especificação do Processo SetwebBill</i> .....	28
2.3.3 <i>Especificação do Processo SipiImpostor</i> .....	32
2.3.4 <i>Especificação do Sistema SSTCC</i> .....	38
2.4 VALIDAÇÃO FORMAL DE SISTEMAS .....	42
2.4.1 <i>Teste</i> .....	42
2.4.2 <i>Simulação</i> .....	42
2.4.3 <i>Verificação</i> .....	43
2.4.3.1 <i>Equivalência Forte (Strong Bisimulation)</i> .....	43
2.4.3.2 <i>Equivalência Fraca (Weak Bisimulation)</i> .....	45
2.5 EXPERIMENTOS NA VALIDAÇÃO FORMAL DO SISTEMA SSTCC.....	46
2.6 RESULTADOS .....	58
<b>3 DETECÇÃO DE INTRUSÃO COM REDES NEURAI ARTIFICIAIS.....</b>	<b>59</b>
3.1 MODELO NEURAL.....	60
3.2 USO DE CLASSIFICADOR NÃO-SUPERVISIONADO – KOHONEN.....	61
3.2.1 <i>Algoritmo Kohonen</i> .....	61
3.2.2 <i>Experimentos</i> .....	63
3.2.2.1 <i>Definição das Características</i> .....	65
3.2.2.2 <i>Treinamento da Rede</i> .....	67
3.2.2.3 <i>Teste da Rede</i> .....	68
3.3 USO DE CLASSIFICADOR SUPERVISIONADO – RBF .....	72
3.3.1 <i>Algoritmo RBF</i> .....	72
3.3.2 <i>Experimentos Iniciais</i> .....	75
3.3.2.1 <i>Definição das Características e dos Agrupamentos</i> .....	75
3.3.2.2 <i>Treinamento e Teste da Rede</i> .....	75
3.3.2.3 <i>Cálculo do Erro</i> .....	76
3.3.3 <i>Experimentos Avançados</i> .....	78
3.3.3.1 <i>Definição das Características e dos Agrupamentos</i> .....	78
3.3.3.2 <i>Treinamento da Rede</i> .....	80
3.3.3.3 <i>Teste da Rede</i> .....	81
3.3.3.4 <i>Cálculo do Erro</i> .....	82
3.4 RESULTADOS.....	85

<b>4</b>	<b>IMPLEMENTAÇÃO DA GERÊNCIA DE SEGURANÇA DISTRIBUÍDA.....</b>	<b>87</b>
4.1	GERÊNCIA DE SEGURANÇA E REDES DE TELECOMUNICAÇÕES.....	88
4.1.1	<i>Ataques e Mecanismos de Segurança</i> .....	89
4.1.2	<i>Tecnologias e Ferramentas na Gerência de Segurança</i> .....	93
4.1.2.1	Arquitetura CORBA.....	93
4.1.2.2	Linguagem Java.....	94
4.1.3	<i>Rede Telefônica</i> .....	96
4.2	IMPLEMENTAÇÃO DOS SISTEMAS SSCC E SIPI.....	102
4.2.1	<i>Módulos Adaptador e Agente</i> .....	102
4.2.2	<i>Módulo Gerente</i> .....	108
4.2.3	<i>Implementação da Gerência em Ambiente Distribuído CORBA</i> .....	110
4.3	IMPLEMENTAÇÃO DO SISTEMA SETWEB.....	111
4.3.1	<i>Módulos Adaptadores</i> .....	111
4.3.2	<i>Módulo Montador de Conta Telefônica</i> .....	114
4.3.3	<i>Bases de Dados do SETWeb</i> .....	115
4.3.3.1	CDR.....	115
4.3.3.2	Wtelecom.....	115
4.3.3.3	Central Local.....	115
4.3.3.4	Central T2.....	116
4.3.3.5	Central T3.....	116
4.3.3.6	Central Trânsito Embratel.....	117
4.3.4	<i>Implementação da Gerência em Ambiente Web</i> .....	118
4.4	RESULTADOS.....	122
<b>5</b>	<b>CONCLUSÃO E FUTUROS TRABALHOS.....</b>	<b>125</b>
5.1	SUMÁRIO DAS CONTRIBUIÇÕES.....	125
5.2	FUTUROS TRABALHOS.....	127
5.3	OUTRAS CONTRIBUIÇÕES.....	128
5.3.1	<i>Publicações e Apresentações</i> .....	128
5.3.2	<i>Prêmios e Citações</i> .....	136
	<b>ANEXO 1 – ISO 8807.....</b>	<b>139</b>
1.1	Especificação.....	139
1.2	Validação.....	140
	<b>ANEXO 2 – REDES NEURAIS ARTIFICIAIS.....</b>	<b>141</b>
2.1	Kohonen.....	141
2.2	RBF - Função de Base Radial.....	145
	<b>ANEXO 3 – CORBA/JAVA.....</b>	<b>166</b>
3.1	Sistemas SSCC e SIPI.....	166
3.2	Sistema SETWeb.....	183
	<b>ÍNDICE REMISSIVO.....</b>	<b>200</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>201</b>

# Lista de Figuras

FIGURA 1.1 – CLONAGEM DE TELEFONE CELULAR.....	3
FIGURA 1.2 – CONTA TELEFÔNICA DE UM CELULAR QUE FOI CLONADO.....	5
FIGURA 1.3 – FUNCIONALIDADE DO SISTEMA SSTCC.....	12
FIGURA 2.1 – REPRESENTAÇÃO GRÁFICA DO SSTCC.....	20
FIGURA 2.2 – ESPECIFICAÇÃO LOTOS DO SSTCC.....	20
FIGURA 2.3 – REPRESENTAÇÃO GRÁFICA DO PROCESSO SstccSERVICE.....	21
FIGURA 2.4 – O SISTEMA SSTCC E SEUS TRÊS SUBSISTEMAS (SSCC, SETWEB E SIPI).....	22
FIGURA 2.5 – REFINAMENTO DA ESPECIFICAÇÃO SSTCC.....	22
FIGURA 2.6 – DOIS PRINCIPAIS PROCESSOS DO SsccCLONE.....	23
FIGURA 2.7 – PROCESSO LOTOS SsccCLONE.....	23
FIGURA 2.8 – PROCESSO CLONECORBAAGENTSET.....	23
FIGURA 2.9 – PROCESSO CLONECORBAMANAGER.....	24
FIGURA 2.10 – DETALHAMENTO DO PROCESSO CLONECORBAAGENTSSET.....	24
FIGURA 2.11 – DETALHAMENTO DO PROCESSO CLONECORBAAGENTSSETJ.....	25
FIGURA 2.12 – PROCESSO CLONECORBAAGENTSSETJ.....	25
FIGURA 2.13 – REPRESENTAÇÃO DO SSCC DETALHADO COM DOIS AGENTES GERENCIADOS.....	27
FIGURA 2.14 – ESPECIFICAÇÃO LOTOS DO SSCC COM DOIS AGENTES.....	28
FIGURA 2.15 – DOIS PRINCIPAIS PROCESSOS DO SETWEBBILL.....	28
FIGURA 2.16 – ESPECIFICAÇÃO LOTOS DO SETWEBBILL.....	28
FIGURA 2.17 – DETALHAMENTO DO PROCESSO BILLCORBAAGENTSSET.....	29
FIGURA 2.18 – DETALHAMENTO DO PROCESSO BILLCORBAAGENTSSETJ.....	30
FIGURA 2.19 – ESPECIFICAÇÃO LOTOS DO PROCESSO BILLCORBAAGENTSSETJ.....	30
FIGURA 2.20 – REPRESENTAÇÃO DO SETWEB DETALHADO COM DOIS AGENTES GERENCIADOS.....	31
FIGURA 2.21 – ESPECIFICAÇÃO LOTOS DO SETWEB COM DOIS AGENTES DISTRIBUÍDOS.....	32
FIGURA 2.22 – DOIS PROCESSOS DO SIPIIMPOSTOR.....	33
FIGURA 2.23 – ESPECIFICAÇÃO LOTOS DO SIPIIMPOSTOR.....	33
FIGURA 2.24 – DETALHAMENTO DO PROCESSO IMPOSTORCORBAAGENTSSET.....	34
FIGURA 2.25 – DETALHAMENTO DO PROCESSO IMPOSTORSORBAAGENTSSETJ.....	35
FIGURA 2.26 – ESPECIFICAÇÃO LOTOS DO PROCESSO IMPOSTORCORBAAGENTSSETJ.....	35
FIGURA 2.27 – REPRESENTAÇÃO DO SIPI DETALHADO COM DOIS AGENTES GERENCIADOS.....	37
FIGURA 2.28 – ESPECIFICAÇÃO LOTOS REFINADA DO SUBSISTEMA SIPI.....	38
FIGURA 2.29 – ARQUITETURA GERAL DO SSTCC.....	39
FIGURA 2.30 – ESPECIFICAÇÃO LOTOS REFINADA DO SISTEMA SSTCC.....	41
FIGURA 2.31 – SISTEMAS NÃO EQUIVALENTES QUANTO À EQUIVALÊNCIA FORTE.....	44
FIGURA 2.32 – SISTEMAS EQUIVALENTES QUANTO À EQUIVALÊNCIA DE OBSERVAÇÃO.....	45
FIGURA 2.33 – GERAÇÃO DO LTS.....	47
FIGURA 2.34 – ESCOLHA NA GERAÇÃO DO LTS.....	47
FIGURA 2.35 – LTS CORRESPONDENTE À ESPECIFICAÇÃO LOTOS SstccPROTOCOL.....	48
FIGURA 2.36 – IMPASSE (DEADLOCK) DETECTADO DURANTE A GERAÇÃO DO LTS.....	49
FIGURA 2.37 – GERAÇÃO DO LTS EM FORMATO ALDEBARAN.....	50
FIGURA 2.38 – SstccPROTOCOL.AUT – LTS NO FORMATO ALDEBARAN.....	50
FIGURA 2.39 – LTS EM 2 DIMENSÕES.....	51
FIGURA 2.40 – LTS COM MUITAS TRANSIÇÕES.....	51
FIGURA 2.41 – REDUÇÃO DO LTS.....	52
FIGURA 2.42 – LTS REDUZIDO POR EQUIVALÊNCIA FORTE.....	52
FIGURA 2.43 – GERAÇÃO DO GRAFO SstccSERVICE.....	53
FIGURA 2.44 – RELAÇÃO DOS ARQUIVOS GERADOS CONTENDO OS RESPECTIVOS TAMANHOS.....	53
FIGURA 2.45 – GRAFO SstccSERVICE NO FORMATO ALDEBARAN.....	54
FIGURA 2.46 – SstccSERVICE LTS.....	54
FIGURA 2.47 – OPÇÃO DE COMPARAÇÃO.....	55
FIGURA 2.48 – RESULTADO DA EQUIVALÊNCIA QUANTO À OBSERVAÇÃO.....	55
FIGURA 2.49 – PROVA FORMAL DE CORREÇÃO DO SISTEMA SSTCC.....	56

FIGURA 3.1 – AMOSTRA DOS PRIMEIROS 10 VETORES DE 8.830 CONSIDERADOS.....	66
FIGURA 3.2 – GERAÇÃO DE AGRUPAMENTOS PELO TREINAMENTO DA REDE (ITERAÇÕES) .....	67
FIGURA 3.3 – LIGAÇÕES TELEFÔNICAS AGRUPADAS. ....	68
FIGURA 3.4 – TESTE DO CLASSIFICADOR. ....	68
FIGURA 3.5 – PESOS UTILIZADOS (5 CARACTERÍSTICAS X 4 PONTOS LINEARES DE SAÍDA). ....	69
FIGURA 3.6 – COMPARAÇÃO DOS RESULTADOS DO EXPERIMENTO (SENSIBILIDADE). ....	70
FIGURA 3.7 – ECONOMIA DA OPERADORA UTILIZANDO O ALGORITMO KOHONEN. ....	71
FIGURA 3.8 – FUNÇÃO DE BASE RADIAL. ....	72
FIGURA 3.9 – ALGORITMO K-MEANS. ....	74
FIGURA 3.10 – CLASSIFICAÇÃO DOS USUÁRIOS. ....	75
FIGURA 3.11 – TAXA DE ERRO DO ALGORITMO. ....	76
FIGURA 3.12A – REDUÇÃO DAS PERDAS COM O USO DA RBF (EM REAIS, BRASIL 1997). ....	77
FIGURA 3.12B – REDUÇÃO DAS PERDAS COM O USO DA RBF (EM DÓLARES, EUA 1998). ....	77
FIGURA 3.13 – CLASSIFICAÇÃO DOS USUÁRIOS EM 10 GRUPOS HIERÁRQUICOS. ....	79
FIGURA 3.14 – MATRIZ $A_{2400,4}$ UTILIZADA PARA A GERAÇÃO DOS CENTROS DOS AGRUPAMENTOS. ....	80
FIGURA 3.15 – MATRIZ $B_{2400,10}$ UTILIZADA PARA TREINAR A REDE.....	81
FIGURA 3.16 – TESTE DA REDE. ....	82
FIGURA 3.17 – EXECUÇÃO ON-LINE E CÁLCULO DO ERRO.....	83
FIGURA 3.18 – FREQUÊNCIA DAS CARACTERÍSTICAS DURAÇÃO, DESTINO E DIA.....	84
FIGURA 4.1 – MONITOR DE REFERÊNCIA.....	88
FIGURA 4.2 – ATAQUES NA SEGURANÇA.....	90
FIGURA 4.3 – EXEMPLO DE UM USUÁRIO EFETUANDO DIVERSOS TIPOS DE LIGAÇÕES.....	99
FIGURA 4.4 – VETOR DE SAÍDA DA REDE NEURAL IDENTIFICANDO O PADRÃO DO USUÁRIO. ....	106
FIGURA 4.5 – TELAS DO AGENTE.....	107
FIGURA 4.6 – JANELA PRINCIPAL.....	108
FIGURA 4.7 – OPÇÃO VIEW MENU.....	109
FIGURA 4.8 – JANELA CONFIG.....	109
FIGURA 4.9 – ARQUIVO DE CHAMADAS ABERTO.....	110
FIGURA 4.10 – PÁGINA INICIAL DO SETWEB.....	114
FIGURA 4.11 – PÁGINA COM A CONTA DISCRIMINADA.....	114
FIGURA 4.12 – BASES DE DADOS LOCALIZADAS NA “MATRIZ” DA EMPRESA TELEFÔNICA.....	115
FIGURA 4.13 – BASE DE DADOS DAS CENTRAIS LOCAIS.....	115
FIGURA 4.14 – BASE DE DADOS DAS CENTRAIS T2.....	116
FIGURA 4.15 – BASES DE DADOS DAS CENTRAIS T3.....	117
FIGURA 4.16 – BASES DE DADOS DE UMA CENTRAL TRÂNSITO EMBRATEL.....	117
FIGURA 4.17 – PERMISSÕES CONCEDIDAS A <i>APPLETS</i> .....	119
FIGURA 4.18 – CADASTRO DE USUÁRIO NO SISTEMA DE EXTRATO TELEFÔNICO VIA WEB.....	120



## Lista de Tabelas

TABELA 2.1 – NÍVEIS DE SEGURANÇA – DEPARTAMENTO DE DEFESA DOS EUA.....	57
TABELA 3.1 – LIGAÇÕES TELEFÔNICAS CLASSIFICADAS POR HORÁRIOS/DIAS.....	66
TABELA 3.2 – NÚMERO DE NEURÔNIOS NA CAMADA ESCONDIDA E RESPECTIVA TAXA DE ERRO.....	76
TABELA 4.1 – ESTABELECIMENTO DE UMA POLÍTICA DE SEGURANÇA.....	91
TABELA 4.2 – EXEMPLO DE TIPOS DE LIGAÇÕES POSSÍVEIS PARA A CIDADE DE FLORIANÓPOLIS.....	98
TABELA 4.3 – DEGRAUS DE TARIFAS INTERNACIONAIS.....	101
TABELA 4.4 – DEGRAUS DE TARIFAS NACIONAIS.....	101
TABELA 4.5 – FERRAMENTAS UTILIZADAS PARA PROVER SEGURANÇA NO SETWEB.....	118

## Lista de Abreviaturas

<i>AMPS</i>	<i>Advanced Mobile Phone System</i>
<i>ANN</i>	<i>Artificial Neural Networks</i>
<i>API</i>	<i>Application Program Interface</i>
<i>ASCII</i>	<i>American National Standard Code for Information Interchange</i>
<i>BD</i>	<i>Banco de Datos</i>
<i>BCG</i>	<i>Binary-Coded Graphs</i>
<i>BOA</i>	<i>Basic Object Adapter</i>
<i>CA</i>	<i>Certification Authority</i>
<i>CADP</i>	<i>Caesar/Aldebaran Development Package</i>
<i>CCA</i>	<i>Carregador de Classes de Applets</i>
<i>CDMA</i>	<i>Code Division Multiple Access</i>
<i>CDR</i>	<i>Call Detailed Register</i>
<i>CDSA</i>	<i>Common Data Security Architecture</i>
<i>CERN</i>	<i>Centre European Research Nuclear</i>
<i>CGI</i>	<i>Common Gateway Interface</i>
<i>COM</i>	<i>Component Object Model</i>
<i>CORBA</i>	<i>Common Object Request Broker Architecture</i>
<i>COSS</i>	<i>Common Object Services Specification</i>
<i>CSP</i>	<i>Criptographic Service Provider</i>
<i>D-AMPS</i>	<i>Digital - Advanced Mobile Phone System</i>
<i>DAS</i>	<i>Digital Signature Algorithm</i>
<i>DES</i>	<i>Data Encryption Standard</i>
<i>DII</i>	<i>Dynamic Invocation Interface</i>
<i>DLL</i>	<i>Dynamic Link Library</i>
<i>DoD - USA</i>	<i>Department of Defense of the United States of America</i>
<i>DSA</i>	<i>Digital Signature Algorithm</i>
<i>ESN</i>	<i>Electronic Serial Number</i>
<i>FBI</i>	<i>Federal Bureau of Investigation</i>
<i>FDT</i>	<i>Formal Description Technique</i>

<i>FPF</i>	<i>Fraud Protection Feature</i>
<i>GIOP</i>	<i>General Inter-ORB Protocol</i>
<i>GSM</i>	<i>Global Systems for Mobile Communications</i>
<i>GSS-API</i>	<i>Generic Security Services API</i>
<i>GUI</i>	<i>Graphic User Interface</i>
<i>HTML</i>	<i>HyperText Markup Language</i>
<i>HTTP</i>	<i>HyperText Transfer Protocol</i>
<i>IDAPI</i>	<i>Integrated Database Application Program Interface</i>
<i>IDC</i>	<i>International Data Corporation</i>
<i>IDL</i>	<i>Interface Definition Language</i>
<i>IIOP</i>	<i>Internet Inter-ORB Protocol</i>
<i>IOR</i>	<i>Interoperable Object Reference</i>
<i>IMAP</i>	<i>Internet Message Access Protocol</i>
<i>IMT 2000</i>	<i>International Mobile Telecommunications 2000</i>
<i>IS-54/95/136</i>	<i>Interim Standard-54 / Interim Standard-95 / Interim Standard-136</i>
<i>ISAPI</i>	<i>Internet Server Application Program Interface</i>
<i>ISO</i>	<i>International Organization for Standardization</i>
<i>ITU-TS</i>	<i>International Telecommunications Union – Telecommunication Standardization Sector</i>
<i>JDK</i>	<i>Java Developer Kit</i>
<i>JRE</i>	<i>Java Runtime Environment</i>
<i>JVM</i>	<i>Java Virtual Machine</i>
<i>LOTOS</i>	<i>Language of Temporal Ordering Specification</i>
<i>LTS</i>	<i>Labelled Transition Systems</i>
<i>MCC</i>	<i>MatLab Compiler Command</i>
<i>MIN</i>	<i>Mobile Identification Number</i>
<i>MVJ</i>	<i>Máquina Virtual Java</i>
<i>NSA</i>	<i>National Security Agency</i>
<i>OA</i>	<i>Object Adapter</i>
<i>ODBC</i>	<i>Open DataBase Connectivity</i>
<i>ODP</i>	<i>Open Distributed Processing</i>
<i>OMG</i>	<i>Object Management Group</i>
<i>ORB</i>	<i>Object Request Broker</i>

<i>OSC</i>	<i>Object Security Controller</i>
<i>OSI</i>	<i>Open Systems Interconnection</i>
<i>PCS</i>	<i>Personal Communications Service</i>
<i>PGP</i>	<i>Pretty Good Privacy</i>
<i>PIN</i>	<i>Personal Identification Number</i>
<i>RBF</i>	<i>Radial Basis Function</i>
<i>RMI</i>	<i>Remote Method Invocation</i>
<i>RNA</i>	Redes Neurais Artificiais
<i>SET</i>	<i>Secure Electronic Transaction</i>
<i>SETWeb</i>	Sistema de Extrato Telefônico On-line via Web
<i>SGBD</i>	Sistema Gerenciador de Banco de Dados
<i>SHTTP</i>	<i>Secure Hypertext Transfer Protocol</i>
<i>SIPI</i>	Sistema de Identificação de Prováveis Inadimplentes
<i>TLS</i>	<i>Transport Layer Security</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>SRM</i>	<i>Security Reference Model</i>
<i>SSCC</i>	Sistema de Segurança contra Clonagem de Celulares
<i>SSH</i>	<i>Secure Shell</i>
<i>SSL</i>	<i>Secure Socket Layer</i>
<i>SSTCC®</i>	Sistema de Segurança para Telecomunicações Contra Clonagem e Inadimplência
<i>TCP/IP</i>	<i>Transmission Control Protocol/Internet Protocol</i>
<i>TCSEC</i>	<i>Trusted Computer System Evaluation Criteria</i>
<i>TDF</i>	Técnica de Descrição Formal
<i>TDMA</i>	<i>Time Division Multiple Access</i>
<i>TMN</i>	<i>Telecommunication Management Networks</i>
<i>TPI</i>	<i>Trust Policy Interface</i>
<i>TPM</i>	<i>Trust Policy Module</i>
<i>UMTS</i>	<i>Universal Mobile Telecommunication System</i>
<i>URL</i>	<i>Uniform Resource Locator</i>
<i>VBC</i>	Verificador de <i>ByteCode</i>
<i>VLSI</i>	<i>Very Large Scale Integrated</i>
<i>WWW</i>	<i>World Wide Web</i>

Resumo da tese apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciência da Computação

## **Concepção, Desenvolvimento e Análise de um Sistema de Gerência de Segurança para Redes de Telecomunicações**

**Mirela Sechi Moretti Annoni Notare**

Março/2000

Orientador: Carlos Becker Westphall

Área de Concentração: Sistemas de Computação

Palavras-chave: Sistemas Distribuídos, Gerência de Segurança, Telecomunicações, LOTOS, Redes Neurais, CORBA

Número de Páginas: 207

### **RESUMO**

Gerência de Segurança contra fraudes e intrusões será um dos principais tópicos de investigação nas próximas gerações de sistemas distribuídos. Um sistema seguro (*secure*) provê proteção contra ataques de usuários não confiáveis, enquanto um sistema correto (*safe*) provê proteção contra erros de usuários confiáveis. Esta tese propõe um sistema de gerenciamento seguro e correto, em sistemas distribuídos em geral, e em sistemas de comunicação sem fio em particular. Inicialmente, apresenta-se a especificação e validação formal, em LOTOS, do sistema distribuído de segurança para provar sua correção. Então, descreve-se como redes neurais podem ser empregadas na gerência de segurança de redes de telecomunicações sem fio, principalmente contra fraudes de clonagem e habilitação. O emprego de redes neurais possibilita o reconhecimento do padrão de cada usuário de telecomunicação móvel, e por conseguinte, detectar intrusões. Os resultados indicam que com o emprego de redes neurais na detecção de intrusão em redes de telecomunicações é possível reduzir, significativamente, as perdas das companhias telefônicas e usuários. Em acréscimo, apresenta-se como a arquitetura CORBA pode ser usada para suportar e elevar a segurança do sistema. Na implementação realizada, o sistema garante controle de acesso, autenticação, confidencialidade, integridade, disponibilidade e não-repúdio. Finalmente, como uma derivação dessa pesquisa, foi desenvolvido um sistema seguro de extrato telefônico via Web, possibilitando que os próprios usuários detectem intrusões e minimizem seus prejuízos.

*Abstract of the thesis presented to UFSC as part of the requirements  
to obtain the degree of Doctor in Computer Science*

***Conception, Development and Analysis of a Security  
Management System for Telecommunication Networks***

***Mirela Sechi Moretti Annoni Notare***

*March/2000*

*Advisor: Carlos Becker Westphall*

*Area of Concentration: Computer Systems*

*Keywords: Distributed Systems, Security Management, Telecommunications,  
LOTOS, Neural Networks, CORBA*

*Number of pages: 207*

## **ABSTRACT**

*Safety and security management against frauds and intrusions will be one of the major issues in the next generation of distributed systems. A secure system provides protection against attacks of untrusted users, while a safe system provides protection against errors of trusted users. This thesis proposes a secure and safe management in distributed systems in general, and wireless communication systems in particular. Initially, a formal specification and validation, in LOTOS, of the distributed security system is presented in order to prove its correctness. Then, it is described how neural networks can be employed in the security management of wireless networks mainly cloning and subscription frauds. The use of neural networks allows the pattern recognition of each user of mobile telecommunication, and thereby detecting intrusions. The experimental results indicate that a significant reduction of the losses of the telecom carriers and users can be obtained using neural networks. This thesis also shows how CORBA architecture can be used to support and enhance the security of the system. In the implementation, the system guarantees access control, authentication, confidentiality, integrity, availability and non-repudiation. Finally, as an outgrowth of this research a secure on-line phone bill system was developed, allowing the end users to detect frauds and minimize their losses.*

*"Safety and Security are two reliable properties of a system.  
A 'safe' system provides protection against errors of trusted users,  
while a 'secure' system protects against errors introduced by untrusted users." (ALEXANDER et al, 1998)*

## 1 Introdução

Esta pesquisa científica é motivada pelas necessidades decorrentes do rápido crescimento das redes sem fio (*wireless*)<sup>1</sup> e da telefonia móvel (*mobile*)<sup>2</sup>. A telefonia móvel está provocando importantes mudanças no cotidiano das pessoas; no entanto, tais mudanças somente serão consolidadas quando os usuários puderem confiar na segurança das redes sem fio. A demanda crescente por serviços seguros e de alto desempenho disponíveis via telefonia móvel com acesso global, tais como pesquisas na Internet, comércio eletrônico e video-conferência, vem constituindo a principal tendência da próxima geração de redes sem fio.

Gerência (AIDAROUS & PLEVYAK, 1998, CHEIKHROUHO, 1999, SLOMAN & TWIDLE, 1994) de segurança (APOSTOLOPOULOS & DASKALOU, 1997, DOWD & MCHENRY, 1998, PFLEEGER & COOPER, 1997, ROZEMBLIT, 1999, STALLINGS, 1995) contra intrusões (DENNING, 1987, STILLERMAN & MARCEAU, 1999) e fraudes (NOTARE et al, 1999) em telefonia móvel (CAMPBELL & GOMEZ, 1999, GIBSON, 1996, GUIZANI et al, 2000, LIM & CHUN, 1999, RIEZENMAN, 2000) representa uma das principais questões associadas à próxima geração de redes sem fio<sup>3</sup> (CALHOUN, 1999, CHAUDHURY et al, 1999, CHUANG & SOLLENBERGER, 1999, DAS et al, 1999, KRAIMECHE, 1999, LU & BI, 1999, PRASAD, 1999). Atualmente, as empresas de telecomunicações estão tendo um grande prejuízo<sup>4</sup> em virtude da carência de soluções eficientes na gerência de segurança contra fraudes na telefonia móvel. Considerando a previsão para o ano de 2003 de um aumento de 70% sobre as atuais perdas<sup>5</sup> causadas pelas fraudes na telefonia móvel, esta pesquisa demonstra que um sistema para a gerência segura e correta das redes de telecomunicações pode ser a solução eficiente para esse problema.

Este Capítulo 1 está organizado como segue. Inicialmente apresenta-se o problema de segurança nas redes de telecomunicações (item 1.1). Em seguida, discute-se os trabalhos relacionados (item 1.2) e apresenta-se a proposta desta tese (item 1.3). Finalmente, mostra-se a estrutura geral do presente documento (item 1.4).

<sup>1</sup> O IEEE 802.11 (NEE, 1999) é o padrão para redes locais sem fio (*Wireless Local Area Networks – WLANs*). O objetivo desse padrão é oferecer um modelo de operação a fim de resolver as questões de incompatibilidade entre fabricantes de equipamentos WLAN.

<sup>2</sup> Em julho de 1999 foi atingido o número de 10.000.000 de aparelhos vendidos no Brasil, sendo que a média de venda nos últimos 6 meses foi de um celular a cada 6 segundos no Brasil. Em 1990 eram apenas 40.000 aparelhos – 1 para cada 2.000 pessoas. Nos Estados Unidos são 23 aparelhos para cada grupo de 100 habitantes (revista *Veja*, 14/07/99, p. 42-43). Em março de 2000 já são 15.000.000 de aparelhos no Brasil – um para cada onze brasileiros (revista *Veja*, 01/03/2000, p. 118-119). Segundo o jornal *Diário Catarinense* (03/01/2000, p. 3), a telefonia celular vai crescer 80% no Brasil no ano 2000. Serão 27 milhões de celulares, o que pelas previsões oficiais só seria atingido em 2003.

<sup>3</sup> Telefones celulares analógicos constituem a primeira geração; digitais, a segunda. A terceira é marcada pela convergência com a Internet, incluindo alta velocidade de acesso sem fio, de 384Kbps a 54Mbps. Atualmente os celulares digitais transmitem dados a velocidades em torno de 14,4Kbps (FREITAS, L. E-tudo: the Web connection. *Varig Inflight Magazine*, n. 133, p.54-122, nov. 1999).

<sup>4</sup> Segundo o IDC ([www2.uol.com.br/info/infonews/091999/](http://www2.uol.com.br/info/infonews/091999/)) mais de meio milhão de dólares diários; segundo a indústria de celular Bell Atlantic ([www.ba.com/nr/96/feb/2-29cellfraud.html](http://www.ba.com/nr/96/feb/2-29cellfraud.html)), uma perda de um milhão e meio de dólares diários apenas nos EUA. No Brasil, os crimes de clonagem e subscrição já atingiram 2% do faturamento anual das empresas (revista *Veja*, 08/10/97, p. 86).

## 1.1 Fraudes de Clonagem e de Habilitação em Telefonia Móvel

Para romper a segurança de um sistema sem fio, basta utilizar um equipamento de rádio portátil, também conhecido como *scanner*. Com esse equipamento é possível registrar as frequências de telefones celulares que estão operando em áreas próximas (em torno de 100 metros) e programar outros telefones (i.e., clones) para realizar chamadas nas frequências capturadas, de modo que as chamadas sejam debitadas nas contas telefônicas dos proprietários que tiveram as frequências captadas. No entanto, à medida que essas fraudes técnicas (leia-se clonagem) tornam-se mais difíceis de serem realizadas devido às novas tecnologias dos aparelhos digitais, os esforços voltam-se para fraudar o processo de habilitação de telefones celulares, que é independente de tecnologia, seja analógica, seja digital<sup>6</sup>. Esse tipo de fraude é favorecida pelas facilidades que as operadoras oferecem para os usuários assinarem seus serviços telefônicos – de modo que as habilitações são realizadas principalmente em nome de terceiros, que não irão pagar pelas chamadas, alegando desconhecê-las.

**Na fraude de habilitação**, também conhecida como fraude de inadimplência ou ainda de subscrição (*subscription*), o criminoso geralmente utiliza o nome de outra pessoa para assinar o serviço de telefonia celular. Esse serviço é utilizado até ser desativado por falta de pagamento (geralmente 30 dias após o início da habilitação do serviço). Dessa forma, um fraudador pode utilizar um celular diferente a cada mês, cada um em nome de um assinante diferente. Mas provavelmente todos esses aparelhos terão um padrão de chamadas em comum<sup>7</sup>.

A **fraude de clonagem** envolve a cobrança de chamadas de clones na conta de um assinante legítimo através do uso indevido dos códigos referentes ao Número Serial Eletrônico (*Electronic Serial Number – ESN*) e ao Número de Identificação do Celular (*Mobile Identification Number – MIN*) capturados do aparelho original<sup>8</sup>. Para uma chamada ser inicializada, o telefone transmite ambos os códigos – ESN e MIN – quando a tecla “enviar” (*send*) é pressionada.

---

<sup>5</sup> O que irá representar 57 milhões de dólares apenas nos Estados Unidos, segundo o IDC (<[www2.uol.com.br/info/infonews/091999/](http://www2.uol.com.br/info/infonews/091999/)>).

<sup>6</sup> A fraude de habilitação, muito difícil de ser detectada via *hardware*, tende a continuar crescendo (em 1996 as fraudes de habilitação representavam 30% das fraudes; em 2000 já representam 40%, contra 20% de clonagem, 5% de *tumbling*, 25% de pré-pagos e 10% de outras).

<sup>7</sup> Detecção via *software* (ao contrário da detecção via *hardware*) pode ser baseada no princípio da existência de um “padrão de chamada” de cada usuário.



Como pode ser observado na Figura 1.1, a clonagem ocorre pela captura, nas ondas eletromagnéticas, dos códigos MIN e ESN de um aparelho de um assinante legal, seguida pela transferência desses códigos para outro telefone celular (i.e., o clone). O celular falso é considerado pela rede como o telefone do assinante em vez de um aparelho clonado. Com isso, as chamadas feitas através do telefone clonado irão ser adicionadas na fatura telefônica do assinante legal. O exemplo típico é o do impostor que atua com um *scanner* em um *shopping center* capturando as informações dos telefones celulares em utilização. O impostor pode então transferir as informações capturadas para vários telefones, produzindo assim diversos clones<sup>9</sup>.

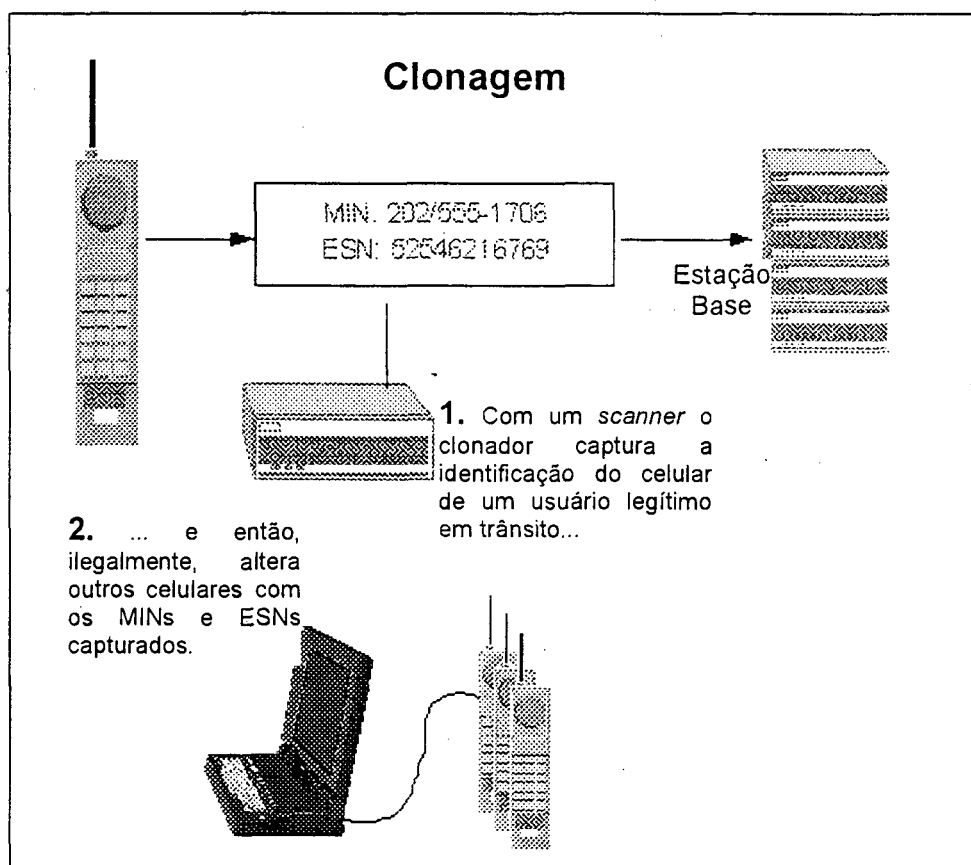


FIGURA 1.1 – CLONAGEM DE TELEFONE CELULAR.

<sup>8</sup> Cada telefone celular é único, e portanto possui códigos ESN e MIN distintos dos demais telefones celulares. O código ESN é definido pelo fabricante e o MIN é programado pelo provedor do serviço telefônico.

<sup>9</sup> Podem existir alguns indícios que um assinante de telefone celular pode identificar para detectar o uso fraudulento do seu celular em antecipação a uma empresa de telefonia: ligações frequentes de números telefônicos errados para o telefone do proprietário original; quedas de conexão; dificuldade em efetuar chamadas; existência de chamadas que constantemente recebem sinal de ocupado e números errados (seria importante questionar aqueles que lhe telefonam frequentemente para saber se existe alguma dificuldade em efetuar uma ligação para o seu número telefônico); e a existência de chamadas indevidas que possam aparecer na sua conta telefônica. Certamente esses itens fornecidos pela AT&T não garantem a segurança da rede, e em acréscimo, é importante enfatizar que a responsabilidade da segurança da rede não é do usuário (que está pagando pelo serviço), mas da empresa telefônica que deve fornecer um serviço seguro.

Existe um aperfeiçoamento dessa fraude de clonagem apresentada na Figura 1.1, chamada de *tumbling fraud* ou então de clone com *tumbling function*. Nesse caso, não é necessário possuir um *scanner*, pois o próprio celular maliciosamente alterado, fica constantemente capturando os pares ESN e MIN (10 + 10 dígitos decimais) e armazenando-os na memória do aparelho (mesmo se o fraudador mudar de cidade ou país). A captura acontece dentro de um raio que varia de 50 a 500 metros (dependendo de local aberto, paredes de concreto, etc) e durante 3 possibilidades: (1) alguém pressiona a tecla “enviar” (*send*) tentando fazer uma chamada; (2) se alguém está recebendo uma chamada (não necessariamente respondida); e (3) periodicamente (cada poucos minutos) o assinante transmite automaticamente seu par ESN/MIN para o seu sítio (*cell site*). Geralmente o clone (que pode estar no modo “*scanning*” ou “*calling*”, capturando pares ESM/MIN ou apto para fazer/receber chamadas, respectivamente) é capaz de manter armazenados os últimos 50 pares capturados (e sem duplicações) e cada chamada utiliza um desses pares. Ou seja, as chamadas realizadas irão ser debitadas cada uma em uma conta telefônica diferente. A grande vantagem de um celular com *tumbling function* é a grande flexibilidade e portabilidade que o fraudador ganha, pois fica independente da re-programação do aparelho, quando após um mês de uso (emissão da fatura mensal) o clone é detectado (e desativado). Em acréscimo, uma das posições da memória (mais precisamente, a posição 99) pode ser facilmente programada (digitando-se M+, 99, M+) para além de realizar chamadas (cada vez com um número diferente) ainda receber chamadas utilizando um determinado número<sup>10</sup>, inclusive o número do aparelho legítimo que o fraudador possuir.

Uma fatura típica de um telefone que foi clonado é apresentada na Figura 1.2.

---

<sup>10</sup> Apenas um dos telefones toca/recebe a chamada, e não existe a possibilidade de linhas cruzadas/escuta.



Porém, mais do que dificultar a clonagem através de novos equipamentos digitais<sup>12</sup>, vários outros métodos de prevenção devem ser instaurados, já que fraudes tais como o uso de nome de terceiros na habilitação não podem ser facilmente impedidas ou minimizadas via *hardware*. Nesse contexto, novas ferramentas para a detecção de fraudes, tais como redes neurais (considerando a existência de padrões de chamada e padrões de usuários) são armas que devem ser usadas em uma guerra já deflagrada. Este trabalho, no escopo da gerência de segurança de redes sem fio, investiga o uso da tecnologia de redes neurais como uma das ferramentas para combater este desafiante problema de fraudes na telefonia móvel.

## 1.2 Trabalhos Relacionados

Existem estratégias distintas através das quais é tratado o problema das fraudes nas telecomunicações móveis: (i) criptografia; (ii) bloqueio; (iii) verificação de usuário; e (iv) análise de tráfego. Neste item 1.2, são discutidos prós e contras de cada uma destas estratégias de prevenção e detecção de fraudes (CAIN et al, 1997, LAWLESS, 1999, RUBIN & GEER, 1999, STEWART, 1999).

**(i) Criptografia:** Uma das formas mais utilizadas para prevenir fraudes, a criptografia<sup>13</sup>, apresenta dois benefícios principais: dificultar a detecção dos pares de código *ESN/MIN* e prevenir o sistema contra a escuta clandestina. A criptografia consiste em transformar uma mensagem legível (*plaintext*) em uma mensagem codificada e não legível (*ciphertext*) para só então transmiti-la. Além de uma chave (*shared key*), a criptografia utiliza um algoritmo para codificar (*encryption algorithm*) e um algoritmo para decodificar (*decryption algorithm*) a mensagem (STALLINGS, 1995). Embora possa ser simples incluir criptografia nos telefones celulares digitais, em virtude de estes utilizarem uma representação binária que pode ser facilmente codificada e decodificada, nos telefones analógicos, por sua vez, a criptografia torna-se extremamente cara e difícil de implementar. Por isso, como ainda existem telefones analógicos, a criptografia ainda não pode ser considerada uma solução eficaz. Em acréscimo, telefones digitais (*GSM*) já foram clonados

<sup>12</sup> Como, por exemplo, os da *Intel Corporation* com a tecnologia *Boot Block Flash*, que codifica o ESN.

<sup>13</sup> Júlio César (101 a.C – 44 a.C), o imperador romano, usava a “cifra de César” para enviar ordens secretas a seus generais. O código consistia em trocar cada letra de uma mensagem pela terceira seguinte. Assim, o “a” virava “d” e assim por diante. Para proteger-se de bisbilhoteiros, o pintor Leonardo da Vinci (1452-1519) se valia de um artifício curioso. Ele escrevia da direita para a esquerda, de modo que seus textos só podiam ser lidos diante de um espelho. A rainha da Escócia Mary Stuart (1542-1587) queria tirar a prima Elizabeth do trono inglês. Ela conspirava em cartas cifradas, com símbolos no lugar das letras. Mas o código era simples demais. Mary foi descoberta e degolada. O desfecho da Segunda Guerra Mundial poderia ter sido outro se não

através da quebra da criptografia (em abril de 1998, Universidade da Califórnia)<sup>14</sup>. Outro obstáculo para a adoção de criptografia consiste na oposição do Escritório Norte-Americano de Investigação Federal (*Federal Bureau of Investigation – FBI*) e da Agência Norte-Americana de Segurança (*National Security Agency – NSA*). Estas agências temem que, se for permitido o uso de criptografia, criminosos poderão codificar suas próprias ligações telefônicas. Tal medida impossibilitaria que o *FBI* e a *NSA* realizassem escutas em chamadas telefônicas suspeitas. Como alternativa, o governo norte-americano está apoiando a adoção de um dispositivo denominado *Clipper Chip*, que através de uma trava de acesso permite a decodificação da comunicação sem conhecer a senha. Dessa forma o governo norte-americano poderia escutar uma conversa criptografada quando necessário. Mas a perspectiva do governo de ter meios para realizar escutas é desconfortável para vários usuários, e como resultado esse assunto tem sido muito controverso. A *AT&T* já anunciou a sua linha de produtos providos de segurança, a qual inclui telefones celulares que utilizam criptografia (o celular *Security 2000* utiliza o *Clipper Chip*). Embora essa estratégia seja uma boa contribuição para sistemas mais seguros, ela não é aplicável em aparelhos analógicos e nem em fraudes de habilitação.

**(ii) Bloqueio:** Com o objetivo de proteção contra prejuízos, algumas operadoras bloqueiam usuários de risco a realizarem alguns tipos de chamadas. A venda de telefones celulares também pode ser restrita, isto é, somente os clientes que comprovarem renda poderão adquirir um aparelho. Entretanto, se a companhia dificultar a assinatura ou chamadas de clientes potenciais, esta não estará somente perdendo clientes mas também a concorrência e possivelmente seu lugar no mercado. O bloqueio de chamadas internacionais é uma estratégia em que a companhia perde muitas possíveis chamadas, e conseqüentemente lucros. A *TIM TELESC* é um exemplo de companhia que utiliza a estratégia de bloqueio das chamadas internacionais de todos os seus clientes<sup>15</sup>. O bloqueio (leia-se não-disponibilidade de serviço) não chega a ser uma solução para a gerência de segurança, mas uma alternativa para sistemas sem segurança.

---

fosse a mente brilhante de Alan Turing (1912-1954). Ele liderou a equipe inglesa de criptoanalistas que desvendou como funcionava a *Enigma*, a máquina usada pelos nazistas para codificar mensagens (Super Interessante. O segredo da Criptografia. p51-53, maio 2000).

<sup>14</sup> VEJA. Segredo Fácil: Hackers provam que celulares digitais podem ser clonados. p. 81, 22 abr. 1998.

<sup>15</sup> Todos os assinantes desta companhia necessitam solicitar o desbloqueio para utilizar o serviço. Em acréscimo, não são avisados da não-disponibilidade deste serviço pelo qual pagam.

**(iii) Verificação de Usuário:** Várias empresas já desenvolveram mecanismos para a verificação de usuário. A *Nynex Mobile* e a *Cellular One* foram as empresas que iniciaram a investigação sobre o uso de Números de Identificação Pessoal (*Personal Identification Numbers – PIN*). A *Brite Voice Systems Inc* recentemente anunciou um produto para reconhecimento de voz chamado *Voice Select Sentry*, o qual utiliza três opções de identificação de usuário. A primeira opção obriga o usuário a digitar ou falar uma senha. A segunda opção trata-se de um método de verificação de voz que obriga o usuário a falar uma senha, e o sistema então verifica se esta coincide com uma voz pré-gravada. A terceira opção constitui um acesso pré-pago, em que um usuário paga previamente por uma chamada de longa distância. Um outro serviço, chamado Serviço de Proteção Contra Fraude (*Fraud Protection Feature – FPF*), é fornecido gratuitamente por algumas empresas de telecomunicações, como a *AT&T*<sup>16</sup>. Neste caso, um número de sete dígitos, discado antes de cada ligação, é gravado e programado de acordo com a velocidade de discagem de um usuário. Sem este número, o fraudador não consegue clonar um telefone. De acordo com a *Nynex*<sup>17</sup>, o dispositivo de senha é a ferramenta mais eficiente disponível no momento. De janeiro a setembro de 1995, o uso de senhas reduziu a fraude em mais de 80% nos mercados analisados. Além de este valor de 80% não corresponder a um eficiente resultado, esta tese considera que o usuário não pode ser imbuído de mais senhas. Pelo contrário, o usuário merece um serviço seguro em função do que está sendo pago.

**(iv) Análise de Tráfego:** Análise de tráfego começa a ser utilizada para detectar padrões de chamadas suspeitas, tais como o aumento repentino da duração das chamadas e o aumento de chamadas internacionais, e também para determinar se é fisicamente possível um assinante efetuar uma chamada no local corrente em relação ao local e horário da última chamada. No entanto, na maioria das companhias, tais como a *Tele Centro Sul*, apenas faturas de valores elevados são investigadas, sem considerar, por exemplo, que um usuário já possui um padrão de chamadas internacionais para determinado número em determinado horário, tendo apenas aumentado a duração das mesmas. Este procedimento é muito pouco eficiente na detecção de fraudes. Em acréscimo, em muitos casos, as chamadas só são analisadas após a emissão da fatura mensal, quando as perdas já serão muito grandes. Outro aspecto muito negativo nesse contexto é o grande número de funcionários necessários para analisar as contas telefônicas de valores elevados e/ou contatar os usuários. Algumas das companhias que estão desenvolvendo *software* para detectar padrões de tráfego suspeitos são a *GTE* e a *Coral Systems*. O *software*

<sup>16</sup> <[http://www.attws.com/general/about\\_us/factsheets/wireless\\_fraud.html](http://www.attws.com/general/about_us/factsheets/wireless_fraud.html)>

<sup>17</sup> <<http://www.craftsreport.com/april96/cellularfraud.html>>

desenvolvido pela *GTE* é conhecido como *CloneGuard*, e o software da *Coral System* é conhecido como *FraudBuster*. A *AT&T* e a *Bell Atlantic* também estão desenvolvendo métodos de análise para detectar discrepâncias nos padrões de chamadas de seus clientes. Porém, uma característica negativa no procedimento atual é que, se uma companhia telefônica conseguir detectar uma discrepância importante no padrão de chamada de um usuário, ela possui a autonomia de bloquear o número telefônico deste usuário sem avisá-lo. Como desvantagem, há a possibilidade de se tratar de um usuário honesto necessitando fazer estas chamadas, ou mesmo que não tenha realizado as chamadas (mas sim o fraudador), o usuário não pode ser impedido de utilizar seu equipamento por este ter sido clonado. Como relatado por um representante da *Bell Atlantic*, em 90% dos casos a operadora local irá notificar o cliente antes do bloqueio do seu número<sup>18</sup>. Entretanto, se a discrepância for detectada por uma operadora de longa distância, esta irá notificar a operadora local que, por sua vez, irá informar o cliente. Neste ínterim, a operadora de longa distância já poderá ter bloqueado o número telefônico com o objetivo de prevenir um uso ilegal. Nesse caso, o cliente poderá tentar fazer uma chamada e não terá sucesso. Em função do desenvolvimento desta tese acredita-se que o número de 90% não representa o melhor resultado e que o atraso em informar os clientes pode provocar profundas perdas para os clientes e para as companhias. Dessa maneira, tal porcentagem necessita ser aumentada, principalmente a partir da utilização de um método automático e imediato de avisar os clientes (sem a necessidade do grande número de funcionários), para somente após bloquear o serviço.

### 1.3 Escopo da Tese – Proposta e Principais Contribuições

Com a crescente popularidade das redes sem fio e com as rápidas mudanças na indústria de telecomunicações móveis, a preocupação com a segurança dos usuários de telefonia celular deveria ser muito maior do que se observa atualmente. Este trabalho propõe o desenvolvimento de um sistema de gerência de segurança para sistemas distribuídos em geral (SIMON, 1996) e para sistemas de comunicação sem fio em particular (LU & BI, 1999, RIEZENMAN, 2000). O gerenciamento de segurança proposto compreende um sistema de detecção de intrusão na telefonia móvel através da análise de tráfego, onde os usuários são classificados em grupos de acordo com seus padrões de utilização do aparelho<sup>19</sup> (NOTARE, 1999).

---

<sup>18</sup> AT&T and GTE Test Way to Cut Phone Fraud, <[www.ba.com/nr/96/feb/2-29cellfraud.html](http://www.ba.com/nr/96/feb/2-29cellfraud.html)>.

<sup>19</sup> A privacidade das informações sobre os usuários continua mantida. O que difere no sistema proposto é o conhecimento prévio destas informações, ou seja, não apenas no término do ciclo mensal quando da impressão da conta telefônica.

A classificação dos usuários em grupos realizada através do emprego de **redes neurais** ajuda o sistema a identificar quando chamadas não correspondem aos padrões do usuário deste telefone, constituindo um possível clone; bem como identificar o padrão de um usuário como muito similar a um padrão de um antigo inadimplente, constituindo um usuário que habilita um celular com a prévia intenção de não pagar pelos serviços<sup>20</sup>. Assim, quando uma ligação telefônica (realizada por um telefone legal ou clonado) é concluída, o sistema verifica se as características da chamada estão dentro dos padrões do determinado usuário (características estas armazenadas previamente em um arquivo de padrões), e também se a chamada é fisicamente possível<sup>21</sup>. Uma mensagem automática (economizando gastos da companhia com pessoal) é enviada ao cliente assim que uma possível fraude de clonagem for detectada. No caso da fraude de habilitação, o sistema identifica um padrão similar a um antigo inadimplente e então investiga para certificar-se da identidade de quem está utilizando o aparelho, ou seja, se é quem realmente diz ser<sup>22</sup>. Essa imediata notificação e investigação, ao contrário da espera até a emissão da fatura no final do mês, ajuda na redução dos prejuízos da companhia, bem como na eliminação dos danos que podem ser repassados para os clientes como, associá-los a criminosos.

Além do gerenciamento contra as fraudes de clonagem e de habilitação a partir das companhias, esta tese também oferece uma aplicação disponível via Web, onde os usuários de telecomunicações podem observar suas contas telefônicas constantemente atualizadas, o que permite que o próprio usuário detecte e minimize fraudes associadas à clonagem. A aplicação engloba os serviços de controle de acesso, autenticação, confidencialidade, integridade, disponibilidade e não-repúdio de comunicação. Esses serviços de segurança são implementados em **Java** com suporte **CORBA**, e incluem mecanismos tais como assinatura e certificado digital, que são muito importantes mas ainda negligenciados por empresas como bancos e companhias de cartão de crédito<sup>23</sup>.

Outra característica muito importante do sistema de gerência proposto é a validação formal a partir das especificações de serviço e de protocolo, em conformidade com o padrão **ISO 8807**.

<sup>20</sup> Neste caso, o fraudador satisfaz-se em usar o aparelho apenas por uma média de 30 dias, quando o mesmo é desabilitado por falta de pagamento. Geralmente o aparelho é comprado em nome de terceiros.

<sup>21</sup> Por exemplo, duas ligações de um mesmo usuário realizadas em um intervalo de 5 minutos e que partiram de localidades distantes 500 Km revelam a existência de um clone, pois uma destas ligações é fisicamente impossível de ter sido realizada pelo mesmo aparelho.

<sup>22</sup> O sistema inicialmente detecta a alteração de padrão, que pode ser um indicio de clonagem; após esse procedimento o padrão detectado é comparado com o padrão de inadimplentes conhecidos, a fim de detectar um possível inadimplente.

<sup>23</sup> Importantes bancos e companhias de cartões de crédito disponibilizam serviços via Web sobre protocolos para segurança, tais como SSL (*Secure Socket Layer*) e SET (*Secure Electronic Transaction*); porém não exigem a certificação digital – a garantia para o usuário de que está acessando o site correto.



Neste escopo, a gerência de segurança para sistemas distribuídos proposta neste trabalho engloba principalmente três tecnologias: (1) redes neurais, para o reconhecimento de padrões de uso da telefonia móvel; (2) CORBA, para a distribuição segura de agentes e gerente; e (3) LOTOS, para a especificação e validação formal, com o objetivo de provar a correção do sistema. Mais especificamente, esta tese visa:

- (i) propor uma gerência de segurança, segura e correta, para sistemas distribuídos, aplicá-la em um sistema de segurança para telecomunicações móveis e sugerir onde esta gerência possa também ser aplicada;
- (ii) oferecer um tutorial para a especificação e verificação formal, de acordo com o padrão ISO 8807, a fim de validar sistemas de segurança distribuídos e garantir matematicamente sua correção, com  $\sigma$  objetivo de atingir o nível máximo de segurança caracterizado pelo Departamento de Defesa dos Estados Unidos;
- (iii) descrever a implementação de como o ODP/OMG CORBA e seu módulo de segurança pode ser usado para suportar, manter e proteger sistemas distribuídos, neste caso um sistema de segurança para telefonia móvel, onde a comunicação entre agentes e gerente é realizada de maneira segura. Para garantir a segurança e a privacidade do usuário, mostra-se como o módulo de segurança oferecido por Java/Web pode contribuir para o sistema distribuído, em que a comunicação entre usuários e servidor incorpora os principais mecanismos de segurança;
- (iv) propor e conceber um serviço de alarme imediato e automático, para contatar, informar e confirmar a existência de uma clone diretamente com o usuário (vítima da fraude). Este serviço é similar aos atuais serviços de “despertador automático”, em que as mensagens (interativas) são recebidas e enviadas sem a necessidade da presença de um funcionário, e dessa forma viabilizar o sistema também de forma mais econômica;
- (v) investigar e desenvolver os sistemas SSCC (Sistema de Segurança Contra Clonagem de Celulares) e SIPI (Sistema de Identificação de Prováveis Inadimplentes) contra fraudes na comunicação móvel (i.e., clonagem e habilitação), que através do emprego de redes neurais monitora e identifica fraudadores. A escolha correta das características e do algoritmo de classificação é decisiva na eficiência da detecção das fraudes, contribuindo para a redução dos prejuízos das companhias telefônicas e dos usuários;

(vi) conceber e desenvolver o sistema SETWeb (Sistema de Extrato Telefônico on-line via Web), que permite aos usuários (previamente cadastrados) observarem suas contas telefônicas, constantemente atualizadas, via Web. O SETWeb garante a segurança do cliente, quando este acessa sua conta telefônica, através de mecanismos implementados em Java com suporte CORBA. A principal característica do SETWeb é possibilitar que os próprios usuários detectem, rapidamente, a existência de ligações telefônicas ilícitas na sua conta telefônica (i.e., detectem ligações realizadas por clones de seu aparelho celular)<sup>24</sup>; e

(vii) conceber um produto para gerência de segurança de redes de telecomunicações.

Veja na Figura 1.3 uma visão geral do sistema de gerência de segurança SSTCC<sup>25</sup> – Sistema de Segurança para Telecomunicações Contra Clonagem e Inadimplência (que engloba os sistemas SSCC – Sistema de Segurança Contra Clonagem de Celulares, SIPI – Sistema de Identificação de Prováveis Inadimplentes e SETWeb – Sistema de Extrato Telefônico on-line via Web).

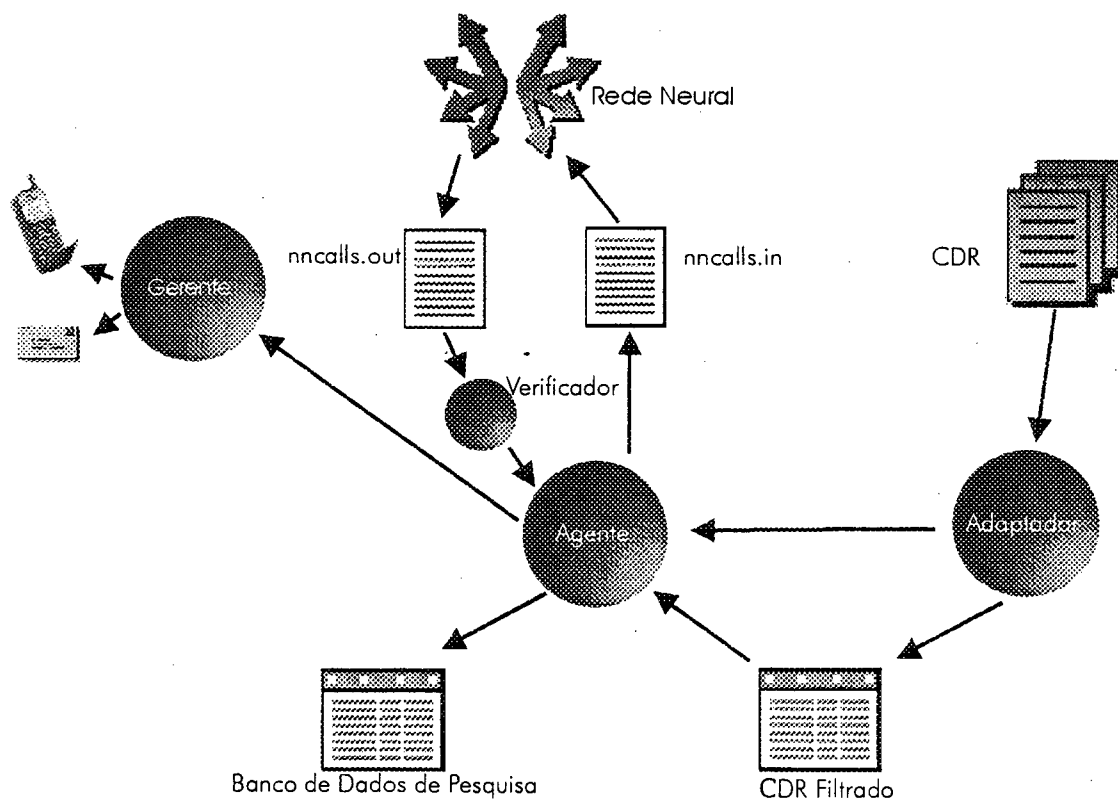


FIGURA 1.3 – FUNCIONALIDADE DO SISTEMA SSTCC.

<sup>24</sup> O extrato telefônico pode ser disponibilizado também via telefone (com mensagens gravadas interativas), fax ou e-mail.

<sup>25</sup> SSTCC® - Este programa (*software*), bem como sua marca, encontra-se protegido contra utilização não autorizada, total ou parcial, conforme a Lei 9.609 de 19/02/1998, regulamentada pelo Decreto 2.556 de 20/04/1998, c/c Lei 9.610 de 19/02/1998, estando devidamente registrado no INPI sob o nº 99001177, ficando os infratores sujeitos às sanções cíveis e penais previstas nos respectivos diplomas legais.

Em resumo, (i) o módulo Adaptador lê as ligações *on-line* do arquivo CDR e armazena as informações necessárias no arquivo CDR Filtrado; (ii) o módulo Agente monitora e detecta possíveis fraudes (utilizando redes neurais) comparando as informações do CDR Filtrado com as do Banco de Dados de Pesquisa (histórico de cada usuário); e (iii) o módulo Gerente recebe notificações do módulo Agente e envia alarmes automáticos e imediatos aos usuários (por telefone e correio) no caso da fraude de clonagem e investiga prováveis futuros inadimplentes no caso da fraude de habilitação. Em acréscimo, a possibilidade de observar o extrato telefônico via Web permite que o próprio usuário identifique clones de seu aparelho. A demonstração da gerência proposta detalhando o desenvolvimento de cada componente do sistema distribuído de segurança é o objeto desta tese.

#### **1.4 Organização deste Trabalho**

Este documento está organizado como segue. O Capítulo 2 descreve o uso do padrão ISO 8807, com a finalidade de especificar e validar formalmente o sistema. O Capítulo 3 demonstra como as redes neurais podem ser utilizadas para a detecção de intrusão em redes de telecomunicações móveis. O Capítulo 4 apresenta a implementação do sistema distribuído de segurança e mostra como CORBA e Java podem suportar, manter e enriquecer a segurança do sistema. O Capítulo 5 sumariza as contribuições e sugere futuros trabalhos.

*“Prevent and Cure. Separate from their classification as management and technology, reliability techniques fall into two categories: (1) a priori techniques, which strive to build software right; and (2) a posteriori techniques, which attempt to right the wrongs. Formal specifications is an example of a priori; testing, of a posteriori. It’s impossible to ignore formal methods. True, they have a bad reputation in some circles as being too heavy and difficult. That reputation, however, is not entirely justified. Formal methods have achieved a number of success. They are the only game in town when it comes to guaranteeing a regulatory authority that you have produced a correct engineering design.” (MEYER, 1999)*

*“To our knowledge, the formality of trusted systems designed to substitute formal proof of security in place of experimental satisfaction – has to date found little place on the Web. We know of no Web use of formal evaluation criteria such as those in the US Defense Department’s Orange Book.” (RUBIN & GEER, 1998)*

*“The requirement for mathematical proof is formidable for something as complex as a general-purpose computer. A system that can provide such verification is referred to as a trusted system. The U.S. Department of Defense in 1981 established the Computer Security Center within the National Security Agency (NSA) with the goal of encouraging the widespread availability of trusted computer systems.” (STALLINGS, 1995)*

## 2 Especificação e Validação Formal (ISO 8807)

Este Capítulo 2 apresenta um tutorial para a especificação (veja Anexo 1.1) e validação (veja Anexo 1.2) formal de sistemas distribuídos. A Técnica de Descrição Formal (TDF) utilizada é o padrão ISO 8807 – LOTOS: *Language of Temporal Ordering Specification* (BRINKSMA, 1988). O **principal objetivo** do emprego desta técnica de alto rigor matemático é prover a prova formal de correção do sistema.

O Sistema de Segurança para Telecomunicações Contra Clonagem e Inadimplência (SSTCC), já apresentado informalmente no item 1.3, é agora especificado formalmente, com a utilização de uma abordagem de refinamentos sucessivos (VISSERS et al, 1988), a qual permite que o sistema seja validado ao longo do seu desenvolvimento – e não somente após a obtenção da especificação final. A validação do sistema emprega a ferramenta *Eucalyptus ToolSet 2.3* (GARAVEL, 1997) em ambiente *Sun Solaris 2.6*.

Este Capítulo 2 está organizado como segue. Inicialmente são descritos os tópicos principais da TDF LOTOS (item 2.1), seguidos do desenvolvimento das especificações de serviço (item 2.2) e de protocolo (item 2.3) do sistema SSTCC. Posteriormente, citam-se as formas de validação formal (item 2.4) e apresentam-se os experimentos na validação do sistema (item 2.5). E finalmente, no item 2.6, são discutidos os resultados, ou seja, a obtenção da prova formal (matemática) de correção do sistema visando atingir o nível máximo de segurança (*ClassA1*) de acordo com o *Orange Book* do Departamento de Defesa dos Estados Unidos.

## 2.1 A Técnica de Descrição Formal LOTOS

Para especificar rigorosamente sistemas distribuídos é conveniente utilizar técnicas de descrição formal a fim de prover maior confiabilidade a tais sistemas, geralmente complexos. As linguagens de especificação formal são baseadas em teorias matemáticas e estão associadas a métodos de especificação precisos e não ambíguos. Técnicas de Descrição Formal (TDFs) servem para definir os aspectos de comportamento e também os aspectos de dados de sistemas (PIRES, 1994, QUEIROZ & CUNHA, 1994).

Pode-se destacar como características de uma boa técnica de descrição formal: descrição mínima e precisa (deve fornecer meios para uma descrição precisa de todas as propriedades com um mínimo de informação extra); facilidade de entendimento (deve prover descrições facilmente compreensíveis); poder de expressão (deve ter capacidade para exprimir uma quantidade muito grande de propriedades importantes para a descrição de sistemas); fundamentação matemática (deve ter os seus elementos de definição e de especificação baseados em um modelo matemático formal); e flexibilidade (deve fornecer meios para uma descrição suficientemente flexível para adaptar-se às pequenas alterações nas propriedades a serem especificadas).

LOTOS (BRINKSMA, 1988) e ESTELLE (*Extended Finite-State Machine Language*) são TDFs normalizadas pela ISO (*International Organization for Standardization*), enquanto SDL (*Specification and Description Language*) é uma TDF normalizada pelo antigo CCITT (*Consultative Committee for International Telegraph and Telephone*) – hoje, ITU-TS (*International Telecommunications Union – Telecommunication Standardization Sector*).

Este trabalho optou por LOTOS, pois, principalmente:

- (1) enquanto SDL e ESTELLE são baseadas em linguagens de programação (Chill e Pascal, respectivamente), LOTOS é independente de linguagem de programação;
- (2) existem ótimas ferramentas para a validação de especificações LOTOS, tais como o CADP (*Caesar/Aldebaran Development Package*) – inclusive agora em ambiente gráfico (*Eucalyptus Toolset*) para *Sun Solaris* – desenvolvido pela equipe VASY (Validação de Sistemas) do INRIA/Grenoble, com a qual se mantêm relações internacionais bilaterais;
- (3) LOTOS é um padrão internacional (ISO 8807); e

(4) o nível de abstração é uma das mais importantes características para o desenvolvimento de sistemas. LOTOS apresenta, além do elevado nível de abstração, outras características inerentes às TDF como o poder de expressão e a estruturação de especificações (QUEIROZ & CUNHA, 1994).

LOTOS é uma TDF que reúne duas álgebras: a primeira álgebra é utilizada para a descrição de aspectos de comportamento e corresponde a uma extensão de Cálculo de Sistemas Comunicantes (*Calculus of Communicating Systems – CCS*) (MILNER, 1980); a segunda álgebra é utilizada para a descrição dos aspectos de dados e corresponde à linguagem *ACT ONE* (EHRIG & MAHR, 1985). Em LOTOS Básico (isto é, a parte de *LOTOS* que representa apenas os aspectos de comportamento dos sistemas) uma especificação é constituída por uma hierarquia de definições de processos.

Pode-se ver um processo como uma caixa preta dotada de portas para a comunicação com o ambiente e com outros processos, via eventos de comunicação. Eventos de comunicação são ações que ocorrem nas portas de comunicação. Ações que não são observáveis externamente são chamadas de ações internas e representadas pelo símbolo *i*.

A definição de especificações e de processos *LOTOS* tem o formato descrito abaixo.

```
specification <id> [lista_de_parametros]:<funcionalidade>
behaviour   <expressao_de_comportamento_da_especificacao>
where process   <id>[<lista_de_parametros>]:<funcionalidade>:=
                <expressao_de_comportamento>
                endproc
                ...
endspec
```

onde:

- (1) *id* identifica o nome da especificação ou o nome de um processo;
- (2) *lista\_de\_parametros* refere-se às portas de comunicação;
- (3) a *funcionalidade* de um processo pode ser *exit* se ele termina com sucesso, habilitando um processo subsequente, ou então, a *funcionalidade* pode ser *noexit* no caso de processos recursivos (infinitos), por exemplo; e
- (4) *expressao\_de\_comportamento* refere-se ao comportamento da especificação ou do processo.

### 2.1.1 Processos *LOTOS* *stop*, *exit* e Infinitos

Os processos *stop* e *exit* pertencem à linguagem *LOTOS*. Nenhum deles possui qualquer evento, seja observável (*Act-(i)*), seja invisível (*i*). O processo *stop* representa a ausência completa de atividade. O processo *exit*, por sua vez, indica uma terminação com sucesso, habilitando a realização de algum comportamento subsequente. Exemplo:

```
process le_escreve[in,out] : noexit :=
  leitura[in] >> escrita[out]      where
  process leitura[in]: exit := in;  exit  endproc
  process escrita[out]: noexit := out; stop  endproc
endproc
```

Nesse caso, após a execução do processo *exit*, o controle é transferido para o estado inicial do processo *escrita[out]*.

Outro meio de expressão de *LOTOS* é a chamada recursiva de processos. Tais chamadas permitem representar comportamentos infinitos. Exemplo:

```
process Ciclico_2[a1,a2]:noexit :=
  a1;a2;Ciclico_2[a1,a2]
endproc
```

Nesse exemplo, o processo *Ciclico\_2* repete a seqüência de eventos *a1;a2* infinitamente.

### 2.1.2 Operadores *LOTOS*

Uma expressão de comportamento em *LOTOS* relata o que pode ser observado em termos de eventos. Nela podem ser usados operadores como:

<code>;</code>	(seqüenciamento de eventos);
<code>[]</code>	(escolhas indeterminísticas entre comportamentos);
<code>   </code>	(composição de processos independentes);
<code>  </code>	(composição de processos dependentes);
<code> [...]  </code>	(composição geral);
<code>hide...in</code>	(ocultação de eventos);
<code>&gt;&gt;</code>	(habilitação de processos); e
<code>[&gt;</code>	(interrupção de processos).

O operador `;` indica seqüenciamento de eventos, permitindo expressar que após a ocorrência de um primeiro evento ocorre um segundo evento. Por exemplo: `a1;a2`. Nesse caso, o evento `a2` somente ocorre após a ocorrência de `a1`. O operador `;` também pode ser usado para expressar seqüenciamento de comportamentos. Por exemplo: `B1; B2`.

O operador `[]` indica uma escolha indeterminística entre dois comportamentos. Por exemplo: `B1[]B2`. Nesse caso, após a escolha indeterminística, pode-se observar o comportamento descrito por `B1` ou então o comportamento descrito por `B2`.

Os operadores `|||`, `||` e `|[...]|` possibilitam a representação de processos concorrentes.

Com o operador `|||` representam-se os processos executados concorrentemente que evoluem sem sincronização entre si. No caso de processos que compartilham nomes de eventos, esses processos sincronizam-se com seus ambientes comuns, mas um processo não se sincroniza com outro. Por exemplo: `calcula_raizes_pares[a,b] ||| calcula_raizes_impares[b,c]`. Nesse caso, ambos os processos são executados em paralelo, mas sem sincronizarem suas portas entre si.

Com o operador `||` representam-se os processos executados concorrentemente que evoluem sincronizadamente. Por exemplo: `recebe_notificações[...] || filtra_notificações[...]`. Nesse caso, os processos são executados em paralelo compartilhando todos os eventos observáveis.

Com o operador `|[...]|` representam-se os processos executados concorrentemente que evoluem sem sincronização, salvo onde há portas compartilhadas pelos processos. Tais portas são explicitadas pelo operador `|[...]|`. Por exemplo: `coleta[oper,notif,dados] |[dados]| filtra[dados,alarm]`. Nesse caso, os processos `coleta` e `filtra` são executados em paralelo compartilhando eventos na porta `dados`.

Com o operador `hide...in` representa-se a ocultação de eventos, tornando-os invisíveis. Essa ocultação é importante para fins de análise e verificação de equivalências.



