

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

SISTEMA DE ARMAZENAMENTO E COMPARTILHAMENTO DE
INFORMAÇÕES COM SEGURANÇA – AMBIENTE *WINDOWS PC*

Fábio Augusto Alves Teixeira

Orientadora

Prof.^a Dr.^a. Geiza Maria Hamazaki da Silva

RIO DE JANEIRO, RJ – BRASIL

Janeiro/2014

SISTEMA DE ARMAZENAMENTO E COMPARTILHAMENTO DE INFORMAÇÕES COM SEGURANÇA – AMBIENTE *WINDOWS PC*

Fábio Augusto Alves Teixeira

Projeto de Graduação apresentado à Escola de Informática Aplicada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO) para obtenção do título de Bacharel em Sistemas de Informação.

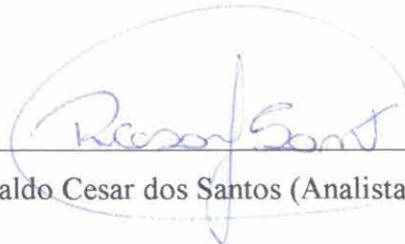
Aprovada por:



Prof.^a. Dr.^a. Geiza Maria Hamazaki da Silva (UNIRIO)



Prof. Dr. Carlos Alberto Vieira Campos (UNIRIO)



Ronaldo Cesar dos Santos (Analista Sênior)

RIO DE JANEIRO, RJ – BRASIL.

Janeiro/2014

AGRADECIMENTOS

À UNIRIO, por proporcionar meios para me manter na universidade através das bolsas permanência e monitoria; pelo conhecimento adquirido; pela experiência e prêmios ganhos com este trabalho; e principalmente pela amizade conquistada com os demais alunos, professores e corpo administrativo;

À FAPERJ, por ter possibilitado e financiado esta pesquisa;

Às pessoas que participaram e ajudaram de alguma forma no desenvolvimento deste trabalho;

Aos professores, pela paciência, ensinamentos, orientação e atenção dispensada ao longo do curso;

Aos amigos, familiares e todos aqueles que contribuíram de alguma forma seja positiva ou negativamente nessa jornada;

Devido ao crescente avanço da tecnologia e informatização dos processos, a busca por novos meios eficientes e eficazes de proteção digital tem como base pesquisas fundamentadas nas áreas da matemática e da computação. Ao visualizar a complexidade da infraestrutura tecnológica que sustenta as organizações [Stamp, 2006], uma das principais preocupações está relacionada ao armazenamento e transmissão de informações sigilosas através de dispositivos móveis e fixos. Existem vários sistemas que garantem a segurança das informações armazenadas localmente, como o *Symantec Encryption*¹ e o *Truecrypt*². No caso da transmissão da informação de forma segura, seja através de dispositivos fixos ou móveis, torna-se necessária a utilização de diversas ferramentas independentes ou integradas que na maioria das vezes são pouco intuitivas exigindo do usuário certo nível de conhecimento técnico.

Este cenário motivou o projeto de um sistema para transmissão de informações sigilosas através de um ambiente de suporte "automático" que realize o armazenamento e a transmissão de documentos criptografados entre dispositivos fixos e móveis. Esse projeto é nomeado Sistema de Armazenamento e Compartilhamento de Informações com Segurança (SACIS) cujo objetivo é garantir a confidencialidade, a integridade, disponibilidade e autenticidade das informações e dos documentos enviados ao destinatário e a interoperabilidade entre as diferentes plataformas operacionais.

O trabalho presente é um subprojeto do projeto SACIS, no qual foi desenvolvido uma infraestrutura para gerenciar as contas dos usuários do sistema, disponibilizar serviços *web* para serem consumidos e gerenciar as chaves informadas automaticamente; e uma ferramenta voltado para dispositivos fixos, *netbooks* e *notebooks* que utilizam o sistema operacional *Windows*. Esta possui as características de segurança proposta pelo SACIS, armazenando localmente arquivos criptografados e enviando mensagens e documentos com segurança para seus destinatários, além de tornar-se uma bancada para a realização de testes de novas tecnologias.

¹ Disponível em <<http://www.symantec.com/pt/br/products-solutions/families/?fid=encryption>>, Acesso em 20 Out. 2013.

² Disponível em <<http://www.truecrypt.org/>>, Acesso em 20 Out. 2013.

Palavras-Chave: Segurança da Informação, *Windows*, Interoperabilidade, Criptografia.

The increasing of the advances in technology and the automatization of processes improve researches, grounded in mathematics and computing, to the development of new efficient and effective tools to guarantee the digital protection. Due the complexity of the technology infrastructure that supports the organizations [Stamp, 2006], a major concern is related to the storage and transmission of confidential information via mobile and fixed devices. There are several systems that guarantee the security of information stored locally, such as Symantec Encryption³ and Truecrypt⁴. In the case of transmitting information securely, either through fixed or mobile devices, it is necessary to use several independent or integrated tools that are most often unintuitive and user demanding the right level of technical knowledge.

*This scenario motivated the design of a system for transmitting sensitive information using an environment that performs the storage and transmission of encrypted documents between fixed and mobile devices. This project is called **Sistema de Armazenamento e Compartilhamento de Informações com Segurança** (SACIS) whose objective is to ensure the confidentiality, integrity, availability and authenticity of the information and documents sent to the recipient and interoperability between different operational systems.*

This is a subproject of SACIS, that aimed the development of an infrastructure to manage the user accounts system, provide web services to be consumed, manage keys automatically informed; and a tool for fixed devices, netbooks and notebooks running under Windows operating system. This tool stores locally encrypted files, sends messages and documents securely to your recipients and aims to become a workbench for conducting tests of new technologies.

Keywords: Information Security, Windows, Interoperability, Encryption.

³ Available in <<http://www.symantec.com/pt/br/products-solutions/families/?fid=encryption>>, Access in 20 Oct. 2013.

⁴ Available in <<http://www.truecrypt.org/>>, Access in 20 Oct. 2013.

Índice

1 – O Prólogo de uma Nova Era	8
1.1 – Onde Motivos, Objetivos e Estado da Arte se Misturam.....	8
1.2 – Organizando a Leitura.....	16
2 – Fundamentando o SACIS.....	17
2.1 – Criptografia e Seus Conceitos Básicos	17
2.2 – Um Breve Histórico	17
2.3 – Criptografia Simétrica.....	19
2.4 – Criptografia Assimétrica	24
2.5 – Assinatura Digital e Função <i>Hash</i>	27
2.6 – <i>Web Service</i>	29
3 – O Que Há Por Trás das Câmeras.....	31
3.1 – Processo de <i>Software</i>	31
3.2 – Arquitetura MVC	32
3.3 – Ambiente de Desenvolvimento	34
3.4 – Repositório de Código e Controle de Versão.....	34
3.5 – Armazenamento dos Dados e Informações.....	35
3.5.1 – Ambiente Local	35
3.5.2 – Servidor <i>Web</i>	35
3.6 – Biblioteca Externa.....	37
4 – SACIS Para <i>Windows PC</i>	39
4.1 – Sistema de Manutenção de Usuários	40
4.2 – Sistema de Manipulação da Informação	43
4.2.1 – Sistema de Gerenciamento de Mensagens	43
4.2.2 – Sistema de Armazenamento de Arquivos	49
4.3 – Gerenciamento de Chaves.....	50
5 – Comparando Ferramentas	52
5.1 – <i>EncryptOnClick</i>	52
5.2 – <i>Kruptos 2 Professional</i>	53
5.3 – <i>Gold Lock 3G</i>	55

6 – Considerações Finais e Projetando o Futuro.....	58
7 – Anexos	61
Anexo I – Regras de Negócio	61
Anexo II – Requisitos Funcionais	62
Anexo III – Requisitos Não-Funcionais.....	63
Anexo IV – Processo de Negócio	64
Anexo V – Caso de Uso: Cadastro de Usuário	65
Anexo VI – Caso de Uso: Alteração de Usuário.....	66
Anexo VII – Caso de Uso: Exclusão de Usuário	68
Anexo VIII – Caso de Uso: Criptografia de Arquivos.....	69
Anexo IX – Caso de Uso: Descriptografia de Arquivos.....	71
Anexo X – Caso de Uso: Envio de Mensagem	73
Anexo XI – Caso de Uso: Visualizar Mensagem	78
Anexo XII – Caso de Uso: Visualizar Anexos.....	79
Anexo XIII – Caso de Uso: Manipulação de Catálogo.....	80
Anexo XIV – Caso de Uso: <i>Login</i> do Gerenciamento de Mensagem.....	81
Anexo XV – Caso de Uso: <i>Login</i> do Armazenamento de Arquivos.....	83
Anexo XVI – Caso de Uso: <i>Login</i> de Manutenção de Usuários.....	84
Anexo XVII – Caso de Uso: Troca de Senha	85
Anexo XVIII – Diagrama de Estado: Cadastro de Usuário	86
Anexo XIX – Diagrama de Estado: Alteração de Usuário	87
Anexo XX – Diagrama de Estado: Exclusão de Usuário.....	88
Anexo XXI – Diagrama de Estado: Criptografia de Arquivos	89
Anexo XXII – Diagrama de Estado: Descriptografia de Arquivos.....	90
Anexo XXIII – Diagrama de Estado: Envio de Mensagem.....	91
Anexo XXIV – Diagrama de Estado: Visualizar Mensagem.....	92
Anexo XXV – Diagrama de Estado: Visualizar Anexos	93
Anexo XXVI – Diagrama de Estado: Manipulação de Catálogo	94
Anexo XXVII – Diagrama de Estado: <i>Login</i> do Gerenciamento de Mensagem ..	95
Anexo XXVIII – Diagrama de Estado: <i>Login</i> do Armazenamento de Arquivos ..	96
Anexo XXIX – Diagrama de Estado: <i>Login</i> de Manutenção de Usuários.....	97
Anexo XXX – Diagrama de Estado: Troca de Senha.....	98

Anexo XXXI – Diagrama de Classes.....	99
Anexo XXXII – Dicionário de Dados.....	100
Anexo XXXIII – Script de Dados.....	101
Anexo XXXIV – <i>Procedure</i> para Verificar a Validade das Chaves Certificadas	102
8 – Referências.....	103

Índice de Figuras

Figura 1 – Níveis de Segurança	9
Figura 2 – Cifra de César	18
Figura 3 – Cifra Simétrica.....	19
Figura 4 – KDC.....	20
Figura 5 – Etapa <i>AddRoundKey</i>	22
Figura 6 – Etapa <i>SubBytes</i>	23
Figura 7 – Etapa <i>ShiftRows</i> 128/192 bits	23
Figura 8 – Matriz de multiplicação para blocos de 128 bits	24
Figura 9 – Etapa <i>MixColumns</i>	24
Figura 10 – Criptografia Assimétrica.....	25
Figura 11 – Assinatura Digital	28
Figura 12 – Criptografia com Assinatura Digital e Função <i>Hash</i>	29
Figura 13 – Descritografia com Assinatura Digital e Função <i>Hash</i>	29
Figura 14 – Arquitetura MVC do Projeto SACIS.....	32
Figura 15 – Camada de Modelo do Projeto SACIS	33
Figura 16 – Camada de Controle do Projeto SACIS.....	33
Figura 17 – Camada de Visão do Projeto SACIS	34
Figura 18 – Modelo Relacional.....	37
Figura 19 – Organograma Geral do Sistema.....	39
Figura 20 – Tela Padrão de <i>Login</i>	39
Figura 21 – Telas de Cadastro de Usuários.....	40
Figura 22 – Mensagem: “Certificado Inválido!”	40
Figura 23 – Mensagem: “Certificado Inválido ou com senha!”	41
Figura 24 – Mensagem: “Selecione um Tipo de Permissão!”	41
Figura 25 – Telas de Alteração de Dados e Exclusão dos Usuários	42
Figura 26 – Tela Nova Senha.....	43
Figura 27 – Tela principal do Sistema de Mensagens.....	44
Figura 28 – Tela de Nova Mensagem	45
Figura 29 – Tela para Anexar Arquivos.....	45
Figura 30 – Mensagem caso destinatário não seja informado ou seja inválido.....	46

Figura 31 – Formato XML para mensagens	46
Figura 32 – Mensagem enviada com sucesso	47
Figura 33 – Lista com destinatários não existentes ou certificados expirados	47
Figura 34 - Aviso caso conteúdo da mensagem tenha sido alterado.....	48
Figura 35 – Tela de Visualização de Mensagem	48
Figura 36 – Telas de Contatos Geral e Pessoal.....	49
Figura 37 – Tela principal do Sistema de Gerenciamento de Arquivos	50
Figura 38 – Aviso de Expiração de Certificado.....	51
Figura 39 – Tela Inicial <i>EncryptOnClick</i>	52
Figura 40 – Solicitação de Senha.....	53
Figura 41 – Tela Principal <i>Kruptos</i>	54
Figura 42 – Solicitação de Senha.....	54
Figura 43 – Tela login.....	56
Figura 44 – Tela principal.....	56
Figura 45 – Abas Para Envio de Mensagens e Arquivos.....	57
Figura 46 – Processo de Negócio.....	64
Figura 47 – Diagrama de Estado: Cadastro de Usuário.....	86
Figura 48 – Diagrama de Estado: Alteração de Usuário.....	87
Figura 49 – Diagrama de Estado: Exclusão de Usuário.....	88
Figura 50 – Diagrama de Estado: Criptografia de Arquivos.....	89
Figura 51 – Diagrama de Estado: Descriptografia de Arquivos	90
Figura 52 – Diagrama de Estado: Envio de Mensagem.....	91
Figura 53 – Diagrama de Estado: Visualizar Mensagem.....	92
Figura 54 – Diagrama de Estado: Visualizar Anexos	93
Figura 55 – Diagrama de Estado: Manipulação de Catálogo.....	94
Figura 56 – Diagrama de Estado: <i>Login</i> do Gerenciamento de Mensagem.....	95
Figura 57 – Diagrama de Estado: <i>Login</i> do Armazenamento de Arquivos	96
Figura 58 – Diagrama de Estado: <i>Login</i> de Manutenção de Usuários.....	97
Figura 59 – Diagrama de Estado: Troca de Senha.....	98
Figura 60 – Diagrama de Classes.....	99

Índice de Tabelas

Tabela 1 – Classificação de Risco.....	9
Tabela 2 – Comparação das Ameaças de 2010 e 2013	10
Tabela 3 – Ferramentas Criptográficas	14
Tabela 3 – Criptografia RSA para a palavra “love”	27
Tabela 4 – Descriptografia RSA para a palavra “love”	27

1 – O Prólogo de uma Nova Era

1.1 – Onde Motivos, Objetivos e Estado da Arte se Misturam

Uma das principais preocupações na gestão corporativa está relacionada à segurança da informação. Ao visualizar a infraestrutura tecnológica que sustenta as organizações corporativas, um de seus principais componentes é o parque computacional, onde o armazenamento e transmissão de informações são realizados através de dispositivos móveis e fixos dentre diversas plataformas e protocolos de rede.

Uma política de segurança consistente deve abranger proteção em três níveis (Figura 1), segundo [Lopez, M.D., 2009]:

1. **Segurança do dispositivo e dos dados nele contidos:** As melhores práticas indicam a necessidade de criptografar todos os dados armazenados no aparelho e em cartões de memória removíveis. Além disso, também é necessário proteger o equipamento de vírus, *worms* e *trojans*⁵. Isso pode ser feito com o uso de *firewall*⁶ e antivírus⁷ para equipamentos móveis.
2. **Segurança na transmissão de dados:** As empresas devem adotar soluções que verifiquem a identidade do remetente e do destinatário e que protejam os dados contra modificação por terceiros à medida que as informações passam pela rede sem fio.
3. **Segurança no acesso à rede corporativa:** Essa é a função mais complexa de uma política de segurança consistente porque deve, ao mesmo tempo, proibir o acesso não autorizado à rede corporativa e permitir que usuários autorizados transfiram informações para dentro e para fora da rede em questão.

Dessa forma, Lopez segue a política de segurança em vigência atual estando de acordo com a norma brasileira ISO/IEC 17799:2005⁸, atualizada em 2007 para a ISO/IEC 27002:2005, e com a norma internacional ISO/IEC 27002:2013⁹ cujo objetivo dos controles criptográficos é proteger a confidencialidade, autenticidade, inte-

⁵ Vírus, *worms* e *trojans* – são programas maliciosos passíveis de causar prejuízos aos sistemas com características distintas.

⁶ *Firewall* – é um programa que controla o tráfego de dados no computador.

⁷ Antivírus – é um programa capaz de detectar e destruir vírus, *worms* e *trojans*.

⁸ Disponível em <http://portal.cjf.jus.br/sigjus/arquivos-diversos/NBR-ISO-IEC-17799-2005.PDF/at_download/file> Acesso em 16 Nov. 2013.

⁹ Disponível em <<http://www.iso27001security.com/html/27002.html>> Acesso em 16 Nov. 2013.

gridade e disponibilidade das informações. Vale a pena ressaltar a propriedade irrefragável ou não-repúdio de uma informação que garante que uma pessoa não consiga negar um ato ou autenticidade de um documento de sua autoria.

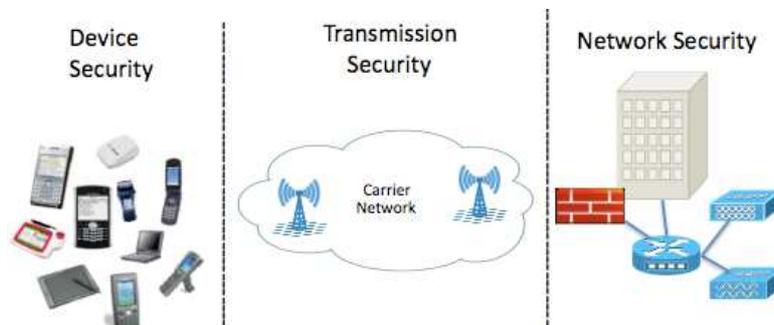


Figura 1 – Níveis de Segurança¹⁰

O desenvolvimento de novas tecnologias contendo novas vulnerabilidades, os avanços realizados pelos atacantes e a construção de sistemas mais complexos e seguros são os principais fatores para a mudança constante no cenário das ameaças para a segurança de uma aplicação e/ou dispositivo seja ele móvel ou fixo. A *Open Web Application Security Project (OWASP)*¹¹, comunidade sem fins comerciais dedicada a aperfeiçoar a segurança nas aplicações, a cada três anos publica as dez maiores ameaças à segurança da informação de acordo com sua prevalência em relação aos dados combinados com uma classificação de risco pré-determinada de exploração, detecção e impacto, representada pela Tabela 1. O conjunto de dados utilizado é fornecido por empresas especializadas em segurança da informação, incluindo consultorias e fornecedores de ferramentas de segurança.

Agentes de Ameaça	Vetores de Ataque	Prevalência da Vulnerabilidade	Deteção Vulnerabilidade	Impactos Técnicos	Impactos no Negócio
Específico da Aplicação	Fácil	Generalizada	Fácil	Severo	Específico do Negócio/ Aplicação
	Média	Comum	Média	Moderado	
	Difícil	Rara	Difícil	Pequeno	

Tabela 1 – Classificação de Risco¹²

¹⁰ Níveis de Segurança [Lopez, M.D., 2009]

¹¹ Disponível em <https://www.owasp.org/index.php/Main_Page> Acesso em 16 Nov. 2013.

¹² Classificação de Risco: Disponível em <http://owasptop10.googlecode.com/files/OWASP_Top_10_-_2013_Brazilian_Portuguese.pdf>. Acesso em 19 Nov. 2013

De acordo com a publicação da OWASP no ano de 2013, as três maiores principais ameaças para a segurança da informação são:

1. **Injeção de Código:** São dados manipulados e não confiáveis enviados pelo atacante, como parte de um comando ou consulta, com o intuito de iludir o interpretador a fim de conseguir acesso aos dados ou a execução do comando.
2. **Quebra de Autenticação e Gerenciamento de Sessão:** É a incorreta implementação de funções relacionadas a autenticação e ao gerenciamento de chaves numa aplicação. O atacante ao explorar essas falhas, compromete senhas e chaves, além de poder assumir o controle das contas dos usuários.
3. **Cross-Site Scripting (XSS):** É a recepção de dados não confiáveis pela aplicação e o seu envio ao navegador sem validá-los ou utilizar filtros adequados. Dessa forma, o atacante tem condições de executar comandos no navegador para atingir seus fins que vão de um simples redirecionamento para sites maliciosos até a captação de sessões do usuário.

Em relação ao ano de 2010 não houve mudança do tipo de ameaça apenas havendo uma inversão na ordem entre a segunda e terceira ameaça. A Tabela 2 apresenta uma comparação das ameaças publicadas em 2010 e 2013.

OWASP Top 10 – 2010 (Anterior)	OWASP Top 10 – 2013 (Novo)
A1 – Injeção de código	A1 – Injeção de código
A3 – Quebra de autenticação e Gerenciamento de Sessão	A2 – Quebra de autenticação e Gerenciamento de Sessão
A2 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Referência Insegura e Direta a Objetos	A4 – Referência Insegura e Direta a Objetos
A6 – Configuração Incorreta de Segurança	A5 – Configuração Incorreta de Segurança
A7 – Armazenamento Criptográfico Inseguro – Agrupado com A9 →	A6 – Exposição de Dados Sensíveis
A8 – Falha na Restrição de Acesso a URL – Ampliado para →	A7 – Falta de Função para Controle do Nível de Acesso
A5 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
<Removido do A6: Configuração Incorreta de Segurança>	A9 – Utilização de Componentes Vulneráveis Conhecidos
A10 – Redirecionamentos e Encaminhamentos Inválidos	A10 – Redirecionamentos e Encaminhamentos Inválidos
A9 – Proteção Insuficiente no Nível de Transporte	Agrupado com 2010-A7 criando o 2013-A6

Tabela 2 – Comparação das Ameaças de 2010 e 2013¹³

¹³ Comparação das Ameaças de 2010 e 2013: Disponível em http://owasptop10.googlecode.com/files/OWASP_Top_10_-_2013_Brazilian_Portuguese.pdf. Acesso em 19 Nov. 2013

Essas vulnerabilidades ao comprometer a segurança dos dados facilitam todo tipo de espionagem nas organizações estatais e privadas. Recentemente a segurança nacional de diversos países, dentre eles o Brasil, foi comprometida devido ao programa de espionagem norte-americana denominada *PRISM* [Karasinski, L., 2013]. Essa ferramenta realiza uma vigilância constante e monitora em tempo real as conversas telefônicas e as informações trafegadas na internet como *e-mails*, transações bancárias, conversas em chats, vídeos e etc. Segundo [Hamman, R., 2013]:

[...] os Estados Unidos estariam espionando pessoas em todo o mundo, graças a sistemas de rastreamento infiltrados em serviços como *Facebook*, *Google* e *Skype*, além de redes móveis e outras conexões de dados. [...]

Ainda neste episódio, algumas das medidas em discussão a serem tomadas pelos países europeus e o Brasil seriam:

[...] o estabelecimento de uma multa de 2% na receita global das empresas que venham a usar a espionagem para obter informações de terceiros.

[...] a obrigatoriedade de os dados dos europeus ficarem hospedados no território europeu. [...] e não onde o provedor americano quer hospedar [...]

[Lobo, A. P, 2013]

No caso do Brasil, ainda está sendo definido se as medidas serão registradas no Marco Civil da Internet¹⁴ ou na Lei de Proteção de Dados Pessoais¹⁵, que está em formulação pelo ministério da Justiça.

Algumas medidas nesse sentido já estavam sendo adotadas pelo governo brasileiro. Em 14 de novembro de 2012, a presidente brasileira sancionou o decreto nº 7.845/2012¹⁶ que:

Regulamenta procedimentos para credenciamento de segurança e tratamento de informação classificada em qualquer grau de sigilo. [...]

O Diário Oficial da União¹⁷ publicado em 19 de fevereiro de 2013 oficializa o desenvolvimento de um Algoritmo de Estado¹⁸. Segue a seguir os trechos referentes ao seu objetivo e quem será responsável pelo seu desenvolvimento, respectivamente:

¹⁴ Marco Civil da Internet – é um projeto de Lei que visa estabelecer direitos e deveres na utilização da Internet no Brasil.

¹⁵ Lei de Proteção de Dados Pessoais – é um anteprojeto proposto que visa assegurar ao cidadão o controle e a titularidade sobre suas próprias informações pessoais, como forma de garantia do direito constitucional à privacidade.

¹⁶ Disponível em < http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2012/Decreto/D7845.htm> Acesso em 16 Nov. 2013.

¹⁷ Disponível em < http://convergenciadigital.uol.com.br/inf/bo_130219_42-43.pdf> Acesso em 16 Nov. 2013.

Normatizar o uso de recurso criptográfico para a segurança de informações produzidas nos órgãos e entidades da Administração Pública Federal - APF [...]

O recurso criptográfico baseado em algoritmo de Estado deverá ser de desenvolvimento próprio ou por órgãos e entidades da administração pública federal, direta ou indireta, mediante acordo ou termo de cooperação, vedada a participação e contratação de empresas e profissionais externos à APF [...]

Logo após os incidentes relacionados à espionagem americana no Brasil, o chefe do Departamento de Segurança da Informação e Comunicações do Gabinete de Segurança Institucional da Presidência da República lançou o novo algoritmo de proteção de dados criado em parceria com a Agência Brasileira de Inteligência segundo [Rodrigues, E., 2013]. Recentemente, em 4 de novembro de 2013, o decreto Nº 8.135¹⁹ foi sancionado pela presidente e publicado no dia seguinte no Diário Oficial da União. Nele fica determinado em seu artigo 1º que:

As comunicações de dados da administração pública federal direta, autárquica e fundacional deverão ser realizadas por redes de telecomunicações e serviços de tecnologia da informação fornecidos por órgãos ou entidades da administração pública federal, incluindo empresas públicas e sociedades de economia mista da União e suas subsidiárias. [...]

Para cumprir este fim e dessa forma ampliar a privacidade e a inviolabilidade de mensagens e comunicações oficiais, a presidente determinou a todos os órgãos do governo federal adotar o programa Expresso V3²⁰ desenvolvido pelo Serviço Federal de Processamento de Dados (Serpro). O programa dispõe de 6 funcionalidades: e-mail, catálogo de contatos, tarefas, calendário, mensagens instantâneas e *web conference*. Essas medidas adotadas pelo governo brasileiro também visam unificar os procedimentos para o uso de sistemas de criptografia uma vez que os diversos órgãos da administração pública usam diferentes ferramentas criptográficas [Grossmann, L.O., 2013].

No mercado existem vários aplicativos desenvolvidos para dispositivos fixos e móveis que podem ser utilizados para a proteção da informação através de técnicas criptográficas, sejam elas simétricas ou assimétricas, bem como para a verificação de

¹⁸ Função matemática utilizada na criptografia e na descriptografia, desenvolvido pelo Estado, para uso exclusivo em interesse do serviço de órgãos ou entidades do Poder Executivo Federal.

¹⁹ Disponível em <<http://www.jusbrasil.com.br/diarios/61251484/dou-secao-1-05-11-2013-pg-2>> Acesso em 18 Nov. 2013.

²⁰ Disponível em <<https://www.serpro.gov.br/noticias/uso-de-e-mail-seguro-torna-se-obrigatorio-em-todo-o-governo-federal>>. Acesso em 18 Nov. 2013.

integridade [Douglas, 2005] [Schneier, 1996], como por exemplo: *Symantec Encryption* e *Gold Lock 3G*²¹. Para todos os tipos de dispositivos dificilmente é permitido a inserção de algoritmos proprietários e a perfeita sincronização entre os dispositivos móveis e os computadores nesse tipo de aplicação.

As ferramentas criptográficas existentes, em sua maioria, contemplam a criptografia de arquivos em discos rígidos e/ou removíveis de forma direta ou através da criação de pastas e diretórios virtuais para o ambiente operacional *Windows*, e poucas possuem também as funcionalidades de envio de mensagens ou *e-mails* com segurança e a comunicação segura através de voz. Além disso, embora haja comunicação entre os aplicativos, as empresas disponibilizam apenas uma funcionalidade por aplicativo independente dela ser gratuita ou não. As exceções ocorrem quando as empresas disponibilizam pacotes contendo um *mix* de funcionalidades, como a *Gold Lock 3G* e a Família *Symantec Encryption*, ou como a *Gpg4win*²² que disponibilizaram as funcionalidades de envio de mensagens e o armazenamento seguro das informações na mesma ferramenta, conforme ilustra a Tabela 3.

Diante deste cenário, surgiu a ideia de criar uma aplicação que possua uma interface simples, não necessite de conhecimento técnico sobre criptografia para utilizar suas funcionalidade e aceite algoritmos criptográficos proprietários. Inicialmente ela estaria voltada para *smartphones* operando nos ambientes *Windows Mobile* e *Android*. Dessa forma, desenvolveu-se o projeto SACIS que visa unir a segurança das informações armazenadas em arquivos, mensagens enviadas e a comunicação por voz em uma única ferramenta utilizando sistemas criptográficos públicos ou proprietários para diferentes sistemas operacionais e dispositivos que se comunicarão através de um servidor *web*.

Para o projeto SACIS tornar-se uma realidade, surgiu a necessidade de dividi-lo em subprojetos. Um destes é o: SACIS - *Windows PC (Personal Computer*²³), cujo objetivo é armazenar documentos criptografados e transmitir informações e documentos sigilosos com segurança em *notebooks*, *netbooks* e computadores fixos que possuam o *Windows* como sistema operacional.