O operador  $\gg$  representa seqüenciamento de processos. Em  $B1 \gg B2$  o processo  $B2$  só é executado após o término com êxito do processo  $B1$ . Exemplo: `recebe[...] \gg responde[...]`. Nesse caso, o processo `responde` só será executado após o término com êxito do processo `recebe`.

O operador  $[>$  representa interrupção de processo. Em  $B1 [> B2$  o processo  $B2$  pode interromper a qualquer instante o processo  $B1$ . Exemplo: `transmite_dados[...] [> interrompe_transmissão[...]`. Nesse exemplo, o processo `interrompe_transmissão` pode interromper o processo `transmite_dados` a qualquer momento (antes ou durante a execução do processo `transmite_dados`).

### 2.1.3 Modelo Semântico

O modelo da TDF *LOTOS* baseia-se em sistemas de transições rotuladas (*LTSs – Labelled Transition Systems*). Sistemas de transições rotuladas são constituídos de transições etiquetadas entre estados que evoluem no decorrer do tempo. As transições representam eventos observáveis ou eventos internos (BRINKSMA, 1988, MILNER, 1980).

Um sistema de transições rotuladas  $sys$  é uma quádrupla  $\langle S, Act, T, s_0 \rangle$ , onde:

- (1)  $s$  é um conjunto não vazio, e seus elementos referem-se aos estados;
- (2)  $Act$  é um conjunto cujos elementos referem-se às ações;
- (3)  $T$  é um conjunto de relações de transições que contém precisamente uma relação  $a \rightarrow \subseteq S \times S$  para cada  $a \in Act$ ; e
- (4)  $s_0$  é o estado inicial de  $sys$ .

Uma transição de um sistema de transições rotuladas é uma tripla  $\langle corrente, a, next \rangle$ , tal que  $\langle corrente, next \rangle \in a \rightarrow$ . Sejam  $B1$  e  $B2$  expressões de comportamento,  $B1 \xrightarrow{a} B2$  significa que do estado onde se tem a expressão de comportamento  $B1$ , após a ocorrência de um evento  $a$ , alcança-se o estado onde se tem a expressão de comportamento  $B2$ . Através desse modelo de transição, a semântica dos operadores *LOTOS* pode ser definida.

## 2.2 Especificação de Serviço do SSTCC

Inicialmente, no nível mais alto de abstração, o sistema SSTCC (Sistema de Segurança para Telecomunicações Contra Clonagem e Inadimplência) pode ser visto como uma caixa preta, dotada de quatro portas de comunicação (porta `mail_alarm`, porta `phone_alarm`, porta `online_bill` e porta `check_owner`), para a troca de mensagens com os usuários de uma dada companhia telefônica. Veja a Figura 2.1.

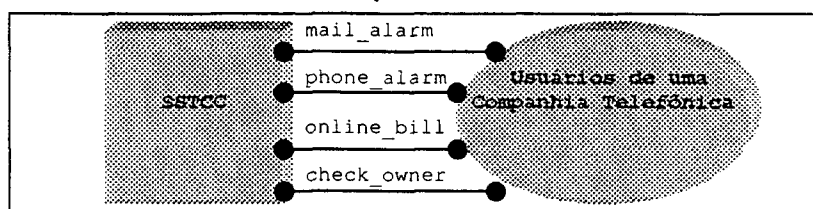


FIGURA 2.1 – REPRESENTAÇÃO GRÁFICA DO SSTCC<sup>26</sup>.

A porta `mail_alarm` é utilizada para que o SSTCC envie alarmes de suspeita de existência de clones, diretamente ao usuário, pelo correio comum. Já a porta `phone_alarm` permite que o SSTCC empregue o telefone celular para esta finalidade. O envio de alarmes através do celular tem, como maior vantagem, o tempo; enquanto que o envio de denúncias pelo correio tem, como maior vantagem, a segurança. A porta `online_bill` é utilizada para que a companhia telefônica possa disponibilizar aos seus usuários a conta mensal atualizada *on-line*. Finalmente, a porta `check_owner` é utilizada pela companhia telefônica para investigar suspeitas de prováveis inadimplentes.

O sistema SSTCC fica permanentemente ativo, o que caracteriza um comportamento infinito desse sistema. Esse comportamento sugere uma especificação LOTOS com funcionalidade `noexit`. Veja a Figura 2.2.

```

specification
SstccService[mail_alarm,phone_alarm,online_bill,check_owner]:noexit
behaviour SstccService[mail_alarm,phone_alarm,online_bill,check_owner]
where
process SstccService[mail_alarm,phone_alarm,online_bill,check_owner]:noexit:=
(i;mail_alarm;
  (phone_alarm;SstccService(mail_alarm,phone_alarm,online_bill,check_owner)
  [] SstccService(mail_alarm,phone_alarm,online_bill,check_owner)))
[] (online_bill;SstccService(mail_alarm,phone_alarm,online_bill,check_owner))
[] (check_owner;SstccService(mail_alarm,phone_alarm,online_bill,check_owner))
endproc
endspec

```

FIGURA 2.2 – ESPECIFICAÇÃO LOTOS DO SSTCC<sup>27</sup>.

<sup>26</sup> Especificada no mais alto nível de abstração.

<sup>27</sup> Especificada no mais alto nível de abstração.

O comportamento do sistema SSTCC é definido pelo processo `SstccService`, o qual pode executar uma ação na porta `mail_alarm`, para o envio de uma denúncia pelo correio comum (considera-se que essa ação é sempre possível).

Em seguida, tem-se uma escolha indeterminística com duas opções. A primeira opção indica o envio de uma denúncia pelo telefone celular, através de uma ação na porta `phone_alarm` (admite-se que essa ação nem sempre é possível), e, em seguida, o processo `SstccService` é chamado, recursivamente, para poder transmitir outra opção. Esta outra opção, por sua vez, trata a situação na qual não foi possível enviar a denúncia pelo telefone celular (por motivo de defeito do aparelho, por estar fora de área, por estar desligado ou por ser chamado em horário inadequado, por exemplo)<sup>28</sup>. Dessa forma, após algum tempo de espera, ocorre uma ação interna `i` (não-observável) e o processo `SstccService` é chamado recursivamente.

Uma representação gráfica dessa especificação em forma de árvore de processos pode ser observada na Figura 2.3 a seguir.

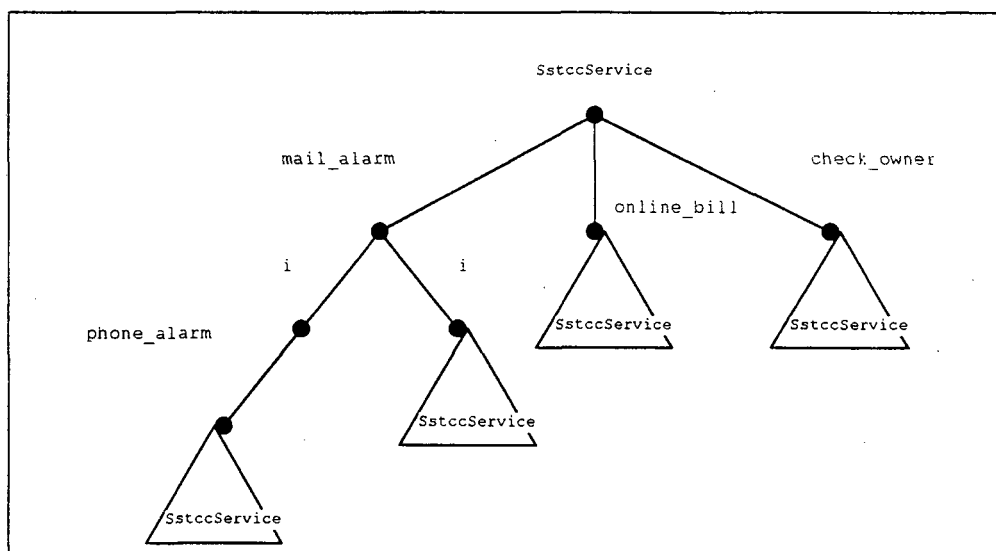


FIGURA 2.3 – REPRESENTAÇÃO GRÁFICA DO PROCESSO SstccSERVICE.

A especificação mais abstrata do sistema SSTCC corresponde a uma formalização dos requisitos dos usuários (serviço) desse sistema. Ela serve de base para os refinamentos posteriores da concepção do SSTCC, ao longo do projeto, e é utilizada na prova de correção da especificação final do sistema (as duas especificações devem ser equivalentes quanto à observação – *observational equivalence*).

## 2.3 Especificações de Protocolo do SSTCC

O sistema SSTCC, já representado no nível mais alto de abstração no item 2.3, pode agora, pela especificação `SstccProtocol`, ser detalhado de modo a considerar seus três subsistemas ou componentes: (1) SSCC – Sistema de Segurança Contra Clonagem de Celular, representado pelo processo `SsccClone`; (2) SETWeb – Sistema de Extrato Telefônico via Web, representado pelo processo `SetwebBill`; e (3) SIPI – Sistema de Identificação de Prováveis Inadimplentes, representado pelo processo `SipiImpostor`. Veja a Figura 2.4.

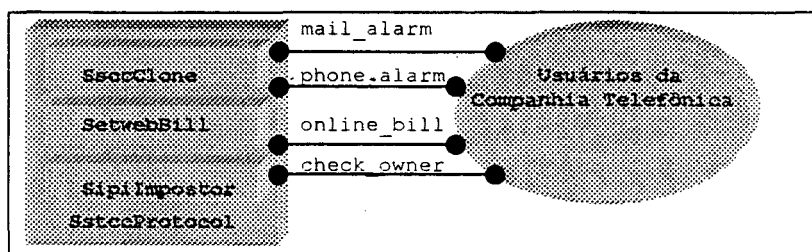


FIGURA 2.4 – O SISTEMA SSTCC E SEUS TRÊS SUBSISTEMAS (SSCC, SETWEB E SIPI).

A especificação LOTOS, correspondente a esse nível de detalhamento, pode ser observada na Figura 2.5.

```

specification
SstccProtocol[mail_alarm,phone_alarm,online_bill,check_owner]:noexit
behaviour
  hide clone_notif,web_notif,impostor_notif in
  SsccClone[mail_alarm,phone_alarm]
  |||SetwebBill[online_bill]
  |||SipiImpostor[check_owner]
  where
  process SsccClone[mail_alarm,phone_alarm]:noexit:= ... endproc
  process SetwebBill[online_bill]:noexit:= ... endproc
  process SipiImpostor[check_owner]:noexit:= ... endproc
endspec
  
```

FIGURA 2.5 – REFINAMENTO DA ESPECIFICAÇÃO SSTCC.

Os três processos apresentados na Figura 2.4 (i.e., `SsccClone`, `SetwebBill` e `SipiImpostor`) são combinados com o uso do operador LOTOS de composição independente (i.e., `|||`), por constituírem processos distribuídos, executados em paralelo, que não compartilham portas (i.e., eventos) entre si. Cada um desses três processos é detalhado a seguir.

<sup>28</sup> Assume-se que o fraudador não utiliza o telefone para receber chamadas (mas somente para realizar chamadas), pois essa é a realidade. Inclusive, é possível programar o telefone clonado para receber chamadas através de um número diferente (i.e., receber não as chamadas que seriam para o usuário legítimo que teve seu aparelho clonado, mas receber chamadas através de um número escolhido pelo fraudador, geralmente do lseu celular legítimo).

### 2.3.1 Especificação do Processo SsccClone

O subsistema SSCC, representado pelo processo LOTOS `SsccClone`, pode ser detalhado de modo a considerar dois dos seus componentes mais importantes: o conjunto de sítios gerenciados, representado pelo processo `CloneCorbaAgentsSet`, e o gerente, representado pelo processo `CloneCorbaManager`. Veja a Figura 2.6.

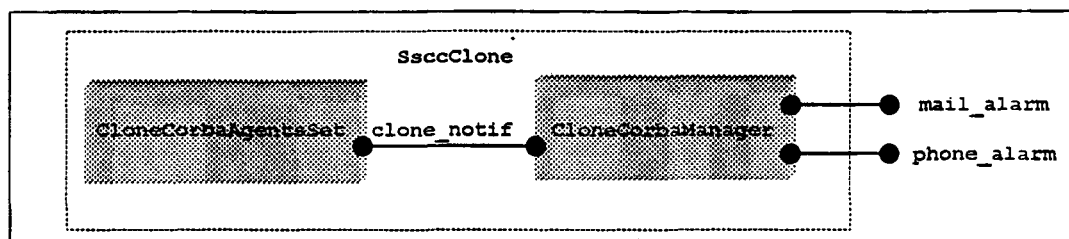


FIGURA 2.6 – DOIS PRINCIPAIS PROCESSOS DO SsccClone.

O processo `CloneCorbaAgentsSet` pode usar a porta `clone_notif` para enviar notificações ao processo `CloneCorbaManager`. Esse processo, por sua vez, após receber uma notificação (através da porta `clone_notif`) pode enviar alarmes aos usuários (através das portas `mail_alarm` e `phone_alarm`). Identifica-se por `SsccClone` a especificação LOTOS dessa concepção refinada. Veja a Figura 2.7.

```

process SsccClone[mail_alarm,phone_alarm]:noexit:=
  hide clone_notif in
  cloneCorbaAgentsSet[clone_notif][clone_notif]
  cloneCorbaManager[clone_notif,mail_alarm,phone_alarm]
  where
  process CloneCorbaAgentsSet[clone_notif]:noexit:=
    ...
  endproc
  process CloneCorbaManager[clone_notif,mail_alarm,phone_alarm]noexit:=
    ...
  endproc
endproc

```

FIGURA 2.7 – PROCESSO LOTOS SsccClone<sup>29</sup>.

O comportamento do processo `CloneCorbaAgentsSet` pode ser especificado em LOTOS como segue: `clone_notif; CloneCorbaAgentsSet [clone_notif]` de modo que podem ocorrer notificações sucessivas, permanentemente. Veja a Figura 2.8.

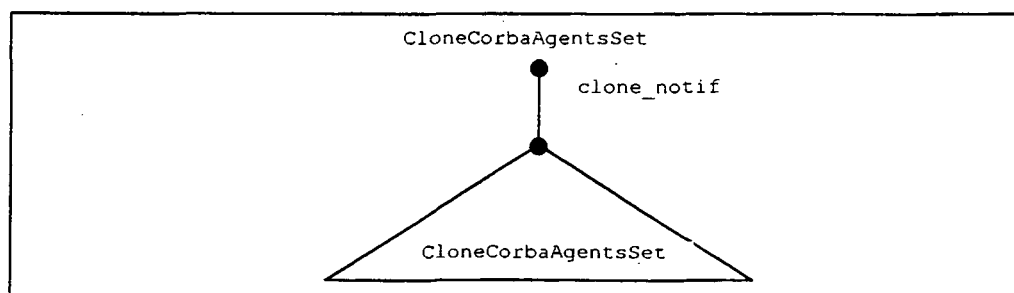


FIGURA 2.8 – PROCESSO CLONECORBAAGENTSET.

<sup>29</sup> Especificada no mais alto nível de abstração.

Já o processo `CloneCorbaManager` pode ter o seu comportamento especificado em LOTOS do modo seguinte:

```
clone_notif;mail_alarm;
(phone_alarm;CloneCorbaManager[clone_notif,mail_alarm,phone_alarm]
[i;CloneCorbaManager[clone_notif,mail_alarm,phone_alarm])
```

e, assim, consegue transmitir alarmes aos usuários após receber notificações. Veja a Figura 2.9.

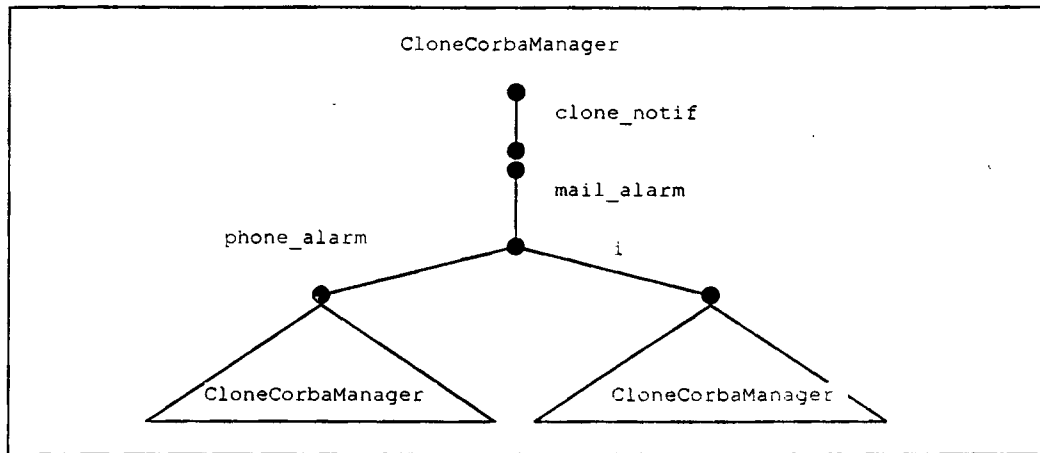


FIGURA 2.9 – PROCESSO CLONECORBAMANAGER.

Os processos `CloneCorbaAgentsSet` e `CloneCorbaManager` são combinados com a utilização do operador de composição geral `| [... ] |`. Nessa combinação fica explicitado que os dois processos compartilham todos os eventos que ocorrem na porta `clone_notif`.

O processo `CloneCorbaAgentsSet` (conjunto de sítios gerenciados) reúne várias instâncias de um mesmo modelo de sítio gerenciado. Cada uma dessas instâncias corresponde a um processo LOTOS que se comunica com o processo `CloneCorbaManager` (gerente do sistema) através da porta `clone_notif`. Veja a Figura 2.10.

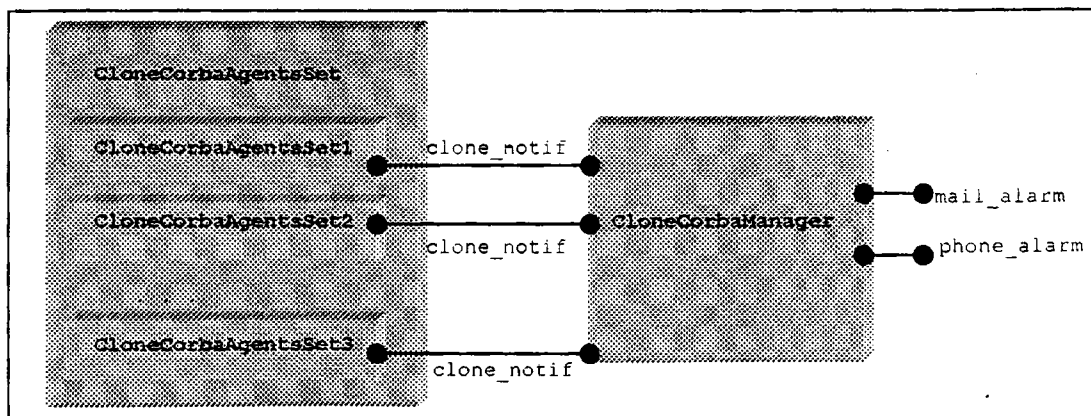


FIGURA 2.10 – DETALHAMENTO DO PROCESSO CLONECORBAAGENTSSET.

Considerando que cada sítio gerenciado atua, isoladamente, para o envio de notificações de possíveis fraudes para o gerente, então pode-se usar o operador de composição independente ( $|||$ ) para combiná-los, obtendo-se a seguinte representação LOTOS:

```
CloneCorbaAgentsSet1 [clone_notif] ||| CloneCorbaAgentsSet2 [clone_notif] |||
. . . ||| CloneCorbaAgentsSetn [clone_notif]
```

O modelo adotado para a representação dos sítios gerenciados é composto de três elementos principais: um agente de gerenciamento, representado pelo processo `CloneCorbaAgentj`; um arquivo de referências (*Baseline*), representado pelo processo `UsersPatternsFilej`; e um arquivo de chamadas telefônicas, representado pelo processo `OnlineCallsFilej`. Desse modo, o refinamento de um desses sítios gerenciados (por exemplo, o sítio *j*) pode ser realizado como mostra a Figura 2.11.

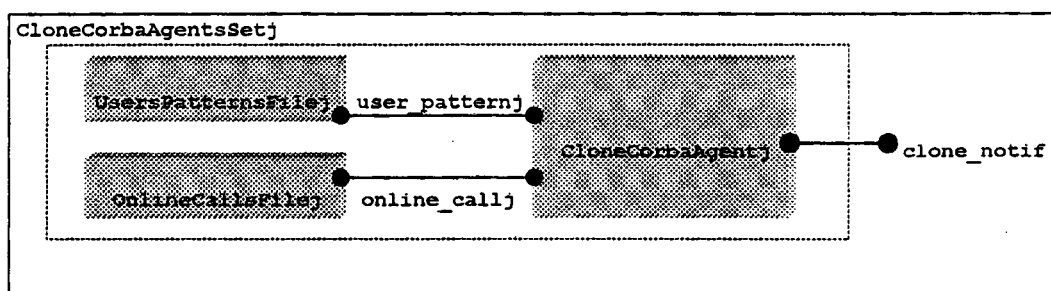


FIGURA 2.11 – DETALHAMENTO DO PROCESSO CLONECORBAAGENTSSETJ.

Na Figura 2.11, o processo `CloneCorbaAgentsSetj` representa um sítio de gerenciamento típico, com os seus três elementos principais. A especificação formal, LOTOS, da arquitetura desse sítio, pode ser observada na Figura 2.12.

```
process CloneCorbaAgentsSetj[clone_notif]:noexit:=
  hide user_patternj,online_callj in
  (UsersPatternsFilej[user_patternj]|||OnlineCallsFilej[online_callj])
  |[user_patternj,online_callj]|
  CloneCorbaAgentj[user_patternj,online_callj,clone_notif]
  where
  process UsersPatternsFilej[users_patternj]:noexit:=
    ...
  endproc
  process OnLineCallsFilej[online_callj]:noexit:=
    ...
  endproc
  process CloneCorbaAgentj[user_patternj,online_callj,clone_notif]:noexit:=
    ...
  endproc
endproc
```

FIGURA 2.12 – PROCESSO CLONECORBAAGENTSSETJ.

O emprego do operador `hide...in`, na especificação detalhada do processo `CloneCorbaAgentsSetj`, permite comparar essa especificação com outra, mais abstrata, do mesmo sítio, que não emprega tal operador. Os processos `UsersPatternsFilej` e `OnLineCallsFilej` atuam independentemente. Considerados em conjunto, esses dois processos compartilham eventos, nas portas `user_patternj` e `online_callj`, com o processo `CloneCorbaAgentj`.

O processo `UsersPatternsFilej` pode executar uma seqüência infinita de eventos na sua porta `user_patternj`, como segue:

```
user_patternj; UsersPatternsFilej[user_patternj]
```

De modo semelhante, o processo `OnLineCallsFilej` pode repetir ações na porta `online_callj`, como segue:

```
online_callj; OnLineCallsFilej[online_callj]
```

Os processos `UsersPatternsFilej` e `OnLineCallsFilej` apresentam comportamentos simples. Já o processo `CloneCorbaAgentj`, entretanto, pode assumir um comportamento um pouco mais complexo: após uma consulta ao arquivo de chamadas telefônicas (com uma ação na porta `online_callj`) o `CloneCorbaAgentj` verifica as características do usuário autorizado, através de um acesso à *Baseline* (com um evento na porta `user_patternj`). Dá-se, então, uma escolha indeterminística com duas alternativas. Essa escolha indeterminística é resolvida, internamente, pelo próprio processo `CloneCorbaAgentj`, como representado a seguir:

```
online_callj;base_j;
(i; CloneCorbaAgentj [user_patternj, online_callj, clone_notif]
 [] i; clone_notif; CloneCorbaAgentj [user_patternj, online_callj, clone_notif])
```

Na primeira alternativa, considera-se a ocorrência de um evento interno, `i`, que representa o caso em que nada de anormal é detectado. Nesse caso, o processo `CloneCorbaAgentj` é chamado recursivamente. Já a segunda alternativa representa o comportamento do `CloneCorbaAgentj` quando há indícios de anormalidade.



Com o objetivo de se considerar um caso simples, pode-se formar, como ilustração do SSCC detalhado, o caso em que se tem o conjunto de sítios gerenciados com apenas dois desses sítios (i.e., CloneCorbaAgent1 e CloneCorbaAgent2). Veja a Figura 2.13.

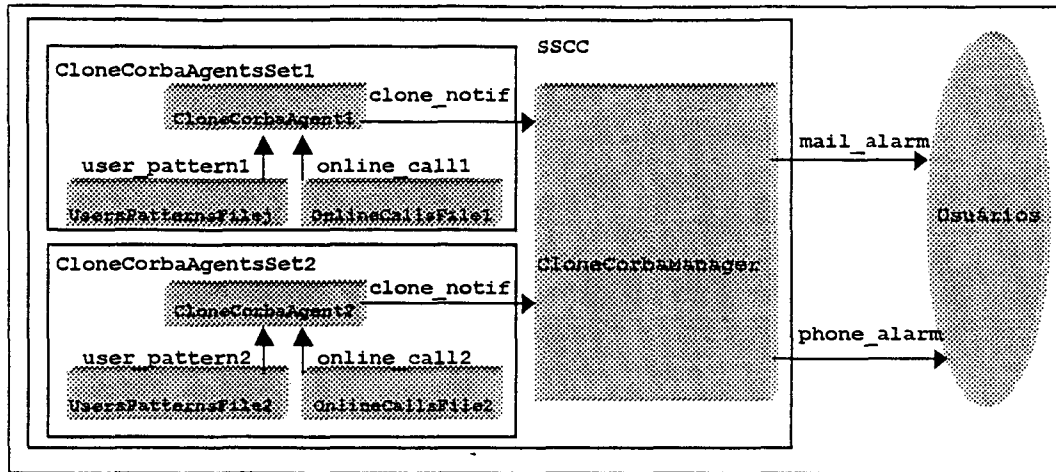


FIGURA 2.13 – REPRESENTAÇÃO DO SSCC DETALHADO COM DOIS AGENTES GERENCIADOS.

A especificação do processo LOTOS correspondente a esta representação gráfica da Figura 2.13, contendo dois sítios (agentes distribuídos), pode ser observada na Figura 2.14.

```

process SsccClone(mail_alarm,phone_alarm):noexit:=
  hide clone_notif in
  CloneCorbaAgentsSet[clone_notif][clone_notif]
  CloneCorbaManager[clone_notif,mail_alarm,phone_alarm]
  where
  process CloneCorbaAgentsSet[clone_notif]:noexit:=
    CloneCorbaAgentsSet1[clone_notif]||CloneCorbaAgentsSet2[clone_notif]
    where
    process CloneCorbaAgentsSet1[clone_notif]:noexit:=
      hide user_pattern1,online_call1 in
      (UsersPatternsFile1[user_pattern1]||OnlineCallsFile1[online_call1])
      |[user_pattern1,online_call1]|
      CloneCorbaAgent1[user_pattern1,online_call1,clone_notif]
      where
      process UsersPatternsFile1[users_pattern1]:noexit:=
        users_pattern1;UsersPatternsFile1[users_pattern1]
      endproc
      process OnLineCallsFile1[online_call1]:noexit:=
        online_call1;OnLineCallsFile1[online_call1]
      endproc
      process CloneCorbaAgent1[user_pattern1,online_call1,clone_notif]:noexit:=
        online_call1;user_pattern1;
        {i;CloneCorbaAgent1[user_pattern1,online_call1,clone_notif]
        []i;clone_notif;CloneCorbaAgent1[user_pattern1,online_call1,clone_notif]}
      endproc
    endproc
    process process CloneCorbaAgentsSet2[clone_notif]:noexit:=
      hide user_pattern2,online_call2 in
      (UsersPatternsFile2[user_pattern2]||OnLineCallsFile2[online_call2])
      |[user_pattern2,online_call2]|
      CloneCorbaAgent2[user_pattern2,online_call2,clone_notif]
      where
      process UsersPatternsFile2[user_patter2]:noexit:=
        user_pattern2;UsersPatternsFile2[user_pattern2]
      endproc
      process OnLineCallsFile2[online_call2]:noexit:=
        online_call2;OnLineCallsFile2[online_call2]
      endproc
      process CloneCorbaAgent2[user_pattern2,online_call2,clone_notif]:noexit:=
        online_call2;user_pattern2;
        {i;CloneCorbaAgent2[user_pattern2,online_call2,clone_notif]
        []i;clone_notif;CloneCorbaAgent2[user_pattern2,online_call2,clone_notif]}
      endproc
    endproc
  endproc
endproc

```

```

endproc
process CloneCorbaManager[clone_notif,mail_alarm,phone_alarm]noexit:=
  clone_notif;mail_alarm;
  (phone_alarm;CloneCorbaManager[clone_notif,mail_alarm,phone_alarm]
  [!i:CloneCorbaManager[clone_notif,mail_alarm,phone_alarm]])
endproc
endproc

```

FIGURA 2.14 – ESPECIFICAÇÃO LOTOS DO SSCC COM DOIS AGENTES.

Esta especificação apresentada na Figura 2.14, com dois agentes distribuídos, é apropriada para uma companhia telefônica que possui dois sítios a serem gerenciados, i.e., duas centrais telefônicas. Ou seja, a especificação deve conter tantos sítios (agentes distribuídos) quanto a quantidade de centrais que a companhia possua.

### 2.3.2 Especificação do Processo SetwebBill

O subsistema SETWeb, representado pelo processo LOTOS setwebBill, pode ser detalhado de modo a considerar dois dos seus componentes mais importantes: o conjunto de sítios gerenciados, representado pelo processo BillCorbaAgentsSet, e o gerente, representado pelo processo BillCorbaManager. Veja a Figura 2.15.

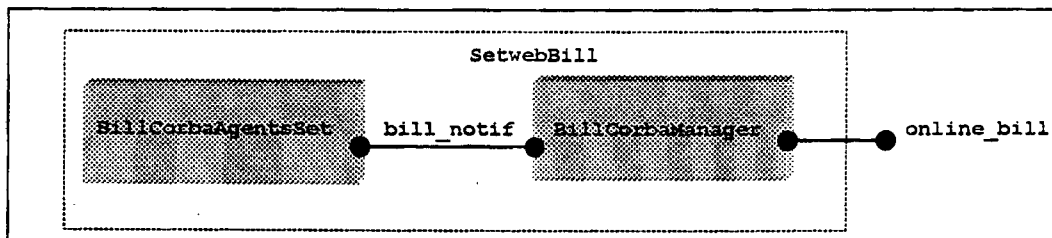


FIGURA 2.15 – DOIS PRINCIPAIS PROCESSOS DO SETWEBBILL.

O processo BillCorbaAgentsSet pode usar a porta `bill_notif` para transmitir notificações ao processo BillCorbaManager, i.e., atualizar os dados calculados nos agentes para o gerente. O gerente, por sua vez, após receber uma notificação (através da porta `bill_notif`) disponibiliza as informações aos usuários através de um WebServer (através da porta `online_bill`). Identifica-se por `SetwebBill` a especificação LOTOS dessa concepção refinada. Veja a Figura 2.16.

```

process SetwebBill[online_bill]:noexit:=
  hide bill_notif in
  BillCorbaAgentsSet[bill_notif]||[bill_notif]||
  BillCorbaManager[bill_notif]
  where
  process BillCorbaAgentsSet[bill_notif]:noexit:= ... endproc
  process BillCorbaManager[bill_notif]:noexit:= ... endproc
endproc

```

FIGURA 2.16 – ESPECIFICAÇÃO LOTOS DO SETWEBBILL<sup>30</sup>.

<sup>30</sup> Especificada no mais alto nível de abstração.

O comportamento do processo `BillCorbaAgentsSet` pode ser especificado, em LOTOS, como segue:

```
Bill_notif; BillCorbaAgentsSet [bill_notif]
```

de modo que podem ocorrer notificações sucessivas permanentemente.

O processo `BillCorbaManager` pode ter o seu comportamento especificado, em LOTOS, do modo seguinte:

```
bill_notif; online_bill; BillCorbaManager[bill_notif, online_bill]
```

e, assim, consegue disponibilizar as informações referentes à conta telefônica aos seus usuários, após recebê-las dos diversos agentes distribuídos (através de notificações).

Os processos `BillCorbaAgentsSet` e `BillCorbaManager` são combinados com a utilização do operador de composição geral `| [...]`. Nessa combinação, fica explicitado que os dois processos compartilham todos os eventos que ocorrem na porta `bill_notif`.

O processo `BillCorbaAgentsSet` (conjunto de sítios gerenciados) reúne várias instâncias de um mesmo modelo de sítio gerenciado. Cada uma dessas instâncias corresponde a um processo LOTOS, que se comunica com o processo `BillCorbaManager` (gerente do sistema), através da porta `bill_notif`. Veja a Figura 2.17.

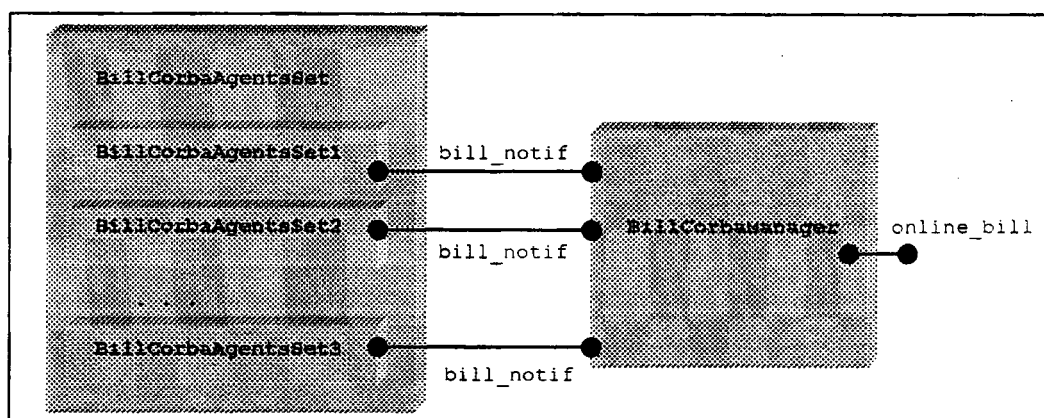


FIGURA 2.17 – DETALHAMENTO DO PROCESSO `BILLCORBAAGENTSSET`.

Considerando que cada sítio gerenciado atua, isoladamente, para o envio de notificações de possíveis fraudes para o Gerente, então pode-se usar o operador de composição independente (`|||`) para combiná-los, obtendo-se a seguinte representação LOTOS:

```
BillCorbaAgentsSet1 [bill_notif] ||| BillCorbaAgentsSet2 [bill_notif] |||
. . . ||| BillCorbaAgentsSetn [bill_notif]
```

O modelo adotado para a concepção dos sítios gerenciados é composto de três elementos principais: um agente de gerenciamento, representado pelo processo `BillCorbaAgentj`; um arquivo onde são gravadas as informações já calculadas que irão compor a conta telefônica, representado pelo processo `BillDBj`; e um arquivo de chamadas telefônicas, representado pelo processo `OnlineCallsFilej`. Desse modo, o refinamento de um desses sítios gerenciados (por exemplo, o sítio `j`) pode ser realizado como mostra a Figura 2.18.

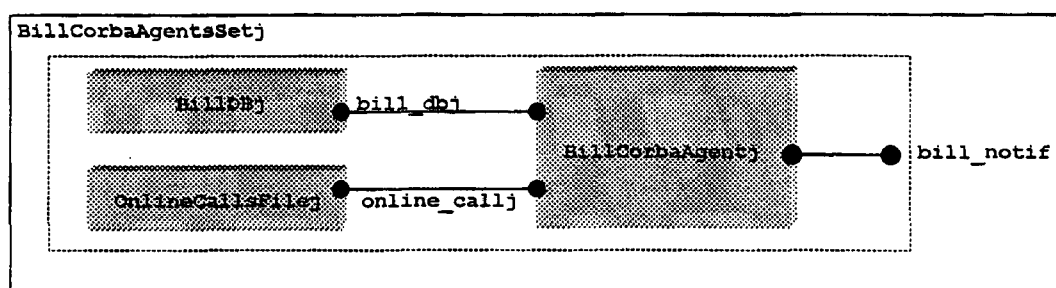


FIGURA 2.18 – DETALHAMENTO DO PROCESSO `BILLCORBAAGENTSSETJ`.

O processo `BillCorbaAgentsSetj` representa um sítio de gerenciamento típico, com os seus três elementos principais (ou seja, `BillCorbaAgentj`, `OnLineCallsFilej` e `BillDBj`). A especificação formal LOTOS da arquitetura desse sítio pode ser observada na Figura 2.19.

```

process BillCorbaAgentsSetj[bill_notif]:noexit:=
  hide bill_dbj,online_callj in
    (BillDBj[bill_dbj]|||OnlineCallsFilej[online_callj])
    |[bill_dbj,online_callj]|
    BillCorbaAgentj[bill_dbj,online_callj,bill_notif]
  where
    process BillDBj[bill_dbj]:noexit:=
      ...
    endproc
    process OnLineCallsFilej[online_callj]:noexit:=
      ...
    endproc
    process BillCorbaAgentj[bill_dbj,online_callj,bill_notif]:noexit:=
      ...
    endproc
  endproc
endproc

```

FIGURA 2.19 – ESPECIFICAÇÃO LOTOS DO PROCESSO `BILLCORBAAGENTSSETJ`.

O emprego do operador `hide...in` nessa especificação do processo `BillCorbaAgentsSetj` permite comparar essa especificação com a especificação de serviço correspondente. Os processos `BillDBj` e `OnLineCallsFilej` atuam independentemente. Considerados em conjunto, esses dois processos compartilham eventos, nas portas `bill_dbj` e `online_callj` com o processo `BillCorbaAgentj`.

O processo `OnLineCallsFilej` pode executar uma seqüência infinita de eventos na sua porta `online_callj`:

```
online_call1j;OnLineCallsFile1j [online_call1j]
```

Isto é, o arquivo de chamadas *on-line* é lido constantemente.

De modo semelhante, o processo BillDBj pode repetir ações na porta bill\_dbj:

```
bill_dbj;BillDBj [bill_dbj]
```

Isto é, as chamadas (*on-line*) são constantemente calculadas e gravadas em uma base de dados.

Os processos BillDBj e OnLineCallsFilej apresentam comportamentos simples, i.e., fazem a leitura de arquivo e a gravação em base de dados, respectivamente, ambos representados por apenas um evento (ou-seja, uma porta de comunicação). Já o processo BillCorbaAgentj executa três eventos: (1) lê as chamadas on-line; (2) calcula e grava as informações na base de dados local – em cada sítio; e (3) envia as informações calculadas para o gerente (que as disponibiliza aos usuários através de um Web Server):

```
online_call1j;bill_dbj;bill_notif;
BillCorbaAgentj[online_call1j,bill_dbj,bill_notif]
```

Assim, os agentes ficam, constantemente, lendo o arquivo de chamadas (evento na porta online\_call1j), calculando e gravando localmente (evento na porta bill\_dbj), e enviando os dados ao gerente (evento na porta bill\_notif).

Com o objetivo de se considerar um caso simples, pode-se formar, como ilustração do SETWeb detalhado, o caso em que se tem o conjunto de sítios gerenciados com apenas dois desses sítios (i.e., BillCorbaAgent1 e BillCorbaAgent2). Veja a Figura 2.20.

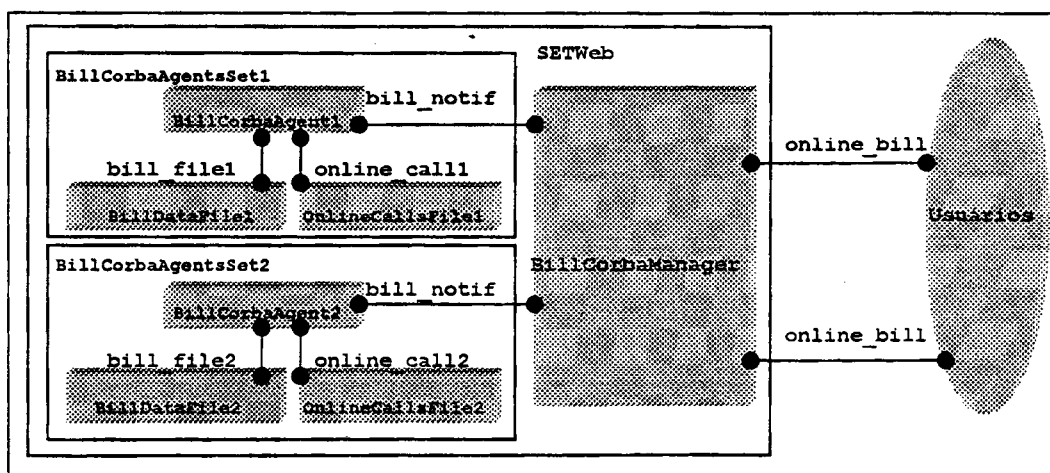


FIGURA 2.20 – REPRESENTAÇÃO DO SETWEB DETALHADO COM DOIS AGENTES GERENCIADOS.

A especificação do processo LOTOS correspondente a esta representação gráfica, contendo dois agentes distribuídos, pode ser observada na Figura 2.21.

```

process SetwebBill[online_bill]:noexit:=
hide web_notif in
BillCorbaAgentsSet[web_notif]||[web_notif]|BillCorbaManager[web_notif,online_bill]
  where
    process BillCorbaAgentsSet[web_notif]:noexit:=
      BillCorbaAgentsSet1[web_notif]|||BillCorbaAgentsSet2[web_notif]
        where
          process BillCorbaAgentsSet1[web_notif]:noexit:=
            hide online_call1,bill_db1 in
              (OnLineCallFile1[online_call1]|||BillDB1[bill_db1])
              |[online_call1,bill_db1]|
            BillCorbaAgent1[online_call1,bill_db1,bill_notif]
              where
                process OnLineCallFile1[online_call1]:noexit:=
                  online_call1;OnLineCallFile1[online_call1]
                endproc
              process BillDB1[bill_db1]:noexit:=
                bill_db1;BillDB1[bill_db1]
              endproc
            process BillCorbaAgent1[online_file1,bill_db1,bill_notif]:noexit:=
              online_file1;bill_db1;bill_notif;BillCorbaAgent1[online_call1,bill_db1,bill_notif]
            endproc
          endproc
        process BillCorbaAgentsSet2[web_notif]:noexit:=
          hide online_call2,bill_db2 in
            (OnLineCallFile2[online_call2]|||BillDB2[bill_db2])
            |[online_call2,bill_db2]|
          BillCorbaAgent2[online_call2,bill_db2,bill_notif]
            where
              process OnLineCallFile2[online_call2]:noexit:=
                online_call2;OnLineCallFile2[online_call2]
              endproc
            process BillDB2[bill_db2]:noexit:=
              bill_db2;BillDB2[bill_db2]
            endproc
          process BillCorbaAgent2[online_file2,bill_db2,bill_notif]:noexit:=
            online_file2;bill_db2;bill_notif;BillCorbaAgent2[online_call2,bill_db2,bill_notif]
          endproc
        endproc
      process BillCorbaManger[bill_notif,online_bill]noexit:=
        bill_notif;online_bill;BillCorbaManger[bill_notif,online_bill]
      endproc
    endproc
  endproc

```

FIGURA 2.21 – ESPECIFICAÇÃO LOTOS DO SETWEB COM DOIS AGENTES DISTRIBUÍDOS.

Esta especificação, com dois agentes distribuídos, permite que novos agentes sejam facilmente adicionados utilizando-se o operador de paralelismo independente (|||).

### 2.3.3 Especificação do Processo Sipi Impostor

O subsistema SIPI, representado pelo processo LOTOS *SipiImpostor*, pode ser detalhado de modo a considerar dois componentes: o conjunto de sítios gerenciados, representado pelo processo *ImpostorCorbaAgentsSet*, e o gerente, representado pelo processo *ImpostorCorbaManager*. Veja a Figura 2.22.

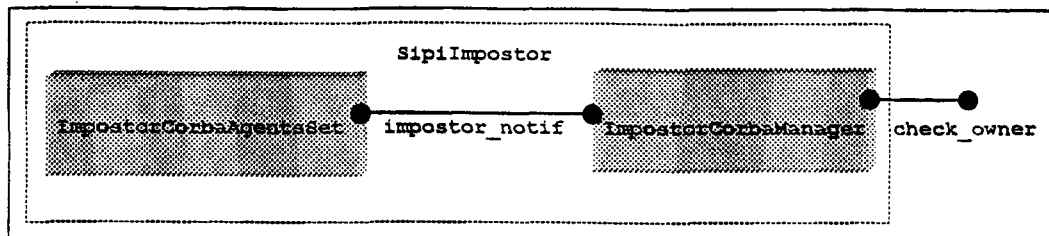


FIGURA 2.22 – DOIS PROCESSOS DO SipiIMPOSTOR.

O processo `ImpostorCorbaAgentsSet` pode usar a porta `impostor_notif` para transmitir notificações ao processo `ImpostorCorbaManager`. Este, por sua vez, após receber uma notificação (através da porta `impostor_notif`), pode verificar a verdadeira identidade dos usuários suspeitos de inadimplência (através da porta `check_owner`). Identifica-se por `SipiImpostor` a especificação LOTOS dessa concepção refinada. Veja a Figura 2.23.

```

process SipiImpostor[check_owner]:noexit:=
  hide impostor_notif in
  ImpostorCorbaAgentsSet[impostor_notif] |[impostor_notif]|
  ImpostorCorbaManager[impostor_notif,check_owner]
  where
  process ImpostorCorbaAgentsSet[impostor_notif]:noexit:= ... endproc
  process ImpostorCorbaManager[impostor_notif,check_owner]:noexit:= ... endproc
endproc
  
```

FIGURA 2.23 – ESPECIFICAÇÃO LOTOS DO SipiIMPOSTOR<sup>31</sup>.

O comportamento do processo `ImpostorCorbaAgentsSet` pode ser especificado, em LOTOS, como segue:

```
impostor_notif; ImpostorCorbaAgentsSet [impostor_notif]
```

de modo que podem existir notificações sucessivas permanentemente.

O processo `ImpostorCorbaManager` pode ter o seu comportamento especificado em LOTOS do modo seguinte:

```
impostor_notif;check_owner;ImpostorCorbaManager[impostor_notif,check_owner]
```

e, assim, consegue sondar usuários (prováveis inadimplentes) após receber notificações.

Os processos `ImpostorCorbaAgentsSet` e `ImpostorCorbaManager` são combinados, com a utilização do operador de composição geral `|[...]|`. Nessa combinação, fica explicitado que os dois processos compartilham todos os eventos que ocorrem na porta `impostor_notif`. O uso do operador `hide..in` oculta a porta interna `impostor_notif`, o que torna possível comparar esta especificação `SipiImpostor` com a especificação inicial de serviço e provar (através de equivalência de observação) que esta última é um refinamento correto da primeira.

O processo `ImpostorCorbaAgentsSet` (Conjunto de Sítios Gerenciados) reúne várias instâncias de um mesmo modelo de sítio gerenciado. Cada uma dessas instâncias corresponde a um processo LOTOS, que se comunica com o processo `ImpostorCorbaManager` (Gerente do Sistema), através da porta `impostor_notif`. Veja a Figura 2.24.

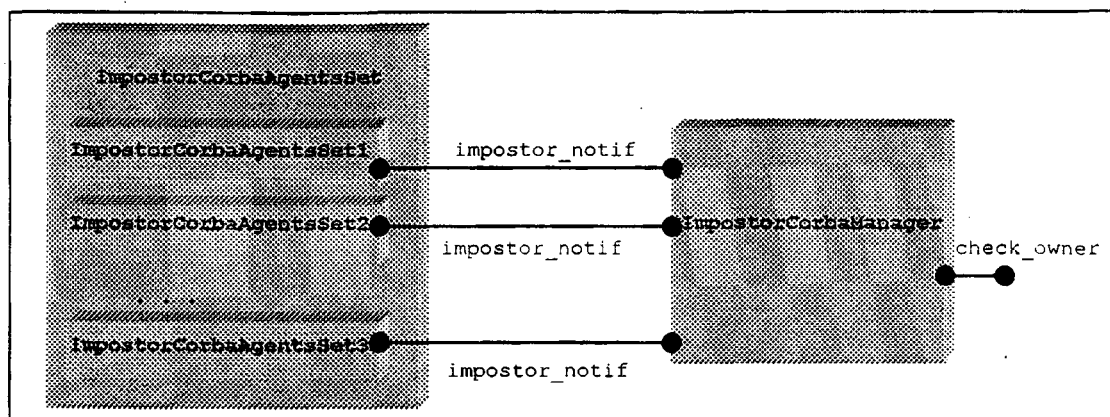


FIGURA 2.24 – DETALHAMENTO DO PROCESSO `IMPOSTORCORBAAGENTSSET`.

Considerando que cada sítio gerenciado atua, isoladamente, para o envio de notificações de possíveis fraudes/inadimplentes ao Gerente, então pode-se usar o operador de composição independente (`|||`) para combiná-los, obtendo-se a seguinte representação LOTOS:

```
ImpostorCorbaAgentsSet1    [impostor_notif]    |||    ImpostorCorbaAgentsSet2
[impostor_notif]    |||    . . .    |||    ImpostorCorbaAgentsSetn [impostor_notif]
```

O modelo adotado para a concepção dos sítios gerenciados é composto de quatro elementos principais: um agente de gerenciamento, representado pelo processo `ImpostorCorbaAgentj`; um arquivo de referências, contendo os padrões dos usuários, representado pelo processo `UsersPatternsFilej`; um arquivo de chamadas telefônicas, representado pelo processo `OnlineCallsFilej`; e um arquivo contendo os usuários inadimplentes já conhecidos, representado pelo processo `ImpostorsPatternsFilej` (na verdade, o processo representa as operações sobre este arquivo). Desse modo, o refinamento de um desses sítios gerenciados (por exemplo, o sítio `j`) pode ser realizado como mostra a Figura 2.25.

<sup>31</sup> Especificada no mais alto nível de abstração.



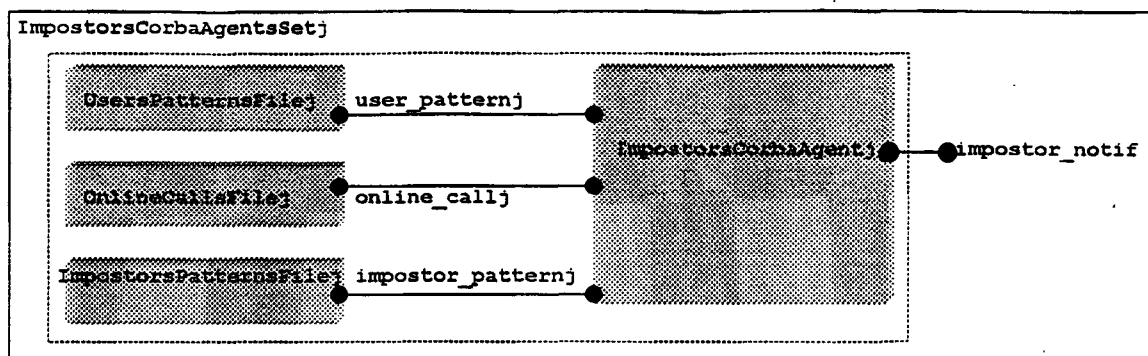


FIGURA 2.25 – DETALHAMENTO DO PROCESSO IMPOSTORSCORBAAGENTSSETJ.

O processo `ImpostorCorbaAgentsSetj` representa um sítio de gerenciamento típico, com os seus três elementos principais. A especificação formal, LOTOS, da arquitetura desse sítio pode ser observada na Figura 2.26.

```

process ImpostorCorbaAgentsSet[impostor_notif]:noexit:=
  ImpostorCorbaAgentsSetj[impostor_notif]|||ImpostorCorbaAgentsSet2[impostor_notif]
  where
  process ImpostorCorbaAgentsSetj[impostor_notif]:noexit:=
    hide user_patternj,impostor_patternj,online_callj in
      (UsersPatternsFilej[user_patternj]|||ImpostorsPatternsFilej[impostor_patternj])
      |||OnLineCallsFilej[online_callj])
      |[user_patternj,impostor_patternj,online_callj]|
      ImpostorCorbaAgentj[user_patternj,online_callj,impostor_patternj,impostor_notif]
      where
        process UsersPatternsFilej[user_patternj]:noexit:=
          ...
        endproc
        process ImpostorsPatternsFilej [user_patternj]:noexit:=
          ...
        endproc
        process OnLineCallsFilej[online_callj]:noexit:=
          ...
        endproc
        process ImpostorCorbaAgentj
          [user_patternj,online_callj,impostor_patternj,impostor_notif]:noexit:=
            ...
          endproc
        endproc
      endproc
    endproc
  endproc

```

FIGURA 2.26 – ESPECIFICAÇÃO LOTOS DO PROCESSO IMPOSTORCORBAAGENTSSETJ.

O operador `hide...in` permite comparar e provar a equivalência desta especificação com outra mais abstrata (i.e., com a especificação de serviço). Os processos `UsersPatternsFilej`, `ImpostorsPatternsFilej` e `OnLineCallsFilej` atuam independentemente. Considerados em conjunto, esses três processos compartilham eventos, nas portas `user_patternj`, `impostor_patternj` e `online_callj` com o processo `ImpostorCorbaAgentj`.

O processo `UsersPatternsFilej` pode executar uma seqüência infinita de eventos na sua porta `user_patternj`:

```
user_patternj;UsersPatternsFilej[user_patternj]
```

Isto é, verificando o arquivo de padrões dos usuários.

De modo semelhante, o processo `ImpostorPatternsFilej` pode repetir ações na porta `impostor_patternj`:

```
impostor_patternj;ImpostorsPatternsFilej[impostor_patternj]
```

Isto é, verificando o arquivo de padrões dos inadimplentes conhecidos.

E de modo semelhante, o processo `OnLineCallsFilej` pode repetir ações na porta `online_callj`:

```
online_callj;OnLineCallsFilej[online_callj]
```

Isto é, verificando as ligações *on-line*.

Os processos `UsersPatternsFilej`, `ImpostorsPatternsFilej` e `OnLineCallsFilej` apresentam comportamentos simples, i.e., apenas um evento. Já o processo `ImpostorCorbaAgentj`, entretanto, pode assumir um comportamento um pouco mais complexo: após uma consulta ao Arquivo de Chamadas Telefônicas (com uma ação na porta `online_callj`), o `ImpostorCorbaAgentj` verifica as características do usuário através de um acesso à BaseLine (com um evento na porta `user_patternj`) e também verifica se existe similaridade dessa ligação com o padrão de um inadimplente, através do acesso ao arquivo de inadimplentes conhecidos (com um evento na porta `impostor_patternj`). Dá-se, então, uma escolha indeterminística com duas alternativas. Essa escolha indeterminística é resolvida internamente pelo próprio processo `ImpostorCorbaAgentj`:

```
online_callj;user_patternj;impostor_patternj;
```

```
(i;ImpostorCorbaAgentj[online_callj,user_patternj,impostor_patternj,impostor_notifj]
[]i;impostor_notif;ImpostorCorbaAgentj[online_callj,user_patternj,impostor_patternj,impostor_notif])
```

Na primeira alternativa, considera-se a ocorrência de um evento interno, *i*, que representa o caso em que nada de anormal é detectado. Nesse caso, o processo `ImpostorCorbaAgentj` é chamado recursivamente. Já a segunda alternativa representa o comportamento do `ImpostorCorbaAgentj` quando há indícios de anormalidade.

Com o objetivo de se considerar um caso simples, pode-se formar, como exemplo do SIPI detalhado, o caso em que se tem o conjunto de sítios gerenciados com apenas dois desses sítios (i.e., ImpostorCorbaAgent1 e ImpostorCorbaAgent2). Veja a Figura 2.27.

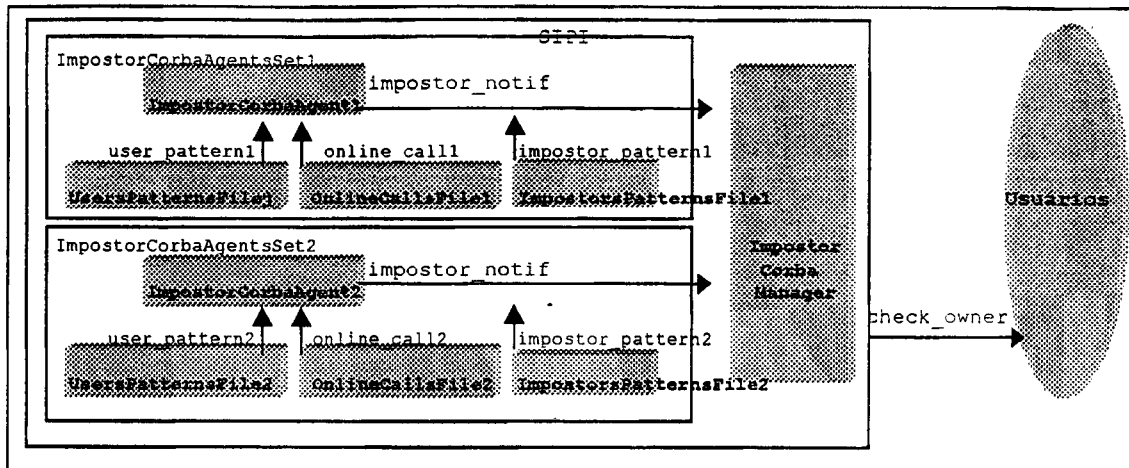


FIGURA 2.27 – REPRESENTAÇÃO DO SIPI DETALHADO COM DOIS AGENTES GERENCIADOS.

A especificação do processo LOTOS correspondente a esta representação gráfica, contendo dois agentes distribuídos, pode ser observada na Figura 2.28.

```

process SipiImpostor(check_owner):noexit:=
  hide impostor_notif in
  ImpostorCorbaAgentsSet{impostor_notif} |[impostor_notif]|
  ImpostorCorbaManager{impostor_notif,check_owner}
  where
  process ImpostorCorbaAgentsSet{impostor_notif}:noexit:=
    ImpostorCorbaAgentsSet1{impostor_notif}|||ImpostorCorbaAgentsSet2{impostor_notif}
    where
    process ImpostorCorbaAgentsSet1{impostor_notif}:noexit:=
      hide user_pattern1,impostor_pattern1,online_call1 in
      (UsersPatternsFile{user_pattern1}|||ImpostorsPatternsFile{impostor_pattern1}
      |||OnLineCallsFile{online_call1})
      |[user_pattern1,impostor_pattern1,online_call1]|
      ImpostorCorbaAgent1[user_pattern1,online_call1,impostor_pattern1,impostor_notif]
      where
      process UsersPatternsFile{user_pattern1}:noexit:=
        user_pattern1; UsersPatternsFile [base_1]
      endproc
      process ImpostorsPatternsFile {user_pattern1}:noexit:=
        user_pattern1; ImpostorsPatternsFile {user_pattern1}
      endproc
      process OnLineCallsFile[online_call1]:noexit:=
        online_call1; OnLineCallsFile[online_call1]
      endproc
    process ImpostorCorbaAgent1
      [user_pattern1,online_call1,impostor_pattern1,impostor_notif]:noexit:=
        online_call1;user_pattern1;impostor_pattern1;
        (i; ImpostorCorbaAgent1[user_pattern1,online_call1,
        impostor_pattern1,impostor_notif]
        [] i;impostor_notif; ImpostorCorbaAgent1[user_pattern1, online_call1,
  
```

```

        impostor_pattern1, impostor_notif))
    endproc
endproc
process process ImpostorCorbaAgent2(impostor_notif):noexit:=
    hide user_pattern2, impostor_pattern2, online_call2 in
        (UsersPatternsFile2[user_pattern2]|||ImpostorsPatternFile2[impostor_pattern2]
|||OnLineCallsFile2[online_call2])|[user_pattern2, impostor_pattern2, online_call2];
ImpostorCorbaAgent2[user_pattern2, impostor_pattern2, online_call2, impostor_notif]
    where
        process UsersPatternsFile2[user_pattern2]:noexit:=
            user_pattern2; UsersPatternsFile2 [base_2]
        endproc
        process ImpostorsPatternsFile2[impostor_pattern2]:noexit:=
            impostor_pattern2; ImpostorsPatternsFile2 [impostor_pattern2]
        endproc
        process OnLineCallsFile2[online_call2]:noexit:=
            online_call2; OnLineCallsFile2[online_call2]
        endproc
        process ImpostorCorbaAgent2[user_pattern2, impostor_pattern2,
            online_call2, impostor_notif]:noexit:=
            online_call2; user_pattern2; impostor_pattern2;
            (i; ImpostorCorbaAgent2[online_call2; user_pattern2;
                impostor_pattern2, impostor_notif]
                []i; impostor_notif; ImpostorCorbaAgent2[online_call2; user_pattern2;
                    impostor_pattern2, impostor_notif])
        endproc
    endproc
endproc
process ImpostorCorbaManager(impostor_notif, check_owner):noexit:=
    impostor_notif;
    (i; check_owner; ImpostorCorbaManager(impostor_notif, check_owner)
    []i; ImpostorCorbaManager(impostor_notif, check_owner))
endproc
endproc
endproc

```

FIGURA 2.28 – ESPECIFICAÇÃO LOTOS REFINADA DO SUBSISTEMA SIPI.

Essa especificação, com dois agentes distribuídos, está relacionada a uma companhia telefônica que possui dois sítios a serem gerenciados, i.e., duas centrais telefônicas. Ou seja, a especificação deve conter tantos agentes distribuídos quanto a quantidade de centrais que a companhia possua.

### 2.3.4 Especificação do Sistema SSTCC

A arquitetura geral do sistema SSTCC, incluindo os três principais processos refinados que a compõem pode ser observada na Figura 2.29.

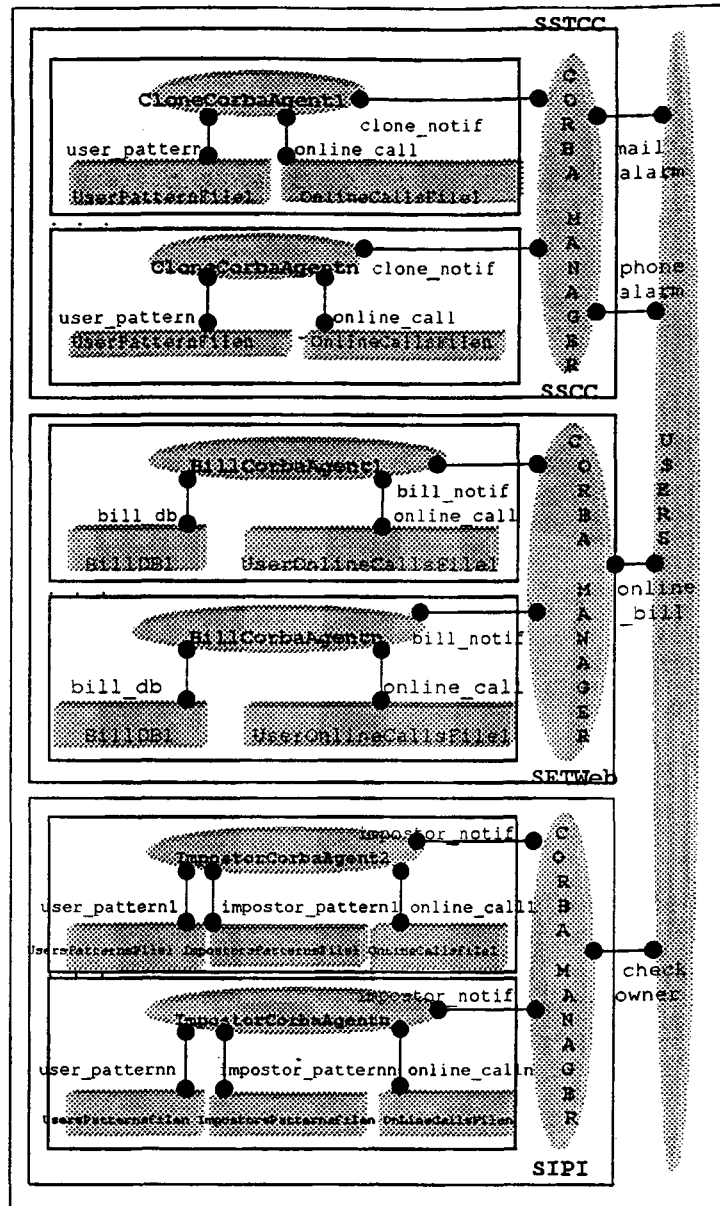


FIGURA 2.29 – ARQUITETURA GERAL DO SSTCC.

A arquitetura representada na Figura 2.29 acima pode ser descrita em LOTOS através da união das especificações refinadas descritas nos itens 2.3.1, 2.3.2 e 2.3.3 como apresentado na especificação *SstccProtocol* da Figura 2.30 a seguir.

```

specification SstccProtocol(mail_alarm,phone_alarm,online_bill,check_owner):noexit
behaviour
hide clone_notif,web_notif,impostor_notif in
SsccClone[mail_alarm,phone_alarm]|||SetwebBill[online_bill]|||SipiImpostor[check_owner]
where
process SsccClone[mail_alarm,phone_alarm]:noexit:=
  hide clone_notif in
  CloneCorbaAgentsSet[clone_notif]||[clone_notif]||
  CloneCorbaManager[clone_notif,mail_alarm,phone_alarm]
  where
  process CloneCorbaAgentsSet[clone_notif]:noexit:=
    CloneCorbaAgentsSet1[clone_notif]|||CloneCorbaAgentsSet2[clone_notif]

```

```

where
process CloneCorbaAgentsSet1[clone_notif]:noexit:=
  hide user_pattern1,online_call1 in
  (UsersPatternsFile1[user_pattern1]|||OnLineCallsFile1[online_call1])
  |[user_pattern1,online_call1]|
CloneCorbaAgent1[user_pattern1,online_call1,clone_notif]
  where
  process UsersPatternsFile1[users_pattern1]:noexit:=
    users_pattern1;UsersPatternsFile1[users_pattern1]
  endproc
  process OnLineCallsFile1[online_call1]:noexit:=
    online_call1;OnLineCallsFile1[online_call1]
  endproc
  process CloneCorbaAgent1[user_pattern1,online_call1,clone_notif]:noexit:=
    online_call1;user_pattern1;
    (i;CloneCorbaAgent1[user_pattern1,online_call1,clone_notif]
    [|i;clone_notif;CloneCorbaAgent1[user_pattern1,online_call1,clone_notif])
  endproc
endproc
process CloneCorbaAgentsSet2[clone_notif]:noexit:=
  hide user_pattern2,online_call2 in
  (UsersPatternsFile2[user_pattern2]|||OnLineCallsFile2[online_call2])
  |[user_pattern2,online_call2]|
CloneCorbaAgent2[user_pattern2,online_call2,clone_notif]
  where
  process UsersPatternsFile2[user_patter2]:noexit:=
    user_pattern2;UsersPatternsFile2[user_pattern2]
  endproc
  process OnLineCallsFile2[online_call2]:noexit:=
    online_call2;OnLineCallsFile2[online_call12]
  endproc
  process CloneCorbaAgent2[user_pattern2,online_call2,clone_notif]:noexit:=
    online_call;user_pattern2;
    (i;CloneCorbaAgent2[user_pattern2,online_call2,clone_notif]
    [|i;clone_notif;CloneCorbaAgent2[user_pattern2,online_call2,clone_notif])
  endproc
endproc
endproc
process CloneCorbaManager[clone_notif,mail_alarm,phone_alarm]noexit:=
  clone_notif;mail_alarm;
  (phone_alarm;CloneCorbaManager{clone_notif,mail_alarm,phone_alarm}
  [|i;CloneCorbaManager{clone_notif,mail_alarm,phone_alarm})
endproc
endproc
process SetwebBill[online_bill]:noexit:=
hide web_notif in BillCorbaAgentsSet[web_notif]|[web_notif]|BillCorbaManager[web_notif,online_bill]
  where
  process BillCorbaAgentsSet[web_notif]:noexit:=
    BillCorbaAgentsSet1[web_notif]|||BillCorbaAgentsSet2[web_notif]
  where
  process BillCorbaAgentsSet1[web_notif]:noexit:=
    hide online_call1,bill_db1 in
    (OnLineCallFile1[online_call1]|||BillDB1[bill_db1])
    |[online_call1,bill_db1]|
    BillCorbaAgent1[online_call1,bill_db1,bill_notif]
    where
    process OnLineCallFile1[online_call1]:noexit:=
      online_call1;OnLineCallFile1[online_call1]
    endproc
    process BillDB1[bill_db1]:noexit:=
      bill_db1;BillDB1[bill_db1]
    endproc
    process BillCorbaAgent1[online_file1,bill_db1,bill_notif]:noexit:=
      online_file1;bill_db1;bill_notif;BillCorbaAgent1[online_call1,bill_db1,bill_notif])
  endproc
endproc
process BillCorbaAgentsSet2[web_notif]:noexit:=
  hide online_call2,bill_db2 in
  (OnLineCallFile2[online_call2]|||BillDb2[bill_db2])
  |[online_call2,bill_db2]|
  BillCorbaAgent2[online_call2,bill_db2,bill_notif]
  where
  process OnLineCallFile2[online_call2]:noexit:=
    online_call2;OnLineCallFile2[online_call2]
  endproc
  process BillDB2[bill_db2]:noexit:=
    bill_db2;BillDB2[bill_db2]
  endproc
  process BillCorbaAgent2[online_file2,bill_db2,bill_notif]:noexit:=
    online_file2;bill_db2;bill_notif;BillCorbaAgent2[online_call2,bill_db2,bill_notif])
  endproc
endproc
process BillCorbaManger[bill_notif,online_bill]noexit:=
  bill_notif;online_bill;BillCorbaManger{bill_notif,online_bill}
endproc
endproc

```

```

process SipiImpostor[check_owner]:noexit:=
  hide impostor_notif in
  ImpostorCorbaAgentsSet[impostor_notif] |[impostor_notif]|
  impostorCorbaManager[impostor_notif,check_owner]
  where
  process ImpostorCorbaAgentsSet[impostor_notif]:noexit:=
    ImpostorCorbaAgentsSet1[impostor_notif]|||ImpostorCorbaAgentsSet2[impostor_notif]
    where
    process ImpostorCorbaAgentsSet1[impostor_notif]:noexit:=
      hide user_pattern1,impostor_pattern1,online_call1 in
      (UsersPatternsFile1[user_pattern1]|||ImpostorsPatternsFile1[impostor_pattern1]
      |||OnLineCallsFile1[online_call1])
      |[user_pattern1,impostor_pattern1,online_call1]|
      ImpostorCorbaAgent1[user_pattern1,online_call1,impostor_pattern1,impostor_notif]
      where
      process UsersPatternsFile1[user_pattern1]:noexit:=
        user_pattern1; UsersPatternsFile1 [base_1]
      endproc
      process ImpostorsPatternsFile1 [user_pattern1]:noexit:=
        user_pattern1; ImpostorsPatternsFile1 {user_pattern1}
      endproc
      process OnLineCallsFile1[online_call1]:noexit:=
        online_call1; OnLineCallsFile1(online_call1)
      endproc
      process ImpostorCorbaAgent1
        (user_pattern1,online_call1,impostor_pattern1,impostor_notif):noexit:=
          online_call1;user_pattern1;impostor_pattern1;
          (i; ImpostorCorbaAgent1(user_pattern1,online_call1,
          impostor_pattern1,impostor_notif)
          [] i;impostor_notif; ImpostorCorbaAgent1(user_pattern1, online_call1,
          impostor_pattern1,impostor_notif))
        endproc
      endproc
      process process ImpostorCorbaAgent2[impostor_notif]:noexit:=
        hide user_pattern2,impostor_pattern2,online_call2 in
        (UsersPatternsFile2[user_pattern2]|||ImpostorsPatternFile2[impostor_pattern2]
        |||OnLineCallsFile2[online_call2])|[user_pattern2,impostor_pattern2,online_call2]|
        ImpostorCorbaAgent2[user_pattern2,impostor_pattern2,online_call2,impostor_notif]
        where
        process UsersPatternsFile2[user_pattern2]:noexit:=
          user_pattern2; UsersPatternsFile2 [base_2]
        endproc
        process ImpostorsPatternsFile2[impostor_pattern2]:noexit:=
          impostor_pattern2; ImpostorsPatternsFile2 {impostor_pattern2}
        endproc
        process OnLineCallsFile2[online_call2]:noexit:=
          online_call2; OnLineCallsFile2(online_call2)
        endproc
        process ImpostorCorbaAgent2[user_pattern2,impostor_pattern2,
          online_call2,impostor_notif]:noexit:=
          online_call2;user_pattern2;impostor_pattern2;
          (i;ImpostorCorbaAgent2(online_call2;user_pattern2;
          impostor_pattern2,impostor_notif)
          []i;impostor_notif;ImpostorCorbaAgent2(online_call2;user_pattern2;
          impostor_pattern2,impostor_notif))
        endproc
      endproc
    endproc
  process ImpostorCorbaManager[impostor_notif,check_owner]noexit:=
    impostor_notif;
    (i;check_owner;ImpostorCorbaManager[impostor_notif,check_owner]
    []i;ImpostorCorbaManager[impostor_notif,check_owner])
  endproc
endproc
endspec

```

FIGURA 2.30 – ESPECIFICAÇÃO LOTOS REFINADA DO SISTEMA SSTCC.