²¹ Disponível em <<https://www.gold-lock.com/en/home/>>. Acesso em 28 Out. 2013.

²² Disponível em <<http://www.gpg4win.org/features.html>>. Acesso em 16 Nov. 2013

²³ *Personal Computer* – Computadores Pessoais.

Ferramenta Criptográfica	Pago	Arquivos	E-mail / Mensagem	Voz	Pasta Virtual	Gerencia Chaves	Ambiente
Symantec Email Encryption	sim		x			sim	pc, mac, linux
Symantec File Share Encryption	sim	x				sim	pc, mac, linux
BoxCryptor	sim	x				Não Informado	Windows, Mac, Linux, iOS e Android
TrueCrypt	não				x	sim	Windows, Mac OS X e Linux
Espionage	sim	x				Não Informado	Mac OS X
Gpg4win	não	x	x			Não Informado	Windows
SafeHouse Professional/Personal	sim				x	Não Informado	Windows
SafeHouse Explorer Encryption	não				x	Não Informado	Windows
EncryptOnClick	não	x				não	Windows
Orphius	sim	x				Não Informado	Windows
Kruptos 2 Professional	sim	x				Não Informado	Windows
Cypherix LE Cryptainer PE	sim	x				Não Informado	Windows
Cypherix LE Secure IT	sim	x				Não Informado	Windows
Droid Crypt	não	x				Não Informado	Android
AnDisk Encryption	não	x				Não Informado	Android
Whispersystems RedPhone	não			x		Não Informado	Nexus S, Nexus One e Android
Whispersystems TextSecure	não		x			Não Informado	Nexus S, Nexus One e Android
Gold Lock 3G	sim		x	x		Não Informado	Windows, OS e Android
SecurStar PocketCrypt	sim				x	Não Informado	Windows Mobile, WinCE
Cellcrypt Mobile	sim			x		Não Informado	iOS, Android e BlackBerry
SACIS	não	x	x	x		sim	Windows, Windows mobile e Android

Tabela 3 – Ferramentas Criptográficas

Este trabalho foi definido como plataforma para o desenvolvimento de novas tecnologias, tais como a comunicação de voz, inserção de algoritmos proprietários e criptografia através da biometria e não tem pretensão de apresentar uma ferramenta criptográfica com novas tecnologias.

Os sistemas desenvolvidos para o SACIS - *Windows PC* são: o Sistema de Armazenamento de Arquivos e Sistema de Gerenciamento de Mensagens. Estes deverão conter uma interface gráfica simples e intuitiva que não exija conhecimento técnico para sua utilização. Eles disponibilizam as seguintes funcionalidades para os usuários:

A) Sistema de Armazenamento de Arquivos

Este sistema será responsável por criptografar e descriptografar o(s) arquivo(s) escolhido(s) através de uma frase-senha definida pelo usuário. Seu objetivo é manter a confidencialidade do(s) documento(s) evitando que seu conteúdo seja acessado por pessoas não autorizadas nos dispositivos locais.

B) Sistema de Gerenciamento de Mensagens

Este sistema será responsável por enviar, abrir e receber mensagens e documentos criptografados, através de chaves certificadas, ou em texto claro. Seu objetivo é garantir a confidencialidade, a integridade, a autenticidade, irretratibilidade e a disponibilidade da informação enviada.

Porém, para o SACIS - *Windows PC* poder realizar o envio das mensagens e dessa forma existir a comunicação entre pares, foi necessário desenvolver para o Projeto SACIS, neste mesmo trabalho, um servidor *web*, um gerenciador de chaves e um Sistema de Manutenção de Usuários. O servidor *web* visa realizar a comunicação entre os sistemas cliente e o servidor. O gerenciador de chaves tem por objetivo garantir que as chaves certificadas estejam sempre aptas para o uso do sistema e evitar sua utilização caso ela não seja válida ou não exista. O Sistema de Manutenção de Usuários irá cadastrar, alterar dados ou excluir um usuário do sistema.

1.2 – Organizando a Leitura

Esta introdução permitiu apresentar a motivação, o estado da arte e os objetivos ao qual esse trabalho se propõe. A seguir, no Capítulo 2, serão apresentados os fundamentos criptográficos utilizados na realização deste projeto bem como um breve histórico sobre estes. No Capítulo 3, as funcionalidades de gerência de mensagens, gerência de usuários, armazenamento de arquivos local, gerência de chaves e as telas propostas do sistema desenvolvido serão pormenorizados. No Capítulo 4 são descritos as especificações do sistema bem como seu desenvolvimento. No Capítulo 5 é realizada uma comparação entre o sistema SACIS – *Windows PC* e outras ferramentas. Por fim, no Capítulo 6 tem-se as considerações finais e os trabalhos futuros.

2 – Fundamentando o SACIS

Neste capítulo são apresentados os conceitos básicos da criptografia, um breve histórico e as principais técnicas utilizadas atualmente para armazenamento e envio de informações através do meio digital. Também serão abordados os fundamentos utilizados para o armazenamento e disponibilização da informação através de um servidor *web*.

2.1 – Criptografia e Seus Conceitos Básicos

A palavra criptografia tem origem grega, onde *kriptos* significa escondido, oculto e *grifo* significa grafia, definindo a arte ou ciência de escrever em cifras ou em códigos. Utilizando um conjunto de técnicas ela torna uma mensagem clara em incompreensível, denominada texto cifrado ou criptografado, através de um processo chamado cifragem. Isso permite que apenas o destinatário desejado consiga decodificar a mensagem cifrada e ler a mensagem clara. Esse processo inverso é denominado decifragem. A criptografia é um subconjunto da criptologia bem como a criptoanálise. Enquanto a primeira busca esconder as informações, a segunda tenta quebrar as técnicas usadas para obter as informações escondidas a partir dos dados codificados sem acessar os segredos necessários para a decodificação.

2.2 – Um Breve Histórico

As técnicas tradicionais de criptografia são divididas em clássicas e modernas. Técnicas clássicas foram utilizadas até surgirem os computadores enquanto as modernas baseiam-se nos computadores dividindo-se em algoritmos simétricos e assimétricos [Abdalla, A. et al].

Antes do advento da computação, a criptografia era utilizada para transportar mensagens com segurança evitando seu entendimento pelo inimigo caso fosse interceptada. Pelo seu baixo grau de complexidade, a criptografia clássica era feita pelas pessoas utilizando somente lápis e papel ou algum equipamento mecânico e técnicas de transposição ou substituição de caracteres. A cifra de César (Figura 2), que substi-

tui uma letra do alfabeto por outro dado um valor de deslocamento [Pinto P., 2013] e a transposição colunar [Azevedo A., 2010], que dispõe o texto claro em uma matriz linha a linha sendo a chave a ordem das colunas, são exemplos de técnicas clássicas de substituição e transposição respectivamente.

Durante a idade média a técnica de decifragem mais utilizada foi a de análise de frequências que consistia em analisar a frequência das letras nas mensagens cifradas comparando-as às frequências médias das letras em textos daquele determinado idioma. Devido as cifras de substituição serem uma técnica simples, essa técnica de criptoanálise permitia a fácil quebra da cifragem.

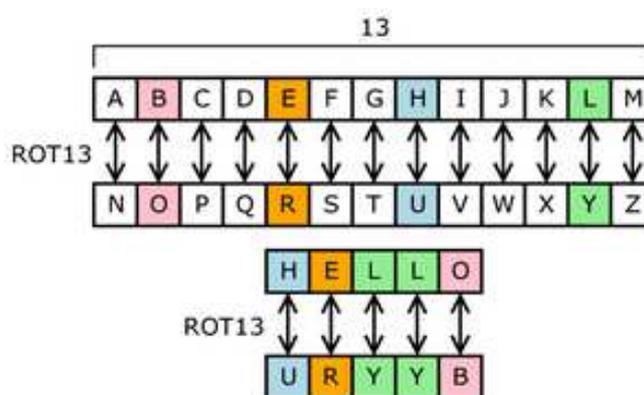


Figura 2 – Cifra de César²⁴

No início do século XX começou-se a utilizar aparelhos mecânicos para a aplicação e remoção de cifras combinando mensagens claras, chaves secretas e operações matemáticas. A 2ª Guerra Mundial proliferou a criptografia e aparelhos para quebra de cifras. Um exemplo foi a máquina Enigma utilizada pelos alemães durante esta guerra para transmitir mensagens cifradas e teve sua cifra quebrada pelos aliados, graças a Allan Turing e sua máquina Colossus, descobrindo dessa forma os segredos militares dos alemães. De 1950 à 1970, a criptografia virou segredo de estado e muito pouco material foi divulgado. A partir de 1970 ela voltou a ser publicada, mas desta vez com teorias matemáticas, de informação e de comunicação fundamentando-a e baseando-se também na computação. Com isso diversas técnicas criptográficas foram desenvolvidas predominando as técnicas de criptografia simétrica e assimétrica até os dias de hoje.

²⁴ Cifra de César: Disponível em <http://pt.wikipedia.org/wiki/Cifra_de_substitui%C3%A7%C3%A3o>. Acesso em 20 Out. 2013

2.3 – Criptografia Simétrica

A criptografia simétrica é a forma mais simples para criptografar textos claros. No processo de cifragem, a informação clara é enviada para um algoritmo que também recebe a chave e como saída ele fornece a informação criptografada que é transmitida ao receptor ou armazenada. No processo de decifragem, a informação cifrada é inserida no algoritmo, juntamente com a mesma chave utilizada para cifrá-la, que realiza o processo inverso recuperando a informação original (Figura 3).

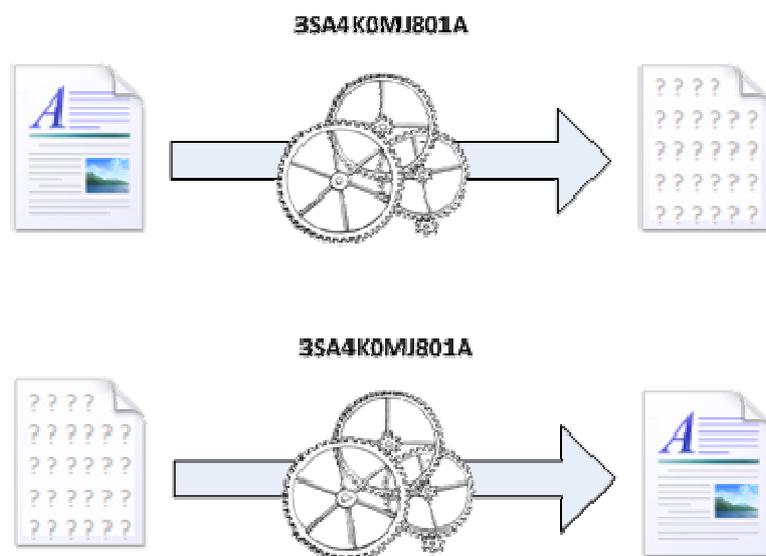


Figura 3 – Cifra Simétrica²⁵

Sua principal vantagem é a alta velocidade de codificação e decodificação, porém ela tem uma grande desvantagem no uso de uma mesma chave para codificar e decodificar a informação: a transmissão dessa chave para que um terceiro possa recuperar o texto original. Esse problema não ocorre quando os textos cifrados estão armazenados localmente e somente uma pessoa conhece a chave. Essa transmissão tem que ser segura o suficiente para não comprometer a segurança da informação cifrada.

Uma forma de contornar esse problema foi a criação do centro de distribuição de chaves (*Key Distribution Center – KDC*) [Kak, A., 2013]. Para utilizar o KDC é necessário o usuário estar registrado nele. No ato do registro uma chave secreta é

²⁵ Cifra Simétrica: disponível em <<http://rarefecundo.wordpress.com/category/sistemas-de-informacao/seguranca-de-sistemas/>>. Acesso em 20 Out. 2013

compartilhada entre ambos para o envio da chave de sessão que será gerada pelo KDC quando esta for solicitada pelo usuário. Dessa forma, uma comunicação segura entre dois usuários registrados deverá ocorrer de acordo com a seguinte sequência (Figura 4):

1. O usuário A solicita ao KDC uma comunicação segura com o usuário B;
2. O KDC gera uma chave de sessão;
3. O KDC envia a chave de sessão gerada para os usuários A e B utilizando a chave secreta KDC de cada um;
4. A e B utilizam a mesma chave de sessão para se comunicar.

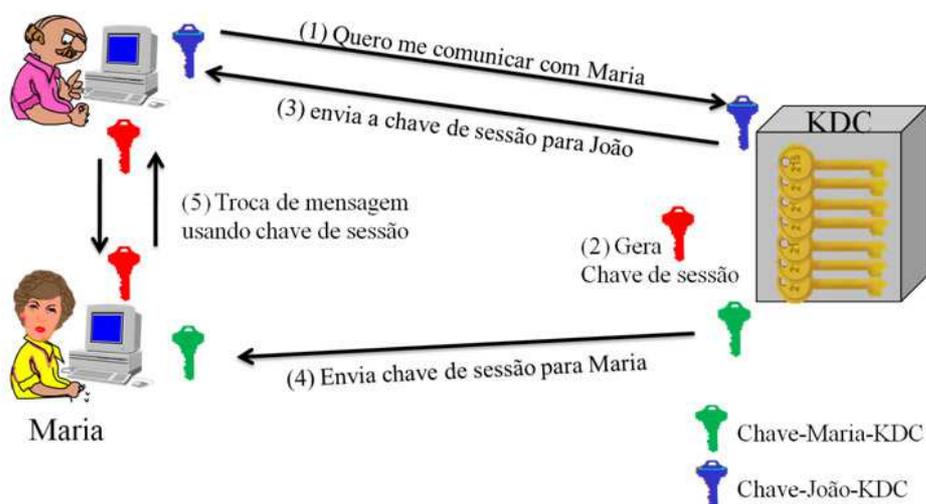


Figura 4 – KDC²⁶

O surgimento da computação permitiu o trabalho com números ao invés de fazê-lo somente com letras e caracteres gráficos, possibilitando um importante passo para o desenvolvimento da criptologia. Assim, na década de 70 do século XX, iniciou-se o desenvolvimento das cifras em blocos que consiste na divisão de um texto claro em blocos de tamanho fixo, em geral de 64 ou 128 bits, que são cifrados através de um dos diferentes modos de operação existentes podendo conferir não só a confidencialidade como também a autenticação.

²⁶ KDC: disponível em http://www.metropledigital.ufrn.br/aulas_avancado/web/disciplinas/seg_redes/aula_03.html. Acesso em 20 Out. 2013

A primeira cifra de blocos criada foi o *Data Encryption Standard* (DES) [Smid, M.E. & Branstad, D.K., 1988]. Ela foi desenvolvida e submetida pela *International Business Machines* (IBM) em 1974 para um concurso aberto pela *National Bureau of Standards* (NBS), atualmente conhecido como *National Institute of Standards and Technology* (NIST), órgão de padrões do governo norte-americano, que havia identificado a necessidade de um padrão governamental para criptografia de informações após uma consulta ao *National Security Agency* (NSA). Embora inicialmente controverso, com um pequeno tamanho de chave e suspeitas de um *backdoor*²⁷ da NSA para que somente ela pudesse ler facilmente as mensagens criptografadas, o Comitê de Inteligência dos Estados Unidos concluiu e provou que o algoritmo poderia funcionar com o pequeno tamanho de chave e que o DES estava livre de fragilidades estatísticas e matemáticas. Em 1990 as suspeitas seriam finalmente acalmadas em definitivo devido a descoberta independente e publicação aberta de Eli Biham e Adi Shamir sobre criptoanálise diferencial [Rinaldi, D.G., 2012], um método genérico de quebra de criptografia.

Vários ataques teóricos foram publicados durante a década de 1990 embora improváveis de serem realizados na prática. Em janeiro de 1999, os esforços empreendidos em conjunto pela *distributed.net* e a *Electronic Frontier Foundation* finalmente deram resultado com a quebra de uma chave DES em 22 horas e 15 minutos. A descoberta das fragilidades no DES evidenciou a necessidade do desenvolvimento de um novo algoritmo criptográfico, o *Advanced Encryption Standard* (AES) [Rouse, M., 2011], e um novo concurso foi realizado. Três anos e meio após o início do concurso, o NIST chega à escolha do vencedor: *Rijndael* [Rouse, M., 2007]. O nome é a fusão dos nomes de Vincent Rijmen e Joan Daemen, os dois belgas criadores do algoritmo. O *Rijndael* concorreu com outros 4 finalistas: *MARS* [Burwick, C. et al, 1999], *RC6* [Robshaw, M.J.B. (2001)], *Serpent*²⁸ e *Twofish*²⁹, sendo os dois últimos largamente utilizados atualmente. Segundo o NIST, o *Rijndael* combina as características de segurança, desempenho, facilidade de implementação e flexibilidade. A-

²⁷ *Backdoor* – são programas instalados no computador que permitem controlá-lo à distância.

²⁸ Disponível em <<http://www.cl.cam.ac.uk/~rja14/serpent.html>>. Acesso em 28 Out. 2013

²⁹ Disponível em <<https://www.schneier.com/twofish.html>>. Acesso em 20 Out. 2013

apresenta também alta resistência a ataques como *power attack* e *timing attack*³⁰ e exige pouca memória, o que o torna adequado para operar em ambientes restritos como *smartcards*³¹ e telefones celulares.

A partir de 2002 o AES ou *Rijndael* foi adotado como novo padrão de criptografia norte-americana. Apesar do AES e *Rijndael* serem considerados sinônimos, existe uma pequena diferença entre eles. O AES é um padrão base que tem um tamanho de bloco de 128 bits e de chave que pode ser de 128, 192 e 256 bits. O *Rijndael* foi desenvolvido de acordo com o padrão base, tendo tamanho de blocos e de chaves qualquer múltiplo de 32 bits devendo estar entre 128 e 256 bits. O AES opera a partir de uma matriz de bytes com 4x4 posições chamado de estado. O *Rijndael* por ter tamanho de bloco maior possui colunas adicionais de estado. Para criptografar um texto claro não há diferenças entre o AES e o *Rijndael*. Ambos irão realizar as mesmas tarefas de cifragem em cada turno que consistem em 4 estágios, exceto o último. No primeiro estágio, denominado *AddRoundKey*, em cada *round* uma subchave é derivada da chave no mesmo tamanho do estado e combinada com cada byte do estado numa operação XOR bit a bit (Figura 5).

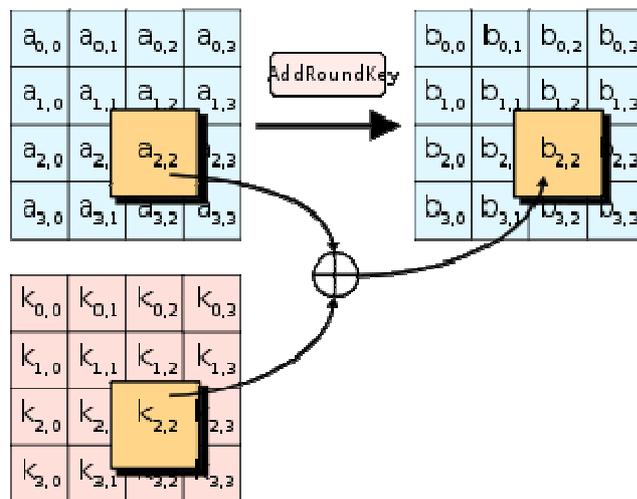


Figura 5 – Etapa *AddRoundKey*³²

³⁰ *Power attack* e *Timing attack* – é a quebra de uma cifra através da dedução da chave secreta e do conhecimento do funcionamento interno do algoritmo criptográfico, respectivamente.

³¹ *Smartcards* – é um cartão que possui capacidade de microprocessamento embutido.

³² Etapa *AddRoundKey*: disponível em

<<http://aescryptography.blogspot.com.br/2012/05/addroundkey-step.html>>. Acesso em 20 Out. 2013

No segundo estágio, o *SubBytes*, ocorre a substituição não linear de cada byte na matriz através do *Rijndael S-Box* que utiliza uma técnica de substituição de 8 bit (Figura 6). O S-Box usado é derivado de uma função inversora multiplicativa sobre $GF(2^8)$ para evitar ataques baseados em propriedades algébricas simples.

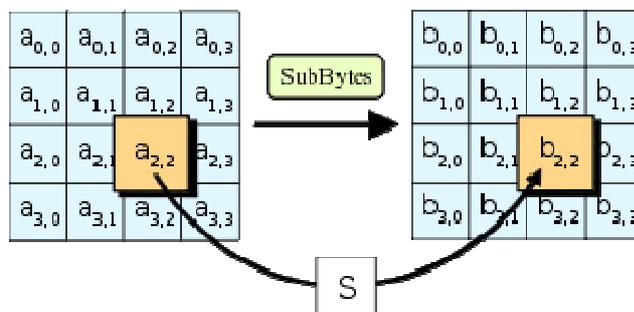


Figura 6 – Etapa *SubBytes*³³

Logo em seguida, no terceiro estágio denominado *ShiftRows*, ele desloca os bytes referentes àquela linha de estados num determinado número de posições. Para os blocos de 128 e 192 bits o deslocamento é o mesmo onde a primeira linha não é modificada e para as demais linhas, os bytes são deslocados $n-1$ linhas para a esquerda, por exemplo, a linha 3 terá um deslocamento de 2 a esquerda (Figura 7). Para os blocos de 256 bits a primeira linha se mantém a mesma enquanto a segunda linha terá um deslocamento de 1 a esquerda. Para as demais linhas, o deslocamento será de n linhas para a esquerda.

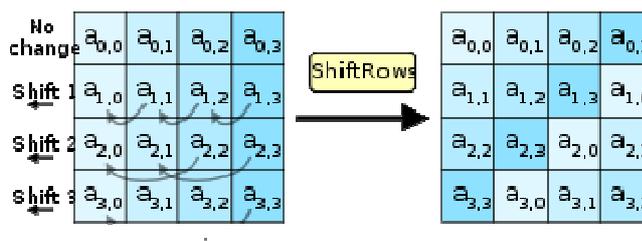


Figura 7 – Etapa *ShiftRows* 128/192 bits³⁴

Por fim, o quarto e último estágio é o *MixColumns*, onde é realizada a operação de mesclagem das colunas (Figura 8) de estado onde os 4 bytes de cada coluna são combinados através de uma transformação linear invertível com o *ShiftRows*. Sua

³³ Etapa *SubBytes*: disponível em <<http://aescryptography.blogspot.com.br/2012/04/subbytes-step.html>>. Acesso em 20 Out. 2013

³⁴ Etapa *ShiftRows* 128/192 bits: disponível em <<http://aescryptography.blogspot.com.br/2012/04/shiftrows-step.html>>. Acesso em 20 Out. 2013

função utiliza os 4 bytes como entrada e gera 4 bytes de saída. Durante essa operação cada coluna é multiplicada por uma matriz interna pré-determinada (Figura 9) cuja multiplicação pelos valores em cada bloco indica o deslocamento a esquerda a ser realizado. Em seguida é efetuada uma operação XOR do valor de deslocamento com os valores iniciais não deslocados. No último turno do processo de cifragem, ao invés de termos um último estágio de *MixColumns*, teremos um estágio *AddRoundKey*.

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Figura 8 – Matriz de multiplicação para blocos de 128 bits³⁵

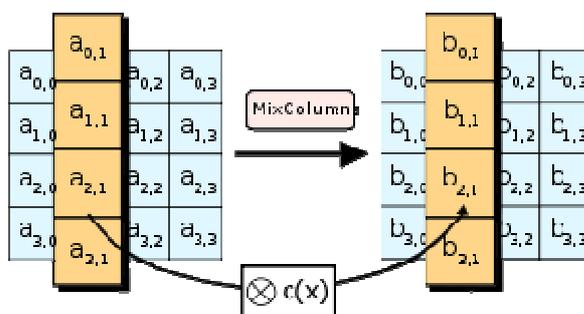


Figura 9 – Etapa *MixColumns*³⁶

Como forma de resolver o principal problema da criptografia simétrica, a transmissão da chave, foi desenvolvida a criptografia assimétrica.

2.4 – Criptografia Assimétrica

A criptografia assimétrica consiste num par de chaves, uma chave pública e uma chave privada. As chaves são correspondentes, ou seja, na criação da chave privada a chave pública é derivada dela, logo as mensagens cifradas com uma das chaves somente poderão ser decifradas com a chave correspondente. Dessa forma, a

³⁵ Matriz de multiplicação para blocos de 128 bits: disponível em <<http://aescryptography.blogspot.com.br/2012/05/mixcolumns-step.html>>. Acesso em 20 Out. 2013

³⁶ Etapa *MixColumns*: disponível em <<http://aescryptography.blogspot.com.br/2012/05/mixcolumns-step.html>>. Acesso em 20 Out. 2013

chave pública pode ser disponibilizada livremente e a chave privada deve ser mantida em segredo.

No processo de cifragem o texto claro é inserido juntamente com a chave pública do destinatário no algoritmo que terá como saída o texto cifrado. Esse texto é enviado ao destinatário que ao recebê-lo, o insere junto com sua chave privada no algoritmo para decifrá-lo obtendo assim o texto claro novamente (Figura 10).

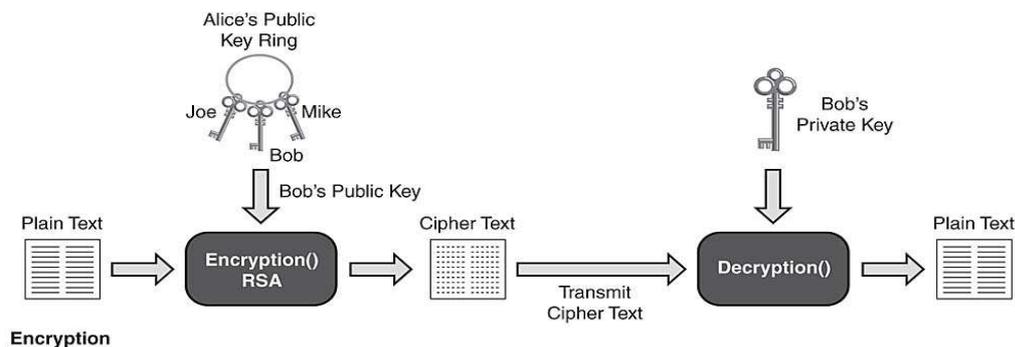


Figura 10 – Criptografia Assimétrica³⁷

A vantagem da criptografia assimétrica é a troca de mensagens cifradas sem a necessidade de envio de chave para descriptografar. Assim, a confidencialidade da mensagem é garantida enquanto a chave privada tiver em poder somente de seu dono. A principal desvantagem desse método é o alto custo computacional dos algoritmos tornando-o mais lento que a criptografia simétrica. Para resolver esse problema, se realiza a cifração do texto claro com a criptografia simétrica e da chave simétrica utilizada com a criptografia assimétrica.

Em 1977 no *Massachusetts Institute of Technology* (MIT), *Ron Rivest*, *Adi Shamir* e *Len Adleman* desenvolveram um algoritmo assimétrico denominado *Rivest-Shamir-Adleman* (RSA) [Silva, E.V.P., 2006]. Esse algoritmo é o mais utilizado atualmente por ser considerado robusto. Porém, o RSA não é o único criptossistema assimétrico existente. Além dele, existem outros como o *Diffie-Hellman* [Ferreira, T., 2007], *Merkle-Hellman* [Shamir, A., 1984], *ElGamal* [Menezes A. & Oorschot P. V. & Vanstone S., 1996] e as Curvas Elípticas [Costa, L.H.M.K. & Duarte, O.C.M.B, 2006].

³⁷ Criptografia Assimétrica: disponível em < <http://www.networkworld.com/subnets/cisco/102208-ch2-ssl-vpn-technology.html>>. Acesso em 20 Out. 2013

A força do RSA basicamente consiste na multiplicação de dois números primos grandes, por ser facilmente calculado pelo computador, e a difícil fatoração do resultado dessa multiplicação para obter os dois números primos novamente. Como exemplo de sua difícil quebra, em 1999 uma chave RSA de 512 bits foi quebrada e para isso foi necessário o trabalho em conjunto de cientistas de 6 países e 300 computadores trabalhando por cerca de 7 meses.

O algoritmo base para gerar as chaves no RSA segundo [Kurose, J. F. & Ross, K. W., 2010] é o seguinte:

1. Escolher dois números primos grandes ‘p’ e ‘q’, da ordem de 1024 bits.
2. Calcular: $n = pq$
3. Calcular: $z = (p - 1)(q - 1)$
4. Escolher um número ‘e’ menor do que ‘n’ e que não tenha fatores comuns com o ‘z’, exceto o 1. A letra ‘e’ será utilizada no processo de cifragem.
5. Achar um número ‘d’, tal que $ed - 1$ seja divisível exatamente por ‘z’. A letra ‘d’ será utilizada para a decifragem. Ou seja, $ed \bmod z = 1$.
6. A chave pública formada é o par de números (n,e) e a chave privada é o par de números (n,d).

Para transformar uma mensagem clara ‘m’ numa mensagem criptografada ‘c’, basta fazer uma potenciação modular através da fórmula: $c = m^e \bmod n$.

Para recuperar a mensagem clara ‘m’ da mensagem criptografada ‘c’ basta fazer outra potenciação modular através da fórmula: $m = c^d \bmod n$.