Ao longo desses refinamentos apresentados, o sistema pode ser constantemente validado através do emprego de ferramentas apropriadas (GARAVEL, 1997) – não sendo necessário, portanto, obter a especificação final para dar início aos procedimentos de validação. No item 2.4, a seguir, são apresentados os procedimentos de validação.

## 2.4 Validação Formal de Sistemas

“À partir d’une représentation d’un système sous forme d’automate, il est possible d’observer et de vérifier certaines propriétés de ce système, telles que la présence de situations de blocage” (ARNOLD, 1989). Uma das grandes vantagens do uso de TDFs é a possibilidade de demonstrar a correção de uma especificação. O termo “validação” pode ser usado para descrever as atividades de demonstração de correção, ou apenas aumento de confiabilidade, de uma especificação ou implementação. Algumas das técnicas de validação (teste, simulação e verificação) são descritas abaixo.

### 2.4.1 Teste

Testes podem ser aplicados em especificações realizadas em alto nível de abstração ou até mesmo em versões de implementação. A maior desvantagem dessa técnica de validação é que os testes não são exaustivos, ou seja, são usados para encontrar erros e não para demonstrar correção (não constituindo, desse modo, uma prova formal de correção). Quanto mais tarde for descoberto um erro, maior será o custo de sua correção.

Com a ferramenta *Cæsar.open* (com o componente *exhibitor*), por exemplo, uma especificação pode ser testada através da procura por seqüências de ações visíveis. A ferramenta mostra as seqüências executadas – se essas forem encontradas. Se tais seqüências não existirem, nada é mostrado. Ainda com a ferramenta *Cæsar.open* (com o componente *executor*), pode ser realizado outro tipo de teste, que explora a execução de uma seqüência randômica em um sistema de transições rotuladas correspondente a uma especificação LOTOS. Nessa execução, escolhe-se um número  $n$  de transições a serem executadas, bem como a estratégia para resolver o indeterminismo (se permitido ou não). As  $n$  ações visíveis executadas são mostradas como resultado.

O teste de um protocolo diante do serviço pode ser realizado definindo seqüências de teste derivadas da especificação de serviço e aplicando-as na especificação de protocolo.

### 2.4.2 Simulação

Na simulação, um modelo executável é desenvolvido e observado. Se o modelo não for muito complexo, a simulação descobre a maioria dos erros. Quando, porém, o modelo torna-se complexo – o que ocorre na grande maioria dos casos – geralmente é impossível simular todos os casos importantes.



A ferramenta *Cæsar*, por exemplo, em uma das fases da compilação (*simulation*), realiza a simulação exaustiva da especificação. Essa fase explora exaustivamente todos os comportamentos possíveis definidos pela rede gerada (uma generalização de Redes de Petri) e produz um grafo de estado. O número de estados, bem como o número de transições do grafo, são somente limitados pela quantidade de memória disponível no equipamento (o que, aliás, não é difícil de ocorrer).

Já a ferramenta *Cæsar.open* (com o componente *simulator*) realiza a simulação interativa através de um laço infinito que lê comandos da entrada-padrão, executa esses comandos e os mostra na saída-padrão. A execução parte do estado inicial, oferecendo ao usuário as opções de estados possíveis a cada transição.

### 2.4.3 Verificação

A verificação é uma prova formal de que uma especificação ou implementação satisfaz propriedades desejáveis, usando métodos matemáticos rigorosos. A análise de especificações permite várias verificações. Por exemplo, comparar uma especificação original com outra mais refinada. Dois tipos de verificação podem ser destacados:

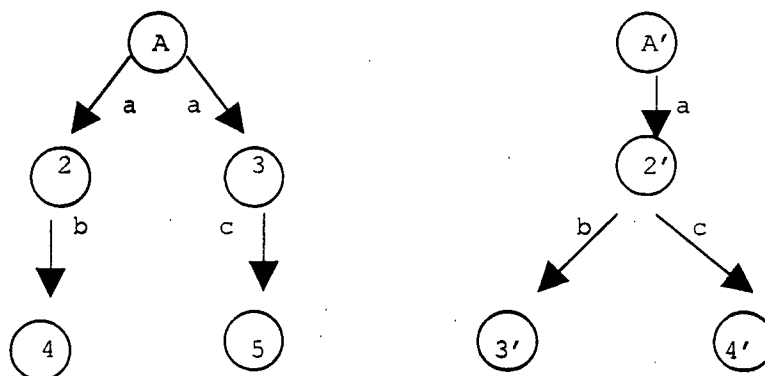
- (1) **Verificação por redução:** os grafos gerados pelos compiladores são, em sua grande maioria, muito complexos para serem verificados. Então, aplicam-se critérios de abstração, reduzindo os grafos de acordo com relações de equivalência, tais como equivalência forte e fraca (veja abaixo nos itens 2.4.3.1 e 2.4.3.2);
- (2) **Verificação por comparação:** uma especificação pode ser verificada através da comparação com outra especificação sob critérios, tais como equivalência forte e fraca. Desse modo, pode-se verificar, por exemplo, se um protocolo provê um dado serviço.

As relações de equivalência forte e fraca (equivalência de observação) são descritas abaixo (ARNOLD, 1989, GARAVEL 1997):

#### 2.4.3.1 Equivalência Forte (*Strong Bisimulation*)

A equivalência forte (*strong bisimulation*) de dois sistemas de transições pode ser provada se for mostrada a existência de uma relação de bissimulação entre os dois conjuntos de estados desses dois sistemas. Tal relação é obtida se observadas todas as transições, incluindo as transições internas com igual importância.

A equivalência forte requer que cada evento em um sistema de transições corresponda a exatamente um evento igual no outro sistema de transições. Um exemplo típico que demonstra que sistemas que podem executar as mesmas seqüências de ações nem sempre são equivalentes é mostrado na Figura 2.31.



Comportamento do sistema A

Comportamento do sistema A'

FIGURA 2.31 – SISTEMAS NÃO EQUIVALENTES QUANTO À EQUIVALÊNCIA FORTE.

A Figura 2.31 apresenta dois sistemas não equivalentes quanto à equivalência forte. Dois estados  $p$  e  $q$  são *bisimilars* se para cada estado  $p'$  alcançável de  $p$  pela execução de uma ação  $a$  existe um estado  $q'$ , alcançável de  $q$  pela execução de uma ação  $a$  tal que  $p'$  e  $q'$  sejam *bisimilars*.

Seja  $G1$  e  $G2$  dois sistemas de transições rotuladas com conjuntos de estados  $S1$  e  $S2$ , respectivamente, e seja  $S = S1 \cup S2$ . Uma relação de equivalência forte sobre  $G1$  e  $G2$  é uma relação  $R \subseteq S \times S$  tal que  $\langle p, q \rangle \in R$  implica

se  $p \xrightarrow{a} p'$  então  $\exists q': q \xrightarrow{a} q'$  e  $\langle p', q' \rangle \in R$ , e

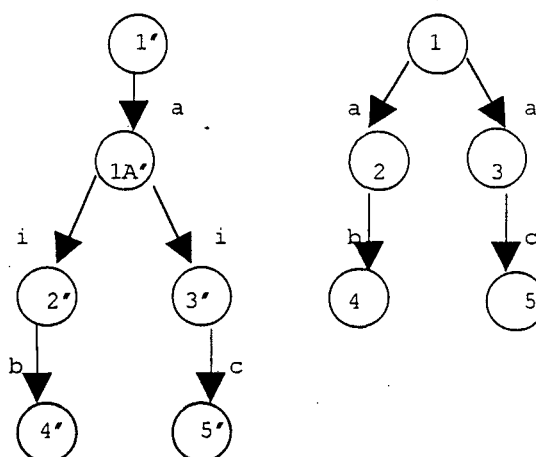
se  $q \xrightarrow{a} q'$  então  $\exists p': p \xrightarrow{a} p'$  e  $\langle p', q' \rangle \in R$ .

Dois sistemas de transições rotuladas  $G1$  e  $G2$  são equivalentes se existe uma relação de equivalência forte relacionando os estados iniciais dos dois sistemas. A equivalência forte é, geralmente, muito forte para os propósitos de verificação por comparação. Abaixo, são descritas duas relações de equivalência usadas na verificação.

### 2.4.3.2 Equivalência Fraca (*Weak Bisimulation*)

Na equivalência fraca (*weak bisimulation*), também chamada de equivalência de observação, um evento em um sistema de transições não necessita estar relacionado a exatamente um evento no outro sistema de transições, pois um evento interno ( $\tau$ -event) de um grafo pode corresponder a zero ou mais eventos internos no outro grafo e vice-versa.

Um sistema pode trocar de estado “espontaneamente” sem qualquer razão aparente ou observável (MILNER, 1980). Na Figura 2.32, mostra-se o comportamento de um sistema  $A''$ , que do estado  $1''$  passa ao estado  $1A''$  após ter sido executada a ação  $a$ . Em seguida, espontaneamente, o sistema passa para o estado  $2''$  ou para o estado  $3''$ . Prosseguindo, o sistema pode executar a ação  $b$  e a ação  $c$ . O comportamento do sistema  $A''$  é equivalente quanto à equivalência de observação ao comportamento do sistema  $A$ .



Comportamento do sistema  $A''$

Comportamento do sistema  $A$

FIGURA 2.32 – SISTEMAS EQUIVALENTES QUANTO À EQUIVALÊNCIA DE OBSERVAÇÃO.

A Figura 2.32 apresenta dois sistemas equivalentes quanto à equivalência de observação. Dados os estados  $p$  e  $p'$ , escreve-se  $p \Rightarrow^a p'$  se  $p$  para executar um evento  $a$  precedido ou sucedido por qualquer número finito (inclusive zero) de eventos internos ( $\tau$ -events) e chegar a um estado  $p'$ . Escreve-se  $p \Rightarrow^\tau p'$  se  $p$  para realizar zero ou mais eventos internos e chegar ao estado  $p'$ .

Seja  $G1$  e  $G2$  dois sistemas de transições rotuladas com conjuntos de estados  $S1$  e  $S2$ , respectivamente, e seja  $S = S1 \cup S2$ . Uma relação de equivalência fraca sobre  $G1$  e  $G2$  é uma relação  $R \subseteq S \times S$  tal que  $\langle p, q \rangle \in R$  implica

se  $p \xrightarrow{a} p'$  então  $\exists q': q \Rightarrow^a q'$  e  $\langle p', q' \rangle \in R$ , e

se  $q \xrightarrow{a} q'$  então  $\exists p': p \Rightarrow^a p'$  e  $\langle p', q' \rangle \in R$ .

Dois grafos  $G1$  e  $G2$  são equivalentes quanto à equivalência de observação se existir uma relação de equivalência fraca relacionando os estados iniciais dos dois grafos.

Se para chegar a um determinado estado em um sistema de transição foi necessária uma ação  $a$ , então, para haver equivalência forte com outro sistema de transição, a mesma ação  $a$  é requerida. Já na equivalência fraca podem ocorrer eventos internos antes e após a ação  $a$ , e os sistemas de transições são considerados equivalentes, pois os eventos internos, nesse caso, são invisíveis.

## 2.5 Experimentos na Validação Formal do Sistema SSTCC

EUCALYPTUS agrega um conjunto de diversas ferramentas LOTOS em uma interface gráfica para ambiente Unix<sup>32</sup>. A principal ferramenta utilizada por este trabalho chama-se CADP (*Caesar/Aldebaran Development Package*), que possui dois principais componentes: (i) CAESAR; e (ii) ALDEBARAN.

(i) **CAESAR**: CAESAR é um compilador que traduz uma especificação LOTOS em um programa C (para ser executado ou simulado) ou em um LTS – Sistema de Transições Rotuladas/*Labelled Transition System* (para ser verificado utilizando ferramentas de bissimulação e/ou de lógica temporal). Por exemplo, é possível comparar o LTS de um protocolo com o LTS do serviço implementado pelo protocolo. Ambos LTSs são gerados utilizando CAESAR e comparados utilizando ALDEBARAN. Também é possível especificar propriedades do protocolo utilizando fórmulas de lógica temporal que podem ser avaliadas no LTS do protocolo. Os algoritmos de tradução CAESAR seguem vários passos. Primeiro, a descrição LOTOS é traduzida em uma álgebra de processos simplificada, chamada SUBLOTOS; em seguida, em um modelo intermediário de Redes de Petri, o qual provê uma representação compacta, estruturada e compreensível pelo usuário de ambos, controle e fluxo de dados. Eventualmente, o LTS é produzido através de análise de alcançabilidade sobre a Rede de Petri. CAESAR aceita LOTOS completo (i.e.,

<sup>32</sup> A partir de 1999, também para ambiente Linux.

ambos, comportamento e dados) com a seguinte restrição em relação à parte de controle: recursão de processo não é permitida à esquerda e à direita de  $[[...]]$ , nem na parte esquerda de  $\gg$  e  $[>$ . Apesar dessas restrições, o subconjunto de LOTOS suportado pelo CAESAR é grande e geralmente suficiente para as necessidades reais. A versão atual do CAESAR permite a geração de grandes LTSs (alguns milhões de estados) em um razoável intervalo de tempo. A versão mais atual do CAESAR oferece a funcionalidade chamada EXEC/CAESAR para a geração de código C. Esse código C pode ser inserido em aplicações, o que permite prototipação rápida diretamente das especificações LOTOS.

(ii) **ALDEBARAN**: ALDEBARAN é uma ferramenta para verificação de sistemas de comunicação, representados por Sistemas de Transições Rotuladas (LTS), i.e., as transições de estados são rotuladas por nomes de ações. Isso permite a redução de LTSs sob várias relações de equivalência (tais como bissimulação forte, equivalência quanto à observação, bissimulação de retardo e equivalência segura). ALDEBARAN provê ao usuário um diagnóstico quando dois LTS são considerados não equivalentes. Os algoritmos de verificação utilizados pelo ALDEBARAN são baseados em estudos, principalmente de Paige-Tarjan e Fernandez-Mounier (GARAVEL, 1997).

Veja, abaixo, os principais passos para a verificação de correção do sistema, i.e., para a obtenção da prova formal de correção do sistema.

### Passo 1 : Geração do LTS

Inicialmente gera-se o LTS (*Labelled Transition System*) correspondente à especificação de protocolo do arquivo `SstccProtocol.lotos`. Para isso, basta escolher a opção “*Generate labelled transition system...*”. Veja a Figura 2.33. Na próxima janela, clica-se em OK. Veja a Figura 2.34.

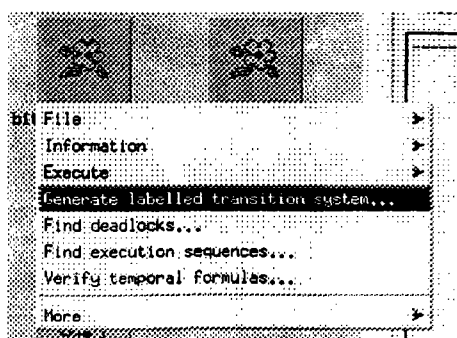


FIGURA 2.33 – GERAÇÃO DO LTS.

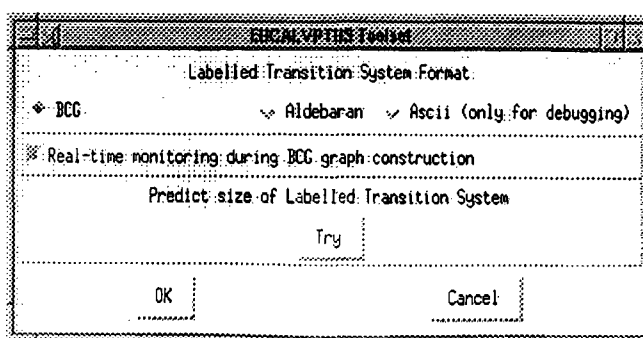


FIGURA 2.34 – ESCOLHA NA GERAÇÃO DO LTS.

Durante a geração, os números de estados e de transições são apresentados. Uma vez que a geração esteja completa, deve-se clicar no botão “done”. Veja a Figura 2.35.

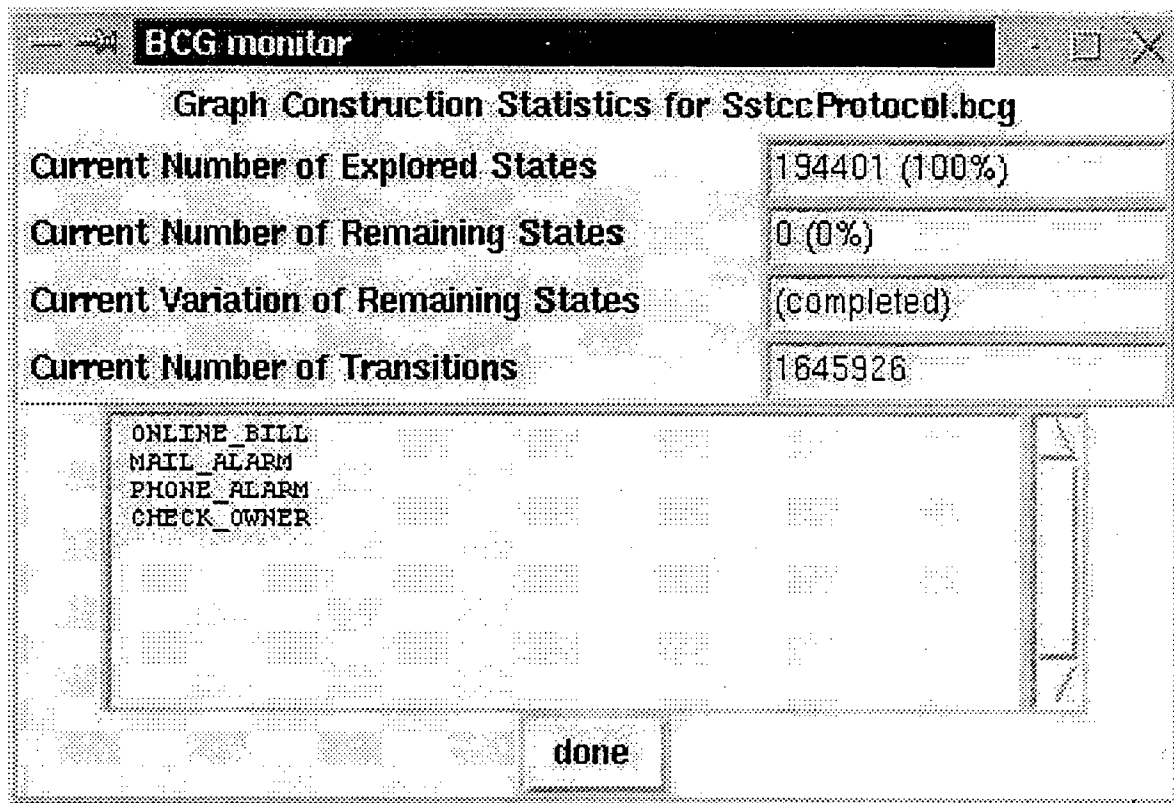


FIGURA 2.35 - LTS CORRESPONDENTE À ESPECIFICAÇÃO LOTOS SstccProtocol.

A Figura 2.35 mostra que o LTS gerado, correspondente à especificação LOTOS SstccProtocol LOTOS, possui 194.401 estados e 1.645.926 transições. O grafo gerado é de formato BCG.

**BCG:** BCG (*Binary-Coded Graphs*) – Gráficos de Código Binário – é um formato de representação de LTSs que inclui uma coleção de bibliotecas e programas associados a esse formato. Comparados aos formatos para LTSs baseados em ASCII (*American National Standard Code for Information Intercange*), o formato BCG utiliza representação binária com técnicas de compressão que resultam em arquivos muito menores (até 20 vezes). BCG é independente de qualquer linguagem, porém mantém os objetos (tipos, funções e variáveis) definidos em programas fontes. As seguintes ferramentas são atualmente disponíveis para esse formato:

- (1) BCG\_IO que realiza conversões entre o formato BCG e uma dúzia de outros formatos;
- (2) BCG\_OPEN que estabelece uma comunicação (*gateway*) entre o formato BCG e o formato do ambiente OPEN/CAESAR;
- (3) BCG\_DRAW que provê uma representação gráfica em duas dimensões dos grafos BCG incluindo automaticamente os estados e transições; e
- (4) BCG\_EDIT que é um editor interativo que permite modificar manualmente a saída (*display*) gerada pelo BCG\_DRAW.

Nesse passo, impasses (*deadlocks*) podem ser detectados (se existirem). Veja a Figura 2.36.

```
In specification SSTCCPROTOCOL [1]
process BILLDB1 [76] is never reached
process BILLDB2 [92] is never reached
a deadlock exists for rendez-vous:
  BILL_DB1 [68]
  synchronized by parallel operator ``|[exit, ONLINE_CALL1, BILL_DB1]|''
  this gate/offer combination is used in the right operand:
    BILLCORBAAGENT1 [exit, ONLINE_CALL1, BILL_DB1, BILL_NOTIF]
  but not in the left operand:
    ONLINECALLFILE1 [exit, ONLINE_CALL1]
a deadlock exists for rendez-vous:
  BILL_DB2 [84]
  synchronized by parallel operator ``|[exit, ONLINE_CALL2, BILL_DB2]|''
  this gate/offer combination is used in the right operand:
    BILLCORBAAGENT2 [exit, ONLINE_CALL2, BILL_DB2, BILL_NOTIF]
  but not in the left operand:
    ONLINECALLFILE2 [exit, ONLINE_CALL2]
```

FIGURA 2.36 – IMPASSE (*DEADLOCK*) DETECTADO DURANTE A GERAÇÃO DO LTS.

Os impasses (*deadlocks*) apresentados na Figura 2.36 foram facilmente resolvidos atualizando-se as linhas 10 e 26 da Figura 2.21, conforme apresentado abaixo:

```
Linha10:   de:   (OnLineCallFile1[online_call1])
           para: (OnLineCallFile1[online_call1]|||BillDb1[bill_db1])
Linha26:   de:   (OnLineCallFile2[online_call2])
           para: (OnLineCallFile2[online_call2]|||BillDb2[bill_db2])
```

Demonstra-se assim, a eficácia da especificação LOTOS na detecção de impasses (*deadlocks*).

Se a opção escolhida na Figura 2.34 for ALDEBARAN (em vez de BCG), então o seguinte resultado é apresentado (veja a Figura 2.37):

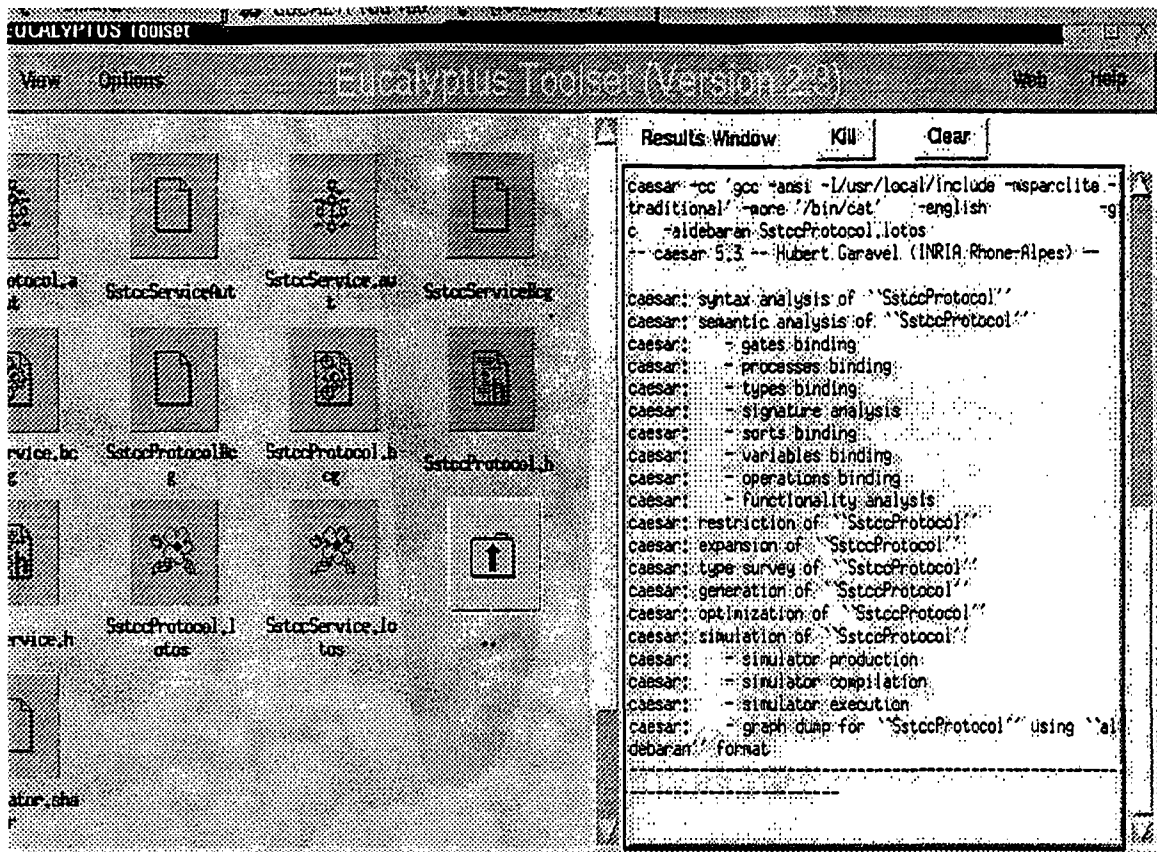


FIGURA 2.37 – GERAÇÃO DO LTS EM FORMATO ALDEBARAN.

Veja parte do grafo gerado no formato ALDEBARAN (formato composto de triplas representando do-estado, evento, para-o-estado) na Figura 2.38.

```

SstccProtocol.aut
des (0, 1645926, 194401)
(0, i, 1)
(0, i, 2)
(0, i, 3)
(0, i, 4)
(0, i, 5)
(0, i, 6)
(1, i, 7)
(1, i, 8)
(1, i, 9)
(1, i, 10)
. . .
(194398, i, 194328)
(194398, i, 194400)
(194398, i, 193367)
(194399, ONLINE_BILL, 194197)
(194399, CHECK_OWNER, 194329)
(194399, MAIL_ALARM, 194400)
(194400, ONLINE_BILL, 194279)
(194400, CHECK_OWNER, 194357)
(194400, PHONE_ALARM, 193968)
(194400, i, 193968)
  
```

FIGURA 2.38 – SSTCCPROTOCOL.AUT – LTS NO FORMATO ALDEBARAN.

A primeira linha do grafo apresentado na Figura 2.38 mostra que o estado inicial é '0', o número de transições é 1.645.926 e o número de estados é 194.401.



## Passo 2 : Visualização do LTS

Um arquivo chamado `sstccProtocol.bcg` é criado. Para visualizá-lo basta clicar em “*Visualize/Draw in 2 Dimensions*”. Veja a Figura 2.39.

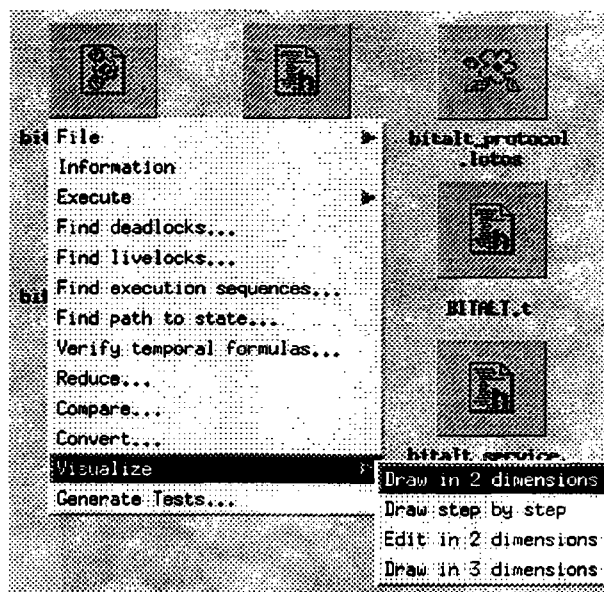


FIGURA 2.39 – LTS EM 2 DIMENSÕES.

O seguinte grafo é apresentado (Figura 2.40):

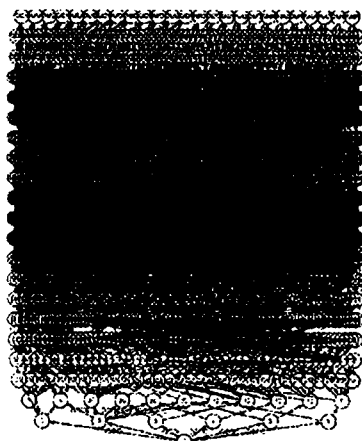


FIGURA 2.40 – LTS COM MUITAS TRANSIÇÕES.

Como pode ser observado, o grafo é muito denso, i.e., o LTS tem muitos estados e transições, tornando impossível uma observação adequada. Uma alternativa é simplificar o grafo, como mostrado no próximo passo. .

### Passo 3 : Redução do LTS

Para reduzir o LTS, basta clicar sobre o arquivo que contém o LTS (`SstccProtocol.bcg`) e selecionar “*Reduce...*”. Veja a Figura 2.41.

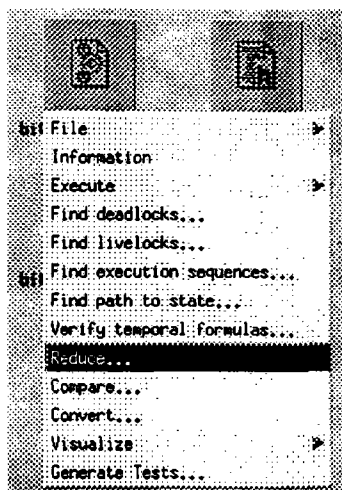


FIGURA 2.41 – REDUÇÃO DO LTS.

Na janela seguinte é possível escolher a ferramenta de redução (*Aldebaran* or *Fc2tools*), o algoritmo e também a equivalência ser utilizada. Inicialmente pode ser selecionado: (1) **Aldebaran** para a ferramenta a ser utilizada; (2) **Strong Equivalence** para a relação de comparação; e (3) **Standard** para o método de decisão. Então, pressiona-se em OK para efetuar a redução. Obtém-se o arquivo `SstccProtocol_bmin.bcg` (veja a Figura 2.42):

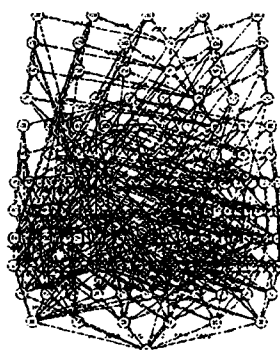


FIGURA 2.42 – LTS REDUZIDO POR EQUIVALÊNCIA FORTE.

Então, foram escolhidas as seguintes opções: (1) **Aldebaran** para a ferramenta a ser utilizada; (2) **Observational Equivalence** para a relação de comparação; e (3) **Standard** para o método de decisão. Então, o arquivo `SstccProtocol_omin.bcg` foi obtido.

## Passo 4 : Geração do LTS

Da mesma maneira como apresentado no Passol, neste passo é gerado o LTS associado à especificação `sstccService.lotos`. Este LTS é pequeno, e por isso não é necessário reduzi-lo. Veja a Figura 2.43.

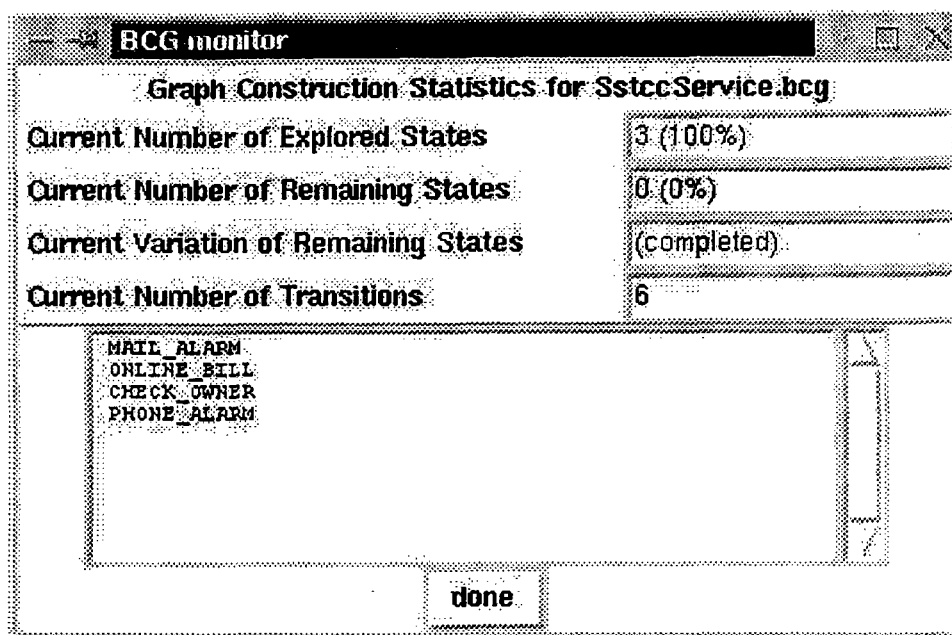


FIGURA 2.43 – GERAÇÃO DO GRAFO SSTCCSERVICE.

A Figura 2.43 mostra que o LTS associado à especificação de serviço contém somente 3 estados e 6 transições.

Observe na Figura 2.44 os tamanhos dos arquivos relacionados às especificações de serviço e protocolo (o grafo associado ao protocolo atinge 90Mb).

```
lrg-gw(mirela)61: ls -l
total 286548
-rw-r--r-- 1 mirela lrg 33866693 Nov 30 18:28 SstccProtocol.aut
-rw-r--r-- 1 mirela lrg 5660892 Nov 30 18:05 SstccProtocol.bcg
-rw-r--r-- 1 mirela lrg 1545 Nov 30 18:03 SstccProtocol.h
-rw-r--r-- 1 mirela lrg 7661 Nov 30 17:58 SstccProtocol.lotos
-rw-r--r-- 1 mirela lrg 90243072 Nov 30 19:36 SstccProtocol.ps
-rw-r--r-- 1 mirela lrg 117464 Nov 30 19:11 SstccProtocolAut
-rw-r--r-- 1 mirela lrg 300731 Dec 1 13:34 SstccProtocol_bmin.bcg
-rw-r--r-- 1 mirela lrg 13216599 Dec 1 14:07 SstccProtocol_bmin.ps
-rw-r--r-- 1 mirela lrg 125 Nov 30 18:24 SstccService.aut
-rw-r--r-- 1 mirela lrg 2600 Nov 30 18:10 SstccService.bcg
-rw-r--r-- 1 mirela lrg 1545 Nov 30 18:01 SstccService.h
-rw-r--r-- 1 mirela lrg 588 Nov 30 17:58 SstccService.lotos
-rw-r--r-- 1 mirela lrg 1602 Dec 1 14:03 aldebaran.seq
-rw-r--r-- 1 mirela lrg 1869051 Oct 25 18:29 installator.shar
lrg-gw(mirela)62:
```

FIGURA 2.44 – RELAÇÃO DOS ARQUIVOS GERADOS CONTENDO OS RESPECTIVOS TAMANHOS.

Se em vez de BCG fosse escolhida a opção ALDEBARAN, iriam ser apresentadas as informações conforme a Figura 2.45.

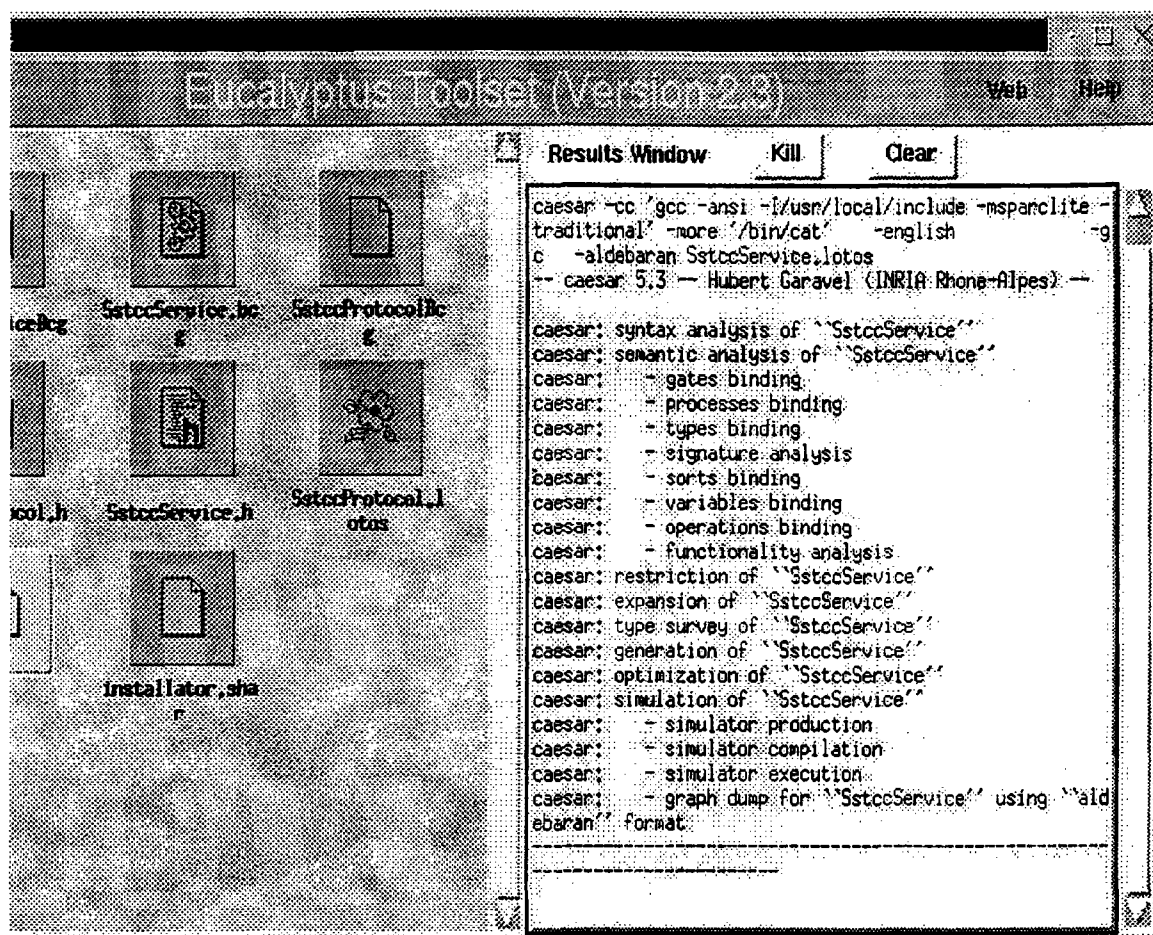


FIGURA 2.45 – GRAFO SSTCCSERVICE NO FORMATO ALDEBARAN.

Como pode ser observado, cada passo (syntax analysis, semantic analysis, etc) foi realizado com sucesso, sem apresentar erros. Veja na Figura 2.46 o LTS no formato Aldebaran (i.e., do-estado, evento, para-estado).

```
SstccService.aut

des (0, 6, 3)
(0, MAIL_ALARM, 1)
(0, ONLINE_BILL, 0)
(0, CHECK_OWNER, 0)
(1, i, 2)
(1, i, 0)
(2, PHONE_ALARM, 0)
```

FIGURA 2.46 – SSTCCSERVICE LTS.

Como pode ser observado na Figura 2.46, o LTS inicia no estado 0 e inclui 3 estados e 6 transições.

## Passo 5 : Comparação de LTS

Agora é possível comparar o LTS que representa o protocolo com o LTS que representa o serviço esperado. Para isso, basta clicar sobre o arquivo `SstccProtocolo_bmin.bcg` e selecionar “*Compare...*”. Veja a Figura 2.47.

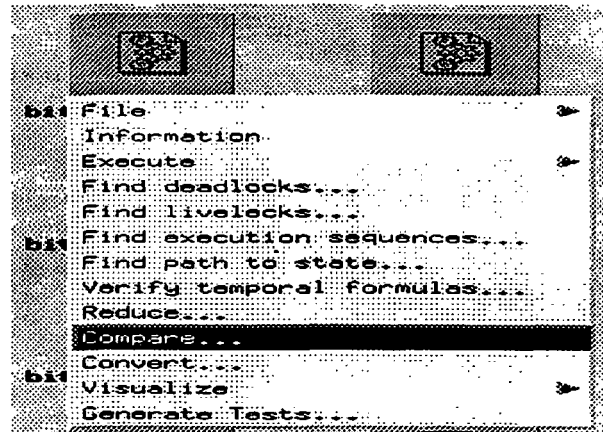


FIGURA 2.47 – OPÇÃO DE COMPARAÇÃO.

Na próxima janela, é possível escolher a ferramenta para comparar os LTSs (Aldebaran ou Fc2tools), bem como a relação de comparação (*Strong Equivalence Bisimulation*, *Observational Equivalence Bisimulation*, ...). Pode-se selecionar: (1) `SstccService.bcg` para o LTS a ser comparado; (2) Aldebaran para a ferramenta a ser utilizada; (3) **Observational Equivalence** para a relação de comparação; e (4) **Standard** para o método de decisão. Após confirmar com OK, o resultado da comparação aparece na parte direita da janela. Veja a Figura 2.48.

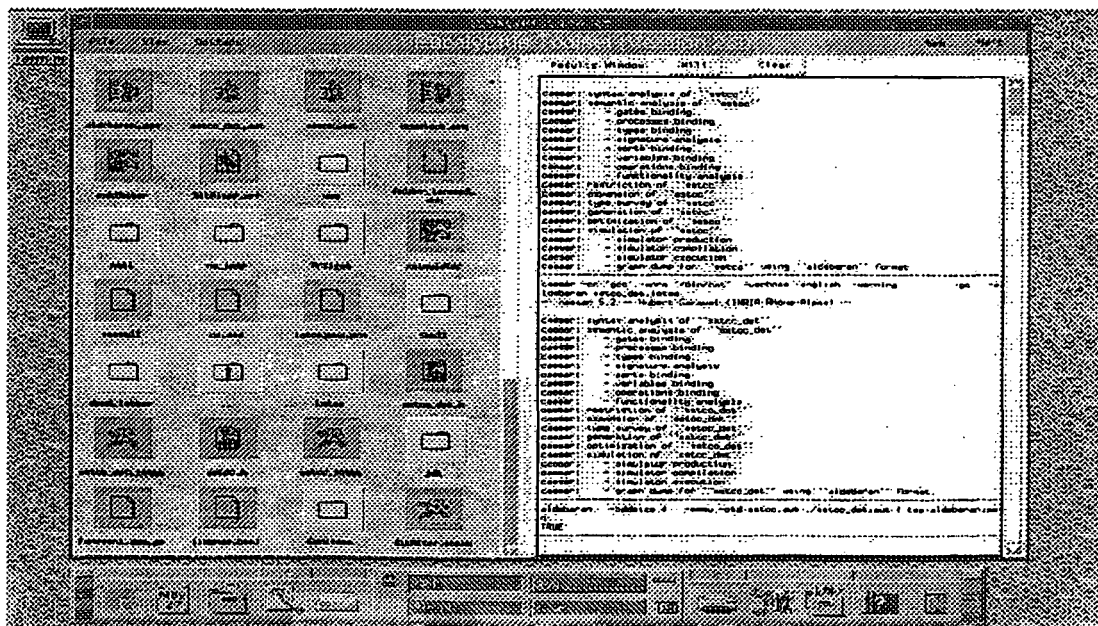


FIGURA 2.48 – RESULTADO DA EQUIVALÊNCIA QUANTO À OBSERVAÇÃO.

O resultado é TRUE, o que significa que o protocolo executa o serviço esperado. Na Figura 2.49, apresenta-se com mais clareza este resultado observado na Figura 2.48, i.e., a prova formal de equivalência quanto à observação entre as especificações SstccService e SstccProtocol. O resultado "TRUE" obtido representa que a especificação inicial dos requisitos do sistema é equivalente à especificação final refinada.

```

caesar: -cc `gcc -ansi -I/usr/local/include -msparclite -traditional' -more `/bin/cat' -
english -9c -aldebaran SstccService.lotos
-- caesar 5.3 - Hubert Garavel (INRIA Rhone-Alpes) --

caesar: syntax analysis of "SstccService"
caesar: semantic analysis of "SstccService"
caesar:   - gates binding
caesar:   - processes binding
caesar:   - types binding
caesar:   - signature analysis
caesar:   - sorts binding
caesar:   - variables binding
caesar:   - operations binding
caesar:   - functionality analysis
caesar: restriction of "SstccService"
caesar: expansion of "SstccService"
caesar: type survey of "SstccService"
caesar: generation of "SstccService"
caesar: optimization of "SstccService"
caesar: simulation of "SstccService"
caesar:   - simulator production
caesar:   - simulator compilation
caesar:   - simulator execution
caesar:   - graph dump for "SstccService" using "aldebaran" format
-----
caesar: -cc `gcc -ansi -I/usr/local/include -msparclite -traditional' -more `/bin/cat' -
english -9c -aldebaran SstccProtocol.lotos
-- caesar 5.3 - Hubert Garavel (INRIA Rhone-Alpes) --

caesar: syntax analysis of "SstccProtocol"
caesar: semantic analysis of "SstccProtocol"
caesar:   - gates binding
caesar:   - processes binding
caesar:   - types binding
caesar:   - signature analysis
caesar:   - sorts binding
caesar:   - variables binding
caesar:   - operations binding
caesar:   - functionality analysis
caesar: restriction of "SstccProtocol"
caesar: expansion of "SstccProtocol"
caesar: type survey of "SstccProtocol"
caesar: generation of "SstccProtocol"
caesar: optimization of "SstccProtocol"
caesar: simulation of "SstccProtocol"
caesar:   - simulator production
caesar:   - simulator compilation
caesar:   - simulator execution
caesar:   - graph dump for "SstccProtocol" using "aldebaran" format
-----
aldebaran -bddsize 4 -oequ -std SstccProtocol_bmin.bcg ./SstccService.bcg | tee
aldebaran.seq
TRUE
-----

```

FIGURA 2.49 - PROVA FORMAL DE CORREÇÃO DO SISTEMA SSTCC.

Note que o processo de validação é considerado composto por simulações, testes e verificações. Enquanto as simulações e os testes são utilizados apenas para encontrar erros, a verificação proporciona a obtenção da prova formal (matemática) de correção do sistema. Essa prova visa satisfazer os requisitos do nível máximo de segurança de acordo com o Departamento de Defesa do EUA. Veja a Tabela 2.1.

TABELA 2.1 – NÍVEIS DE SEGURANÇA – DEPARTAMENTO DE DEFESA DOS EUA.

Nível de segurança	Principal característica introduzida
Division D	<i>Minimal protection</i> – proteção mínima.
Division C	<i>Discretionary protection</i> – proteção discreta.
Class C1	<i>Discretionary security protection</i> – proteção discreta de segurança, i.e. o usuário decide o nível de proteção requerida para cada objeto de propriedade do usuário.
Class C2	<i>Controlled access protection</i> – proteção de acesso controlado; auditoria é requerida para contabilizar as ações dos usuários.
Division B	<i>Mandatory protection</i> – proteção obrigatória.
Class B1	<i>Labelled security protection</i> – proteção de segurança rotulada. Cada usuário tem um nível de segurança (chamada de uma desobstrução/liberação de segurança) e cada objeto tem um nível de sensibilidade. O sistema decide se um usuário pode acessar um objeto comparando os valores de desobstrução/liberação e sensibilidade.
Class B2	<i>Structure security protection</i> – proteção de segurança estruturada. Projeto rigoroso (usando modelos formais de segurança), teste e prova de segurança.
Class B3	<i>Security domain</i> – domínio de segurança. Requerimentos suplementares de projeto rigoroso e garantia de segurança.
Division A	<i>Verified protection</i> – proteção verificada.
Class A1	<i>Verified design</i> – projeto verificado. Análise formal e prova matemática de que o sistema de computação casa com a política de segurança do sistema e suas especificações de projeto. Distribuição confiável que garante a segurança do sistema.

Dessa maneira, o procedimento de análise formal e de obtenção da prova matemática de correção apresentada neste Capítulo 2 visam garantir o projeto verificado, em conformidade com o mais alto nível de segurança (*Class A1*) apresentado na Tabela 2.1.

## 2.6 Resultados

O uso da Técnica de Descrição Formal LOTOS (ISO 8807) mostrou-se de grande eficiência e eficácia no desenvolvimento do sistema, garantindo a segurança de um padrão que possui rigor matemático para a especificação e ainda grande poder de análise e projeto de sistemas distribuídos complexos.

Através de ferramentas como a *Eucalyptus*<sup>33</sup>, as especificações puderam ser validadas ao longo dos seus refinamentos, i.e., desde a especificação mais abstrata até a mais refinada do sistema. Das formas de validação, i.e., simulações, testes e verificações, atenção especial foi dada às verificações, pois, enquanto simulações e testes têm (apenas) a finalidade de encontrar erros, as verificações vão além, provendo prova formal de correção do sistema.

O procedimento utilizado para obter a prova de correção entre os refinamentos da especificação foi a geração de sistemas de transições rotuladas (LTS), tanto da (1) especificação mais abstrata *SstccService*, associada ao serviço requerido, quanto da (2) especificação mais refinada *SstccProtocol*, associada ao protocolo obtido. Através do formalismo matemático da técnica de descrição LOTOS, verificou-se formalmente que a especificação de protocolo implementa corretamente o serviço esperado. A prova formal de correção obtida, de acordo com o padrão ISO 8807, visa garantir o nível máximo (*ClassA1*) da Tabela de Segurança do Departamento de Defesa dos EUA descrita no *Orange Book* (SIMON, 1996).

Como trabalhos futuros, pretende-se continuar os refinamentos da especificação de protocolo do sistema de modo a agregar os tipos abstratos de dados. Dessa maneira, o código C gerado automaticamente a partir das especificações LOTOS poderá contribuir ainda mais substancialmente em rapidez e segurança nas atividades de implementação.

---

<sup>33</sup> A equipe VASY (Validação de Sistemas) do INRIA (Grenoble, França), que desenvolveu a ferramenta CADP, participa ativamente na atualização do padrão ISO 8807. Inclusive, o editor deste padrão, Ed Brinksma, foi membro da banca de defesa de PhD de uma pesquisadora da equipe.



*“The classifier efficiency is directly proportionally to the object characteristics choosed. A good classification performance requires the selection of effective characteristics and also a classifier that make a well use of these characteristics – considering limited trainee data and computational resources. Training the network successfully involves many choices and training experiments. From these experiments, the developer learns which configurations train the network most successfully for the application in hand. The developer is thus an architect for a neural network.” (DAYHOFF, 1990)*

### **3 Detecção de Intrusão com Redes Neurais Artificiais**

Este Capítulo apresenta a gerência de segurança para detecção de intrusão (BONIFÁCIO et al, 1998, DENNING, 1987, LUNT, 1988, LUNT et al, 1989) e fraudes (STEWART, 1999) em redes de telecomunicações móveis (GIBSON, 1996) com a utilização de redes neurais artificiais (BARRETO, 1997) para o reconhecimento de padrões (DUDA & HART, 1973, HUSH & HORNE, 1993, LIPPMANN, 1989) dos usuários. Dois classificadores de padrões (algoritmos) são investigados: (1) Kohonen, não-supervisionado (veja Anexo 2.1) (KOHONEN, 1982); e (2) Funções de Base Radial (*Radial Basis Function*), supervisionado (veja Anexo 2.2) (TODESCO, 1995).

O **principal objetivo** do emprego de redes neurais para o reconhecimento de padrões neste trabalho é detectar fraudes em telecomunicações móveis (clonagem e inadimplência) já no seu início (i.e., nas primeiras chamadas ilegais e não apenas quando a fatura mensal for emitida), e dessa forma minimizar os prejuízos. Este Capítulo 3 apresenta como o Sistema de Segurança para Telecomunicações Contra Clonagem e Inadimplência (SSTCC) reconhece os padrões de cada usuário de telecomunicações e, também, como detecta rapidamente as chamadas fora dos padrões de cada usuário. A implementação dos algoritmos de redes neurais artificiais (Kohonen e RBF) é realizada através do emprego das ferramentas *MatLab 5.2.1* (MATLAB, 1992) e *ToolBox* (DEMUTH & BEALE, 1998) em ambiente *Windows*, com equipamento *Pentium II*.

Este Capítulo 3 está organizado como segue. Inicialmente é apresentada uma breve introdução sobre o Modelo Neural (item 3.1). A seguir, são analisados dois algoritmos de classificação – Kohonen (item 3.2) e RBF (item 3.3). E finalmente, no item 3.4, são discutidos os resultados visando a redução dos prejuízos associados à intrusão em redes móveis de telecomunicações.

### 3.1 Modelo Neural

Devido a sua aplicabilidade, tem havido grande interesse dos pesquisadores em relação às Redes Neurais Artificiais (*Artificial Neural Networks – ANN*). Este campo tem tido grande crescimento nas indústrias de computação e sistemas de telecomunicações de grande porte. Redes neurais são redes essencialmente interconectadas paralelamente e uma de suas mais relevantes propriedades é a possibilidade de aprendizagem. Redes Neurais estimam uma função sem requerer uma descrição matemática da funcionalidade da saída da rede em relação à entrada – elas aprendem por exemplos. Pelo aprendizado, uma rede neural pode descobrir padrões e a relação entre eles, além de organizá-los para realizar associações. Como consequência, elas são amplamente utilizadas para resolver problemas de classificação.

Redes neurais (DAYHOFF, 1990), quando vistas como uma rede adaptativa, podem ser analisadas como uma máquina de memória distribuída que é naturalmente capaz de armazenar conhecimento experimental e colocá-lo em disponibilidade para uma utilização qualquer. Redes neurais são similares a mente em dois aspectos: (i) o conhecimento é adquirido pela rede através de processos de aprendizagem; e (ii) os pesos das conexões entre os neurônios, conhecidos como sinapses, são armazenados como conhecimento. O procedimento utilizado para representar o processo de aprendizagem, comumente chamado de algoritmo de aprendizagem, tem a função de modificar os pesos das conexões das redes visando alcançar um objetivo anteriormente planejado.

Nos itens 3.2 e 3.3 seguintes, descreve-se como as redes neurais podem ser utilizadas na indústria de telecomunicações (o item 3.2 apresenta a utilização do algoritmo Kohonen e o item 3.3 apresenta a utilização do algoritmo RBF). Especificamente, é apresentado como a tecnologia de redes neurais pode identificar fraudes, impostores e usuários não confiáveis em operações de telefonia móvel – que é uma preocupação crescente hoje em dia, na maioria das empresas de telecomunicações.

## 3.2 Uso de Classificador Não-Supervisionado – Kohonen

Diversos modelos de rede neural têm sido propostos com a finalidade de classificar dados de acordo com alguma relação de similaridade. Neste item 3.2 emprega-se o modelo Kohonen (INGBER, 1993, KOHONEN, 1982, MEYER, 1994, SCHNEIDER, 1999) para a detecção de fraudes em telefonia móvel a partir da classificação dos usuários em grupos, de acordo com a similaridade de uso do aparelho.

O modelo Kohonen provê um modelo de rede neural de organização adaptativa de mapas topológicos com características que provêm componentes importantes para sistemas de reconhecimento de padrões complexos, conforme descrito no próximo item 3.2.1.

### 3.2.1 Algoritmo Kohonen

No modelo Kohonen, os pesos sinápticos iniciam-se no estado desligado (*off*), i.e., os valores são baixos, e um valor de entrada é provido para a rede sem a especificação da saída desejada. O fato de não especificar a saída caracteriza um algoritmo não-supervisionado. De acordo com o valor de entrada, um neurônio de saída deve prover a melhor resposta, sendo então o vencedor (i.e., o neurônio que gerar a menor distância euclidiana). Tanto o neurônio vencedor quanto seus vizinhos têm seus pesos sinápticos ajustados, e a partir desse momento eles irão responder melhor a este valor de entrada.

Uma vez que todo o conjunto de treinamento tenha sido oferecido à rede, ela é considerada treinada. A próxima fase, i.e., fase de teste, é similar à fase de treinamento, porém nesse caso os pesos não mais são alterados. Se a rede reconhece as entradas de maneira eficiente, então a rede é considerada treinada com sucesso.

Independentemente das entradas, se uma rede possui  $x$  neurônios na camada de saída, então haverá  $x$  saídas possíveis. Em acréscimo, se existem  $x$  entradas, então existirão  $x$  conexões sinápticas entre cada neurônio de saída e os  $x$  neurônios de entrada. A função recebe um vetor de entrada para cada camada e retorna uma saída de zeros para todos os neurônios, exceto para o vencedor. A saída dos vencedores é igual a 1.

O algoritmo Kohonen descreve um mapa  $\phi$  de um espaço de entrada  $V$  em um espaço de saída  $A$ . O espaço de saída consiste em nós  $n_j$ , os quais são organizados em uma ordem topológica. Para cada nó  $n_j$  em  $A$  existe um ponteiro  $w_j$  em  $V$ . Cada ponto  $v$  de  $V$  é mapeado para o nó  $n_i$ , cujo ponteiro  $w_i$  está mais próximo de  $v$ , i.e.,

$$n_i : d^V(w_i, v) = \underset{n_j \in A}{\text{mim}} d^V(w_j, v), \quad (1)$$

onde  $d^V(w_j, v)$  é a distância entre  $w_j$  e  $v$  no espaço  $V$ . Dessa forma, o mapa  $\phi$  é dado pelos ponteiros  $w_j$ . Durante cada passo de iteração, esses ponteiros são ajustados. Uma iteração começa com a geração de um estímulo  $v \in V$ , de acordo com uma função de densidade de probabilidade  $p(v)$ . Então os ponteiros  $w_i$  e ponteiros  $w_j$  dos nós da vizinhança de  $n_i$  são alterados em um pequeno passo em direção a  $v$ :

$$\delta w_j = \epsilon h_{j,i}^0(d^A(n_j, n_i))(v - w_j) \quad \forall n_j \in A. \quad (2)$$

A extensão da vizinhança de  $n_i$  é dada pela função  $h_{j,i}^0$ , da qual depende da distância  $d^A(n_j, n_i)$  entre  $n_j$  e  $n_i$  no espaço de saída. Uma escolha típica é

$$h_{j,i}^0(d) = e^{-d^2/2\sigma^2} \quad (3)$$

A etapa  $\epsilon$  e o raio da vizinhança  $\sigma$  são parâmetros influenciando a convergência do algoritmo. Eles devem decrescer enquanto o número dos passos de aprendizagem aumenta. Geralmente decrescem exponencialmente. A fase de aprendizado começa com a inicialização de todos  $w_j$ , a qual pode ser desenvolvida randomicamente ou baseada em uma informação *a priori* sobre o mapa. O número de passos de aprendizado pode ser dado explicitamente ou então, por um critério de terminação (p.ex., um raio de vizinhança menor que  $\sigma_{min}$ ).

$v_1$ ,  $v_2$  e  $v_3$  são três estímulos no espaço de entrada e  $n_{i1}$ ,  $n_{i2}$ , e  $n_{i3}$  são os respectivos nós do espaço de saída, no qual o estímulo é mapeado. Dessa forma, o mapa  $\phi$  preserva a vizinhança completamente se para cada tripla  $v_1$ ,  $v_2$  e  $v_3$  ocorrer o seguinte:

$$\begin{aligned} d^V(v_2, v_1) \geq d^V(v_3, v_1) &\Leftrightarrow d^A(n_{i2}, n_{i1}) \geq d^A(n_{i3}, n_{i1}) \\ \text{and} & \\ d^V(v_2, v_1) \leq d^V(v_3, v_1) &\Leftrightarrow d^A(n_{i2}, n_{i1}) \leq d^A(n_{i3}, n_{i1}) \end{aligned} \quad (4)$$

Por essas condições serem simétricas em relação a  $V$  e  $A$ , o mapa inverso  $\phi^{-1}$  também preserva a vizinhança completamente. Essa condição (4) é suficiente, mas não necessária para a preservação completa da vizinhança.

Uma vez que a etapa de aprendizado (*learning*) tenha sido finalizada, obtém-se conhecimento suficiente para iniciar sua utilização (*utilization*). No caso deste trabalho, a intenção é utilizar esta rede treinada para a detecção de fraudes em operações de telefonia móvel. Mais especificamente, para diferenciar ligações telefônicas realizadas por usuários legais das realizadas por fraudadores, através do conhecimento dos padrões de cada usuário.

### 3.2.2 Experimentos

O objetivo destes experimentos é mostrar de que forma as redes neurais podem ser aplicadas como ferramenta de identificação de fraudadores que utilizam telefones celulares de maneira ilegal. E além disso, investigar particularmente o impacto do modelo Kohonen na performance do sistema neural de detecção de fraudes em operações de telefonia móvel.

Classificar os usuários de telecomunicações em grupos tem por objetivo possibilitar que o sistema SSTCC identifique ligações telefônicas que não correspondam aos padrões de utilização de um usuário que pertença a determinado grupo. Existem três tipos principais de fraudadores que o sistema SSTCC pode identificar: (i) aqueles que alteram o padrão das ligações telefônicas; (ii) aqueles que compram um telefone móvel somente por um mês (i.e., com a prévia intenção de não pagar pelo serviço); e (iii) aqueles que compram telefones móveis utilizando nome de terceiros, também com a intenção de não pagar. Assim, quando uma chamada telefônica (de um celular legal ou clonado) é finalizada, o sistema verifica se as características desta chamada está de acordo com as características (i.e., padrão) do usuário em questão, armazenadas previamente em um arquivo de padrões. Então, um aviso pode ser enviado ao cliente caso uma possível fraude seja detectada. Esta imediata notificação, ao contrário da espera até o fim do ciclo mensal quando a conta é emitida, colabora para a redução dos prejuízos para ambos, companhias e usuários honestos.

Estes experimentos visam a concepção e o desenvolvimento do modelo de rede neural requerido, a fim de prover a melhor performance (eficiência) para aplicações específicas – neste caso, fraudes em telefonia móvel – e sugerir alguma recomendação (principalmente no escopo da gerência de redes) para a futura geração de sistemas sem fio (*wireless*). Dentre as questões investigadas com esses experimentos (demandando muito tempo) estão:

- (1) determinar as características que devem ser consideradas em cada ligação telefônica, bem como a quantidade de redes neurais a serem utilizadas. Nos experimentos foram analisadas as seguintes possibilidades:
  - (a) 1 rede neural e 4 características (`called_number`, `duration`, `time`, `day`);
  - (b) 2 redes neurais (`weekdays`, `weekend`) e 3 características (`called_number`, `duration`, `time`); e
  - (c) 10 redes neurais (`weekdays00-08h`, `weekdays08-12h`, `weekdays12-14h`, `weekdays14-18h`, `weekdays18-24h`, `weekend00-08Sat`, `weekend08-12Sat`, `weekend12-24Sat`, `weekend00-12Sun`, `weekend12-24Sun`) e 2 características (`called_number`, `duration`).
  
- (2) determinar a quantidade de dados (i.e., ligações telefônicas) necessárias para treinar o sistema, de forma que este possa identificar, eficientemente, os fraudadores. Foram analisadas as seguintes possibilidades:
  - (a) um dia (um arquivo com todas as 24 horas de ligações ou 5 arquivos associados a 5 diferentes horários/tarifas do dia);
  - (b) dois dias (uma segunda-feira e um domingo);
  - (c) três dias (uma segunda-feira, um domingo e um feriado);
  - (d) uma semana (5 dias de uma semana e 2 dias de um fim de semana); e
  - (e) um mês (muitos usuários fazem mais ligações na primeira semana do mês, por exemplo).
  
- (3) determinar o melhor tamanho de saída e o número máximo de agrupamentos (i.e., quantidade de pontos no vetor de saída). Foram analisadas as seguintes possibilidades:
  - (a)  $2^4 = 16$  – suficiente apenas para a combinação `called_number` (`local`, `interurban`, `international`, `special`) e `duration` (`short`, `median`, `long`, `extra long`);
  - (b)  $2^5 = 32$  – suficiente para prover a combinação `called_number`, `duration` e `date` (`weekdays`, `weekend`);
  - (c)  $2^9 = 512$  – suficiente para prover a combinação `called_number`, `duration`, `date` e `time` (`weekdays00-08h`, `weekdays08-12h`, `weekdays12-14h`, `weekdays14-18h`, `weekdays18-24h`, `weekend00-08Sat`, `weekend08-12Sat`, `weekend12-24Sat`, `weekend00-12Sun`, `weekend12-24Sun`); e
  - (d)  $2^{10} = 1024$  – suficiente para prover agrupamentos ainda mais refinados (i.e., prover maior sensibilidade).

A investigação de todas essas questões possibilitou que a rede neural fosse construída e treinada de forma eficiente para a aplicação específica das fraudes nas telecomunicações móveis. A rede resultante foi então avaliada pela sua performance. Por não oferecer níveis de performance provados teoricamente, a performance da rede neural deve ser realizada via testes (DAYHOFF, 1990).

### 3.2.2.1 Definição das Características

Uma boa performance na classificação requer uma boa seleção de características efetivas e um classificador que faça uso dessas características de modo eficiente, considerando os dados de treinamento limitados e os recursos computacionais disponíveis.

Nos experimentos, utilizou-se um equipamento Pentium II com 64Mb de RAM e 4Gb de espaço em disco. Os arquivos de dados (i.e., ligações dos usuários) foram coletados de uma companhia telefônica brasileira<sup>34</sup>. Considerando que a maioria dos usuários possui padrões diferentes nos dias de semana e fins de semana, bem como nos diferentes horários/níveis de descontos, foi utilizado nos experimentos um conjunto de dados representativo que pode ser suportado pelos recursos computacionais disponíveis (limitações de hardware e software). Devido aos recursos computacionais limitados, já que dificilmente uma aplicação real de grande porte poderá utilizar a totalidade dos dados disponíveis, mas sim uma amostra representativa, foi utilizado o seguinte: (1) o uso de duas redes, uma com dias de semana e outra com fins de semana e feriados – e neste caso sem considerar a característica `date`; e (2) o uso de dez redes, ou seja, além de considerar as redes de acordo com os tipos de dia, também foram consideradas distintas redes para cada horário de tarifa durante o dia – e neste caso a característica `time` também não foi considerada. Esta última opção permitiu o uso de um maior número de dados de entrada para treinar a rede, o que melhorou a performance. Foram selecionadas as seguintes características:

- (1) `caller_number` (somente para teste, não para treinamento);
- (2) `called_number` (para identificar o tipo da chamada, i.e., local, internacional, etc);
- (3) `time` (para classificar de acordo com as diferentes tarifas/horários durante o dia);
- (4) `duration` (para identificar longas chamadas fora do padrão do usuário); e
- (5) `date` (para identificar os diferentes padrões durante os dias de semana e finais de semana).

O primeiro procedimento é transformar os dados selecionados em valores dentro do intervalo  $[0, 1]$ . Na Figura 3.1, são apresentadas as primeiras linhas de um arquivo de treinamento que contém 8.830 vetores de entrada (i.e., 8.830 ligações de um domingo, das 12h às 14h).

called_number	duration	local	interurban	international	special	duration
9021516511570	42	1.0000	0	0	0	3.2571
9021516511570	52	1.0000	0	0	0	0.3524
9021516511570	116	1.0000	0	0	0	0.9619
9021512449650	29	1.0000	0	0	0	0.1333
5320489634750	228	1.0000	0	0	0	1.0000
905399796373	1	0	0	1.0000	0	1.0000
905399794723	0	0	0	1.0000	0	0
905399791301	1	0	0	1.0000	0	1.0000
905399756275	122	0	0	1.0000	0	1.0000
905199121379	55	0	0	1.0000	0	0.3810
...						

(a) – Vetores de entrada originais.

(b) – Vetores de entrada transformados.

FIGURA 3.1 – AMOSTRA DOS PRIMEIROS 10 VETORES DE 8.830 CONSIDERADOS.

A Figura 3.1(a) contém dois tipos de dados, `called_number` e `duration`. A Figura 3.1(b) contém cinco tipos de dados: os primeiros quatro tipos de dados são relacionados ao `called_number` e o último é relacionado à `duration`. No caso da `duration`, esta representa uma transformação linear (já que os dados são contínuos). Então, foi atribuído 0 minutos para o valor mínimo (=0), e 30 minutos para o valor máximo (= 1). No caso do `called_number`, foi usada uma transformação binária, onde uma coluna foi transformada em quatro colunas, cada qual correspondendo a um tipo específico de ligação, i.e., `local`, `interurban`, `international`, e `special` (800, 900, 102, 135, etc.). Veja a Tabela 3.1.

TABELA 3.1 – LIGAÇÕES TELEFÔNICAS CLASSIFICADAS POR HORÁRIOS/DIAS.

	qtd ligacoes	menor (min)	media (min)	maior (min)
Sun00-08	6.766	1	29	5056
Sun12-14	6.520	1	48	3711
Sun14-18	14.779	1	40	3455
Mon00-08	19.540	1	44	47239
Mon12-14	13.192	1	46	46997
Mon14-18	10.876	1	44	44774

Usando o conjunto de dados de entrada transformados, a rede pode então ser treinada e depois disso testada para que seja avaliada sua performance. Observa-se que treinar a rede eficientemente envolve muitas escolhas e experimentos, o que demanda muito tempo<sup>35</sup>.

<sup>34</sup> Os arquivos utilizados incluem uma média de 2 milhões de ligações por dia.

<sup>35</sup> Conforme as variáveis escolhidas e a quantidade de dados utilizados, nem todos testes puderam ser realizados devido ao tempo consumido (com os recursos disponíveis). Vários testes necessitaram ser interrompidos após pares de horas em execução sem que os mesmos fossem finalizados. Por exemplo, para considerar a sazonalidade nos padrões das ligações, uma maior quantidade de dados necessita ser considerada.



### 3.2.2.2 Treinamento da Rede

Para treinar o classificador, foram utilizadas de duas até as quatro características consideradas, e assim variando a quantidade de redes de uma a dez. Quanto mais redes, menos características necessitam ser consideradas, e então maior quantidade de dados pode ser utilizada. Por exemplo, se forem utilizadas duas redes, uma para dias de semana e outra para fins de semana, a característica *date* não necessita ser considerada, o que possibilita utilizar mais registros (chamadas) em um arquivo de igual tamanho. E, quanto maior o número de chamadas consideradas, mais exata será a classificação.

Utilizando o conjunto de dados (vetores de entrada), mostrado previamente nas Figuras 3.1(a) e 3.1(b), que contém duas características (*called\_number* e *duration*), a rede foi treinada com quatro pontos no vetor de saída (o que provê um máximo de 16 agrupamentos). Nesse caso, as características *day* e *time* não foram consideradas devido ao fato de que todos os vetores pertencerem ao mesmo *day* (Sunday, August 1<sup>st</sup>, 1999) e ao mesmo *time/desconto* (das 12h às 14h). Na Figura 3.2 é mostrada parte da saída do treinamento, i.e., as iterações e as atualizações de pesos.

```

Ans =
Columns 1 through 12
 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 0 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 13 through 24
...
Columns 61 through 72
 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 0 1 0 0 0 1
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 73 through 84
 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1
 0 1 1 1 0 0 0 0 1 0 0 1
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 85 through 93
 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1
 1 1 1 1 0 0 0 0 0
 1 1 1 1 0 0 0 0 0

```

FIGURA 3.2 – GERAÇÃO DE AGRUPAMENTOS PELO TREINAMENTO DA REDE (ITERAÇÕES).

Como ilustrado na Figura 3.2, é possível observar que quatro diferentes agrupamentos (*clusters*) foram gerados – 1111, 1011, 1101 e 1100 – cada um agrupando um conjunto de usuários destas duas horas (12h às 14h) e deste dia (fim de semana) considerado. Em outras palavras, as ligações telefônicas foram classificadas em agrupamentos (padrões) de acordo com uma relação de similaridade. Veja a Figura 3.3.



(a) – Ligações antes do agrupamento. (b) – Ligações após o agrupamento.

FIGURA 3.3 – LIGAÇÕES TELEFÔNICAS AGRUPADAS.

A Figura 3.3 mostra o agrupamento das ligações telefônicas, i.e., a classificação das ligações em grupos de acordo com suas similaridades entre as quatro características consideradas. Para cada número de características e redes utilizadas, novo treinamento deve ser realizado.

### 3.2.2.3 Teste da Rede

Para testar a rede treinada, submete-se o conjunto de dados (após a transformação no intervalo  $[0, 1]$ ) que representam as ligações de um determinado usuário. Então, a rede gera o vetor de saída contendo o padrão deste usuário. Esta saída ( $ans$ ) representa o padrão deste usuário e será utilizado pelo sistema para detectar ligações suspeitas (Veja a Figura 3.4). Este procedimento é realizado para cada usuário. O número do telefone e seu respectivo padrão são armazenados no arquivo de padrões chamado *Baseline*.

caller_nbr	called_nbr	duration	ans =
5599978334	679874243	4	1
5599978334	679874243	4	0
5599978334	679874243	12	1
5599978334	679874243	12	0
5599978334	679874243	12	
5599978334	679874243	12	
5599978334	679874243	17	
5599978334	679874243	17	
5599978334	679874243	22	
5599978334	679874243	22	

(a) Dados de entrada de um usuário.