Como exemplo, suponha que João queira gerar um par de chaves utilizando o algoritmo RSA descrito acima. Ele escolhe $p = 5$ e $q = 7$. Logo temos $n = 35$ e $z = 24$. O valor para ‘e’ escolhido por ele é o 5 por não ter fatores em comum com o valor de ‘z’ e assim um $d = 29$ por ser exatamente divisível por 24. A chave pública de João é o par (35, 5) e a chave privada é o par (35, 29). Caso Maria, namorada do João, queira enviar a palavra “love” para ele criptografada, ela deve usar a chave pública do destinatário (João) e teríamos o seguinte resultado (considere que cada letra do alfabeto é representado de 1 a 26 sendo ‘a’ = 1 e ‘z’ = 26):

Texto Claro	m	m ^e	c = m ^e mod n
l	12	248832	17
o	15	759375	15
v	22	5153632	22
e	5	3125	10

Tabela 4 – Criptografia RSA para a palavra “love”

Quando João recebe a palavra criptografada pela Maria ele pode descriptografar utilizando sua chave privada. Dessa forma teríamos o seguinte:

Texto Criptografado	c ^d	c ^d mod n	Texto Claro
17	48196857210675 x 10 ²⁶	12	l
15	127834039403948 x 10 ²⁴	15	o
22	851643319086537 x 10 ²⁴	22	v
10	1 x 10 ³⁰	5	e

Tabela 5 – Descriptografia RSA para a palavra “love”

Nem sempre há necessidade de confidenciar um dado através da criptografia, seja ela simétrica ou assimétrica. Às vezes é necessário apenas identificar quem a escreveu e evitar que as informações sejam modificadas. Para isso existem alguns mecanismos que ajudam a atingir esses objetivos.

2.5 – Assinatura Digital e Função Hash

A assinatura digital [Trinta, F.A.M., Macedo, R.C., 1998] é um mecanismo que torna possível o envio de uma mensagem garantindo a autenticidade e autoria do remetente. Nesse processo, a mensagem é criptografada com a chave privada do remetente e anexada juntamente com a mensagem original para ser enviada ao destinatário (Figura 11). Para confirmar a autenticidade do remetente e a integridade da mensagem recebida, o destinatário deve descriptografar a mensagem assinada com a chave pública do remetente e realizar a comparação entre eles.

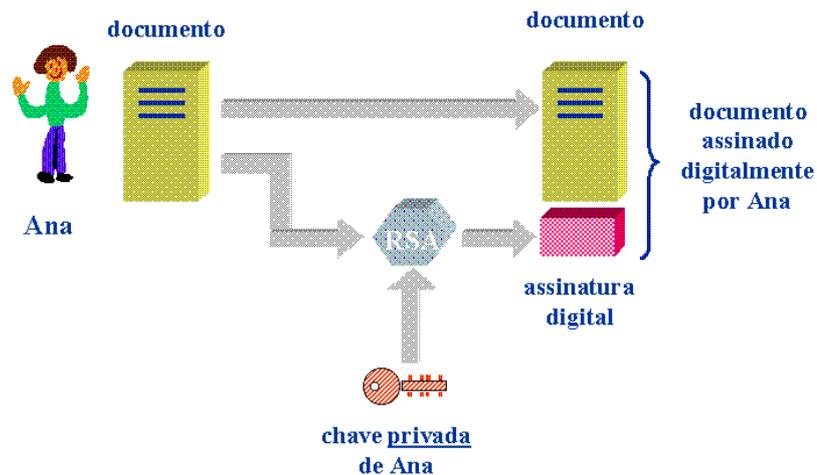


Figura 11 – Assinatura Digital³⁸

Como é inviável, devido a sua lentidão, utilizar algoritmos criptográficos assimétricos para criptografar as mensagens, a solução é a função *hash* [Pisa, P., 2012]. Ela oferece uma assinatura digital rápida e integridade confiável derivado da mensagem que se deseja assinar independente do seu tamanho, funcionando como uma impressão digital dela. Assim, qualquer modificação no conteúdo da mensagem original irá gerar um resumo ou *hash* diferente do original sendo esta modificação detectada na comparação entre *hashes*. A função *hash* também é chamada de *Message Digest*, *One-Way Hash Function*, Função de Condensação ou Função de Espalhamento Unidirecional, sendo as principais funções utilizadas na criptografia o *Message-Digest Algorithm 5* (MD5) [Rivest, R., 1992] e o *Secure Hash Algorithm* (SHA-1) [Eastlake, D. & Jones, P., 2001].

Dessa forma, a assinatura digital passa a ter como procedimento a entrada da mensagem a ser assinada no algoritmo *Message Digest* que irá gerar um valor *hash* da mensagem. Esse *hash* é criptografado com a chave privada do remetente e anexada a mensagem original (Figura 12). Para confirmar a autenticidade do remetente e a integridade da mensagem recebida, o destinatário deve descriptografar o *hash* com a chave pública do remetente, gerar um *hash* da mensagem e realizar a comparação entre eles. (Figura 13).

³⁸ Assinatura Digital: disponível em <http://www.training.com.br/lpmaia/pub_seg_cripto.htm>. Acesso em 20 Out. 2013

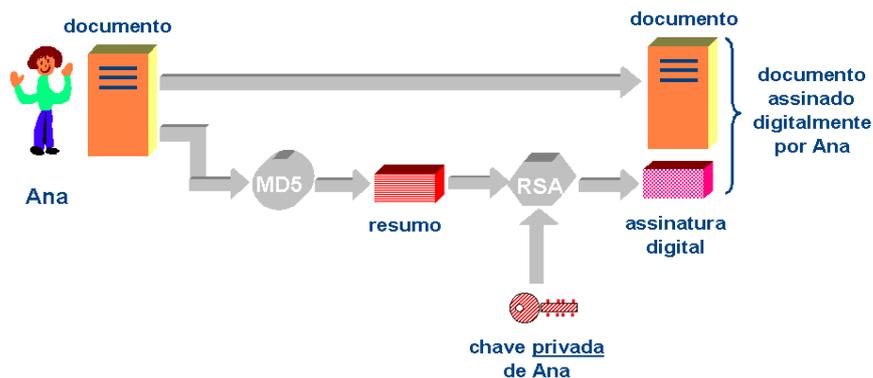


Figura 12 – Criptografia com Assinatura Digital e Função Hash³⁹

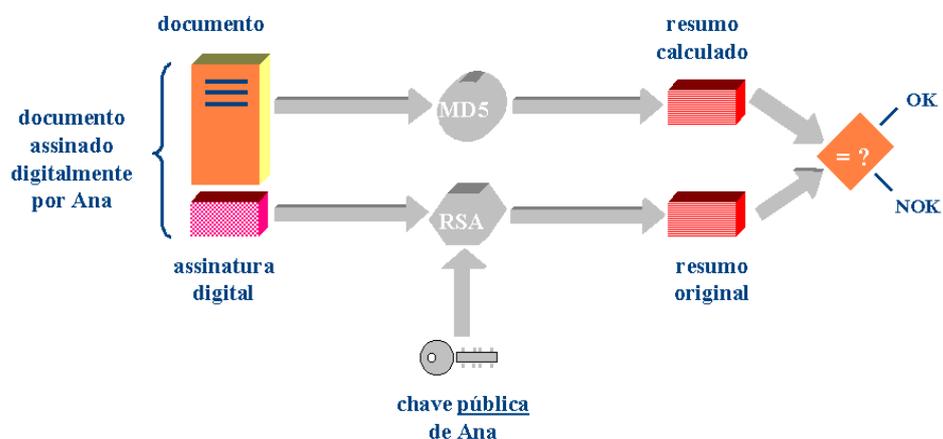


Figura 13 – Descryptografia com Assinatura Digital e Função Hash⁴⁰

2.6 – Web Service

Desde a popularização da internet nos anos 90, novas tecnologias têm sido pesquisadas e desenvolvidas para permitir uma maior integração entre os diversos aplicativos existentes e serviços nas diferentes plataformas através da *web*. Esse tipo de solução é denominada *web service* [Haddad, R., 2013]. Um *web service* utiliza os protocolos padrões desenvolvidos para a web, como o *HiperText Transmission Protocol* (HTTP), para realizar a transmissão de dados que são enviados no formato *Exten-*

³⁹ Criptografia com Assinatura Digital e Função Hash: disponível em <http://www.training.com.br/lpmaia/pub_seg_cripto.htm>. Acesso em 20 Out. 2013

⁴⁰ Descryptografia com Assinatura Digital e Função Hash: disponível em <http://www.training.com.br/lpmaia/pub_seg_cripto.htm>. Acesso em 20 Out. 2013

sible Markup Language (XML) [Pereira, A.P., 2009] e encapsulados pelo protocolo *Simple Object Access Protocol* (SOAP) [Dantas, D.C.T., 2007].

O SOAP é um protocolo baseado em XML para acessar aplicações remotamente ou realizar troca de mensagens num ambiente independente da plataforma ou linguagem de programação utilizada. Este protocolo irá realizar o acesso ao *web service*. Outro componente utilizado é o documento *Web Services Description Language* (WSDL) [Viegas, C., 2008] que é uma linguagem, também baseada em XML, para descrever as interfaces, operações, codificação dentre outras informações pertinentes ao *web service*. Assim, quando o cliente solicitar um serviço, o *web service* envia para ele a descrição do serviço solicitado e o cliente poderá construir sua mensagem passando os dados de acordo com a definição existente no WSDL.

Como principais vantagens dessa solução temos a integração entre sistemas permitindo a reutilização de códigos e sua fácil compreensão, a atualização do sistema de forma centralizada, transparência para o *firewall* uma vez que a comunicação é feita por meio de uma string XML e isola a base de dados dos serviços oferecidos. No próximo capítulo será mostrada a contribuição que a engenharia de software teve na elaboração do projeto SACIS - *Windows PC*.

3 – O Que Há Por Trás das Câmeras

Este capítulo tem o propósito de apresentar a documentação e especificações geradas para o desenvolvimento do projeto SACIS, a definição da sua arquitetura, do gerenciamento dos dados e informações, o ambiente para o qual será desenvolvido, a disponibilidade do código-fonte e o controle de versões utilizado.

3.1 – Processo de *Software*

O processo de *software* é uma área relacionada a todos os aspectos da produção de um sistema, desde o início de suas especificações até sua manutenção, com o objetivo de garantir sua confiabilidade e eficiência [Sommerville, I., 2007][Bauer, F.L., 1968]. Ele é definido por um conjunto de atividades direcionadas ao projeto como a análise de requisitos, projeto, codificação, teste, instalação, recursos necessários e os procedimentos a serem adotados na realização de cada uma das atividades.

O estudo das disciplinas ligadas aos processos de software na academia proporcionou o acesso ao conhecimento dos processos sistematizados capazes de auxiliar e direcionar as atividades relacionadas ao projeto SACIS de forma mais objetiva e estruturada. Dessa forma, foi possível realizar a modelagem do sistema utilizando a linguagem *Unified Modeling Language* (UML) 2.0⁴¹, que possui uma notação que segue os conceitos da orientação a objetos, definindo:

- Regras de Negócio (Anexo I);
- Requisitos Funcionais (Anexo II);
- Requisitos Não-Funcionais (Anexo III);
- Processo de Negócio (Anexo IV);
- Casos de Uso das principais funcionalidades do Sistema (Anexos V a XVII);
- Diagrama de Estado para cada Caso de Uso (Anexos XVIII a XXX);
- Diagrama de Classe (Anexo XXXI).

⁴¹ Disponível em < <http://www.uml.org/> >. Acesso em 24 Nov. 2013.

3.2 – Arquitetura MVC

A arquitetura MVC [Bastos, D.F., 2011] foi adotada para o desenvolvimento do projeto a fim de facilitar a extensão do sistema, o reaproveitamento e a manutenção do código e diminuir sua complexidade. A principal característica do padrão MVC é a divisão em 3 camadas (Modelo-Visão-Controle) das funcionalidades da aplicação, onde cada camada tem funções bem definidas. A arquitetura MVC desenvolvida para o projeto SACIS está ilustrada na figura 14.

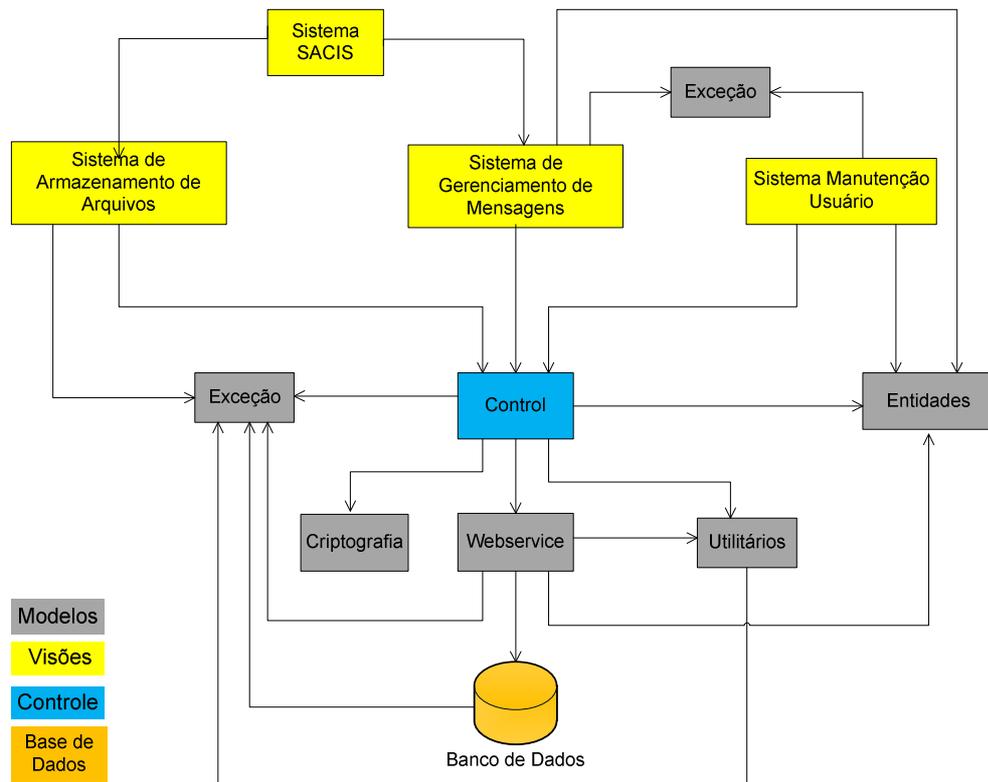


Figura 14 – Arquitetura MVC do Projeto SACIS

A camada de Modelo disponibiliza funcionalidades que permitem à camada de Controle o acesso encapsulado aos dados. Esta camada tem total acesso aos dados e seu objetivo é armazenar as informações enviadas pelo usuário, realizar consultas, manipular, gerar e modelar os dados de acordo com as regras de negócio definidas para o acesso e modificação da informação. No projeto, todas as informações solicitadas ou enviadas pela camada de Controle são manipuladas e informadas pelo *web service*. Os pacotes *sacis.model* (Figura 15) representam essa camada.

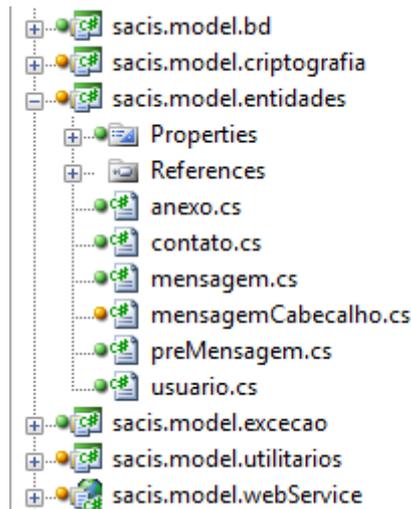


Figura 15 – Camada de Modelo do Projeto SACIS

A camada de Controle é responsável por controlar todo o fluxo de informações entre a camada de Visão e a de Modelo. Cabe a ela controlar e mapear todas as ações do usuário selecionando o modelo correto disponível que retorne a informação solicitada pelo usuário. Esta camada é definida no projeto pelo pacote `sacis.control.servlet`, conforme ilustra a figura 16, que contém as lógicas para cada sistema desenvolvido para o projeto.

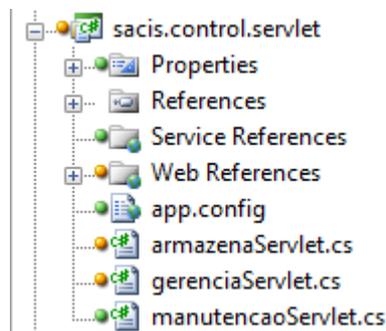


Figura 16 – Camada de Controle do Projeto SACIS

A última camada, Visão, é responsável pela interface ao qual o usuário irá visualizar e receber as informações de acordo com as ações requisitadas por ele. No projeto SACIS, esta camada é definida pelas telas ao qual o usuário tem acesso para realizar as ações de *login*, manipulação dos arquivos locais, gerenciamento de mensagens e de seus contatos. Na aplicação elas estão evidenciadas nos pacotes `sacis.view.sistema` conforme ilustra a figura 17.

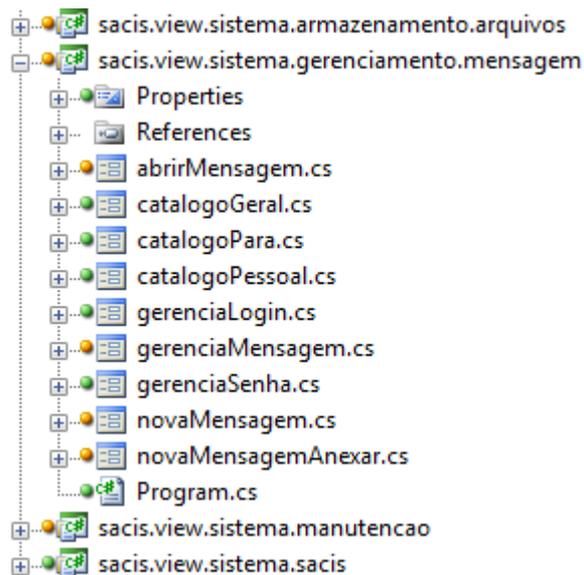


Figura 17 – Camada de Visão do Projeto SACIS

3.3 – Ambiente de Desenvolvimento

Este projeto implementa uma aplicação para a família de dispositivos que possuem sistema operacional *Windows*. A fim de garantir a compatibilidade entre as diferentes versões do sistema operacional, a implementação da aplicação foi realizada na linguagem de programação C# tendo como ambiente de desenvolvimento o *software Visual Studio 2008*⁴².

As dificuldades encontradas durante o desenvolvimento da aplicação foram resolvidas em sua maioria com pesquisas. Houve apenas uma dificuldade relacionada à criptografia assimétrica que demandou uma pesquisa mais profunda, cuja solução será apresentada na seção 3.6 neste mesmo capítulo.

3.4 – Repositório de Código e Controle de Versão

Para disponibilizar o código fonte e a documentação do projeto SACIS foi escolhido o *Google Code*⁴³ como repositório de código. O *Google Code* é um serviço oferecido pela empresa *Google* para hospedagem colaborativa de projetos com código-

⁴² Disponível em < <http://microsoft-visual-studio-2008.software.informer.com/>>. Acesso em 20 Nov. 2013.

⁴³ O código fonte e a documentação do projeto SACIS podem ser acessados em <<https://code.google.com/p/sacis/>>.

go aberto e disponibiliza 2 Gb de espaço para hospedagem de código e um sistema de gerenciamento de versões compatível com o *Git*, *Subversion* e *Mercurial*.

O sistema de gerenciamento de versão escolhido foi o *subversion* por sua compatibilidade com o *Visual Studio* sendo necessária apenas a instalação do *plug-in Visual SVN*⁴⁴. Através desse *plug-in* é possível gerenciar de forma fácil e intuitiva as modificações realizadas e existentes entre os códigos local e remoto.

3.5 – Armazenamento dos Dados e Informações

Os dados e informações utilizados pelo projeto SACIS são armazenados em dois lugares distintos.

3.5.1 – Ambiente Local

O projeto prevê que o *login* e o *hash* da senha dos usuários que o utilizam necessitam ser armazenados num arquivo criado por ele no dispositivo. Esses dados são necessários para que o usuário possa ter acesso e utilizar o Sistema de Armazenamento de Arquivos que independe de conexão com o servidor *web*. O risco de se armazenar o *hash* da senha no ambiente local é um atacante conseguir acesso ao sistema ao enviá-lo junto com o *login* através de um ataque de injeção de código.

3.5.2 – Servidor Web

Um servidor *web* é responsável por armazenar e realizar trocas de informações no formato HTML ou XML entre cliente-servidor utilizando protocolo HTTP. Por essas características estarem de acordo com as necessidades do projeto, o servidor *web* desenvolvido para o SACIS disponibiliza para a troca de informações através do *web service* seus serviços em um WSDL⁴⁵ para serem consumidos independente da tecnologia utilizada do lado cliente.

⁴⁴ Disponível em: < <http://visualstudiogallery.msdn.microsoft.com/DBD60715-FE57-44B5-ABEA-F18618068C1E> >. Acesso em 20 Nov. 2013.

⁴⁵ O WSDL com os serviços disponíveis pelo projeto SACIS pode ser acessado em <<http://sacis.com.br/ws/Service1.asmx?wsdl>>.

O armazenamento das informações e dados é feito de duas formas. A primeira forma está relacionada com os 3 tipos de arquivos manipulados no servidor *web*: as mensagens enviadas, os catálogos de contatos pessoais e as chaves certificadas. Cada usuário possui uma pasta individual contendo subpastas (contatos, entrada e enviados) que é criada para ele ao ser cadastrado no sistema. As mensagens enviadas e recebidas são armazenadas no formato XML (o qual permite a interoperabilidade da aplicação entre as diversas plataformas sejam elas de dispositivos móveis ou fixos) com extensão ‘.msg’ nas pastas de ‘entrada’ ou ‘enviadas’ de acordo com seu tipo informado na base de dados. Importante notar que o envio das mensagens é realizado através do protocolo HTTP e não pelo protocolo *Simple Mail Transfer Protocol* (SMTP)⁴⁶ por não serem e-mails. O catálogo de contatos pessoais também é salvo no formato XML, porém com extensão ‘.cnt’ na pasta ‘contato’. As chaves certificadas dos usuários ficam armazenadas na pasta ‘chaveiro’, que não faz parte da pasta individual dos usuários.

A segunda forma é através da utilização de um Sistema de Gerenciamento de Banco de Dados (SGBD) para salvar os dados referentes aos usuários do sistema e as informações básicas das mensagens a fim de facilitar sua listagem na tela principal do Sistema de Mensagens. Um SGBD proporciona um melhor gerenciamento, manipulação e organização dos dados. O projeto SACIS utiliza a versão 5.1.20 do *MySQL*⁴⁷. Ele é um SGBD gratuito, de fácil usabilidade, rápido, confiável e capaz de suportar grandes tráfegos que disponibiliza uma ferramenta gráfica com o intuito de auxiliar a visualização e manipulação dos dados e tabelas de forma intuitiva, o *MySQL Query Browser*⁴⁸. A versão 1.1.20 é a utilizada. Para armazenar os dados que garantem o acesso e listam as mensagens recebidas e enviadas pelos usuários foram criadas três tabelas: *usuario*, *mensagem* e *seq_mensagem*, conforme ilustra o modelo relacional na figura 18.

Cada tabela possui uma função definida. A tabela ‘*usuario*’ armazena os registros referentes ao cadastro; a de ‘*mensagem*’ é responsável por armazenar as in-

⁴⁶ Disponível em <<http://computer.howstuffworks.com/e-mail-messaging/email3.htm>>. Acesso em 18 Jan. 2013.

⁴⁷ Disponível em <<http://www.mysql.com/>>. Acesso em 19 Nov. 2013.

⁴⁸ Disponível em <<http://imasters.com.br/artigo/8530/mysql/trabalhando-com-o-mysql-query-browser-parte-01/>>. Acesso em 19 Nov. 2013.

formações básicas das mensagens enviadas e recebidas; enquanto a ‘seq_mensagem’ provê um valor numérico sequencial para a tabela mensagem. O dicionário de dados e os scripts para gerar a base de dados podem ser visualizados nos anexos XXXII e XXXIII, respectivamente.

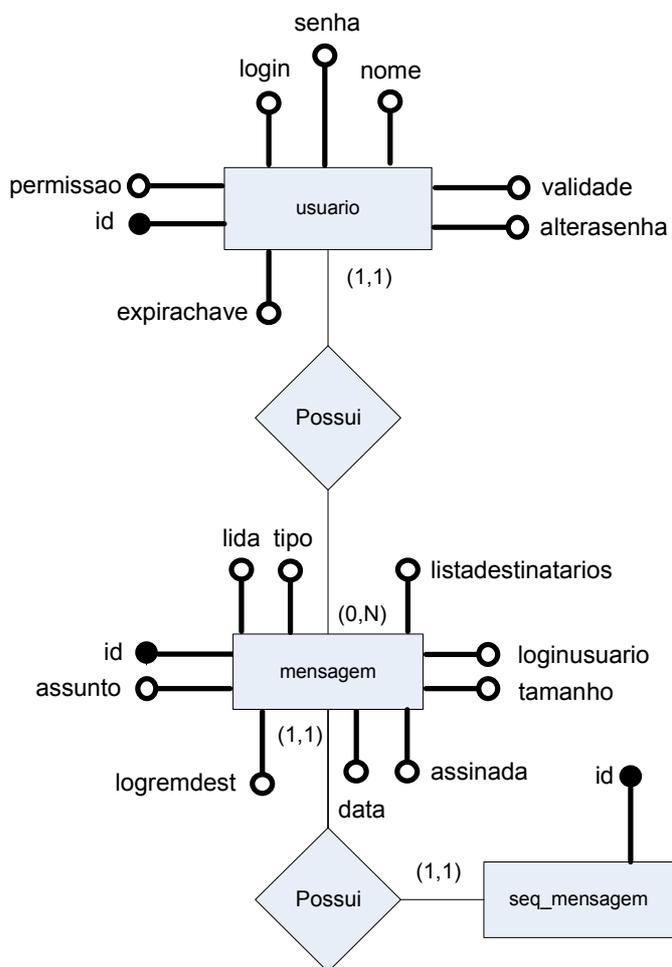


Figura 18 – Modelo Relacional

3.6 – Biblioteca Externa

Os programas utilizados para gerar um par de chaves certificada e testar o projeto foram o *KeyStore Explorer* 4.0.1⁴⁹ e o *OpenSSL*⁵⁰, ambos *open source*. A chave certificada pode ser originária de qualquer autoridade certificadora⁵¹ no formato

⁴⁹ Disponível em: < <http://keystore-explorer.sourceforge.net/>>. Acesso em 10 Dez. 2013.

⁵⁰ Disponível em: < <http://www.openssl.org/>>. Acesso em 10 Dez. 2013.

⁵¹ Disponível em: < <http://serasa.certificadodigital.com.br/perguntas-frequentes/o-que-e-a-autoridade-certificadora-ac/>>. Acesso em 16 Jan. 2014.

Privacy-Enhanced Mail (PEM) [Kent, S.T., 2006] e não pode conter senha. Para garantir uma criptografia com maior segurança é recomendado um tamanho mínimo da chave de 1024 bytes. Como a leitura da chave privada no formato PEM não é possível utilizando os métodos criptográficos disponíveis na linguagem de programação C#, foi necessário utilizar uma biblioteca criptográfica externa⁵², desenvolvida pela organização *Bouncy Castle*⁵³. Dessa forma, é possível ler a chave privada e realizar a criptografia assimétrica e a assinatura digital.

O seguir será descrito o funcionamento e apresentada a ferramenta desenvolvida que visa proporcionar o envio de mensagens e armazenamento de arquivos de forma segura.

⁵² Disponível em: <<http://www.bouncycastle.org/csharp/>>. Acesso em 10 Dez. 2013.

⁵³ Disponível em: <<http://www.bouncycastle.org/index.html>>. Acesso em 10 Dez. 2013.

4 – SACIS Para *Windows PC*

Neste capítulo é apresentado o sistema SACIS - *Windows PC* e seu funcionamento. A aplicação é dividida em dois sistemas (Figura 19): o *Sistema de Manutenção de Usuários*, utilizado pelo administrador, e o *Sistema de Manipulação de Informação*, utilizado pelo usuário. Este último é subdividido no *Sistema de Armazenamento de Arquivos*, no qual é realizado o armazenamento local de dados criptografados, e o *Sistema de Gerenciamento de Mensagens*, que é responsável pelo gerenciamento de chaves e envio de mensagens e arquivos criptografados.

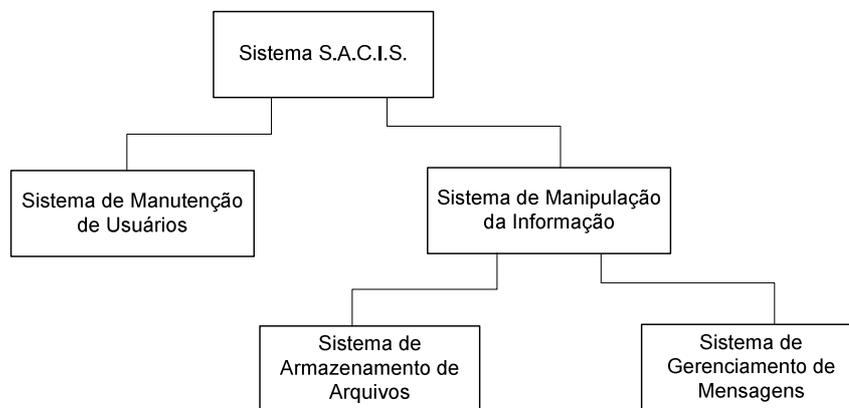


Figura 19 – Organograma Geral do Sistema

Cada um dos 3 sistemas são acessados através de uma tela padrão (Figura 20) na qual o usuário deverá inserir seu *login* e senha que serão validados. As telas de acesso de cada sistema possuem validações específicas que serão descritas no decorrer do capítulo.



Figura 20 – Tela Padrão de Login

4.1 – Sistema de Manutenção de Usuários

Neste sistema o administrador poderá cadastrar, alterar e excluir o usuário através de abas específicas para cada funcionalidade. Para cadastrar o usuário, o administrador deverá inserir os dados solicitados (nome e *login*), definir o tipo de permissão que o cadastrado terá (usuário ou administrador) e, além disso, incluir o certificado informado pelo usuário, conforme ilustra a Figura 21.

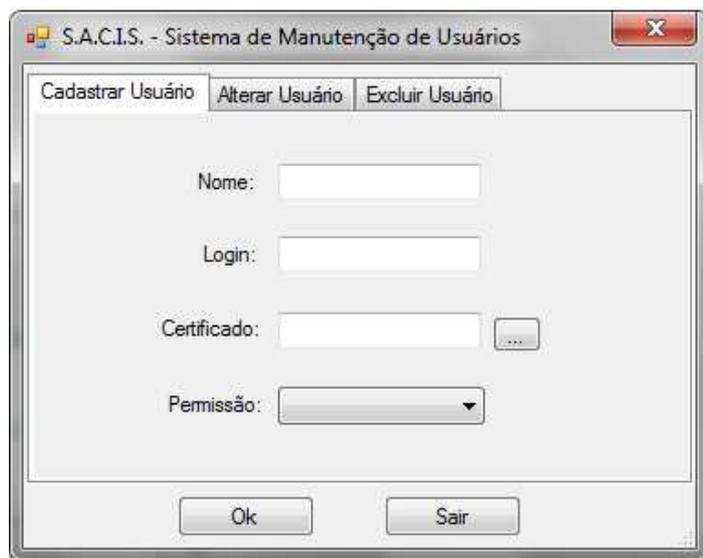


Figura 21 – Telas de Cadastro de Usuários

Para o cadastro ser realizado com sucesso, cada campo precisa ser validado. O primeiro campo a ser validado é o ‘Certificado’. A mensagem “Certificado Inválido!” (Figura 22) será exibido ao usuário se ele não informar nenhum arquivo. Caso ele informe um certificado inválido ou com senha, a mensagem “Certificado Inválido ou com senha!” irá aparecer (Figura 23).

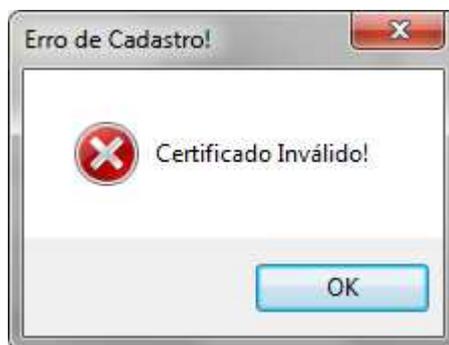


Figura 22 – Mensagem: “Certificado Inválido!”

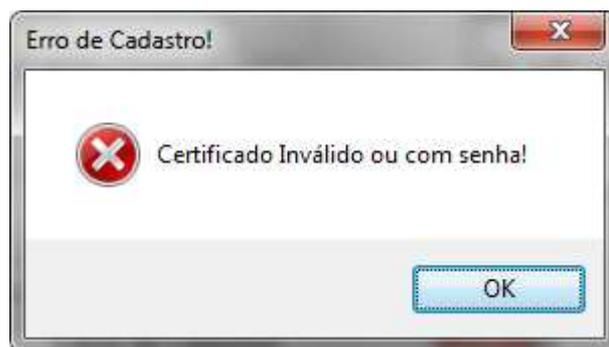


Figura 23 – Mensagem: “Certificado Inválido ou com senha!”

Caso a data seja menor do que a atual a mensagem “Certificado Expirado!” irá retornar ao usuário. O segundo campo é o da ‘Permissão’. Se nenhuma opção for selecionada o sistema também retornará a mensagem “Selecione um Tipo de Permissão!” (Figura 24).

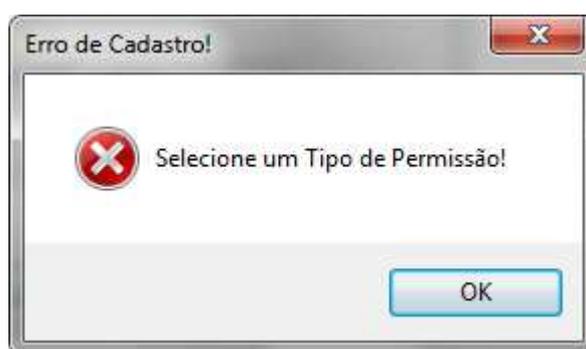


Figura 24 – Mensagem: “Selecione um Tipo de Permissão!”

Os próximos campos são os de ‘Nome’ e ‘Login’ cujos conteúdos devem atender as seguintes condições:

- Não podem ser iguais;
- Não podem ter menos de 8 caracteres;
- Não podem ser vazios ou nulos;
- Não podem iniciar ou terminar com caracteres de espaço e *delete*;
- Seus caracteres devem estar entre 32 e 255 da tabela ASCII⁵⁴;
- Deve conter apenas um espaço entre os caracteres.

Após realizar essas validações no lado cliente, o sistema verificará a existência do *login* informado no banco de dados. Caso este não exista, os dados serão inse-

⁵⁴ Disponível em < <http://www.asciitable.com/> >. Acesso em 23 Nov. 2013.

ridos na base de dados, o qual armazenará as informações de cadastro do usuário (nome, *login*, *hash* da senha, chave pública, validade da chave pública e permissão). Caso contrário será informado ao administrador a existência do usuário.

Continuando o cadastro no banco de dados, o sistema cria uma pasta individual no servidor para o usuário. Esta contém subpastas para as mensagens recebidas, enviadas e contatos. Ainda nesta fase, o sistema irá padronizar a chave pública certificada (conforme será descrito na seção 4.3) e salvá-la no chaveiro existente no servidor para que ele possa utilizá-la de forma mais eficiente no processo de cifração e decifração.

Na tela de alteração do usuário, o administrador deve informar obrigatoriamente o *login* do usuário a ter seus dados alterados e selecionar ou digitar a(s) opção(ões) desejada(s) (Alterar Senha, Nome, Certificado ou Permissão conforme Figura 25). Ao ser solicitada a alteração dos dados requeridos, o sistema realizará as validações conforme faz no ato do cadastro. Uma vez validados, os dados são salvos no banco de dados e pastas de acordo com o solicitado. Para efetuar a exclusão, o administrador deve informar um *login* a ser verificado pelo sistema. Confirmada sua existência, o sistema irá excluir todos os dados referentes ao usuário na base de dados, bem como as pastas e certificados a ele associados no servidor.

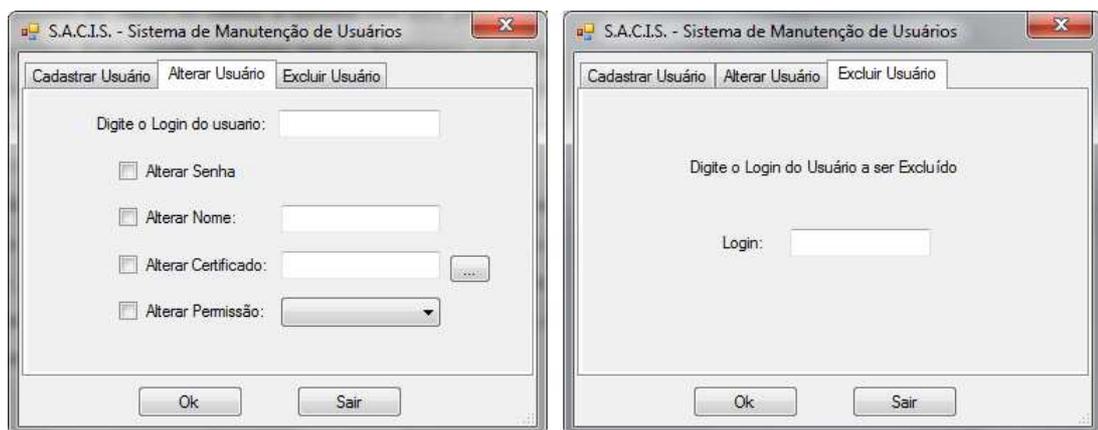


Figura 25 – Telas de Alteração de Dados e Exclusão dos Usuários

Para acessar o Sistema de Manutenção o usuário deverá ter privilégios de administrador o qual é verificado no momento do acesso. Os demais usuários utilizarão apenas o sistema destinado para a manipulação de informações.

4.2 – Sistema de Manipulação da Informação

Este sistema é dividido em duas funcionalidades: criptografia/descriptografia de arquivos locais (Sistema de Armazenamento de Arquivos) e o envio/recebimento de mensagens com ou sem anexos, os quais poderão ser cifrados ou não (Sistema de Gerenciamento de Mensagens).

4.2.1 – Sistema de Gerenciamento de Mensagens

Para acessar o sistema de envio e recebimento de mensagens é necessário que o usuário preencha os campos de *login* e senha da tela de acesso. Antes de permitir sua entrada, o sistema verifica se o usuário necessita trocar a senha. Isso ocorre no caso de ser o primeiro acesso após o cadastro de seu perfil ou ao solicitar a mudança de senha. Nessas situações, o usuário não precisa preencher o campo senha. Detectada a necessidade de troca de senha, através do *login*, será exibida uma mensagem informando a expiração da senha e a tela para digitar uma nova ficará disponível (Figura 26). Uma vez validada a nova senha é salva na base de dados.

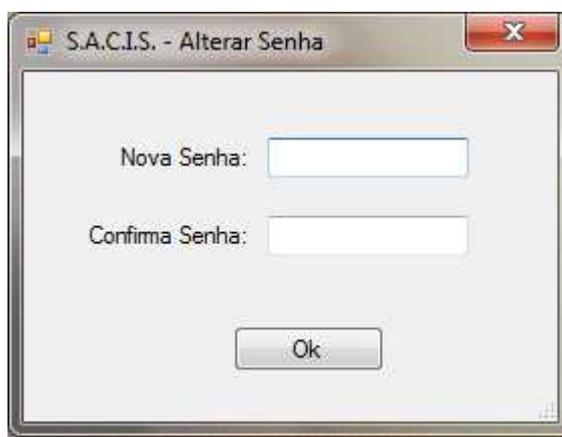
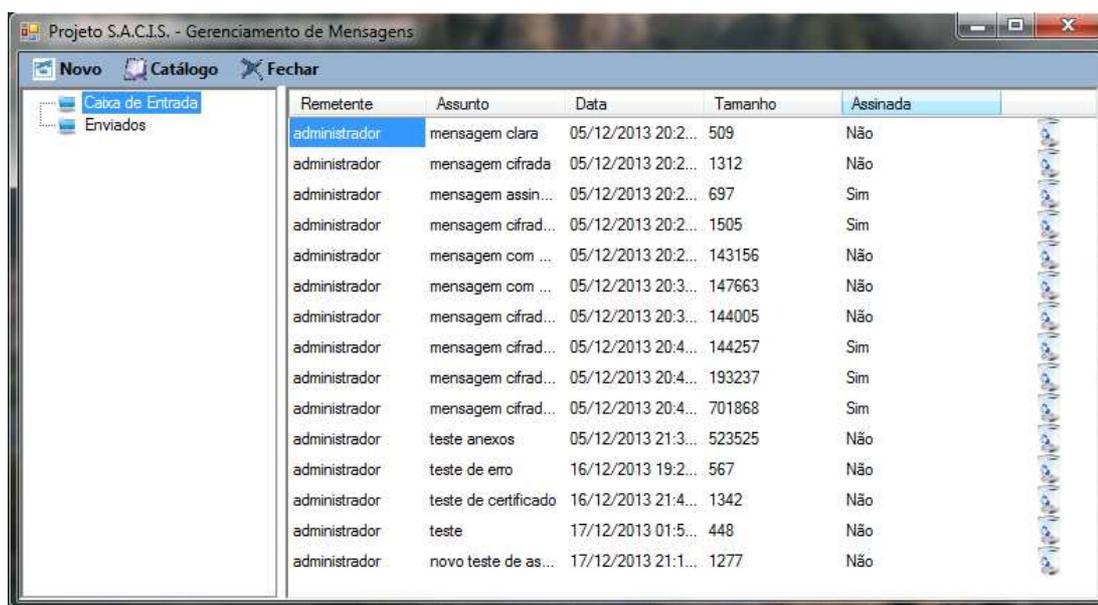


Figura 26 – Tela Nova Senha

Não havendo a necessidade de troca de senha, as validações básicas das credenciais para permitir ao usuário o acesso são feitas. Ao ser garantido seu acesso, o sistema irá verificar localmente a existência do registro do usuário. Caso não exista o registro, este será realizado copiando dados do servidor para uma pasta local no dispositivo. Caso o usuário exista e a sua senha não coincida localmente, o servidor irá

atualizar o arquivo local automaticamente naquele dispositivo em que foi realizado o acesso ao Sistema de Gerenciamento de Mensagens.

Ao ser garantido seu acesso, uma tela de gerenciamento de mensagens é exibida ao usuário, mostrando as pastas de mensagens (Caixa de Entrada e Enviados) e o *menu* contendo as opções de nova mensagem, catálogo e fechar (Figura 27). Os dados são visualizados do servidor, permitindo sua manipulação (visualização e remoção) na caixa de entrada e enviados.



Remetente	Assunto	Data	Tamanho	Assinada
administrador	mensagem clara	05/12/2013 20:2...	509	Não
administrador	mensagem cifrada	05/12/2013 20:2...	1312	Não
administrador	mensagem assin...	05/12/2013 20:2...	697	Sim
administrador	mensagem cifrad...	05/12/2013 20:2...	1505	Sim
administrador	mensagem com ...	05/12/2013 20:2...	143156	Não
administrador	mensagem com ...	05/12/2013 20:3...	147663	Não
administrador	mensagem cifrad...	05/12/2013 20:3...	144005	Não
administrador	mensagem cifrad...	05/12/2013 20:4...	144257	Sim
administrador	mensagem cifrad...	05/12/2013 20:4...	193237	Sim
administrador	mensagem cifrad...	05/12/2013 20:4...	701868	Sim
administrador	teste anexos	05/12/2013 21:3...	523525	Não
administrador	teste de erro	16/12/2013 19:2...	567	Não
administrador	teste de certificado	16/12/2013 21:4...	1342	Não
administrador	teste	17/12/2013 01:5...	448	Não
administrador	novo teste de as...	17/12/2013 21:1...	1277	Não

Figura 27 – Tela principal do Sistema de Mensagens

Ao solicitar a criação de uma nova mensagem é exibida uma tela (Figura 28) onde pode-se selecionar os contatos pessoais, editar o assunto, anexar arquivos, digitar a mensagem e assinalar a criptografia/assinatura da mensagem. Para anexar um documento é apresentado uma tela para a seleção e indicação de quais serão criptografados (Figura 29). A seleção dos contatos será realizada através do catálogo pessoal do usuário que está armazenado no servidor ou digitando diretamente o identificador (endereço eletrônico) no campo do(s) destinatário(s).



Figura 28 – Tela de Nova Mensagem

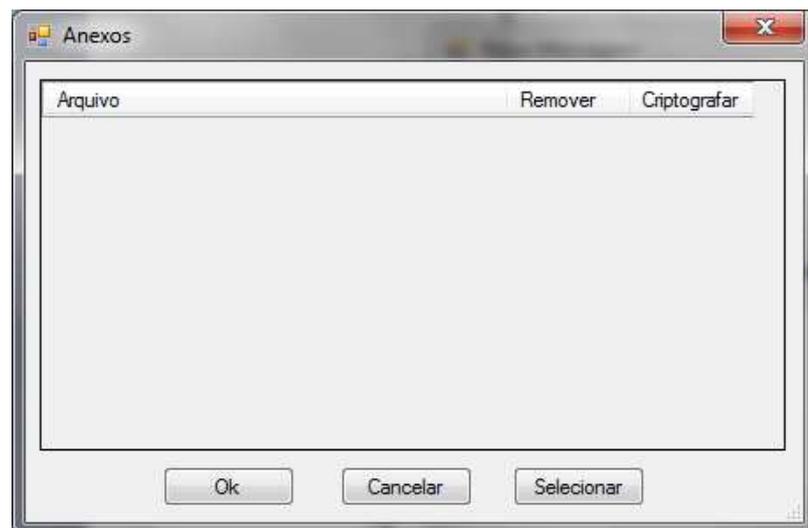


Figura 29 – Tela para Anexar Arquivos

Ao solicitar o envio da mensagem, o sistema irá validar os destinatários e verificar se suas chaves certificadas não expiraram. Caso o destinatário não seja informado ou seja inválido, os seus respectivos avisos “Mensagem não pode ser enviada sem um destinatário!” e “Verifique Contato Inválido!” serão exibidos para o usuário (Figura 30).

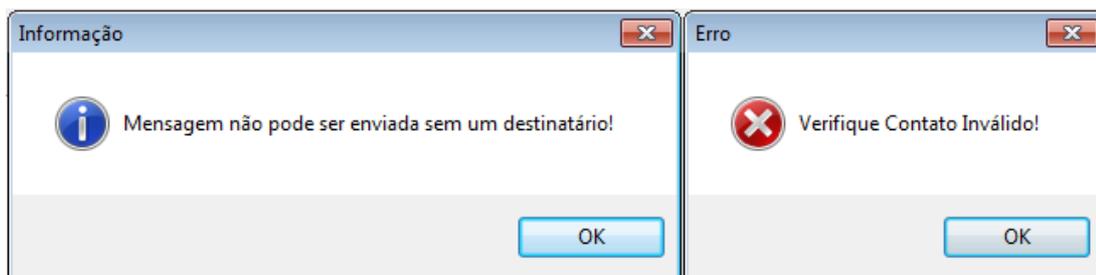


Figura 30 – Mensagem caso destinatário não seja informado ou seja inválido

Em seguida, será verificado quais arquivos anexados deverão ser cifrados e se a mensagem deverá ser cifrada e/ou assinada. A mensagem e o(s) arquivo(s) anexado(s) serão cifrados simetricamente, quando solicitados, onde uma chave será gerada automaticamente pelo sistema e esta será cifrada assimetricamente com a chave pública do destinatário. Se a mensagem estiver sendo assinada, será solicitada ao usuário sua chave privada. Ao final da montagem da mensagem, esta é transformada em XML e enviada ao servidor *web* onde será armazenada em um arquivo (Figura 31) na pasta de entrada do destinatário e uma cópia é salva na pasta de enviados do remetente.

```

<mensagem>
  <de> Remetente </de>
  <para> Destinatario </para>
  <assunto> Assunto da mensagem </assunto>
  <texto> Corpo da mensagem </texto>
  <assinatura> Assinatura da mensagem </assinatura>
  <criptografar> Valor booleano </criptografar>
  <assinar> Valor booleano </assinar>
  <anexos>
    <anexo>
      <nome> Nome do arquivo </nome>
      <cripto> Valor booleano </cripto>
      <chave |>
      <conteudo> Conteudo do arquivo </conteudo>
    </anexo>
  </anexos>
</mensagem>

```

Figura 31 – Formato XML para mensagens

Ao terminar o envio da mensagem, ocorre uma notificação ao usuário. Esta irá se referir ao envio da mensagem sem nenhum problema (Figura 32) ou quando ela é enviada para parte da lista de destino contendo, dessa forma, destinatários não existentes ou certificados expirados (Figura 33).

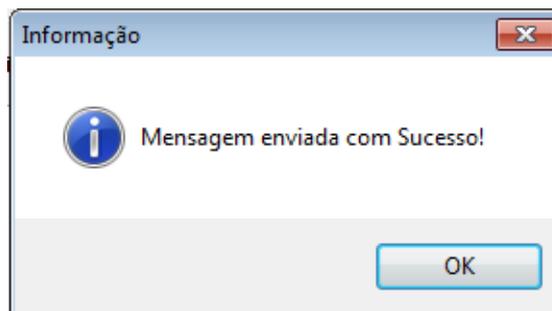


Figura 32 – Mensagem enviada com sucesso

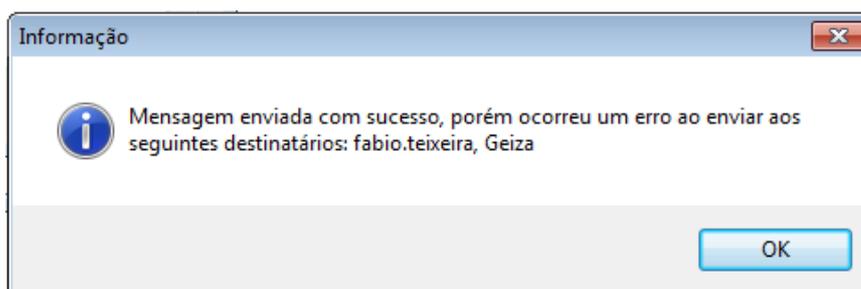


Figura 33 – Lista com destinatários não existentes ou certificados expirados

Para facilitar a manipulação das mensagens que o usuário possui, as informações como o remetente, o destinatário, o tipo da mensagem (enviada ou recebida), a data do envio, o assunto, o tamanho da mensagem, a lista de destinatários e se ela foi lida ou não são armazenadas no banco de dados. A geração de chaves simétricas no Sistema de Gerenciamento de Mensagens é realizada utilizando um algoritmo baseado no MERSENNE⁵⁵, um gerador de números aleatórios.

Para o usuário acessar suas mensagens, elas são exibidas automaticamente na caixa de entrada. Ao solicitar sua abertura, o sistema verifica se a mensagem tem seu corpo de texto cifrado e/ou assinado. Ele decifra a chave simétrica usada para criptografar a mensagem, após solicitar ao usuário sua chave privada, e realiza a descryptografia do texto. Caso seja bem sucedido será exibido todo o conteúdo da mensagem em texto claro. Se assinada, o sistema utilizará a chave pública do remetente para resgatar o *hash* do texto, garantindo a autenticidade do remetente. Em seguida ele cria um *hash* do texto recebido e compara os dois *hashes*, o assinado digitalmente e o criado, para verificar sua integridade. Caso os dois resumos não coincidam, a mensa-

⁵⁵ Disponível em <<http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html>>. Acesso em 28 Out. 2013

gem “Não foi possível abrir a mensagem. Motivo: Mensagem Corrompida” será exibida ao usuário (Figura 34).

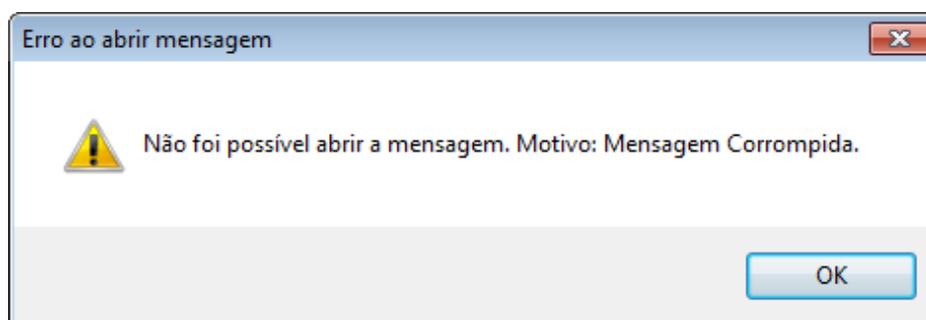


Figura 34 - Aviso caso conteúdo da mensagem tenha sido alterado.

Um duplo clique sobre o anexo permite sua visualização. O sistema vai ao servidor *web* buscar o conteúdo do anexo e armazena este numa pasta temporária para sua abertura. Neste primeiro momento, os anexos somente poderão ser salvos através do programa associado a sua extensão. Caso ele esteja cifrado, o sistema irá executar os mesmos passos usados para descriptografar a mensagem, conforme citado acima, antes de enviá-lo para a pasta temporária. Esta pasta é limpa quando a mensagem é enviada ou quando o usuário sai do sistema.

Através da tela de visualização de mensagens (Figura 35), é possível responder ou encaminhar uma mensagem executando o mesmo processo de criação de uma mensagem nova.

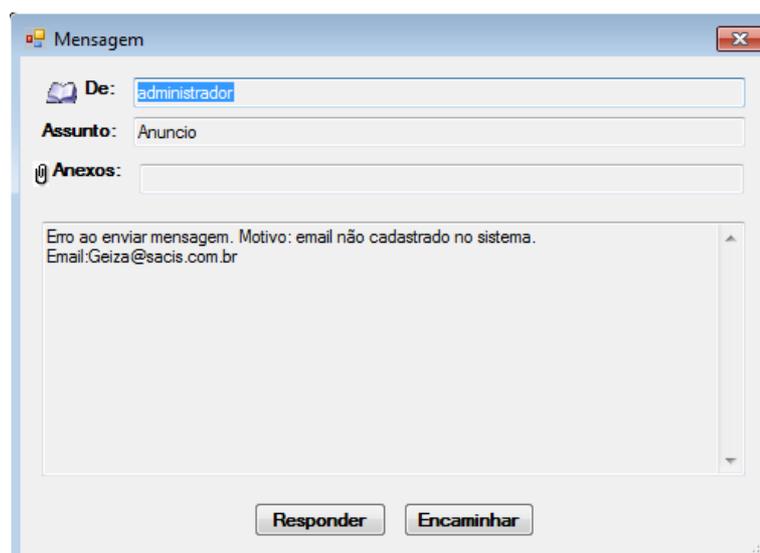


Figura 35 – Tela de Visualização de Mensagem

Na tela principal do Sistema de Gerenciamento de Mensagens, os contatos pessoais poderão ser adicionados ao catálogo pessoal do usuário através do catálogo geral que lista todos os usuários cadastrados no sistema. Acessando o catálogo pessoal é possível excluir os contatos desejados que foram incluídos pelo usuário conforme ilustra a Figura 36.

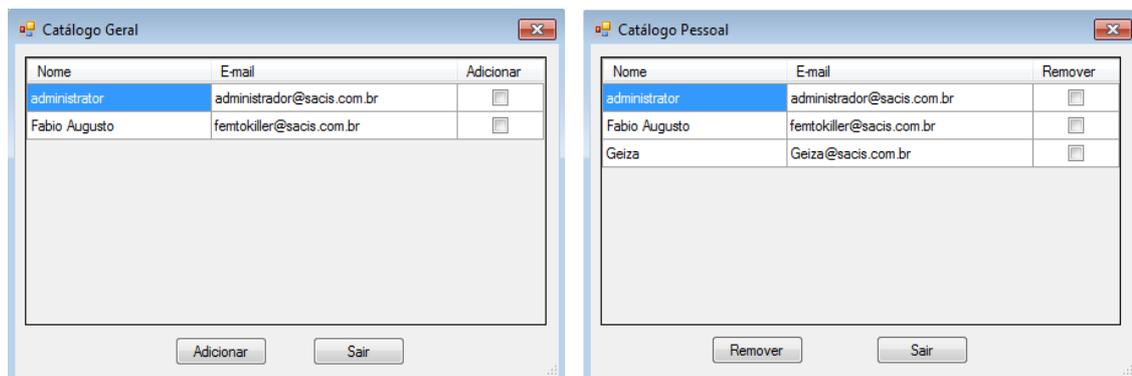


Figura 36 – Telas de Contatos Geral e Pessoal

Além de gerenciar as mensagens e arquivos a serem enviados a um destinatário, o Sistema de Manipulação da Informação também oferece o armazenamento dos arquivos com segurança no próprio dispositivo.

4.2.2 – Sistema de Armazenamento de Arquivos

O sistema de armazenamento de arquivos realiza a cifração e decifração de arquivos locais. Para entrar no sistema é necessário um *login* “semi-independente” da conexão com o servidor, pois para utilizar o sistema o usuário deverá ter realizado ao menos um acesso ao servidor *web* para que possa ser registrado localmente. Neste processo, alguns dados do servidor são copiados para uma pasta local no dispositivo ficando armazenado num arquivo o *hash* da senha e o *login* do usuário. Dessa forma não há o comprometimento da segurança da informação do usuário.

Dado que o usuário esteja registrado localmente será aberta a tela principal do sistema com as abas para criptografar e descriptografar (Figura 37). Na aba criptografar, após escolher os arquivos através do botão ‘Selecionar’, será solicitado ao usuário, ao executar a ação, uma frase-senha que será a chave utilizada pelo sistema para criptografar os arquivos com o AES. Cada arquivo é criptografado com a mes-

ma frase-senha individualmente e tem sua extensão alterada para o padrão do sistema.

A aba de descriptografar funciona de forma semelhante à de criptografar, porém ao solicitar sua ação, o sistema irá pedir a frase-senha de cada arquivo criptografado escolhido para executar sua função restaurando o arquivo original na pasta destinada pelo usuário.