(b) Padrão de saída do usuário.

FIGURA 3.4 – TESTE DO CLASSIFICADOR.

Note que quanto mais dados (ligações de um usuário) forem submetidos, mais vetores (de 0s e 1s) são gerados (veja a Figura 3.4). Então, o classificador escolhe o vetor que ocorre mais vezes como o padrão mais provável deste usuário. Por exemplo, se fossem submetidos dez vetores de entrada, sendo seis deles distintos e quatro deles iguais, então os

quatro iguais seriam escolhidos como padrão deste usuário. Pequenas quantidades de dados de entrada para testes podem não serem eficientes para o sistema de segurança. Os experimentos realizados indicam que significantes resultados foram obtidos quando testadas dez ligações de cada usuário de cada vez. Diversos testes (execuções) foram realizados com o intuito de observar a performance de classificação da rede e dos dados utilizados. Os resultados foram muito encorajadores. Os agrupamentos gerados foram eficazes no reconhecimento dos padrões dos usuários e, por conseguinte, na detecção de alterações desses padrões que possam constituir fraudes.

O passo seguinte é determinar qual deve ser a sensibilidade da rede, ou seja, quão grande deve ser a alteração de padrão para caracterizar uma fraude ou para caracterizar uma margem de alterações normais no padrão dos usuários. Dessa forma, a seguir é mostrado o estudo do padrão dos usuários durante os fins de semana, durante os dias de semana, e também durante os diferentes períodos de desconto (tarifas) do dia.

Visando investigar as diferenças de padrão entre dias de semana e fins de semana foram utilizados os seguintes conjuntos de dados:

- (1) ligações de domingo, 1º de agosto, das 12h às 14h – totalizando 8.830 ligações; e
- (2) ligações de Segunda-feira, 2 de agosto, das 12h às 14h – totalizando 13.456 ligações.

Além disso, para investigar as alterações de padrões dos usuários durante os fim de semana e dias de semana, ambas as redes (i.e., weekends e weekdays) necessitam usar os mesmos pesos. Dessa maneira, os pesos gerados (automaticamente) e ajustados na primeira rede (veja a Figura 3.5) são salvos e então utilizados na segunda rede.

Local	interurban	international	special	duration
0.4951	0.4304	0.3535	0.4640	0.4790
-0.4393	0.4282	-0.0903	0.3892	0.6812
0.2861	-0.1166	0.3091	-0.8668	-0.2403
-0.0202	-0.6942	0.4209	-0.1359	0.5675

FIGURA 3.5 – PESOS UTILIZADOS (5 CARACTERÍSTICAS X 4 PONTOS LINEARES DE SAÍDA).

Como esperado, as observações indicaram que o vetor de saída gerado para diversos usuários usando as ligações de domingo foi diferente do vetor gerado usando as ligações de segunda-feira do mesmo usuário. Como resultado, verificou-se uma diferença significativa nos padrões dos usuários durante o fim de semana (Sunday) e durante os dias de semana (Monday).

Similarmente, foram examinados os padrões dos usuários durante os diferentes períodos de desconto do dia. Foram utilizados os seguintes conjuntos de dados:

- (1) ligações de domingo, 1º de agosto, das 12h às 14h – totalizando 8.830 ligações; e
- (2) ligações de domingo, 1º de agosto, da 0h às 8h – totalizando 7.030 ligações.

Aqui, novamente, os padrões obtidos foram diferentes, apesar de neste caso não serem tão significativamente diferentes como no caso de dias de semana (Mondays) e fins de semana (Sundays).

Visando analisar a sensibilidade da rede treinada, na Figura 3.6 são sumarizados os resultados obtidos quando usados várias quantidades de redes e diferentes números de características<sup>36</sup>, considerando conjuntos de dez ligações com dez usuários cada. Cada um destes dez usuários testados foram classificados nos agrupamentos correspondentes – onde quatro agrupamentos tiveram a melhor sensibilidade. Como pode ser visto, a melhor performance foi obtida utilizando-se dez redes, duas características (quatro agrupamentos gerados), ligações de dois dias e nove pontos no vetor de saída ( $2^9 = 512$  possíveis agrupamentos). Isso porque day (i.e., weekend/weekday) e time (00:00-08:00, 12:00-14:00) agregam diferentes padrões dos usuários.

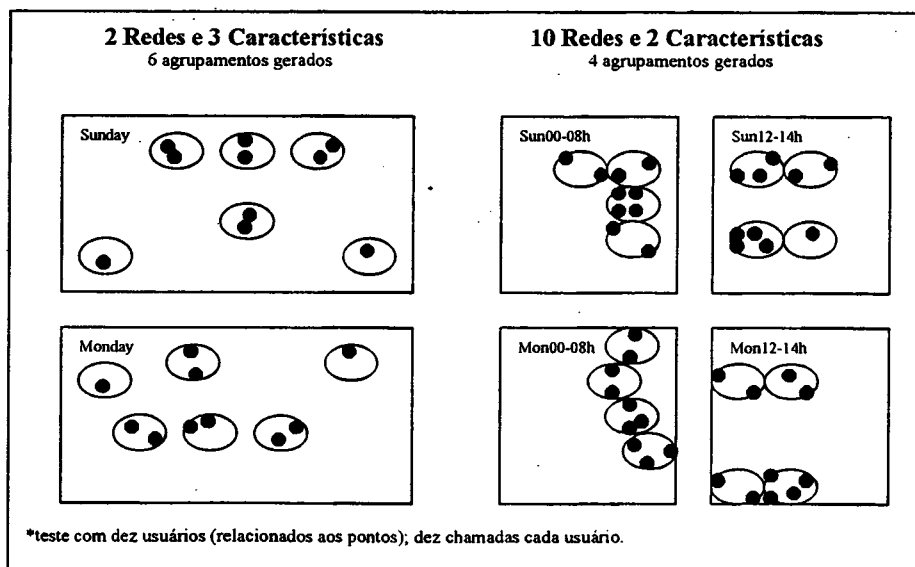


FIGURA 3.6 – COMPARAÇÃO DOS RESULTADOS DO EXPERIMENTO (SENSIBILIDADE).

<sup>36</sup> O resultado final considerando uma rede e as quatro características não foi possível obter devido às limitações de software e hardware.

Os resultados obtidos na Figura 3.6 estão fortemente associados ao aumento da sensibilidade dos dados de treinamento para um conjunto particular de dados que exibem alta similaridade, i.e., as ligações já pertencem a um dia específico (Sunday/Monday) e a um horário/tarifa específico (das 12h às 14h, e da 0h às 8h).

A fase de teste permitiu trabalhar com a sensibilidade de classificação. Pequenas alterações inseridas no conjunto de dados devem gerar – e geraram – a mesma saída (agrupamento), a fim de evitar que falsos alarmes sejam enviados. De maneira similar, grandes alterações devem resultar – e resultaram – em um diferente agrupamento para o usuário, a fim de garantir que o usuário seja avisado sobre a existência de uma fraude.

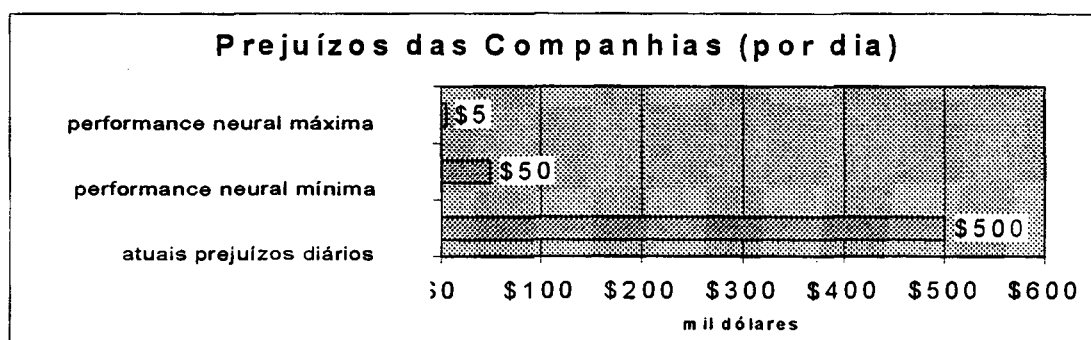


FIGURA 3.7 - ECONOMIA DA OPERADORA UTILIZANDO O ALGORITMO KOHONEN.

Na Figura 3.7 é apresentado quanto as companhias de telecomunicações podem economizar utilizando o sistema com esta rede neural. Por exemplo, considerando a performance mínima, foi obtida uma redução dos prejuízos de US\$500.000 dólares<sup>37</sup> para US\$50.000 diários, o que representa uma redução de 10%; enquanto em máxima performance da rede neural obteve-se uma redução para apenas 1% dos atuais prejuízos. Na verdade, existem diversos parâmetros que devem ser considerados, os quais são independentes do sistema:

- (1) a disponibilidade das ligações, que pode levar minutos ou horas, dependendo da tecnologia da empresa de telecomunicações;
- (2) o tempo para treinar e testar a rede neural, que depende da quantidade de dados, da quantidade de pontos do vetor de saída e dos recursos computacionais disponíveis; e
- (3) a sensibilidade do classificador sobre a variação dos dados *on-line*<sup>38</sup>.

Todavia, este sistema indica uma significativa redução dos prejuízos das companhias de telecomunicações, e em acréscimo espera-se reduzi-los ainda mais<sup>39</sup>.

<sup>37</sup> <www.uol.com.br/info/infonews/091999/>.

### 3.3 Uso de Classificador Supervisionado – RBF

Neste Capítulo 3.3 é proposto o emprego do algoritmo supervisionado Função de Base Radial (*Radial Basis Function* – RBF) para o reconhecimento dos padrões dos usuários de telecomunicações (LEE & RHEE, 1993). RBF é um excelente classificador e também um aproximador universal de funções (TODESCO, 1995).

#### 3.3.1 Algoritmo RBF

A arquitetura da RBF consiste em uma camada de entrada, uma camada escondida e uma camada de saída. Os nós da camada de saída formam uma combinação linear e usam o classificador de tipo Kernel. Veja a Figura 3.8.

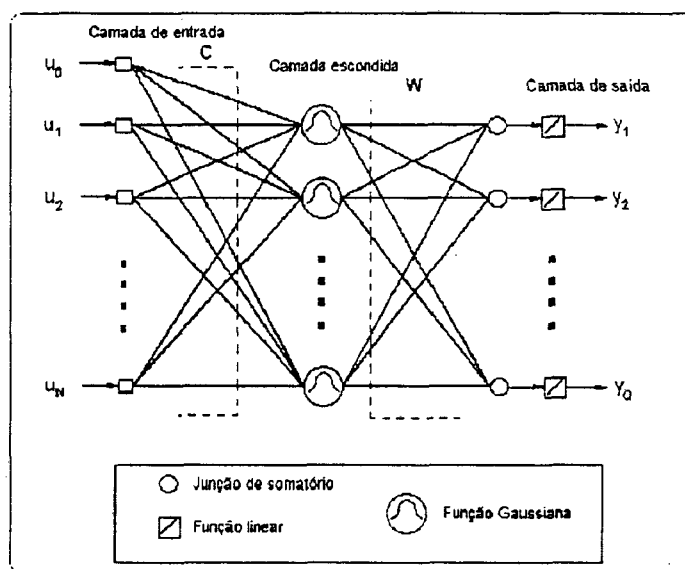


FIGURA 3.8 – FUNÇÃO DE BASE RADIAL.

A função RBF em sua camada escondida produz uma resposta para os estímulos de entrada (padrões). Quando a entrada está dentro de uma pequena região localizada no espaço de entrada, a função RBF produz uma resposta significativamente diferente de zero. A entrada representa nodos de entrada, i.e., unidades sensoriais. Cada função de ativação requer um

<sup>38</sup> É importante enfatizar que a primeira chamada de um clone ou impostor nunca é evitada.

<sup>39</sup> Note que, em 1998, as empresas nos EUA anunciaram um prejuízo de US\$33.400 milhões e, em 1997, as companhias brasileiras anunciaram um prejuízo de US\$33.000 milhões, por exemplo.

centro e um parâmetro escalar. A função *Gauss* é utilizada nesta investigação como função de ativação (LEE & RHEE, 1993). Nesse caso, a rede neural pode ser utilizada para tomar decisões de máxima correção, i.e., determinar quais dos vários centros é mais similar ao vetor de entrada. Dado  $x-c$  como vetor de entrada, a saída de um nó pode ser definida como segue:

$$f = (x - c) = \frac{1}{(2\pi)^{n/2} \sigma_1 \sigma_2 \dots \sigma_n} \exp \left\{ -\frac{1}{2} \sum_{j=1}^n \left( \frac{x_j - c_j}{\sigma_j} \right)^2 \right\} \quad (5)$$

onde  $n$  é o número de dados de entrada, e  $\sigma_1, \sigma_2, \dots, \sigma_n, j=[1,n]$  determina a dispersão escalar em cada direção. Para aumentar a funcionalidade da função  $f$ , propõe-se usar a distância *Mahalanobis* na função *Gauss*, definida como segue:

$$f = (x - c) = \frac{1}{(2\pi)^{n/2} |K|^{1/2}} \exp \left\{ -\frac{1}{2} (x - c)^T K^{-1} (x - c) \right\} \quad (6)$$

onde  $K^{-1}$  é o inverso da matriz co-variância  $X$ , associada com o nó da camada escondida  $C$ .

Dados  $n$ -vetores (dados de entrada) de  $p$ -amostras, representando  $p$ -classes, a rede pode ser inicializada com o conhecimento dos centros (i.e., localização das amostras). Se a amostra do vetor  $J$ -<sup>th</sup> é representada, então os pesos da matriz  $C$  podem ser definidos como  $C = [c_1 c_2 \dots c_3]^T$ , de modo que os pesos na camada escondida do nodo  $j$  são compostos de um vetor de centros. A camada escondida é a soma ponderada das saídas da camada escondida. Quando apresentado um vetor de entrada para a rede, a rede implementa  $Y$ , como segue:

$$y = W \cdot f(\|x - c\|) \quad (7)$$

onde  $f$  representa o vetor de saída da camada escondida, e  $C$  representa o vetor do centro correspondente. Após fornecer alguns dados com os resultados desejados, os pesos da Matriz  $W$  podem ser determinados usando-se o algoritmo *Least Mean Square* (LMS). O aprendizado na camada escondida é executado usando-se um método não-supervisionado, tal como um algoritmo heurístico ou de agrupamento, ou supervisionado, i.e., nodos  $C$  que representam as conexões entre a camada de entrada e a camada escondida. O algoritmo *K-means* é a técnica mais comum<sup>40</sup> empregada para determinar esses centros (TODESCO, 1995). Uma estrutura geral desse algoritmo é apresentada na Figura 3.9.

<sup>40</sup> Outros estudos têm usado ou algoritmos *supervised learning*, *self-organised learning*, ou *minimum orthogonal least squares algorithm* para encontrar esses centros.

**Initialize** the centers of clusters  $C_m$ ,  $m = 1, 2, \dots, M$ : can be randomly or the first  $M$  training patterns.  
**Repeat**  
 // To group all patterns with the nearest cluster center.  
**For all**  $u_n$  **do**  
 Let  $u_n$  to  $\Theta_m$ , where  $\|u_n - c_m\| = \min_m \|u_n - c_m\|$ .  
**End;**  
 // To calculate the clusters centers.  
**For all**  $c_m$  **do**  

$$c_m = (1/M_m) \sum_{u_n \in \Theta_m} u_n$$
**End;**  
**Until** do not exist more changes in the clusters attributions.

FIGURA 3.9 – ALGORITMO K-MEANS.

Para determinar  $\sigma^2$ , o parâmetro de variação para a função *Gauss*, pode-se escolher: (i) aproximá-lo da média da distância entre os dados de treinamento; (ii) calcular as distâncias entre os centros em cada dimensão e usar alguma percentagem desta distância como fator escalar para aproximar  $\sigma^2$ ; e (iii) usar o algoritmo *p-nearest neighbour*. Neste trabalho foi utilizada a última opção. As maiores motivações em utilizar redes neurais artificiais para a classificação de usuários de telefonia móvel são: (1) elas têm a capacidade intrínseca de aprender com dados de entrada e generalizar; (2) a rede é não-paramétrica e faz mais suposições considerando a distribuição dos dados do que os tradicionais métodos estatísticos (*Baysian*); e (3) a rede é capaz de criar limites de decisões que são altamente não-lineares no espaço de características (TODESCO, 1995).

Dessa forma, o algoritmo RBF apresentado acima, construído para a classificação dos usuários, tem como finalidade gerar o arquivo *Baseline*, que contém os números dos usuários da companhia telefônica com seus respectivos padrões. Esse arquivo constitui a base de dados utilizada pela implementação CORBA (Capítulo 4), onde as ligações *on-line* são comparadas com essa base de dados a fim de detectar possíveis fraudes, i.e., identificar se ligações de um usuário estão fora do padrão que consta para o mesmo nesse arquivo.



### 3.3.2 Experimentos Iniciais

O algoritmo para a classificação dos usuários de telecomunicações segundo seus padrões de utilização do telefone apresentado acima (item 3.3), que compreende *K-Means*, *P-NearestNeighbour* e *Gauss* (para a obtenção dos centros, variância e saída da camada escondida, respectivamente), foi implementado com os *softwares MatLab* (MATLAB, 1992) e *ToolBox* (DEMUTH & BEALE, 1998).

Neste item 3.3.2, os experimentos de classificação com o método supervisionado *RBF* foram realizados com o uso dos conhecidos dados de *Copenhagen*, utilizados por vários pesquisadores (TODESCO, 1995) em vez de dados reais (i.e., ligações telefônicas reais, que serão utilizadas no próximo item 3.3.3). Esse procedimento permitiu comprovar a eficiência do algoritmo (*RBF*), uma vez que tais dados já foram anteriormente classificados utilizando-se outros algoritmos – e dessa forma as taxas de erro de cada algoritmo puderam ser comparadas.

#### 3.3.2.1 Definição das Características e dos Agrupamentos

Os dados, referenciados aqui como *Copenhagen data*, foram classificados em sete tipos. Veja a Figura 3.10:

- |   |
|---|
| <ol style="list-style-type: none"> <li>1) usuários que fazem apenas ligações locais curtas;</li> <li>2) usuários que fazem muitas ligações locais;</li> <li>3) usuários que fazem poucas ligações de longa distância;</li> <li>4) usuários que fazem muitas ligações de longa distância;</li> <li>5) usuários que fazem poucas ligações curtas internacionais;</li> <li>6) usuários que fazem muitas ligações internacionais curtas; e</li> <li>7) usuários que fazem muitas ligações internacionais longas.</li> </ol> |
|---|

FIGURA 3.10 – CLASSIFICAÇÃO DOS USUÁRIOS.

Salienta-se que a classe 2 inclui a classe 1, a classe 3 inclui a classe 2, e assim por diante.

#### 3.3.2.2 Treinamento e Teste da Rede

Os dados foram armazenados em quatro arquivos (A1.dat, A2.dat, B1.dat, e B2.dat), onde A1 e B1 incluem 4.061 amostras e A2 e B2 4.050 amostras. A1 é utilizado para a geração dos centros, A2 para treinar a rede, B1 para testar a rede e B2 para a gerar a taxa de erro.

Cada amostra de A1 e A2 contém três características: 1) o número do telefone chamado; 2) o dia/hora da ligação; e 3) a duração da ligação. Cada amostra de A2 e B2 contém duas informações: 1) o número do telefone que fez a ligação; e 2) o padrão a que essa ligação pertence. Cada amostra nos arquivos A1 e B1 corresponde a um padrão nos arquivos A2 e B2, respectivamente. É dessa maneira que a rede aprende de forma supervisionada.

### 3.3.2.3 Cálculo do Erro

Os dois conjuntos de dados (1 e 2) foram cruzados como dados de treinamento e de testes a fim de gerar a taxa de erro de classificação final. Veja na Tabela 3.2 a taxa de erro obtida.

TABELA 3.2 – NÚMERO DE NEURÔNIOS NA CAMADA ESCONDIDA E RESPECTIVA TAXA DE ERRO.

Neurônios na camada escondida	Taxa de erro
50	5.0185
107	4.3758
100	4.4252
110	4.2027
111	4.2027
127	4.3511

Durante os experimentos, variou-se a quantidade de neurônios da camada escondida entre 50 e 150. Os resultados obtidos estão sumarizados na Tabela 3.2. Como pode ser observado, um resultado muito bom, de 4,2027%, foi obtido utilizando-se 110 neurônios na camada escondida.

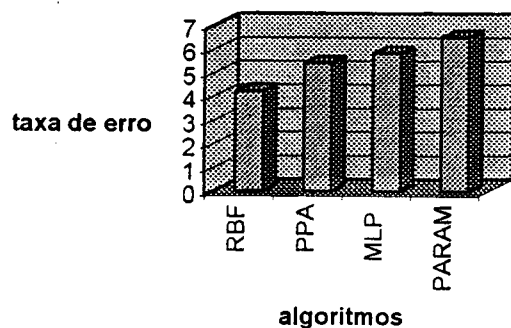


FIGURA 3.11 – TAXA DE ERRO DO ALGORITMO.

Este resultado de 4,2025% é considerado bom, i.e., eficiente, pois, utilizando-se estes mesmos dados com outros algoritmos, a taxa de erro foi maior. Pode-se citar *Back Propagation* (PPA), *Multi-Layer Perceptrons* (MLP) e *Parametrical*, que apresentaram taxas de erro de 5,4%, 5,8%, e 6,5%, respectivamente (veja a Figura 3.11).

O resultado obtido com o classificador Função de Base Radial (*Radial Basis Function*) para a detecção de fraudes na telefonia móvel, considerando a taxa de erro de 4,2%, pode reduzir significativamente as perdas das companhias telefônicas, conforme pode ser observado na Figura 3.12.

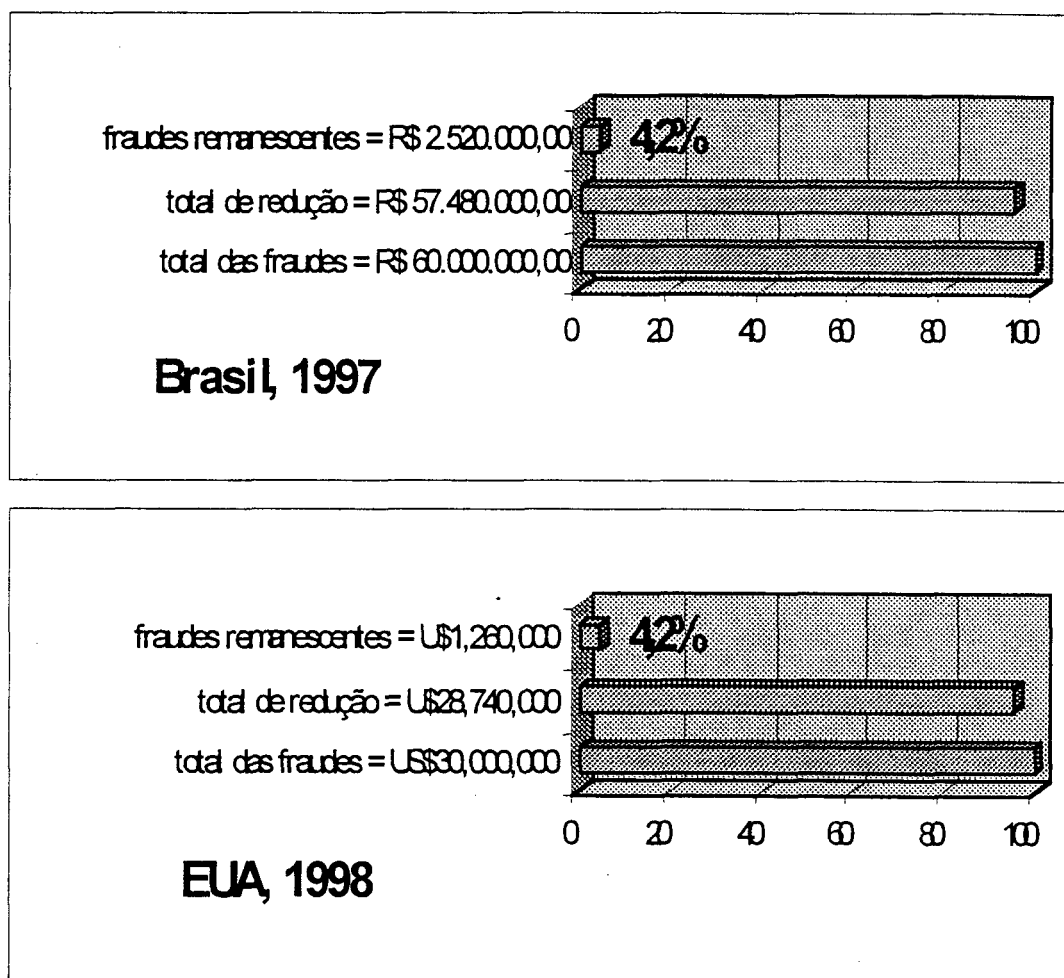


FIGURA 3.12A – REDUÇÃO DAS PERDAS COM O USO DA RBF (EM REAIS, BRASIL 1997).

FIGURA 3.12B – REDUÇÃO DAS PERDAS COM O USO DA RBF (EM DÓLARES, EUA 1998).

Dessa forma, considerando as perdas no Brasil<sup>41</sup>, em 1997, estas poderiam ser reduzidas de R\$ 60.000.000,00 para R\$ 2.520.000,00; e considerando as perdas nos Estados Unidos, em 1998, estas poderiam ser reduzidas de US\$ 33.400.000 para US\$ 1.402.800. Segundo dados do IDC<sup>42</sup>, de setembro de 1999, atualmente as companhias perdem meio milhão de dólares por dia com esses tipos de fraudes.

### 3.3.3 Experimentos Avançados

Após obter um resultado encorajador com o uso da RBF através dos dados de *Copenhagen* (item 3.3.2) – o que permitiu a comparação com outros algoritmos, neste item 3.3.3 o mesmo algoritmo é utilizado agora com dados reais, ou seja, ligações de uma companhia telefônica brasileira<sup>43</sup>.

#### 3.3.3.1 Definição das Características e dos Agrupamentos

A escolha de um classificador supervisionado (neste caso, RBF) é realizada quando existem informações suficientes sobre os dados a serem classificados, ou seja, quais são os grupos em que os dados podem ser classificados considerando suas características relevantes. Por exemplo: (i) opta-se por supervisionado: para agrupar usuários de telecomunicações de acordo com seus hábitos de utilização do aparelho, pois existem elementos suficientes para tal (por exemplo, pode-se agrupar os usuários que fazem apenas ligações locais curtas, os que fazem poucas ligações curtas interurbanas, os que fazem muitas ligações internacionais longas, etc.); e (ii) opta-se por não-supervisionado: para agrupar produtos que clientes de um supermercado costumam comprar na mesma compra, para deixá-los em prateleiras próximas (o exemplo clássico é de que não é possível supor que cerveja e fralda pertencem ao mesmo grupo).

Dessa forma, considerando as características das chamadas e analisando os arquivos de ligações de uma companhia telefônica, foi possível definir os agrupamentos. Os usuários foram agrupados em uma ordem hierárquica crescente (do menos provável ao mais provável padrão de fraude) de acordo com o tipo de ligações que efetuam<sup>44</sup>. Veja a Figura 3.13.

<sup>41</sup> GONÇALVES, D.N. Olho na conta: cresce o número de vítimas de telefones celulares clonados. *Veja*, São Paulo, p. 86, 8 out. 1997.

<sup>42</sup> <[www2.uol.com.br/info/infonews/091999/](http://www2.uol.com.br/info/infonews/091999/)>, <[www2.uol.com.br/info/infonews/091999/](http://www2.uol.com.br/info/infonews/091999/)>, <[www.uol.com.br/idgnow/entre/entre22.htm](http://www.uol.com.br/idgnow/entre/entre22.htm)>.

<sup>43</sup> Nesses experimentos são consideradas todas as ligações efetivadas pelos usuários da Telefônica Celular no Rio Grande do Sul, nos dias 01/08/99 (domingo) e 02/08/99 (segunda-feira), totalizando 4.255.973 ligações.

<sup>44</sup> Investigações poderão ser realizadas a fim de verificar o comportamento (performance) de um maior refinamento de grupos, isto é, classificar as ligações internacionais por continentes e as interurbanas por regiões.

1)	local;	
2)	interurbana,	dia/horário com desconto, curta;
3)	interurbana,	dia/horário sem desconto, curta;
4)	interurbana,	dia/horário com ou sem desconto, longa;
5)	especial,	dia/horário com ou sem desconto, curta;
6)	especial,	dia/horário com ou sem desconto, longa;
7)	internacional,	dia/horário com desconto, curta;
8)	internacional,	dia/horário sem desconto, curta;
9)	internacional,	dia/horário com desconto, longa; e
10)	internacional,	dia/horário sem desconto, longa.

FIGURA 3.13 – CLASSIFICAÇÃO DOS USUÁRIOS EM 10 GRUPOS HIERÁRQUICOS.

No menor nível (1) estão os usuários que fazem apenas ligações locais (em qualquer horário e de qualquer duração). No maior nível (10) estão os usuários que fazem inclusive ligações internacionais, em horários sem desconto e de longa duração.

Para que esse agrupamento seja possível, quatro características precisam ser consideradas em cada ligação telefônica que é efetivada:

- (1) `CalledNumber` (se local, interurbana, especial ou internacional);
- (2) `Day` (se em dia útil ou não);
- (3) `Time` (0h às 8h, 8h às 12h, 12h às 14h, 14h às 18h ou 18h às 24h); e
- (4) `Duration` (até 1, 15, 30, 60, 120 ou mais de 300 minutos).

Através da característica `CalledNumber` é possível identificar se a ligação é local, interurbana, internacional ou especial (tais como 0900). Através da característica `Day` é possível verificar se a ligação foi realizada em dia útil ou final de semana/feriado e da característica `Time` é possível verificar o horário em que a ligação foi iniciada. Estas duas últimas características (i.e., `Day` e `Time`) permitem identificar a tarifa da ligação (i.e., com ou sem desconto), muito importante para a detecção de fraudes, já que a maioria dos usuários procura restringir suas ligações em dias e horários em que a tarifa é menor (ao contrário do fraudador, que não tem essa preocupação). Por fim, através da característica `Duration` pode-se identificar longas chamadas, também mais comuns em ligações a partir de usuários fraudadores.

As combinações de todos os valores que as quatro características podem assumir totalizam  $4 \times 2 \times 5 \times 6 = 240$ . Desse modo, a base de dados a ser utilizada deve contemplar todas essas 240 combinações, a fim de que a rede possa aprender (classificar) a que grupo pertence cada um dos tipos possíveis de ligações telefônicas. Nesse experimento, com o objetivo de obter maior precisão na aprendizagem, utilizam-se não só 240 ligações, mas 10 conjuntos dessas combinações, ou seja, 2.400 ligações telefônicas<sup>45</sup>. Esses dados constituem o arquivo `Ligacoes.dat`<sup>46</sup> (sendo o mesmo uma matriz  $A_{2400,4}$ ) a partir do qual os 10 grupos são gerados. Veja a Figura 3.14.

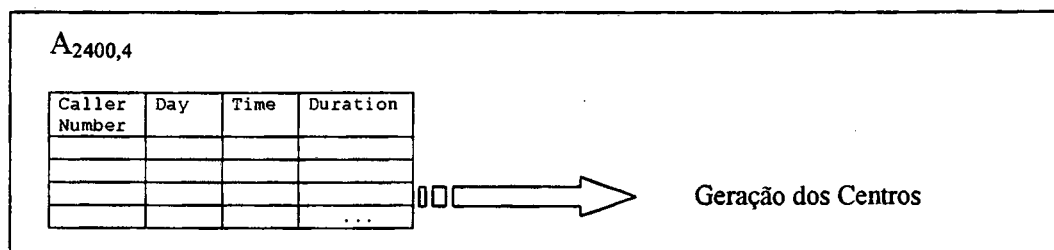


FIGURA 3.14 – MATRIZ  $A_{2400,4}$  UTILIZADA PARA A GERAÇÃO DOS CENTROS DOS AGRUPAMENTOS.

A geração dos agrupamentos a partir desses dados é então realizada com os algoritmos *k-means* e *p-nearest-neighbour* (para geração dos centros e raios, respectivamente) como mostrado previamente no item 3.3.1.

### 3.3.3.2 Treinamento da Rede

Para que a rede aprenda quais ligações pertencem a determinado agrupamento (i.e., classifique adequadamente), é necessário existir uma matriz  $B_{2400,10}$  (arquivo `Padroes.dat`)<sup>47</sup> contendo o mesmo número de linhas (registros/ligações) que a matriz  $A_{2400,4}$ , de modo que cada registro/ligação da matriz  $A_{2400,4}$  seja associado a um registro da matriz  $B_{2400,10}$ , i.e., o seu padrão correspondente. Cada um dos 10 padrões é definido por um vetor de 10 elementos, sendo nove elementos iguais a "0" e um elemento igual a "1". A posição do vetor em que o elemento é igual a "1" caracteriza o agrupamento/nível hierárquico a que pertence a ligação. Veja a Figura 3.15.

<sup>45</sup> Investigações poderão ser realizadas com 100, 1.000, 10.000 conjuntos de 240 ligações, por exemplo, a fim de que se possa comparar a performance de classificação – mas isso depende do uso de melhores equipamentos (hardware).

<sup>46</sup> Dividiu-se esse arquivo `Ligacoes.dat` (matriz  $A_{2400,4}$ ) em dois: `Lig_a.dat` e `Lig_b.dat`.

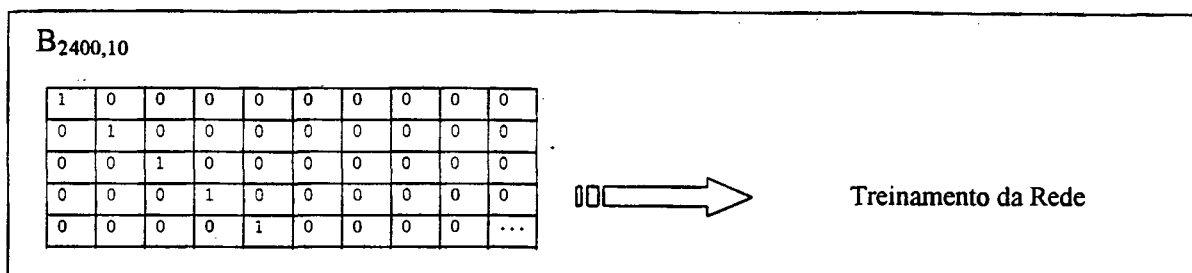


FIGURA 3.15 – MATRIZ  $B_{2400,10}$  UTILIZADA PARA TREINAR A REDE.

Dessa forma, cada linha (registro/ligação) da matriz  $A_{2400,4}$  tem seu padrão correspondente na matriz  $B_{2400,10}$ , o que caracteriza o treinamento – supervisionado – da rede. Por exemplo, a linha 1 e a linha 100 da matriz  $A_{2400,4}$  podem ter como características, respectivamente:

- 1) ligação local; domingo; 15h10min; e
- 2) ligação local; Segunda-feira; 3h2min.

E então ambas devem possuir mesmo padrão (constituindo uma ligação local, curta e de horário de tarifa reduzida) associado à matriz  $B_{2400,10}$ , ou seja, os registros 1 e 100 conterão ambos os vetores idênticos, iguais a: “1 0 0 0 0 0 0 0 0 0” .

### 3.3.3.3 Teste da Rede

Depois que a rede é treinada, ou seja, após a rede aprender a reconhecer a qual grupo uma ligação pertence a partir das quatro características consideradas, é necessário associar cada usuário a um dos 10 padrões.

Para tanto, as ligações de cada usuário (contidas no arquivo `Ligacoes.dat`) são separadas em arquivos distintos, ou seja, cada usuário tem um arquivo associado (`UserX.dat`)<sup>48</sup> contendo unicamente as suas ligações (quantidade de ligações x 4) . Então a rede é executada para cada arquivo de usuário, de forma a gerar o vetor/padrão correspondente para cada usuário (`UserXpattern.dat`). Todos esses resultados das execuções da rede (i.e., os vetores contendo os números dos usuários e seus respectivos padrões) são armazenados em um arquivo chamado `Baseline.dat` Veja a Figura 3.16.

<sup>47</sup> Dividiu-se esse arquivo `Padroes.dat` (matriz  $B_{2400,10}$ ) em dois: `Pad_a.dat` e `Pad_b.dat`.

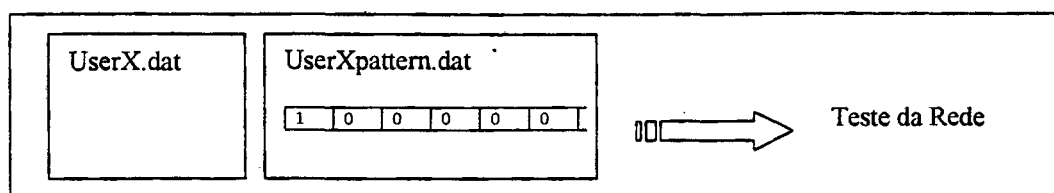


FIGURA 3.16 – TESTE DA REDE.

Ou seja, a rede é executada quantas vezes for o número de usuários; e para cada um deles é gerado o seu vetor/padrão.

### 3.3.3.4 Cálculo do Erro

Após a fase de teste, a rede é utilizada para detectar fraudes e verificar o erro, na medida que as chamadas são completadas (*on-line*). A cada 10 ligações de um usuário estas geram 10 vetores de padrões, sendo um desses o vencedor, ou seja, é o padrão do usuário correspondente a estas 10 ligações<sup>49</sup>. Esse padrão é então comparado com o padrão que consta no arquivo *Baseline.dat* para este usuário. Se os padrões não forem iguais, então o alarme de fraude é enviado ao gerente do sistema. Dessa forma, os alarmes falsos são contabilizados como erro.

O procedimento de reunir 10 ligações para só então executar a rede contribui para o controle da quantidade de ligações (principalmente internacionais), realizadas por cada usuário, uma vez que uma das características das fraudes é o uso constante do telefone, enquanto que um telefone legal não possui a mesma frequência de uso. É importante comentar, que as 10 ligações de um mesmo usuário, armazenadas pelo módulo agente, antes mesmo de serem enviadas para a rede neural, já são analisadas quanto à existência de ligações impossíveis, p.ex., a existência de uma ligação realizada em Florianópolis às 14h e outra ligação realizada no Rio de Janeiro às 14h10min (pois o tempo mínimo de deslocamento entre essas duas cidades é superior aos 10 minutos do exemplo).

<sup>48</sup> x é o número do telefone do usuário.

<sup>49</sup> Este número é configurável no agente Java, através de uma barra de rolagem. Os testes realizados apontam dez como o valor adequado para *default*.



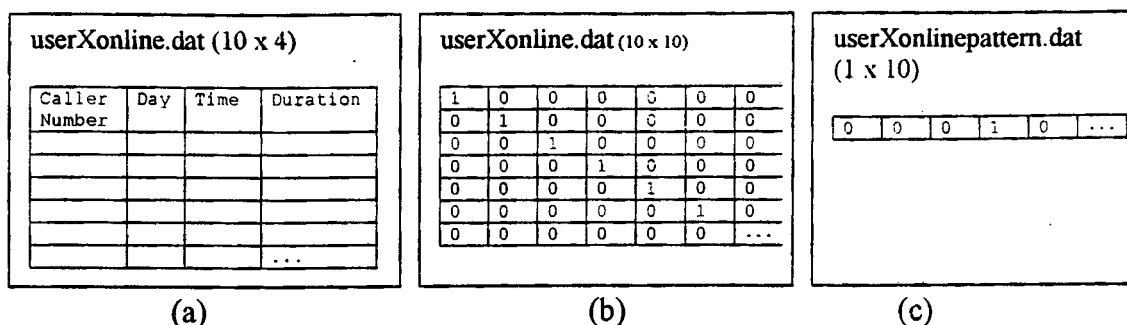


FIGURA 3.17 – EXECUÇÃO ON-LINE E CÁLCULO DO ERRO.

A Figura 3.17 mostra que após a reunião de 10 ligações realizadas *on-line* de cada usuário (a) são gerados 10 vetores correspondentes (b). O vetor vencedor (c) é o padrão atual do usuário associado as suas últimas 10 ligações. Caso esse padrão atual tenha um valor superior ao valor armazenado anteriormente (no arquivo *Baseline.dat*) um alarme é enviado para este usuário. Os alarmes falsos são contabilizados como erro. A taxa de erro obtida com a utilização de dados reais foi ainda melhor que quando utilizados os dados de *Copenhagen*. A menor taxa de erro obtida foi de 2,5%, com 80 neurônios na camada escondida.

Um resumo do procedimento para a obtenção dessa taxa de erro de 2,5% é apresentado a seguir.

- (1) Inicialmente foi realizada uma análise heurística nos padrões de celulares clonados (contas telefônicas) e nos padrões de uso dos usuários legais. As informações necessárias foram obtidas a partir de arquivos de ligações dos usuários de uma companhia telefônica e através de informações obtidas por um engenheiro de controle de fraudes de uma companhia telefônica brasileira.
- (2) Após esta análise foram consideradas quatro características relevantes em cada ligação, bem como os valores que cada uma poderia assumir; e o número de agrupamentos (*clusters*) adequado foi definido em 10.
- (3) Utilizou-se então o algoritmo não supervisionado Kohonen para a geração automática dos agrupamentos a fim de comprovar a eficiência do método heurístico. A execução

do Kohonen, com 2.400 amostras e um espaço de saída de 6 pontos, gerou 8 agrupamentos, o que confirma que 10 agrupamentos é um bom número para os dados considerados. O uso do Kohonen poderá ser também útil quando em futuras investigações as quatro características consideradas forem mais refinadas (tais como internacionais em diversos países e interurbanas em diversos estados) e os agrupamentos então não forem mais tão óbvios para serem identificados heurísticamente.

- (4) A amostra foi validada como sendo representativa através da geração de gráficos correspondentes à frequência de ocorrências em cada variável da amostra. Ou seja, a amostra apresentou a mesma frequência (das características Duration, CalledNumber e Day) que o arquivo de dados completo. Veja as Figuras 3.18a, 3.18b e 3.18c.

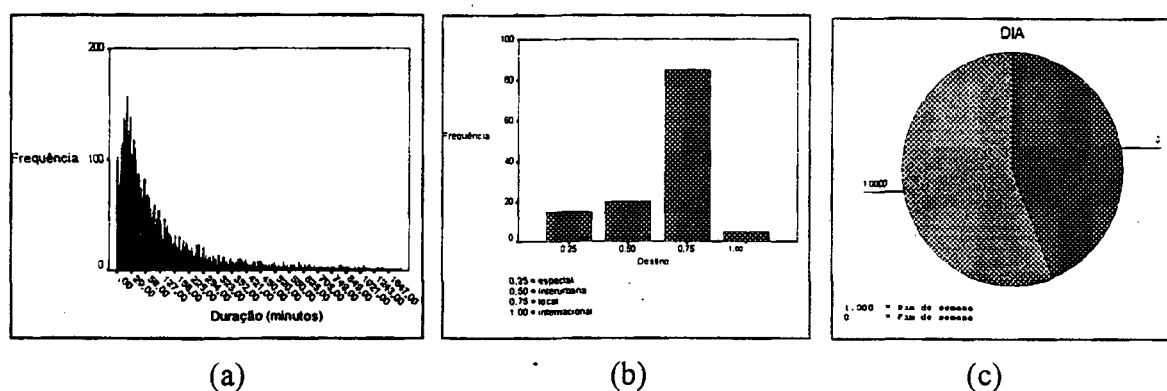


FIGURA 3.18 – FREQUÊNCIA DAS CARACTERÍSTICAS DURAÇÃO, DESTINO E DIA.

Estes gráficos, gerados com a ferramenta estatística *Statistical Package for the Social Sciences – SPSS*<sup>50</sup>, mostram que a maioria das ligações tem a duração de 30 minutos, são locais, e praticamente a metade em dia de semana e a metade em fim de semana.

- (5) A seguir, considerando as quatro características e os valores que as mesmas podem assumir ( $4 \times 2 \times 5 \times 6 = 240$ ), construíram-se duas matrizes com 2.400 amostras cada, ou seja, contendo 10 conjuntos com todas as combinações possíveis a fim de ensinar à rede a que padrão cada um dos 240 tipos de ligações telefônicas pertence. Embora considera-se que 10 conjuntos seja uma amostra adequada (i.e., representativa), este número também está relacionado aos limites de processamento. Pretende-se como trabalhos futuros testar dois

<sup>50</sup> [http://www.nyu.edu/acf/pubs/SPSS\\_Win/SPSSwindoc\\_ToC.html](http://www.nyu.edu/acf/pubs/SPSS_Win/SPSSwindoc_ToC.html)

arquivos (já formatados) de 7.000 conjuntos associados a dois dias completos de ligações telefônicas, procedimento que os limites computacionais atuais disponíveis para esta pesquisa não suportam.

- (6) Estas duas matrizes de 2.400 observações foram associadas a duas outras matrizes com os padrões correspondentes (de 1 a 10) e dessa forma a rede pôde ser treinada de modo supervisionado através do algoritmo RBF.

As taxas de erro obtidas variaram conforme as configurações testadas na rede. Variando a quantidade de observações (de 2 a 10 conjuntos) e a quantidade de neurônios na camada escondida (de 10 a 150), a melhor taxa de erro obtida foi de 2,5% com 10 conjuntos de 240 amostras (ou seja, 2.400 em cada uma das duas matrizes) e 80 neurônios na camada escondida. Esta taxa de erro de 2,5% foi o melhor resultado de classificação obtido por esta investigação científica.<sup>51</sup>

### 3.4 Resultados

Se, por um lado, existe uma grande quantidade de pesquisas associadas ao *hardware* das redes de telecomunicações sem fio, ou seja, ao aparelho celular digital, tornando-o muito mais seguro contra as fraudes de clonagem, por outro lado, existem poucas investigações associadas a soluções eficazes contra fraudes via *software*<sup>52</sup>. Importante salientar que as fraudes associadas à habilitação (*subscription*) ainda são muito independentes do *hardware* utilizado e que as soluções via *software* são as mais adequadas atualmente.

Este Capítulo 3 apresentou uma solução via *software* para a detecção de intrusão em redes móveis. A solução consiste em um sistema que emprega redes neurais contra as fraudes de clonagem e de habilitação, já que padrões alterados podem ser comparados com padrões de antigos inadimplentes. Apresentou-se também a performance do sistema, que incluiu dados reais de uma companhia telefônica. Os resultados mostraram a eficácia do sistema SSTCC em identificar fraudes e, desta forma, reduzir os prejuízos das companhias telefônicas e os danos que poderiam causar aos seus clientes. Considerando a taxa de erro obtida na classificação (de 2,5% utilizando 80 neurônios na camada escondida), os atuais prejuízos de US\$ 500.000 diários podem ser reduzidos para US\$ 12.500 diários<sup>53</sup>.

<sup>51</sup> Observou-se que testes com dados inadequados (tipo e quantidade) podem produzir taxas de erros que chegam a 30%.

<sup>52</sup> Ericsson e Nortel são dois exemplos de empresas que desenvolvem *software* contra fraudes na telefonia móvel.

<sup>53</sup> Este valor não contabiliza as fraudes detectadas via SETWeb (pelos próprios usuários), nem pelo descréscimo natural das fraudes, visto que a fraude torna-se menos lucrativa para os fraudadores.

O sistema de detecção e eliminação do clone tem como importante característica o monitoramento e aviso automático ao usuário. Dessa maneira, a companhia não necessita de muitos funcionários dedicados a essa atividade. O aviso ao usuário é realizado através de uma mensagem interativa gravada do tipo “despertador automático”, além do aviso que segue por correio. Os resultados indicam que o sistema reduz significativamente os prejuízos das companhias telefônicas, bem como as perdas/riscos que os (inocentes) clientes podem ter. Dessa forma, as companhias podem reduzir o preço dos seus serviços (ligações) e, por conseguinte, beneficiar os seus usuários.

Como trabalhos futuros, planeja-se investigar a performance do sistema quando as quatro características consideradas forem subdivididas; por exemplo, as ligações internacionais divididas em países e as interurbanas em estados. Essa divisão é muito importante para a detecção de inadimplentes, pois nesta fraude não se compara o atual padrão com o padrão armazenado do mesmo usuário, mas sim a similaridade do atual padrão com o dos inadimplentes anteriores – e nesse caso dez agrupamentos não são suficientes. Também pretende-se investigar a determinação do padrão do usuário *on-line* (a cada dez ligações do mesmo) não mais como o padrão vencedor (que mais vezes ocorre), mas sim analisando quantos padrões são diferentes do atual padrão do usuário. O objetivo desta investigação tem sua justificativa, pois, mesmo considerando que ligações fraudulentas ocorrem com maior frequência do que as ligações do próprio usuário (o que justifica optar apenas pelo vencedor), poderia também ocorrer o seguinte: o “vencedor” ter apenas duas ligações iguais e idênticas ao padrão do usuário no meio de oito diferentes, e dessa forma uma fraude poderia não ser identificada. Mas no caso de optar pela “maioria diferente” das dez em vez do “vencedor”, o sistema conseguiria considerar uma maioria (por exemplo oito) de ligações distintas e diferentes do padrão atual do usuário.

Em acréscimo, planeja-se investigar os resultados de classificação utilizando outras técnicas, tais como Raciocínio Baseado em Casos (RAMOS & ALVES, 1999, CRUZ et al., 1998) e ferramentas estatísticas como *SPSS*. Finalmente, considera-se a possibilidade de estender esse sistema para outras áreas, como a clonagem de cartões de crédito e o marketing direcionado pela determinação de padrões de consumo.

*"The choice of CORBA and Java combined with the open Interface Definition Language Interface leads to a highly open, extensible, and distributed solution. By having a protocol-defined interface, CORBA forces the separation of an object interface and implementation from its use."*  
(HAGGERTY & SEETHEARAMAN, 1998)  
*"CORBA offers a useful methodology and middleware for building interoperable databases."*  
(DOGAC et al, 1998)

## 4 Implementação da Gerência de Segurança Distribuída

Este Capítulo 4 apresenta a implementação da gerência de segurança em sistemas distribuídos através do suporte da arquitetura CORBA (HAGGERTY & SEETHEARAMAN, 1998, OMG, 1998). O **principal objetivo** do emprego desta técnica (arquitetura de distribuição) no escopo deste trabalho é prover a segurança adequada na comunicação entre agentes e gerente e na comunicação entre usuários e servidor Web. Para isso, diversos mecanismos são implementados em Java (ANEROUSIS, 1998, MCGRAW & FELTEN, 1997, NAUGHTON, 1986, OPPLIGER, 1999) a fim de garantir que a comunicação não seja interrompida, interceptada, modificada ou fabricada ilicitamente. Criptografia é um dos principais mecanismos utilizados, que juntamente com os mecanismos de assinatura e certificado digital tornam o controle de acesso e a comunicação muito mais seguros.

O Sistema SSTCC é implementado em Java sobre a arquitetura CORBA, de modo a se beneficiar dos recursos de segurança oferecidos por ambas as técnicas através de uma política de segurança adequada que fornece os serviços de controle de acesso, autenticação, confidencialidade, integridade, disponibilidade e não-repúdio. A implementação do sistema é realizada através do emprego das ferramentas *Java Developer Kit JDK 1.2.1*, *Symantec Visual Café 3.0 Data Base Engine Trial* e *Visibroker 3.4 for Java*.

Este Capítulo 4 está organizado como segue. Inicialmente, no item 4.1, são apresentados aspectos relevantes da gerência de segurança em sistemas distribuídos, considerando os tipos de ataques e os serviços de segurança correspondentes (item 4.1.1), as tecnologias e ferramentas utilizadas para a implementação desses serviços (item 4.1.2) e as características relevantes das redes de telefonia para as quais o sistema foi desenvolvido (item 4.1.3). No item 4.2, é apresentada a implementação dos Sistemas SSCC e SIPI e de seus mecanismos de segurança sobre a arquitetura CORBA (veja Anexo 3.1). No item 4.3 é descrita a implementação do Sistema SETWeb e de seus mecanismos de segurança implementados em Java (Veja Anexo 3.2). Finalmente, no item 4.4, são discutidos os resultados.

## 4.1 Gerência de Segurança e Redes de Telecomunicações

Uma das formas mais utilizadas para a proteção de informações contra ataques está baseada no conceito de níveis de segurança. Muito utilizada na área militar, é caracterizada pela categorização da informação em não classificada (*unclassified*), confidencial (*confidential*), secreta (*secret*), ultra secreta (*top secret*). Quando múltiplas categorias ou níveis de dados são definidos, então tem-se uma segurança multi-nível (*multilevel security*). Um sistema de segurança multi-nível deve garantir as seguintes regras:

- (1) leitura acima negada (*no read up*): um sujeito pode somente ler um objeto de nível de segurança igual ou inferior ao seu;
- (2) escrita abaixo negada (*no write down*): um sujeito pode somente escrever em um objeto de nível de segurança igual ou superior ao seu.

Essas duas regras, se garantidas corretamente, provêm a segurança multi-nível. Uma abordagem em que se tem pesquisado muito é baseada no conceito de monitor de referência (*reference monitor*). Veja abaixo na Figura 4.1, o monitor de referência (STALLINGS, 1995).

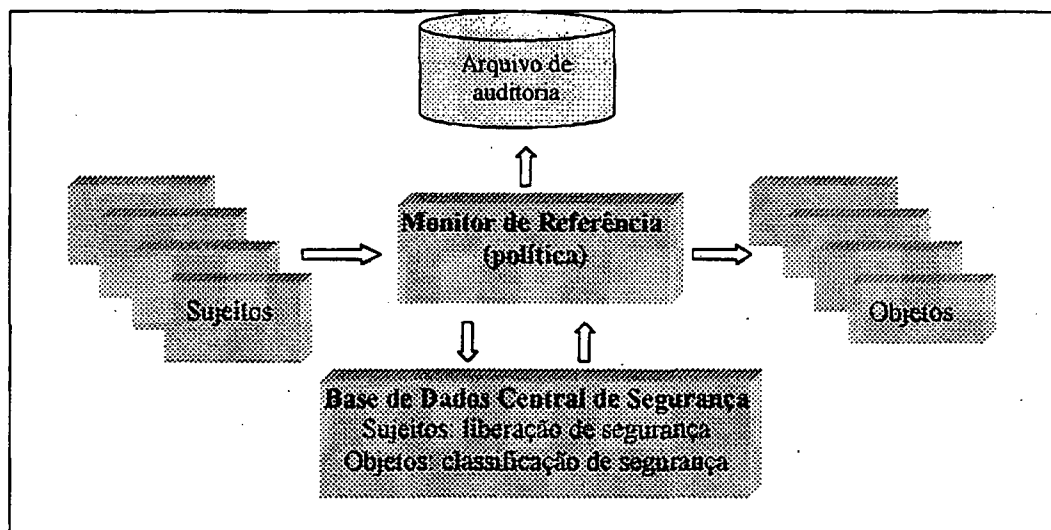


FIGURA 4.1 – MONITOR DE REFERÊNCIA.

O monitor de referência (*reference monitor*) pode estar em *hardware* ou no sistema operacional e controla o acesso de sujeitos a objetos conforme parâmetros de segurança dos sujeitos e dos objetos. O monitor de referência tem acesso ao arquivo base de dados central de segurança (*security kernel database*) que lista os privilégios de acesso, i.e., liberação/desobstrução de segurança (*security clearance*) de cada sujeito e atributos de

proteção, ou seja, nível de classificação (*classification level*) de cada objeto. Eventos importantes, como detecção de violações de segurança, por exemplo, são armazenados em um arquivo de auditoria (*audit file*). O Monitor de Referência faz cumprir as regras de segurança (leitura, escrita) e tem as seguintes propriedades:

- (1) completa mediação: as regras de segurança são cumpridas em cada acesso (e não somente quando um arquivo é aberto, por exemplo);
- (2) isolamento: o monitor de referência e a base de dados são protegidos de modificação não autorizada; e
- (3) verificabilidade: a correção do Monitor de Referência deve ser provável. Isto é, deve ser possível demonstrar matematicamente que o Monitor de Referência cumpre as regras de segurança e provê completa mediação e isolamento. Um sistema que provê verificação, i.e., **prova matemática** é referido como um Sistema Confiável (*Trusted System*).

Em um esforço para satisfazer suas próprias necessidades, e também como um serviço para a população, o Departamento de Defesa dos Estados Unidos, em 1981, estabeleceu o Centro de Computação Segura (*Computer Security Center*), dentro da Agência Nacional de Segurança (*National Security Agency – NSA*), com o objetivo de encorajar a ampla disponibilidade de Sistemas Confiáveis para Computadores (*Trusted Computer Systems*). O Centro classifica os produtos de acordo com o alcance das características de segurança que eles oferecem. Em essência, esse Centro visa avaliar produtos disponíveis comercialmente em relação aos requerimentos de segurança acima apresentados (SIMON, 1996).

#### 4.1.1 Ataques e Mecanismos de Segurança

Com o objetivo de ser alcançado um grau adequado de segurança para os sistemas distribuídos é definida uma **política de segurança** (i.e., os requisitos de segurança definidos por cada sistema). Uma política de segurança é descrita a partir da utilização de **modelos de segurança** (regras). Um **esquema de segurança** é a realização de uma política de segurança através de um conjunto de mecanismos de segurança. Um **mecanismo de segurança** é projetado para detectar, prevenir ou recuperar a rede de um ataque de segurança. Um **ataque de segurança** é qualquer ação que compromete a segurança das informações de uma organização, i.e., um **risco de segurança** intencionalmente efetivado. Os **serviços de segurança**, oferecidos por aplicações de gerência, fazem uso de um ou mais mecanismos de segurança e são definidos de acordo com uma política de segurança.

Existem quatro categorias gerais às quais os ataques<sup>54</sup> podem ser associados (STALLINGS, 1995). Veja a Figura 4.2.

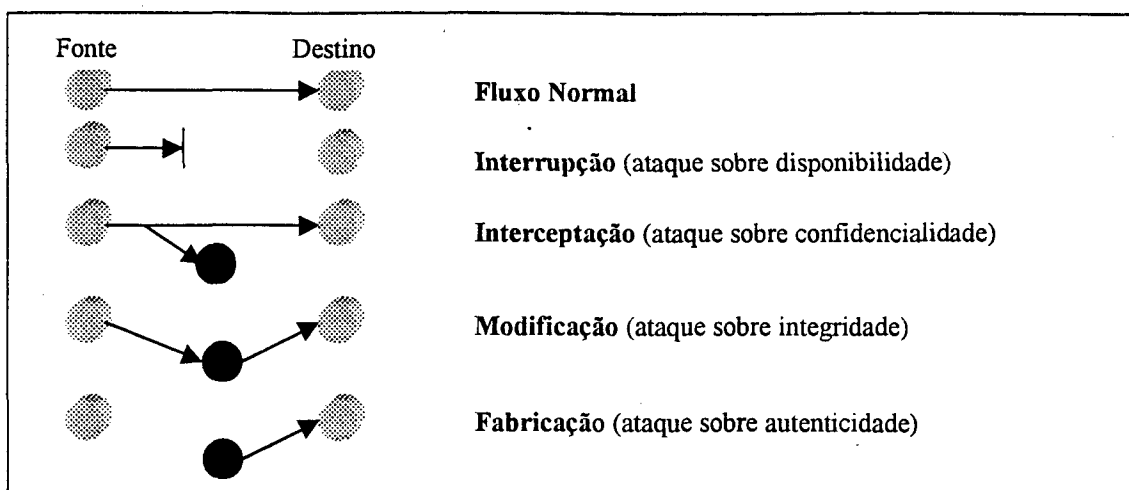


FIGURA 4.2 – ATAQUES NA SEGURANÇA.

Normalmente, uma **política** de segurança inclui:

- (1) procedimentos relevantes para a gerência de segurança (o que constitui atividades legítimas e autorizadas e os mecanismos de monitoramento da segurança);
- (2) níveis de responsabilidade (sob quais circunstâncias e para quem a responsabilidade é delegada); e
- (3) quais mecanismos de segurança devem ser empregados para garantir a política de segurança.

Os principais **mecanismos** que podem ser empregados para se alcançar as demandas de uma política de segurança são (SIMON, 1996):

- (1) autenticação é o processo de estabelecimento de prova de identidade. Dois níveis de autenticação são requeridos: 1) durante a conexão do cliente ao servidor (p.ex., associações erradas ou *user login*); e 2) para cada mensagem;
- (2) autorização é o processo de estabelecer quais principais<sup>55</sup> têm o direito de acessar um objeto particular. Uma vez que um usuário é autenticado, mecanismos de controle de acesso são requeridos para determinar quais recursos um usuário pode acessar e de quais maneiras;

<sup>54</sup> De acordo com pesquisa em 148 instituições financeiras brasileiras pela Módulo Security Solutions (responsável pela segurança na entrega das declarações do Imposto de Renda pela Internet), 35% dos *hackers* são funcionários da própria empresa e 50% dos ataques foram registrados nos últimos 6 meses (revista Veja, 14/07/1999, p.94).

<sup>55</sup> Principal é o termo utilizado para designar usuários ou processos.



- (3) confidencialidade de dados é o processo de garantir que os conteúdos das mensagens sejam revelados somente aos principais autorizados. A principal técnica utilizada para prevenir perdas de confidencialidade é a criptografia;
- (4) não repúdio é o processo de autenticar o originador da mensagem. Um principal deve ser capaz de ser convencido da identidade do originador (*proof of origin*). Em acréscimo, um principal deve ser capaz de convencer uma terceira parte da origem da mensagem, de modo que não haja dúvida ou negação da origem. De modo oposto, não-repúdio de um receptor é o processo de provar a uma terceira parte que a mensagem foi verdadeiramente recebida por um identificável principal, o qual não pode duvidar ou negar que a mensagem foi recebida (*proof of delivery*). Estas provas estão associadas aos mecanismos de assinatura digital e certificado digital; e
- (5) administração segura é o processo de gerenciar o ambiente de segurança e garantir que a política de segurança seja cumprida e que a segurança não esteja sendo violada.

A relação entre riscos, ataques, serviços e mecanismos de segurança a ser analisada para a definição de uma política de segurança pode ser observada na Tabela 4.1 abaixo (MATOS, 1999).

TABELA 4.1 – ESTABELECIMENTO DE UMA POLÍTICA DE SEGURANÇA.

Riscos	Ataques	Serviços	Mecanismos
Destruição de informação	Vírus Roubo de senhas Dano físico	Controle de Acesso Confidencialidade	Criptografia Assinatura Digital
Modificação ou deturpação da informação	<i>IP Spoofing</i> Monitoração de roteador <i>Masquerading</i> Modificação <i>Replay</i> (intercepção e armazenamento)	Controle de Acesso Confidencialidade Integridade Autenticação Não-Repúdio	Criptografia Assinatura Digital Certificado Digital
Roubo ou perda de informação	Roubo de senhas Falsificação	Controle de Acesso Autenticação	Criptografia Assinatura Digital
Revelação de informação	<i>IP Spoofing</i> Monitoração de roteador	Controle de Acesso Confidencialidade Integridade Autenticação Não-Repúdio	Criptografia Assinatura Digital Certificado Digital
Interrupção de serviços	Sobrecarga deliberada do servidor	Integridade Não-Repúdio	Criptografia Assinatura Digital Certificado Digital <i>Time stamp</i>

Os serviços listados na Tabela 4.1, bem como alguns protocolos existentes que colaboram para a obtenção de uma gerência de segurança eficiente, são comentados a seguir.

Um plano eficaz de gerência de segurança deve englobar os elementos necessários para prover os seguintes serviços (DOWD & MCHENRY, 1998): 1) controle de acesso (usuários autorizados); 2) confidencialidade (informações permanecem privadas); 3) autenticação (originador da mensagem é quem diz ser); 4) não-repúdio (originador da mensagem não pode negar que a enviou); e 5) integridade (mensagem não foi modificada em trânsito). Porém, não necessariamente uma política de segurança deve prover o nível máximo de segurança; mas, sim, deve-se procurar o nível adequado de segurança para as necessidades de cada usuário e/ou aplicação de modo a também oferecer boa performance e custo adequado.

Existem **protocolos** que têm por objetivo impedir tipos específicos de ataques; e outros que visam dar suporte a mecanismos de segurança a fim de que usuários possam construir suas próprias soluções de segurança. Algumas ferramentas associadas a protocolos de segurança estão listados abaixo:

SSH (*Secure Shell*) – provê segurança através de criptografia antes mesmo da autenticação. Útil em aplicações tais como telnet e rlogin;

SSL (*Secure Socket Layer*) – apresenta mecanismos para autenticação de clientes baseados em certificados digitais, mas não oferece recursos para a implementação de assinaturas digitais;

IPSEC (*IP Security*) – novos serviços de autenticação e criptografia;

PGP (*Pretty Good Privacy*) – criptografia em *e-mails* onde a distribuição das chaves públicas é realizada pelos proprietários das mesmas;

PEM (*Privacy Enhanced Mail*) – criptografia em *e-mails* onde existe uma autoridade centralizada para a distribuição de chaves; e

SET (*Secure Eletronic Transaction*) – independente de plataforma e do protocolo de transporte utilizado foi concebido para transações comerciais na Internet (*Visa e Mastercard*) e implementa todos os cinco serviços de segurança (acesso, confidencialidade, autenticação, não-repúdio e integridade).

Tendo conhecimento dos serviços necessários e dos protocolos disponíveis, o desenvolvedor pode fazer uso da arquitetura CORBA e da linguagem Java para implementar uma política de segurança adequada para sua aplicação<sup>56</sup>. As técnicas e ferramentas associadas a CORBA e Java são apresentadas no próximo item 4.1.2.

---

<sup>56</sup> Newsletter of the IEEE Computer Society's TC on Security and Privacy, <<http://www.itd.nrl.navy.mil/ITD/5540/ieee/cipher/>>; CERT Coordination Center, <<http://www.cert.org/advisories/>>.

## 4.1.2 Tecnologias e Ferramentas na Gerência de Segurança

Este item 4.1.2 apresenta as principais tecnologias, bem como suas ferramentas, que colaboram para prover aplicações distribuídas seguras. No item 4.1.2.1 apresenta-se a arquitetura CORBA e no item 4.1.2.2 a linguagem Java. Enquanto CORBA preocupa-se com a transparência da rede, Java se preocupa com a transparência da implementação.

### 4.1.2.1 Arquitetura CORBA

A arquitetura CORBA (*Common Object Request Broker Architecture*) permite que aplicações se comuniquem umas com as outras com transparência de localização de objetos, linguagens, sistemas operacionais e redes. CORBA define as interfaces APIs (*Application Program Interfaces*) e a linguagem IDL (*Interface Definition Language*) que permitem a interação cliente-servidor com uma implementação de um ORB (*Object Request Broker*). O ORB é o elemento intermediário (*middleware*) que estabelece relações cliente-servidor entre os objetos. Usando um ORB, um cliente pode invocar de forma transparente um método em um objeto servidor, que pode estar em qualquer lugar da rede. O ORB é responsável por encontrar um objeto que possa implementar a requisição, passar os parâmetros, invocar o método e retornar os resultados. Assim, o ORB provê a interoperabilidade entre aplicações de sistemas distribuídos heterogêneos.

O Modelo de Referência de Segurança CORBA (*Security Reference Model – SRM*) inclui as seguintes funcionalidades:

- (1) auditoria – para contabilizar as ações relacionadas à segurança do sistema;
- (2) segurança de comunicação – para garantir a segurança da comunicação entre objetos;
- (3) não-repúdio – para comprovar a origem dos dados do originador;
- (4) administração da segurança – para aplicar as políticas de segurança;
- (5) identificação e autenticação – para verificar a identidade do usuário que pede um serviço; e
- (6) autorização e controle de acesso – para verificar quais serviços podem ser prestados ao usuário.

O SRM requer uma política adequada a fim de considerar não somente a segurança, mas também a portabilidade, a interoperabilidade, a performance e a independência tecnológica (OMG, 1998):

- (1) portabilidade – o objeto não precisa se preocupar com a segurança estabelecida no ambiente, podendo migrar para ambientes cujo nível de segurança é diferente;
- (2) interoperabilidade – é necessário manter a consistência de segurança mesmo entre sistemas heterogêneos, com diferentes ORBs. Muitos ORBs implementam esta interoperabilidade através do uso de *gateways*, que conectam ambientes distintos. Através do uso dos protocolos GIOP (*General Inter-ORB Protocol*) e IIOIP (*Internet Inter-ORB Protocol*), objetos podem ser executados e verificados em ambientes de alto grau de segurança<sup>57</sup>;
- (3) performance – é importante que a inclusão de serviços de segurança não degrade a performance do sistema; e
- (4) independência tecnológica – cliente e servidor devem utilizar os mesmos mecanismos de segurança, de forma a permitir o estabelecimento da comunicação.

O *Visibroker for Java*<sup>58</sup>, utilizado nesta implementação, é um ORB CORBA 2.0 que suporta o desenvolvimento e a gerência de aplicações de objetos distribuídos através de uma variedade de plataformas e de sistemas operacionais. Os objetos construídos com o *VisiBroker* são facilmente acessados por aplicações baseadas na Web que se comunicam usando o protocolo IIOIP – que é o padrão para a comunicação entre objetos distribuídos.

#### 4.1.2.2 Linguagem Java

A fim de prover segurança em aplicações distribuídas, a arquitetura de segurança Java oferece três níveis para a verificação de aplicações escritas em Java:

- (1) Verificador de *Bytecode* (VBC) – antes que seja executado, o resultado da compilação de uma *applet*<sup>59</sup> (*bytecode*) é verificado, de forma a garantir que ponteiros não foram forjados, restrições não tenham sido violadas e que não tenha ocorrido acesso a objetos proibidos;
- (2) Carregador de Classes de *Applets* (CCA) – tendo sido feita a verificação estática do *bytecode*, inicia-se a sua verificação dinâmica. Neste ponto, o CCA determina quando e

<sup>57</sup> A comunicação entre os objetos pode ser gerenciada através da utilização do protocolo IIOIP. No entanto, o protocolo IIOIP não é totalmente seguro. Para que comunicações entre objetos sejam mais seguras, utiliza-se SSL como forma de prover autenticação, confidencialidade, não-repúdio, integridade e controle de acesso.

<sup>58</sup> VISIGENIC. *Visibroker for Java Programers's Guide*. v. 3.3.

como uma *applet* pode adicionar classes a um ambiente que esteja executando Java. Para isso, o CCA divide e associa essas classes com *Espaços de Nomes*, de modo a garantir que *applets* maliciosas não tentem modificar conteúdos do ambiente em execução; e

(3) Gerente de Segurança – é o módulo responsável por verificações em tempo de execução cujas funções são: prevenir da instalação de novos CCAs, proteger *threads* de interferências mútuas, controlar operações de *read/write*, controlar operações de *socket* (*connect* e *accept*) e controlar o acesso a pacotes Java.

Dentro do contexto dos três níveis identificados acima, a MVJ (Máquina Virtual Java) que está incluída na maioria dos navegadores (*browsers*) possui grande importância. As verificações feitas pela MVJ relacionadas à garantia da segurança são (MATOS, 1999): (1) verificação de referências a objetos que utilizam classes diferentes; (2) acesso estruturado à memória (i.e., a não-utilização de ponteiros); (3) liberação automática de memória (*garbage collection*); (4) verificação de limites de vetores; e (5) verificação de referências nulas. Tais verificações efetuadas pela MVJ impedem a possibilidade de ataques cuja intenção é converter a referência a um objeto, ou seja, burlar a segurança de forma que um determinado objeto execute operações que não tenham sido definidas para o mesmo objeto. Porém, mesmo com tais níveis de verificação, ataques através de *applets* maliciosas ainda são possíveis. Uma característica do novo modelo de segurança Java é a implementação de Domínios de Proteção, onde um arquivo-texto contém as restrições de acesso a determinadas informações. A adoção de um domínio pode ser realizada de duas maneiras: (1) através da ferramenta *policytool*; e (2) através da API `java.security.Permission` ou qualquer classe `Permission`, p. ex., `java.io.Permission`.

Java<sup>60</sup> e CORBA (CHEN et al, 1997, ORFALI & HARKEY, 1997) são meios importantes para a obtenção de aplicações distribuídas e dinâmicas, principalmente em contexto Web (HAUW et al, 1997). Os serviços de portabilidade de chaves criptográficas assinatura de objetos, criptografia de mensagens e recursos SSL já podem ser encontrados – e configurados – nas atuais versões de navegadores.