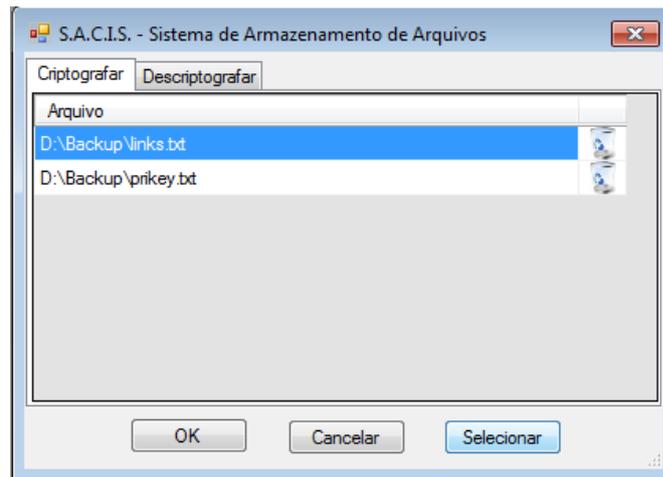


Figura 37 – Tela principal do Sistema de Gerenciamento de Arquivos

Para todo o Sistema de Manipulação da Informação funcionar adequadamente, também se faz necessário que haja um controle sobre as chaves informadas pelos usuários para apoiar o uso das técnicas criptográficas utilizadas. Para essa tarefa o sistema possui um gerenciamento de chaves que será abordado a seguir.

4.3 – Gerenciamento de Chaves

As chaves certificadas informadas, no momento do cadastro ou alteração da mesma, são padronizadas. Essa padronização ocorre da seguinte forma:

1. O conteúdo da chave certificada é lido no lado cliente da ferramenta;
2. O conteúdo é enviado ao servidor *web*;
3. É criado um novo arquivo no chaveiro do servidor *web* composto pelo *login* do usuário associado à chave e a extensão '*key*' (ex: fabio.key);
4. O conteúdo da chave é salvo no arquivo criado.

O objetivo da padronização da chave certificada é automatização do seu uso pelo sistema no envio de mensagens e arquivos criptografados e na verificação da assinatura. Além disso, a data de expiração da chave é salva na base de dados para que esta seja checada diariamente por uma *procedure*⁵⁶ (anexo XXXIV). Ao faltar trinta dias para a sua expiração, o sistema irá avisar a cada *login* que o usuário fizer na aplicação os dias restantes de uso do sistema (Figura 38). Ao atingir a data de sua expiração, a chave não poderá mais ser utilizada até que a mesma seja trocada.

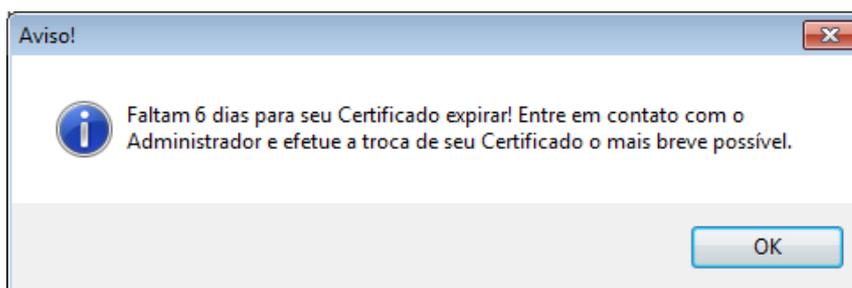


Figura 38 – Aviso de Expiração de Certificado

As chaves simétricas usadas para criptografar mensagens e arquivos tanto no Sistema de Gerenciamento de Mensagens quanto no Sistema de Armazenamento de Arquivos são geradas aleatoriamente para cada um deles, não havendo dessa forma repetição de chaves. O próximo capítulo irá realizar algumas comparações entre as ferramentas existentes com o sistema desenvolvido neste trabalho.

⁵⁶ *Procedure* – É uma sub-rotina executada pelo banco de dados para cumprir uma determinada tarefa.

5 – Comparando Ferramentas

Este capítulo tem o propósito de realizar algumas comparações entre as ferramentas *EncryptOnClick*, *Kruptos 2 Professional* e *Gold Lock 3G* com as funcionalidades oferecidas pelo SACIS – *Windows PC*.

5.1 – *EncryptOnClick*

É uma ferramenta gratuita para criptografar arquivos locais utilizando criptografia simétrica AES 256. Ao acessar o programa, ele apresenta quatro botões. Dois botões são referentes a criptografia de todos os arquivos numa determinada pasta ou arquivo soltos e os outros dois são referentes a descriptografia dos mesmos como segue figura 39.



Figura 39 – Tela Inicial *EncryptOnClick*

Para criptografar os arquivos, basta clicar no botão referente a sua ação, escolher o arquivo ou pasta e colocar uma senha de qualquer tamanho (Figura 40). O programa irá realizar a criptografia renomeando o nome do arquivo isolado ou contidos na pasta para extensão “.EOC” e apaga o arquivo original. A descriptografia segue de forma semelhante a criptografia sendo o arquivo com sua extensão original restaurada e o arquivo cifrado apagado.

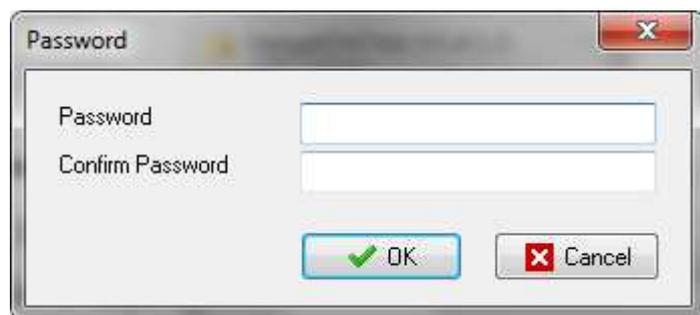


Figura 40 – Solicitação de Senha

Como ponto positivo, a ferramenta é bem mais simples e intuitiva em relação ao SACIS – *Windows PC*. Porém, ela não parece oferecer segurança adequada aos arquivos cifrados por permitir senhas com tamanho de 1 caractere. Ao contrário, o sistema proposto por este trabalho só aceita senhas de 32 caracteres.

Outro problema detectado na ferramenta analisada foi o livre acesso que qualquer pessoa tem a ela. Supondo que uma pessoa com más intenções acesse o dispositivo, este pode cifrar todos os arquivos indiscriminadamente impossibilitando o usuário de visualizar o conteúdo de seus arquivos. Nesse ponto, o SACIS – *Windows PC*, através de sua tela de *login*, só permite o acesso aos usuários cadastrados no sistema diminuindo o risco desse tipo de situação ocorrer.

5.2 – *Kruptos 2 Professional*

É uma ferramenta paga para criptografar arquivos locais, simples que utiliza criptografia simétrica *blowfish 256* na versão inglesa⁵⁷. Para criptografar arquivos, basta selecioná-lo através do botão “*add files*” ou “*add folder*” ou arrastá-lo a partir da aba lateral. Os arquivos escolhidos irão aparecer na tela principal (Figura 41). Após isso, é só clicar no botão “*Encrypt*” onde será solicitada uma senha de no mínimo 6 caracteres e opcionalmente uma pasta destino (Figura 42). Todos os arquivos listados na tela principal serão cifrados com ela.

⁵⁷ Nota: Para este experimento foi utilizada a versão *Trial* de 30 dias.

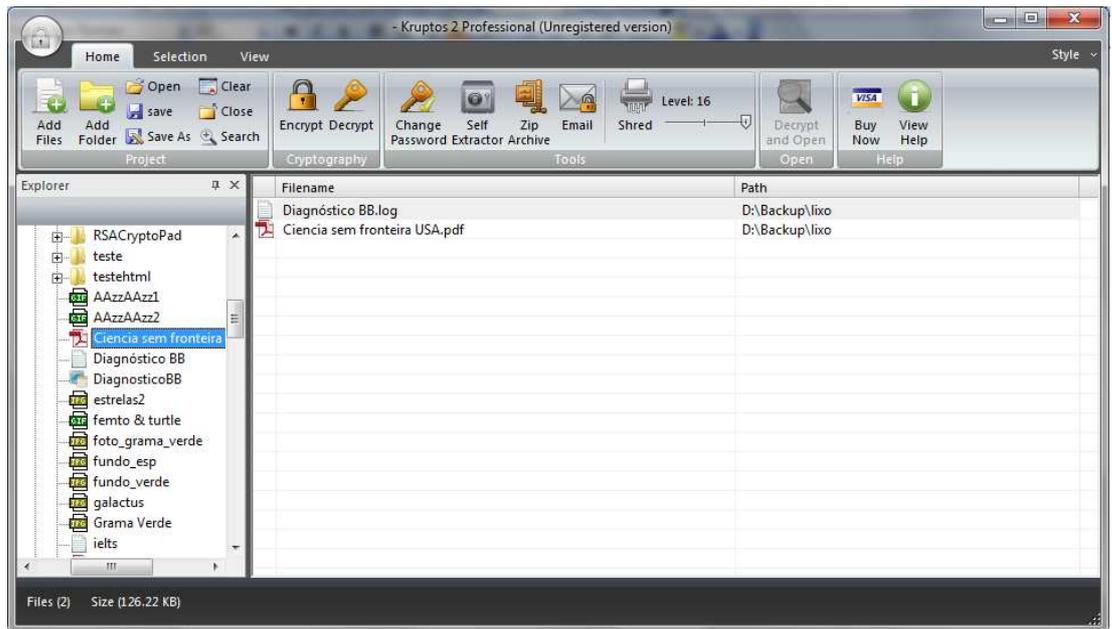


Figura 41 – Tela Principal *Kruptos*



Figura 42 – Solicitação de Senha

Ao realizar a criptografia, a ferramenta cria um arquivo com o conteúdo criptado e extensão “.k2p” e apaga o arquivo original. Para executar a descriptografia dos arquivos, que segue de forma semelhante à criptografia, o usuário deve clicar no botão “*decrypt*” e informar a senha do(s) arquivo(s) para ter seu arquivo restaurado ao estado original. Caso o(s) arquivo(s) tenham senhas diferentes, o programa não irá descriptografá-los, sendo necessário repetir o processo informando as outras senhas.

A *Kruptos 2 Professional* é uma ferramenta cujo funcionamento é muito semelhante ao SACIS – *Windows PC*, porém ela oferece algumas funcionalidades a mais como:

- A associação de arquivos à ferramenta para rápida visualização de seu conteúdo em texto claro, sem converter o arquivo ao estado original, quando clicado duplamente, bastando informar a senha;
- A possibilidade de alterar a senha de um arquivo já cifrado (desde que saiba a senha antiga);
- A compactação de arquivos; e
- A possibilidade de “picotar” um arquivo, ou seja, excluir o arquivo sobrecrevendo ele e não somente apagando o ponteiro que o referencia na memória.

Como ponto negativo, ela permite uma senha com tamanho mínimo de 6 caracteres, o acesso a ferramenta por qualquer pessoa que esteja manuseando o computador, possibilitando que este possa cifrar os arquivos indiscriminadamente como ocorre com a ferramenta anterior, *EncryptOnClick*, e o botão “*Email*” que apenas cifra os arquivos listados, compacta-os e abre o sistema de envio de mensagens padrão do sistema operacional (ex: *Outlook Express*, *Microsoft Outlook*) anexando o arquivo cifrado à mensagem. O SACIS – *Windows PC* solicita uma senha de 32 caracteres ao criptografar os arquivos locais, tem um maior controle sobre o acesso ao sistema de criptografia local, através de seu *login* para evitar problemas conforme explicado na seção 5.1 neste mesmo capítulo, e possui um sistema próprio de envio de mensagens que permite enviar informações e documentos cifrados por ele mesmo, não dependendo de outras ferramentas para este fim.

5.3 – Gold Lock 3G

O *Gold Lock 3G* é uma ferramenta paga⁵⁸ para envio de mensagens e arquivos e comunicação por voz com segurança que utiliza algoritmos criptográficos *Diffie Hellman* 4096 bits, AES 256 bits e curva elíptica 384 bits. Para utilizar a ferra-

⁵⁸ Nota: Para este experimento foi utilizada a versão *Trial* de 15 dias

menta é necessário fazer primeiramente um cadastro no site da empresa. De posse de suas credenciais, o usuário consegue acessar o sistema através da tela de *login* (Figura 43). Ao entrar no sistema pela primeira vez, é solicitado um nome para associar o dispositivo que está sendo usado com o *login*. Uma vez dentro do sistema, a tela principal apresenta a aba “contato” e a barra de menu (Figura 44).



Figura 43 – Tela login



Figura 44 – Tela principal

Para adicionar os contatos, é necessário saber seu nome, pois a ferramenta não disponibiliza a opção de busca e não permite a exclusão do mesmo. Com o con-

tato online, as opções para comunicação com ele são disponibilizadas. O usuário ao selecionar as opções “Text” ou “Send File”, abre-se uma aba para realizar a comunicação com o contato (Figura 45).

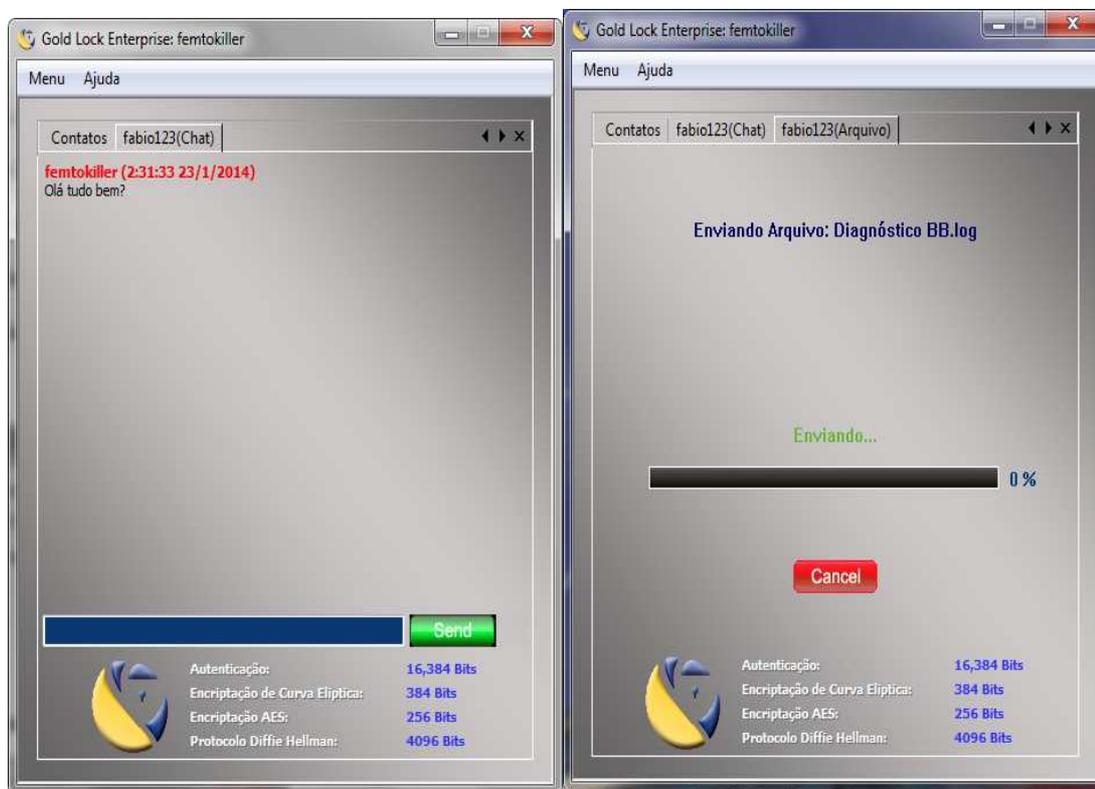


Figura 45 – Abas Para Envio de Mensagens e Arquivos

Apesar de ser uma ferramenta simples, ela só permite o acesso a uma única conta por dispositivo devido a associação que ela faz com o dispositivo. Também não há nenhum indicativo de quando a criptografia está sendo utilizada, pois, ao que tudo indica, ela é feita de forma automática e pré-definida para a comunicação sem permitir ao usuário escolher a desejada.

Tanto a *Gold Lock* quanto o *SACIS – Windows PC*, para enviar mensagens, necessitam estar conectados com a internet. A grande diferença entre ambos é que a primeira funciona de modo similar a um *chat* para enviar as mensagens e arquivos enquanto a segunda realiza esta mesma função de forma parecida com um e-mail⁵⁹ e permite qualquer pessoa cadastrada acessar o sistema de um mesmo dispositivo.

⁵⁹ Nota: Este trabalho não utiliza o protocolo SMTP o qual caracteriza um e-mail.

6 – Considerações Finais e Projetando o Futuro

De acordo com um dos objetivos propostos para este trabalho, o desenvolvimento do *web service* cumpre sua função por disponibilizar os serviços a serem consumidos pelo cliente. Assim, ele realiza a comunicação com os sistemas de gerenciamento de mensagens, administração de usuários e armazenamento local e salva as informações vindas deles. Além disso, ele possibilita a interoperabilidade e a conversa entre os diferentes sistemas, devido ao uso do padrão XML nas mensagens, e isola os serviços disponibilizados dos dados não oferecendo riscos à segurança destes.

Com relação ao envio de mensagens, nenhuma das ferramentas apresentou um sistema similar ao proposto neste projeto. Algumas apenas anexam um arquivo cifrado no gerenciador de e-mail padrão ou então utilizam o chat para enviar mensagens. É possível que o *Symantec Email Encryption* seja parecido, porém não foi possível analisá-la por ela ser paga e a empresa não fornecer uma versão de teste. Com o sistema de gerenciamento de mensagens consegue-se enviar os dados cifrados, através da combinação da criptografia simétrica e assimétrica, e armazená-los com segurança garantindo não somente a confidencialidade das informações como também sua integridade, disponibilidade e autenticidade. Nesse ponto, o gerenciamento de chaves desenvolvido para o projeto facilita o sistema a utilizar as chaves informadas pelo usuário e diminui a necessidade de conhecimento técnico por parte do usuário. Assim, as interfaces gráficas desenvolvidas tornam-se mais simples.

Para uma maior segurança das mensagens, todas as tarefas de cifração e decifração devem ser realizadas no lado do cliente, isentando o servidor dessa tarefa. Para ser realizada no lado do servidor, exigiria que a ferramenta garantisse a segurança dos dados nele contra ataques de qualquer natureza, uma vez que o servidor por si só é um alvo em potencial. Dessa forma, embora a ferramenta aumente o uso da banda ao enviar a mensagem para cada destinatário, fornece uma maior segurança aos dados enviados e armazenados no servidor.

A comparação de algumas ferramentas com as funcionalidades propostas neste trabalho mostra que o SACIS – *Windows PC* possui um melhor controle das pessoas que irão utilizá-lo por possuir um sistema gerenciador de usuários. Das ferramentas analisadas, apenas uma, *Gold Lock 3G*, possui um cadastro de usuários *online*

para envio de mensagens e arquivos. As demais, que realizam a criptografia de arquivos locais, não possuem nenhuma restrição de acesso às suas funcionalidades, permitindo qualquer pessoa utilizá-las. Entende-se que uma ferramenta de segurança não deve ser utilizada por agentes não autorizados a fim de não se tornar uma opção para prejudicar terceiros. Nesse sentido, o sistema de armazenamento de arquivos consegue evitar que pessoas não autorizadas utilizem o sistema.

Para este trabalho tornar-se uma plataforma para o desenvolvimento de novas tecnologias, esta versão do SACIS - *Windows PC* necessita de alguns ajustes e melhorias como:

- Criação da funcionalidade ‘Responder a todos’;
- Implementação de uma função para remoção de usuários excluídos da base de dados nos arquivos de logs existentes nos dispositivos locais;
- Implementação do botão enviar e receber mensagens para evitar que o sistema faça uma requisição ao servidor a cada clique na pasta de enviados e de entrada;
- Realização de testes para detecção de erros, de desempenho, de usabilidade e para detectar falhas na segurança da ferramenta;
- Implementação de um método para salvar anexos; e
- Criação de método para marcar mensagens como lidas ou não lidas;

Para uma nova versão seria interessante disponibilizar funcionalidades como:

- A inserção de algoritmos criptográficos proprietários;
- Visualização do conteúdo de arquivos com extensão ‘.sac’ através da descryptografia automática;
- Inclusão de métodos para tornar a ferramenta acessível através do teclado.
- A criação de uma pasta ou diretório virtual;
- Desenvolvimento de um túnel seguro de comunicação entre o lado cliente e servidor;
- A inserção de um gerenciamento de sessão para controle dos acessos ao sistema;

- A comunicação através do *HyperText Transfer Protocol Secure* (HTTPS)⁶⁰;
- Envio de mensagem SMS pelo servidor *web*;
- A criação de uma área segura para os arquivos decifrados das mensagens não serem acessados por agentes; e
- A utilização do protocolo PKCS [Muzzi, F.A.G. & Tamae, R.Y., 2004] para leitura de outros formatos de chaves assimétricas e para permitir a interoperação com outras ferramentas de comunicação.

Não foi possível comparar com todas as ferramentas pesquisadas por atenderem a outros sistemas operacionais e dispositivos ou por serem ferramentas não gratuitas. Porém, elas poderão ser abordadas em outro momento com o desenvolvimento de novas tecnologias para o ambiente *Windows* ou conforme o avanço dos estudos que estão sendo realizados voltados para a expansão do projeto como o seu desenvolvimento para o ambiente *Android*⁶¹, a utilização da biometria para criação de chaves criptográficas e a criptografia de voz para a comunicação em tempo real. Estes ampliarão o espectro da tecnologia gerada por este projeto, potencializando a sua utilização em diversas aplicações e áreas.

⁶⁰ HTTPS – É a combinação do protocolo HTTP com o protocolo SSL/TLS.

⁶¹ *Android* – Sistema operacional desenvolvido pela Google para *smartphones*.

Anexo I – Regras de Negócio

- O sistema deve armazenar os dados dos usuários numa única base de dados no servidor *web*;
- O usuário cadastrado deve possuir um único cadastro e identificação;
- Para acessar o sistema, todo usuário deve estar cadastrado;
- Para acessar o sistema, o usuário deve entrar com *login* e senha válidos;
- Somente o administrador pode cadastrar, alterar ou excluir usuários;
- Todo usuário deve informar sua chave pública certificada obtida através de qualquer certificadora no formato PEM e sem conter senha para ser cadastrado;
- Para usar o sistema de armazenamento local o usuário deve fazer ao menos um acesso ao sistema de gerenciamento de mensagens;
- O envio de mensagens só pode ser feito para outro usuário cadastrado no sistema;
- Os usuários devem possuir pastas individuais para armazenamento das mensagens recebidas e enviadas no servidor *web*;
- Os usuários devem ter a opção de enviar suas mensagens cifradas ou não.
- Deve haver um catálogo geral contendo todos os usuários do sistema;
- Deve haver um catálogo pessoal com todos os contatos adicionados pelo usuário no servidor *web*;
- O usuário deve informar a localização de sua chave privada quando solicitado pelo sistema;
- Os arquivos cifrados pelo sistema terão a extensão ‘SAC’;
- Todas as mensagens enviadas e recebidas devem ter as informações de cabeçalho salvas na base de dados;
- Os arquivos de contatos deverão ter extensão ‘CNT’;
- Todas as mensagens deverão ser identificadas por um código numérico e extensão ‘MSG’;
- O *hash* utilizado deve ser o SHA-512;
- O algoritmo utilizado para a cifra simétrica deve ser o *Rijndael*;
- O vetor de inicialização deve conter 32 caracteres;
- O algoritmo utilizado para a cifra assimétrica deve ser o RSA.

Anexo II – Requisitos Funcionais

- Cadastrar usuário no sistema;
- Excluir usuário no sistema;
- Alterar usuário no sistema;
- Selecionar chave pública;
- Selecionar chave privada;
- Fazer *login*;
- Selecionar aplicação a ser utilizada;
- Selecionar arquivos a serem criptografados;
- Informar pasta para armazenar arquivos criptografados;
- Informar frase-senha para criptografia de arquivos locais;
- Selecionar arquivos a serem descriptografados;
- Informar pasta para armazenar arquivos descriptografados;
- Informar frase-senha para descriptografia de arquivos locais;
- Alterar senha;
- Enviar mensagens cifradas e/ou assinadas ou claras;
- Encaminhar mensagens recebidas cifradas e/ou assinadas ou claras;
- Responder mensagens recebidas cifradas e/ou assinadas ou claras;
- Abrir mensagens recebidas cifradas ou claras;
- Excluir mensagens recebidas cifradas ou claras;
- Cancelar mensagens;
- Consultar usuários no catálogo pessoal;
- Remover usuários no catálogo pessoal;
- Consultar usuários do catálogo geral.
- Adicionar contatos do catálogo geral para o pessoal;
- Selecionar destinatários da mensagem;
- Selecionar arquivos a serem anexados na mensagem;
- Escolher arquivos anexados na mensagem a serem criptografados.

Anexo III – Requisitos Não-Funcionais

- A senha do usuário é gerada e convertida em *hash* no ato do cadastro ou alteração de senha;
- O sistema verifica a validade dos dados informados pelo usuário;
- O sistema cria pastas individuais para cada usuário cadastrado;
- Uma nova senha é solicitada ao usuário ao fazer *login* pela primeira vez ou ao solicitar sua mudança;
- O sistema verifica a validade dos contatos informados pelo usuário ao enviar a mensagem;
- O sistema adiciona ao registro local o *login* e o *hash* da senha dos usuários que acessaram o Sistema de Gerenciamento de Mensagem;
- O sistema verifica a existência do usuário e senha no registro local;
- A tela inicial do usuário apresenta as opções para acessar o Sistema de Armazenamento Local e o Sistema de Gerenciamento de Mensagens;
- A tela de acesso aos sistemas solicita o *login* e senha do usuário cadastrado;
- O sistema salva automaticamente na base de dados as informações das mensagens enviadas e recebidas;
- O sistema salva as mensagens enviadas e recebidas nas respectivas pastas do remetente e destinatários;
- O sistema converte a mensagem e anexos criptografados em hexadecimal e vice-versa;
- O sistema verifica a data de expiração da chave pública dos usuários do sistema;
- O sistema avisa ao usuário do sistema sobre a proximidade da expiração de sua chave pública.

Anexo IV – Processo de Negócio

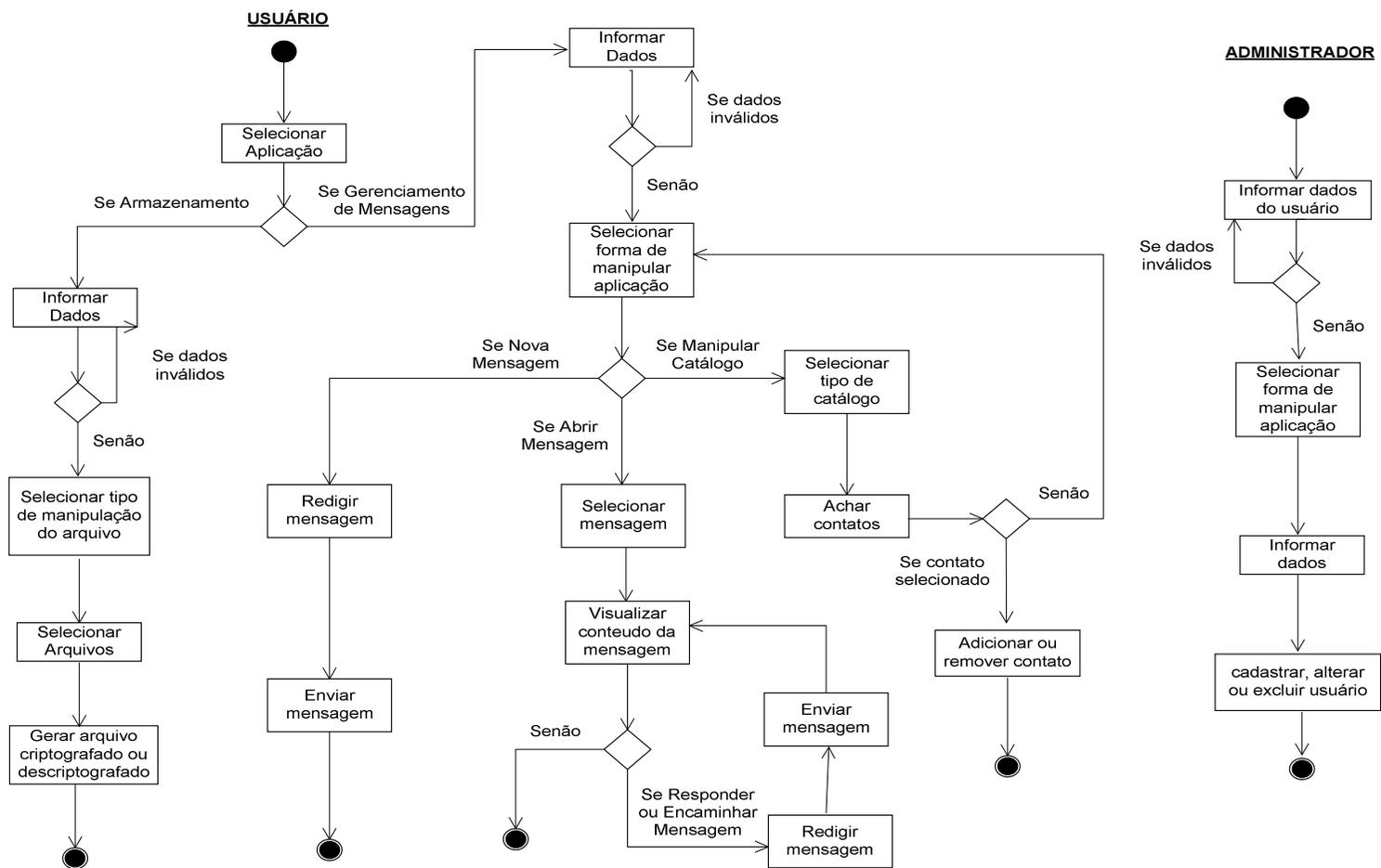


Figura 46 – Processo de Negócio

Anexo V – Caso de Uso: Cadastro de Usuário

Nome: Cadastramento de usuário

Objetivo: Cadastrar dados do usuário que terá acesso ao sistema

Ator: Administrador

Pré-condição: O administrador possuir os dados do usuário a ser cadastrado (Nome, *Login*, Certificado e Permissão).