<sup>59</sup> *Applet* é um programa Java localizado em um servidor, que ao ser acessado é transmitido e executado no cliente.

<sup>60</sup> A tendência atual associada à Java é a rede que interliga qualquer eletrodoméstico – aliás, a idéia inicial da linguagem Java, quando esta surgiu. A *Sun Microsystems* aposta na linguagem *Jini*, baseada em Java, para interligar os aparelhos eletrodomésticos sem a necessidade de PCs (um novo refrigerador *Frigidaire*, permite manter a lista de compras atualizada através de um código de barras que lê os produtos que estão para acabar e viram pedidos a supermercados via Internet). Isso significa que a garantia de segurança em aplicações distribuídas é cada vez mais uma exigência atual no cotidiano das pessoas.

### 4.1.3 Rede Telefônica

O Brasil entrou para a era das telecomunicações tão logo esta foi inaugurada no mundo. As primeiras linhas telegráficas foram instaladas em 1852, e a telefonia passou a ter mais expressão em 1877. Em 1962, foi promulgada a Lei 4.117, que instituiu o Código Brasileiro de Telecomunicações e definiu o Sistema Nacional de Telecomunicações. Em 1963, foi aprovado o Plano Nacional de Telecomunicações e o Regulamento Geral, definindo a Empresa Brasileira de Telecomunicações (Embratel) como a executora do Sistema Básico de Telecomunicações. Em 1967, foi introduzido o DDD no país e em cinco anos todas as capitais e principais cidades estavam interligadas – e simultaneamente, foi inaugurado o serviço internacional.

Quanto à infra-estrutura industrial de serviços e fabricação, o Brasil está apto a outro salto de categoria, como o que foi dado na década de 60. A história das telecomunicações brasileiras inclui fatos de pioneirismo de desbravação, como as instalações de linhas telegráficas pelo Marechal Cândido Rondon, patrono das telecomunicações brasileiras, bem como fatos de experiências técnico-científicas inusitadas, como as do padre Landau de Moura (BARRADAS, 1995).

A telefonia celular no Brasil (iniciada em 1991) opera na frequência de 800 MHz em duas bandas, A e B. A Anatel (Agência Nacional de Telecomunicações) está decidindo sobre qual será a frequência da Banda C, em breve disponível no Brasil. São duas opções:

- (1) frequência 1900 MHz, com o sistema PCS (Sistema de Comunicações Pessoais – *Personal Communications System*) utilizado nos Estados Unidos, Canadá e Argentina, por exemplo, que possibilita a escolha dentre todas as tecnologias (CDMA, TDMA e GSM) e é compatível com as bandas A e B; e
- (2) frequência 1800 MHz, com o sistema GSM Móveis (Sistema Global para Comunicações – *Global System for Mobile Communications*) utilizado na Europa.

A estrutura atual de uma rede de telecomunicações convencional compreende diversas centrais telefônicas distribuídas pela área de abrangência de uma empresa de telecomunicações. Essas centrais são classificadas e dispostas segundo o tipo de ligações que podem registrar e também conforme o número de centrais que a região a que ela pertence necessita (SOUZA & LEITE, 1999).

**Centrais Locais:** São centrais telefônicas nas quais se ligam linhas de assinantes. A central local tem um terminal para cada assinante em um raio típico de até 6 km e possui juntores para ligação com outras centrais. Possuem prefixo indicativo que também compõe o número do assinante. Estas centrais registram apenas impulsos, ou seja, ligações que são efetuadas para um número de telefone pertencente à mesma cidade. As ligações registradas por este tipo de central pertencem ao degrau de tarifação local (DL).

**Centrais Tandem:** Uma central tandem interliga centrais através de juntores e não possui terminais de assinantes, isto é, não liga linha de assinantes. Os dispositivos comuns são destinados exclusivamente ao encaminhamento de chamadas. Uma central “tandem” pode ser sinônimo de central trânsito quanto ao aspecto de interligar centrais de comutação entre si. Entretanto, estes dois termos podem ser aplicados de maneira diferenciada no que se refere ao encaminhamento das chamadas. Também é utilizado o termo “central tandem local” para se referir a uma central trânsito que tem por função interligar centrais locais. Uma central IU (interurbana) é a central destinada essencialmente a distribuir as chamadas IU terminadas em uma área local.

**Centrais Trânsito:** A central trânsito ou IU comuta chamadas originadas em centrais locais ou provenientes de centrais tandem. A trânsito permite a conexão de centrais por meio físico ou através do espaço livre e também não possui terminais de assinante. Sua principal função é interligar outras centrais de comutação entre si. A central trânsito interurbana é a central trânsito usada no encaminhamento de chamadas IU. Estas centrais registram apenas ligações que não são locais, ou seja, registram todas as chamadas telefônicas completadas e realizadas por uma central. A base de dados formada pelos registros de chamadas telefônicas, em inglês *Call Detail Record* (CDR), traduzidos para o português como “bilhetes de tarifação”, contém diversos campos de informação sobre o evento de utilização do serviço de telefonia, entre eles a identificação do assinante que origina a chamada e o assinante-destino. As ligações registradas por este tipo de central podem pertencer a degraus de tarifação distintos conforme listados na Tabela 4.2.

TABELA 4.2 – EXEMPLO DE TIPOS DE LIGAÇÕES POSSÍVEIS PARA A CIDADE DE FLORIANÓPOLIS<sup>61</sup>

Degráus de Tarifação	Tipos de Ligações
DL	prefixoLocal + _____ (telefones pertencentes a Florianópolis)
D1	prefixoInterurbano + _____ (telefones com DDD 048)
D2	04x + _____ - _____ (onde x é diferente de 8)
D3	0y_ + _____ - _____ (onde y é diferente de 4 e 0)
DI	00 + códigoPaís + códigoCidade + númeroTelefone

O Degrau D1 corresponde a ligações não locais, mas destinadas a localidades que possuem o mesmo código de área; o Degrau D2 corresponde a ligações não locais, para localidades que possuem código de área onde apenas o último dígito o diferencia do código de origem da ligação; o Degrau D3 corresponde a ligações não locais, destinadas a lugares que possuem código de área diferente dos códigos pertencentes aos degraus de tarifação D1 e D2; e por fim o Degrau DI corresponde a ligações internacionais.

**Servidores Regionais:** São computadores que concentram lotes de CDRs de centrais telefônicas que pertencem à área de abrangência destes servidores. Tais servidores são denominados de forma diferente dependendo da empresa de telecomunicações a que pertencem. Cada lote de CDRs de uma central é enviado ao seu respectivo Servidor Regional. Este envio é efetuado em intervalos de tempo ou após o acúmulo de uma determinada quantidade de ligações, dependendo da configuração da central.

A Figura 4.3 mostra o encaminhamento dos diversos tipos de ligações que podem ser efetuadas por um usuário de uma companhia telefônica. Neste caso, tomou-se como exemplo um telefone da localidade de Florianópolis (233-5983, DDD 48).

<sup>61</sup> Este trabalho não contempla o novo sistema DDD, pois o mesmo foi adotado quando este trabalho já estava em fase adiantada de desenvolvimento. Tais alterações estão assinaladas para trabalho futuros.



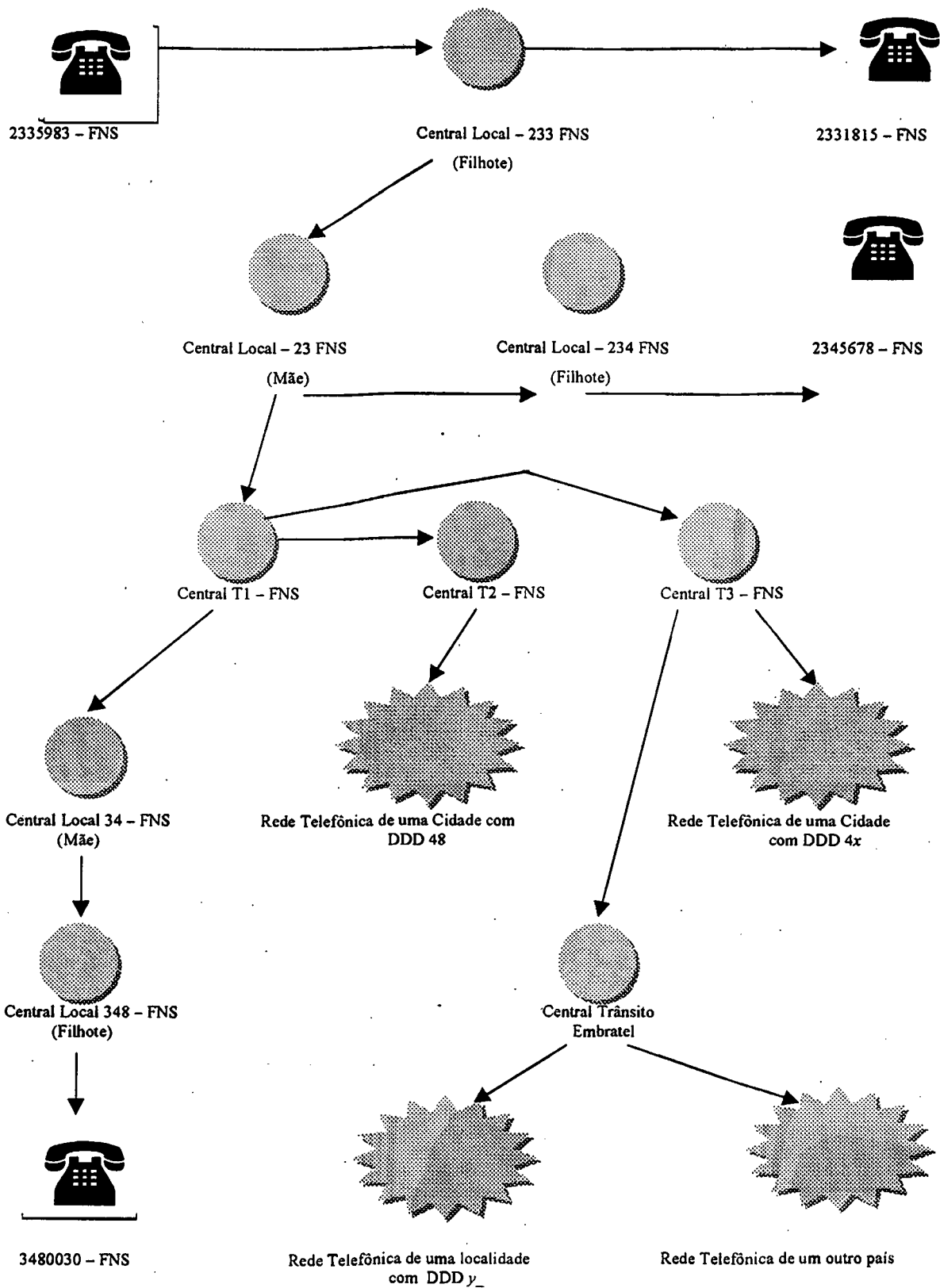


FIGURA 4.3 - EXEMPLO DE UM USUÁRIO EFETUANDO DIVERSOS TIPOS DE LIGAÇÕES.

Todas as ligações realizadas pelo telefone exemplificado, independentemente do tipo, passam pela central local à qual o usuário pertence (neste caso, a central “filhote” 233). O que acontece, na verdade, é que toda ligação efetuada por um usuário que chega à sua **central local** é avaliada por esta e, dependendo do número que está sendo discado, a central pode ou não encaminhar a ligação para uma outra central, dependendo das seguintes situações: 1) no caso de o número discado possuir um prefixo igual ao prefixo da central “filhote” do telefone de onde a ligação está sendo originada, a central “filhote” encaminhará a ligação diretamente para o telefone-destino, pois ambos pertencem à mesma central; e 2) quando a ligação é destinada a um número de telefone com prefixo diferente, a ligação é encaminhada a sua central “mãe” (no exemplo, a central “mãe” 23).

Na central “mãe”, a ligação é avaliada novamente. De acordo com o número discado, a ligação pode ter outros 2 destinos: 1) se a ligação possuir os dois primeiros dígitos iguais aos do código da central “mãe” onde ela está (neste caso, 23), a ligação é encaminhada diretamente para a central “filhote” pertencente ao telefone destino e esta se encarrega de concretizar a ligação; e 2) caso o número chamado não possua o mesmo código da central mãe, a ligação é encaminhada para a central T1 de Florianópolis, conforme o exemplo.

A central T1, por sua vez, avalia o prefixo do telefone chamado. De acordo com este prefixo, a ligação pode ter outros 3 destinos: 1) uma nova central mãe (veja central 34 no exemplo), caso o prefixo da ligação seja de um telefone da mesma cidade; 2) a central T2 de Florianópolis, caso a ligação seja destinada a um telefone com mesmo código de área, mas fora da cidade; e 3) a central T3 de Florianópolis, caso o número discado pertença a um outro código de área.

Na central T2, a ligação é encaminhada a várias outras centrais que compõem o caminho até o destino da ligação, ou seja, ela é encaminhada para a rede telefônica da cidade-destino da ligação.

Na central T3, a ligação pode ser encaminhada a dois destinos distintos: 1) se o código de área do número destino da ligação diferir do código origem da ligação apenas pelo último dígito, a ligação será encaminhada à rede telefônica da região que possuir aquele código DDD; e 2) mas se a ligação possuir apenas o primeiro dígito igual ao primeiro dígito do código de área do número de telefone que originou a ligação, a chamada é encaminhada para a Central Trânsito Embratel.

Na Central Trânsito Embratel, a ligação pode ter dois destinos: 1) uma companhia telefônica de um outro país, no caso de ligação internacional; e 2) a companhia telefônica responsável pela região onde o destino da ligação se localiza.

O sistema de tarifação está baseado na programação das centrais telefônicas. Quando o usuário retira o fone do gancho e começa a teclar um número, os equipamentos interpretam os algarismos iniciais como código de encaminhamento. Essa configuração numérica identifica o tipo de ligação, e quando a chamada se completa começa o cálculo do tempo de duração, considerando-se dia e hora. O cálculo do valor de ligações é efetuado conforme tabelas de tarifas vigentes na empresa de telecomunicações. As tabelas são divididas segundo os tipos de ligações (destino da ligação/divisão em degraus). Cada degrau também tem sua tarifa variada de acordo com os horários e dias da semana nos quais as ligações poderão ser efetuadas, conforme apresentado nas Tabelas 4.3 e 4.4 abaixo.

DESTINO E GRUPOS DE TARIFICAÇÃO	MINUTO INICIAL		CADA 6 SEGUNDOS ADICIONAIS		HORÁRIOS DE TARIFICAÇÃO REDUZIDA (1)
	Normal	Reduzido	Normal	Reduzido	
Mercosul (Argentina, Chile, Paraguai e Uruguai)	1,72234	1,37787	0,14477	0,11382	
EUA (Inclusive Havai)	1,33869	1,07167	0,09796	0,07832	
Canadá e demais países das Américas e Antilhas	1,72234	1,37787	0,16197	0,12932	00:00 às 05:00
Portugal, Alemanha, Áustria, Suíça, Holanda, Espanha, Bélgica, Dinamarca, Irlanda	1,72234	1,37787	0,17222	0,13778	20:00 às 24:00
Finlândia, França, Grécia, Itália, Lichtenstein, Noruega, Reino Unido, Suécia e Suíça	1,77203	1,42003	0,17222	0,13778	
Demais países da Europa e Oriente Médio	2,21879	1,77503	0,19137	0,15309	
Austrália e Japão	2,49892	1,99913	0,20032	0,16041	01:00 às 06:00 e 13:00 às 17:00
Demais países da África	3,68875	2,95100	0,29953	0,23963	01:00 às 05:00 e 21:00 às 24:00
Demais países da Ásia, Oceania e Ilhas do Pacífico	1,66815	1,33400	0,29953	0,23963	01:00 às 06:00 e 13:00 às 17:00

TARIFA - R\$

(1) - Domingo - 00:00 às 24:00  
De segunda a sábado - nos horários acima

TABELA 4.3 – DEGRAUS DE TARIFAS INTERNACIONAIS.

(valores em 1 pulso a cada x segundos)

FAIXA DE HORÁRIOS	DIAS ÚTEIS				SÁBADOS				DOMINGOS E FERIADOS NAC.			
	DL	D1	D2	D3	DL	D1	D2	D3	DL	D1	D2	D3
00:00 às 06:00	PPC				PPC							
06:00 às 07:00		128,9	77,3	51,6		128,9	77,3	51,6				
07:00 às 09:00		64,4	38,7	25,8		64,4	38,7	25,8				
09:00 às 12:00		32,2	19,3	12,9		32,2	19,3	12,9				
12:00 às 14:00	240	64,4	38,7	25,8		64,4	38,7	25,8				
14:00 às 18:00		32,2	19,3	12,9		32,2	19,3	12,9				
18:00 às 21:00		64,4	38,7	25,8	PPC	128,9	77,3	51,6				
21:00 às 24:00		128,9	77,3	51,6		128,9	77,3	51,6				

Super Reduzida (75% da Normal)   
 Reduzida (50% da Normal)  
 Normal   
 Atréscida (100% da Normal)

\* PPC - Cobrança de 1(um) impulso por ligação.

\* DL - Ligações Locais.  
\* D1 - Ligações não DL para mesmo código de área.  
\* D2 - Ligações não DL, não D1, para código de área 04X.  
\* D3 - Ligações não DL, não D1, não D2, para código de área 0XY.

\* Valor do impulso - R\$ 0,08

TABELA 4.4 - DEGRAUS DE TARIFAS NACIONAIS.

A Tabela 4.3 apresenta as tarifas internacionais, enquanto a Tabela 4.4 apresenta as tarifas nacionais vigentes no Brasil em 1999.

As ligações locais são calculadas conforme a leitura do número de pulsos que são registrados em um contador que está ligado diretamente à linha do assinante, através de sua respectiva central local. Por sua vez, este contador é controlado por um relógio que, dependendo do horário de tarifação no qual a ligação está sendo realizada, faz com que o contador seja incrementado de um pulso a cada “x” segundos. Diferentemente das ligações locais, o valor de ligações não locais é calculado utilizando-se os dados referentes à duração, à data e ao horário em que a ligação foi efetuada, juntamente com os valores da tabela de tarifas da empresa. As ligações interurbanas e internacionais são registradas pela Embratel. Todas as ligações não locais são aferidas num processo chamado de “bilhetado”, no qual existe discriminação de destino, data, hora de realização, e tempo de duração da chamada. Mensalmente, é feita a totalização de pulsos e minutos utilizados, para lançamento das contas telefônicas.

## **4.2 Implementação dos Sistemas SSCC e SIPI**

O item 4.2.1 apresenta a implementação em Java com suporte CORBA dos módulos Adaptador e Agente, que são distribuídos por toda a área de cobertura de determinada companhia telefônica (HERMIDA & VALE, 1999, MILIOLI & CASTELLO, 1999). Esses módulos estão associados a cada central telefônica da companhia. O item 4.2.2 apresenta a implementação Java com suporte CORBA do módulo Gerente, responsável por receber notificações de possíveis fraudes dos diversos agentes distribuídos e enviar alarmes aos usuários da companhia (SPÍNDOLA, 1999)<sup>62</sup>. O item 4.3.3 apresenta a implementação da segurança distribuída com suporte CORBA.

### **4.2.1 Módulos Adaptador e Agente**

Considerando que as ligações telefônicas (armazenadas em CDRs - *Call Detailed Register*) ficam distribuídas na rede telefônica e que existe um processo Agente junto a cada CDR, os processos Agentes também se encontram distribuídos. Para realizar a comunicação entre os agentes distribuídos e o Gerente (que é um módulo centralizado) emprega-se CORBA como tecnologia de objetos distribuídos (PAVLOU, 1997, DOGAC et al, 1998).

---

<sup>62</sup> Os módulos Adaptador, Agente e Gerente estão apresentados graficamente na Figura 1.3.

A arquitetura de distribuição CORBA permite que o Agente envie notificações ao Gerente de forma simples e transparente para os administradores do sistema da operadora telefônica. É necessário apenas que existam ORBs (*Object Request Broker*) nos servidores onde o módulo Gerente e os módulos Agentes estão instalados.

O SSCC utiliza-se de duas bases de dados: 1) uma, denominada CDR Filtrado, que serve de ligação entre o Adaptador (que lê os CDRs antes de enviar ao Agente) e o Agente; e 2) a outra, denominada Base de Consulta, onde o Agente consulta o padrão previamente calculado para cada usuário. O uso de uma base de dados para tal finalidade justifica-se pelo simples fato de o Agente necessitar dos dados de um certo número de ligações vindas do CDR. Caso fosse necessário fazer uma consulta diretamente no CDR, esta deveria ser sequencial e levaria muito tempo, devido ao tamanho do arquivo.

Os campos que compõem o CDR filtrado são: `telUsuario` – o número do telefone que originou a chamada; `telChamado` – o número do telefone chamado; `duracao` – o tempo de duração da chamada; `dataChamada` – a data em que ocorreu a chamada; e `horaChamada` – a hora em que se iniciou a chamada. Os campos que compõem a Base de Consulta são: `telUsuario` – o número do telefone de um usuário; e `classeUsuario` – o padrão de comportamento do usuário.

O módulo Adaptador é responsável por esperar o recebimento de novas ligações que chegam a todo instante em uma central telefônica. Para cumprir essa tarefa o mesmo deve analisar os registros de chamada da companhia telefônica. Esses registros contêm informações detalhadas sobre cada uma das chamadas efetuadas e, por isso, chamados de CDR (*Call Detailed Register*). Em cada central telefônica existe um arquivo que contém dados armazenados sequencialmente em formato ASCII. Este arquivo é o que se denomina de CDR. CDRs são normalmente arquivos muito grandes e que recebem atualizações muito rápidas, pois para cada ligação efetuada um novo registro (linha) é inserido nos mesmos. CDRs são utilizados para a contabilização da conta telefônica dos clientes da operadora. No final de cada mês esses arquivos são coletados, enviados a uma central de processamento e então são geradas as contas telefônicas a serem pagas. Neste trabalho, utiliza-se o CDR em formato *Ericsson*. No entanto, para o uso de outros formatos de CRDs, a mudança no código do programa é mínima, apenas modificando-se a posição dentro do arquivo onde os dados seriam coletados.

A funcionalidade básica do Adaptador é: 1) ler e monitorar o CDR; 2) filtrar os dados lidos e armazená-los em um banco de dados; 3) verificar se um dado cliente atingiu um certo número de ligações; e 4) avisar ao Agente quando for atingido o número de ligações especificado. O motivo de existir um módulo que serve somente para ler e filtrar as informações de um arquivo é que esse arquivo é muito variável de companhia telefônica para companhia telefônica. Assim, não é necessário modificar o Agente quando o CDR for diferente do originalmente usado, bastando alterar o Adaptador, que é um módulo muito mais simples que o Agente. Dessa forma, o Adaptador lê o CDR e disponibiliza as informações ao Agente através de uma Base de Dados. O conjunto Agente/Adaptador deve estar instalado em cada uma das centrais telefônicas onde houver um CDR. O tipo de chamada (local, interurbana regional, interurbana nacional, internacional) é determinado pela localização da estação onde se encontra o CDR – por exemplo, uma ligação internacional fica em uma central da Embratel.

Quando inicializado o sistema, o módulo Adaptador lê o CDR de cima para baixo, do início até quando encontrar a última linha. Durante esse processo de leitura, o Adaptador verifica, a cada inserção no banco de dados, se o usuário que originou a ligação corrente já atingiu o número especificado de ligações. Esse número é especificado pelo administrador do sistema e esse valor regula a sensibilidade do sistema. São estas  $n$  ligações<sup>63</sup> que são enviadas para a rede neural para que a mesma determine o padrão de determinado usuário. A sensibilidade do sistema é afetada a medida que esse valor for modificado. Existe um intervalo ótimo onde a precisão do sistema é maior. Caso esse valor seja muito maior ou muito menor que esse intervalo ótimo, a precisão será reduzida. Atingido o total de ligações para o usuário corrente, o Adaptador emite um aviso ao Agente indicando que ocorreu essa condição e envia o número do telefone do usuário. Encontrando o final do arquivo, o Adaptador entra em estado de espera por um período predeterminado. Passado este tempo o Adaptador fica novamente ativo e verifica se houve mudança no CDR. Caso tenha ocorrido alguma modificação (inserção de registros), o Adaptador lê estes novos dados, e a cada inserção no banco de dados verifica, como feito anteriormente, se o número especificado de ligações para este usuário foi alcançado. Desse ponto em diante o sistema continuará funcionando dessa maneira, alternando períodos de espera e de leitura.

O módulo Adaptador foi implementado usando-se o conceito de linhas de código (*threads*), para maior eficiência e menor consumo de processamento. Dessa maneira conseguiu-se sanar

---

<sup>63</sup> Nos testes realizados, o melhor desempenho foi conseguido com  $n=10$ . O objetivo é controlar a 'quebra' excessiva de padrão ou então o agrupamento excessivo de padrões. E isso é particular a cada central telefônica (i.e., agente).

um problema encontrado anteriormente, que era o da concorrência de acesso ao CDR. O sistema consegue manter o arquivo aberto para leitura permanentemente e mesmo assim o CDR pode ser atualizado sem nenhum problema. Por se tratar de um módulo que atua independentemente do usuário (i.e., em *background*), o Adaptador não possui interface com o usuário.

O módulo Agente é responsável, juntamente com a rede neural, pela detecção da possível fraude de clonagem propriamente dita. É ele quem envia para a rede neural o conjunto contendo o número especificado de chamadas lidas pelo Adaptador e é ele também quem recebe o resultado do processamento da rede neural. De posse do resultado emitido pela rede neural, verifica se o padrão calculado está de acordo com o padrão previamente definido para o usuário. Se o padrão estiver de acordo, ele entra em estado de dormência até que o Adaptador o invoque novamente. Caso contrário, envia um alerta para o Gerente.

A funcionalidade básica do Agente é: 1) ler o número determinado de ligações do banco de dados e excluir os *n* registros lidos; 2) preparar os dados a serem enviados para a rede neural; 3) receber o padrão calculado pela rede neural; 4) comparar o padrão calculado com o padrão previamente determinado para o usuário; e 5) enviar o alarme ao Gerente caso constate possíveis fraudes.

De posse desse número de telefone, o Agente faz uma pesquisa na base de dados e recupera, em uma estrutura do tipo *ResultSet* (estrutura da linguagem Java que armazena os dados vindos de uma consulta SQL), essas *n* ligações. Esses dados retornados da consulta são armazenados em um arquivo formato ASCII, para que a rede neural possa recuperá-los e processá-los. Depois de recuperadas e salvas, essas informações são excluídas da base de dados. A exclusão se justifica por dois motivos: em primeiro lugar para evitar que a base de dados cresça indefinidamente, e em segundo lugar porque não teria sentido que as informações referentes às ligações fossem guardadas depois de processadas. O módulo Gerente (veja item 4.2.2), ao contrário, armazena os dados inclusive para fins de auditoria.

O próximo passo é a invocação da rede neural pelo módulo Agente, para que a mesma processe os dados salvos<sup>64</sup>. Para cada *n* ligações de um usuário que são submetidas à rede neural, esta devolve como saída uma seqüência de dez dígitos, dos quais apenas um será igual a 1. Todos os demais serão iguais a 0. Este valor 1 indica o padrão do usuário. Veja a Figura 4.4.

---

<sup>64</sup> Atualmente a interação com a rede neural é realizada através de arquivos ASCII. Pretende-se incorporar os procedimentos da rede neural, hoje em MatLab, dentro do código Java do Agente com o auxílio do MCC (*MatLab Compiler Command*).

Número de padrões:	1	2	3	4	5	6	7	8	9	10
Saída da rede neural:	0	0	0	0	1	0	0	0	0	0

-----Usuário de padrão 5.

FIGURA 4.4 – VETOR DE SAÍDA DA REDE NEURAL IDENTIFICANDO O PADRÃO DO USUÁRIO.

Pelo fato de o Agente estar sempre em estado de dormência, optou-se por criar um pequeno módulo (que por sua vez também é uma *thread*) que tem a finalidade única e exclusiva de verificar se o arquivo de saída da rede neural foi modificado. Acontecendo a modificação, esse módulo, denominado *Verifier*, avisa o Agente de que essa condição ocorreu. Quando o Agente perceber o chamado do módulo *Verifier* avisando que o arquivo de saída da rede neural foi modificado, ele, o Agente, irá ler o padrão que a RN calculou e inseriu nesse arquivo e compará-lo com o padrão previamente calculado e armazenado na Base de Consulta.

Nesse ponto, o Agente pode agir de duas formas. Se o padrão enviado pela rede neural for igual ao padrão previamente calculado, o Agente simplesmente ignora o usuário, considerando que não houve fraude de clonagem. Entretanto, se o padrão enviado pela RN não coincidir com o padrão previamente calculado, o Agente presume que existe a possibilidade de uma fraude de clonagem e envia uma mensagem de alarme para o módulo Gerente, contendo o número do telefone sob suspeita, o padrão real do usuário e o padrão recém-calculado pela rede neural.

O Agente tem ainda como função gerar um *log* contendo todas as suspeitas de clonagem que foram enviadas para o Gerente. De maneira análoga ao Adaptador, o Agente também foi implementado usando-se o conceito de linhas de código (*threads*). Ele fica em estado de dormência até que o Adaptador o invoque. Isso evita consumo de capacidade de processamento, pois o Agente só entrará em atividade quando realmente for necessário. Por ser uma *thread*, ele também consegue evitar o problema de concorrência com o arquivo de saída da rede neural, da mesma forma como ocorreu com o CDR.

Por se tratar do módulo principal do sistema, esse módulo possui uma interface com o usuário usando os componentes do *Java Foundation Classes*. A partir dessa interface, o operador do sistema poderá iniciar ou encerrar todas as atividades aqui descritas. É possível ainda configurar alguns parâmetros referentes ao sistema. Veja na Figura 4.9 algumas telas do módulo Agente.



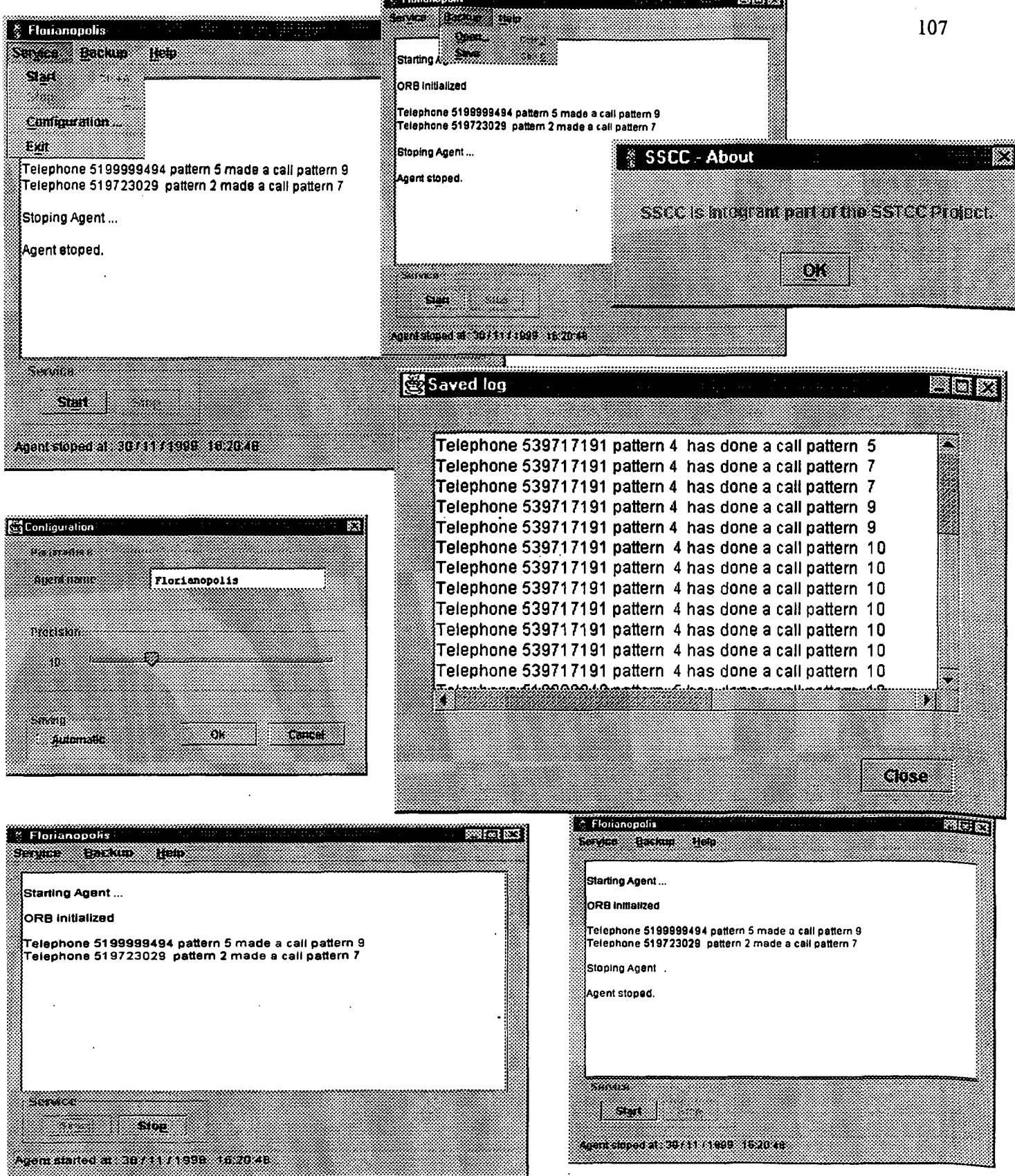


FIGURA 4.5 – TELAS DO AGENTE.

A Figura 4.5 apresenta algumas telas onde é possível observar o módulo Agente monitorando e detectando possíveis fraudes.

### 4.2.2 Módulo Gerente

A função do gerente é receber notificações dos diversos módulos Agentes e avisar o proprietário da linha sobre uma possível fraude nesta. O cliente é avisado de duas formas: 1) através de uma mensagem automática e imediata via o próprio telefone; e 2) através de uma mensagem impressa enviada pelo correio para o endereço que consta no cadastro. Em ambos os casos o aviso tem como objetivo confirmar ou não a realização das chamadas pelo usuário. Cabe então à empresa tomar as devidas providências caso haja a confirmação da fraude por parte do cliente. Em acréscimo, o gerente disponibiliza informações de similaridade de padrões das atuais chamadas com inadimplentes anteriores.

A Figura 4.6 mostra a janela inicial do módulo Gerente, com os alarmes que chegam dos agentes distribuídos.

arrive hour	User Info	Send	Send	Modify	Modify
	phone number	mail alarm	phone alarm	user resp	pattern update
12:30	9836012	18/11/99	18/11/99	22/11/99	22/11/99
13:12	9854234	19/11/99	19/11/99	21/11/99	21/11/99

FIGURA 4.6— JANELA PRINCIPAL.

Nos campos **mail alarm** e **phone alarm**, é mostrada a data em que o avisos foram mandados ao cliente (se em branco, então os alarmes não forma enviados). O campo **user resp** mostra a data em que o usuário forneceu a resposta, e em **pattern update** é apresentada a data em que a base de dados foi atualizada.

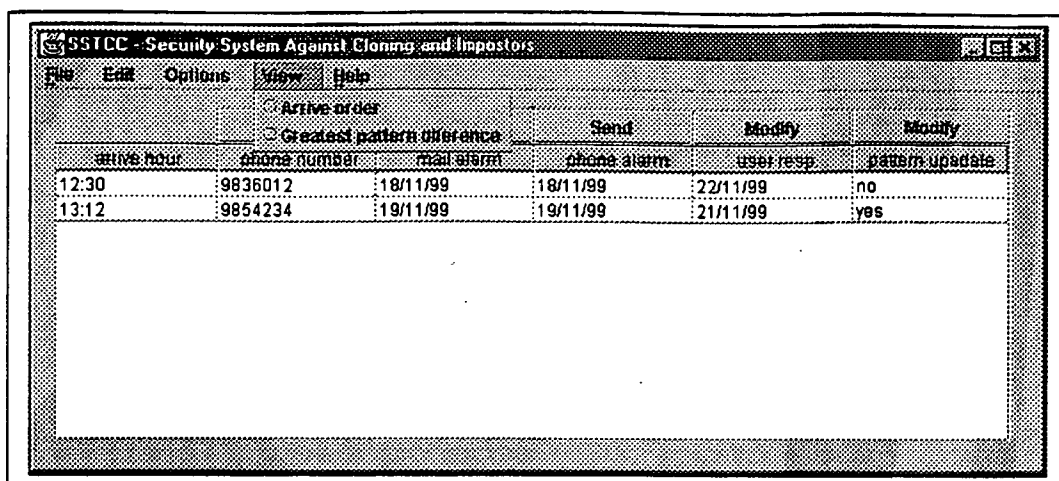


FIGURA 4.7 – OPÇÃO VIEW MENU.

Conforme pode ser observado na Figura 4.7, a opção View permite que as notificações sejam visualizadas não só por ordem de chegada, mas também por maiores diferenças detectadas (esta opção é permitida em arquivos de notificações já salvos anteriormente).

Na Figura 4.8, observam-se as opções para o envio dos alarmes aos clientes (i.e., por email, por telefone, por ambos)<sup>65</sup>.

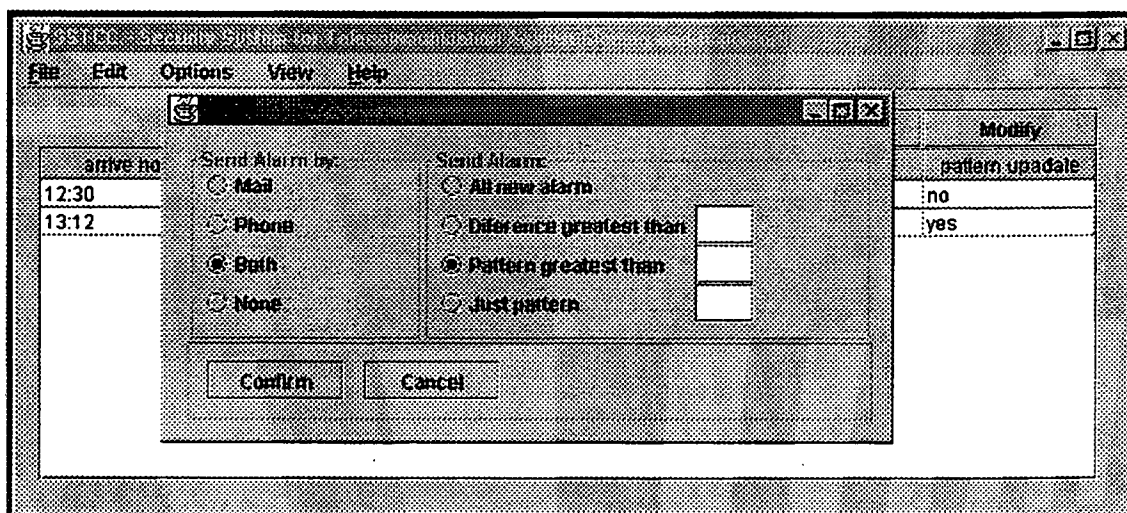


FIGURA 4.8 – JANELA CONFIG.

Em acréscimo, como é possível observar na Figura 4.8, os alarmes automáticos podem ser configurados (por cada companhia, de acordo com cada época). Ou seja, as notificações vindas do módulo Agente podem só ser enviadas automaticamente aos usuários quando a diferença de padrões (atual e novo) for maior que x; quando o padrão for maior que x; ou então somente quando o padrão for igual a x.

<sup>65</sup> Os alarmes poderão ser também enviados via fax e e-mail.

A Figura 4.9 apresenta uma janela de um arquivo de notificações recebidas aberto. Nela há uma série de informações (como o da janela principal), e além disso há mais algumas informações como as contidas na janela de **Config**, onde é possível contactar os clientes que ainda não foram contactados.

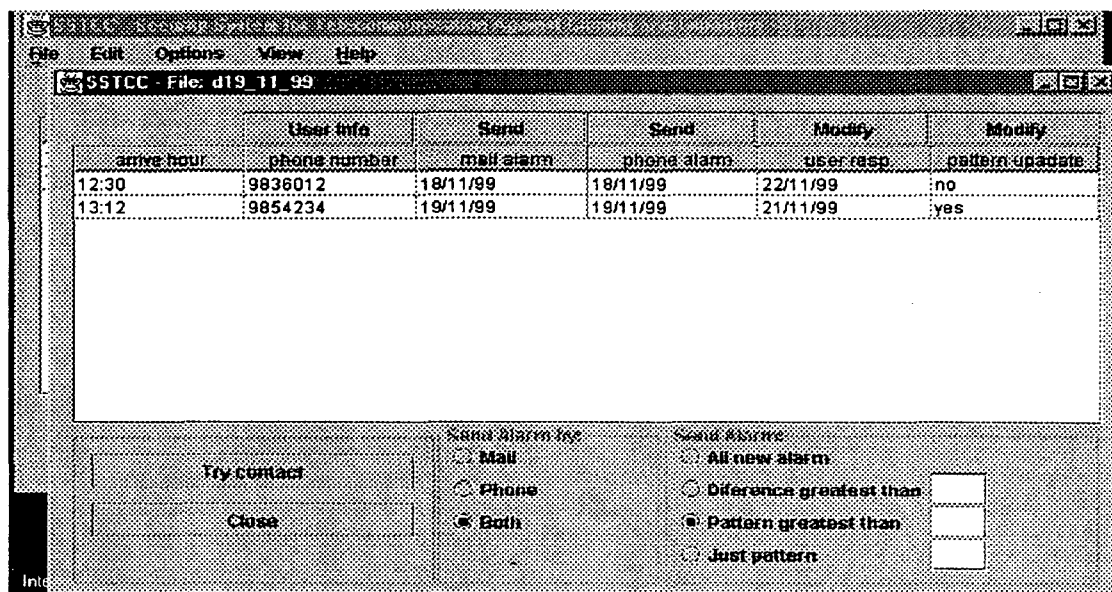


FIGURA 4.9 – ARQUIVO DE CHAMADAS ABERTO.

Esta opção apresentada na Figura 4.9 possibilita filtrar os alarmes a serem enviados aos clientes, conforme as necessidades da companhia.

#### 4.2.3 Implementação da Gerência em Ambiente Distribuído CORBA

Além da segurança prevista na comunicação com suporte CORBA (OMG, 1998) da implementação da segurança oferecida pelas ferramentas disponíveis no JDK, o SSTCC provê ainda um esquema específico de autenticação e autorização via ORB (DÉSIRÉ, 1999). Esse mecanismo de autenticação baseia-se em um pseudo-objeto do CORBA e os interceptores do *Visibroker*. Esse mecanismo consiste em que o cliente (Agente), após fornecer sua identificação (*ID/login*) de usuário e sua senha para entrar em uma sessão de aplicação, obtenha um identificador de sessão, ou *ticket*. Esse *ticket* contém a identificação e a senha do usuário em uma forma criptografada, que juntamente com a requisição do cliente é enviado pelo ORB cada vez que o cliente faz uma invocação em um objeto. Quando a requisição chega ao servidor (Gerente), ela é interceptada por um interceptor e o *ticket* é checado para averiguar se o cliente tem a permissão.

Em acréscimo, é de grande importância<sup>66</sup> garantir que as notificações enviadas pelos Agentes cheguem ao Gerente. Para garantir a disponibilidade (i.e., a não-interrupção) das notificações, o sistema utiliza o mecanismo de selo de tempo (*time-stamp*). Esse mecanismo permite não só observar se as notificações estão realmente chegando ao Gerente, mas também, se estão chegando dentro do intervalo de tempo previsto.

### 4.3 Implementação do Sistema SETWeb

O item 4.3.1 apresenta a implementação do Sistema SETWeb em Java, o qual permite que os usuários das empresas de telecomunicações verifiquem seus extratos telefônicos atualizados *on-line* durante todos os dias do mês. Isso possibilita a imediata detecção – pelos próprios usuários – de ligações indevidas que possam ocorrer por causa de fraudes de clonagem de celulares<sup>67</sup>. O sistema SETWeb é composto por módulos Adaptadores distribuídos (item 4.3.1) por todos os Servidores Regionais. Tais adaptadores são responsáveis por ler cada chamada registrada e disponibilizá-las em bancos de dados para que o Módulo Montador de Conta (item 4.3.2) as requisite. Após uma prévia validação da identificação do usuário através da *applet*, este mostra a página gerada pelo Módulo Montador (SOUZA & LEITE, 1999).

#### 4.3.1 Módulos Adaptadores

Estes módulos se localizam junto de cada um dos Servidores Regionais distribuídos na área de abrangência da empresa de telecomunicações e são responsáveis por analisar os registros de chamada das suas respectivas centrais (locais, T2, T3 e Embratel). Esses registros contêm informações detalhadas sobre cada uma das chamadas efetuadas e são usualmente chamados de CDR (*Call Detailed Register*). A contabilização do valor de cada uma das ligações é baseada na tabela de tarifação correspondente ao horário e dia. O tipo da ligação não é considerado nessa tabela de tarifas, pois as ligações já se encontram separadas por tipos conforme a central que as registra. Por exemplo, uma tabela de tarifas de uma central do tipo T2 apenas conterá tarifas de ligações pertencentes ao degrau de tarifação D1, pois este é o único tipo de ligação registrada por centrais T2.

---

<sup>66</sup> Neste caso, mais importante que a confidencialidade, por exemplo.

<sup>67</sup> Salienta-se que isso não significa que a responsabilidade de detecção de fraude não passa a ser do usuário; o extrato telefônico via Web é apenas mais um serviço disponibilizado ao usuário.

Após a filtragem dos dados relevantes para a contabilização e também após a realização do cálculo da ligação, os dados referentes a cada ligação são armazenados em outra estrutura de dados (uma tabela do *Microsoft Access*) e ficam disponíveis para que o Módulo Montador de Conta (que será detalhado a seguir, no item 4.3.2) envie ao *applet* a página com a conta telefônica do usuário que realizou a consulta.

O módulo Adaptador depende do tipo de central (local, trânsito tipo T2, T3 ou Embratel):

- (1) Centrais Locais – o módulo Adaptador para centrais locais é desnecessário, uma vez que estas centrais apenas registram o montante de impulsos resultantes de ligações locais realizadas por cada um dos clientes. Sendo assim, é preciso apenas que seja feita uma leitura do contador de impulsos do usuário;
- (2) Centrais Trânsito tipo T2 e T3 – o módulo Adaptador destas centrais diferem apenas na sua tabela de tarifas, pois cada uma delas registra tipos de ligações diferentes e, conseqüentemente, com tarifas diferentes; e
- (3) Central Trânsito Embratel – este tipo de central, diferentemente dos outros tipos de centrais, necessita de um módulo Adaptador que, além de filtrar os dados necessários ao cálculo do valor das ligações e efetuar este cálculo, precisa avaliar o tipo de ligação que está sendo efetuada, pois centrais trânsito Embratel podem registrar ligações interurbanas, internacionais e também todas as ligações a cobrar realizadas no território nacional (cada tipo de ligação possuindo uma tabela de tarifação diferente)<sup>68</sup>.

A funcionalidade desses módulos pode ser descrita em diferentes passos:

- (1) leitura do CDR – o módulo lê cada uma das linhas do arquivo-texto, que corresponde às ligações que foram registradas nas centrais (T2, T3 ou Embratel). Enquanto forem enviados novos lotes de ligações registradas pelas centrais, o módulo efetuará a leitura dos mesmos. Os passos abaixo são realizados para cada linha lida, ou seja, para cada ligação registrada no arquivo;
- (2) validação do usuário – o módulo verifica se o telefone do usuário que efetuou a ligação que está sendo avaliada faz parte do cadastro de usuários do SETWeb. Essa consulta é feita na tabela *cadSETWeb* da base de dados da central. Caso o usuário não utilize o serviço SETWeb, a ligação é ignorada e o módulo avaliará a próxima linha do arquivo. Do contrário, os passos abaixo são realizados;

---

<sup>68</sup> Assim como a leitura dos CDRs das companhias telefônicas só é possível com a permissão das respectivas companhias, os CDRs da Embratel também só podem ser lidos com a devida permissão da mesma.

- (3) filtragem dos dados – uma única ligação é formada por uma quantidade de campos bastante grande. Dependendo do formato do CDR, o número de campos pode chegar a 50 ou mais. Por isso, é necessário filtrar apenas os dados relevantes para o cálculo do valor da ligação e os dados que identificam a mesma. O módulo Adaptador se encarregará de selecionar apenas os seguintes dados do CDR: dataInicio, horaInicio, duração, numDeA, numDeB. Posteriormente, esses dados podem vir a ser subdivididos para permitir que outros passos possam ser realizados. Por exemplo, para se obter a localidade do destino da ligação a partir do prefixo e/ou DDD do numDeB;
- (4) cálculo do valor da ligação – o cálculo do valor da ligação é feito através de uma comparação dos dados filtrados (data, hora e duração da ligação) com os dados que compõem a tabela de tarifação da central (ver Tabelas 4.3 e 4.4), e toma-se por base o número de Seg/impulso referente ao dia e hora em que a ligação foi realizada (buscado na tabela de tarifação);
- (5) busca do destino da ligação – um dos dados referentes à ligação que deve ser mostrado na conta discriminada é o local correspondente ao número chamado. Cada central contém os códigos de área e prefixos de telefones que fazem parte da área de abrangência desta central, juntamente com os respectivos nomes das localidades a que correspondem aqueles códigos de área e prefixos. Dessa forma, o Módulo Adaptador avalia o dado numDeB (número destino) extraído do CDR e busca a cidade correspondente a esse número na tabela de localidades da central;
- (6) classificação da ligação – como centrais T2 e T3 registram sempre o mesmo tipo de ligações, a classe dessas ligações será sempre a mesma. Por exemplo, ligações registradas em T2 e T3 pertencerão à classe DDD; então os Módulos Adaptadores presentes nessas centrais já possuem a classe da ligação como um atributo. Na central Embratel é preciso avaliar o tipo da ligação, uma vez que esse tipo de central registra ligações interurbanas e internacionais. Através dos primeiros dígitos do número chamado é possível diferenciar o tipo da ligação. Ligações interurbanas pertencem à classe DDD e internacionais à classe DDI. Nessas centrais, os módulos Adaptadores classificam a ligação durante a leitura da mesma; e
- (7) disponibilização dos dados da ligação – realizado o cálculo do valor da ligação, e capturado o local de destino e classe da mesma, os dados relacionados à conta devem ficar disponíveis em uma tabela de uma base de dados. Essa base de dados será composta por todas as ligações de usuários do SETWeb da central, sendo que cada ligação será representada apenas pelos dados que aparecerão no extrato: Número de Origem, Classe, Data, Hora, Duração, Destino, Número Chamado, Valor em Reais.

### 4.3.2 Módulo Montador de Conta Telefônica

A funcionalidade deste módulo é fazer consultas aos diversos bancos de dados distribuídos, ou seja, solicitar todas as ligações do cliente que está efetuando a consulta. Essa requisição só é realizada após a validação da identificação do usuário (número de telefone e senha) que consta na base de dados de usuários da companhia telefônica que solicitaram tal serviço. O Montador de Conta Telefônica é responsável por realizar a contabilização total (valor gasto pelo cliente até o momento da consulta) a partir da contabilização parcial (valor de cada ligação realizada pelo cliente) que é consultada nos bancos de dados que contêm ligações do cliente em questão. Além disso, este módulo é responsável por mostrar a resposta da consulta (conta discriminada) na página da Internet. O formato dessa resposta é similar ao formato da conta recebida no final do mês.

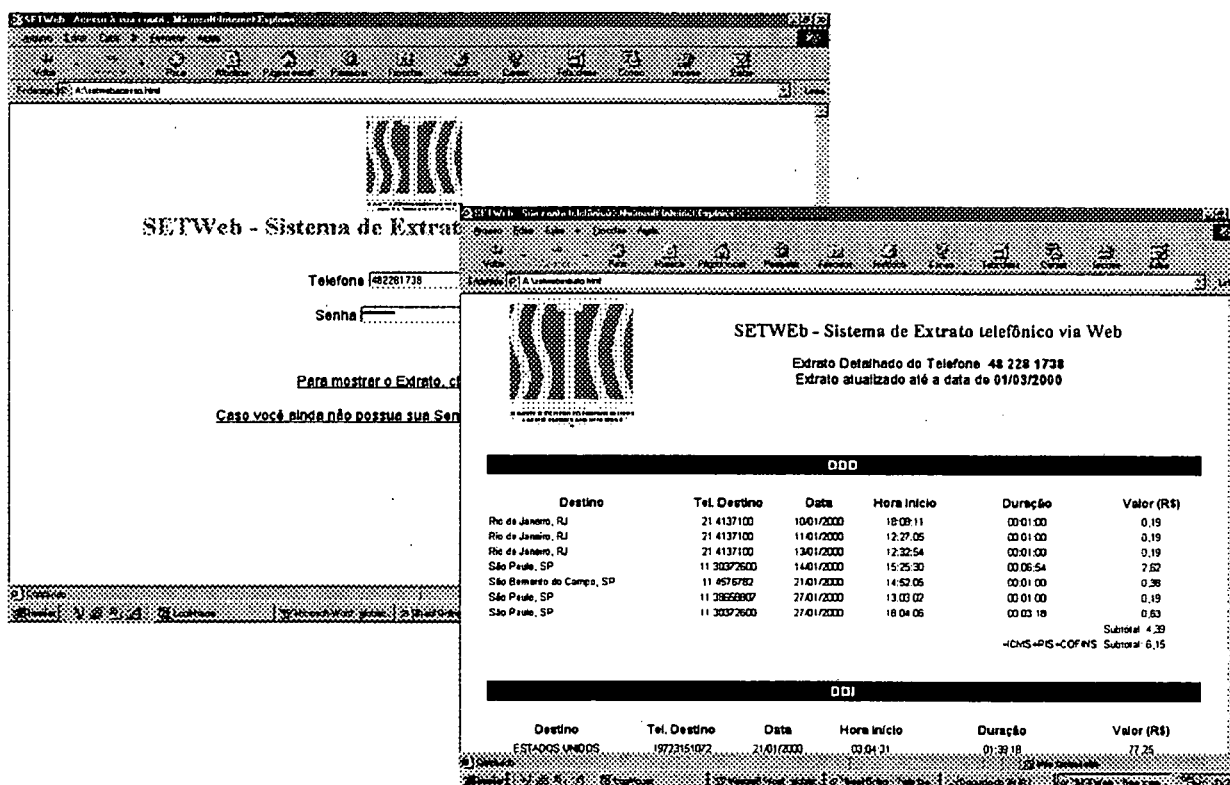


FIGURA 4.10 – PÁGINA INICIAL DO SETWEB.

FIGURA 4.11 – PÁGINA COM A CONTA DISCRIMINADA.

O módulo Montador tem como interface com o usuário a Web, que disponibiliza o Extrato Telefônico através de uma *applet*. A Figura 4.10 mostra a solicitação do número do telefone e a senha correspondente ao usuário. A Figura 4.11 mostra o extrato atualizado até o momento da consulta.



### 4.3.3 Bases de Dados do SETWeb

Para manipular, armazenar e disponibilizar todos os dados que fazem parte deste sistema, foi necessário o uso de bases de dados do *Microsoft Access* e de arquivos-texto. As bases de dados utilizadas estão apresentadas abaixo.

#### 4.3.3.1 CDR

As centrais possuem um arquivo texto contendo, em cada linha (CDR) os seguintes campos: (1) dataInicio: data em que a ligação se iniciou; (2) horaInicio: horário em que a ligação se iniciou; (3) duração: tempo de duração da ligação (em segundos); (4) numDeA: número de origem; e (5) numDeB: número de destino.

#### 4.3.3.2 Wtelecom

Esta base de dados consiste em três tabelas: (1) cadClientes: contém o cadastro de dados pessoais dos clientes da empresa; (2) cadClientesSETWeb: contém os números de telefones e senhas dos usuários da empresa que solicitaram o serviço SETWeb; e (3) mensalidades: contém os tipos de mensalidades cobradas pela empresa.

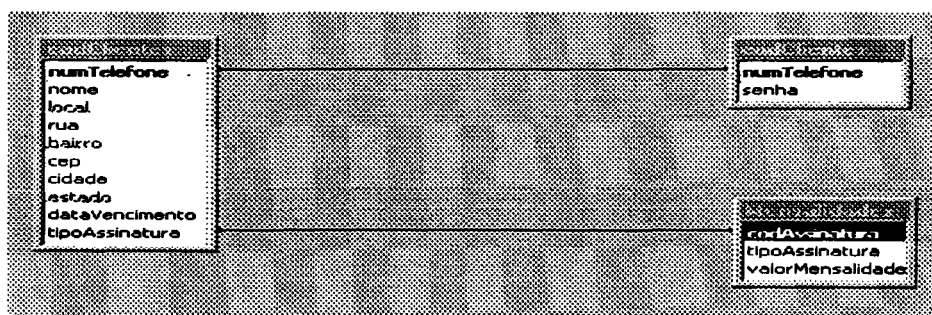


FIGURA 4.12– BASES DE DADOS LOCALIZADAS NA “MATRIZ” DA EMPRESA TELEFÔNICA.

A Figura 4.12 apresenta o relacionamento entre as tabelas cadClientes, cadClientesWeb e mensalidades.

#### 4.3.3.3 Central Local

Consiste de somente uma tabela: (1) pulsosC233FNS: contém todos os telefones que pertencem à determinada central local (neste exemplo, a central 233 de Florianópolis) e a respectiva quantidade de pulsos de cada um dos telefones.

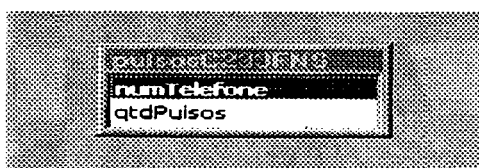


FIGURA 4.13 – BASE DE DADOS DAS CENTRAIS LOCAIS.

Essa tabela apresentada na Figura 4.13 está associada ao contador de pulsos.

#### 4.3.3.4 Central T2

Esta base de dados consiste nas seguintes tabelas: (1) cadT2SETWeb: cadastro dos clientes que fazem parte da área de abrangência desta central e que utilizam o serviço; (2) localidadesT2: cadastro de todos os prefixos de telefones que fazem parte da área de abrangência desta central, juntamente com os respectivos nomes das localidades a que correspondem aqueles prefixos, pois é necessário que na conta telefônica esteja discriminado, por extenso, o nome da cidade para onde a ligação foi realizada; (3) tarifasT2: tabela de valores de tarifas de ligações pertencentes ao degrau de tarifação D1 que serão utilizados para o cálculo das ligações registradas por T2; e (4) CDRFiltrado: tabela com todas as informações referentes às ligações efetuadas pelos usuários, e que serão mostradas aos mesmos. Juntamente com esses dados, está o nome da cidade-destino da ligação, o valor da ligação ( ) e o número do telefone originador da ligação.



FIGURA 4.14 – BASE DE DADOS DAS CENTRAIS T2.

A Figura 4.14 apresenta as quatro tabelas associadas às Centrais T2.

#### 4.3.3.5 Central T3

Esta base de dados consiste nas seguintes tabelas: (1) cadT3SETWeb: cadastro de todos os clientes que fazem parte da área de abrangência desta central e que utilizam o serviço; (2) localidadesT3: cadastro de todos os códigos de área e prefixos de telefones que fazem parte da área de abrangência desta central, juntamente com os respectivos nomes das localidades a que correspondem aqueles códigos de área e prefixos; (3) tarifasT3: tabela de valores de tarifas de ligações pertencentes ao degrau de tarifação D2 que serão utilizados para o cálculo das ligações registradas por T3; e (4) CDRFiltrado: tabela com as informações referentes às ligações efetuadas pelos usuários, com o valor em reais de cada ligação registrada pela central. Os dados pertencentes a esta tabela são os dados que serão mostrados aos usuários. Nesta mesma tabela, também consta o número do telefone originador da ligação.

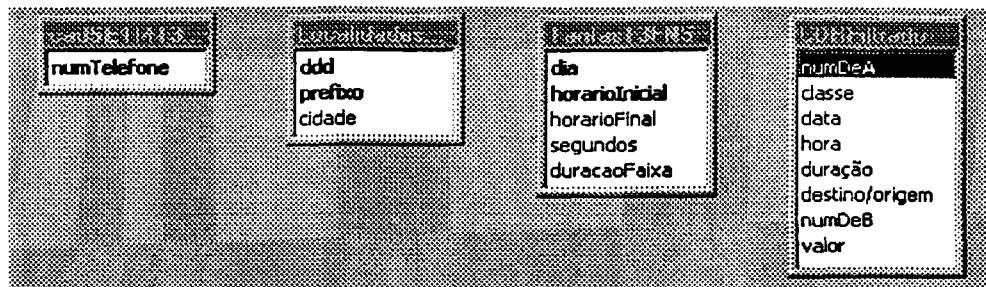


FIGURA 4.15 – BASES DE DADOS DAS CENTRAIS T3.

A Figura 4.15 apresenta as quatro tabelas associadas às Centrais T3.

#### 4.3.3.6 Central Trânsito Embratel

Esta base de dados consiste nas seguintes tabelas: (1) `cadSETWebSC`: cadastro de todos os clientes da empresa de telecomunicações (no exemplo, de Santa Catarina) que utilizam o serviço, composto do código DDD seguido pelo número do telefone; (2) `localidadesEmbrIU`: cadastro de todos os códigos de área, prefixos de telefones e nomes das cidades de todo o país (exceto do estado de Santa Catarina, neste exemplo); (3) `tarifasEmbrIU`: tabela de valores de tarifas de ligações pertencentes ao degrau de tarifação D3 que serão utilizados para o cálculo das ligações interurbanas registradas por esta central; (4) `tarifasEmbrDI`: tabela de valores de tarifas de ligações pertencentes ao degrau de tarifação DI (internacional) que serão utilizados para o cálculo das ligações internacionais registradas por esta central; e (5) `CDRFiltrado`: tabela com as informações referentes às ligações efetuadas pelos usuários, com o valor em reais de cada ligação registrada pela central e o nome da cidade para onde a ligação foi efetuada. Os dados pertencentes a esta tabela são os dados que serão mostrados aos usuários. Nesta mesma tabela, também consta o número do telefone originador da ligação. Esse número é necessário para que possa ser efetuada a busca de ligações de um usuário a partir do seu número de telefone.

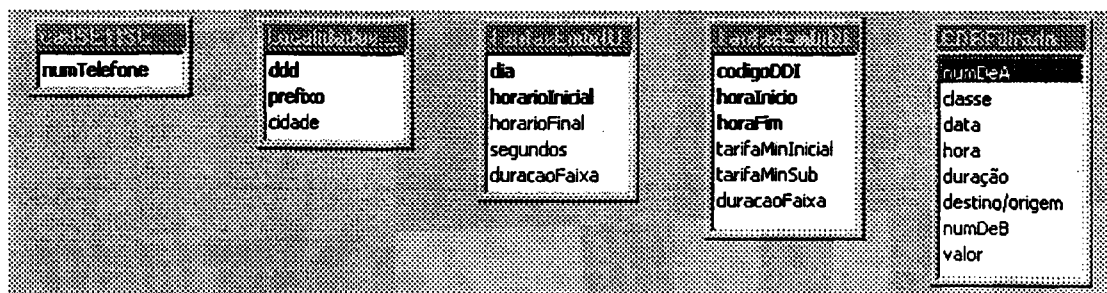


FIGURA 4.16 – BASES DE DADOS DE UMA CENTRAL TRÂNSITO EMBRATTEL.

A Figura 4.16 apresenta as cinco tabelas associadas à Central de Trânsito Embratel.

#### 4.3.4 Implementação da Gerência em Ambiente Web

Ao se utilizar o protocolo HTTP para o acesso a um endereço Web se estabelece uma conexão TCP com o servidor que disponibiliza o conteúdo de tal endereço. A conexão TCP é finalizada após os dados serem transmitidos. O conteúdo de endereços Web pode incluir, por exemplo, *applets*, cuja função é a execução dinâmica de um código – o que torna clientes e servidores vulneráveis a ataques. O “contra-ataque” envolve a implementação de mecanismos de segurança para o cliente, o servidor e o protocolo.

Para a implementação dos mecanismos de segurança do Sistema SETWeb, investigou-se: 1) a segurança oferecida pelos navegadores (*browsers*); 2) os serviços de segurança da arquitetura CORBA e da linguagem Java; e 3) as facilidades para a autenticação dos clientes autorizados são disponibilizadas através de protocolos tais como SSL (*Secure Sockets Layer*), SSH (*Secure Shell*), HTTPS (*Hypertext Transfer Protocol sobre SSL* – um exemplo de uma extensão ao protocolo HTTP para o estabelecimento de transações WWW seguras) e o protocolo SET (*Secure Eletronic Transaction* – um esforço conjunto da Visa, *Mastercard*, da *Netscape* e da IBM para o estabelecimento de um protocolo seguro para o transporte de informações em Sistemas de Pagamento Eletrônico).

As principais ferramentas utilizadas para a implementação dos mecanismos de segurança no Sistema SETWeb estão relacionadas na Tabela 4.5 a seguir.

TABELA 4.5 – FERRAMENTAS UTILIZADAS PARA PROVER SEGURANÇA NO SETWEB.

<b>Visibroker for Java</b>	ORB CORBA, cuja tradução de IDLs é específica para a linguagem Java.
<b>Java Developer Kit</b>	Ferramenta de desenvolvimento de aplicações na linguagem Java que permite administrar e configurar políticas de segurança com serviços de criptografia, controle de acesso e gerência de chaves e certificados digitais.
<b>HTML Converter for Java</b>	Conversor de páginas escritas HTML para a inclusão de <i>tags</i> que identifiquem a presença de uma <i>applet</i> escrita na versão 1.2 do JDK.
<b>Policytool</b>	Ferramenta do JDK, cuja função é permitir que sejam estabelecidas permissões de acesso e modificação de componentes através de <i>applets</i> .
<b>Keytool</b>	Ferramenta do JDK, cuja função é gerar chaves criptográficas para os usuários.

A gerência de permissões é feita pela ferramenta `policytool`, ou através da configuração manual do arquivo `java.policy` localizado no diretório `\jdk1.2\jre\lib\security`. No SETWeb, a mesma foi utilizada com o intuito de permitir que as informações do usuário fossem armazenadas em um BD localizado no *host* servidor, assim como permitir que o usuário, consultasse informações de sua chave privada. A Figura 4.17 exibe as permissões que foram necessárias para a aplicação.

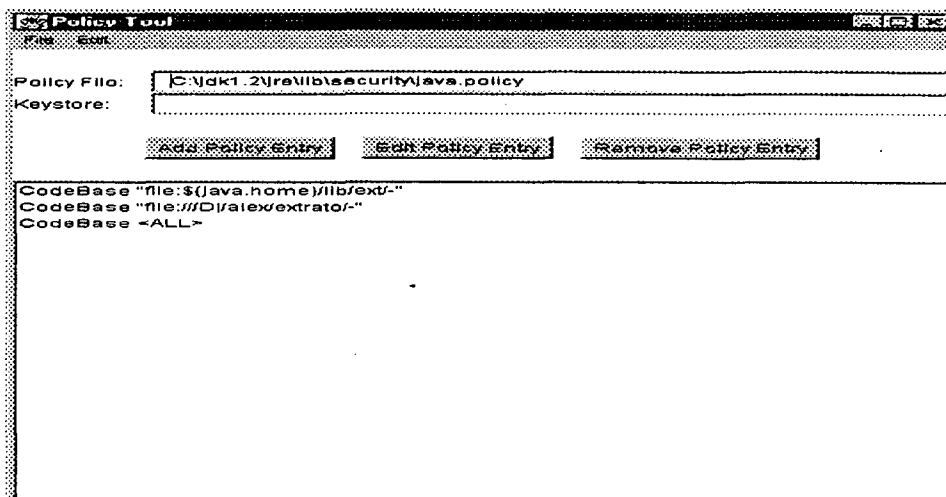


FIGURA 4.17 – PERMISSÕES CONCEDIDAS A *APPLETS*.

No entanto, os *browsers* atuais ainda não incluem todas as classes utilizadas pelo JDK 1.2, reconhecidas pelas MVJs. Assim, um programa adicional (*plug-in*) é necessário para que tais classes possam ser reconhecidas. Neste caso, foi necessária a utilização da ferramenta *Java HTML Converter*, cuja função é incluir informações adicionais nas páginas HTML que possuem chamadas a classes produzidas no JDK 1.2 (MATOS, 1999).

A gerência de segurança do SETWEB inicia-se pela inclusão de novos usuários e, conseqüentemente, há geração de chaves criptográficas para que estes possam cadastrar suas senhas de acesso e fornecer suas informações pessoais. Após cadastrado, o usuário recebe em casa uma aplicação que instala em sua máquina informações sobre o *site* do serviço e também a sua chave privada. Nesse instante é que o usuário fornece sua senha, pois dessa forma sua chave privada, que é utilizada para criptografar a senha, não pode ser interceptada pela rede, por ter sido transportada por correio. A Figura 4.18 apresenta a interface onde o usuário fornece seus dados pessoais. Sua inclusão no cadastro automaticamente dispara a criação das chaves criptográficas.

SETWeb - Acesso à sua conta - Microsoft Internet Explorer

Arquivo Editar Exibir Ferramentas Avançadas Ajuda

Voltar Avançar Parar Atualizar Recarregar Favoritos Histórico Cache Ferramentas Correio Imprimir E-mail

Endereço A:\setweb\senha.html

**SETWeb - Sistema de Extrato telefônico via Web**

**Requisição de senha segura**

Telefone

Senha

Essa senha será criptografada e enviada para você via correio; somente após a instalação da senha será possível acessar seu Extrato via Web.

Por favor, preencha os dados abaixo:

Nome:  Endereço:

Cidade, Estado:  Outros telefones, fax:

Endereço Eletrônico:  CPF:

Clique  para confirmar a Requisição de Senha Segura.

Concluído Meu computador

Iniciar Logoff Microsoft Word - globe Microsoft Office - Tabela Planilha de Excel SETWeb - Acesso

FIGURA 4.18 – CADASTRO DE USUÁRIO NO SISTEMA DE EXTRATO TELEFÔNICO VIA WEB.

As chaves criadas são armazenadas em um Banco de Dados. Esse procedimento facilita a tarefa de modificação de senha do usuário, assim como permite o fornecimento imediato da instalação da chave privada do usuário quando o mesmo perdê-la, ou até mesmo quiser recadastrar-se.

Verificou-se na carga da *applet* um leve declínio de performance, devido a necessidade da carga do *plug-in* que permite a interpretação da classe produzida no JDK 1.2. No entanto, esse fato não impede de se considerar o uso de Java/CORBA a solução mais indicada, pois se trata de um serviço distribuído, e o tempo de carga da *applet* apenas compromete no primeiro instante (porque, após a primeira utilização da *applet*, a mesma já está carregada na pilha de execução da MVJ). Uma alternativa descartada foi CGI/HTML, pois a cada nova requisição é necessário um novo acesso a um programa CGI.

Para a implementação dos serviços de segurança foram utilizadas classes da linguagem Java pertencentes à API `java.security`, tais como:

- 1) `java.security.KeyPairGenerator`: responsável por contactar um Provedor de Serviços de Criptografia para que o mesmo dispare o processo de criação de chaves;
- 2) `java.security.PrivateKey`: responsável pela criação de chaves privadas;
- 3) `java.security.PublicKey`: responsável pela criação de chaves públicas;
- 4) `java.security.Signature`: responsável por contactar um Provedor de Serviços de Criptografia (CSP) para que o mesmo dispare o processo de criação de assinaturas digitais;
- 5) `java.security.KeyFactory`: utilizada para instanciar uma chave pública produzida com o algoritmo DAS (Digital Signature Algorithm); e
- 6) `java.security.spec.X509EncodedKeySpec`: utilizada para instanciar chaves a partir de um arquivo externo, caso o usuário já possua sua chave privada. É verificada sua conformidade com o padrão X.509.

A fim de que a autenticidade do *site* seja constatada, é necessário que o usuário confie (comprove) que está utilizando o serviço do local correto. Pode-se, então, prover um certificado digital para o *site*, através de autoridades de certificação que disponibilizam este serviço na Internet (como, p.ex., o da *Verisign*, <<http://digitalid.verisign.com>>). Através de um certificado digital a empresa pode certificar seus usuários de que está sendo provido o serviço esperado e não um outro *site*. Em acréscimo, é importante que o usuário utilize sempre as novas versões dos *browsers*, pois estas possuem as mais atuais implementações dos mecanismos de segurança, o que permite, por exemplo, que o usuário perceba a presença de um *site* certificado ao emitir uma mensagem de aviso para o mesmo.

Em resumo, a implementação dos mecanismos de criptografia, assinatura e certificado digital, apresentados acima, garante os serviços de controle de acesso, confidencialidade, autenticidade, integridade e não-repúdio.