Pós-condição: Usuário cadastrado no sistema.

O caso de uso começa quando o sistema apresenta a aba Cadastrar Usuário no Sistema de Manutenção.

- Fluxo Normal

1. O administrador entra com os dados (Nome, *Login*, Certificado e Permissão) do usuário a ser cadastrado.
2. O sistema valida os dados digitados e selecionados.
3. O sistema verifica se não existe cadastrado o *Login* do usuário digitado.
4. O sistema gera automaticamente a senha e o *Hash* dela.
5. O sistema salva os dados do usuário na base de dados.
6. O sistema cria pastas para armazenamento de mensagens para o usuário cadastrado
7. Fim do caso de uso.

- Fluxo Alternativo 2: Sistema invalida Certificado.

- 2.1. O sistema apresenta mensagem “Certificado com senha ou inexistente”.
- 2.2. Retorna ao passo 1.

- Fluxo Alternativo 2: Tipo de permissão não selecionada

- 2.1. O sistema apresenta mensagem “Selecione um tipo de permissão”.
- 2.2. Retorna ao passo 1.

- Fluxo Alternativo 2: Sistema invalida Nome ou *Login*.

- 2.1. O sistema apresenta mensagem “Erro nos dados”.
- 2.2. Retorna ao passo 1.

- Fluxo Alternativo 3: Usuário já cadastrado.

- 3.1. O sistema apresenta mensagem “Usuário Cadastrado”.
- 3.2. Retorna ao passo 1.

Anexo VI – Caso de Uso: Alteração de Usuário

Nome: Alteração dos dados do usuário

Objetivo: Alterar dados solicitados pelo usuário

Ator: Administrador

Pré-condição: O administrador saber os dados a serem alterados (Nome, Senha, Certificado e Permissão) e o *Login* do usuário.

Pós-condição: Dados do usuário alterados no sistema.

O caso de uso começa quando o administrador seleciona a aba Alterar Usuário no Sistema de Manutenção.

- Fluxo Normal

1. O administrador digita o *Login* do usuário
2. O administrador seleciona a opção resetar senha
3. O administrador escolhe a opção OK
4. O sistema verifica se pelo menos uma opção foi selecionada.
5. O sistema verifica a existência do *login* digitado.
6. O sistema altera automaticamente o(s) dado(s) na base de dados.
7. O sistema apresenta mensagem “Dados alterados com sucesso”.
8. Fim do caso de uso.

- Fluxo Alternativo 2: Administrador seleciona opção Alterar Nome

- 2.1. O administrador digita novo nome

- Fluxo Alternativo 2: Administrador seleciona opção Alterar Certificado

- 2.1. O administrador digita caminho do novo certificado

- Fluxo Alternativo 2.1: Administrador busca certificado

2.1.1. O sistema apresenta tela para escolha do certificado com as opções Abrir e Cancelar

2.1.2. O administrador seleciona arquivo do certificado

2.1.3. O administrador seleciona opção abrir

2.1.4. O sistema mostra caminho selecionado

- Fluxo Alternativo 2.1.1: Opção Cancelar escolhida

2.1.1.1. Retorna ao passo 1

- **Fluxo Alternativo 2:** Administrador seleciona opção Alterar Permissão
 - 2.1. O administrador seleciona opção (administrador ou usuário)
- **Fluxo Alternativo 3:** Opção sair selecionada.
 - 3.1. O sistema apresenta mensagem “Deseja realmente sair do sistema de manutenção?”.
 - 3.2. O administrador seleciona a opção OK
 - 3.3. Vai para o passo 8.
- **Fluxo Alternativo 3.2:** Opção Cancelar selecionada.
 - 3.2.1. Vai para o passo 1.
- **Fluxo Alternativo 4:** Nenhuma opção selecionada.
 - 4.1. O sistema apresenta mensagem “Selecione pelo menos uma opção”.
 - 4.2. Retorna ao passo 1.
- **Fluxo Alternativo 4:** Opção alterar nome selecionado
 - 4.1. O sistema verifica validade do nome
 - 4.2. O sistema valida nome
 - 4.3. Vai para passo 5
- **Fluxo Alternativo 4.2:** Sistema invalida nome
 - 4.2.1. O sistema apresenta mensagem “Nome Inválido”.
 - 4.2.2. Retorna ao passo 1.
- **Fluxo Alternativo 4:** Opção alterar Certificado selecionado
 - 4.1. O sistema verifica validade do certificado
 - 4.2. O sistema valida certificado
 - 4.3. Vai para passo 5
- **Fluxo Alternativo 4.2:** Sistema invalida certificado
 - 4.2.1. O sistema apresenta mensagem “Certificado inexistente ou inválido”.
 - 4.2.2. Retorna ao passo 1.
- **Fluxo Alternativo 5:** Usuário inexistente.
 - 5.1. O sistema apresenta mensagem “Login Inválido”.
 - 5.2. Retorna ao passo 1.

Anexo VII – Caso de Uso: Exclusão de Usuário

Nome: Exclusão de usuário

Objetivo: Excluir usuário do sistema

Ator: Administrador

Pré-condição: O administrador saber o *Login* do usuário.

Pós-condição: Usuário excluído do sistema.

O caso de uso começa quando o administrador seleciona a aba Excluir Usuário no Sistema de Manutenção.

- Fluxo Normal

1. O administrador digita o *Login* do usuário
2. O administrador escolhe a opção OK
3. O sistema verifica a existência do *Login* digitado.
4. O sistema exclui automaticamente os dados na base de dados, as pastas e arquivos existentes no servidor.
5. Fim do caso de uso.

Anexo VIII – Caso de Uso: Criptografia de Arquivos

Nome: Criptografia de arquivos

Objetivo: Cifrar arquivos selecionados localmente

Ator: Usuário

Pré-condição: O usuário ter acesso ao aplicativo Sistema de Armazenamento

Pós-condição: Arquivos cifrados localmente

O caso de uso começa quando o usuário seleciona a aba Criptografar no Sistema de Armazenamento.

- Fluxo Normal

1. O sistema apresenta a tela inicial com as opções Ok, Cancelar e Selecionar
2. O usuário escolhe a opção Selecionar
3. O sistema apresenta tela para escolha de arquivos com as opções Abrir e Cancelar
4. O usuário escolhe arquivo(s) a ser(em) manipulado(s)
5. O usuário escolhe a opção Abrir
6. O sistema exibe arquivo(s) escolhido(s) com a opção de excluí-lo na tela inicial
7. O usuário escolhe a opção OK na tela inicial
8. O sistema solicita uma frase senha de 32 caracteres
9. O sistema valida a frase senha
10. O sistema exibe tela para escolha da pasta destino com as opções Ok, Cancelar e Criar Nova Pasta
11. O usuário escolhe a opção Ok
12. O sistema executa o processo de criptografia
13. O sistema exibe a mensagem “Criptografia realizada com sucesso!”
14. O sistema salva o arquivo cifrado na pasta destino
15. Fim do caso de uso

- Fluxo Alternativo 1: Usuário escolhe opção Ok sem nenhum arquivo selecionado

- 1.1. O sistema exibe a mensagem “Selecione pelo menos um arquivo!”
- 1.2. Vai para o passo 1.

- **Fluxo Alternativo 1:** Usuário escolhe opção Ok com arquivo(s) selecionado(s)
 - 1.1. Vai para o passo 8.
- **Fluxo Alternativo 1:** Usuário escolhe a opção Cancelar
 - 1.1. O sistema exibe a mensagem “Deseja realmente cancelar o armazenamento?” com as opções Ok e Cancelar
 - 1.2. O usuário escolhe a opção OK
 - 1.3. Fim do caso de uso
- **Fluxo Alternativo 1.2:** O usuário escolhe a opção Cancelar
 - 1.2.1. Vai para o passo 1
- **Fluxo Alternativo 5:** Usuário escolhe Cancelar
 - 5.1. Vai para o passo 1
- **Fluxo Alternativo 6:** Usuário escolhe opção excluir arquivo
 - 6.1. O sistema exclui o arquivo da lista.
 - 6.2. Vai pro passo 6
- **Fluxo Alternativo 9:** Sistema invalida a frase senha
 - 9.1. O sistema exibe a mensagem “Digite uma senha valida com 32 caracteres”
 - 9.2. Vai pro passo 6
- **Fluxo Alternativo 10:** Opção Cancelar escolhida
 - 10.1. Vai pro passo 6
- **Fluxo Alternativo 10:** Opção Criar Nova Pasta escolhida
 - 10.1. O sistema cria nova pasta localmente

Anexo IX – Caso de Uso: Descriptografia de Arquivos

Nome: Decifração de arquivos

Objetivo: Decifrar arquivos selecionados localmente

Ator: Usuário

Pré-condição: O usuário ter acesso ao aplicativo Sistema de Armazenamento

Pós-condição: Arquivos decifrados localmente

O caso de uso começa quando o usuário seleciona a aba Descriptografar no Sistema de Armazenamento.

- Fluxo Normal

1. O sistema apresenta a tela inicial com as opções Ok, Cancelar e Selecionar
2. O usuário escolhe a opção Selecionar
3. O sistema apresenta tela para escolha de arquivos com as opções Abrir e Cancelar
4. O usuário escolhe arquivo(s) a ser(em) manipulado(s)
5. O usuário escolhe a opção Abrir
6. O sistema exibe arquivo(s) escolhido(s) com a opção de excluí-lo na tela inicial
7. O usuário escolhe a opção OK na tela inicial
8. O sistema solicita uma frase senha de 32 caracteres
9. O sistema valida a frase senha
10. O sistema exibe tela para escolha da pasta destino com as opções Ok, Cancelar e Criar Nova Pasta
11. O usuário escolhe a opção Ok
12. O sistema verifica a validade da frase senha
13. O sistema executa o processo de descriptografia
14. O sistema exibe a mensagem “Descriptografia realizada com sucesso!”
15. O sistema salva o arquivo decifrado na pasta destino
16. Fim do caso de uso

- Fluxo Alternativo 1: Usuário escolhe opção Ok sem nenhum arquivo selecionado

- 1.1. O sistema exibe a mensagem “Selecione pelo menos um arquivo!”
- 1.2. Vai para o passo 1.

- **Fluxo Alternativo 1:** Usuário escolhe opção Ok com arquivo(s) selecionado(s)
 - 1.1. Vai para o passo 8.
- **Fluxo Alternativo 1:** Usuário escolhe a opção Cancelar
 - 1.1. O sistema exibe a mensagem “Deseja realmente cancelar o armazenamento?” com as opções Ok e Cancelar
 - 1.2. O usuário escolhe a opção OK
 - 1.3. Fim do caso de uso
- **Fluxo Alternativo 1.2:** O usuário escolhe a opção Cancelar
 - 1.2.1. Vai para o passo 1
- **Fluxo Alternativo 5:** Usuário escolhe Cancelar
 - 5.1. Vai para o passo 1
- **Fluxo Alternativo 6:** Usuário escolhe opção excluir arquivo
 - 6.1. O sistema exclui o arquivo da lista.
 - 6.2. Vai pro passo 6
- **Fluxo Alternativo 9:** Sistema invalida a frase senha
 - 9.1. O sistema exibe a mensagem “Digite uma senha valida com 32 caracteres”
 - 9.2. Vai pro passo 6
- **Fluxo Alternativo 10:** Opção Cancelar escolhida
 - 10.1. Vai pro passo 6
- **Fluxo Alternativo 10:** Opção Criar Nova Pasta escolhida
 - 10.1. O sistema cria nova pasta localmente
- **Fluxo Alternativo 12:** Chave inválida
 - 12.1. O sistema exibe a mensagem “Chave inválida!”
 - 12.2. Vai para o passo 1

Anexo X – Caso de Uso: Envio de Mensagem

Nome: Envio de mensagem

Objetivo: Enviar mensagens cifradas ou claras com ou sem arquivos anexados cifrados ou claros.

Ator: Usuário.

Pré-condição: O usuário possuir o destinatário adicionado no catálogo local.

Pós-condição: Mensagem enviada.

O caso de uso começa quando o usuário escolhe opção Novo na tela de Gerenciamento de Mensagens.

- Fluxo Normal

1. O sistema apresenta tela com as opções Para, Anexar, Criptografar Mensagem, Assinar Mensagem, Enviar e Descartar e os campos para o destinatário da mensagem, o assunto e mensagem.
2. O usuário digita o endereço eletrônico do destinatário da mensagem.
3. O usuário digita assunto da mensagem.
4. O usuário seleciona a opção Anexar.
5. O sistema abre tela principal de seleção de Anexos com as opções de Ok, Cancelar, Selecionar, Remover e Criptografar.
6. O usuário seleciona a opção Selecionar.
7. O sistema abre tela para seleção de arquivos.
8. O usuário seleciona os arquivos.
9. O sistema lista os arquivos escolhidos na tela principal de seleção de Anexos.
10. O usuário seleciona arquivos listados para cifrar e nenhum para remover.
11. O usuário seleciona a opção Ok.
12. O sistema mostra os arquivos a anexar no campo destinado.
13. O usuário digita o corpo da mensagem.
14. O usuário seleciona a opção Assinar Mensagem.
15. O usuário seleciona a opção Criptografar Mensagem.
16. O usuário seleciona a opção Enviar.
17. O sistema valida contatos.
18. O sistema exibe tela solicitando chave privada do usuário.
19. O sistema executa processo de assinatura da mensagem.

20. O sistema executa processo de criptografia simétrica nos arquivos selecionados e na mensagem.
21. O sistema executa processo de criptografia assimétrica das chaves simétricas.
22. O sistema cria a mensagem com os anexos.
23. O sistema envia a mensagem para o servidor *web*
24. O sistema salva alguns dados da mensagem na base de dados.
25. O sistema envia mensagem ao destinatário.
26. Fim do Caso de Uso.

- Fluxo Alternativo 1: Opção Para escolhida

- 1.1. O sistema apresenta uma tela listando os contatos do catalogo local do usuário com as opções Ok e Cancelar.
- 1.2. O usuário seleciona o(s) contato(s) desejado(s)
- 1.3. O usuário escolhe a opção Ok
- 1.4. O sistema exibe os contatos selecionados no campo destinatário
- 1.5. Vai para o passo 3

- Fluxo Alternativo 1.2: Opção Cancelar escolhida

- 1.2.1. Retorna ao passo 1

- Fluxo Alternativo 4: Usuário não seleciona opção Anexar

- 4.1. O usuário digita o corpo da mensagem.
- 4.2. O usuário seleciona a opção Assinar Mensagem.
- 4.3. O usuário seleciona a opção Criptografar Mensagem.
- 4.4. O usuário seleciona a opção Enviar.
- 4.5. O sistema verifica a validade dos contatos.
- 4.6. O sistema exibe tela solicitando chave privada do usuário.
- 4.7. O sistema executa processo de assinatura da mensagem.
- 4.8. O sistema executa processo de criptografia simétrica na mensagem.
- 4.9. O sistema executa processo de criptografia assimétrica das chaves simétricas
- 4.10. O sistema cria a mensagem
- 4.11. Vai para o passo 23.

- Fluxo Alternativo 4.2: Usuário não seleciona a opção Assinar Mensagem

4.2.1. O usuário seleciona a opção Criptografar

4.2.2. O usuário seleciona a opção Enviar

4.2.3. O sistema valida contatos.

4.2.4. Vai para o passo 4.8.

- Fluxo Alternativo 4.2.1: Usuário não seleciona a opção Criptografar

4.2.1.1. O usuário seleciona a opção Enviar.

4.2.1.2. O sistema valida contatos.

4.2.1.3. Vai para o passo 4.10.

- Fluxo Alternativo 4.2.1.2: Contatos Inválidos

4.2.1.2.1 O sistema exibe mensagem “Verifique Contato Inválido!”.

4.2.1.2.2 O sistema retorna ao passo 1

- Fluxo Alternativo 4.2.3: Contatos Inválidos

4.2.3.1. O sistema exibe mensagem “Verifique Contato Inválido!”.

4.2.3.2. O sistema retorna ao passo 1

- Fluxo Alternativo 4.3: Usuário não seleciona a opção Criptografar

4.3.1. O usuário seleciona a opção Enviar

4.3.2. O sistema valida contatos.

4.3.3. O sistema exibe tela solicitando chave privada do usuário.

4.3.4. O sistema executa processo de assinatura da mensagem

4.3.5. Vai para o passo 4.10.

- Fluxo Alternativo 4.3.2: Contatos Inválidos

4.3.1. O sistema exibe mensagem “Verifique Contato Inválido!”.

4.3.2. O sistema retorna ao passo 1

- Fluxo Alternativo 4.5: Contatos Inválidos

4.5.1. O sistema exibe mensagem “Verifique Contato Inválido!”.

4.5.2. O sistema retorna ao passo 1

- **Fluxo Alternativo 9:** Opção Cancelar escolhida
 - 9.1. Retorna ao passo 1.
- **Fluxo Alternativo 9:** Opção remover selecionada
 - 9.1. O usuário escolhe a opção Ok.
 - 9.2. O sistema remove da lista o(s) arquivo(s) selecionado(s).
 - 9.3. Vai para o passo 12.
- **Fluxo Alternativo 14:** Usuário não seleciona a opção Assinar Mensagem
 - 14.1. O usuário seleciona a opção Criptografar.
 - 14.2. O usuário seleciona a opção Enviar.
 - 14.3. O sistema valida contatos.
 - 14.4. Vai para o passo 20.
- **Fluxo Alternativo 14.1:** Usuário não seleciona a opção Criptografar
 - 14.1.1. O usuário seleciona a opção Enviar.
 - 14.1.2. O sistema valida contatos.
 - 14.1.3. Vai para o passo 22.
- **Fluxo Alternativo 14.1.2:** Contatos Inválidos
 - 14.1.2.1. O sistema exibe mensagem “Verifique Contato Inválido!”.
 - 14.1.2.2. O sistema retorna ao passo 1.
- **Fluxo Alternativo 14.3:** Contatos Inválidos
 - 14.3.1. O sistema exibe mensagem “Verifique Contato Inválido!”.
 - 14.3.2. O sistema retorna ao passo 1.
- **Fluxo Alternativo 15:** Usuário não seleciona a opção Criptografar
 - 15.1. O usuário seleciona a opção Enviar
 - 15.2. O sistema valida contatos.
 - 15.3. O sistema executa processo de assinatura da mensagem
 - 15.4. O sistema executa processo de criptografia simétrica nos arquivos selecionados.
 - 15.5. O sistema executa processo de criptografia assimétrica das chaves simétricas.
 - 15.6. Vai para o passo 22.

- Fluxo Alternativo 15.2: Contatos Inválidos

15.2.1. O sistema exibe mensagem “Verifique Contato Inválido!”.

15.2.2. O sistema retorna ao passo 1

- Fluxo Alternativo 16: Opção Descartar escolhida

16.1. O sistema exibe mensagem “Deseja descartar mensagem?” com as opções Ok e Cancelar.

16.2. O usuário escolhe Ok.

16.3. O sistema descarta a mensagem.

16.4. Fim do Caso de Uso.

- Fluxo Alternativo 16.1: Opção Cancelar escolhida

16.1.1. Volta para o passo 1.

- Fluxo Alternativo 17: Contatos Inválidos

17.1. O sistema exibe mensagem “Verifique Contato Inválido!”.

17.2. O sistema retorna ao passo 1

Anexo XI – Caso de Uso: Visualizar Mensagem

Nome: Visualizar mensagens

Objetivo: Visualizar mensagens recebidas/enviadas

Ator: Usuário

Pré-condição: Existir mensagens recebidas/enviadas

Pós-condição: Mensagem visualizada

O caso de uso começa quando o usuário escolhe a opção de Gerenciamento de Mensagens.

- Fluxo Normal

1. O usuário seleciona a pasta das mensagens recebidas ou enviadas
2. O sistema apresenta as mensagens existentes da pasta selecionada na tela.
3. O usuário escolhe uma mensagem.
4. O sistema recupera os dados da mensagem no servidor.
5. O sistema apresenta numa tela o conteúdo da mensagem, o nome dos arquivos anexados, o assunto e o remetente.
6. Fim do caso de uso.

- Fluxo Alternativo 4: Mensagem está criptografada

- 4.1. O sistema exibe tela solicitando chave privada do usuário.
- 4.2. O sistema executa processo de descriptografia da mensagem.
- 4.3. O sistema retorna ao passo 5.

- Fluxo Alternativo 4.2: Mensagem está Assinada

- 4.2.1. O sistema busca chave certificada do usuário no servidor.
- 4.2.2. O sistema executa processo de verificação de assinatura digital.
- 4.2.3. O sistema retorna ao passo 5.

- Fluxo Alternativo 4: Mensagem está assinada

- 4.1. O sistema busca chave certificada do usuário no servidor.
- 4.2. O sistema executa processo de verificação de assinatura digital.
- 4.3. O sistema retorna ao passo 5.

Anexo XII – Caso de Uso: Visualizar Anexos

Nome: Visualizar Anexos

Objetivo: Visualizar anexos das mensagens

Ator: Usuário

Pré-condição: Existir anexos nas mensagens

Pós-condição: Anexo visualizado

O caso de uso começa quando o usuário seleciona um anexo na mensagem.

- Fluxo Normal

1. O sistema recupera anexo selecionado da mensagem.
2. O sistema abre o arquivo anexado selecionado.
3. Fim do caso de uso

- Fluxo Alternativo 1: Arquivo está criptografado

- 1.1. O sistema exibe tela solicitando chave privada do usuário.
- 1.2. O sistema executa processo de descriptografia do anexo.
- 1.3. O sistema retorna ao passo 2.

Anexo XIII – Caso de Uso: Manipulação de Catálogo

Nome: Manipulação de Catálogo

Objetivo: Manipular contatos do catálogo geral e local

Ator: Usuário

Pré-condição: O usuário estar cadastrado no sistema

Pós-condição: Usuários do sistema incluídos ou excluídos do catálogo local.

O caso de uso começa quando o usuário escolhe a opção Catálogo na tela de Gerenciamento de Mensagens.

- Fluxo Normal

1. O sistema exibe as opções Geral e Pessoal
2. O usuário escolhe a opção Geral
3. O sistema lista todos os contatos cadastrados na base de dados e as opções Adicionar e Cancelar
4. O usuário seleciona os contatos desejados
5. O usuário escolhe a opção Adicionar
6. O sistema adiciona os contatos selecionados no catálogo pessoal
7. Fim do caso de uso

- Fluxo Alternativo 1: O usuário escolhe a opção Local

- 1.1. O sistema lista todos os contatos cadastrados no catálogo pessoal e as opções de Remover e Cancelar
- 1.2. O usuário seleciona os contatos desejados
- 1.3. O usuário escolhe a opção Remover
- 1.4. O sistema exclui os contatos selecionados do catálogo pessoal
- 1.5. Fim do caso de uso.

- Fluxo Alternativo 1.1: Opção Cancelar escolhida

- 1.1.1. Fim do caso de uso

- Fluxo Alternativo 3: Opção Cancelar escolhida

- 3.1. Fim do caso de uso

Anexo XIV – Caso de Uso: *Login* do Gerenciamento de Mensagem

Nome: *Login* do Gerenciamento de Mensagem

Objetivo: Permitir acesso ao Sistema de Gerenciamento de Mensagem ao usuário

Ator: Usuário

Pré-condição: O usuário estar cadastrado no sistema

Pós-condição: Sistema permite acesso ao usuário.

O caso de uso começa quando o usuário acessa a tela de *login* do Sistema de Gerenciamento de Mensagens.

- Fluxo Normal

1. Sistema apresenta tela com os campos de *login* e senha.
2. Usuário digita seu *login* e senha.
3. Sistema gera *hash* da senha.
4. Sistema envia *login* e *hash* da senha para o servidor.
5. Servidor valida dados enviados pelo usuário.
6. Usuário acessa tela inicial do Sistema de Gerenciamento de Mensagens.
7. Fim do caso de uso.

- Fluxo Alternativo 5: *Login* de usuário inválido

- 5.1. Sistema exibe mensagem “Digite Usuário Válido!” com a opção Ok.
- 5.2. Volta para o passo 1

- Fluxo Alternativo 5: Senha de usuário inválido

- 5.1. Sistema exibe mensagem “Acesso Negado! *Login* ou Senha não existe!” com a opção Ok.
- 5.2. Volta para o passo 1

- Fluxo Alternativo 5: Senha de usuário expirado

- 5.1. Sistema exibe mensagem "Senha expirada! É necessária sua alteração para acessar novamente o sistema." com a opção Ok.
- 5.2. Sistema exibe tela para troca de senha.
- 5.3. Usuário informa nova senha.
- 5.4. Sistema troca a senha.
- 5.2. Volta para o passo 1

- Fluxo Alternativo 5: Chave certificada expirada

5.1. O sistema exibe mensagem “Acesso Negado! Chave Certificada Expirada!” com a opção Ok.

5.2. Volta para o passo 1

Anexo XV – Caso de Uso: *Login* do Armazenamento de Arquivos

Nome: *Login* do Armazenamento de Arquivos.

Objetivo: Permitir acesso ao Sistema de Armazenamento de Arquivos ao usuário.

Ator: Usuário

Pré-condição: O usuário ter realizado um acesso ao Sistema de Gerenciamento de Mensagem.

Pós-condição: Sistema permite acesso ao usuário.

O caso de uso começa quando o usuário acessa a tela de *login* do Sistema de Gerenciamento de Mensagens.

- Fluxo Normal

1. Sistema apresenta tela com os campos de *login* e senha.
2. Usuário digita seu *login* e senha.
3. Sistema gera *hash* da senha.
4. Sistema valida dados digitados pelo usuário.
5. Usuário acessa tela inicial do Sistema de Armazenamento de Arquivos.
6. Fim do caso de uso.

- Fluxo Alternativo 5: *Login* de usuário inválido

- 5.1. O sistema exibe mensagem “Digite Usuário Válido!” com a opção Ok.
- 5.2. Volta para o passo 1

- Fluxo Alternativo 5: Senha de usuário inválido

- 5.1. O sistema exibe mensagem “Acesso Negado!” com a opção Ok.
- 5.2. Volta para o passo 1

Anexo XVI – Caso de Uso: *Login* de Manutenção de Usuários

Nome: *Login* de Manutenção de Usuários

Objetivo: Permitir acesso ao Sistema de Manutenção de Usuários ao usuário.

Ator: Usuário

Pré-condição: O usuário ter permissão de Administrador.

Pós-condição: Sistema permite acesso ao usuário.

O caso de uso começa quando o usuário acessa a tela de *login* do Sistema de Manutenção de Usuários.

- Fluxo Normal

1. Sistema apresenta tela com os campos de *login* e senha.
2. Usuário digita seu *login* e senha.
3. Sistema gera *hash* da senha.
4. Sistema envia *login* e *hash* da senha para o servidor.
5. Servidor valida dados enviados pelo usuário.
6. Servidor verifica tipo de permissão do usuário.
7. Usuário acessa tela inicial do Sistema de Gerenciamento de Manutenção de Usuários.
8. Fim do caso de uso.

- Fluxo Alternativo 5: *Login* ou senha de usuário inválido.

- 5.1. O sistema exibe mensagem “Acesso Negado!” com a opção Ok.
- 5.2. Volta para o passo 1.

- Fluxo Alternativo 6: Usuário sem permissão de Administrador

- 6.1. O sistema exibe mensagem “Acesso Negado!” com a opção Ok.
- 6.2. Volta para o passo 1.

Anexo XVII – Caso de Uso: Troca de Senha

Nome: Troca de senha

Objetivo: Realizar a troca de senha do usuário no sistema.

Ator: Usuário

Pré-condição: O usuário acessar o Sistema de Gerenciamento de Mensagens pela primeira vez ou ter solicitado alteração de senha.

Pós-condição: Sistema altera senha do usuário.

O caso de uso começa quando o usuário tenta acessar o Sistema de Manutenção de Usuários com a senha expirada.

- Fluxo Normal

1. Sistema apresenta tela de troca de senha com campos para senha nova e confirmação de senha.
2. Usuário digita senha nova e sua confirmação.
3. Sistema valida senha.
4. Sistema gera *hash* da senha.
5. Sistema envia *login* e *hash* da senha para o servidor.
6. Servidor troca senha.
7. Fim do caso de uso.

- Fluxo Alternativo 3: Senha inválida.

- 5.1. O sistema exibe mensagem “Erro ao Trocar Senha! Digite Senha Válida.” com a opção Ok.
- 5.2. Volta para o passo 1.