Investigações também foram realizadas considerando as Funções de Relatório de Alarmes de Segurança (SARF), baseadas nas recomendações estabelecidas pelo *International Telecommunication Union* (ITU). A implementação, também com suporte CORBA, tem o objetivo de proporcionar que o gerenciamento de segurança do SETWeb possa ser realizado remotamente, pela própria Web, e dessa forma o gerente responsável pela segurança do SETWeb ganha maior flexibilidade para as tarefas de gerenciamento (BORGES, 1999).

## 4.4 Resultados

O crescimento da Internet e suas oportunidades de negócios<sup>69</sup> é algo que não pode ser mais ignorado. Somente nos EUA o faturamento em 1999 das atividades vinculadas à Internet foi de US\$ 507 bilhões. Isso significa que as transações na rede já representam o primeiro setor da economia – o setor aéreo vem em segundo lugar com US\$ 335 bilhões e a telefonia em terceiro com US\$ 300 bilhões<sup>70</sup>. Considerando esse exemplo, pode-se facilmente imaginar a crescente necessidade de prover soluções de segurança eficientes e adequadas para tais sistemas.

Neste Capítulo 4, apresentaram-se investigações de segurança na Web em três escopos distintos, i.e., (1) protocolos – associados à soluções para a camada de transporte; (2) ambiente – associado à segurança provida pelos navegadores; e (3) linguagens e arquiteturas – associadas aos recursos disponíveis por Java e CORBA. Dessa forma, a proposta apresentada neste trabalho para o desenvolvimento de sistemas de gerência de segurança engloba diferentes serviços: (1) a autenticidade do usuário e do provedor de serviços é verificada; (2) a confidencialidade e integridade de senhas e informações é garantida, i.e., os dados trafegam de maneira confiável; (3) a disponibilidade dos serviços e informações é garantida; (4) o não-repúdio é impedido, ou seja, o usuário e o provedor de serviços não podem desmentir o fato de participarem do processo de consulta, se tiverem participado; e (5) o controle de acesso é implementado, inclusive de forma a facilitar mecanismos futuros de auditoria.

A implementação de segurança foi distinta em relação aos módulos do sistema. No caso da implementação dos Sistemas SSCC e SIPI, por exemplo, mecanismos para a prevenção de interrupção na comunicação entre agentes e gerente (envio de notificações) são de maior importância do que mecanismos de confidencialidade. Já no caso da implementação do Sistema SETWeb, a importância maior está nos mecanismos de autenticação e confidencialidade no acesso às contas telefônicas pelos usuários do que na interrupção deste serviço. Em acréscimo, a política de segurança adotada prevê um nível de segurança

---

<sup>69</sup> Os negócios eletrônicos entre empresas irão dobrar a cada 12 meses nos próximos 5 anos. O valor negociado entre companhias, que foi de US\$ 43 bilhões em 1998, deve atingir US\$ 1,3 trilhão em 2003. Uma justificativa para esse crescimento é o valor da transação. Pode-se citar como exemplo que a compensação de um cheque em uma agência custa em torno de US\$ 1,08; por telefone cai para a metade desse valor; por computador em sistemas do tipo *home banking* custa US\$ 0,26; enquanto que pela Internet o custo fica em apenas US\$ 0,13 [Freitas].

<sup>70</sup> DIÁRIO CATARINENSE. Crescimento da Internet nos Estados Unidos. out. 1999.



maior para os usuários quando acessam suas contas telefônicas do que quando os mesmos acessam informações gerais, tais como valores de tarifas no servidor Web da companhia telefônica. Também, observou-se uma melhor performance na implementação desses mecanismos de segurança quando implementados em uma máquina diferente daquela onde está o servidor Web.

As soluções de segurança implementadas para autenticação e autorização utilizam o CORBA para o envio de identificação e senha dos usuários, de forma criptografada. Em acréscimo, especificamente para o sistema SETWeb, como solução para o controle de acesso dos usuários que não possuem JVM com as classes CORBA (ainda a maioria), utiliza-se essencialmente a segurança provida pelo JDK 1.2 através das ferramentas `policytool` e `keytool`. A segurança implementada no sistema SETWeb oferece assinatura e certificado digital. Enquanto a maioria dos sites de comércio eletrônico (inclusive bancos) atualmente disponíveis oferecem mecanismos para assinatura digital (a fim de que os sites comprovem que o usuário é quem diz ser), pouquíssimos sites oferecem mecanismos para certificado digital (a fim de que os usuários tenham a certeza de que estão acessando o site que pensam ser). Para exemplificar o perigo que pode ser a falta de um mecanismo de certificação digital, considere que um usuário digite como endereço do seu banco <<http://www.bancodobrasil.com.br>> (endereço errado) em vez de <<http://www.bancobrasil.com.br>> (endereço correto). Imagine, também, que um fraudador fez uma réplica (aliás, muito fácil de ser feita) das informações da página verdadeira na página falsa. O usuário, inocentemente, coloca sua identificação e senha na tela inicial. Grande e desagradável surpresa será descobrir posteriormente que uma *applet* acaba de transferir todo o saldo da sua conta para uma outra.

E a preocupação relacionada à segurança na Web (WWW) é cada vez maior, pois a cada dia agregam-se mais usuários e mais tecnologias. Pode-se citar como uma nova tecnologia o MMM<sup>71</sup> (Modo de Mídia Móvel), em que a Internet é acessada por telefone celular. Essa tecnologia está sendo lançada pela empresa *Nokia*<sup>72</sup>, da Finlândia, país onde sete em cada dez habitantes têm telefone celular. *Nokia, Motorola, Ericsson, Samsung e IBM* criaram o Protocolo para Aplicações Sem Fio (*Wireless Application Protocol – WAP*), um padrão

<sup>71</sup> MMM é um trocadilho com WWW de ponta-cabeça.

<sup>72</sup> Nokia é o nome de um rio no interior da Finlândia.

para a comunicação entre os portáteis digitais. É a terceira geração<sup>73</sup>, em que os telefones multimídia integram Internet, TV e pager. Em acréscimo, *Ericsson, IBM, Nokia, Intel e Toshiba* criaram em 1998 um sistema de radiotransmissão<sup>74</sup> que possibilita que o celular troque informações com computadores, abra o portão de casa, receba mensagens da secretária eletrônica e até acenda as luzes da casa. Definitivamente, a terceira geração só terá sucesso se provida de forte segurança.

Como trabalhos futuros, pretende-se atualizar o sistema de acordo com a nova numeração de DDD (com número da operadora e complementar o SETWeb com a telefonia móvel que agregou o algarismo 9 em todos os números). Em acréscimo, implementar o mecanismo de mensagens automáticas (para enviar os alarmes ao usuários via telefone), o que só será possível quando o sistema for implantado em uma companhia telefônica. Atualmente ela é realizada com uma simulação através dos recursos multimídia dos computadores. Finalmente, em conformidade com o padrão WAP da terceira geração de telefonia móvel, disponibilizar a conta telefônica on-line no próprio aparelho – incluindo opção de pagamento.

---

<sup>73</sup> A primeira foi a geração dos analógicos; e a segunda, a dos digitais.

<sup>74</sup> Realizado por um chip chamado *Bluetooth* que envia ondas eletromagnéticas por uma saída de infravermelho. A propósito, o nome é uma homenagem a um chefe viking que unificou tribos inimigas na Dinamarca no Século IX.

## **5 Conclusão e Futuros Trabalhos**

É chegada a hora de tratar a segurança das redes com seriedade. Telefones móveis estão alterando e possivelmente irão alterar ainda mais os hábitos da população. No entanto os usuários precisam ter serviços seguros; e as empresas precisam reduzir seus prejuízos decorrentes das fraudes na telefonia móvel. Assim, as empresas de telecomunicações (SHAW, 1999) devem agregar aos seus serviços uma gerência de segurança eficiente e eficaz. As empresas que investirem em segurança serão bastante beneficiadas – podendo ser as únicas a sobreviver.

Considerando as cinco áreas funcionais de gerência de redes, i.e., configuração, falhas, performance, contabilização e segurança, esta última não tem recebido a atenção devida. Com o aumento da popularidade da telefonia móvel (*mobile*) e das transações comerciais eletrônicas (*e-commerce*) observa-se o crescimento das necessidades de segurança – tanto das empresas quanto dos usuários potenciais das redes sem fio (*wireless*). Os resultados apresentados nesta tese validam a eficiência da proposta para o gerenciamento seguro e correto de sistemas distribuídos e redes de comunicação sem fio.

Neste Capítulo 5 estão sumarizadas as contribuições da pesquisa científica desta tese, associadas à gerência de segurança de redes sem fio no combate às fraudes de clonagem e de habilitação em telefonia móvel (item 5.1). Em acréscimo, são listados tópicos para possíveis trabalhos futuros (item 5.2). Finalmente, no item 5.3, encontram-se outras contribuições, tais como publicações e citações.

### **5.1 Sumário das Contribuições**

As contribuições desta tese associadas à gerência de segurança em sistemas distribuídos, às redes de comunicação sem fio e às fraudes de clonagem e de habilitação na telefonia móvel estão sumarizadas a seguir:

- (i) uma gerência que reúne três tecnologias (LOTOS, Redes Neurais e CORBA) para a gerência de segurança segura, correta e interoperável de sistemas distribuídos, com validação demonstrada através do desenvolvimento de um sistema de segurança para telecomunicações móveis;
- (ii) um modelo para a obtenção da prova formal de correção de sistemas distribuídos, com o objetivo de atingir o nível de segurança *ClassA1* do *Orange Book* (ao contrário dos atuais métodos não formais de verificação de correção de sistemas);
- (iii) uma implementação CORBA/Java para suportar a segurança e a privacidade na comunicação entre agentes e gerente de sistemas distribuídos e Internet (considerando os serviços de segurança de controle de acesso, autenticação, confidencialidade, integridade, disponibilidade e não-repúdio);
- (iv) um serviço de aviso automático e imediato para obter a confirmação das fraudes suspeitas. A vantagem desse aviso ser automático é porque proporciona economia da empresa com pessoal; e a vantagem de ser imediato é porque proporciona economia com a significativa redução de prejuízos em função das fraudes;
- (v) um sistema eficaz contra fraudes de clonagem e habilitação na comunicação móvel que detecta intrusões através do emprego de redes neurais. Os resultados obtidos mostram que, usando 80 neurônios na camada intermediária da rede neural, foi obtida uma taxa de erro de classificação muito boa (2,5%), e dessa forma o sistema pode contribuir significativamente para a redução das perdas das companhias telefônicas em vários milhões de dólares;
- (vi) um sistema provido de assinatura e certificação digital, em que os usuários ao longo do mês podem acessar suas contas telefônicas via Web, permitindo que monitorem e detectem por si próprios a existência de clones de seu aparelho; e
- (vii) um produto - SSTCC® - Sistema de Segurança para Telecomunicações contra Clonagem e Inadimplência - patenteado no INPI (Instituto Nacional de Propriedade Industrial) sob o n. 99001177.

Os resultados obtidos por esta investigação científica indicam que o sistema de gerência de segurança concebido para o desenvolvimento de sistemas distribuídos proposto auxilia fortemente os usuários de telecomunicações móveis contra os danos que podem sofrer, quer sejam danos econômicos (terem que pagar por chamadas que não fizeram), quer sejam danos morais (serem envolvidos com pessoas e/ou negócios de que não tenham conhecimento). Em acréscimo, os resultados demonstram que o sistema de segurança SSTCC reduz significativamente os prejuízos das companhias telefônicas, tornando-as mais competitivas, e que por sua vez podem oferecer serviços mais baratos para toda a população.

## 5.2 Futuros Trabalhos

A partir da pesquisa técnico-científica realizada nesta tese pode-se citar várias sugestões para futuras investigações, como pode ser observado a seguir:

- (i) empregar a gerência proposta em outras aplicações distribuídas, tais como a clonagem de cartões de crédito e a divulgação direcionada de produtos e serviços (*marketing* direcionado). Tais aplicações, assim como as fraudes na telefonia, também são distribuídas, necessitam de forte segurança e são baseadas em reconhecimento de padrões dos usuários. Em acréscimo, estender a implementação de notificação de fraude aos usuários para além dos atuais avisos por celular e por correio, de forma a possibilitar também os avisos por fax e correio eletrônico (*e-mail*), como forma de agregar ainda mais segurança ao sistema;
- (ii) incorporar os tipos abstratos de dados nas especificações LOTOS, de modo que o código C gerado automaticamente a partir dessas especificações LOTOS possa contribuir ainda mais com segurança e rapidez de implementação;
- (iii) investigar e desenvolver novos algoritmos de classificação, além de subdividir as características consideradas (i.e., dividir “ligações internacionais” e “ligações interurbanas” por regiões) visando reduzir ainda mais a taxa de erro obtida nesta pesquisa. Investigações também podem considerar Raciocínio Baseado em Casos (*Case Base Reasonic*), *DataMining* e métodos estatísticos nas tarefas de classificação;

- (iv) alterar a interface entre o Agente Java/CORBA e a Rede Neural/MatLab, atualmente via arquivo ASCII. Ou seja, obter melhor performance do sistema ao utilizar o MCC (*MatLab Compiler Command*) para traduzir o programa MatLab para C/Java e então inseri-lo no código do Agente;
- (v) atualizar o sistema de acordo com as modificações realizadas recentemente no sistema de telecomunicações brasileiro, i.e., as alterações no DDD (Discagem Direta à Distância) com opção de escolha da operadora, e também o acréscimo de um dígito na numeração dos celulares;
- (vi) implantar o sistema em companhias telefônicas, inicialmente na companhia telefônica brasileira que colabora com este trabalho, de modo a possibilitar a implementação efetiva do aviso automático aos usuários (somente possível neste ambiente), e em acréscimo possibilitar a análise do desempenho do sistema em ambiente real; e
- (vii) pesquisar a localização dos fraudadores.

Considera-se que estas linhas de investigações sugeridas são importantes contribuições no escopo desta pesquisa científica.

### 5.3 Outras Contribuições

Em acréscimo às contribuições técnico-científicas apresentadas anteriormente no item 5.1, este item 5.3 lista parte das publicações (item 5.3.1) e citações (item 5.3.2) obtidas ao longo da realização dessa tese.

#### 5.3.1 Publicações e Apresentações

##### Livro/Periódico

BOUKERCHE, A., NOTARE, M.S.M.A. Applications of neural networks to mobile communication systems. In: ZOMAYA, A.Y. *Solutions to parallel and distributed computing problems: lessons from Biological Sciences*. New York: Wiley & Sons, 2000.

WESTPHALL, C.B., KORMANN, L.F., NOTARE, M.S.M.A., RISO, B.G., CRUZ, F.A.S. Editores. IEEE LANOMS'99 - LATIN AMERICAN NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM. *Proceedings ...* Rio de Janeiro, dez 1999. ISBN 85-900382-3-8.

WESTPHALL, C.B., KORMANN, L.F., NOTARE, M.S.M.A., RISO, B.G., CRUZ, F.A.S. Network Management – Solutions and Trends for the Latin and Global Markets. *Journal of Network System Managemen*, New York: Kluwer Academic/Plenum Publishers, vol. 8, n. 2, p. 299-304, abr-maio 2000.

**IEEE**

- NOTARE, M. S. M. A, RISO, B. G., LORENA, P. S., PENNA, M. C. DE O., WESTPHALL, C. B. Formal design of a telecommunications networks management system. AT&T. In: IEEE ISCC'97 - INTERNATIONAL SYMPOSIUM ON COMPUTERS AND COMMUNICATIONS. *Proceedings...* Alexandria, Egito, jul. 1997. p.146-150.
- NOTARE, M. S. M. A, RISO, B. G., LORENA, P. S., PENNA, M. C. DE O., WESTPHALL, C. B. Formal design of a platform for telecommunication heterogeneous networks management. University of Western Sydney - Department of Computing, Nepean, Austrália. In: IFIP/IEEE DSOM'97 - INTERNATIONAL WORKSHOP ON DISTRIBUTED SYSTEMS OPERATIONS AND MANAGEMENT. *Proceedings...* Sydney, Austrália, 21-23 out. 1997, p.263-278.
- NOTARE, M.S.M.A., CRUZ, F.A.S., SOBRAL, J.B.M., ALVES, J.B.M., RISO, B.G., WESTPHALL, C.B. Distributed management in the security area for cloned mobile phones. In: IEEE DSOM'98 - IFIP/IEEE INTERNATIONAL WORKSHOP ON DISTRIBUTED SYSTEMS: Operations & Management. *Proceedings...* University of Delaware, Newark, Delaware, USA, 26-28 out. 1998, p. 14-24.
- NOTARE, M.S.M.A., CRUZ, F.A., RISO, B.G., WESTPHALL, C.B. Wireless communications: security management for cloned cellular phones. In: IEEE WCNC'99 - IEEE WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE. *Proceedings...* New Orleans, LA, USA, 21-24 set. 1999, p.1412-1416.
- NOTARE, M.S.M.A., CRUZ, F.A., RISO, B.G., WESTPHALL, C.B. Security management against cloned cellular telephones. In: IEEE ICON'99 - THE IEEE CONFERENCE ON NETWORKS. *Proceedings...* Brisbane, Queensland, Australia, 28 set.-01 out. 1999, p.356-363.
- NOTARE, M.S.M.A., A. BOUKERCHE, CRUZ, F.A S., RISO, B.G., WESTPHALL, C.B. Security management against cloning mobile telephones. In: IEEE GLOBECOM'99 - GLOBAL COMMUNICATIONS. *Proceedings...* Rio de Janeiro, RJ, Brasil, 05 - 09 dez. 1999, p.1969-1973.
- BOUKERCHE, A., NOTARE, M.S.M.A. Neural fraud detection in mobile phone operations. In: 14 th IPSPS - INTERNATIONAL PARALLEL PROCESSING SYMPOSIUM & 11th SPDP - SYMPOSIUM ON PARALLEL & DISTRIBUTED PROCESSING / 3rd WORKSHOP ON BIOLOGICALLY INSPIRED SOLUTIONS TO PARALLEL PROCESSING PROBLEMS (BioSP3). IEEE Computer Society. *Proceedings...* Spring Verlag, Cancun, 01 maio 2000, p. 636-644.
- NOTARE, M.S.M.A., BOUKERCHE, A., WESTPHALL, C.B. Safety and security for 2000 telecommunications. In: IEEE EUROCOMM 2000 - EUROPE COMMUNICATIONS: Information systems for enhanced public safety & security. Session 6C: Tactical and governmental communications: II. *Proceedings...* Munich, Alemanha, 17-19 maio 2000.

**Internacional**

- NOTARE, M. S. M. A, MACIEL, C., FURLANETTO, E.M, RISO, B. G., WESTPHALL, C. B. Proyecto formal de una plataforma para gerenciamento de redes de telecomunicaciones. In: III ICIE - INTERNATIONAL CONGRESS ON INFORMATION ENGINEERING. *Proceedings...* University of Buenos Aires. 16-17 abr. 1997, p. 487. Resumo.
- NOTARE, M.S.M.A., FURLANETTO, E.M., CRUZ, F.A.S., RISO, B.G, WESTPHALL, C.B. Proyecto formal de una plataforma de gerenciamento. In: CACIC 97 - Congresso Argentino de Ciências de la Computación. *Proceedings...* La Plata, Argentina. 29 set.- 4 out. 1997.
- NOTARE, M.S.M.A., FURLANETTO, E.M., RISO, B.G., WESTPHALL, C. B. Proyecto formal de una plataforma de gerenciamento. In: III CACIC. *Proceedings...* La Plata, Argentina. 29 set. - 1 out. 1997, p.51-61.

- CRUZ, F.A. DA S., NOTARE, M.S.M.A., REIS, M.M., GAUTHIER, F., ALVES, J.B.M., RISO, B.G., WESTPHALL, C.B. Uso de inteligência artificial na implementação de um sistema de gerência proativo para redes ATM. In: XXIII CONFERENCIA LATINOAMERICANA DE INFORMÁTICA. *Proceedings...* Valparaíso, Chile, 10-15 nov. 1997, p.465-474.
- NOTARE, M.S.M.A., ROGERIO, K.O., MACIEL, C., RISO, B.G., WESTPHALL, C.B. Formal design of management objects. In: XXIII CONFERENCIA LATINOAMERICANA DE INFORMÁTICA. *Proceedings...* Valparaíso, Chile, 10-15 nov. 1997, p. 455-464.
- CRUZ, F.A., NOTARE, M.S.M.A., MARTINS, A., WEBER-LEE, R., BARCIA, R.M. Using case-based reasoning in an intelligent management system. In: EIS'98 - INTERNATIONAL SYMPOSIUM ON ENGINEERING OF A INTELLIGENT SYSTEMS. *Proceedings...* University of La Laguna, Tenerife, Spain, 11-13 fev. 1998. CD-ROM, p. 1093-1099.
- MACIEL, C., RISO, B.G., NOTARE, M.S.M.A. Especificação Formal com Uso da biblioteca bibLOTOS. In: IDEAS'98 - WORKSHOP IBEROAMERICANO DE ENGENHARIA DE REQUISITOS E AMBIENTES DE SOFTWARE. *Proceedings...* Torres, RS, 01-03 abr. 1998, p 26-37.
- BLAY, E.A., NOTARE, M.S.M.A., RISO, B.G. Especificação LOTOS do comportamento de elementos de rede: um estudo de caso. In: IDEAS'98 - WORKSHOP IBEROAMERICANO DE ENGENHARIA DE REQUISITOS E AMBIENTES DE SOFTWARE. *Proceedings...* Torres, RS, 01-03 abr. 1998, p 313-314.
- NOTARE, M.S.M.A., CRUZ, F.A., ERNST, C., ALVES, J.B. M., RISO, B.G., WESTPHALL, C.B. Object-oriented modeling and design of a intelligent management system for ATM networks. In: IDEAS'98 - WORKSHOP IBEROAMERICANO DE ENGENHARIA DE REQUISITOS E AMBIENTES DE SOFTWARE. *Proceedings...* Torres, RS, 01-03 abr. 1998, p 322-323.
- NOTARE, M.S.M.A., CRUZ, F.A., RISO, B.G., ALVES, J.B.M., WESTPHALL, C. B. A Security telecommunication management system. In: EUROAMERITEL'98. *Proceedings...* Santiago, Chile, 08-10 jun. 1998.
- CRUZ, F.A., NOTARE, M.S.M.A., REIS, M.M., ALVES, J.B.M., RISO, B.G., WESTPHALL, C.B. Use of artificial intelligence techniques in the implementation of a performane management system for telecommunications networks. In: SPECTS 98. *Proceedings ...* Reno, Nevada. 19-22 jul. 1998.
- SOUZA, F.P., LEITE, K.C., NOTARE, M.S.M.A., WESTPHALL, C.B. Gerência distribuída na área de segurança para redes de telecomunicações de alta velocidade. In: CACIC'98. *Proceedings...* Neuquén, Argentina, 26-31 out. 1998.
- MATOS, A. V., NOTARE, M.S.M.A., RISO, B.G. Uso de LOTOS na gerência de segurança de aplicações distribuídas em Java. In: CACIC'98. *Proceedings ...* Neuquén, Argentina, 26-31 out. 1998.
- SOUZA, F.P., LEITE, K.C., NOTARE, M.S.M.A., WESTPHALL, C.B. Uma aplicação distribuída para redes de telecomunicações utilizando tecnologias Web e CORBA. In: WICC99 - WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN. *Proceedings ...* San Juan, Argentina, 27-28 maio 1999.
- CRUZ, F.A., NOTARE, M.S.M.A., RISO, B.G., WESTPHALL, C.B. Gerência proativa de redes com técnicas de inteligência artificial. In: V Congreso Internacional de Ingeniería Informática. *Proceedings...* Universidade de Buenos Aires, Facultad de Ingeniería, Departamento de Computación. Argentina, 18-20 ago. 1999.
- CRUZ, F.A., NOTARE, M.S.M.A., RISO, B.G., ALVES, J.B.M., WESTPHALL, C.B. Gerência proativa de redes com técnicas de inteligência artificial. In: XXV CLEI'99 - CONFERENCIA LATINOAMERICANA DE INFORMÁTICA / IV CONGRESO INTERNACIONAL DE TECNOLOGIAS Y APLICACIONES INFORMATICAS / X PANEL NACIONAL DE INFORMÁTICA. *Proceedings...* 30 ago.-03 set. 1999.
- NOTARE, M.S.M.A., CRUZ, F.A., WESTPHALL, C.B. An intrusion detection system to mobile phone networks. In: World Exposition EXPO'2000. *Proceedings ...* Hannover, Alemanha, 11-12 jul. 2000.



**SBC / SBT**

NOTARE, M.S.M.A., MACIEL, C., FURLANETTO, E.M., RISO, B.G., WESTPHALL, C.B. bibLOTOS: Une bibliothèque de constructions prédéfinies pour un projet de gestion de réseau. In : SFBSID'97 - II SEMINÁRIO FRANCO-BRASILEIRO EM SISTEMAS INFORMÁTICOS DISTRIBUÍDOS - ARQUITETURAS MULTIMÍDIAS PARA AS TELECOMUNICAÇÕES / 2 ÈME SEMINAIRE FRANCO-BRASILIEN SUR LES ARCHITECTURES DES SYSTÈMES DISTRIBUÉS - ARCHITECTURES MULTIMÉDIAS POUR LES TÉLÉCOMMUNICATIONS. *Proceedings...* UFC/RNP-CE, Lab. PRISM/UVSQ, Fr. Univ. de Versailles, France. Fortaleza, CE, 03-07 nov.19 97, p.565-578.

MACIEL, C., FURLANETTO, E.M., NOTARE, M. S. M. A., RISO, B.G. Projeto de sistemas par a gerência de redes utilizando construções predefinidas: a biblioteca bibLOTOS. In: XVII SEMISH. *Proceedings...* Brasília, DF. 02-08 ago.19 97

WESTPHALL, C.B., RISO, B.G., KORMANN, L.F., NOTARE, M.S.M.A., MARTINS, J.S.B., LORENA, P.S., PENNA, M.C.O. PLAGERE Project - platforms for network management. proTeM-CC conference - projects Fase II. In: WORKSHOP OF INTERNATIONAL EVALUATION OF THE PROTEM-CC PROGRAM. *Proceedings...* Belo Horizonte, MG, 02-03 abr. 1998, p.283-316.

CRUZ, F.A., NOTARE, M.S.M.A., ALVES, J.B.M., RISO, B.G., WESTPHALL, C.B., Design of an intelligent management system for ATM networks. In: XVI SBRC. *Proceedings...* UFF, Rio de Janeiro, RJ, 25-28 maio 1998, p.356.

NOTARE, M.S.M.A., CRUZ, F.A., RISO, B.G., WESTPHALL, C.B. Network management against cloning of telephones. In: SBT'99. *Proceedings...* Vila Velha, ES, 07-10 set. 1999.

NOTARE, M.S.M.A., WESTPHALL, C.B. Concepção, Desenvolvimento e Análise de um Sistema de Segurança para Redes de Telecomunicações. In: SEMISH00. *Proceedings...* Curitiba, PR, 17-21 jul. 2000, p.19-20.

**Nacional**

NOTARE, M.S.M.A., RISO, B.G., LORENA, P.S., PENNA, M.C. DE O. N., WESTPHALL, C.B. Formal design of a telecommunications networks management system. In: V SEMANA DA PESQUISA UFSC. *Anais...* Florianópolis, SC, 13-17 out. 1997, p.41. Resumo.

NOTARE, M.S.M.A., CRUZ, F.A. DA S., WEBER-LEE, R., MARTINS, A., BARCIA, R. Using case-based reasoning in an intelligent management system. In: V SEMANA DA PESQUISA UFSC. *Anais...* Florianópolis, SC, 13-17 out. 1997, p.17. Resumo.

WESTPHALL, C.B., NOTARE, M.S.M.A., RISO, B.G., LORENA, P.S., PENNA, M.C. DE O. N. Formal design of a platform for telecommunication heterogeneous network management. In: V SEMANA DA PESQUISA UFSC. *Anais...* Florianópolis, SC, 13-17 out. 1997, p.35. Resumo.

RISO, B.G., NOTARE, M. S. M.A., MACIEL, C., FURLANETTO, E.M., WESTPHALL, C.B. Proyecto formal de una plataforma de gerenciamiento. In: V SEMANA DA PESQUISA UFSC. *Anais...* Florianópolis, SC, 13-17 out. 1997, p. 34. Resumo.

RISO, B.G., NOTARE, M.S.M.A., ROGERIO, K.O., MACIEL, C., WESTPHALL, C.B. Formal design of Managed objects. In: V SEMANA DA PESQUISA UFSC. *Anais...* Florianópolis, SC, 13-17 out. 1997, p. 33. Resumo.

RISO, B.G., MACIEL, C., NOTARE, M.S.M.A., WESTPHALL, C.B. bibLOTOS: Uma biblioteca de construções predefinidas para o projeto da gerência de redes. In: V SEMANA DA PESQUISA UFSC. *Anais...* Florianópolis, SC, 13-17 out. 1997, p.32. Resumo.

- FURLANETTO, E.M., MACIEL, C., NOTARE, M.S.M.A., RISO, B.G. Duas experiências em engenharia de sistemas com LOTOS no Laboratório de Redes e Gerência. In: V SEMANA DA PESQUISA UFSC. *Anais...* Florianópolis, SC, 13-17 out. 1997, p.36. Resumo.
- FURLANETTO, E.M., NOTARE, M.S.M.A., RISO, B.G. Engenharia de sistemas de gerência de redes com LOTOS no Laboratório de Redes e Gerência. In: VII SIC SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UFSC. *Anais...* Florianópolis, SC, 13-17 out. 1997. Resumo.
- CRUZ, F.A. DA S., NOTARE, M.S.M.A., ERNST, C., RISO, B.G., WESTPHALL, C.B. Object-oriented modeling and design of an intelligent management system for ATM networks using CORBA. In: V SEMANA DA PESQUISA UFSC. *Anais...* Florianópolis, SC, 13-17 out. 1997, p. 37. Resumo.
- CRUZ, F.A. DA S., NOTARE, M.S.M.A., REIS, M.M., GAUTHIER, F., RISO, B.G., WESTPHALL, C.B. Uso de inteligência artificial na implementação de um sistema de gerência proativo para redes ATM. In: V SEMANA DA PESQUISA UFSC. *Anais...* Florianópolis, SC, 13-17 out. 1997, p. 38. Resumo.
- CRUZ, F.A. DA S., NOTARE, M.S.M.A., KORMANN, L.F., MARTINS, J.S.B, RISO, B.G., WESTPHALL, C.B. Use of artificial intelligence in the implementation of a proactive management system for ATM networks. In: V SEMANA DA PESQUISA UFSC. *Anais...* Florianópolis, SC, 13-17 out. 1997, p. 39. Resumo.
- NOTARE, M.S.M.A., MACIEL, C., RISO, B.G., WESTPHALL, C.B. BibLOTOS: une bibliothèque de constructions prédéfinies pour un projet de gestion de réseau. In: V SEMANA DA PESQUISA UFSC. *Anais...* Florianópolis, SC, 13-17/ out. 1997, p. 40. Resumo.
- NOTARE, M.S.M.A., CRUZ, F.A.S., ALVES, J.B.M. Applying Prolog programming to implement an ATM MIB. In: V SEMANA DA PESQUISA UFSC. *Anais...* Florianópolis, SC, 13-17/ out. 1997, p. 42. Resumo.
- CRUZ, F.A., NOTARE, M.S.M.A., MATIAS JR, R., TODESCO, J.L., RISO, B.G., WESTPHALL, C.B. Gerência inteligente com o emprego de técnicas de reconhecimento de padrões. In: II WORKSHOP DE SOLUÇÕES NÃO CONVENCIONAIS EM SISTEMAS DE GERÊNCIA DE REDES. *Anais...* Itapipoca, CE, 27-28 jan. 1998.
- CASTELLO BRANCO, W.N., MILIOLI, C.F., NOTARE, M.S.M.A., SOBRAL, J.B.M., WESTPHALL, C.B. Sistema de segurança para telefones celulares clonados (Painel n. 153). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul - Centro de Eventos de Florianópolis, 15-18 set. 1998, p. 157.
- BLAY, E.A., NOTARE, M.S.M.A., RISO, B.G. Especificação LOTOS do comportamento de elementos de redes: um estudo de caso (Painel n. 519). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul - Centro de Eventos de Florianópolis, 15-18 set. 1998, p. 520.
- COSTA, T.F.S., NOTARE, M.S.M.A., RISO, B.G. Dedução de protocolos, com LOTOS, no ensino orientado para restrições (Painel n. 521). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul - Centro de Eventos de Florianópolis, 15-18 set. 1998, p. 521.
- CRUZ, F.A., NOTARE, M.S.M.A., WESTPHALL, C.B., MARTINS, A., WEBBER-LEE, R., BARCIA, R.M. Using case-based reasoning in an intelligent management system (Painel n. 522). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul - Centro de Eventos de Florianópolis, 15-18/ set. 1998, p. 521.
- CRUZ, F.A., NOTARE, M.S.M.A., REIS, M.M., ALVES, J.B.M., RISO, B.G., WESTPHALL, C.B. Use of artificial intelligence techniques in the implementation of a performance management system for telecommunications networks (Painel n. 523). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul - Centro de Eventos de Florianópolis, 15-18 set. 1998, p. 522.

- CRUZ, F.A., NOTARE, M.S.M.A., MATIAS, R.J., TODESCO, J.L., RISO, B.G., WESTPHALL, C.B. Gerência inteligente com o emprego de técnicas de reconhecimento de padrões (Painel n. 524). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul – Centro de Eventos de Florianópolis, 15-18 set. 1998, p. 522.
- CRUZ, F.A., NOTARE, M.S.M.A., ALVES, J.B.M., RISO, B.G., WESTPHALL, C.B. Design of an intelligent management system for ATM networks (Painel n. 525). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul - Centro de Eventos de Florianópolis, 15-18 set. 1998, p. 523.
- FURLANETTO, E.M., MELILO, E.C., MATOS, A.V., NOTARE, M.S.M.A., RISO, B.G., WESTPHALL, C.B. Um estudo das propriedades dos protocolos (Painel n. 526). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul - Centro de Eventos de Florianópolis, 15-18 set. 1998, p. 523.
- MACIEL, C., RISO, B.G., NOTARE, M.S.M.A. Especificação formal com o uso da biblioteca bibLOTOS (Painel n. 527). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul - Centro de Eventos de Florianópolis, 15-18 set. 1998, p. 524.
- NOTARE, M.S.M.A., CRUZ, F.A., RISO, B.G., ALVES, J.B.M., WESTPHALL, C.B. A security telecommunication management system (Painel n. 528). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul - Centro de Eventos de Florianópolis, 15-18 set. 1998, p. 524.
- NOTARE, M.S.M.A., CRUZ, F.A., ERNST, C., ALVES, J.B.M., WESTPHALL, C.B. Object-oriented modeling and design of an intelligent management system for ATM networks (Painel n. 529). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul - Centro de Eventos de Florianópolis, 15-18 set. 1998, p. 525.
- NOTARE, M.S.M.A., CRUZ, F.A., SOBRAL, J.B.M., ALVES, J.B.M., RISO, B.G., WESTPHALL, C.B. Distributed management in the security area for cloned mobile phones (Painel n. 530). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul - Centro de Eventos de Florianópolis, 15-18 set. 1998, p. 525.
- SCHNEIDER, M.L., COSTA, T.F.S., NOTARE, M.S.M.A., RISO, B.G. Um estudo LOTOS associado a redes neurais (Painel n. 532). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul – Centro de Eventos de Florianópolis, 15-18 set. 1998, p. 526.
- WESTPHALL, C.B., RISO, B.G., NOTARE, M.S.M.A., KORMANN, L.F., MARTINS, J.S.B., LORENA, P.S., PENNA, M.C.O.N. PLAGERE. Project: platforms for network management (Painel n. 534). In: VI SEMANA DA QUALIDADE E INOVAÇÃO TECNOLÓGICA 98 - VI SEMANA DA PESQUISA DA UFSC. *Anais...* CentroSul - Centro de Eventos de Florianópolis, 15-18 set. 1998, p. 527.
- SOUZA, F.P., LEITE, K.C., NOTARE, M.S.M.A., WESTPHALL, C.B. Gerência distribuída na área de segurança para redes de telecomunicações de alta velocidade. In: IX SEMAC98, CEC - Centro de Estudos da Computação. *Anais...* IBILCE-UNESP São José do Rio Preto, SP, 25 set. 1998.
- MATOS, A.V., NOTARE, M.S.M.A., RISO, B.G. Tratamento Formal à Segurança de Carga de Applets. In: WORKCOMP'98 - WORKSHOP DE COMPUTAÇÃO DO ITA. *Anais...* São José dos Campos, SP, 06-09 out. 1998.
- SOUZA, F.P., LEITE, K.C., NOTARE, M.S.M.A., WESTPHALL, C.B. Gerência distribuída na área de segurança para redes de telecomunicações. In: VIII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UFSC. *Anais...* UFSC, Pró-Reitoria de Pesquisa e Pós-Graduação, Departamento de Apoio à Pesquisa. Florianópolis, SC, 25-26 nov. 1998, p. 271.
- MILIOLI, C.F., BRANCO, W.C., NOTARE, M.S.M.A., SOBRAL, J.B.M., WESTPHALL, C.B., RISO, B.G. Sistema de segurança de telefones celulares clonados. In: VIII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UFSC. *Anais...* UFSC, Pró-Reitoria de Pesquisa e Pós-Graduação, Departamento de Apoio à Pesquisa. Florianópolis, SC, 25-26 nov. 1998, p. 270.

- CRUZ, F.A.S., NOTARE, M.S.M.A., RISO, B.G., ALVES, J.B.M., WESTPHALL, C.B. Gerência proativa de redes com o emprego de inteligência artificial. In: SIMP99 - SIMPÓSIO DE INFORMÁTICA DO PLANALTO MÉDIO. *Anais...* Universidade de Passo Fundo, RS, 14-16 abr. 1999.
- MATTOS, A.V., NOTARE, M.S.M.A., RISO, B.G., SOBRAL, J.B.M., WESTPHALL, C.B. Gerência de segurança em aplicações de banco de dados na Web. In: SIMP99 - SIMPÓSIO DE INFORMÁTICA DO PLANALTO MÉDIO. *Anais...* Universidade de Passo Fundo, RS, 14-16 abr. 1999.
- SOUZA, F.P., LEITE, K.C., NOTARE, M.S.M.A., WESTPHALL, C.B. Uma aplicação distribuída para redes de telecomunicações utilizando as tecnologias Web e CORBA. In: SIMP99 - SIMPÓSIO DE INFORMÁTICA DO PLANALTO MÉDIO. *Anais...* Universidade de Passo Fundo, RS, 14-16 abr. 1999.
- NOTARE, M.S.M.A., CRUZ, F.A., RISO, B.G., WESTPHALL, C.B. Sistema de segurança para telecomunicações contra clonagem e inadimplência. In: FENASOFT'99. Centro de Convenções Anhembi, São Paulo, SP, jun. 1999.
- NOTARE, M.S.M.A., CRUZ, F.A., RISO, B.G., WESTPHALL, C.B. Distributed security management system for mobile telecommunication. In: COMDEX'99. Centro de Convenções Anhembi. São Paulo SP, 16 ago. 1999.
- NOTARE, M.S.M.A., CRUZ, F.A.S., RISO, B.G., WESTPHALL, C.B. Security in telecommunications networks. In: SSI - SIMPÓSIO DE SEGURANÇA EM INFORMÁTICA. ITA - INSTITUTO TECNOLÓGICO DE AERONÁUTICA. *Anais...* São Paulo, 14-16 set. 1999.
- MATTOS, A.V., NOTARE, M.S.M.A., WESTPHALL, C.B. Segurança e flexibilidade em aplicações de banco de dados distribuídos. In: SSI - SIMPÓSIO DE SEGURANÇA EM INFORMÁTICA. ITA - INSTITUTO TECNOLÓGICO DE AERONÁUTICA. *Anais...* São Paulo, 14-16 set. 1999.
- NOTARE, M.S.M.A., BOUKERCHE, A., CRUZ, F.A.S., RISO, B.G., WESTPHALL, C.B. Projeto e implementação de um sistema de segurança distribuído para redes de telecomunicações. In: CONTTEIN'99/INFOTECH - PRÊMIO NACIONAL DE SOFTWARE PARA TELECOMUNICAÇÕES. Londrina, PR, 14-18 set. 1999.
- NOTARE, M.S.M.A., CRUZ, F.A., HERMIDA, A., ALVES, J.B., RISO, B.G., WESTPHALL, C.B. Segurança no combate a fraude de telefone celular clonado. (Painel 158). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* Florianópolis, 20-23 set. 1999, p. 151.
- NOTARE, M.S.M.A., HERMIDA, A., VALE, W., WESTPHALL, C.B. Sistema de segurança contra celulares clonados. (Painel 159). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* Florianópolis, 20-23 set. 1999, p. 150.
- MATOS, A.V. DE, RISO, B.G., NOTARE, M.S.M.A. Tratamento formal à segurança da carga de applets. (Painel 427). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* Florianópolis, 20-23 set. 1999, p. 427.
- NOTARE, M.S.M.A., CRUZ, F.A., RISO, B.G., WESTPHALL, C.B. Wireless communications: security management for cloned cellular phones. (Painel 429). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* Florianópolis, 20-23 set. 1999, p. 428.
- NOTARE, M.S.M.A., CRUZ, F.A., RISO, B.G., WESTPHALL, C.B. Security management against cloning mobile telephones. (Painel 432). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* Florianópolis, 20-23 set. 1999, p. 430.
- NOTARE, M.S.M.A., CRUZ, F.A., RISO, B.G., WESTPHALL, C.B. Security management against cloned cellular telephones. (Painel 431). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* Florianópolis, 20-23 set. 1999, p. 429.
- NOTARE, M.S.M.A., BOUKERCHE, A., WESTPHALL, C.B. A Secure management against cloned telephones. (Painel 430). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* Florianópolis, 20-23 set. 1999, p. 429.

- CRUZ, F. A DA S. , NOTARE, M.S.M.A., RISO, B. G. , WESTPHALL, C. B. Gerência proativa de redes com técnicas de inteligência artificial. (Painel 422). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* 20-23 set. 1999, p. 425.
- CRUZ, F. A DA S. , NOTARE, M.S.M.A., ALVES, J. B. M. , WESTPHALL, C.B. Gerência proativa de redes com o emprego de inteligência artificial. (Painel 421). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* Florianópolis, 20-23 set. 1999, p. 424.
- NOTARE, M.S.M.A., CRUZ, F. A. DA S. , RISO, B. G. , WESTPHALL, C.B. SSTCC - Sistema de segurança para telecomunicações contra clonagem e inadimplência. (Painel 433). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* Florianópolis, 20-23 set. 1999, p. 430.
- CRUZ, F.A., NOTARE, M.S.M.A., RISO, B. G. , WESTPHALL, C.B. Gerência proativa de redes com técnicas de inteligência artificial. (Painel 423). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* Florianópolis, 20-23 set. 1999, p. 425.
- WESTPHALL, C.B. , LEITE, K.C. , NOTARE, M.S.M.A., SOUZA, F. P. Uma aplicação distribuída para redes de telecomunicações utilizando tecnologias Web e CORBA. (Painel 426). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* Florianópolis, 20-23 set. 1999, p. 427.
- NOTARE, M.S.M.A., CRUZ, F.A., RISO, B. G. , WESTPHALL, C.B. Distributed security management system for mobile telecommunication (Painel 428). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* Florianópolis, 20-23 set. 1999, p. 428.
- WESTPHALL, C.B., LEITE, K.C. , NOTARE, M.S.M.A., SOUZA, F. P. Uma aplicação distribuída para redes de telecomunicações utilizando as tecnologias Web e CORBA. (Painel 425). In: VII SEMANA DA PESQUISA, UFSC. *Anais...* Florianópolis, 20-23 set. 1999, p. 426.
- MATOS, A.V., NOTARE, M.S.M.A., SOBRAL, J. B. M. , WESTPHALL, C.B. Gerência de segurança em aplicações de banco de dados na Web. (Painel 424). In: VII SEMANA DA PESQUISA, UFSC. *Anais ...* Florianópolis, 20-23 set. 1999, p. 426.
- HERMIDA, A.C., WESTPHALL, C.B., NOTARE, M.S.M.A., VALE, W., SPÍNDOLA, F.S. Sistema de segurança contra celulares clonados. In: IX SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UFSC. *Anais...* Florianópolis, 26-28 out. 1999, p.398.
- SOUZA, F.P., WESTPHALL, C.B., NOTARE, M.S.M.A., SOBRAL, J.B.M. Sistema de extrato telefônico via Web em Java com suporte CORBA. In: IX SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UFSC. *Anais...* Florianópolis, 26-28 out. 1999, p.399.
- VALLE, W., WESTPHALL, C.B., NOTARE, M.S.M.A., SPÍNDOLA, F.S., HERMIDA, A.C. SIPI - Sistema de identificação de prováveis inadimplentes. In: IX SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UFSC. *Anais...* Florianópolis, 26-28 out. 1999, p.401.

### **Relatório de Pesquisa**

- NOTARE, M.S.M.A at al. Formalização do Protótipo Final da Plataforma de Gerência Prática Telebrás. Projeto PLAGERE/ProTeM-CC-II/CNPq, Lista PLAGERE: 166, UFSC - Universidade Federal de Santa Catarina, Florianópolis, SC, 06/02/97.
- NOTARE, M.S.M.A at al. Protótipo Mínimo de um Sistema Distribuído para a Gerência de Segurança em Redes de Telecomunicações. Projeto Integrado CNPq/AUTOGERE - Automação da Gerência de Redes. Relatório de Pesquisa AUTOGERE 106. Atividades A, E, G, H e I. UFSC, 28/02/99.
- NOTARE, M.S.M.A at al. Implementação de um Sistema de Gerência Distribuído, Utilizando as Tecnologias Web, Java e CORBA, nas Áreas de Contabilização e Segurança. Projeto Integrado CNPq/AUTOGERE - Automação da Gerência de Redes. Relatório de Pesquisa AUTOGERE 108. Atividades A, E, H e I. UFSC, 28/02/99.
- NOTARE, M.S.M.A at al. Implementação de um Sistema Distribuído de Gerência de Segurança Contra Clonagem de Celulares, Utilizando a Tecnologia CORBA. Projeto Integrado CNPq/AUTOGERE - Automação da Gerência de Redes. Relatório de Pesquisa AUTOGERE 110. Atividades A, E, H e I. UFSC, 28/02/99.

### 5.3.2 Prêmios e Citações

**Prêmio.** Sucesu-SP'98 de Tecnologia da Informação. Primeira Fase. São Paulo, SP, 1999.

**Prêmio.** Co-Orientação do trabalho classificado entre os 10 primeiros no Concurso de Iniciação Científica do CNPq. Extrato Telefônico via Web com CORBA e Java. Aluna Fernanda Pereira de Souza. PUC-Rio, 21/07/1999.

**Indicação a Prêmio.** Prêmio Max Award. Fenasoft'99. Anhembi. São Paulo. 19-24/07/99.

**Prêmio Nacional de Software para Telecomunicações.** Terceiro Lugar, UNOPAR – Universidade Norte do Paraná, Londrina, PR, 16-18/09/1999.

**Citação em Site.** INRIA – Institut de Recherche en Informatique et en Automatique, Unité de Recherche INRIA Rhône-Alpes, Montbonnot, Saint Martin, Grenoble, France, <<http://www.inrialpes.fr/vasy/pub/CaseStudies/ProActNetworks.html>>.

**Citação em Relatório de Atividades.** INRIA – Institut de Recherche en Informatique et en Automatique, Unité de Recherche INRIA Rhône-Alpes, INRIA – Rapport d'activité scientifique 1997, Action Vasy-RA – Validation de Systèmes – Recherche et Applications, Citação em relações bilaterais “Relations bilatérales internationales – Amérique du Sud”, Montbonnot, Saint Martin, Grenoble, France, p.12.

**Citação em Anais.** O Estado Atual da Pesquisa e Desenvolvimento em Gerenciamento de Redes no Brasil por José Marcos Silva Nogueira. II SFBSID'97 – Segundo Seminário Franco-Brasileiro em Sistemas Informáticos Distribuídos – Arquiteturas Multimídias para as Telecomunicações, Fortaleza, CE, 03-07/11/97 Citação de publicações nas referências bibliográficas, p 81-98.

**Reportagem em Jornal.** UFSC melhora qualidade nas redes de telecomunicações – UFSC coordena projeto para melhorar a qualidade nas telecomunicações. Citação em reportagem do Jornal Indústria e Comércio, Florianópolis, SC, 19/11/97, p.10.

**Reportagem em Site.** Sistema de Segurança contra Clonagem de Celulares. Reportagem no Jornal on-line do Site da Universidade Aberta. <[www.unaberta.ufsc.br](http://www.unaberta.ufsc.br)>. Florianópolis, SC, 23-24/05/98.

**Reportagem em Rádio.** Sistema de Segurança contra Clonagem de Celulares. Reportagem na Rádio CBN Programa Universidade Aberta. Florianópolis, SC, (em fita cassete).

**Reportagem em TV.** Sistema de Segurança contra Clonagem de Celulares. Reportagem na TV Educativa. Universidade Aberta. Florianópolis, SC, 25/05/98 (em fita VHS).

**Reportagem em TV.** Sistema de Segurança contra Clonagem de Celulares. Reportagem na RBS TV. Programa Estúdio SC. Florianópolis, SC, 17/05/98 (em fita VHS).

**Reportagem em TV.** Sistema de Segurança contra Clonagem de Celulares. Reportagem na RBS TV. Programa Jornal do Almoço. Florianópolis, SC, 18/05/98 (em fita VHS).

**Reportagem em TV.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Canal Universitário. 18h, Florianópolis, SC, 12/02/99 (em fita VHS).

**Reportagem em TV.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Canal Universitário. Programa Minuto no Campus. Repórter Cleide. Florianópolis, SC, 13/02/99 (em fita VHS).

**Reportagem em Rádio.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem ao Vivo na Rádio CBN. Programa Notícia na Manhã. Apresentação Mário Motta. 11h, Florianópolis, SC, 15/02/99 (em fita cassete e em fita VHS).

**Reportagem em TV.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem na TV Cultura Programa Jornal da Cultura. Repórter Evandro. Florianópolis, SC, 21h30min.

**Reportagem em Jornal.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Jornal Gazeta Mercantil. Caderno Santa Catarina. Florianópolis, SC, 17/02/99, p. D-2.

**Reportagem em Jornal.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Jornal Diário Catarinense, Coluna Cacau Menezes. Florianópolis, SC, 19/02/99, p. 35.

**Reportagem em Jornal.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Jornal O Estado de Santa Catarina. Florianópolis, SC, 19/02/99, p. 10.

**Reportagem em Jornal.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Jornal Diário da Manhã. Chapecó, SC, 19/02/99.

**Reportagem em TV.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem ao Vivo na RBS TV. Programa Bom Dia Santa Catarina. Apresentação Ângelo Ribeiro. Florianópolis, SC, 6h45min 19/02/99.

**Reportagem em TV.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem na TV Record. Jornal da Record. Apresentação Nádia. Florianópolis, SC, 18h40min, 19/02/99.

**Reportagem em Jornal.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Jornal Diário Catarinense, Caderno Revista DC, Coluna Juliana Wosgraus. Florianópolis, SC, 20/02/99, p. 3.

**Reportagem em Site.** Sistema de Segurança Contra Clonagem de Celulares. Agecom On-line. Florianópolis, SC, 01/03/99.

**Reportagem em Jornal.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Jornal A Notícia Capital, Coluna Ricardinho Machado, Florianópolis, SC, 01/03/99, p. 7.

**Reportagem em Site.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem na Agência Estado, 17/03/99, <www.agemado.com.br/cet/caplic/caplic.htm>.

**Reportagem em Jornal.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Jornal Diário Catarinense, Caderno Principal, Florianópolis, SC, 18/03/99, p.37.

**Reportagem em Rádio.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem na Rádio CBN-Rio de Janeiro. Apresentação Marcos Gomes. 16h, Rio de Janeiro, RJ, 27/03/99.

**Reportagem em Rádio.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem ao Vivo na Rádio CBN-Florianópolis. Programa Notícia na Manhã. Apresentação Mário Motta. 11h30min, Florianópolis, SC, 04/99 (em fita VHS).

**Reportagem em Site.** Sistema de Segurança Contra Clonagem de Celulares. Artigo na Seção Ciência Tecnologia, Brasília, DF, 04/99, <www.radiobras.gov.br/c&t/artigos>.

**Reportagem em Jornal.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Jornal O Nacional, Caderno Principal, Coluna Empresas & Negócios, de Céia Giongo, 04-04/04/99, p.18.

**Reportagem em Site.** Sistema de Segurança Contra Clonagem de Celulares. Seção Ciência e Tecnologia, Brasília, DF, 09/04/99, <http://www.radiobras.gov.br/c&t.htm#9>.

**Reportagem em Jornal.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Jornal Diário Catarinense, Caderno Principal, Seção Visor, Florianópolis, SC, 15/04/99, pág.3.

**Reportagem em TV.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Canal Universitário (Canal 15 da Net). Programa Minuto no Campus. Repórter Giovana. Florianópolis, SC 20/04/99 (em fita VHS).

**Reportagem em Jornal.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Jornal Diário Catarinense, Caderno Principal, Seção Visor, Florianópolis, SC, 22/04/99, p.3.

**Reportagem em TV.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Canal Barriga Verde (TV Bandeirantes). Repórter Paulo Santhias. Florianópolis, SC, 23/04/99 (em fita VHS).

**Reportagem em TV.** Comércio Eletrônico. RBS TV (Globo). Programa Jornal da RBS. Repórter Edimilson Ortiz. 19h, 24/04/99.

**Reportagem em Jornal.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Jornal Diário Catarinense, Caderno Principal, Seção Visor, Florianópolis, SC, 30/04/99, p.3.

**Reportagem em Revista.** Sistema de Segurança Contra Clonagem de Celulares. Revista do Conselho Regional de Engenharia, Arquitetura e Agronomia de Santa Catarina - CREA-SC. Florianópolis, SC, Ano3 n.5, Maio 1999, p. 4.

**Reportagem em TV.** SBT TV. Programa SC 2000 – Escolha sua Profissão/Gente de Sucesso. Apresentação Rita. 14h, 07-11/06/99.

**Reportagem em TV.** TV Globo. Programa Jornal da Globo. Apresentação Lillian Vitte Fibe 14/07/99.

- Reportagem em Site.** Opinião – Clonagem de Celulares. Revista Inovar On-line. Florianópolis, SC, 07/99.
- Reportagem em Jornal.** Sistema de Segurança Contra Clonagem e Inadimplência. Reportagem no Jornal Diário Catarinense, Caderno Principal, Cacau Menezes/Romí de Liz – Interina, Florianópolis, SC, 20/07/99 p.43.
- Reportagem em Jornal.** Sistema de Segurança Contra Clonagem e Inadimplência. Reportagem no Jornal Diário Catarinense, Caderno Principal, Cacau Menezes/Romí de Liz – Interina, Florianópolis, SC, 21/07/99, p.51.
- Reportagem em Jornal.** Sistema de Segurança Contra Clonagem e Inadimplência/SETWeb. Reportagem no Jornal Diário Catarinense, Caderno Variedades, Juliana Wosgraus, Florianópolis, SC, 23/07/99, pág.3.
- Reportagem em Jornal.** Sistema de Segurança Contra Clonagem e Inadimplência. Reportagem no Jornal Diário Catarinense, Caderno Principal, Seção Visor. Florianópolis, SC, 24/07/99, pág.3.
- Reportagem em Jornal.** Sistema de Segurança para Telecomunicações Contra Clonagem e Inadimplência. Reportagem no Jornal Diário Catarinense, Caderno Variedades, Juliana Wosgraus, Florianópolis, SC 16/08/99, p.3.
- Reportagem em Jornal.** Sistema de Segurança para Telecomunicações Contra Clonagem e Inadimplência. Reportagem no Jornal Diário Catarinense, Caderno Variedades, Juliana Wosgraus, Florianópolis, SC 21/02/00, p.3.
- Reportagem em Jornal.** Sistema de Segurança para Telecomunicações Contra Clonagem e Inadimplência. Reportagem no Jornal Diário Catarinense, Caderno Variedades, Juliana Wosgraus, Florianópolis, SC 22/02/00, p.3.
- Reportagem em Jornal.** Sistema de Segurança Contra Clonagem e Inadimplência. Reportagem no Jornal Diário Catarinense, Caderno Principal, Seção Visor. Florianópolis, SC, 27/02/99, pág.3.
- Reportagem em Site.** Sistema de Extrato telefônico via Web. Seção Ciência, Tecnologia & Meio Ambiente, Brasília, DF, 25/02/00. [Www.radiobras.gov.br/c&t/2000/materia\\_250200\\_3.htm](http://www.radiobras.gov.br/c&t/2000/materia_250200_3.htm)
- Citação em Relatório de Atividades.** INRIA – Institut de Recherche en Informatique et en Automatique, Unité de Recherche INRIA Rhône-Alpes, INRIA – Rapport d’activité scientifique 1999, Action Vasy-RA – Validation de Systèmes – Recherche et Applications, Citação em Casos de Estudo, Montbonnot, Saint Martin, Grenoble, France, p.28.
- Citação em Site.** INRIA – Institut de Recherche en Informatique et en Automatique, Unité de Recherche INRIA Rhône-Alpes, Montbonnot, Saint Martin, Grenoble, France, <<http://www.inrialpes.fr/vasy/pub/CaseStudies/>>, 1999.
- Reportagem em Jornal.** Sistema de Segurança Contra Clonagem de Celulares. Reportagem no Jornal Diário Catarinense, Coluna Cacau Menezes. Florianópolis, SC, 07/03/00, p. 43.
- Reportagem em TV.** TV Barriga Verde. Programa Educação & Cidadania. Apresentação Maria Odete Olsen. 16/03/00.
- Reportagem em Site.** Sistema de Segurança contra Clonagem de Celulares. Reportagem no Jornal on-line do Site da Universidade Aberta. <[www.unaberta.ufsc.br](http://www.unaberta.ufsc.br)>. Florianópolis, SC, 31/03/00.
- Reportagem em Site.** Sistema de Segurança contra Clonagem de Celulares. Reportagem no Jornal on-line do Site da Universidade Aberta. <[www.unaberta.ufsc.br](http://www.unaberta.ufsc.br)>. Florianópolis, SC, 11/05/2000.
- Reportagem em TV.** TV Cultura/Unaberta. 11/05/2000.
- Reportagem em Revista.** UFSC 2000. (em andamento).





```

endproc
endproc
process BillCorbaManager[bill_notif,online_bill]:noexit:=
bill_notif;online_bill;BillCorbaManager(bill_notif,online_bill)
endproc
endproc
process SipiImpostor[check_owner]:noexit:=
hide impostor_notif in
ImpostorCorbaAgentsSet[impostor_notif] |[impostor_notif]|
ImpostorCorbaManager[impostor_notif,check_owner]
where
process ImpostorCorbaAgentsSet[impostor_notif]:noexit:=
ImpostorCorbaAgentsSet1[impostor_notif] |[impostor_corba_agents_set2[impostor_notif]
where
process ImpostorCorbaAgentsSet1[impostor_notif]:noexit:=
hide user_pattern1,impostor_pattern1,online_call1 in
(UsersPatternsFile1[user_pattern1] |[impostorsPatternsFile1[impostor_pattern1]
|[OnlineCallsFile1[online_call1]]
|[user_pattern1,impostor_pattern1,online_call1]|
ImpostorCorbaAgent1[user_pattern1,online_call1,impostor_pattern1,impostor_notif]
where
process UsersPatternsFile1[user_pattern1]:noexit:=
user_pattern1;UsersPatternsFile1[user_pattern1]
endproc
process ImpostorsPatternsFile1[user_pattern1]:noexit:=
user_pattern1; ImpostorsPatternsFile1[user_pattern1]
endproc
process OnlineCallsFile1[online_call1]:noexit:=
online_call1;OnlineCallsFile1[online_call1]
endproc
process ImpostorCorbaAgent1
[user_pattern1,online_call1,impostor_pattern1,impostor_notif]:noexit:=
online_call1;user_pattern1;impostor_pattern1;
(i;ImpostorCorbaAgent1[user_pattern1,online_call1,impostor_pattern1,impostor_notif]
|[i;impostor_notif;
ImpostorCorbaAgent1[user_pattern1,online_call1,impostor_pattern1,impostor_notif])
endproc
endproc
process ImpostorCorbaAgentsSet2[impostor_notif]:noexit:=
hide user_pattern2,impostor_pattern2,online_call2 in
(UsersPatternsFile2[user_pattern2] |[impostorsPatternsFile2[impostor_pattern2]
|[OnlineCallsFile2[online_call2]] |[user_pattern2,impostor_pattern2,online_call2]|
ImpostorCorbaAgent2[user_pattern2,impostor_pattern2,online_call2,impostor_notif]
where
process UsersPatternsFile2[user_pattern2]:noexit:=
user_pattern2;UsersPatternsFile2[user_pattern2]
endproc
process ImpostorsPatternsFile2[impostor_pattern2]:noexit:=
impostor_pattern2;ImpostorsPatternsFile2[impostor_pattern2]
endproc
process OnlineCallsFile2[online_call2]:noexit:=
online_call2;OnlineCallsFile2[online_call2]
endproc
process ImpostorCorbaAgent2[user_pattern2,impostor_pattern2,online_call2,impostor_notif]:noexit:=
online_call2;user_pattern2;impostor_pattern2;
(i;ImpostorCorbaAgent2[online_call2,user_pattern2,impostor_pattern2,impostor_notif]
|[i;impostor_notif;ImpostorCorbaAgent2[online_call2,user_pattern2,impostor_pattern2,impostor_notif])
endproc
endproc
endproc
process ImpostorCorbaManager[impostor_notif,check_owner]:noexit:=
impostor_notif;
(i;check_owner;ImpostorCorbaManager[impostor_notif,check_owner]
|[i;ImpostorCorbaManager[impostor_notif,check_owner]
endproc
endproc
endspec

```

## 1.2 Validação

SstccService.aut	(5, 1, 23)
	(5, 1, 25)
des (0, 6, 3)	(5, 1, 26)
(0, MAIL_ALARM, 1)	(6, 1, 12)
(0, ONLINE_BILL, 0)	(6, 1, 17)
(0, CHECK_OWNER, 0)	(6, 1, 21)
(1, 1, 2)	(6, 1, 24)
(1, 1, 0)	(6, 1, 26)
(2, PHONE_ALARM, 0)	(6, 1, 27)
	(7, 1, 28)
	(7, 1, 29)
SstccProtocol.aut	(7, 1, 30)
	(7, 1, 31)
des (0, 1645926, 194401)	(7, 1, 32)
(0, 1, 1)	(7, 1, 33)
(0, 1, 2)	(7, 1, 34)
(0, 1, 3)	
(0, 1, 4)	
(0, 1, 5)	(194395, ONLINE_BILL, 194029)
(0, 1, 6)	(194395, CHECK_OWNER, 194256)
(1, 1, 7)	(194395, PHONE_ALARM, 193364)
(1, 1, 8)	(194395, 1, 193364)
(1, 1, 9)	(194395, 1, 194400)
(1, 1, 10)	(194396, ONLINE_BILL, 194030)
(1, 1, 11)	(194396, CHECK_OWNER, 194265)
(1, 1, 12)	(194396, PHONE_ALARM, 193365)
(2, 1, 8)	(194396, 1, 193365)
(2, 1, 13)	(194396, 1, 194400)
(2, 1, 14)	(194397, ONLINE_BILL, 194063)
(2, 1, 15)	(194397, CHECK_OWNER, 194280)
(2, 1, 16)	(194397, PHONE_ALARM, 193366)
(2, 1, 17)	(194397, 1, 194300)
(3, 1, 9)	(194397, 1, 194400)
(3, 1, 14)	(194397, 1, 193366)
(3, 1, 18)	(194398, ONLINE_BILL, 194116)
(3, 1, 19)	(194398, CHECK_OWNER, 194301)
(3, 1, 20)	(194398, PHONE_ALARM, 193367)
(3, 1, 21)	(194398, 1, 194328)
(4, 1, 10)	(194398, 1, 194400)
(4, 1, 15)	(194398, 1, 193367)
(4, 1, 19)	(194399, ONLINE_BILL, 194197)
(4, 1, 22)	(194399, CHECK_OWNER, 194329)
(4, 1, 23)	(194400, MAIL_ALARM, 194400)
(4, 1, 24)	(194400, ONLINE_BILL, 194279)
(5, 1, 11)	(194400, CHECK_OWNER, 194357)
(5, 1, 16)	(194400, PHONE_ALARM, 193968)
(5, 1, 20)	(194400, 1, 193968)

## Anexo 2 – Redes Neurais Artificiais

### 2.1 Kohonen

```

% implementacao do algoritmo de kohonen
clf;
figure(gcf)
colordef(gcf, 'none')
setfsize(300,250);
techo on
clc

V1 = [
    0.9
    0.7
   -0.5
];

V2 = [
   -0.5
    0.5
    0.8
];

%V1 e v2 sao os vetores de entrada da RN -> cada ponto representaria uma ligacao do arquivo
%da amostra.
Q = 200;
P = [randn(3,Q)*0.2 + V1*ones(1,Q), randn(3,Q)*0.2 + V2*ones(1,Q)];
P = normc(P);

pause % Pressione qualquer tecla p/ ver os dados
clc

% Aqui os vetores de entrada normalizados são plotados na
% superfície de uma esfera.

hold on
[x,y,z] = sphere(10);
h = surf(x*0.9,y*0.9,z*0.9,x.^2.*y.^2.*z.^2);
set(h, 'facecolor', 'interp')
plot3(P(1,:) * 0.99, P(2,:) * 0.99, P(3,:) * 0.99, 'b', 'markersize', 10)
view([160 -20])
colormap(cool)
title('Treinamento da rede neural')

% O objetivo desta demonstração é fazer o neurônio de Kohonen aprender a reconhecer
% o vetor protótipo no meio dos dados.

clc

%W = randnr(4,3); % 1 neuron with 3 inputs.
%W2=W
W = [
   -0.7862    0.2269    0.1325;
   -0.8833    0.1032    0.4572;
    0.7714   -0.3990   -0.4956;
    0.2525   -0.5066   -0.8244;
];

b = -0.1*ones(4,1);

plot3(W(1,1),W(1,2),W(1,3), 'w+', W(2,1),W(2,2),W(2,3), 'w+', W(3,1),W(3,2),W(3,3), 'w+', W(4,1),W(4,2),W(4,3), 'w+', W(1,1),W(1,2),W(1,3), 'r.',
W(2,1),W(2,2),W(2,3), 'r.', W(3,1),W(3,2),W(3,3), 'r.', W(4,1),W(4,2),W(4,3), 'r.', 'markersize', 10)

pause;
clc
hardlim(W*p,b)
pause;

lr = 0.05; % Learning rate.
%loop p/ atualização dos pesos.
for q = 1:Q^2
    p = P(:,q); % Pick qth input vector.
    a = hardlim(W*p,b); % Calculate output.
    dW = learnk(W,p,a,lr); % Apply learning rule.
    W = W + dW; % New weights.

plot3(W(1,1),W(1,2),W(1,3), 'r.', W(2,1),W(2,2),W(2,3), 'r.', W(3,1),W(3,2),W(3,3), 'r.', W(4,1),W(4,2),W(4,3), 'r.', 'markersize', 10, 'erasemode', 'none');

%W
end
plot3(W(1,1),W(1,2),W(1,3), 'w+', W(2,1),W(2,2),W(2,3), 'w+', W(3,1),W(3,2),W(3,3), 'w+', W(4,1),W(4,2),W(4,3), 'w+', 'markersize', 10, 'erasemode', 'none')
W = normr(W);
hardlim(W*p,b)
pause;

%hardlim(W*[0.7;0.5;-0.3],b) - V1, p/ colocar na apres. depois do treinam. no teste
%hardlim(W*[-0.4;0.8;0.2],b) - V2, p/ colocar na apres. depois do treinam. no teste
disp('Fim do treinamento')

W + dW; % New weights.

plot3(W(1,1),W(1,2),W(1,3), 'r.', W(2,1),W(2,2),W(2,3), 'r.', W(3,1),W(3,2),W(3,3), 'r.', W(4,1),W(4,2),W(4,3), 'r.', 'markersize', 10, 'erasemode', 'none');

%W
end
p = P(:,q); % Pick qth input vector.
a = hardlim(W*p,b); % Calculate output.
dW = learnk(W,p,a,lr); % Apply learning rule.
W = W + dW; % New weights.

plot3(W(1,1),W(1,2),W(1,3), 'r.', W(2,1),W(2,2),W(2,3), 'r.', W(3,1),W(3,2),W(3,3), 'r.', W(4,1),W(4,2),W(4,3), 'r.', 'markersize', 10, 'erasemode', 'none');

%W
end
p = P(:,q); % Pick qth input vector.

```

```

a = hardlim(W*p,b); % Calculate output.
dW = learnk(W,p,a,lr); % Apply learning rule.
W = W + dW; % New weights.

plot3(W(1,1),W(1,2),W(1,3), 'r.',W(2,1),W(2,2),W(2,3), 'r.',W(3,1),W(3,2),W(3,3), 'r.',W(4,1),W(4,2),W(4,3), 'r.', 'markersize',10, 'erasemode', 'none');
%W
end
p = P(:,q); % Pick qth input vector.
a = hardlim(W*p,b); % Calculate output.
dW = learnk(W,p,a,lr); % Apply learning rule.
W = W + dW; % New weights.

plot3(W(1,1),W(1,2),W(1,3), 'r.',W(2,1),W(2,2),W(2,3), 'r.',W(3,1),W(3,2),W(3,3), 'r.',W(4,1),W(4,2),W(4,3), 'r.', 'markersize',10, 'erasemode', 'none');
%W
end
p = P(:,q); % Pick qth input vector.
a = hardlim(W*p,b); % Calculate output.
dW = learnk(W,p,a,lr); % Apply learning rule.
W = W + dW; % New weights.

plot3(W(1,1),W(1,2),W(1,3), 'r.',W(2,1),W(2,2),W(2,3), 'r.',W(3,1),W(3,2),W(3,3), 'r.',W(4,1),W(4,2),W(4,3), 'r.', 'markersize',10, 'erasemode', 'none');
%W
end
p = P(:,q); % Pick qth input vector.
a = hardlim(W*p,b); % Calculate output.
dW = learnk(W,p,a,lr); % Apply learning rule.
W = W + dW; % New weights.

plot3(W(1,1),W(1,2),W(1,3), 'r.',W(2,1),W(2,2),W(2,3), 'r.',W(3,1),W(3,2),W(3,3), 'r.',W(4,1),W(4,2),W(4,3), 'r.', 'markersize',10, 'erasemode', 'none');
%W
end
p = P(:,q); % Pick qth input vector.
a = hardlim(W*p,b); % Calculate output.
dW = learnk(W,p,a,lr); % Apply learning rule.
W = W + dW; % New weights.

plot3(W(1,1),W(1,2),W(1,3), 'r.',W(2,1),W(2,2),W(2,3), 'r.',W(3,1),W(3,2),W(3,3), 'r.',W(4,1),W(4,2),W(4,3), 'r.', 'markersize',10, 'erasemode', 'none');
%W
end
p = P(:,q); % Pick qth input vector.
a = hardlim(W*p,b); % Calculate output.
dW = learnk(W,p,a,lr); % Apply learning rule.
W = W + dW; % New weights.

plot3(W(1,1),W(1,2),W(1,3), 'r.',W(2,1),W(2,2),W(2,3), 'r.',W(3,1),W(3,2),W(3,3), 'r.',W(4,1),W(4,2),W(4,3), 'r.', 'markersize',10, 'erasemode', 'none');
%W
end
p = P(:,q); % Pick qth input vector.
a = hardlim(W*p,b); % Calculate output.
dW = learnk(W,p,a,lr); % Apply learning rule.
W = W + dW; % New weights.

plot3(W(1,1),W(1,2),W(1,3), 'r.',W(2,1),W(2,2),W(2,3), 'r.',W(3,1),W(3,2),W(3,3), 'r.',W(4,1),W(4,2),W(4,3), 'r.', 'markersize',10, 'erasemode', 'none');
%W
end
p = P(:,q); % Pick qth input vector.
a = hardlim(W*p,b); % Calculate output.
dW = learnk(W,p,a,lr); % Apply learning rule.
W = W + dW; % New weights.

plot3(W(1,1),W(1,2),W(1,3), 'r.',W(2,1),W(2,2),W(2,3), 'r.',W(3,1),W(3,2),W(3,3), 'r.',W(4,1),W(4,2),W(4,3), 'r.', 'markersize',10, 'erasemode', 'none');
%W
end
p = P(:,q); % Pick qth input vector.
a = hardlim(W*p,b); % Calculate output.
dW = learnk(W,p,a,lr); % Apply learning rule.
W = W + dW; % New weights.

plot3(W(1,1),W(1,2),W(1,3), 'w+',W(2,1),W(2,2),W(2,3), 'w+',W(3,1),W(3,2),W(3,3), 'w+',W(4,1),W(4,2),W(4,3), 'w+', 'markersize',10, 'erasemode', 'none')
W = norm(W);
hardlim(W*P,b)
ans =

```

Columns 1 through 12

```

0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1

```

Columns 13 through 24

```

0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1

```

Columns 25 through 36

```

0 0 0 0 0 0 0 0 0 0 0 0

```

0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 37 through 48

0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 49 through 60

0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 61 through 72

0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 73 through 84

0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 85 through 96

0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 97 through 108

0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 109 through 120

0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 121 through 132

0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 133 through 144

0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 145 through 156

0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 157 through 168

0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 169 through 180

0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 181 through 192

0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0  
 1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1

Columns 193 through 204

0 0 0 0 0 0 0 0 1 1 1 1  
 0 0 0 0 0 0 0 0 1 1 1 1  
 1 1 1 1 1 1 1 1 0 0 0 0  
 1 1 1 1 1 1 1 1 0 0 0 0

Columns 205 through 216

1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1  
 0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0

Columns 217 through 228

1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1  
 0 0 1 0 0 0 0 0 0 0 0 0  
 0 0 1 0 0 0 0 0 0 0 0 0

Columns 229 through 240

1 1 1 1 1 1 1 1 1 1 1 1  
 1 1 1 1 1 1 1 1 1 1 1 1  
 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0

Columns 241 through 252

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0
    
```

Columns 253 through 264

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
    
```

Columns 265 through 276

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0
    
```

Columns 277 through 288

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0
    
```

Columns 289 through 300

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
    
```

Columns 301 through 312

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
    
```

Columns 313 through 324

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
    
```

Columns 325 through 336

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 1 0
    
```

Columns 337 through 348

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 1 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0
    
```

Columns 349 through 360

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
    
```

Columns 361 through 372

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 1 0 0 1 0 0 0
0 0 0 0 0 1 0 0 1 0 0 0
    
```

Columns 373 through 384

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
    
```

Columns 385 through 396

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
    
```

Columns 397 through 400

```

1 1 1 1
1 1 1 1
0 0 0 0
0 0 0 0
    
```

```

pause;
%hardlim(W*[0.7;0.5;-0.3],b) - V1, p/ colocar na apres. depois do treinam. no teste
%hardlim(W*[-0.4;0.8;0.2],b) - V2, p/ colocar na apres. depois do treinam. no teste
disp('Fim do treinamento')
Fim do treinamento
»
    
```

```

» hardlim(W*[0.7;0.5;-0.3])
ans =
    
```

```

0
0
1
1
    
```

## 2.2 RBF - Função de Base Radial

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%rbf3.m - Implementacao RBF com dados de Copenhagen
%Algoritmo da RBF para classificar com 3 features
clear;
echo on;
t=cputime;
load a3.dat;
[cent,varia] = kmeans(a3(1:1000,:),110); % retorna os centroides e a variancia
% em cada centroide
varial=4*spread(cent,1); % calcula a variancia entre os centros
out=gauss(cent,a3,varial); % out sao os neuronios da camada escondida
% apos usar a funcao de Gauss
clear a3;
%treina.m - treina usando a pseudo inversa
load a7.dat;
gm=inv(out'*out)*out';
W=(gm*a7)'; % determinacao dos pesos da
% camada de saida
clear gm a7;
%testa.m - testa um conjunto de dados
load b3.dat;
S=gauss(cent,b3,varial); % S neuronios na camada escondida
% para os dados de teste
clear b3;
load b7.dat;
A=purelin(W*S)'; % A e' a matriz com as saidas da rede
[num,error]=compara(b7,A); % B7 e' a matriz com as saidas desejadas
% a funcao compara traz a porcentagem de
% erro e o numero de ligacoes erradas
clear b7 S A ans num;
cputime-t, clear ans t;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%kmeans.m
%Algoritmo K-means
function [c,r]=kmeans(x,nc)
[N,p]=size(x);
c=x(1:nc,:); % Init. clusters from first nc data pattern.
tmp=zeros(N,nc);
Change=1; iter=0;
np=zeros(1,N);
while Change == 1,
cp = zeros(nc,p);
cndx=zeros(nc,1);
for i=1:nc,
tmp(:,i) = ((x-ones(N,1)*c(i,:)).^2)*ones(p,1);
end;
[ymn,n]=min(tmp');
for i=1:N,
k=n(i);
cp(k,:)=cp(k,:)+x(i,:);
cndx(k) = cndx(k)+1;
end;
for i=1:nc,
c(i,:)=cp(i,:)/(cndx(i)+eps);
end;
if n == np,
[nm,i]=min(cndx);
if nm==0,
jj = round(461*rand(1))+1;
c(i,:)=x(jj,:);
else,
Change=0;
end;
else,
np=n;
iter = iter +1;
end;
nm,
% Compute the cluster spread.
r=zeros(nc,1);
for i=1:N,
k=n(i);
st=x(i,:)-c(k,:);
r(k)=r(k)+st*st';
end;
r=r./cndx;
c: % centers
r: % spread of centers

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%tgauss.m
%Função de Gauss - transferência da primeira camada para a camada escondida
function out=gauss(c,x,cvar)
%USAGE out = gauss(center,datin,desvio)
% out - Matriz com (mxn)
% onde : m - numero de inputs
% n - numero de clusters
% center - Matriz dos centros calculado pelo Kmean
% datain - Matriz com inputs
% desvio - Vetor com o desvio padrao dos centros
[nc,p] = size(c);
[N,p]=size(x);
% To calculate Gauss function
out=zeros(N,nc);
for i=1:nc
out(i,i) = (exp(-0.5*((x - ones(N,1)*c(i,:)).^2))*ones(p,1)/cvar(i));
end
function [num,error]=compara(A,B)
%USAGE
% [num,error]=compara(A,B)
% A - Matriz com targets
% B - Matriz com saidas da rede
[l,c]=size(A);
[Y1,I1]=max(A');
[Y2,I2]=max(B');
cont=0;
for i=1:l
if (I1(i) == I2(i) )
cont=cont+1;
end
end
num=cont;
error=(l-num)/l*100;

```









```

out(:,i) = (exp(-0.5*((x - ones(N,1)*c(i,:)).^2))*ones(p,1)/cvar(i));
end
% algoritmo K-means e p-nearest neighbor para determinar a variância entre os centros,
% função de Gauss para a transferência da camada de entrada para a camada escondida
% e função compata, utilizada para comparar a saída da rede com a saída desejada
%Algoritmo K-means
function [c,r]=kmeans(x,nc)
%USAGE (center,spread)=kmeanr2(data,number centers)
% Inputs : x = (N x p) : feature vectors.
% nc= number of clusters.
% Outputs : c = (nc x p) cluster vectors.
% r = c(nc,1) cluster spread.
%
% Page 247 of "Fundamentals of speech recognition."
%
[N,p]=size(x);
% xmax=max(max(abs(x)));
% x=x/xmax; % Normalize the data set.
% c=rand(nc,p); % Init. clusters randomly.
% c=x(1:nc,:); % Init. clusters from first nc data pattern.

tmp=zeros(N,nc);
Change=1; iter=0;
np=zeros(1,N);
%xvar=var(x);
%R=cov(x); [V,D]=eig(R); D=diag(D)'; xvar=D;
%xvar=ones(N,1)*xvar;
while Change == 1,
cp = zeros(nc,p);
cndx=zeros(nc,1);
for i=1:nc,
%tmp(:,i) = (((x-ones(N,1)*c(i,:)).^2)./xvar)*ones(p,1);
tmp(:,i) = ((x-ones(N,1)*c(i,:)).^2)*ones(p,1);
end;
[ymin,n]=min(tmp');
for i=1:N,
k=n(i);
cp(k,:)=cp(k,:)+x(i,:);
cndx(k) = cndx(k)+1;
end;
for i=1:nc,
c(i,:)=cp(i,:)/(cndx(i)+eps);
end;
if n == np,
[nm,i]=min(cndx);
if nm==0,
jj = round(461*rand(1))+1;
c(i,:)=x(jj,:);
else,
Change=0;
end;
else,
np=n;
end;
iter = iter +1;
end;
nm,
% Compute the cluster spread.
r=zeros(nc,1);
for i=1:N,
k=n(i);
st=x(i,:)-c(k,:);
r(k)=r(k)+st*st';
end;

```

```

r=r./cndx;
c: % centers
s: % spread of centers
% RBF para classificar com 4 features
% Algoritmo da RBF
clear;
echo on;
t=cputime;
load liga.dat;
[cent,varia] = kmeanr2(liga(1:2400,:),80); % retorna os centroides e a variancia
% em cada centroide
%varial=4*varia; % ampliacao do raio de cada centroide
varial=4*spread(cent,1); % calcula a variancia entre os centros
%varia=0.4*varial+0.4*varia2; % ampliacao do raio de cada centroide
out=gauss(cent,liga,varial); % out sao os neuronios da camada escondida
% apos usar a funcao de Gauss
clear liga;
%treina.m - treina usando a pseudo inversa
load pada.dat;
gm=inv(out'*out)*out';
W=(gm*pada)'; % determinacao dos pesos da
% camada de saida
clear gm pada;
%testa.m - testa um conjunto de dados
load ligb.dat;
S=gauss(cent,ligb,varial); % S neuronios na camada escondida
% para os dados de teste
clear ligb;
load padb.dat;
A=purelin(W*S)'; % A e' a matriz com as saidas da rede
[num,error]=compara(padb,A); % padb e' a matriz com as saidas desejadas
hold on
[x,y,z] = sphere(10);
h = surf(x*0.9,y*0.9,z*0.9,x.^2.*y.^2.*z.^2);
set(h,'facecolor','interp')
%plot3(A(:,1)*0.99,A(:,2)*0.99,A(:,4)*0.99,A(:,4)*0.99,A(:,5)*0.99,A(:,6)*0.99,A(:,10)*0.99,'.b','markersize',10)
plot3(A(:,1)*0.99,A(:,2)*0.99,A(:,4)*0.99,'.b','markersize',10)

view([170 -30])
colormap(cool)
title('Teste - Normalized input vectors')
% a funcao compara traz a porcentagem de
% erro e o numero de cromossomos errados
%clear padb S A ans num;
%tputime-t, clear ans t;
clear padb;

cputime-t, clear ans t;
function [num,error]=compara(A,B)
%USAGE
% [num,error]=compara(A,B)
% A - Matriz com targets
% B - Matriz com saidas da rede
[l,c]=size(A);
[Y1,I1]=max(A');
[Y2,I2]=max(B');
cont=0;
for i=1:l
if (I1(i) == I2(i) )
cont=cont+1;
end
end
num=cont;
error=(l-num)/l*100;

```

liga.dat				900230103	134517	79	20899
9586380	14803	4	20899	800230202	131234	125	20899
2233477	428	17	20899	800230206	121117	425	20899
99795722	947	34	20899	800990001	154012	2	20899
800230101	17	79	20899	519789980	141713	14	20899
800230201	11	125	20899	519738919	142857	36	20899
800230205	13117	425	20899	9855544	175212	111	20899
9586381	84803	4	20899	4861238	144321	126	20899
2233478	81428	17	20899	4871138	141809	551	20899
99795723	91247	34	20899	800542111	180141	3	20899
800230103	104517	79	20899	512249200	180548	17	20899
800230202	111234	125	20899	9810313	180317	42	20899
800230206	111117	425	20899	9655853	180649	110	20899
9586381	124803	4	20899	3331010	180849	142	20899
2233478	121428	17	20899	2193894	180544	609	20899
99795723	131247	34	20899	9586380	14803	4	10899

2233477	428	17	10899	9721717	2029	27	10899
99795722	947	34	10899	102	325	103	10899
800230101	17	79	10899	519753329	130	201	10899
800230201	11	125	10899	519753322	1530	501	10899
800230205	13117	425	10899	519753320	31130	2	10899
9586381	84803	4	10899	2275101	92308	15	10899
2233478	81428	17	10899	9721718	92029	27	10899
99795723	91247	34	10899	190	100825	103	10899
800230103	104517	79	10899	519753320	111230	201	10899
800230202	111234	125	10899	519753323	111530	501	10899
800230206	111117	425	10899	519753321	121130	2	10899
9586381	124803	4	10899	2275102	123308	15	10899
2233478	121428	17	10899	9721719	132029	27	10899
99795723	131247	34	10899	130	130825	103	10899
800230103	134517	79	10899	519753321	131230	201	10899
800230202	131234	125	10899	519753324	121530	501	10899
800230206	121117	425	10899	800990002	154012	2	10899
800990001	154012	2	10899	519799981	141713	14	10899
519789980	141713	14	10899	519738910	142857	36	10899
519738919	142857	36	10899	9855545	175212	111	10899
9855544	175212	111	10899	4861239	144321	126	10899
4861238	144321	126	10899	4871139	141809	551	10899
4871138	141809	551	10899	800542112	180141	6	10899
800542111	180141	6	10899	51224921	180548	19	10899
512249200	180548	19	10899	9810314	180317	34	10899
9810313	180317	34	10899	9655854	180649	90	10899
9655853	180649	90	10899	3331011	180849	132	10899
3331010	180849	132	10899	2193895	180544	909	10899
2193894	180544	909	10899	482281730	612	10	10899
482281739	12	10	10899	473314326	94412	30	10899
473314325	84412	30	10899	219921213	131010	25	36404
219921212	121010	25	36404	119724544	150906	18	36404
119724543	140906	14	36404	543141905	190204	19	36404
543141905	180204	16	36404	482281731	3319	25	20899
482281730	19	29	20899	47331437	134612	24	20899
47331436	124412	28	20899	219921214	231010	21	36405
219921213	211010	22	20899	472282345	81532	25	20899
482282345	81532	29	20899	219804652	153023	21	20899
489804652	143023	20	20899	482281738	12	33	10899
482281739	12	33	10899	473314326	84412	40	10899
473314325	84412	40	10899	219921213	121010	50	36404
219921212	121010	50	36404	119724544	140906	55	36404
119724543	140906	55	36404	543141906	180204	46	36404
543141905	180204	46	36404	482281731	19	119	20899
482281730	19	119	20899	473314376	124412	48	20899
47331436	124412	48	20899	219921214	211010	222	20899
219921213	211010	222	20899	482282344	81532	59	10899
482282345	81532	59	20899	489804653	143023	40	20899
489804652	143023	40	20899	900139854	12	15	10899
900339854	12	13	10899	900121212	94412	12	10899
900221212	84412	10	10899	900214545	121010	11	36404
900114545	121010	10	36404	900219999	140906	19	36404
900119999	140906	15	36404	900318877	180204	11	36404
900218877	180204	16	36404	900312020	19	9	20899
900212020	19	19	20899	900581010	124412	18	20899
900481010	124412	8	20899	900409099	211010	2	20899
900540909	211010	22	20899	900114141	81532	19	20899
900414141	81532	9	20899	900981010	143023	20	20899
900991010	143023	30	20899	900339855	12	73	10899
900339854	12	133	10899	900221215	84412	90	10899
900221212	84412	100	10899	900114546	121010	160	36404
900114545	121010	60	36404	900119995	140906	105	36404
900119999	140906	115	36404	900218875	180204	106	36404
900218877	180204	116	36404	900212025	19	160	20899
900212020	19	190	20899	900481015	124412	97	20899
900481010	124412	80	20899	900540905	211010	212	20899
900540909	211010	212	20899	900414145	81532	900	20899
900414141	81532	90	20899	900991015	143023	350	20899
900991010	143023	300	20899	19723151072	12	15	10899
19723151072	12	10	10899	443157953	94412	27	10899
443157953	84412	30	10899	338763452929		131010	22
338763452929		25	36404	19763152354	142906	11	36404
19763152354	140906	14	36404	19733157878	182204	13	36404
19733157878	180204	16	36404	19623148765	319	21	20899
19623148765	19	29	20899	1876316223	123412	20	20899
18763162323	124412	28	20899	229909954646		221010	28
229909954646		22	20899	19722151082	81532	19	20899
19723151082	81532	29	20899	448722943545		143023	90
448722143545		20	20899	19723191072	12	69	10899
19723151072	12	60	10899	443159953	84412	95	10899
443157953	84412	90	10899	338763452929		121010	79
338763452929		69	36404	19763952354	140906	55	36404
19763152354	140906	59	36404	19733957878	180204	91	36404
19733157878	180204	93	36404	19623948765	19	124	20899
19623148765	19	120	20899	18763962323	124412	118	20899
18763162323	124412	128	20899	229909954646		211010	190
229909954646		150	20899	19993151082	91532	329	20899
19793151082	81532	229	20899	449962143545		153023	420
449962143545		320	20899	518002258	5248	3	20899
519753329	1130	2	20899	9718420	3914	20	20899
2275100	2308	15	20899	519792558	1605	36	20899
9721717	2029	27	20899	513641474	4427	109	20899
102	825	103	20899	513641433	4427	160	20899
519753329	130	201	20899	513641434	4629	960	20899
519753322	1530	501	20899	518002259	85248	3	20899
519753320	81130	2	20899	9718421	83914	20	20899
2275101	92308	15	20899	519792559	91605	36	20899
9721718	92029	27	20899	513641475	94427	109	20899
190	100825	103	20899	513641434	104427	160	20899
519753320	111230	201	20899	513641435	114629	960	20899
519753323	111530	501	20899	518002250	125248	3	20899
519753321	121130	2	20899	9718422	123914	20	20899
2275102	123308	15	20899	519792550	121605	36	20899
9721719	132029	27	20899	513641476	134427	109	20899
130	130825	103	20899	513641435	134427	160	20899
519753321	131230	201	20899	513641436	144629	960	20899
519753324	121530	501	20899	800990003	154012	2	20899
2275102	123308	15	20899	519789982	141713	14	20899
9721719	132029	27	20899	519738911	142857	36	20899
130	130825	103	20899	9855546	175212	111	20899
519753321	131230	201	20899	4861230	144321	126	20899
519753324	121530	501	20899	4871130	141809	551	20899
800990002	154012	2	20899	800542113	180141	6	20899
519789981	141713	14	20899	51224922	190548	19	20899
519738910	142857	36	20899	9810315	200317	34	20899
9855545	175212	111	20899	9655855	210649	90	20899
4861239	144321	126	20899	3331012	220849	132	20899
4871139	141809	551	20899	2193896	230544	909	20899
800542112	180141	6	20899	518002258	5248	3	10899
51224921	180548	19	20899	9718420	3914	20	10899
9810314	180317	34	20899	519792558	1605	36	10899
9655854	180649	90	20899				
3331011	180849	132	20899				
2193895	180544	909	20899				
519753329	1130	2	10899				
2275100	2308	15	10899				

513641474	4427	109	10899	99795983	413	218	10899
513641433	4427	160	10899	99795583	613	718	10899
513641434	4629	960	10899	517151032	110109	7	10899
518002259	85248	3	10899	9718132	83406	18	10899
9718421	83914	20	10899	518002497	95332	30	10899
519792559	91605	36	10899	99795994	85413	118	10899
513641475	94427	109	10899	99795984	101413	218	10899
513641434	104427	160	10899	99795584	111613	718	10899
513641435	114629	960	10899	517151033	120109	7	10899
518002250	125248	3	10899	9718133	123406	18	10899
9718422	123914	20	10899	518002498	125332	30	10899
519792550	121605	36	10899	99795995	134413	118	10899
513641476	134427	109	10899	99795985	131413	218	10899
513641435	134427	160	10899	99795585	131613	718	10899
513641436	144629	960	10899	800990002	154012	2	10899
800990003	154012	2	10899	519789981	141713	14	10899
519789982	141713	14	10899	519738910	142857	36	10899
519738911	142857	36	10899	9855545	175212	111	10899
9855546	175212	111	10899	4861239	144321	126	10899
4861230	144321	126	10899	4871139	141809	551	10899
4871130	141809	551	10899	800542114	180141	2	10899
800542113	180141	2	10899	51224923	200548	29	10899
51224922	190548	19	10899	9810316	210317	44	10899
9810315	200317	34	10899	9655856	220649	80	10899
9655855	210649	90	10899	3331013	230849	152	10899
3331012	220849	132	10899	2193897	230544	1109	10899
2193896	230544	909	10899	482281732	612	20	10899
482281731	612	20	10899	473314328	104412	29	10899
473314327	94412	29	10899	219921215	132010	30	36404
219921214	131010	30	36404	119724546	160906	14	36404
119724545	150906	14	36404	543141908	191204	13	36404
543141907	190204	13	36404	482281733	5319	21	20899
482281732	3319	21	20899	47331439	134912	25	20899
47331438	134612	25	20899	219921216	231610	20	36405
219921215	231010	20	36405	5372282345	111532	15	20899
3372282345	101532	25	20899	419804652	173023	210	20899
119804652	163023	21	20899	482231638	12	38	10899
482231638	12	33	10899	473334226	84412	45	10899
473314226	84412	44	10899	219931113	121010	54	36404
219921113	121010	53	36404	119734144	140906	51	36404
119724144	140906	56	36404	543131106	180204	49	36404
543141106	180204	48	36404	482231131	19	126	20899
482231131	19	129	20899	473334175	124412	47	20899
47331417	124412	49	20899	219931114	211010	112	20899
219921114	211010	232	20899	482232144	81532	42	20899
482232144	81532	49	20899	489834153	143023	48	20899
489834153	143023	44	20899	900339754	412	15	10899
900339854	312	15	10899	900321712	104412	12	10899
900321212	94412	12	10899	900414745	132010	11	36404
900414545	122010	11	36404	900419799	151806	18	36404
900419999	141906	18	36404	900518777	181604	11	36404
900518877	181204	11	36404	900512070	549	9	20899
900512020	519	9	20899	900681070	134712	18	20899
900681010	134412	18	20899	900540979	221810	1	20899
900540909	221010	2	20899	900014171	91542	19	20899
900014141	91552	19	20899	900961070	153723	20	20899
900961010	153023	20	20899	900339857	12	330	10899
900339855	12	730	10899	900221275	84412	490	10899
900221215	84412	290	10899	900114576	121010	169	36404
900114546	121010	69	36404	900119975	140906	85	36404
900119995	140906	115	36404	900218885	180204	96	36404
900218875	180204	126	36404	900212075	19	140	20899
900212025	19	130	20899	900481075	124412	99	20899
900481015	124412	97	20899	900540975	211010	232	20899
900540905	211010	202	20899	900414175	81532	130	20899
900414145	81532	100	20899	900999075	143023	170	20899
900991015	143023	150	20899	19729951072	12	16	10899
19729951072	12	15	10899	443199953	94412	18	10899
443199953	94412	17	10899	338799452929	131010	13	36404
338799452929	131010	12	36404	19769952354	142906	12	36404
19769952354	142906	1	36404	19739957878	182204	4	36404
19739957878	182204	3	36404	19629948765	319	3	20899
19629948765	319	1	20899	18769962323	123412	8	20899
18769962323	123412	7	20899	22999994646	221010	9	20899
22999994646	221010	8	20899	19722151082	91532	9	20899
19722151082	91532	9	20899	448722143545	153023	27	20899
448722143545	153023	27	20899	19723991072	312	169	10899
19723991072	112	169	10899	443199953	104412	295	10899
443199953	94412	295	10899	338799452929	121610	379	36404
338799452929	131010	279	36404	19769952354	160906	155	36404
19769952354	150906	155	36404	19739957878	190504	591	36404
19739957878	190204	191	36404	19629948765	319	924	20899
19629948765	219	224	20899	18769962323	135412	618	20899
18769962323	134412	318	20899	22999994646	224010	390	20899
22999994646	221010	490	20899	19999951082	111532	829	20899
19999951082	101532	429	20899	449999143545	173023	920	20899
449999143545	163023	620	20899	4713666	803	2	20899
517151031	109	7	20899	9718514	305	17	20899
9718131	3406	18	20899	516681160	253	38	20899
518002496	5332	30	20899	9804101	836	108	20899
99795993	5413	118	20899	9804401	836	308	20899
99795983	413	218	20899	9804411	1836	808	20899
99795583	613	718	20899	4713666	81203	2	20899
517151032	110109	7	20899	9718515	83005	17	20899
9718132	83406	18	20899	516681161	92253	38	20899
518002497	95332	30	20899	9804102	83612	108	20899
99795994	85413	118	20899	9804402	83615	308	20899
99795984	101413	218	20899	9804412	101836	808	20899
99795584	111613	718	20899	4713666	121203	2	20899
517151033	120109	7	20899	9718516	133005	17	20899
9718133	123406	18	20899	516681162	132253	38	20899
518002498	125332	30	20899	9804103	133612	108	20899
99795995	134413	118	20899	9804403	123615	308	20899
99795985	131413	218	20899	9804413	121836	808	20899
99795585	131613	718	20899	800990005	154012	2	20899
800990002	154012	2	20899	519789985	141713	14	10899
519789981	141713	14	20899	519738915	142857	36	10899
519738910	142857	36	20899	9855547	175212	111	10899
9855545	175212	111	20899	4861237	144321	126	10899
4861239	144321	126	20899	4871137	141809	551	10899
4871139	141809	551	20899	800542115	190141	1	10899
800542114	180141	2	20899	51224924	210548	24	10899
51224923	200548	29	20899	9810317	220317	34	20899
9810316	210317	44	20899	9655857	230649	84	20899
9655856	220649	80	20899	3331014	231849	152	20899
3331013	230849	152	20899	2193897	233544	1509	20899
2193897	230544	1109	20899	4713666	803	2	10899
517151031	109	7	10899	9718514	305	17	10899
9718131	3406	18	10899	516681160	253	38	10899
518002496	5332	30	10899	9804101	836	108	10899
99795993	5413	118	10899	9804401	836	308	10899















3481300	113630	117	10899	2865999	115512	1202	10899
3481200	113630	317	10899	135	121934	6	10899
3481210	113630	1017	10899	513612349	122801	13	10899
99791730	121222	3	10899	621008	134814	25	10899
8005102	125505	17	10899	2865993	132512	102	10899
9525909	135646	37	10899	2865996	132512	302	10899
3481301	133630	117	10899	2865999	135512	1202	10899
3481201	133630	317	10899	800991006	154012	4	10899
3481211	133630	1017	10899	519781986	141713	16	10899
800990006	154012	2	10899	519731916	142857	31	10899
519789986	141713	14	10899	9855518	175212	112	10899
519738916	142857	36	10899	4861218	144321	125	10899
9855548	175212	111	10899	4871118	141809	553	10899
4861238	144321	126	10899	800542118	190641	4	10899
4871138	141809	551	10899	51224927	190848	20	10899
800542117	1800141	1	10899	9810310	180917	39	10899
51224926	180548	24	10899	9655850	220049	100	10899
9810319	180317	34	10899	3331017	234549	230	10899
9655859	220649	84	10899	2193891	235544	1009	10899
3331016	231849	152	10899	482482832	612	2	10899
2193890	233544	1509	10899	473313228	104412	2	10899
482481832	612	20	10899	219833315	132010	4	36404
473314228	104412	29	10899	119623246	160906	4	36404
219831315	132010	30	36404	543153208	191204	9	36404
119624246	160906	14	36404	482483233	5319	12	20899
543151208	191204	13	36404	47361329	134912	12	20899
482481233	5319	21	20899	219823216	231610	14	36405
47361429	134912	25	20899	713391345	9352	19	20899
219821216	231610	20	36405	619802252	173023	12	20899
413391345	103532	9	20899	482252638	112	48	10899
119802652	166023	10	20899	473542226	94412	35	10899
482251638	112	48	10899	219952113	122010	56	36404
473544226	94412	35	10899	119752144	141906	52	36404
219951113	122010	56	36404	543152106	181204	39	36404
119754144	141906	52	36404	482252131	219	166	20899
543151106	181204	39	36404	473352417	124712	77	20899
482251131	219	166	20899	219952114	211210	152	20899
473354171	124712	77	20899	482252144	81542	52	20899
219951114	211210	152	20899	489852153	143223	38	20899
482252144	81542	52	20899	900379614	112	11	10899
489854153	143223	38	20899	900356722	113412	14	10899
900399614	212	1	10899	900456795	133310	21	36404
900396722	114412	1	10899	900456789	154306	19	36404
900496795	133010	2	36404	900556797	184304	12	36404
900496789	154806	9	36404	900556090	543	17	20899
900596797	184604	2	36404	900656090	133712	19	20899
900596090	545	7	20899	900556999	223810	14	20899
900696090	130712	9	20899	900056191	91342	11	20899
900596999	220810	4	20899	900956090	153723	7	20899
900096191	91042	11	20899	900112857	402	930	10899
90096090	150723	26	20899	900112275	110612	930	10899
900842857	432	910	10899	900112576	130610	339	36404
900842275	114612	810	10899	900112975	150046	235	36404
900842576	134610	319	36404	900222885	190014	436	36404
900842975	150446	215	36404	900222075	309	430	20899
900842885	190514	416	36404	900222075	130642	234	20899
900842075	319	440	20899	900222975	230410	432	20899
900842075	134642	284	20899	90022175	85072	230	20899
900852975	231410	402	20899	900222075	160323	237	20899
900842175	85472	290	20899	19799951072	12	4	10899
900892075	153323	287	20899	773999953	814412	7	10899
19799951072	412	19	10899	338999452929		133010	9
773999953	114412	29	10899	19799952354	152906	15	36404
338999452929	123010	19	36404	19799957878	212204	18	36404
19799952354	162906	5	36404	19699948765	319	14	20899
19799957878	202204	8	36404	18799962323	125412	14	20899
19699948765	519	4	20899	22999999646		233010	12
18799962323	135412	18	20899	3999991082	84532	6	20899
22999999646	235010	19	20899	44899993545		145023	9
19929991082	114532	3	20899	443199953	111412	125	10899
448799993545	175023	4	20899	338799452929		11150	1319
443199953	111412	1295	10899	19769952354	164906	955	36404
338799452929	11150	1379	36404	19739957878	205504	191	36404
19769952354	164906	1155	36404	19629948765	219	124	20899
19739957878	205504	591	36404	18769962323	165412	118	20899
19629948765	219	924	20899	22999999646		234010	390
18769962323	165412	118	20899	39999990083	105532	329	20899
22999999646	234010	3390	20899	669999903544		141023	620
19999990083	95532	1529	20899	2233522	10923	2	20899
449999903544	171023	1920	20899	513587289	12340	13	20899
135	5934	6	20899	323885	11241	37	20899
513612349	2801	13	20899	2102984	14349	100	20899
621008	4814	25	20899	2102982	4349	300	20899
2865993	12512	102	20899	2102582	51349	900	20899
2865996	2512	302	20899	2233522	110923	2	20899
2865999	15512	1202	20899	513587280	112340	13	20899
135	81934	6	20899	323886	111241	37	20899
513612349	92801	13	20899	2102985	84349	100	20899
621008	94814	25	20899	2102984	94349	300	20899
2865993	102512	102	20899	2102585	91349	900	20899
2865996	102512	302	20899	2233523	130923	2	20899
2865999	115512	1202	20899	513587281	132340	13	20899
135	121934	6	20899	323887	131241	37	20899
513612349	122801	13	20899	2102986	124349	100	20899
621008	134814	25	20899	2102985	124349	300	20899
2865993	132512	102	20899	2102586	131349	400	20899
2865996	132512	302	20899	800990009	154012	2	20899
2865999	135512	1202	20899	519789989	141713	14	20899
800991006	154012	4	20899	519738919	142857	36	20899
519781986	141713	16	20899	9855448	175212	111	20899
519731916	142857	31	20899	4861538	144321	126	20899
9855518	175212	112	20899	4871338	141809	551	20899
4861218	144321	125	20899	900542119	191641	5	20899
4871118	141809	553	20899	51224928	192848	27	20899
800542118	190641	4	20899	9810311	184917	37	20899
51224927	190848	20	20899	9655851	224049	109	20899
9810310	180917	39	20899	3331018	235549	240	20899
9655850	220049	100	20899	2193892	232544	1909	20899
3331017	234549	230	20899	2233522	10923	2	10899
2193891	235544	1009	20899	513587289	12340	13	10899
135	5934	6	10899	323885	11241	37	10899
513612349	2801	13	10899	2102984	14349	100	10899
621008	4814	25	10899	2102982	4349	300	10899
2865993	12512	102	10899	2102582	51349	900	10899
2865996	2512	302	10899	2233522	110923	2	10899
2865999	15512	1202	10899	513587280	112340	13	10899
135	81934	6	10899	323886	111241	37	10899
513612349	92801	13	10899	2102985	84349	100	10899
621008	94814	25	10899	2102984	94349	300	10899
2865993	102512	102	10899	2102585	91349	900	10899
2865996	102512	302	10899	2233523	130923	2	10899
2865999	115512	1202	10899	513587281	132340	13	10899
135	121934	6	10899	323887	131241	37	10899
513612349	122801	13	10899	2102986	124349	100	10899
621008	134814	25	10899	2102985	124349	300	10899
2865993	132512	102	10899	2102586	131349	400	10899
2865996	132512	302	10899	800990009	154012	2	10899
2865999	135512	1202	10899	519789989	141713	14	10899
800991006	154012	4	10899	519738919	142857	36	10899
519781986	141713	16	10899	9855448	175212	111	10899
519731916	142857	31	10899	4861538	144321	126	10899
9855518	175212	112	10899	4871338	141809	551	10899
4861218	144321	125	10899	900542119	191641	5	10899
4871118	141809	553	10899	51224928	192848	27	10899
800542118	190641	4	10899	9810311	184917	37	10899
51224927	190848	20	10899	9655851	224049	109	10899
9810310	180917	39	10899	3331018	235549	240	10899
9655850	220049	100	10899	2193892	232544	1909	10899
3331017	234549	230	10899	2233522	10923	2	10899
2193891	235544	1009	10899	513587289	12340	13	10899
135	5934	6	10899	323885	11241	37	10899
513612349	2801	13	10899	21029			















## Anexo 3 – CORBA/Java

### 3.1 Sistemas SSCC e SIPI

```

-----*/
/* Arquivo FrmAgente.java - Interface para o Agente. */
/*
   SSCC - Sistema de Seguranca Contra Celulares Clonados
   Autores: Alexandre Campos Hermida
           Walter do Valle

   Orientacao : Carlos Becker Westphall
   Co-orientacao : Mirela Secchi Moreti Anoni Notare
   Colaboracao : Joao Bosco Manguieira Sobral
               Daniela Claro
*/

//imports de interface
import java.awt.*;
import com.sun.java.swing.*;
import com.symantec.itools.swing.borders.BevelBorder;
import com.symantec.itools.swing.borders.TitledBorder;
import symantec.itools.awt.ImagePanel;

//imports de aplicacao
import java.io.*;
import java.util.Vector;
import Agent;
import symantec.itools.awt.StatusBar;

public class FrmAgente extends com.sun.java.swing.JFrame
{
    static String iniFileName = "sscc.ini";

    int i=0;
    private String nomeAgente;
    private boolean slvAuto;
    private int toleranciaAgente;

    Agent oAgente;

    public FrmAgente()
    {
        //((INIT_CONTROLS
        setJMenuBar(JMenuBar2);
        getContentPane().setLayout(null);
        setSize(457, 373);
        setVisible(false);
        titledBorder1.setTitle("Service");
        //SS titledBorder1.move(0, 384);
        bevelBorder2.setBorderType(com.sun.java.swing.border.BevelBorder.LOWERED);
        //SS bevelBorder2.move(24, 384);
        jPanelServico.setBorder(titledBorder1);
        jPanelServico.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));
        jPanelServico.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));
        getContentPane().add(jPanelServico);
        jPanelServico.setBounds(12, 276, 168, 60);
        jButtonIniciar.setText("Start");
        jButtonIniciar.setActionCommand("Iniciar");
        jButtonIniciar.setMnemonic((int)'A');
        jPanelServico.add(jButtonIniciar);
        jButtonIniciar.setBounds(19, 25, 63, 25);
        jButtonParar.setText("Stop");
        jButtonParar.setActionCommand("Parar");
        jButtonParar.setMnemonic((int)'P');
        jButtonParar.setEnabled(false);
        jPanelServico.add(jButtonParar);
        jButtonParar.setBounds(87, 25, 61, 25);
        JScrollPane.setViewportView(true);
        getContentPane().add(JScrollPane);
        JScrollPane.setBounds(12, 12, 432, 264);
        JTextAreaMensagens.setSelectionColor(new java.awt.Color(204, 204, 255));
        JTextAreaMensagens.setLineWrap(true);
        JTextAreaMensagens.setWrapStyleWord(true);
        JTextAreaMensagens.setDisabledTextColor(new java.awt.Color(153, 153, 153));
        JTextAreaMensagens.setEditable(false);
        JScrollPane.setViewportView().add(JTextAreaMensagens);
        JTextAreaMensagens.setBounds(0, 0, 429, 261);
        try {
            iconImage.setImageURL(symantec.itools.net.RelativeURL.getURL("Images/ssccicon-menor.gif"));
        }
        catch (java.net.MalformedURLException error) { }
        catch (java.beans.PropertyVetoException e) { }
        try {
            iconImage.setStyle(symantec.itools.awt.ImagePanel.IMAGE_SCALED_TO_FIT);
        }
        catch (java.beans.PropertyVetoException e) { }
        iconImage.setLayout(null);
        getContentPane().add(iconImage);
        iconImage.setBounds(264, 288, 50, 50);
        iconImage.setVisible(false);
        try {
            statusBar1.setBorderStyle(symantec.itools.awt.StatusBar.BEVEL_LOWERED);
        }
        catch (java.beans.PropertyVetoException e) { }
        try {
            statusBar1.setIPadBottom(2);
        }
        catch (java.beans.PropertyVetoException e) { }
        try {
            statusBar1.setIPadSides(2);
        }
        catch (java.beans.PropertyVetoException e) { }
        try {
            statusBar1.setPaddingTop(8);
        }
        catch (java.beans.PropertyVetoException e) { }
        try {

```

```

        statusBar1.setAlignStyle(symantec.itools.awt.StatusBar.ALIGN_LEFT);
    }
    catch (java.beans.PropertyVetoException e) {}
    getContentPane().add(statusBar1);
    statusBar1.setFont(new Font("Dialog", Font.PLAIN, 12));
    statusBar1.setBounds(0, 336, 456, 36);
    //SS JMenuBar2.move(48, 384);
    serviceMenu.setText("Service");
    serviceMenu.setActionCommand("File");
    serviceMenu.setMnemonic((int)'V');
    JMenuBar2.add(serviceMenu);
    startItem.setHorizontalTextPosition(com.sun.java.swing.SwingConstants.RIGHT);
    startItem.setText("Start");
    startItem.setActionCommand("Start");
    startItem.setAccelerator(com.sun.java.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_A,
java.awt.Event.CTRL_MASK));
    stopItem.setMnemonic((int)'A');
    serviceMenu.add(stopItem);
    stopItem.setHorizontalTextPosition(com.sun.java.swing.SwingConstants.RIGHT);
    stopItem.setText("Stop");
    stopItem.setActionCommand("Stop");
    stopItem.setAccelerator(com.sun.java.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_P,
java.awt.Event.CTRL_MASK));
    stopItem.setMnemonic((int)'P');
    serviceMenu.add(stopItem);
    serviceMenu.add(JSeparator2);
    configItem.setText("Configuration...");
    configItem.setActionCommand("Configuration");
    configItem.setMnemonic((int)'C');
    serviceMenu.add(configItem);
    serviceMenu.add(JSeparator3);
    exitItem.setText("Exit");
    exitItem.setActionCommand("Exit");
    exitItem.setMnemonic((int)'X');
    serviceMenu.add(exitItem);
    backupMenu.setText("Backup");
    backupMenu.setActionCommand("File");
    backupMenu.setMnemonic((int)'B');
    JMenuBar2.add(backupMenu);
    openItem.setHorizontalTextPosition(com.sun.java.swing.SwingConstants.RIGHT);
    openItem.setText("Open...");
    openItem.setActionCommand("Open...");
    openItem.setAccelerator(com.sun.java.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_O,
java.awt.Event.CTRL_MASK));
    openItem.setMnemonic((int)'O');
    backupMenu.add(openItem);
    saveItem.setHorizontalTextPosition(com.sun.java.swing.SwingConstants.RIGHT);
    saveItem.setText("Save");
    saveItem.setActionCommand("Save");
    saveItem.setAccelerator(com.sun.java.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_S,
java.awt.Event.CTRL_MASK));
    saveItem.setMnemonic((int)'S');
    backupMenu.add(saveItem);
    helpMenu.setText("Help");
    helpMenu.setActionCommand("Help");
    helpMenu.setMnemonic((int)'H');
    JMenuBar2.add(helpMenu);
    aboutItem.setHorizontalTextPosition(com.sun.java.swing.SwingConstants.RIGHT);
    aboutItem.setText("About...");
    aboutItem.setActionCommand("About...");
    aboutItem.setMnemonic((int)'A');
    helpMenu.add(aboutItem);

    setIconImage(iconImage.getImage());
    //}

    //({INIT_MENUS
    //})

    //({REGISTER_LISTENERS
    SysAction lSysAction = new SysAction();
    startItem.addActionListener(lSysAction);
    stopItem.addActionListener(lSysAction);
    exitItem.addActionListener(lSysAction);
    SysWindow aSysWindow = new SysWindow();
    this.addWindowListener(aSysWindow);
    configItem.addActionListener(lSysAction);
    aboutItem.addActionListener(lSysAction);
    jBtnIniciar.addActionListener(lSysAction);
    jBtnParar.addActionListener(lSysAction);
    openItem.addActionListener(lSysAction);
    saveItem.addActionListener(lSysAction);
    //})
}

public FrmAgente(String sTitle)
{
    this();
    setTitle(sTitle);
}

public void setVisible(boolean b)
{
    if (b)
        setLocation(50, 50);
    super.setVisible(b);
}

static public void main(String args[])
{
    (new FrmAgente()).setVisible(true);
}

//-----
private void inicioAgente()
{
    try {
        jTxtAreaMensagens.append("\nStarting Agent ... \n\n");

        oAgente = new Agent(this.jTxtAreaMensagens, nomeAgente, slvAuto, toleranciaAgente, "nncalls.in");
        oAgente.start();

        setText_TxtFieldInfos("Agent started at ");
        setTitle(oAgente.recurnName());

    } catch (java.lang.Exception e) {
        com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Error", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
    }
}

//-----

```

```

public void setValores(String nom, int tolerancia, boolean auto){
    nomeAgente = nom;
    setTitle(nomeAgente);
    toleranciaAgente = tolerancia;
    if (auto) {saveItem.setEnabled(false);slvAuto=true;}
    else {saveItem.setEnabled(true);slvAuto=false;}
}

public void setText_TxtFieldInfos(String info)
{
    try{
        statusBar1.setStatusText(info + " : "+oAgente.returnDate() + " " + oAgente.returnTime());
    }catch(Exception e){com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Error #0015", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);}
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension size = getSize();

    super.addNotify();

    if (frameSizeAdjusted)
        return;
    frameSizeAdjusted = true;

    // Adjust size of frame according to the insets and menu bar
    Insets insets = getInsets();
    com.sun.java.swing.JMenuBar menuBar = getRootPane().getJMenuBar();
    int menuBarHeight = 0;
    if (menuBar != null)
        menuBarHeight = menuBar.getPreferredSize().height;
    setSize(insets.left + insets.right + size.width, insets.top + insets.bottom + size.height + menuBarHeight);
}

// Used by addNotify
boolean frameSizeAdjusted = false;

//((DECLARE_CONTROLS
com.symantec.itools.swing.borders.TitledBorder titledBorder1 = new com.symantec.itools.swing.borders.TitledBorder();
com.symantec.itools.swing.borders.BevelBorder bevelBorder2 = new com.symantec.itools.swing.borders.BevelBorder();
com.sun.java.swing.JPanel jPanelService = new com.sun.java.swing.JPanel();
com.sun.java.swing.JButton btnIniciar = new com.sun.java.swing.JButton();
com.sun.java.swing.JButton btnParar = new com.sun.java.swing.JButton();
com.sun.java.swing.JScrollPane jScrollPane = new com.sun.java.swing.JScrollPane();
com.sun.java.swing.JTextArea jTextAreaMensagens = new com.sun.java.swing.JTextArea();
symantec.itools.awt.ImagePanel iconImage = new symantec.itools.awt.ImagePanel();
symantec.itools.awt.StatusBar statusBar1 = new symantec.itools.awt.StatusBar();
com.sun.java.swing.JMenuBar jMenuBar2 = new com.sun.java.swing.JMenuBar();
com.sun.java.swing.JMenu serviceMenu = new com.sun.java.swing.JMenu();
com.sun.java.swing.JMenuItem startItem = new com.sun.java.swing.JMenuItem();
com.sun.java.swing.JMenuItem stopItem = new com.sun.java.swing.JMenuItem();
com.sun.java.swing.JSeparator jSeparator2 = new com.sun.java.swing.JSeparator();
com.sun.java.swing.JMenuItem configItem = new com.sun.java.swing.JMenuItem();
com.sun.java.swing.JSeparator jSeparator3 = new com.sun.java.swing.JSeparator();
com.sun.java.swing.JMenuItem exitItem = new com.sun.java.swing.JMenuItem();
com.sun.java.swing.JMenu backupMenu = new com.sun.java.swing.JMenu();
com.sun.java.swing.JMenuItem openItem = new com.sun.java.swing.JMenuItem();
com.sun.java.swing.JMenuItem saveItem = new com.sun.java.swing.JMenuItem();
com.sun.java.swing.JMenu helpMenu = new com.sun.java.swing.JMenu();
com.sun.java.swing.JMenuItem aboutItem = new com.sun.java.swing.JMenuItem();
//))

//((DECLARE_MENUS
//))

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == startItem)
            startItem_actionPerformed(event);
        else if (object == stopItem)
            stopItem_actionPerformed(event);
        else if (object == exitItem)
            exitItem_actionPerformed(event);
        else if (object == configItem)
            configItem_actionPerformed(event);
        else if (object == aboutItem)
            aboutItem_actionPerformed(event);
        else if (object == btnIniciar)
            btnIniciar_actionPerformed(event);
        else if (object == btnParar)
            btnParar_actionPerformed(event);
        else if (object == openItem)
            openItem_actionPerformed(event);
        else if (object == saveItem)
            saveItem_actionPerformed(event);
    }

    void startItem_actionPerformed(java.awt.event.ActionEvent event)
    {
        if (jBtnIniciar.isEnabled()){
            jBtnParar.setEnabled(true);
            jBtnIniciar.setEnabled(false);

            startItem.setEnabled(false);
            stopItem.setEnabled(true);

            inicieAgente();
        }
    }

    void stopItem_actionPerformed(java.awt.event.ActionEvent event)
    {
        if (jBtnParar.isEnabled()){
            jTextAreaMensagens.append("\nStopping Agent ... \n");
            if (!oAgente.stopAgent())
                jTextAreaMensagens.append("\nThe Agent can't be stoped. \n");
            else
                {
                    jBtnIniciar.setEnabled(true);
                    jBtnParar.setEnabled(false);
                }
        }
    }
}

```

```

        startItem.setEnabled(true);
        stopItem.setEnabled(false);

        setText_txtFieldInfos("Agent stopped at ");
        jTxtAreaMensagens.append("\nAgent stopped.\n");
    }
}

void exitItem_actionPerformed(java.awt.event.ActionEvent event)
{
    try {
        Toolkit.getDefaultToolkit().beep();
        int reply = com.sun.java.swing.JOptionPane.showConfirmDialog(this,
            "Do you really want to exit?",
            "SSCC - Exit",
            com.sun.java.swing.JOptionPane.YES_NO_OPTION,
            com.sun.java.swing.JOptionPane.QUESTION_MESSAGE);

        if (reply == com.sun.java.swing.JOptionPane.YES_OPTION)
        {
            stopItem_actionPerformed(event);
            this.setVisible(false);
            this.dispose();
            System.exit(0);
        }
    } catch (Exception e) {
    }
}

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == FrmAgente.this)
            FrmAgente_windowClosing(event);
    }

    public void windowOpened(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == FrmAgente.this)
            FrmAgente_windowOpened(event);
    }
}

void FrmAgente_windowOpened(java.awt.event.WindowEvent event)
{
    try{
        FileInputStream oArqIni = new FileInputStream(iniFileName);
        DataInputStream oDataInArqIni = new DataInputStream(oArqIni);

        jBtnParar.setEnabled(false);

        nomeAgente = oDataInArqIni.readLine();
        setTitle(nomeAgente);
        toleranciaAgente = Integer.valueOf(oDataInArqIni.readLine()).intValue();

        String sAuto = oDataInArqIni.readLine();
        if (sAuto.equalsIgnoreCase("1"))(slvAuto=true;saveItem.setEnabled(false);)
        else (slvAuto=false;saveItem.setEnabled(true);)

        oDataInArqIni.close();
        oArqIni.close();
    }catch (FileNotFoundException fnfe) (com.sun.java.swing.JOptionPane.showMessageDialog(null, fnfe,
        "Error", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);)
    catch (IOException ioe) (com.sun.java.swing.JOptionPane.showMessageDialog(null, ioe, "Error", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);)
}

void configItem_actionPerformed(java.awt.event.ActionEvent event)
{
    try {
        JDialogConfiguracoes JDialogll = new JDialogConfiguracoes(this);
        JDialogll.setModal(true);
        JDialogll.setTitle("Configuration");
        JDialogll.show();
    } catch (java.lang.Exception e) (com.sun.java.swing.JOptionPane.showMessageDialog(null, "Error on open configuration
        window.", "Error", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);)
}

void aboutItem_actionPerformed(java.awt.event.ActionEvent event)
{
    try {
        JAboutDialogl aboutDialog = new JAboutDialogl(this);
        aboutDialog.setModal(true);
        aboutDialog.show();
    } catch (Exception e) (com.sun.java.swing.JOptionPane.showMessageDialog(null, "Error on open about dialog.",
        "Error", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);)
}

void jBtnIniciar_actionPerformed(java.awt.event.ActionEvent event)
{
    startItem_actionPerformed(event);
}

void jBtnParar_actionPerformed(java.awt.event.ActionEvent event)
{
    stopItem_actionPerformed(event);
}

void FrmAgente_windowClosing(java.awt.event.WindowEvent event)
{
    exitItem_actionPerformed(null);
}

void openItem_actionPerformed(java.awt.event.ActionEvent event)
{
    JFrmBackupReader reader = new JFrmBackupReader();
    reader.jBtnSave.setVisible(false);
    reader.leiaArquivoBackup();
    reader.setTitle("Saved log");
    reader.show();
}

void saveItem_actionPerformed(java.awt.event.ActionEvent event)

```

```

    try {
        if(saveItem.isEnabled()){
            Vector clonesSalvar = new Vector();
            clonesSalvar = oAgente.returnClones();
            JFrmBackupReader backupReader = new JFrmBackupReader();
            backupReader.setTitle("Unsaved log informations");
            backupReader.show();
            backupReader.mostraDados(oAgente, clonesSalvar);
        }
    } catch (Exception e) {
        com.sun.java.swing.JOptionPane.showMessageDialog(null,
            "Agent not started!",
            "Error",
            com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
    }
}

/*-----*/
/* Arquivo Agent.java - A implementacao do Agente. */
import java.util.Date;
import java.util.Vector;
import java.sql.*;
import Adaptador;
import java.io.*;
import java.text.DateFormat;
import Verifier;
import com.sun.java.swing.JTextArea;
import SSTCC.SSCCManager;

class Agent extends Thread {
    private String name;
    private int precisao;
    private String initDate;
    private String initTime;
    private boolean autoSave;

    private String testTelephonNumber;
    String arqInName;
    Vector clonesSalvar = new Vector();
    Adaptador oAdaptador;
    Verifier aVerifier;

    File NNOutFile;

    Connection con;
    Statement stat;

    //CORBA
    SSTCC.SSCCManager theManager;
    org.omg.CORBA.ORB orb;

    JTextArea aTextArea;

    /* Inicio do Construtor */
    Agent(JTextArea theTextArea, String agentName, boolean salvaAuto, int slideTemp, String arqInName)
    {
        try{
            aTextArea = theTextArea;

            // inicializando o ORB
            orb = org.omg.CORBA.ORB.init();
            aTextArea.append("ORB initialized\n\n");

            // Bind para o Objeto agentI
            theManager = SSTCC.SSCCManagerHelper.bind(orb, "agentI");

            NNOutFile = new File("nncalls.out");

            arqInName = arqInName;

            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con = DriverManager.getConnection("jdbc:odbc:AGAD","","");
            stat = con.createStatement();

        } catch (org.omg.CORBA.SystemException e)
        {
            com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Error Initializing CORBA", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
        }
        catch (Exception e)
        {
            com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Error #2109", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
        }

        name = agentName;
        autoSave = salvaAuto;
        precisao = slideTemp;
        getSystemDate();

        oAdaptador = new Adaptador(this, "cdr.txt", precisao);
        oAdaptador.start();

        aVerifier = new Verifier(this, "nncalls.out", 5000);
        aVerifier.start();
    }
    /* Fim do Construtor */

    /* Inicio dos Metodos Privados */
    private void saveCallFile(ResultSet ligacoes)/*, String arqName*/
    {
        try {
            FileOutputStream nncalls = new FileOutputStream(arqInName, false);
            PrintWriter oOutNncalls = new PrintWriter(nncalls);

            String output = "";

            while (ligacoes.next())
            {
                output = ligacoes.getString("telUsuario")+" "+
                    ligacoes.getString("dataChamada")+" "+
                    ligacoes.getString("horaChamada")+" "+
                    ligacoes.getString("duracao");
                oOutNncalls.println(output);
            }
            oOutNncalls.close();
            nncalls.close();
        }
    }
}

```



```

    }
    catch (IOException ioe){
        com.sun.java.swing.JOptionPane.showMessageDialog(null, ioe, "I/O Error #2287", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
    }
    catch (SQLException sqle){
        com.sun.java.swing.JOptionPane.showMessageDialog(null, sqle, "SQL Error #2287", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
    }
    catch (Exception e){
        com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Error #2287", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
    }
}

/* Inicio dos Metodos Publicos */
public boolean stopAgent()
{
    try{
        aVerifier.stop();
        oAdaptador.stop();
        this.stop();
        return true;
    }catch(Exception e)
    {
        com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Error #4423", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
        return false;
    }
}

public void adapterCallingAgent(String tel)
{
    try{
        testTelephonNumber = tel;
        ResultSet rs = stat.executeQuery("SELECT * FROM Transf WHERE telUsuario='"+tel+"'");

        this.saveCallFile(rs);
        stat.executeUpdate("DELETE * FROM Transf WHERE telUsuario='"+tel+"'");
    }catch(Exception e)
    {
        com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Error #4236", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
    }
}

public void verifierCallingAgent()
{
    String input;

    try{
        FileInputStream NNOutInStrmFile;
        DataInputStream DtInStrmNNOutFile;

        NNOutInStrmFile = new FileInputStream("nncalls.out");
        DtInStrmNNOutFile = new DataInputStream(NNOutInStrmFile);

        input = DtInStrmNNOutFile.readLine();

        this.verifyCloning(input);
    }catch(Exception e)
    {
        com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Error #3382", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
    }
}

private void verifyCloning(String input)
{
    int position = input.indexOf("1") + 1;

    try{
        ResultSet rsTestOfClon = stat.executeQuery("SELECT * FROM UsuariosClasses WHERE telUsuario='"+testTelephonNumber+"'");
        rsTestOfClon.next();
        int pattern = rsTestOfClon.getInt("classeUsuario");

        /*
        *****
        /* Mandar mensagem pro gerente se detectado possivel clone *
        *****
        */

        if(pattern != position)
        {
            //position eh o padrao calculado
            //pattern eh o padrao atual

            String output = testTelephonNumber + " " + position + ";" + pattern + "/" + " "+ "/;

            /* salvamento automatico */
            if (autoSave)
            {
                saveLogFile("backup.log", output+" ");
            }
            else
            {
                clonesSalvar.addElement(output);
            }
            /* ----- */

            /* Mandando mensagem pro gerente via CORBA */
            System.out.println("MANDOU MENSAGEM PRO GERENTE -> "+ output);

            if (theManager.recebaMensagem(output))
            {
                this.aTextArea.append("Telephone " +testTelephonNumber + " pattern " + pattern + " made a call pattern " + position + "\n");
            }
            else
            {
                this.aTextArea.append("Nao houve comunicacao com o Gerente");
            }
            /* ----- */
        }
    }catch(Exception e)
    {
        System.out.println(e.toString());
        com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Error #1243", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
    }
}

public void changeName(String newName)
{
    name = newName;
}

public String returnName()

```

```

    {
        return name;
    }

    public void setRelaxacao(int precision)
    {
        precisao = precision;
    }

    public int returnRelaxacao()
    {
        return precisao;
    }

    public String returnDate()
    {
        return initDate;
    }

    public String returnTime()
    {
        return initTime;
    }

    public Vector returnClones()
    {
        return clonesSalvar;
    }

    public void clearClones()
    {
        clonesSalvar.removeAllElements();
    }

    /* Fim dos Metodos Publicos */

    /* Inicio dos Metodos Privados */

    private void getSystemDate()
    {
        Date umaData = new Date();

        String day = String.valueOf(umaData.getDate());
        String month = String.valueOf(umaData.getMonth()+1);

        String year = String.valueOf(umaData.getYear() + 1900);

        initDate = day+" / "+month+" / "+year;
        initTime = umaData.getHours()+":"+umaData.getMinutes()+":"+umaData.getSeconds();
    }

    private void saveLogFile(String arcName, String output) throws IOException
    {
        FileOutputStream backupLog = new FileOutputStream(arcName, true);
        PrintWriter oOutBackupLog = new PrintWriter(backupLog);
        oOutBackupLog.println(output);
        oOutBackupLog.close();
        backupLog.close();
    }
}

/*-----*/

/* Arquivo Adaptador.java - responsável por ler os registros do CDR */

import java.io.*;
import java.sql.*;

class Adaptador extends Thread
{
    String CDRName;

    /*tempo de espera em milissegundos*/
    long tmpEspera = 5000;

    long backLastMod=0;
    long lastMod = 0;

    int precisao;

    Statement stmt = null;
    Connection con = null;

    FileInputStream oArqCDR;
    DataInputStream oDataInArqCDR;
    File arquivo;

    Agent anAgent;

    Adaptador (Agent theAgent, String ArcName, int pres)
    {
        CDRName = ArcName;
        precisao = pres;

        try{
            /*registra o driver*/
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            /*Cria e estabelece a conexao*/
            con = DriverManager.getConnection("jdbc:odbc:AGAD","","");
            /*Cria um comando*/
            stmt = con.createStatement();

            oArqCDR = new FileInputStream(CDRName);
            oDataInArqCDR = new DataInputStream(oArqCDR);
            arquivo = new File(CDRName);

            anAgent = theAgent;
        }
        catch (SQLException e){
            com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "SQL Error", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
            System.exit(1);
        }
        catch (FileNotFoundException e){
            com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "File Not Found Error", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
            System.exit(1);
        }
        catch (Exception e){
            com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Generic Error #2237", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

```

        System.exit(1);
    }

    //{{{INIT_CONTROLS
    //}}
}

public void run ()
{
    String input = "";
    try
    {
        /* *** Deprecated ***
        Statement stmtTransf = con.createStatement();
        stmtTransf.executeUpdate("DELETE * FROM Transf");
        */

        while(true){

            this.sleep(tmpEspera);

            backLastMod = lastMod;
            lastMod = arquivo.lastModified();

            if ((backLastMod != lastMod) && (backLastMod !=0))
            {
                //Já passou a primeira vez e está pegando a partir do fim agora
                while ((input = oDataInArqCDR.readLine()) != null)
                {
                    try {
                        stmt.executeUpdate("INSERT INTO Transf (telUsuario, telChamado, duracao, dataChamada) VALUES
                        ("'+input.substring(13,23)+'", "'+input.substring(24,34)+'", "'+input.substring(44,53)+'", "'+input.substring(35,43)+'");");
                        if (verificaPrecisao(input.substring(13,23)))
                        {
                            /*chama Agente*/
                            anAgent.adapterCallingAgent(input.substring(13,23));
                            System.out.println("CHAMOU O AGENTE! Telefone: "+ input.substring(13,23));
                        }
                    } catch (Exception e){
                        com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Error #2683", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
                    }
                }
            }
            else
            {
                if ((backLastMod == 0))
                {
                    //Está passando a primeira vez
                    stmt.executeUpdate("DELETE * FROM Transf");
                    while ((input = oDataInArqCDR.readLine()) != null)
                    {
                        try {
                            stmt.executeUpdate("INSERT INTO Transf (telUsuario, telChamado, duracao, dataChamada) VALUES
                            ("'+input.substring(13,23)+'", "'+input.substring(24,34)+'", "'+input.substring(44,53)+'", "'+input.substring(35,43)+'");");
                            if (verificaPrecisao(input.substring(13,23)))
                            {
                                /*chama Agente*/
                                anAgent.adapterCallingAgent(input.substring(13,23));
                                System.out.println("CHAMOU O AGENTE! Telefone: "+ input.substring(13,23));
                            }
                        } catch (Exception e){
                            com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Error #1514", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
                        }
                    }
                }
                //----- Fim da primeira passagem -----
            }
        }
    } //Fim do while(true)
} //Fim do try
catch (FileNotFoundException e){
    com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "File Not Found Error", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
}
catch (IOException e){
    com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "I/O Error", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
}
catch (Exception e){
    com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Error #2419", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
}
}

private boolean verificaPrecisao(String telefone)
{
    int counter = 0;

    /* Procurar um comando de SQL para contar número de registros */
    try {
        ResultSet rs = stmt.executeQuery("SELECT * FROM Transf WHERE telUsuario='"+telefone+"'");

        while (rs.next()){counter++;}
        if(counter == precisao){counter = 0;return true;}
        else {counter = 0;return false;}
    } catch (Exception e){return false;}
}

//{{{DECLARE_CONTROLS
//}}
} //Fim da classe

/*-----*/
Arquivo Verifier.java - responsável por monitorar se já houve 10 ligações

import java.io.*;

public class Verifier extends Thread
{
    int waitTim = 0;
    long backLastMod=0;
    long lastMod = 0;
    String arqNam;
    File aFile;
    Agent anAgent;

    Verifier(Agent theAgent, String arqName, int waitTime)
    {
        arqNam = arqName;
        waitTim = waitTime;
        anAgent = theAgent;
    }
}

```

```

        this.setPriority(MIN_PRIORITY);

        //{{{INIT_CONTROLS
        //}}}
    }

    public void run()
    {
        try{
            aFile = new File(arqNam);
            while(true){
                this.sleep(waitTim);

                backLastMod = lastMod;
                lastMod = aFile.lastModified();

                if ((backLastMod != lastMod) && (backLastMod !=0))
                {
                    System.out.println("Chamando o Agente...");
                    anAgent.verifierCallingAgent();
                }
            }//while
        }catch(Exception e)
        {
            com.sun.java.swing.JOptionPane.showMessageDialog(null, e, "Error #1982", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
        }

        //{{{DECLARE_CONTROLS
        //}}}
    }
}
-----*/
/* Arquivo JFrmBackupReader.java - Responsável por ler e gravar
dados do log ainda nao gravados */

/*
        A basic implementation of the JFrame class.
*/

import java.awt.*;
import com.sun.java.swing.*;
import com.symantec.itools.swing.borders.TitledBorder;
import java.io.*;
import java.util.Vector;

public class JFrmBackupReader extends com.sun.java.swing.JFrame
{
    Vector clonesSalvar;
    Agent theAgent;

    public JFrmBackupReader()
    {
        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.
        //{{{INIT_CONTROLS
        setResizable(true);
        getContentPane().setLayout(null);
        setSize(436, 316);
        setVisible(false);
        JScrollPane.setOpaque(true);
        getContentPane().add(JScrollPane);
        JScrollPane.setBounds(24, 24, 384, 206);
        JTextAreaLog.setEditable(false);
        JScrollPane.getViewport().add(JTextAreaLog);
        JTextAreaLog.setBounds(0, 0, 381, 203);
        titledBorder1.setTitle("Service");
        setLocation(100, 100);
        //$$ titledBorder1.move(0, 324);
        jButtonSave.setText("Save");
        jButtonSave.setActionCommand("Save");
        getContentPane().add(jButtonSave);
        jButtonSave.setBounds(252, 264, 63, 25);
        jButtonClose.setText("Close");
        jButtonClose.setActionCommand("Exit");
        getContentPane().add(jButtonClose);
        jButtonClose.setBounds(336, 264, 67, 25);
        //}}}

        //{{{INIT_MENUS
        //}}}

        //{{{REGISTER_LISTENERS
        SymAction lSymAction = new SymAction();
        jButtonClose.addActionListener(lSymAction);
        jButtonSave.addActionListener(lSymAction);
        //}}}
    }

    public JFrmBackupReader(String sTitle)
    {
        this();
        setTitle(sTitle);
    }

    public void setVisible(boolean b)
    {
        if (b)
            setLocation(50, 50);
        super.setVisible(b);
    }

    public void mostreDados(Agent anAgent, Vector clonesSave){
        clonesSalvar = clonesSave;
        theAgent = anAgent;

        if (clonesSalvar.size() == 0){
            jButtonSave.setEnabled(false);
        }
        else {
            for (int i=0; i< clonesSalvar.size(); i++){
                String input = (String)clonesSalvar.elementAt(i);
                this.JTextAreaLog.append("Telephone " +
                    input.substring(0, 9) +
                    " pattern " +
                    input.substring(13, 14) +
                    " has done a call pattern " +
                    input.substring(10, 13) +

```

```

        "\n");
    }
    //fim do else:
}

public void leiaArquivoBackup(){
try{
    FileInputStream oArqBackup = new FileInputStream("backup.log");
    DataInputStream oDataInArqBkp = new DataInputStream(oArqBackup);
    String input;
    while ((input = oDataInArqBkp.readLine()) != null){
        JTextAreaLog.append("Telephone " +
            input.substring(0, 9) +
            " pattern " +
            input.substring(13, 15) +
            " has done a call pattern " +
            input.substring(10, 13) +
            "\n");
    }
    oDataInArqBkp.close();
    oArqBackup.close();
} catch (Exception e){
    com.sun.java.swing.JOptionPane.showMessageDialog(null,
        e,
        "Error #1514",
        com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
}

static public void main(String args[])
{
    (new JFrmBackupReader()).setVisible(true);
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension size = getSize();

    super.addNotify();

    if (frameSizeAdjusted)
        return;
    frameSizeAdjusted = true;

    // Adjust size of frame according to the insets and menu bar
    Insets insets = getInsets();
    com.sun.java.swing.JMenuBar menuBar = getRootPane().getJMenuBar();
    int menuBarHeight = 0;
    if (menuBar != null)
        menuBarHeight = menuBar.getPreferredSize().height;
    setSize(insets.left + insets.right + size.width, insets.top + insets.bottom + size.height + menuBarHeight);
}

// Used by addNotify
boolean frameSizeAdjusted = false;

//((DECLARE_CONTROLS
com.sun.java.swing.JScrollPane JScrollPanel = new com.sun.java.swing.JScrollPane();
com.sun.java.swing.JTextArea JTextAreaLog = new com.sun.java.swing.JTextArea();
com.symantec.itools.swing.borders.TitledBorder titledBorder1 = new com.symantec.itools.swing.borders.TitledBorder();
com.sun.java.swing.JButton jBtnSave = new com.sun.java.swing.JButton();
com.sun.java.swing.JButton jBtnClose = new com.sun.java.swing.JButton();
//))

//((DECLARE_MENUS
//))

class SynAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == jBtnClose)
            jBtnClose_actionPerformed(event);
        else if (object == jBtnSave)
            jBtnSave_actionPerformed(event);
    }

    void jBtnClose_actionPerformed(java.awt.event.ActionEvent event)
    {
        this.setVisible(false);
    }

    void jBtnSave_actionPerformed(java.awt.event.ActionEvent event)
    {
        if (jBtnSave.isEnabled()){
try{
        FileOutputStream backupLog = new FileOutputStream("backup.log", true);
        PrintWriter oOutBackupLog = new PrintWriter(backupLog);

        for (int i = 0; i < clonesSalvar.size(); i++)
        {
            oOutBackupLog.println(clonesSalvar.elementAt(i) + " ");
        }

        oOutBackupLog.close();
        backupLog.close();
        theAgent.clearClones();
        com.sun.java.swing.JOptionPane.showMessageDialog(null,
            "Save log successful!",
            "Information",
            com.sun.java.swing.JOptionPane.INFORMATION_MESSAGE);

        jBtnSave.setEnabled(false);
        JTextAreaLog.setText("");
} catch (FileNotFoundException fnfe){
    com.sun.java.swing.JOptionPane.showMessageDialog(null,
        fnfe,
        "Error",
        com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
} catch (IOException ioe){
    com.sun.java.swing.JOptionPane.showMessageDialog(null,
        ioe,
        "Error",
        com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
} catch (Exception e){

```



```

        setLocation(150, 150);
        super.setVisible(b);
    }

    static public void main(String args[])
    {
        (new JDialogConfiguracoes()).setVisible(true);
    }

    private void saveIniFile() throws IOException
    {
        try{
            FileOutputStream iniFileStream = new FileOutputStream(iniFileName, false);
            PrintWriter iniFilePW = new PrintWriter(iniFileStream);

            iniFilePW.println(jTxtPanaNome.getText());
            //iniFilePW.println(jTxtPanaRelaxacao.getText());
            iniFilePW.println(JSliderTempo.getValue()+"");

            if (JChkBoxAutomatico.isSelected()){
                iniFilePW.println("1");
            }
            else{
                iniFilePW.println("0");
            }

            iniFilePW.close();
            iniFileStream.close();
        }
        catch(Exception ioe){com.sun.java.swing.JOptionPane.showMessageDialog(null, ioe, "Error", com.sun.java.swing.JOptionPane.ERROR_MESSAGE);}
    }

    public void addNotify()
    {
        Dimension size = getSize();

        super.addNotify();

        if (frameSizeAdjusted)
            return;
        frameSizeAdjusted = true;

        Insets insets = getInsets();
        setSize(insets.left + insets.right + size.width, insets.top + insets.bottom + size.height);
    }

    boolean frameSizeAdjusted = false;

    //{DECLARE CONTROLS
    com.sun.java.swing.JPanel jPanelMensagens = new com.sun.java.swing.JPanel();
    com.sun.java.swing.JButton jBtnOk = new com.sun.java.swing.JButton();
    com.sun.java.swing.JButton jBtnCancelar = new com.sun.java.swing.JButton();
    com.symantec.itools.swing.borders.TitledBorder titledBorder2 = new com.symantec.itools.swing.borders.TitledBorder();
    com.sun.java.swing.JPanel jPanel1 = new com.sun.java.swing.JPanel();
    com.sun.java.swing.JLabel jLabel2 = new com.sun.java.swing.JLabel();
    com.sun.java.swing.JTextArea jTxtPanaNome = new com.sun.java.swing.JTextArea();
    com.sun.java.swing.JPanel jPanel2 = new com.sun.java.swing.JPanel();
    com.sun.java.swing.JCheckBox JChkBoxAutomatico = new com.sun.java.swing.JCheckBox();
    com.symantec.itools.swing.borders.TitledBorder titledBorder1 = new com.symantec.itools.swing.borders.TitledBorder();
    com.sun.java.swing.JPanel jPanel3 = new com.sun.java.swing.JPanel();
    com.sun.java.swing.JSlider JSliderTempo = new com.sun.java.swing.JSlider();
    com.sun.java.swing.JLabel jLabelTempo = new com.sun.java.swing.JLabel();
    com.symantec.itools.swing.borders.TitledBorder titledBorder3 = new com.symantec.itools.swing.borders.TitledBorder();
    //}

    class SymMouse extends java.awt.event.MouseAdapter
    {
        public void mouseClicked(java.awt.event.MouseEvent event)
        {
            Object object = event.getSource();
            if (object == jBtnCancelar)
                jBtnCancelar_mouseClicked(event);
            else if (object == jBtnOk)
                jBtnOk_mouseClicked(event);
        }
    }

    void jBtnCancelar_mouseClicked(java.awt.event.MouseEvent event)
    {
        jBtnCancelar_mouseClicked_Interaction1(event);
    }

    void jBtnCancelar_mouseClicked_Interaction1(java.awt.event.MouseEvent event)
    {
        try {
            this.setVisible(false);
        } catch (java.lang.Exception e) {
        }
    }

    class SymChange implements com.sun.java.swing.event.ChangeListener
    {
        public void stateChanged(com.sun.java.swing.event.ChangeEvent event)
        {
            Object object = event.getSource();
            if (object == JSliderTempo)
                JSliderTempo_stateChanged(event);
        }
    }

    void JSliderTempo_stateChanged(com.sun.java.swing.event.ChangeEvent event)
    {
        JSliderTempo_stateChanged_Interaction1(event);
    }

    void JSliderTempo_stateChanged_Interaction1(com.sun.java.swing.event.ChangeEvent event)
    {
        try {
            if ((JSliderTempo.getValue() <= 5) || (JSliderTempo.getValue() >= 15))
            {
                JSliderTempo.setForeground(Color.orange);
                JSliderTempo.repaint();
            }
            else
            {
                JSliderTempo.setForeground(new Color(153,153,204));
                JSliderTempo.repaint();
            }

            jLabelTempo.setText(java.lang.String.valueOf(JSliderTempo.getValue()));
        }
    }

```

```

        } catch (java.lang.Exception e) {
        }
    }

    void jButtonOk_mouseClicked(java.awt.event.MouseEvent event)
    {
        jButtonOk_mouseClicked_Interaction1(event);
    }

    void jButtonOk_mouseClicked_Interaction1(java.awt.event.MouseEvent event)
    {
        try {
            saveIniFile();
            if (JCheckBoxAutomatico.isSelected())
            {
                parentDesteframe.setarValores(jTxtPaneNome.getText(), JSliderTempo.getValue(), true);
            }
            else{
                parentDesteframe.setarValores(jTxtPaneNome.getText(), JSliderTempo.getValue(), false);
            }
            this.setVisible(false);
        } catch (java.lang.Exception e) {
        }
    }

    class SymWindow extends java.awt.event.WindowAdapter
    {
        public void windowOpened(java.awt.event.WindowEvent event)
        {
            Object object = event.getSource();
            if (object == JDialogConfiguracoes.this)
                JDialogConfiguracoes_windowOpened(event);
        }

        void JDialogConfiguracoes_windowOpened(java.awt.event.WindowEvent event)
        {
            try{
                FileInputStream oArqIni = new FileInputStream(iniFileName);
                DataInputStream oDataInArqIni = new DataInputStream(oArqIni);

                jTxtPaneNome.setText(oDataInArqIni.readLine());
                //jTxtPaneRelaxacao.setText(oDataInArqIni.readLine());
                String intervaloProc = oDataInArqIni.readLine();

                JSliderTempo.setValue(Integer.valueOf(intervaloProc).intValue());
                jLabelTempo.setText(intervaloProc);
                if((JSliderTempo.getValue() <= 5) || (JSliderTempo.getValue() >= 15))
                {
                    JSliderTempo.setForeground(Color.orange);
                    JSliderTempo.repaint();
                }
                else
                {
                    JSliderTempo.setForeground(new Color(153,153,204));
                    JSliderTempo.repaint();
                }

                String sAuto = oDataInArqIni.readLine();
                if (sAuto.equalsIgnoreCase("1")) JCheckBoxAutomatico.setSelected(true);
                else JCheckBoxAutomatico.setSelected(false);

                oDataInArqIni.close();
                oArqIni.close();
            }catch(FileNotFoundException fnfe){
                com.sun.java.swing.JOptionPane.showMessageDialog(null,
                    fnfe,
                    "Error #2212",
                    com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
            }
            catch(IOException ioe)(com.sun.java.swing.JOptionPane.showMessageDialog(null,
                ioe, "Error #2212",
                com.sun.java.swing.JOptionPane.ERROR_MESSAGE);
            }
        }
    }

    /*-----*/

```

Arquivo JAboutDialog1.java - Janela com dados sobre o sistema.