Anexo XVIII – Diagrama de Estado: Cadastro de Usuário

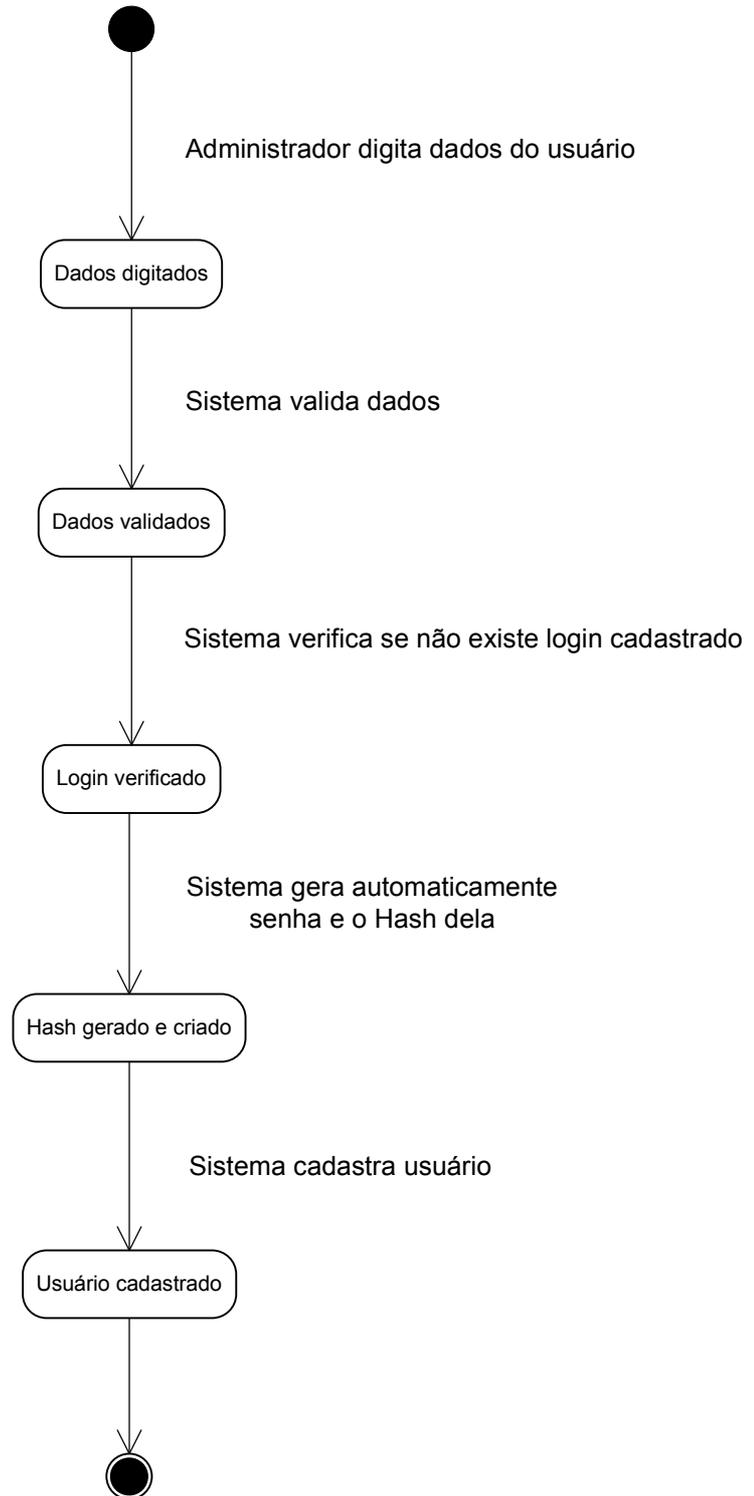


Figura 47 – Diagrama de Estado: Cadastro de Usuário

Anexo XIX – Diagrama de Estado: Alteração de Usuário

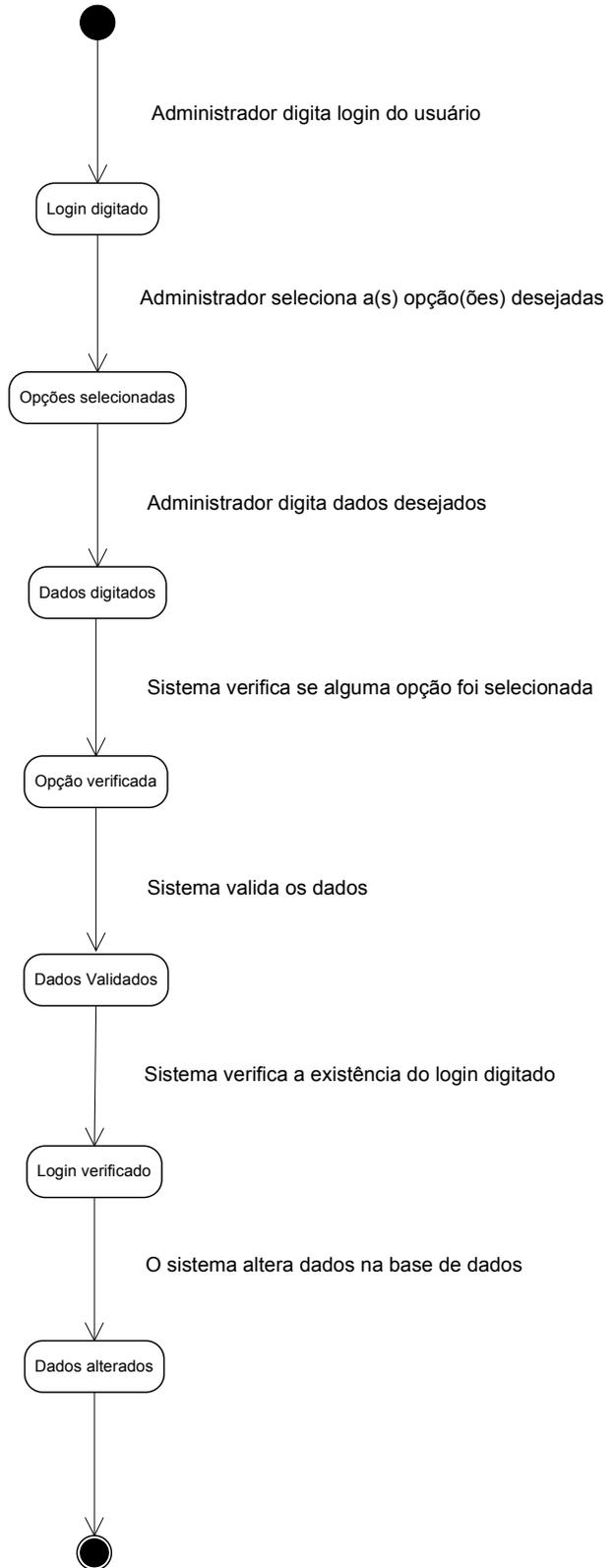


Figura 48 – Diagrama de Estado: Alteração de Usuário

Anexo XX – Diagrama de Estado: Exclusão de Usuário

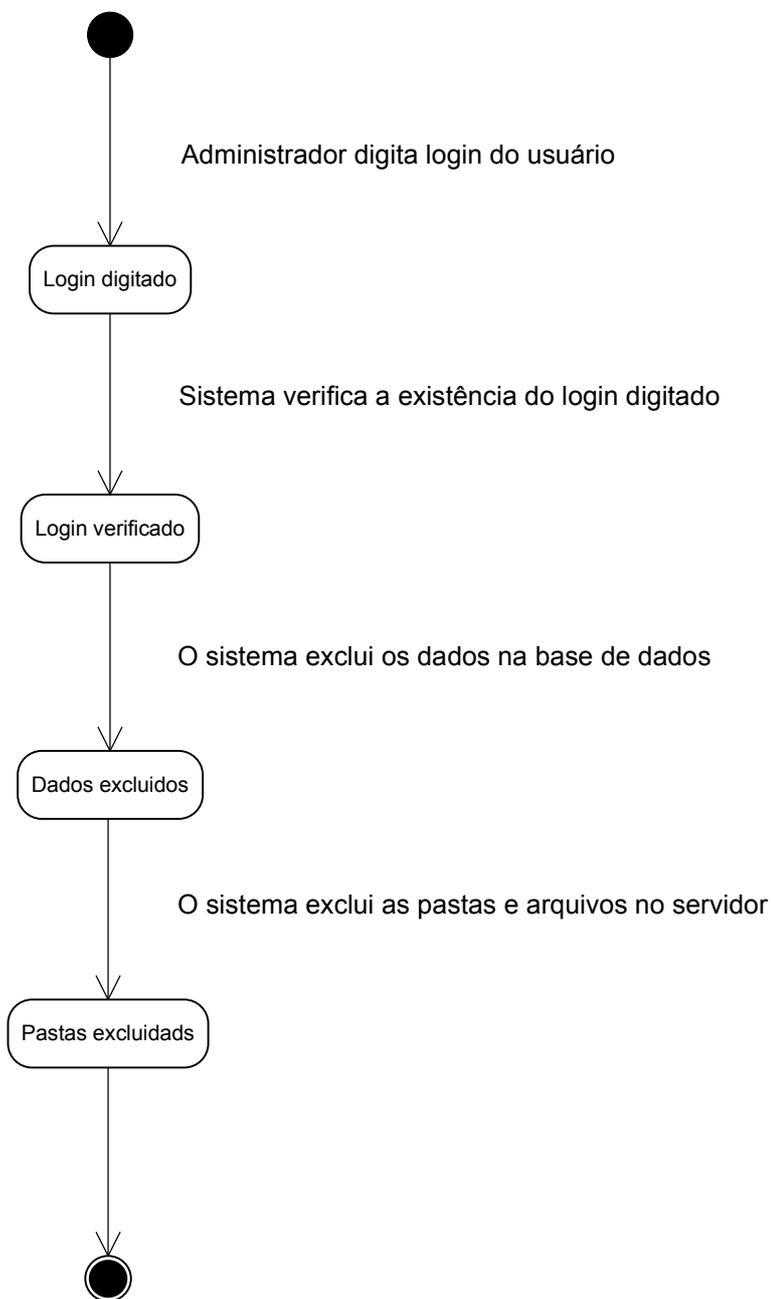


Figura 49 – Diagrama de Estado: Exclusão de Usuário

Anexo XXI – Diagrama de Estado: Criptografia de Arquivos

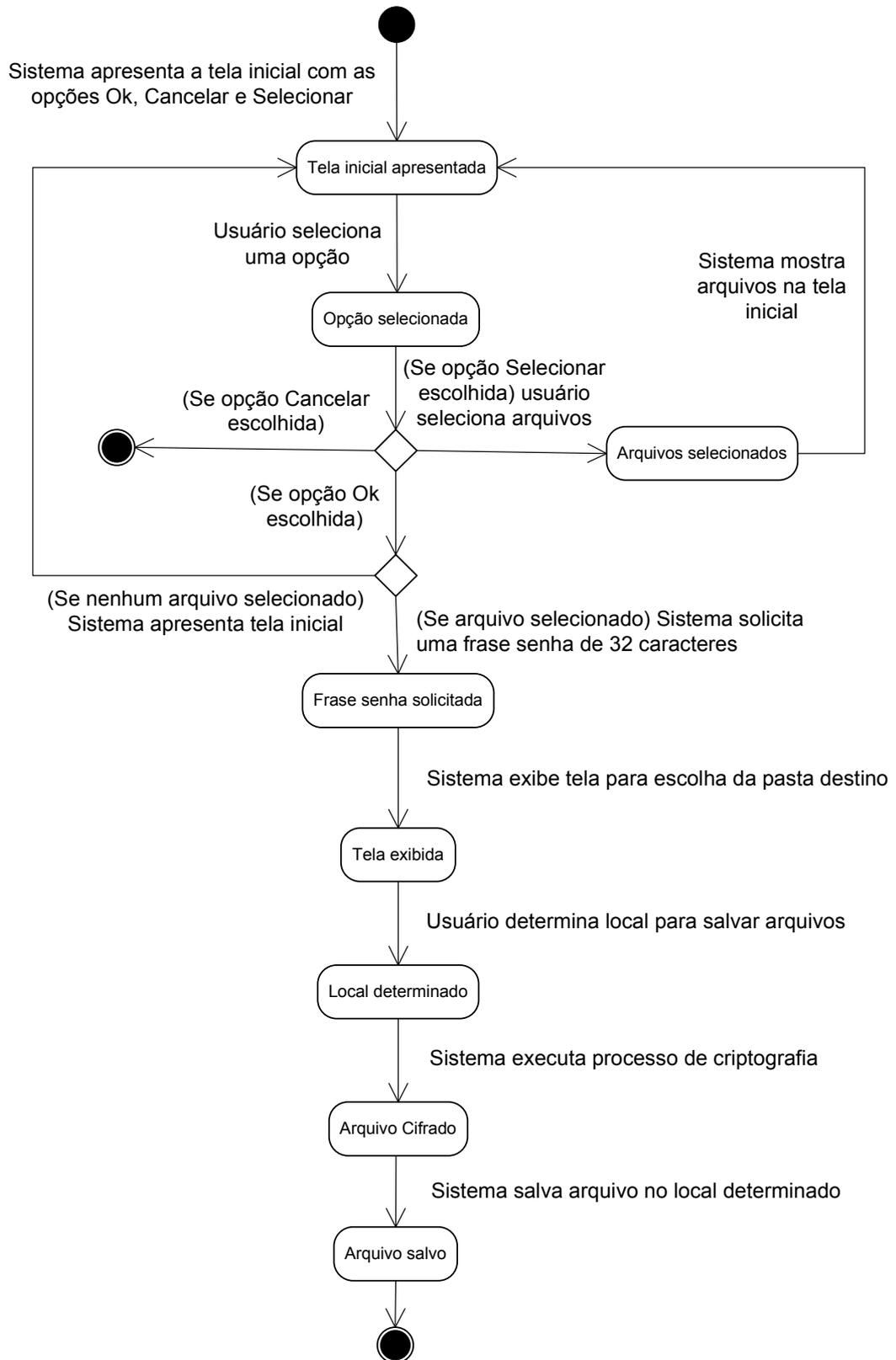


Figura 50 – Diagrama de Estado: Criptografia de Arquivos

Anexo XXII – Diagrama de Estado: Descriptografia de Arquivos

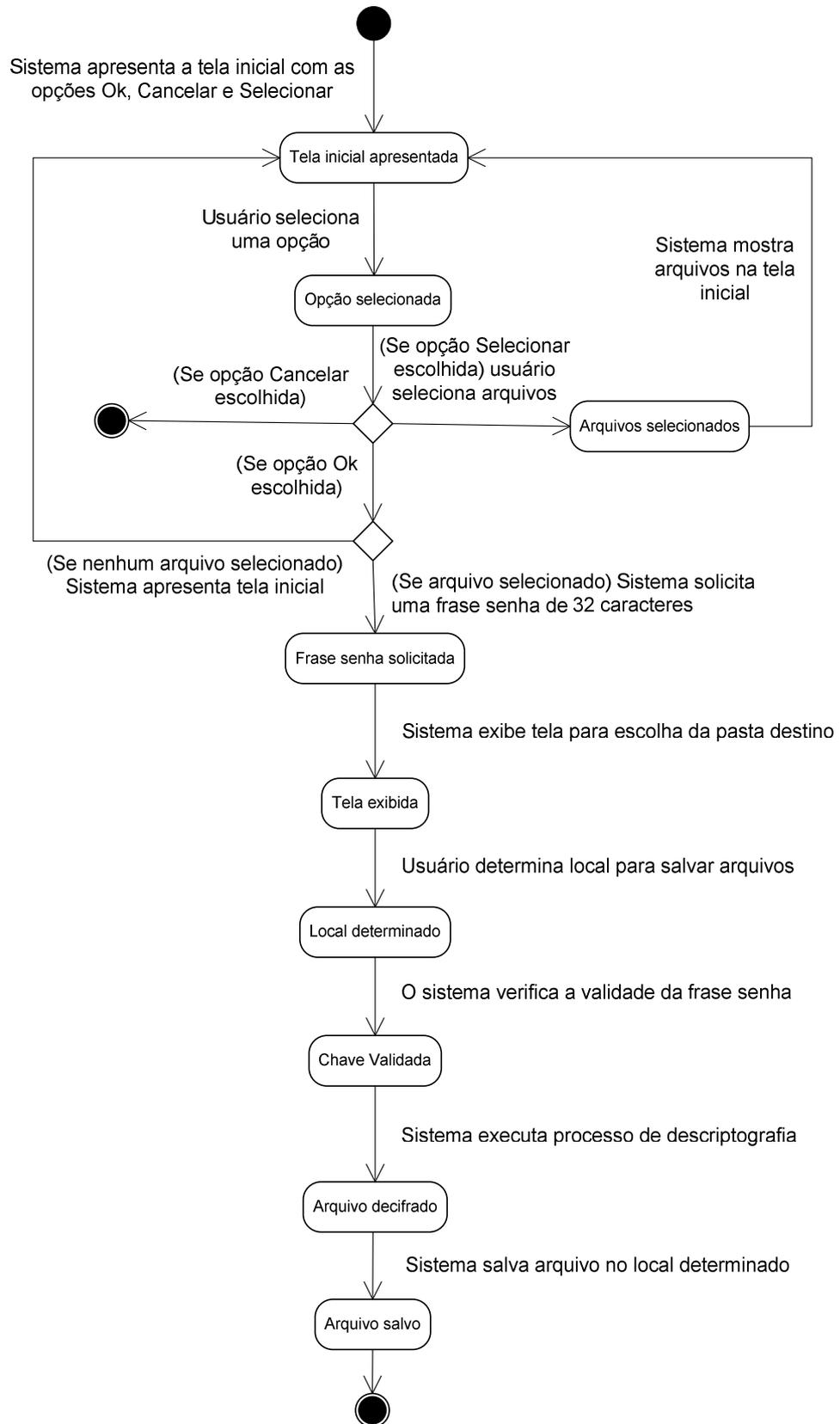


Figura 51 – Diagrama de Estado: Descriptografia de Arquivos

Anexo XXIII – Diagrama de Estado: Envio de Mensagem

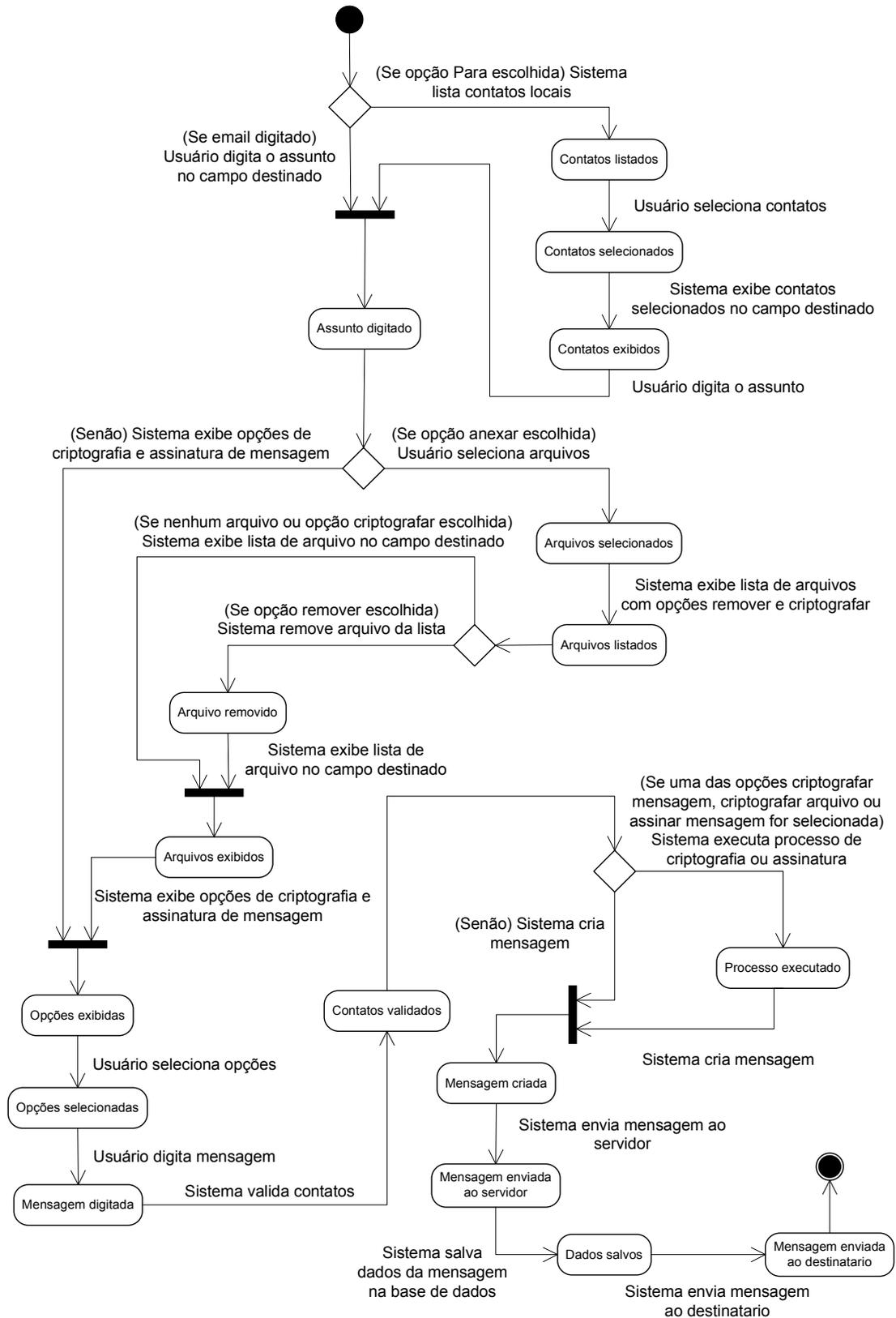


Figura 52 – Diagrama de Estado: Envio de Mensagem

Anexo XXIV – Diagrama de Estado: Visualizar Mensagem

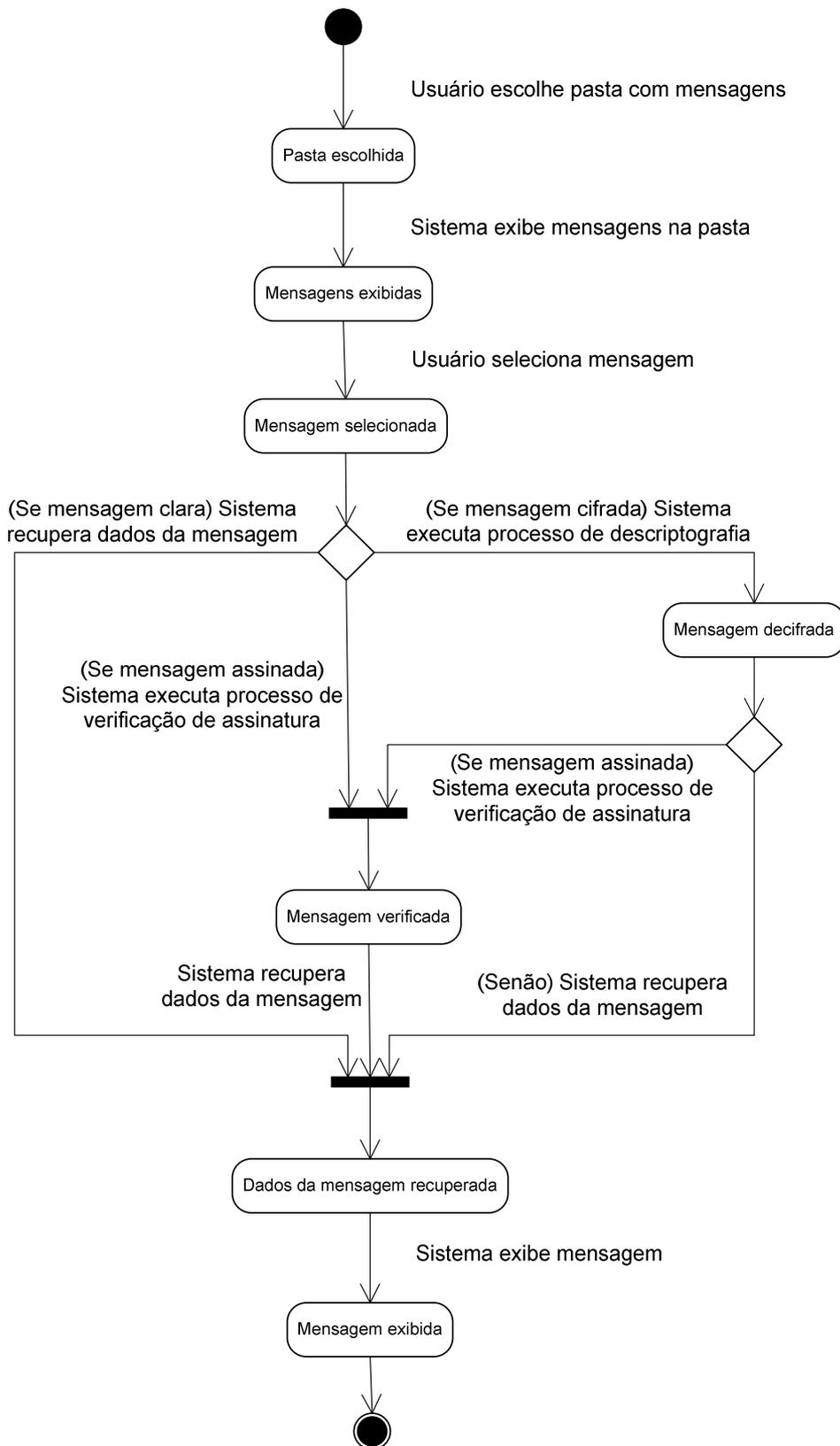


Figura 53 – Diagrama de Estado: Visualizar Mensagem

Anexo XXV – Diagrama de Estado: Visualizar Anexos

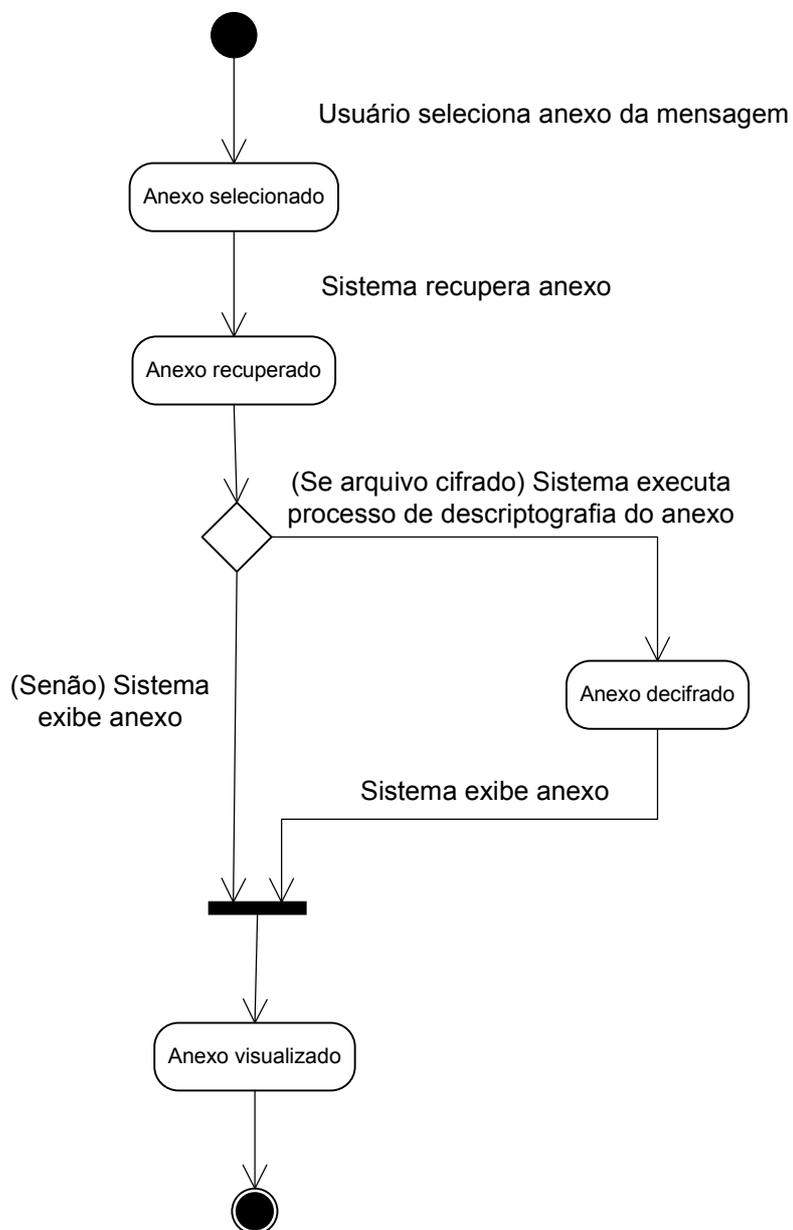


Figura 54 – Diagrama de Estado: Visualizar Anexos

Anexo XXVI – Diagrama de Estado: Manipulação de Catálogo

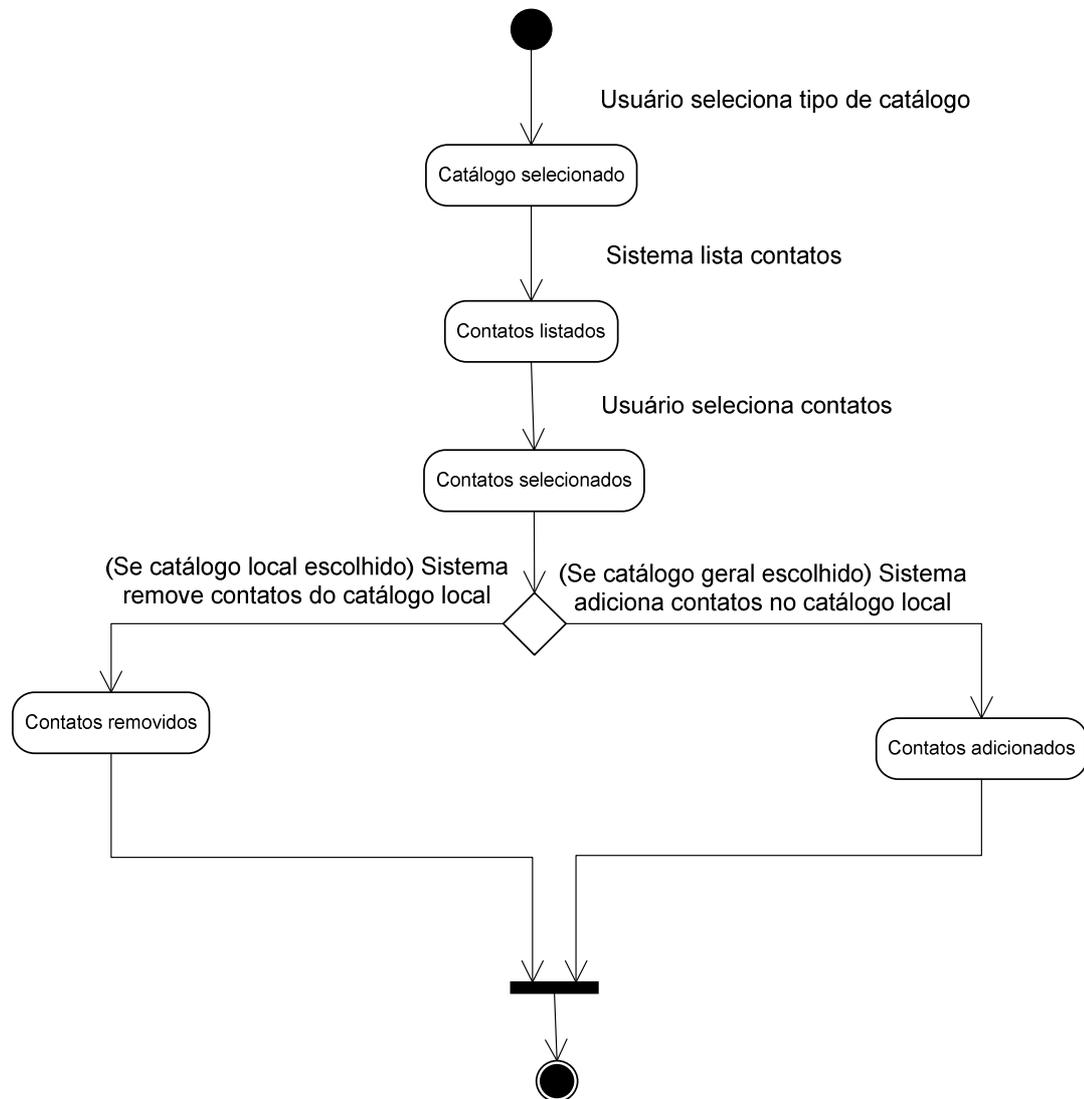


Figura 55 – Diagrama de Estado: Manipulação de Catálogo

Anexo XXVII – Diagrama de Estado: *Login* do Gerenciamento de Mensagem

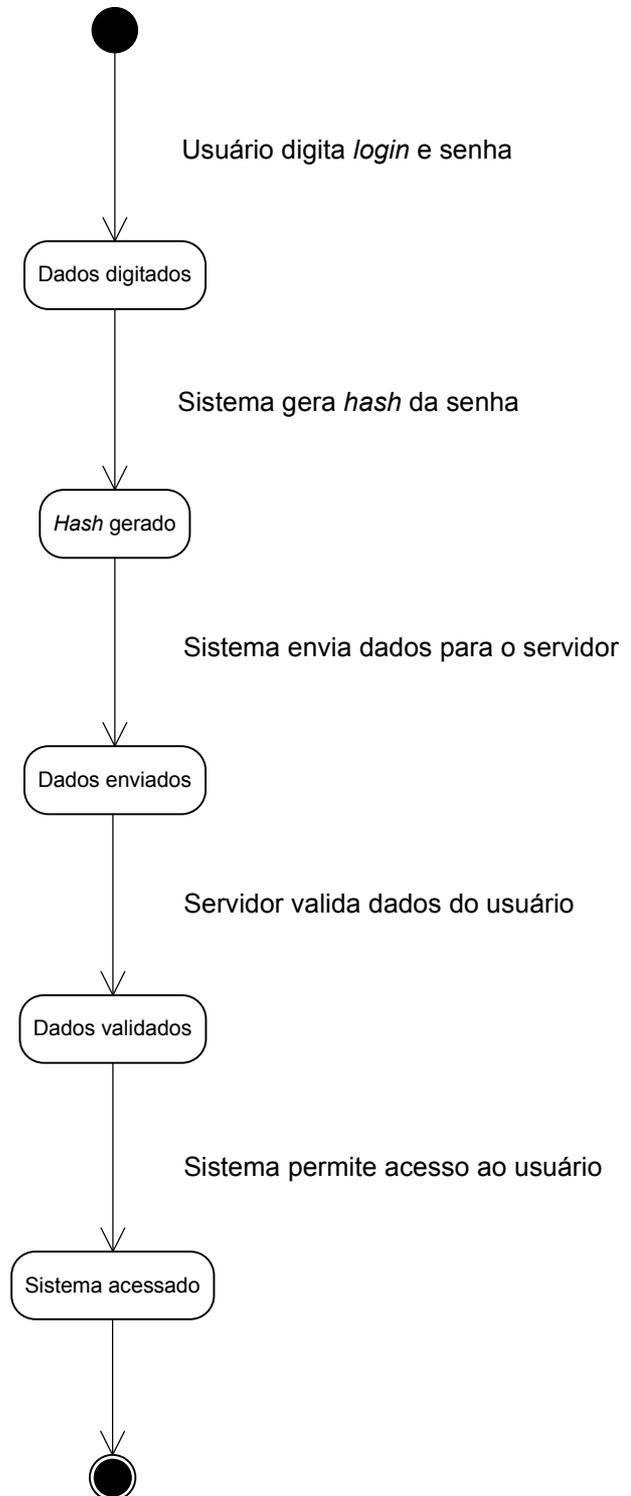


Figura 56 – Diagrama de Estado: *Login* do Gerenciamento de Mensagem

Anexo XXVIII – Diagrama de Estado: *Login* do Armazenamento de Arquivos

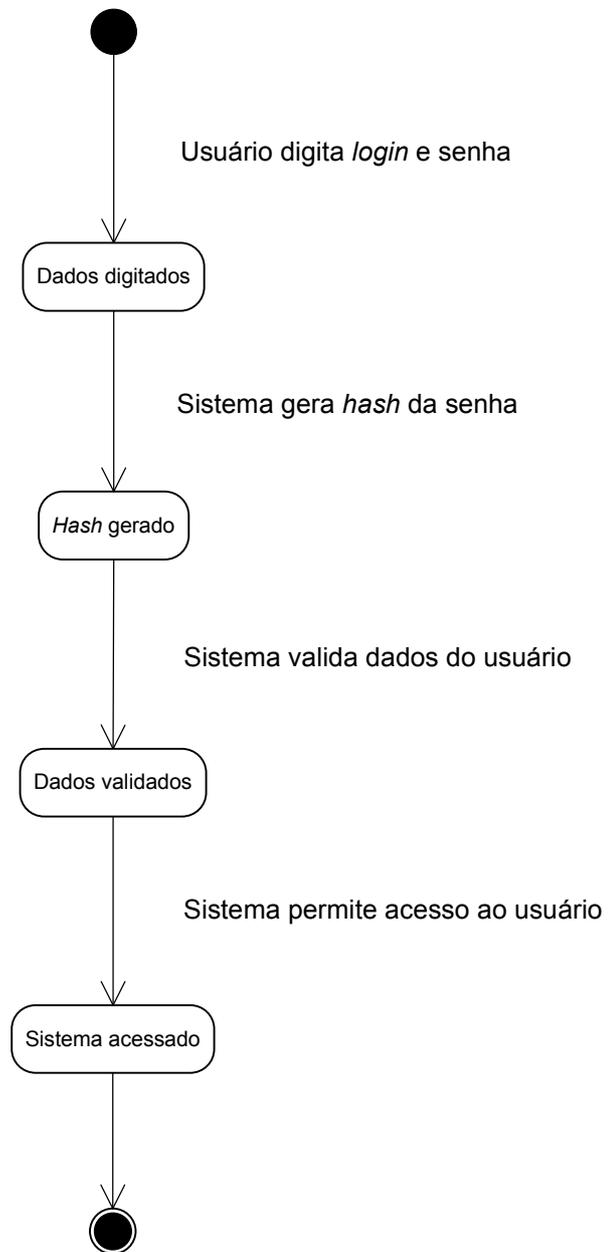


Figura 57 – Diagrama de Estado: *Login* do Armazenamento de Arquivos

Anexo XXIX – Diagrama de Estado: *Login* de Manutenção de Usuários

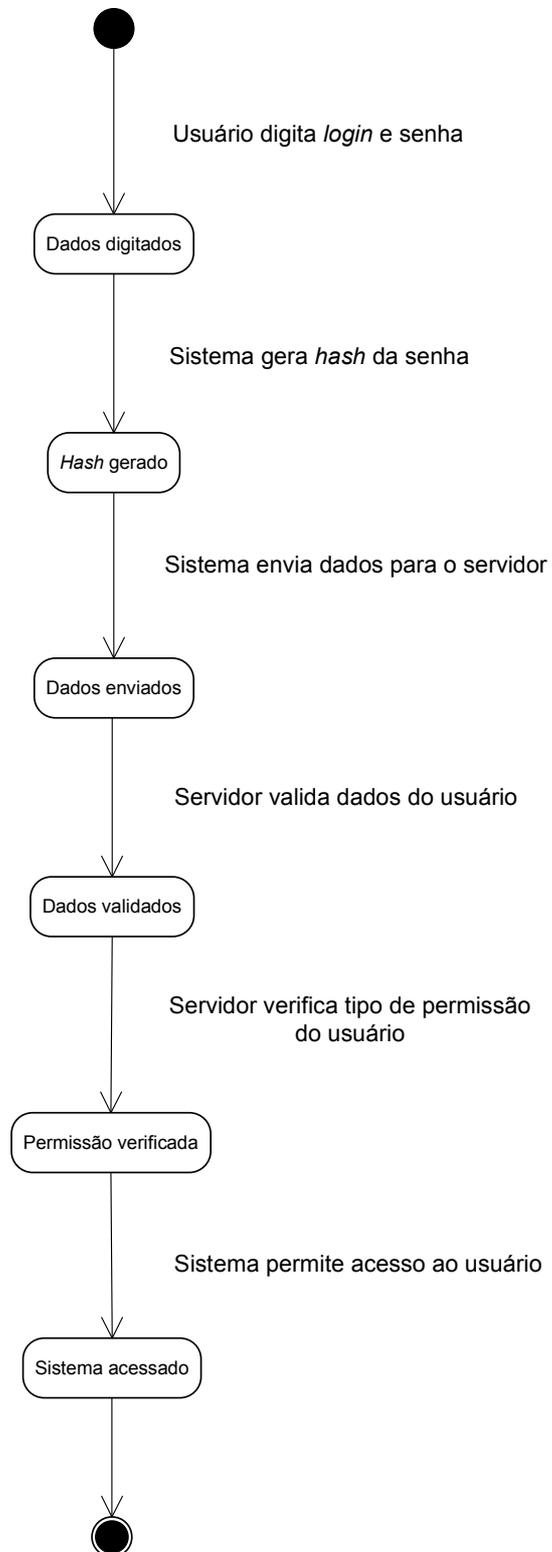


Figura 58 – Diagrama de Estado: *Login* de Manutenção de Usuários

Anexo XXX – Diagrama de Estado: Troca de Senha

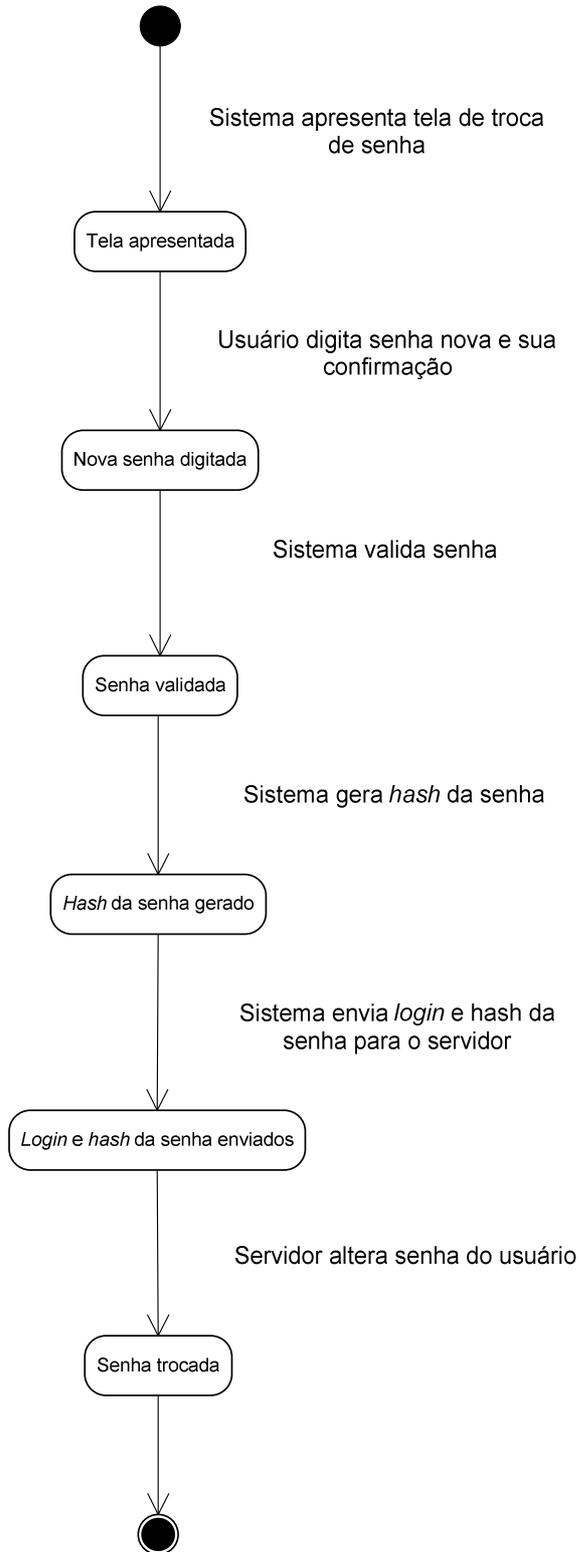


Figura 59 – Diagrama de Estado: Troca de Senha

Anexo XXXI – Diagrama de Classes

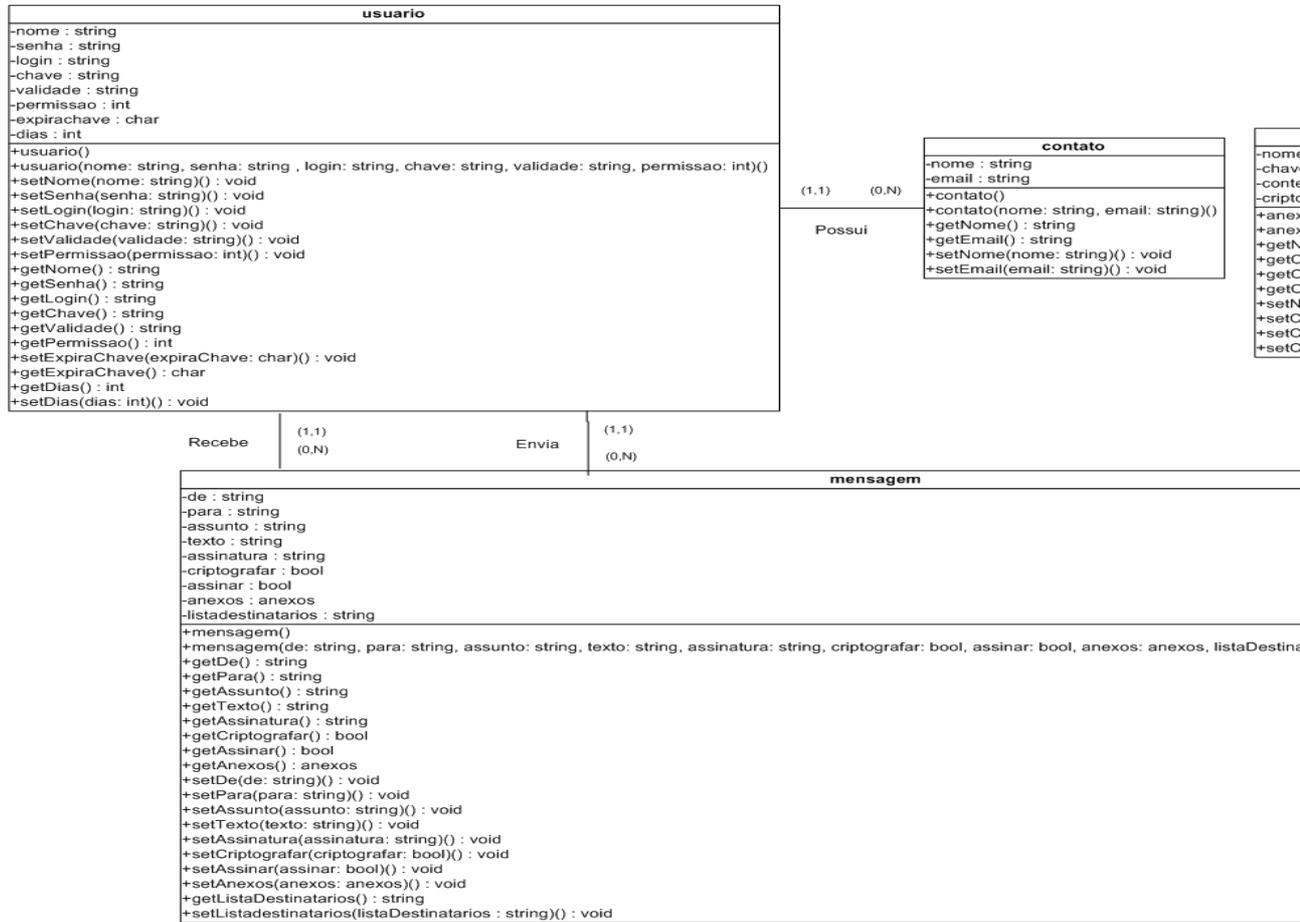


Figura 60 – Diagrama de Classes

Anexo XXXII – Dicionário de Dados

Tabela usuario

usuario = id, login, senha, nome, validade, permissao, alterasenha, expirachave, dias

id = { caracteres-numericos }

login = { caracteres }

senha = { caracteres }

nome = { caracteres }

validade = { caracteres-numericos } + { caracteres-validos }

permissao = { caracteres-alfabeticos }

alterasenha = { caracteres- booleanos }

expirachave = { caracteres- opcionais }

dias = { caracteres-numericos }

caracteres = { caracteres-alfabeticos } + { caracteres-numericos } + { caracteres-validos }

caracteres-alfabeticos = [a – z | A – Z]

caracteres-validos = [. | - | _ | @ | # | \$ | & | : | ;]

caracteres-numericos = [0 – 9]

caracteres-booleanos = [s | n]

caracteres-opcionais = [s | n | t]

Tabela mensagem

mensagem = id, logremdest, assunto, lida, data, tamanho, tipo, loginusuario, listades-
tinatarios, assinada

id = { caracteres-numericos }

logremdest = { caracteres }

assunto = { caracteres }

lida = [0 | 1]

data = { caracteres-numericos }

tamanho = { caracteres-numericos }

tipo = { caracteres-alfabeticos }

loginusuario = { caracteres }

listadestinatarios = { caracteres }

assinada = [0 | 1]

caracteres = { caracteres-alfabeticos } + { caracteres-numericos } + { caracteres-validos }

caracteres-alfabeticos = [a – z | A – Z]

caracteres-validos = [. | - | _ | @ | # | \$ | & | : | ;]

caracteres-numericos = [0 – 9]

Tabela seq_mensagem

seq_mensagem = id

id = { caracteres-numericos }

caracteres-numericos = [0 – 9]

Anexo XXXIII – Script de Dados

```
create table usuario (  
    id int(10) unsigned not null auto_increment,  
    login varchar(45) not null default "",  
    senha varchar(512) not null default "",  
    nome varchar(100) not null default "",  
    validade timestamp not null default '0000-00-00 00:00:00',  
    permissao char(1) not null default 'u',  
    alterasenha char(1) not null default 's',  
    expirachave char(1) not null default 'n',  
    dias int(10) unsigned not null default '0',  
    primary key (id)  
);  
  
insert into usuario(login,senha,nome,validade,permissao,alterasenha,expirachave)  
values ('administrador',  
'7fcf4ba391c48784edde599889d6e3f1e47a27db36ecc050cc92f259bfac38afad2c68a1  
ae804d77075e8fb722503f3eca2b2c1006ee6f6c7b7628cb45fffd1d', 'Administrador',  
'2030-03-18 07:30:00', 'a', 'n', 'n', 0);  
  
create table mensagem (  
    id int(10) unsigned not null auto_increment,  
    loginusuario varchar(45) not null default "",  
    assunto varchar(100) not null default "",  
    lida tinyint(1) not null default '0',  
    logremdest varchar(45) not null default "",  
    tipo varchar(45) not null default "",  
    data timestamp not null default '0000-00-00 00:00:00',  
    tamanho int(10) unsigned not null default '0',  
    listadestinatarios varchar(1000) not null default "",  
    assinada tinyint(1) not null default '0',  
    primary key (id),  
    key cluster (loginusuario) using btree  
);  
  
create table seq_mensagem ( id int not null);  
insert into seq_mensagem (id) values(1);
```

Anexo XXXIV – Procedure para Verificar a Validade das Chaves Certificadas

```
set global event_scheduler = on;

-- drop event verifica_validade_chave;
create event verifica_validade_chave
on schedule every 1 say
do call valida_chave();

delimiter $$

-- drop procedure if exists `bdsacis`.`valida_chave` $$
create procedure `bdsacis`.`valida_chave` ()

begin
    declare usuarioid integer;
    declare done integer default 0;
    declare dias_chave integer;
    declare total_usuarios integer;
    declare expira char(1);

    declare cursorid cursor for select id from usuario;
    declare continue handler for not found set done = 1;
    open cursorid;

    read_loop: loop

        fetch cursorid into usuarioid;

        select datediff(validade, now()), expirachave into dias_chave, expira from usu-