```

import java.awt.*;
import com.sun.java.swing.*;

```

```

public class JAboutDialog1 extends com.sun.java.swing.JDialog
{
    public JAboutDialog1(Frame parentFrame)
    {
        super(parentFrame);
        //((INIT_CONTROLS
        setTitle("SSCC - About");
        setModal(true);
        setDefaultCloseOperation(com.sun.java.swing.JFrame.DISPOSE_ON_CLOSE);
        getContentPane().setLayout(new GridBagLayout());
        setSize(248, 94);
        setVisible(false);
        okButton.setText("OK");
        okButton.setActionCommand("OK");
        okButton.setOpaque(false);
        okButton.setMnemonic((int)'O');
        getContentPane().add(okButton, new
com.symantec.itools.awt.GridBagConstraintsD(2,1,1,1,0,0,0,java.awt.GridBagConstraints.CENTER,java.awt.GridBagConstraints.NONE,new
Insets(0,0,10,0),0,0));
        okButton.setBounds(98,59,51,25);
        aboutLabel.setHorizontalAlignment(com.sun.java.swing.SwingConstants.CENTER);
        aboutLabel.setText("SSCC is integrant part of the SSTCC Project.");
        getContentPane().add(aboutLabel, new
com.symantec.itools.awt.GridBagConstraintsD(0,0,3,1,1,0,1,0,java.awt.GridBagConstraints.CENTER,java.awt.GridBagConstraints.BOTH,new
Insets(0,0,0,0),0,0));
        aboutLabel.setBounds(0,0,248,59);
        //))

        //((REGISTER_LISTENERS
        SymWindow aSymWindow = new SymWindow();
        this.addWindowListener(aSymWindow);
        SymAction lSymAction = new SymAction();
        okButton.addActionListener(lSymAction);
        //))
    }
}

```



```

    }

    public void setVisible(boolean b)
    {
        if (b)
        {
            Rectangle bounds = (getParent()).getBounds();
            Dimension size = getSize();
            setLocation(bounds.x + (bounds.width - size.width)/2,
                        bounds.y + (bounds.height - size.height)/2);
        }

        super.setVisible(b);
    }

    public void addNotify()
    {
        // Record the size of the window prior to calling parents addNotify.
        Dimension d = getSize();

        super.addNotify();

        if (fComponentsAdjusted)
            return;
        // Adjust components according to the insets
        Insets insets = getInsets();
        setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
        Component components[] = getContentPane().getComponents();
        for (int i = 0; i < components.length; i++)
        {
            Point p = components[i].getLocation();
            p.translate(insets.left, insets.top);
            components[i].setLocation(p);
        }
        fComponentsAdjusted = true;
    }

    // Used for addNotify check.
    boolean fComponentsAdjusted = false;

    /**(DECLARE_CONTROLS
    com.sun.java.swing.JButton okButton = new com.sun.java.swing.JButton();
    com.sun.java.swing.JLabel aboutLabel = new com.sun.java.swing.JLabel();
    /**)

    class SymWindow extends java.awt.event.WindowAdapter
    {
        public void windowClosing(java.awt.event.WindowEvent event)
        {
            Object object = event.getSource();
            if (object == JAboutDialog1.this)
                JAboutDialog_windowClosing(event);
        }

        void JAboutDialog_windowClosing(java.awt.event.WindowEvent event)
        {
            JAboutDialog_windowClosing_Interaction1(event);
        }

        void JAboutDialog_windowClosing_Interaction1(java.awt.event.WindowEvent event) {
            try {
                this.setVisible(false);
            } catch (Exception e) {
            }
        }

        class SymAction implements java.awt.event.ActionListener
        {
            public void actionPerformed(java.awt.event.ActionEvent event)
            {
                Object object = event.getSource();
                if (object == okButton)
                    okButton_actionPerformed(event);
            }

            void okButton_actionPerformed(java.awt.event.ActionEvent event)
            {
                okButton_actionPerformed_Interaction1(event);
            }

            void okButton_actionPerformed_Interaction1(java.awt.event.ActionEvent event) {
                try {
                    this.setVisible(false);
                } catch (Exception e) {
                }
            }
        }
    }

    /*-----*/
    /* Arquivo sssc.idl - define a interface dos objetos distribuidos via Corba */
    module SSTCC
    {
        interface SSSCCManager
        {
            boolean recebeMensagem(in string alertMessage);
        };
    };

    /*-----*/

    /* Arquivos gerados pelo Visibroker for Java para implementacao dos objetos remotos
    SSSCCManager.java
    SSSCCManagerHolder.java
    SSSCCManagerHelper.java
    _st_SSSCCManager.java
    _SSCCManagerImplBase.java
    SSSCCManagerOperations.java
    _tie_SSSCCManager.java
    _example_SSSCCManager.java
    */

    /*-----*/
    /* Arquivo SSSCCManager.java */
    package SSTCC;

```

```

public interface SSCCManager extends org.omg.CORBA.Object {
    public boolean recebeMensagem(java.lang.String alertMessage);
}

/*-----*/

/* Arquivo SSCCManagerHolder.java */

package SSTCC;
/**
<p>
<ul>
<li> <b>Java Class</b> SSTCC.SSCCManagerHolder
<li> <b>Source File</b> SSTCC/SSCCManagerHolder.java
<li> <b>IDL Source File</b> ssc.idl
<li> <b>IDL Absolute Name</b> ::SSTCC::SSCCManager
<li> <b>Repository Identifier</b> IDL:SSTCC/SSCCManager:1.0
</ul>
<b>IDL definition:</b>
<pre>
    interface SSCCManager {
        boolean recebeMensagem(
            in string alertMessage
        );
    };
</pre>
</p>
*/
final public class SSCCManagerHolder implements org.omg.CORBA.portable.Streamable {
    public SSTCC.SSCCManager value;
    public SSCCManagerHolder() {
    }
    public SSCCManagerHolder(SSTCC.SSCCManager value) {
        this.value = value;
    }
    public void _read(org.omg.CORBA.portable.InputStream input) {
        value = SSCCManagerHelper.read(input);
    }
    public void _write(org.omg.CORBA.portable.OutputStream output) {
        SSCCManagerHelper.write(output, value);
    }
    public org.omg.CORBA.TypeCode _type() {
        return SSCCManagerHelper.type();
    }
}

/*-----*/

/* Arquivo SSCCManagerHelper.java */
package SSTCC;
/**
<p>
<ul>
<li> <b>Java Class</b> SSTCC.SSCCManagerHolder
<li> <b>Source File</b> SSTCC/SSCCManagerHolder.java
<li> <b>IDL Source File</b> ssc.idl
<li> <b>IDL Absolute Name</b> ::SSTCC::SSCCManager
<li> <b>Repository Identifier</b> IDL:SSTCC/SSCCManager:1.0
</ul>
<b>IDL definition:</b>
<pre>
    interface SSCCManager {
        boolean recebeMensagem(
            in string alertMessage
        );
    };
</pre>
</p>
*/
final public class SSCCManagerHelper implements org.omg.CORBA.portable.Streamable {
    public SSTCC.SSCCManager value;
    public SSCCManagerHelper() {
    }
    public SSCCManagerHelper(SSTCC.SSCCManager value) {
        this.value = value;
    }
    public void _read(org.omg.CORBA.portable.InputStream input) {
        value = SSCCManagerHelper.read(input);
    }
    public void _write(org.omg.CORBA.portable.OutputStream output) {
        SSCCManagerHelper.write(output, value);
    }
    public org.omg.CORBA.TypeCode _type() {
        return SSCCManagerHelper.type();
    }
}

/*-----*/

/* Arquivo _st_SSCCManager.java */

package SSTCC;
/**
<p>
<ul>
<li> <b>Java Class</b> SSTCC._st_SSCCManager
<li> <b>Source File</b> SSTCC/_st_SSCCManager.java
<li> <b>IDL Source File</b> ssc.idl
<li> <b>IDL Absolute Name</b> ::SSTCC::SSCCManager
<li> <b>Repository Identifier</b> IDL:SSTCC/SSCCManager:1.0
</ul>
<b>IDL definition:</b>
<pre>
    interface SSCCManager {
        boolean recebeMensagem(
            in string alertMessage
        );
    };
</pre>
</p>
*/
public class _st_SSCCManager extends org.omg.CORBA.portable.ObjectImpl implements SSTCC.SSCCManager {
    protected SSTCC.SSCCManager _wrapper = null;
    public SSTCC.SSCCManager _this() {
        return this;
    }
    public java.lang.String[] _ids() {
        return __ids;
    }
    private static java.lang.String[] __ids = {

```



```

    };
  </pre>
</p>
</p>
*/
public interface SSSCCManagerOperations {
  /**
  <p>
  Operation: <b>::SSTCC::SSCCManager::recebaMensagem</b>.
  <pre>
    boolean recebaMensagem(
      in string alertMessage
    );
  </pre>
  </p>
  </p>
  */
  public boolean recebaMensagem(
    java.lang.String alertMessage
  );
}

/*-----*/

/* Arquivo _tie_SSSCCManager.java */

package SSTCC;
/**
<p>
<ul>
<li> <b>Java Class</b> SSTCC._tie_SSSCCManager
<li> <b>Source File</b> SSTCC/_tie_SSSCCManager.java
<li> <b>IDL Source File</b> sssc.idl
<li> <b>IDL Absolute Name</b> ::SSTCC::SSCCManager
<li> <b>Repository Identifier</b> IDL:SSTCC/SSCCManager:1.0
</ul>
<b>IDL definition:</b>
<pre>
  interface SSSCCManager {
    boolean recebaMensagem(
      in string alertMessage
    );
  };
</pre>
</p>
</p>
*/
public class _tie_SSSCCManager extends SSTCC._SSCCManagerImplBase {
  private SSTCC.SSSCCManagerOperations _delegate;
  public _tie_SSSCCManager(SSTCC.SSSCCManagerOperations delegate, java.lang.String name) {
    super(name);
    this._delegate = delegate;
  }
  public _tie_SSSCCManager(SSTCC.SSSCCManagerOperations delegate) {
    this._delegate = delegate;
  }
  public SSTCC.SSSCCManagerOperations _delegate() {
    return this._delegate;
  }
  /**
  <p>
  Operation: <b>::SSTCC::SSCCManager::recebaMensagem</b>.
  <pre>
    boolean recebaMensagem(
      in string alertMessage
    );
  </pre>
  </p>
  </p>
  */
  public boolean recebaMensagem(
    java.lang.String alertMessage
  ) {
    return this._delegate.recebaMensagem(
      alertMessage
    );
  }
}

/*-----*/

/* Arquivo _example_SSSCCManager.java */

package SSTCC;
/**
<p>
<ul>
<li> <b>Java Class</b> SSTCC._example_SSSCCManager
<li> <b>Source File</b> SSTCC/_example_SSSCCManager.java
<li> <b>IDL Source File</b> sssc.idl
<li> <b>IDL Absolute Name</b> ::SSTCC::SSCCManager
<li> <b>Repository Identifier</b> IDL:SSTCC/SSCCManager:1.0
</ul>
<b>IDL definition:</b>
<pre>
  interface SSSCCManager {
    boolean recebaMensagem(
      in string alertMessage
    );
  };
</pre>
</p>
</p>
*/
public class _example_SSSCCManager extends SSTCC._SSCCManagerImplBase {
  /** Construct a persistently named object. */
  public _example_SSSCCManager(java.lang.String name) {
    super(name);
  }
  /** Construct a transient object. */
  public _example_SSSCCManager() {
    super();
  }
  /**
  <p>
  Operation: <b>::SSTCC::SSCCManager::recebaMensagem</b>.
  <pre>
    boolean recebaMensagem(
      in string alertMessage
    );
  </pre>
  </p>
  </p>
  */
  public boolean recebaMensagem(
    java.lang.String alertMessage
  ) {

```



```

//adiciona o label ao frame
add(lbl);

//define o tratador de eventos de janela do
frame como sendo o frame
addWindowListener(this);
}

//-----
public void actionPerformed(ActionEvent e)
{
    //verifica se o source do evento foi o
    botao btnParar
    if (e.getSource() == btnParar)
    {
        //esconde a janela
        setVisible(false);
        System.exit(0);
    }
    else
    if (e.getSource() ==
    btnIniciar)
    {
        lbl.setText("Inicializando...");
        try
        {
            leCDR();
        }
        catch(Exception
        erro)
        {
            lbl.setText("Erro na Leitura do CDR.");
        }
    }
}

//-----
public void windowActivated(WindowEvent e)
{
}

//-----
public void windowDeactivated(WindowEvent e)
{
}

//-----
public void windowClosing(WindowEvent e)
{
    setVisible(false);
    System.exit(0);
}

//-----
public void windowClosed(WindowEvent e)
{
}

//-----
public void windowOpened(WindowEvent e)
{
}

//-----
public void windowIconified(WindowEvent e)
{
}

//-----
public void windowDeiconified(WindowEvent e)
{
}

//-----
public static void main(String args[])
{
    AdaptadorCT2FNS oAdaptador = new
    AdaptadorCT2FNS();
    oAdaptador.setVisible(true);
}

//-----
public void leCDR() throws Exception
{
    //declara as variaveis que serao lidas do
    CDR
    String duracao, dataInicio, horaInicio,
    numDeA, numDeB, localDestino;
    double valor;
    //linha onde sera guardado um CDR
    String CDR;
    //associa o arquivo do CDR ao objeto
    arquivo

```

```

DataInputStream arquivo = new
DataInputStream(new FileInputStream("cdr.txt"));

//loop que fica lendo o CDR
CDR = arquivo.readLine();
do
{
    //atribui os campos do CDR as
    variaveis
    numDeA = CDR.substring(0,9);
    numDeB = CDR.substring(9,18);
    dataInicio = CDR.substring
    (18,26);
    horaInicio = CDR.substring
    (26,32);
    duracao = CDR.substring
    (32,38);
    lbl.setText("Lendo CDR...");
    //verifica se o telefone que
    efetuou a ligacao eh usuario SETWeb
    if (ehUsuarioSETWeb(numDeA))
    {
        //busca o nome da
        cidade do numDeB
        localDestino =
        buscaLocalDestino(numDeB.substring(2,5));
        //calcula o valor da
        ligacao lida
        valor =
        calculaValorLigacao(dataInicio, horaInicio, duracao);
        //gera um registro
        //geraCDRfiltrado
        (numDeA, numDeB, dataInicio, horaInicio, duracao, localDestino,
        valor);
    }
    while((CDR = arquivo.readLine()) != null);
}

//-----
public boolean ehUsuarioSETWeb(String numOrigem)
{
    //define inicialmente o no. de registros da
    consulta em zero
    int numReg = 0;
    //define uma variavel que vai guardar o
    valor retornado
    boolean resultado = false;
    //conecta ao Banco de Dados Central48T2FNS
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection conBD =
        DriverManager.getConnection("jdbc:odbc:Central48T2FNS", "", "");
        Statement stmtBD =
        conBD.createStatement();
        String sql = "SELECT
        cadSETI48T2FNS.numTelefone FROM cadSETI48T2FNS WHERE
        (cadSETI48T2FNS.numTelefone) = '" + numOrigem + "'";
        ResultSet rs =
        stmtBD.executeQuery(sql);
        if (rs.next())
        {
            numReg++;
        }
        if (numReg != 0)
        {
            resultado = true;
        }
    }
    catch(Exception e)
    {
        lbl.setText("Erro:
        "+e.toString());
        System.out.println
        (e.toString());
    }
    return resultado;
}

//-----
public String buscaLocalDestino(String prefixoDestino)
{
    //define uma variavel que vai guardar o
    valor retornado
    String resultado = "";
    int numReg = 0;
    //conecta ao Banco de Dados Central48T2FNS
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection conBD =
        DriverManager.getConnection("jdbc:odbc:Central48T2FNS", "", "");
        Statement stmtBD =
        conBD.createStatement();
        String sql = "SELECT
        LocalidadesT2FNS.cidade FROM LocalidadesT2FNS WHERE
        (LocalidadesT2FNS.prefixo) = '" + prefixoDestino + "'";
        ResultSet rs =
        stmtBD.executeQuery(sql);
        rs.next();
        resultado =
        rs.getString("cidade");
    }
}

```



```

}
}

//-----
public static void main(String args[])
{
    AdaptadorCT3FNS oAdaptador = new
    AdaptadorCT3FNS();
    oAdaptador.setVisible(true);
}

//-----

public void leCDR() throws Exception
{
    //declara as variaveis que serao lidas do
    CDR
    String duracao, dataInicio, horaInicio,
    numDeA, numDeB, localDestino;
    Double valor;

    //linha onde sera guardado um CDR
    String CDR;

    //associa o arquivo do CDR ao objeto
    DataInputStream arquivo = new
    DataInputStream(new FileInputStream("cdrCT3FNS.txt"));

    //loop que fica lendo o CDR
    CDR = arquivo.readLine();
    do
    {
        //atribui os campos do CDR as
        variaveis
        numDeA = CDR.substring(0,9);
        numDeB = CDR.substring(9,18);
        dataInicio = CDR.substring
        (18,26);
        horaInicio = CDR.substring
        (26,32);
        duracao = CDR.substring
        (32,38);

        labell.setText("Lendo
        CDRs...");

        //verifica se o telefone que
        efetuou a ligacao eh usuario SETWeb
        if (ehUsuarioSETWeb(numDeA))
        {
            //busca o nome da
            cidade do numDeB
            labell.setText("Buscando cidade destino...");
            localDestino =
            buscaLocalDestino(numDeB.substring(0,2), numDeB.substring(2,5));

            //calcula o valor da
            ligacao lida
            labell.setText("Calculando valor da ligacao...");
            valor =
            calculaValorLigacao(dataInicio, horaInicio, duracao);

            //gera um registro
            no BD com as ligacoes ja calculadas
            labell.setText("Gerando CDR filtrado...");
            geraCDRFiltrado
            (numDeA, numDeB, dataInicio, horaInicio, duracao, localDestino,
            valor);
        }
        while((CDR = arquivo.readLine()) != null);
        labell.setText("Finalizou leitura.");
    }

//-----

public boolean ehUsuarioSETWeb(String numOrigem)
{
    //define inicialmente o no. de registros da
    consulta em zero
    int numReg = 0;

    //define uma variavel que vai guardar o
    valor retornado
    boolean resultado = false;

    //conecta ao Banco de Dados Central4T3FNS
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection conBD =
        DriverManager.getConnection("jdbc:odbc:Central4T3FNS", "", "");
        Statement stmtBD =
        conBD.createStatement();

        String sql = "SELECT
        cadSETI4T3FNS.numTelefone FROM cadSETI4T3FNS WHERE
        (cadSETI4T3FNS.numTelefone) = " + numOrigem + """;
        ResultSet rs =
        stmtBD.executeQuery(sql);

        if (rs.next())
        {
            numReg++;
        }
        if (numReg != 0)
        {
            resultado = true;
        }
    }
    catch(Exception e)
    {
}
}
}

```

```

labell.setText("Erro:
"+e.toString());
return resultado;
}

//-----

public String buscaLocalDestino(String codAreaDestino,
String prefixoDestino)
{
    //define uma variavel que vai guardar o
    valor retornado
    String resultado = "";

    //conecta ao Banco de Dados Central4T3FNS
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection conBD =
        DriverManager.getConnection("jdbc:odbc:Central4T3FNS", "", "");
        Statement stmtBD =
        conBD.createStatement();

        String sql = "SELECT
        LocalidadesT3FNS.cidade FROM LocalidadesT3FNS WHERE
        (LocalidadesT3FNS.cdd) = " + codAreaDestino + "" AND
        (LocalidadesT3FNS.prefixo) = " + prefixoDestino + """;
        ResultSet rs =
        stmtBD.executeQuery(sql);

        rs.next();
        resultado =
        rs.getString("cidade");
    }
    catch(Exception e)
    {
        labell.setText("Erro:
"+e.toString());
return resultado;
}

//-----

public Double calculaValorLigacao(String dataI, String
horaI, String duracaoI)
{
    int numReg = 0;
    double valor = 0.00;
    boolean sai = false;
    boolean terminou = false;
    String hiTabela = "";
    String hfTabela = "";
    String segundos = "";
    String dia = "";
    String duracaoFaixa = "";
    int mt = 0;
    int st = 0;
    int hrt = 0;
    int mil = 0;
    int sil = 0;
    int hril = 0;
    long durLigacao, hfCall, hiCall;
    long horaFinalTabela = 24;

    //cria objetos da classe Date
    java.util.Date inicioLigacao = null;
    java.util.Date fimLigacao = null;

    //busca os dados referentes a data e hora
    de inicio da ligacao
    inicioLigacao = buscaDadosInicioI (dataI,
    horaI);

    //obtem o dia do mes de inicio da ligacao
    int diaInicioI = inicioLigacao.getDate ();

    //busca os dados referentes a data e hora
    de fim da ligacao
    fimLigacao = buscaDadosfimI (inicioLigacao,
    duracaoI);

    //obtem o dia do mes de inicio da ligacao
    int diaFimI = fimLigacao.getDate ();

    //busca o dia da semana de inicio da
    ligacao
    int diaSemInicio = buscaDiaSemanaInicioI
    (inicioLigacao);

    //obtem hora, minuto, segundo da data e
    hora final da ligacao
    int hora = fimLigacao.getHours();
    int minuto = fimLigacao.getMinutes();
    int segundo = fimLigacao.getSeconds();

    //declara as variaveis de hora, min e
    segundo em string
    String strHora, strMin, strSeg;

    //coloca um zero a esquerda do string qdo
    necessario
    if (hora < 10)
    {
        strHora = "0" +
        Integer.toString(hora);
    }
    else
    {
        strHora =
        Integer.toString(hora);
    }

    if (minuto < 10)
    {
}
}
}

```



```

Integer.toString(minuto);
    }
    else
    {
        strMin =
Integer.toString(minuto);
    }
    if (segundo < 10)
    {
        strSeg = "0" +
Integer.toString(segundo);
    }
    else
    {
        strSeg =
Integer.toString(segundo);
    }
    //concatena hora, minuto, segundo final da
    String hfLigacao = strHora + strMin +
strSeg;

    //conecta ao Banco de Dados de
    Central4T3FNS (Tabela Tarifas)
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        DriverManager.getConnection("jdbc:odbc:Central4T3FNS", "", "");
        Statement stmtBD =
conBD.createStatement();

        //atribui a instrucao sql a uma
        String sql = "SELECT * FROM
TarifasT3FNS ORDER BY dia, horarioInicial";

        //obtem a tabela de tarifas via
        ResultSet rs =
stmtBD.executeQuery(sql);

        //obtem a primeira linha de
        //obtem a tarifa do inicio da
        while ( sai != true)
        {
            rs.next();
            //Obtem os campos da
            //consulta
            dia =
rs.getString("dia");
            hiTabela =
rs.getString("horarioInicial");
            hfTabela =
rs.getString("horarioFinal");

            // converte o dia da
            int diaT =
Integer.valueOf(dia).intValue();

            //converte a hora da
            long horaLigLong =
Long.valueOf(horaL).longValue();

            //converte a hora
            long hiTabLong =
Long.valueOf(hiTabela).longValue();

            //converte a hora
            long hfTabLong =
Long.valueOf(hfTabela).longValue();

            //obtem a primeira
            if ((diaT ==
diaSemInicio) && (hiTabLong <= horaLigLong) && (horaLigLong <
hfTabLong))
            {
                segundos
= rs.getString("segundos");
                duracaoFaixa = rs.getString("duracaoFaixa");
                sai =
true;
            }

            //converte a duracao da ligacao
            durLigacao =
Long.valueOf(duracaoL).longValue();

            //converte a hora final da
            hfCall =
Long.valueOf(hfLigacao).longValue();

            //converte a hora inicial da
            hiCall =
Long.valueOf(horaL).longValue();

            while (terminou != true)
            {
                //converte a faixa
                long durFaixaL =
Long.valueOf(duracaoFaixa).longValue();

                //converte a hora
                long hfTabela =
Long.valueOf(hfTabela).longValue();

                //converte os
                double
segundosTabela = Double.valueOf(segundos).doubleValue();

                horaFinalTabela =
hfTabela;

                if ((hfTabela ==
240000) && (diaIniciol != diaFimL))
                {
                    horaFinalTabela = 0;
                    diaIniciol++;
                }

                if ((hfCall <
horaFinalTabela) && (durLigacao <= durFaixaL))
                {
                    valor =
durLigacao/segundosTabela*pulso;
                    terminou
= true;
                }
                else
                {
                    //hfTabela - hiCall
                    //obtem
                    a hora final da tabela e a hora
                    //inicial da ligacao em milisegundos
                    String
hft = Long.toString(hfTabela);
                    if
                    (hft.length() == 5)
                    {
                        st =
Integer.valueOf(hft.substring(3,5)).intValue();
                        mt =
Integer.valueOf(hft.substring(1,3)).intValue();
                        hrt =
Integer.valueOf(hft.substring(0,1)).intValue();
                    }
                    if
                    (hft.length() == 6)
                    {
                        st =
Integer.valueOf(hft.substring(4,6)).intValue();
                        mt =
Integer.valueOf(hft.substring(2,4)).intValue();
                        hrt =
Integer.valueOf(hft.substring(0,2)).intValue();
                    }
                    String
hil = Long.toString(hiCall);
                    int
tamhil = hil.length();
                    switch
                    (tamhil)
                    {
                        case
                    4:
                        sil =
Integer.valueOf(hil.substring(2,4)).intValue();
                        mil =
Integer.valueOf(hil.substring(0,2)).intValue();
                        hril = 0; break;
                    case
                    5:
                        sil =
Integer.valueOf(hil.substring(3,5)).intValue();
                        mil =
Integer.valueOf(hil.substring(1,3)).intValue();
                        hril =
Integer.valueOf(hil.substring(0,1)).intValue();
                        break;
                    case
                    6:
                        sil =
Integer.valueOf(hil.substring(4,6)).intValue();
                        mil =
Integer.valueOf(hil.substring(2,4)).intValue();
                        hril =
Integer.valueOf(hil.substring(0,2)).intValue();
                        break;
                    case
                    3:
                        sil =
Integer.valueOf(hil.substring(1,3)).intValue();
                        mil =
Integer.valueOf(hil.substring(0,1)).intValue();
                        hril = 0; break;
                    case
                    2:
                        sil =
Integer.valueOf(hil.substring(0,2)).intValue();
                        mil =
Integer.valueOf(hil.substring(0,2)).intValue();
                        hril = 0; break;
                    case
                    0:
                        hril = 0; break;
                }
            }
        }
    }
}

```

```

case
1:
0:
0:
    hrt,mt,st);
    java.util.Date hFimTab = new java.util.Date(0,0,0,
    java.util.Date hILig = new java.util.Date(0,0,0,
    hril,mil,sil);

    //calcula o valor da ligacao
    valor =
    valor + (((hFimTab.getTime() -
    hILig.getTime())/1000)/segundosTabela) * pulso);

    if
    (dia.equals("6") && hFTabela.equals("240000"))
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection conBDSab =
        DriverManager.getConnection("jdbc:odbc:Central4T3FNS", "", "");
        Statement stmtBDSab = conBDSab.createStatement();
        String sql = "SELECT * FROM TarifasT3FNS ORDER BY dia,
        horarioInicial";
        rs = stmtBDSab.executeQuery(sql);
        rs.next();
    }
    else
    {
        rs.next();
    }

    = rs.getString("horarioInicial");
    = rs.getString("horarioFinal");
    = rs.getString("segundos");
    duracaoFaixa = rs.getString("duracaoFaixa");
    dia =
    rs.getString("dia");

    //converte a hora inicial da tabela para long
    long
    hiTable = Long.valueOf(hiTabela).longValue();

    //redefine a hora inicial da ligacao (prox pedaco)
    //com a
    hora inicial da faixa de tarifas da tabela
    //e a
    duracao da ligacao
    hiCall =
    hiTable;

    durLigacao = durLigacao - ((hFimTab.getTime() -
    hILig.getTime())/1000);
    }
    catch(Exception e)
    {
        e.toString();
        label1.setText("Erro: "+
        e.toString());
    }
    Double valor = new Double (valor*100);
    int vl = valor.intValue();
    Double valorL = new Double (vl/100.00);
    return valorL;
}

//-----
public int buscaDiaSemanaIniciol (java.util.Date
iniciol)
{
    int resultado = 0;
    //obtem um inteiro que representa o dia da
    semana da data
    resultado = iniciol.getDay();

    return resultado;
}

//-----
public java.util.Date buscaDadosIniciol (String dataL,
String horal)
{
    // Separa a data de inicio da ligacao em
    int dia =
    Integer.valueOf(dataL.substring(0,2)).intValue();
    int mes =
    Integer.valueOf(dataL.substring(2,4)).intValue();

    int ano =
    Integer.valueOf(dataL.substring(4,8)).intValue();

    // Separa a hora de inicio da ligacao em
    hora, minuto e segundo
    int seg =
    Integer.valueOf(horal.substring(4,6)).intValue();
    int min =
    Integer.valueOf(horal.substring(6,8)).intValue();
    int hora =
    Integer.valueOf(horal.substring(0,2)).intValue();

    //cria objeto da classe Date com dados do
    inicio da ligacao
    java.util.Date inicioLigacao = new
    java.util.Date(ano-1900, mes-1, dia, hora, min, seg);

    return inicioLigacao;
}

//-----
public java.util.Date buscaDadosFimL (java.util.Date
iniciol,
String duracaoL)
{
    //cria objetos da classe Date
    java.util.Date fimLigacao = new
    java.util.Date();
    //transforma a duracao de string para long
    long duracao =
    Long.valueOf(duracaoL).longValue();
    //obtem os dados do fim da ligacao a partir
    da hora inicial
    //e duracao em ms
    fimLigacao.setTime(iniciol.getTime()+
    duracao*1000 );
}

//-----
public void geraCDRfiltrado (String numDeA, String
numDeB,
String dataInicio, String horaInicio, String duracao,
String localDestino, Double valor)
{
    //conecta ao Banco de Dados Central4T3FNS
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection conBD =
        DriverManager.getConnection("jdbc:odbc:Central4T3FNS", "", "");
        Statement stmtBD =
        conBD.createStatement();
        String sql = "INSERT INTO
        CDRfiltrado VALUES (" + numDeA + ", " + classe + ", " +
        dataInicio + ", " + horaInicio + ", " + duracao + ", " +
        localDestino + ", " + numDeB + ", " + valor + ")";
        stmtBD.executeUpdate(sql);
    }
    catch(Exception e)
    {
        label1.setText("Erro:
        "+e.toString());
    }
}

//-----
// Classe Adaptador Central Embratel
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.sql.*;
import java.util.*;
import java.lang.*;
import java.text.*;

class AdaptadorEmbratel extends Frame implements ActionListener,
WindowListener
{
    //declara atributos da central
    double pulso = 0.08;
    String classe;

    //declara objetos de interface
    private Button btnParar;
    private Button btnIniciar;
    private Label label1;

    AdaptadorEmbratel ()
    {
        //chama o construtor da superclasse
        super();

        //define o titulo da janela
        setTitle("Módulo Adaptador - Central
        Embratel");

        //define o posicionamento e dimensoes da
        janela
        setBounds(0,0,323,144);

        //define o layout
        setLayout(null);
        setResizable(false);

        //define o fundo do frame
}

```

```

setBackground(Color.lightGray);

//cria os botoes
btnIniciar = new Button("Iniciar");
btnParar = new Button("Parar");

//define o posicionamento e dimensoes dos
botoes
btnIniciar.setBounds(36,36,108,28);
btnParar.setBounds(180,36,108,28);

//define o tratador do evento tipo
como sendo o frame
ActionEvent dos botoes
btnIniciar.addActionListener(this);
btnParar.addActionListener(this);

//adiciona os botoes ao frame
add(btnIniciar);
add(btnParar);

//cria o label
label1 = new Label();

//define o posicionamento e dimensoes do
label
label1.setBounds(36,84,252,24);

//define o alinhamento do texto do label
label1.setAlignment(Label.CENTER);

//define o texto do label
label1.setText("Adaptador não
inicializado.");

//adiciona o label ao frame
add(label1);

//define o tratador de eventos de janela do
frame como sendo o frame
addWindowListener(this);
}

//-----
public void actionPerformed(ActionEvent e)
{
    //verifica se o source do evento foi o
    botao btnParar
    if (e.getSource() == btnParar)
    {
        //esconde a janela
        setVisible(false);
        System.exit(0);
    }
    else
        if (e.getSource() ==
        btnIniciar)
        {
            label1.setText("Inicializando...");
            try
            {
                leCDR();
            }
            catch(Exception
            erro)
            {
                label1.setText("Erro na Leitura do CDR.");
            }
        }
    }

//-----
public void windowActivated(WindowEvent e)
{
}

//-----
public void windowDeactivated(WindowEvent e)
{
}

//-----
public void windowClosing(WindowEvent e)
{
    setVisible(false);
    System.exit(0);
}

//-----
public void windowClosed(WindowEvent e)
{
}

//-----
public void windowOpened(WindowEvent e)
{
}

//-----
public void windowIconified(WindowEvent e)
{
}

```

```

//-----
public void windowDeiconified(WindowEvent e)
{
}

//-----
public static void main(String args[])
{
    AdaptadorCEmbratel oAdaptador = new
    AdaptadorCEmbratel();
    oAdaptador.setVisible(true);
}

//-----
public void leCDR() throws Exception
{
    //declara as variaveis que serao lidas do
    CDR
    String duracao, dataInicio, horaInicio,
    numDeA, numDeB, codInter;
    Double valor=new Double(0.00);
    String localDestino;
    String codPais = "";

    //linha onde sera guardado um CDR
    String CDR;

    //associa o arquivo do CDR ao objeto
    arquivo
    DataInputStream arquivo = new
    DataInputStream(new FileInputStream("cdrEmbratel.txt"));

    //loop que fica lendo o CDR
    CDR = arquivo.readLine();
    do
    {
        //seta local destino com string
        vazio
        localDestino = " ";

        //atribui os campos do CDR as
        variaveis
        numDeA = CDR.substring(0,9);
        codInter = CDR.substring
        (9,11);

        if (codInter.equals("00"))
        {
            classe = "DD1";
            int tamCDR =
            CDR.length();
            duracao =
            CDR.substring(tamCDR-6,tamCDR);
            horaInicio =
            CDR.substring(tamCDR-12,tamCDR-6);
            dataInicio =
            CDR.substring(tamCDR-20,tamCDR-12);
            numDeB =
            CDR.substring(9,tamCDR-20);
            codPais =
            CDR.substring(11,tamCDR-29);
        }
        else
        {
            classe = "DDD";
            numDeB = CDR.substring(9,18);
            dataInicio = CDR.substring
            (18,26);
            horaInicio = CDR.substring
            (26,32);
            duracao = CDR.substring
            (32,38);
        }
        label1.setText("Lendo
        CDRs...");

        //verifica se o telefone que
        efetuou a ligacao eh usuario SETWeb
        if (ehUsuarioSETWeb(numDeA))
        {
            if
            {
                //busca o nome da
                cidade do numDeB se a ligacao for nacional

                label1.setText("Buscando cidade destino...");
                localDestino =
                buscaLocalDestino(numDeB.substring(0,2), numDeB.substring(2,5));

                //calcula o valor da
                ligacao lida

                label1.setText("Calculando valor da ligacao...");
                valor =
                calculaValorLigacaoNac(dataInicio, horaInicio, duracao);
            }
            else
            {
                if
                {
                    label1.setText("Calculando valor da ligacao...");
                    valor =
                    calculaValorLigacaoInt(dataInicio, horaInicio, duracao, codPais);
                }
            }
        }

        //gera um registro
        no BD com as ligacoes ja calculadas

        label1.setText("Gerando CDR filtrado...");
    }
}

```













```

        CEP = rs.getString(6);
        Dace = rs.getString(7);
    }
    rs.close();
    stmt.close();
    con.close();

    CPFTF.setText (CPF);
    nomeTF.setText (Nome);
    enderecoTF.setText (Ender);
    cidadeTF.setText (City);
    estadoTF.setText (State);
    CEPTF.setText (CEP);
    dataTF.setText (Date);

} catch (Exception e) {
    System.out.println ("Erro fatal na consulta");
}
return true;
}

public boolean cadDados() {
    try {
        KeyPairGenerator keyGen = KeyPairGenerator.getInstance ("DSA",
"Sun");
        SecureRandom random = SecureRandom.getInstance ("SHA1PRNG",
"Sun");

        KeyPair pair = keyGen.generateKeyPair();
        priv = pair.getPrivate();
        pub = pair.getPublic();
    } catch (Exception e) {
        System.err.println ("Exceção ativada: " + e.toString());
    }

    try {
        String CPF = CPFTF.getText().trim();
        String Nome = nomeTF.getText().trim();
        String Ender = enderecoTF.getText().trim();
        String City = cidadeTF.getText().trim();
        String State = estadoTF.getText().trim();
        String CEP = CEPTF.getText().trim();
        String Date = dataTF.getText().trim();

        Connection con =
DriverManager.getConnection("jdbc:odbc:conta", "conta", "conta");
        Statement stmt = con.createStatement();

        ResultSet rs = stmt.executeQuery("INSERT
INTO conta (CPF,nome,endereco,cidade,estado,CEP,data,chvp,chnpb)
values
('"+CPF+"','"+Nome+"','"+Ender+"','"+City+"','"+State+"','"+CEP+"',
 '"+Date+"','"+priv+"','"+pub+"')");
        stmt.close();
        con.close();
        System.out.println ("Operacao bem sucedida
");

    } catch (Exception e) {
        System.out.println ("Erro!");
    }

    CPFTF.setText ("");
    nomeTF.setText ("");
    enderecoTF.setText ("");
    cidadeTF.setText ("");
    estadoTF.setText ("");
    CEPTF.setText ("");
    dataTF.setText ("");

    return true;
}

/* essa é a aplicação que o usuário vai utilizar em casa. A
politica será a seguinte: em casa, o usuário recebe sua chave
privada e sua assinatura para uma senha sugerida. De posse dessa
assinatura, o usuário fornece a senha que é assinada e verificada
localmente pelo servidor que já possui todos os dados */

import java.lang.*;
import java.awt.*;
import java.sql.*;
import java.io.*;
import java.net.*;
import java.security.*;
import java.security.spec.*;
import java.applet.*;

// o usuário pode cadastrar sua senha e assiná-la, usando a chave
privada

public class acesso extends Applet {

    protected Label CPFL, senhaL;
    protected TextField CPFTF, senhaTF;
    Button OK, Cancel;
    Panel p1,p2;
    URL url;
    String realSign, sig;
    PrivateKey priv;

    static {
        try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    public static void main (String args[]) {
        Decasa teste = new Decasa();
        teste.init();
    }

    public void init () {
        // setLayout (new GridLayout(2,2));
        setBackground (Color.black);
        setForeground (Color.white);

        p1 = new Panel();
        p2 = new Panel();

        OK = new Button ("OK ");
        Cancel = new Button ("Cancel ");

        CPFL = new Label ("CPF:");
        senhaL = new Label ("Senha:");

        CPFTF = new TextField (20);
        senhaTF = new TextField (20);
        senhaTF.setEchoCharacter ("*");

        p1.add (OK);
        p1.add (Cancel);

        p2.setLayout (new GridLayout (3,2));

        p2.add(CPFL);
        p2.add(CPFTF);
        p2.add(senhaL);
        p2.add(senhaTF);

        add(p2,"North");
        add(p1,"South");

        senhaTF.requestFocus();
        setSize (500,300);
        show();
    }

    public boolean handleEvent (Event evt) {
        Object pEvtSource = evt.target;
        if (pEvtSource == OK && evt.id == Event.ACTION_EVENT) {
            return this.assina ();
        }
        else if (pEvtSource == Cancel && evt.id ==
Event.ACTION_EVENT) {
            CPFTF.setText("");
            senhaTF.setText("");
            try {
                this.url = new URL
("file:///d:/alex/extrato/pitti.html");
                getAppletContext().showDocument(this.url);
            } catch (MalformedURLException e) {
                System.out.println("Bad URL: " + this.url);
            }
            else if (pEvtSource == this && evt.id ==
Event.WINDOW_DESTROY) {
                return this.sair();
            }
            return false;
        }
    }

    public boolean sair() {
        setVisible (false);
        System.exit (0);
        return true;
    }

    public boolean assina() {
        /* Para que a assinatura seja enviada, será necessária a chave
privada que gerará a assinatura. Essa informação está disponível no
BD signus.
Esse BD, é automaticamente gravado em c:\wtelec, quando da
instalação do aplicativo pelo cliente.

Por outro lado, a assinatura gerada fica guardada na variável
realSig. Esse procedimento permite que o host servidor da World
Telecom confirme a autenticidade do usuário. */

        String gosign = senhaTF.getText().trim();
        /* gosign é a senha, ou seja, o item que será assinado para
posterior verificação*/

        try {
            Connection con =
DriverManager.getConnection("jdbc:odbc:signus", "signus",
"signus");
            Statement stat = con.createStatement();

            ResultSet rs = stat.executeQuery("Select *
From signus ");
            while (rs.next())
            {
                priv = rs.getString(1);
            }

            rs.close();
            stmt.close();
            con.close();
        } catch (Exception e) {
            System.out.println ("Erro!");
        }

        try {
            Signature dsa = Signature.getInstance ("SHAwithDSA",
"Sun");
            dsa.initSign (priv);

            byte[] bl;
            gosign.getBytes(0, gosign.length(),bl, 0 );

            /* como o método update apenas recebe objetos byte[]
como entrada, é feita conversão antes */

            dsa.update (bl);
        }
    }
}

```

```

byte[] assinatura = dsa.sign();
// a assinatura enfim, é realizada

sig = new String(assinatura,0);
/* sig é transformada em string para ser comparada a
assinatura é comparada com o a assinatura já armazenada no servidor
da WTelecom */
catch (Exception e) {
    System.err.println ("Caught exception "+
e.toString());
}

try {
    String procura = CPPTF.getText().trim();

    Connection con =
DriverManager.getConnection("jdbc:odbc:conta", "conta", "conta");
    Statement stmt = con.createStatement();

    ResultSet rs = stmt.executeQuery("Select *
From conta Where (CPF = '"+procura+"')");
    while (rs.next())
    {
        realsign = rs.getString(10);
    }

    rs.close();
    stmt.close();
    con.close();
}
catch (Exception e) {
    System.out.println ("Erro!");
}

```

```

if (sig != realsign)
{
    try {
        this.url = new URL
("file:///d:/alex/extrato/pitti.html");
        getAppletContext().showDocument(this.url);
        senhaTF.setText ("");
    }
    catch (MalformedURLException e)
    {
        System.out.println("Bad URL: " + this.url);
    }
    ;
    else
    {
        try {
            this.url = new URL
("file:///d:/alex/extrato/ok.html");
            getAppletContext().showDocument(this.url);
            senhaTF.setText ("");
        }
        catch (MalformedURLException e)
        {
            System.out.println("Bad URL: " + this.url);
        }
    }
}

CPPTF.setText ("");
senhaTF.setText ("");
return true;
}
}

//-----Seguranca Java para o Sistema SetWeb-----
fim-----

```

CPF	premio	cidade
11	201	Sao Paulo
11	203	Sao Paulo
11	205	Sao Paulo
11	207	Sao Paulo
11	213	Sao Paulo
11	214	Sao Paulo
11	215	Sao Paulo
11	217	Sao Paulo
11	236	Sao Paulo
11	255	Sao Paulo
11	260	Sao Paulo
11	262	Sao Paulo
11	265	Sao Paulo
11	267	Sao Paulo
11	268	Sao Paulo
11	270	Sao Paulo
11	297	Sao Paulo
11	426	Sao Paulo
11	429	Sao Paulo
11	610	Sao Paulo
11	680	Sao Paulo
11	810	Sao Paulo
11	812	Sao Paulo
11	825	Sao Paulo
11	840	Sao Paulo
11	885	Sao Paulo
11	957	Sao Paulo
21	243	Rio de Janeiro
21	252	Rio de Janeiro
21	259	Rio de Janeiro
21	274	Rio de Janeiro
21	288	Rio de Janeiro
21	289	Rio de Janeiro

21	372	Rio de Janeiro
21	413	Rio de Janeiro
21	431	Rio de Janeiro
21	493	Rio de Janeiro
21	509	Rio de Janeiro
21	511	Rio de Janeiro
21	512	Rio de Janeiro
21	523	Rio de Janeiro
21	532	Rio de Janeiro
21	533	Rio de Janeiro
21	580	Rio de Janeiro
21	589	Rio de Janeiro
21	593	Rio de Janeiro
21	594	Rio de Janeiro
21	595	Rio de Janeiro
21	597	Rio de Janeiro
21	601	Rio de Janeiro
21	701	Rio de Janeiro
21	717	Rio de Janeiro
21	756	Rio de Janeiro
27	200	Vitoria
27	222	Vitoria
27	223	Vitoria
27	225	Vitoria
27	227	Vitoria
27	235	Vitoria
27	322	Vitoria
31	212	Belo Horizonte
31	222	Belo Horizonte
31	223	Belo Horizonte
31	224	Belo Horizonte
31	227	Belo Horizonte
31	261	Belo Horizonte
31	273	Belo Horizonte

31	274	Belo Horizonte
31	278	Belo Horizonte
31	281	Belo Horizonte
31	292	Belo Horizonte
31	299	Belo Horizonte
31	332	Belo Horizonte
31	334	Belo Horizonte
31	371	Belo Horizonte
31	388	Belo Horizonte
31	411	Belo Horizonte
31	421	Belo Horizonte
31	424	Belo Horizonte
31	428	Belo Horizonte
31	432	Belo Horizonte
31	442	Belo Horizonte
31	444	Belo Horizonte
31	446	Belo Horizonte
31	453	Belo Horizonte
31	464	Belo Horizonte
31	482	Belo Horizonte
31	486	Belo Horizonte
31	491	Belo Horizonte
51	218	Porto Alegre
51	222	Porto Alegre
51	328	Porto Alegre
51	333	Porto Alegre
51	346	Porto Alegre
61	224	Brasilia
61	226	Brasilia
61	234	Brasilia
61	248	Brasilia
61	328	Brasilia
61	346	Brasilia
61	364	Brasilia
61	381	Brasilia
61	387	Brasilia
61	500	Brasilia
61	568	Brasilia
61	591	Brasilia

62	215	Goiânia
62	225	Goiânia
62	295	Goiânia
62	846	Goiânia
71	336	Salvador
85	278	Fortaleza
85	287	Fortaleza
91	210	Belem
91	212	Belem
91	222	Belem
91	224	Belem
91	226	Belem
91	227	Belem
91	228	Belem
91	229	Belem
91	231	Belem
91	233	Belem
91	235	Belem
91	236	Belem
91	237	Belem
91	241	Belem
91	242	Belem
91	243	Belem
91	245	Belem
91	246	Belem
91	255	Belem
91	266	Belem
91	613	Belem
92	232	Manaus
92	234	Manaus
92	238	Manaus
92	622	Manaus
92	633	Manaus
92	635	Manaus
92	642	Manaus
92	656	Manaus
92	663	Manaus

Contorno-DPI	horaInicio	horaFim	VariaçãoAnual	VariaçãoSub	Quantidade
1	000000	050000	1,0717	0,0783	18000
1	050000	200000	1,3396	0,0979	54000
1	200000	240000	1,0717	0,0783	14400
27	000000	050000	2,951	0,2396	18000
27	050000	200000	3,6888	0,2995	54000
27	200000	240000	2,951	0,2396	14400
33	000000	050000	1,42	0,1378	18000
33	050000	200000	1,775	0,1722	54000
33	200000	240000	1,42	0,1378	14400

34	000000	050000	1,3779	0,1378	18000
34	050000	200000	1,7223	0,1722	54000
34	200000	240000	1,3779	0,1378	14400
52	000000	050000	1,3779	0,1296	18000
52	050000	200000	1,7223	0,162	54000
52	200000	240000	1,3779	0,1296	14400
54	000000	050000	1,3779	0,1158	18000
54	050000	200000	1,7223	0,1448	54000
54	200000	240000	1,3779	0,1158	14400
81	000000	010000	2,4989	0,2005	3600
81	010000	060000	1,9991	0,1604	18000
81	060000	130000	2,4989	0,2005	25200
81	130000	170000	1,9991	0,1604	14400
81	170000	240000	2,4989	0,2005	25200
91	000000	010000	3,6888	0,2995	3600
91	010000	060000	2,951	0,2396	18000
91	060000	130000	3,6888	0,2995	25200
91	130000	170000	2,951	0,2396	14400
91	170000	240000	3,6888	0,2995	25200
972	000000	050000	1,3779	0,1378	18000
972	050000	200000	1,7223	0,1722	54000
972	200000	240000	1,3779	0,1378	14400

Site	Non-Residential	Forest of Forest	Residence	Duration of Lease
0	000000	060000	103,1	21600
0	060000	240000	51,6	64800
1	000000	060000	103,1	21600
1	060000	070000	51,6	3600
1	070000	090000	25,8	7200
1	090000	120000	12,9	10800
1	120000	140000	25,8	7200
1	140000	180000	12,9	14400
1	180000	210000	25,8	10800
1	210000	240000	51,6	10800
2	000000	060000	103,1	21600
2	060000	070000	51,6	3600
2	070000	090000	25,8	7200
2	090000	120000	12,9	10800
2	120000	140000	25,8	7200
2	140000	180000	12,9	14400
2	180000	210000	25,8	10800
2	210000	240000	51,6	10800
3	000000	060000	103,1	21600
3	060000	070000	51,6	3600
3	070000	090000	25,8	7200
3	090000	120000	12,9	10800
3	120000	140000	25,8	7200
3	140000	180000	12,9	14400
3	180000	210000	25,8	10800
3	210000	240000	51,6	10800

4	000000	060000	103,1	21600
4	060000	070000	51,6	3600
4	070000	090000	25,8	7200
4	090000	120000	12,9	10800
4	120000	140000	25,8	7200
4	140000	180000	12,9	14400
4	180000	210000	25,8	10800
4	210000	240000	51,6	10800
5	000000	060000	103,1	21600
5	060000	070000	51,6	3600
5	070000	090000	25,8	7200
5	090000	120000	12,9	10800
5	120000	140000	25,8	7200
5	140000	180000	12,9	14400
5	180000	210000	25,8	10800
5	210000	240000	51,6	10800
6	000000	060000	103,1	21600
6	060000	070000	51,6	3600
6	070000	140000	25,8	25200
6	140000	240000	51,6	36000

## Índice Remissivo

### A

Agente ... 25, 30, 34, 102, 103, 104, 105, 106, 109  
Agrupamentos ..... 75, 78

### C

Características ..... 65, 78  
Classificação ..... 79, 113  
Clone ..... 23  
Comparação ..... 43  
CORBA ..... 10, 13, 14, 39, 87, 92, 93, 94, 95, 102,  
103, 110, 111, 118, 120, 122, 123, 136, 166

### E

Equivalência de observação ..... 43, 45, 46  
Equivalência forte ..... 43, 44, 46  
Equivalência segura ..... 43  
Especificação formal ..... 15

### G

Gauss ..... 75, 145, 148, 149, 150  
Gerência ..... 13, 14, 87, 88, 93, 110, 118  
Gerente ..... 23, 24, 25, 28, 29, 32, 34, 95, 102, 103,  
105, 106, 108

### H

Habilitação ..... 2, 17, 85, 125

### J

Java ..... 11, 87, 92, 93, 94, 95, 102, 105, 106, 111,  
118, 120, 122, 136, 194, 196  
JDK ..... 11, 87, 118, 119, 120

### K

Kohonen ..... 59, 61, 83, 141

### L

LOTOS ..... 11, 13, 14, 15, 16, 17, 19, 20, 22, 23,  
24, 25, 27, 28, 29, 30, 32, 33, 34, 35, 37, 38,  
39, 41, 42, 58

### M

Mecanismos ..... 89, 91  
Mobile ..... 10, 11, 2, 5  
Móveis ..... 1, 11, 59, 85, 126, 127  
Móvel ..... 1, 11, 125

### N

Notificações ..... 18

### O

Observational equivalence ..... 45, 46

### P

Protocolo ..... 14, 22, 43  
Prova ..... 56

### R

RBF ..... 59, 72, 75, 78, 145, 148, 150  
Recursividade de processos ..... 17  
Rede Neural ..... 82  
Redução ..... 43  
Riscos ..... 91

### S

Segurança ..... 12, 13, 14, 20, 22, 57, 58, 59, 87, 88,  
89, 93, 95, 118, 136, 137, 138  
Sem fio ..... 13, 1, 85, 125  
Serviço ..... 14, 20, 43, 93  
SETWeb ..... 12, 22, 28, 31, 32, 39, 87, 111, 112,  
113, 115, 118, 119, 122, 138, 184, 186, 190  
Simulação ..... 42  
Simulação exaustiva ..... 43  
Simulação interativa ..... 43  
SIPI ..... 12, 22, 32, 37, 38, 39, 87, 102, 122, 166  
Sistemas Distribuídos ..... 1, 2, 13, 14  
SSCC ..... 12, 22, 23, 27, 28, 37, 39, 87, 102, 103,  
122, 166  
SSTCC ..... 12, 14, 20, 21, 22, 27, 38, 39, 41, 46, 56,  
59, 87  
Subscrição ..... 13, 1

### T

TDF ..... 19  
Telecomunicações ..... 1, 12, 13, 14, 59, 88, 96,  
136, 138  
Teste ..... 42, 68, 81, 82, 150  
Treinamento ..... 67, 75, 80, 141

### V

Validação ..... 14, 15, 42, 46, 58, 112, 140  
Verificação ..... 42, 43, 44

## Referências Bibliográficas

- AIDAROUS, S., PLEVYAK, T. *Telecommunications network management: technologies and implementations*. Piscataway, NJ, USA: IEEE Press, 1998. 352 p. ISBN 0-7803-3454-X.
- ALEXANDER, D. S., ARBAUGH, W.A., KEROMYTIS, A.D., SMITH, J.M. Safety and security of programmable networks infrastructures, *IEEE Communications Magazine*, New York, v.36, n.10, p.84-92, out.1998.
- ANEROUSIS, N. Scalable management services using Java and the World Wide Web. In: DSOM'98 – DISTRIBUTED SYSTEMS: OPERATIONS & MANAGEMENT. University of Delaware, USA, 26-28 oct. 1998, p. 79-90.
- APOSTOLOPOULOS, t., daskalou, v. The role of the time parameter in a network security management model. In: IEEE ISCC'97 – INTERNATIONAL SYMPOSIUM ON COMPUTER COMMUNICATIONS. Alexandria, Egito, 1-3 jul. 1997, p. 528-532.
- ARNOLD, A. Systèmes de transitions finis et sémantique des processus communicants. In: TECHNIQUE et science informatiques. Université Bordeaux, 1989, p.193-216.
- BARRADAS, O. Você e as telecomunicações. *Telebrasil* – Associação Brasileira de Telecomunicações. Rio de Janeiro, RJ: Editora Interciência, 1995. 277p.
- BARRETO, J.M. *Inteligência artificial no limiar do século XXI*. Florianópolis: Duplic, 1997.
- BONIFÁCIO, J. M., MOREIRA, E. S., CANSIAN, A. M., CARVALHO, A. C. P. L. F. Um ambiente de segurança distribuído para integração de firewalls com sistemas de detecção de intrusão. In: SBRC'98 – SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES. Universidade Federal Fluminense, Rio de Janeiro, 25-28 maio 1998, p. 561-576.
- BORGES, P. R. *Gerenciamento de um serviço de extrato telefônico na internet por funções de relatório de alarmes de segurança*. Florianópolis, 1999. Projeto de Conclusão de Curso (Curso de Graduação em Ciência da Computação) - Departamento de Informática e de Estatística, Centro Tecnológico, Universidade Federal de Santa Catarina.
- BRINKSMA, E. *ISO 8807: LOTOS (Language of Temporal Ordering Specifications)*, 1988.
- CAIN, L., LAUNDY, N., MORRIS, C. *Motorola and Coral Systems offer cellular fraud solution for wireless carriers*. Motorola, <www.mot.com>, jun. 1997.

- CALHOUN, G. *Third generation wireless communications: post shannon architectures*. Norwood, MA, USA: Artech House Publishers, 1999. 300p. ISBN 1-58053- 043-5.
- CAMPBELL, A. T., GOMEZ, J. An overview of cellular IP. In: IEEE WCNC'99 – WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE. *Proceedings...* New Orleans, USA, 21-24 set. 1999, v.2, p. 606-610.
- CHAUDHURY, P., MOHR, W., ONOE, S. The 3GPP proposal for IMT-2000. *IEEE Communications Magazine*, New York, v.37, n.12, p.72-81, dez.1999.
- CHEIKHROUHOU, M. N. BDI-oriented agents for network management. In: GLOBECOM'99 – GLOBAL TELECOMMUNICATIONS CONFERENCE. Symposium on Enterprise Applications and Services. *Proceedings...* Rio de Janeiro, Brasil, 5-9 dez. 1999, v.3, p. 1964-1968.
- CHEN, G., RIXON, J., KONG, Q. Integration CORBA and Java for ATM connection management. In: IEEE DSOM'97 - INTERNATIONAL WORKSHOP ON DISTRIBUTED SYSTEMS OPERATIONS AND MANAGEMENT. *Proceedings...* Sydney, Australia, 1997. p.104-117.
- CHUANG, J., SOLLENBERGER, N. Wideband wireless data access based on OFDM and dynamic packet assignment. In: IEEE WCNC'99 – WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE. *Proceedings...* New Orleans, USA, 21-24 set. 1999, v.2, p. 757-761.
- CRUZ, F.A.S., NOTARE, M.S.M.A., WESTPHALL, C.B, MARTINS, A., WEBER-LEE, R., BARCIA, R.M. Using case-based reasoning in an intelligent management system. In: EIS'98 - INTERNATIONAL SYMPOSIUM ON ENGINEERING OF A INTELLIGENT SYSTEMS. *Proceedings...* Tenerife, Spain, 11-13 fev.1998. p.1093-1099.
- DAS, K. S., CHATTERJEE, M., KAKAM, N. K. QoS provisioning in wireless multimedia networks. In: IEEE WCNC'99 – WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE. *Proceedings...* New Orleans, USA, 21-24 set. 1999, v.3, p. 1493-1497.
- DAYHOFF, J. *Neural network architectures: an introduction*, Ed. Van Nostrand Reinhold, 1990.
- DEMUTH, H., BEALE, M. Neural network Tollbox: for use with MatLab. *User Guide*, Version3, 1998. p.1-7, 7, 33.
- DENNING, D. An intrusion-detection model. *IEEE Transactions on Software Engineering*, New York, v.13, n.2, p.222-232, 1987.
- DÉSIRÈ, N. *Um modelo de gerência de segurança baseado em objetos distribuídos*. Florianópolis, 1999. Trabalho Individual (Curso de Graduação em Ciência da



Computação) - Departamento de Informática e de Estatística, Centro Tecnológico, Universidade Federal de Santa Catarina.

DOGAC, A., DENGI, C., ÖSZU, M.T. Distributed object computing platforms. *Communications of the ACM*, New York, v.41, n.9, p.95-103, set.1998.

DOWD, P., MCHENRY, J.T. Network security: it's time to take it seriously. *IEEE Computer Magazine*, New York, v.31, n.9, p.24-28, set.1998.

DUDA, R. O., HART, P.E. *Pattern classification and scene analysis*. New York, NY: John Wiley & Sons, Inc., 1973.

EHRIG, H., MAHR, B. *Fundamentals of algebraic specifications*. Ed. Springer-Verlag, 1985.

GARAVEL, H. *CADP: Eucalyptus manual*. Grenoble, France: INRIA/VASY, 1997. <<http://www.inrialpes.fr/vasy/cadp/>>.

GIBSON, J.D. *The mobile communications handbook*. Piscataway, NJ, USA: IEEE Press, 1996. 726p. ISBN 0-8493- 8573-3.

GUIZANI, m., RAYES, A., ATIQUZZAMAN, M., et al. Internet Telephony. *IEEE Communications Magazine*, New York, v. 38, n. 4, p. 44-103, abr. 2000.

HAGGERTY, P., SEETHARAMAN, K. The benefits of CORBA-based network management. *Communications of the ACM*, New York, v.41, n.10, p.73-79, out.1998.

HAUW, L.H., CANELA, Z., VOYER, F. A CORBA-based TMN prototype with Web access. In: IEEE DSOM'97 - INTERNATIONAL WORKSHOP FOR DISTRIBUTED SYSTEMS OPERATIONS AND MANAGEMENT. *Proceedings...* Sydney, Australia, 21-23 out. 1997. p.81-93.

HERMIDA, A., VALE, W. *Implementação Java dos módulos agentes e adaptador do SSTCC - Sistema de Segurança para Telecomunicações Contra Clonagem e Inadimplência*. Florianópolis, 1999. Projeto de Conclusão de Curso (Curso de Graduação em Ciência da Computação) - Departamento de Informática e de Estatística, Centro Tecnológico, Universidade Federal de Santa Catarina.

HUSH, D.R., HORNE, B.G. Progress in supervised neural networks: what's new since Lippmann. *IEEE Signal Processing Magazine*, p.8-39, jan.1993.

INGBER, L. Simulated annealing: practive versus theory. *Mathematical Computing Modelling*, v.18, n.11, p.29-57, 1993.

KOHONEN, T. Self-organized formation of topologically correct feature maps. In: *BIOLOGICAL Cybernetics*, 1982, p.59-69.

- KRAIMECHE, B. Wireless ATM: current standards and issues. In: IEEE WCNC'99 – WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE. *Proceedings...* New Orleans, USA, 21-24 set. 1999, v.1, p. 56-60.
- LAWLESS, S. *FraudOffice – Software*. Ericsson Web Services, <<http://www.ericsson.com>>, mar. 1999.
- LEE, S., RHEE, M. A. Gaussian potential function network with hierarquically self-organizing learning. *Pattern Recognition Letters*, v.14, n.3, p.221- 227, 1993.
- LIM, H., CHUNG, W. Interworking of SMS between GSM based system using I-SMC. In: IEEE WCNC'99 – WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE. *Proceedings...* New Orleans, USA, 21-24 set. 1999, v.3, p. 1432-1436.
- LIPPMANN, R.P. Pattern classification using neural networks. *IEEE Communications Magazine*, New York, p. 47-64, nov.1989.
- LIU, C., CANT, T., OZOLS, M., HENDERSON, M. Formalizing certificate management systems. In: IEEE ICON'99 – INTERNATIONAL CONFERENCE ON NETWORKS. Brisbane, Austrália, 28 set. - 01 out., p.340-348.
- LU, W., BI, Q. Wireless mobile ATM technologies for third-generation wireless communications. *IEEE communications Magazine*, New York, v.37, n.11, p.36-82, nov.1999.
- LUNT, T. Automated audit trail analysis and intrusion detection: a survey. In: INTERNATIONAL COMPUTER SECURITY CONFERENCE. *Proceedings...*, 1988. p. 65-73.
- LUNT, T. et. al. Knowledge-based intrusion detection. In: ARTIFICIAL INTELLIGENCE SYSTEMS IN GOVERNEMENT CONFERENCE. *Proceedings...* mar. 1989.
- MATLAB USER'S GUIDE - For Microsoft Windows: High-Performance Numeric Computation and Visualization Software. Englewood Cliffs, NJ: The Math Works Inc., Prentice Hall, , 1992.
- MATOS, A.V. *Gerência de segurança em aplicações de bancos de dados na Web*. Florianópolis, 1999. Dissertação (Curso de Graduação em Ciência da Computação) - Departamento de Informática e de Estatística, Centro Tecnológico, Universidade Federal de Santa Catarina.
- McGRAW, G., FELTEN, E. *Java security*. New York: John Wiley & Sons, 1997. 192p. ISBN 0-471-17842-X.
- MEYER, J.W. Self-Organizing Processes. In: CONPAR'94 - VAPP VI. *Lecture Notes in Computer Science 824*. Berlin : Spring-Verlag , 1994, p.842-853.

- MEYER, B. Every little bit counts: toward more reliable software. *IEEE Computer Magazine*. New York, v.32, n.11, p.131-135, nov. 1999.
- MILIOLI, C.F., CASTELLO, W. *Sistema de segurança contra telefones celulares clonados*. Florianópolis, 1999. Projeto de Conclusão de Curso (Curso de Graduação em Ciência da Computação) - Departamento de Informática e de Estatística, Centro Tecnológico, Universidade Federal de Santa Catarina.
- MILNER, R. *A calculus of communicating systems*. Berlin: Springer-Verlag, 1980.
- NAUGHTON, P. *The Java handbook: the authoritative guide to the Java Revolution*. Berkeley, USA: McGraw-Hill, 1986. 424p. ISBN 0-07-882199-1.
- NEE, R. van, AWATER, G., MORIKURA, M, TAKANASHI, H., WEBSTER, M., HALFORD, K. New high-rate wireless LAN standards. *IEEE Communications Magazine*, New York, v.37, n.12, p.82-88. dez. 1999.
- NOTARE, M.S.M.A., CRUZ, F. A. S., SOBRAL, J.B.M., ALVES, J.B.M., RISO, B. G., WESTPHALL, C. B. Distributed Management in the Security Area for Cloned Mobile Phones, In: IEEE DSOM'98 - IEEE INTERNATIONAL WORKSHOP ON DISTRIBUTED SYSTEMS: OPERATIONS & MANAGEMENT. *Proceedings...* Newark, Delaware, USA, 26-28 out. 1998. p.14-24.
- OMG. *Security service specification in CORBA services: common objects services specification*. 1998.
- OPPLIGER, R. *Security technologies for the World Wide Web*. Norwood, MA, USA: Artech House Publisher, 1999. ISBN 1-58053-045-1.
- ORFALI, R., HARKEY, D. *Client/server programming with Java and CORBA*. New York: John Wiley & Sons, 1997.
- PAVLOU, G. From protocol-based to distributed object-based management architectures. In: IEEE DSOM'97 - INTERNATIONAL WORKSHOP FOR DISTRIBUTED SYSTEMS OPERATIONS AND MANAGEMENT. *Proceedings...* Sydney, Australia, 21-23 out. 1997. p. 25-40.
- PFLEEGER, C., COOPER, D. Security and privacy: promising advances. *IEEE Software*, New York, p.110-111, sep./out. 1997.
- PIRES, L.F. *Architectural notes: a framework for distributed systems development*. Enschede, The Netherlands: CIP - Gegevens Koninklijke Bibliotheek, 1994. ISBN 90-9007461-9.
- PRASAD, R. *Third generation mobile communications systems*. Norwood, MA, USA: Artech House Publishers, 1999. ISBN 1-58053-082-6.
- QUEIROZ, J.A.M., CUNHA, P.R.F. *Sistemas distribuídos: de especificações LOTOS a implementações*. Recife: UFPE-DI, jul. 1994.

- RAMOS, A.M., ALVES, J.B.M. Experiential knowledge. In: IEEE LANOMS'99. *Proceedings...* Rio de Janeiro, RJ, p. 236-244. ISBN: 85-900382-3-8.
- RIEZENMAN, M.J. Technology 2000 analysis & forecast: communications. *IEEE Spectrum Magazine*, New York, v.37, n.1, p.33-39, jan. 2000.
- ROZEMBLIT, M. *Security for telecommunications network management*. Piscataway, NJ: IEEE Press. 1999. ISBN 0-7803-3490-6.
- RUBIN, A. D., GEER, D. E. A survey of Web security. *IEEE Computer*, New York, v.1.31, n.9, p.34-41, sept. 1998.
- RUBIN, P., JANECEK, P. *Supersleuth fraud solution to wireless operators*. Nortel Networks Corporation, <<http://www.nortel.com>>, 1999.
- SCHNEIDER, Maria Laura. *Detecção de intrusão em redes móveis através de Redes Neurais*. Florianópolis, 1999. Dissertação (Curso de Graduação em Ciência da Computação) - Departamento de Informática e de Estatística, Centro Tecnológico, Universidade Federal de Santa Catarina.
- SHAW, J. *Strategic Management in Telecommunications*. Norwood, MA, USA: Artech House Publishers, 1999. ISBN 1-58053-018-4.
- SIMON, E. *Distributed Informations Systems: From client/server to distributed multimedia*. Maidenhead, Berkshire, England: McGraw-Hill, 1996. 414p. ISBN 0-07-709076-4.
- SLOMAN, M., TWIDLE, K. *Domains: a framework for structuring management policy*. Wokingham, UK: Addison-Wesley, 1994.
- SOUZA, F., LEITE, K. *Sistema de extrato telefônico via Web em Java com suporte CORBA*. Florianópolis, 1999. Projeto de Conclusão de Curso (Curso de Graduação em Ciência da Computação) - Departamento de Informática e de Estatística, Centro Tecnológico, Universidade Federal de Santa Catarina.
- SPÍNDOLA, Fernando. *Implementação Java do módulo gerente do SSTCC - Sistema de Segurança para Telecomunicações Contra Clonagem e Inadimplência*. Florianópolis, 1999. Projeto de Conclusão de Curso (Curso de Graduação em Ciência da Computação) - Departamento de Informática e de Estatística, Centro Tecnológico, Universidade Federal de Santa Catarina.
- STALLINGS, W. *Network and internetwork security: principles and practice*. Englewood Cliffs, NJ, USA: Prentice-Hall/IEEE Press, 1995. 462p. ISBN 0-02-415483-0.
- STEWART, K. A., EE 4984 *Telecommunication Networks Project 1: Cellular Telephone Fraud*, <[http://fiddle.ee.vt.edu/courses/ee4984/proj\\_95/stewart.html](http://fiddle.ee.vt.edu/courses/ee4984/proj_95/stewart.html)>.

STILLERMAN, M., MARCEAU, C. Intrusion Detection for Distributed Applications, *Communications of the ACM*, New York, v.42, n. 7, p.62-69, July 1999.

TODESCO, J. *Reconhecimento de padrões usando redes neurais artificiais com função de base radial: uma aplicação para classificação de cromossomos humanos*. Florianópolis, 1995. Tese (Curso de Engenharia de Produção e Sistemas) – Departamento de Engenharia de Produção, Centro Tecnológico, Universidade Federal de Santa Catarina.

VISSERS, C.A., SCOLLO, G., SINDEREN, M.V. *Architecture and specification style in formal descriptions of distributed systems*. University of Twente, The Netherlands, 1988.