ario where id = usuarioid;

        if done then leave read_loop;
        end if;

        if dias_chave <= 30 and dias_chave > 0 then update usuario set expirachave =
't', dias = dias_chave where id = usuarioid;
        elseif dias_chave <= 0 then update usuario set expirachave = 's', dias = 0 where
id = usuarioid;
        end if;

    end loop read_loop;

    close cursorid;
end $$
delimiter;
```

8 – Referências

- Abdalla, A. et al** “Cryptography: Past, Present, and Future”,
<http://users.ece.gatech.edu/~owen/Academic/CS4235/Summer2006/GR3%20Final%20Project.doc>, Janeiro.
- Azevedo, A. (2010)** “Cifras de Transposição”,
<http://ademirlord.blogspot.com.br/2010/10/cifras-de-transposicao.html>, Janeiro.
- Bastos, D.F. (2011)** “O que é Model-view-controller”,
http://www.oficinadanet.com.br/artigo/desenvolvimento/o_que_e_model-view-controller_mvc, Novembro.
- Bauer, F.L. (1969)** “*Software Engineering*. NATO Science Committee, pagina 231. *Scientific Affairs Division*.
- Burwick, C. et al (1999)** “*MARS - a candidate cipher for AES*”,
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.35.6084&rep=rep1&type=pdf>, Outubro.
- Costa, L.H.M.K. & Duarte, O.C.M.B (2006)** “Curvas Elípticas”,
http://www.gta.ufrj.br/grad/06_2/renan/ecc_renan.html, Outubro.
- Dantas, D.C.T. (2007)** “*Simple Object Access Protocol (SOAP)*”,
http://www.gta.ufrj.br/grad/07_2/daniel/, Novembro.
- Douglas, R. S. (2005)** “*Cryptography Theory and Practice*”, Editora Chapman & Hall/CRC Press, 3ª edição.
- Eastlake, D. & Jones, P. (2001)** “*US Secure Hash Algorithm 1 (SHA1)*”,
<ftp://ftp.rfc-editor.org/in-notes/rfc3174.txt>, Outubro.
- Ferreira, T. (2007)** “Algoritmo Diffie-Hellman”,
http://imasters.com.br/artigo/6954/seguranca/algoritmo_diffie-hellman/, Outubro.
- Grossmann, L.O. (2013)** “Governo vai desenvolver criptografia própria”,
<http://convergenciadigital.uol.com.br/cgi/cgilua.exe/sys/start.htm?inford=33104&sid=18#.UoeZ8sTBPTA>, Novembro.

- Haddad, R. (2013)** “*Web Services*”, <http://msdn.microsoft.com/pt-br/library/cc564893.aspx>, Novembro.
- Hamman, R. (2013)** “Espionagem americana no Brasil: o que é preciso saber?”, <http://www.tecmundo.com.br/politica/41800-espionagem-americana-no-brasil-o-que-e-preciso-saber-.htm>, Outubro.
- Kak, A. (2013)** “*Key Distribution for Symmetric Key Cryptography and Generating RandomNumbers*”, <https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture10.pdf>, Outubro.
- Karasinski, L. (2013)** “PRISM: entenda toda a polêmica sobre como os EUA controlam você”, http://www.tecmundo.com.br/privacidade/40816-prism-entenda-toda-a-polemica-sobre-como-os-eua-controlam-voce.htm?utm_source=artigo_bottom_saibamais&utm_medium=tecmundo, Outubro.
- Kent, S.T. (2006)** “*Internet Privacy Enhanced Mail*”, <http://www.acsac.org/secshelf/book001/17.pdf>, Dezembro.
- Kurose, J. F. & Ross, K. W. (2010)** “Redes de computadores e a Internet: uma abordagem top-down”, São Paulo, Addison Wesley, p. 502 – 504.
- Lobo, A. P. (2013)** “Para deter os EUA, Brasil e Europa negociam regras comuns para computação em nuvem” <http://convergiadigital.uol.com.br/cgi/cgilua.exe/sys/start.htm?infoid=35208&sid=51#.Umm0uvmBPTA>, Outubro.
- Lopez, M.D. (2009)** “*Successful Mobile Deployments Require Robust Security*”, http://us.blackberry.com/business/leading/Successful_Mobile_Deployments.pdf, Outubro.
- Menezes A. & Oorschot P. V. & Vanstone S. (1996)** “*Handbook of Applied Cryptography*”, <http://cacr.uwaterloo.ca/hac/about/chap8.pdf>, Outubro.
- Muzzi, F.A.G. & Tamae, R.Y. (2004)** “Padrão de Criptografia Baseada no PKCS”, http://faef.revista.inf.br/imagens_arquivos/arquivos_destaque/OgMCnWz3AsnO9BS_2013-5-24-11-16-41.pdf, Outubro.

- Pereira, A.P. (2009)** “O que é XML?”,
<http://www.tecmundo.com.br/programacao/1762-o-que-e-xml-.htm>, Outubro.
- Pinto, P. (2013)** “Conheça a Historia da Criptografia”,
<http://pplware.sapo.pt/informacao/conheca-a-historia-da-criptografia/>, Janeiro.
- Pisa, P. (2012)** “O que é Hash?”,
<http://www.techtudo.com.br/artigos/noticia/2012/07/o-que-e-hash.html>, Outubro.
- Rinaldi, D.G. (2012)** “Análise do AES e sua criptoanálise diferencial”,
<http://www.lume.ufrgs.br/bitstream/handle/10183/54139/000855639.pdf?sequence=1>,
, Outubro.
- Rivest, R. (1992)** “*The MD5 Message-Digest Algorithm*”,
<http://tools.ietf.org/html/rfc1321>, Outubro.
- Robshaw, M.J.B. (2001)** “*RC6 and the AES*”,
<ftp://ftp.rsasecurity.com/pub/rsalabs/rc6/rc6%2Baes.pdf>, Outubro.
- Rodrigues, E. (2013)** “GSI lançará algoritmo de proteção de dados federais”,
<http://www.estadao.com.br/noticias/nacional,gsi-lancara-algoritmo-de-protecao-de-dados-federais,1063899,0.htm>, Novembro.
- Rouse, M. (2007)** “*Rijndael*”,
<http://searchsecurity.techtarget.com/definition/Rijndael>, Outubro.
- Rouse, M. (2011)** “*Advanced Encryption Standard (AES)*”,
<http://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard>, Junho.
- Schneier, B. (1996)** “*Applied Cryptography: Protocols, Algorithms, and Source Code in C*”, Editora John Wiley and Sons, 2ª edição.
- Shamir, A. (1984)** “*A Polynomial-Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem*”,
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.123.5840&rep=rep1&type=pdf>, Outubro.

- Silva, E.V.P. (2006)** “Introdução à criptografia RSA”,
http://www.impa.br/opencms/pt/eventos/downloads/jornadas_2006/trabalhos/jornadas_elen_pereira.pdf, Outubro.
- Smid, M.E. & Branstad, D.K. (1988)** “*The Data Encrypt Standard past and future*”, http://media.wiley.com/product_data/excerpt/28/07803535/0780353528.pdf,
Outubro.
- Sommerville, I. (2007)** “Engenharia de Software”, Editora Pearson Addison-Wesley.
- Stamp, M. (2006) “*Information Security: Principles and Practice*”, Editora John Wiley and Sons.
- Trinta, F.A.M. & Macedo, R.C. (1998)** “Um Estudo sobre Criptografia e Assinatura Digital”, <http://www.di.ufpe.br/~flash/ais98/cripto/criptografia.htm>, Junho.
- Viegas, C. (2008)** “Anatomia WSDL”,
<http://www.aqueleblogdesoa.com.br/2008/08/anatomia-do-wsdl/>, Novembro.