UNIVERSIDADE DO EXTREMO DO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

DIEGO MACHADO MEDEIROS

AUDISOFT - FERRAMENTA PARA AUDITORIA EM DESENVOLVIMENTO DE SISTEMAS UTILIZANDO CRIPTOGRAFIA DE DADOS POR MEIO DO ALGORITMO RSA

CRICIÚMA, JULHO DE 2008

DIEGO MACHADO MEDEIROS

AUDISOFT - FERRAMENTA PARA AUDITORIA EM DESENVOLVIMENTO DE SISTEMAS UTILIZANDO CRIPTOGRAFIA DE DADOS POR MEIO DO ALGORITMO RSA

Trabalho de Conclusão de Curso apresentado para a obtenção do Grau de Bacharel em Ciência da Computação da Universidade do Extremo Sul Catarinense.

Orientador: Prof. MSc. Paracelso de Oliveira Caldas

CRICIÚMA, JULHO DE 2008

DIEGO MACHADO MEDEIROS

AUDISOFT - FERRAMENTA PARA AUDITORIA EM DESENVOLVIMENTO DE SISTEMAS UTILIZANDO CRIPTOGRAFIA DE DADOS POR MEIO DO ALGORITMO RSA

Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

_____

Profa. MSc. Ana Claudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação

Banca Examinadora:

_____

Prof. MSc. Paracelso de Oliveira Caldas (UNESC)
Orientador

_____

Profa. MSc. Ana Claudia Garcia Barbosa (UNESC)

_____

Prof. MSc. Alessandro Zanini (UNISUL/UNIBAVE)

Dedico este trabalho a minha família, principalmente aos meus pais e a minha esposa, que sempre estiveram ao meu lado e acreditaram no meu potencial, e a lembrança de meu primo Murilo Medeiros Teixeira.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Ariovaldo e Jucenilda, que sempre fizeram de tudo para me dar uma boa educação.

A minha esposa, Samara Simões Rocha, pela paciência e compreensão tida durante a realização da Graduação na Universidade e principalmente no Trabalho de Conclusão do Curso.

Aos meus amigos e colegas, por terem paciência para me ajudar nas horas complicadas, em especial aos meus amigos e colegas: Lucélio Possamai Valdati e Luiz Juventino.

Ao meu Orientador, Prof. MSc. Paracelso de Oliveira Caldas por acreditar no meu projeto, em minha capacidade e por toda atenção e ajuda prestada durante todo o andamento desta pesquisa.

RESUMO

O presente trabalho apresenta uma pesquisa sobre a Auditoria executada na Engenharia de *Software* (ES) no Desenvolvimento de Sistemas. Para a segurança dos dados coletados e manipulados nesta atividade foi utilizada a Criptografia, como Mecanismo de Segurança. A pesquisa realiza um estudo detalhado da Auditoria em geral, chegando até a área de Auditoria de Sistemas, uma divisão da Auditoria que tem o foco principal desta pesquisa. A Auditoria de Sistemas por sua vez, divide-se em duas grandes partes: a primeira muito utilizada e dizimada é a Auditoria de Sistemas (Sistemas Pronto, em Operação) e a segunda é a Auditoria no Desenvolvimento de Sistemas, esta última é a que vai ser implementada nesta pesquisa. Além disso, este trabalho apresenta um estudo de mecanismos de segurança e aponta a criptografia de dados, junto com o Algoritmo *Rivest - Shamir - Adleman* (RSA), como o meio para se alcançar a proteção e sigilo dos dados da implementação do projeto. Por fim, são apresentadas as etapas de desenvolvimento dessa implementação, mostrando como foi projetado e qual foi à metodologia utilizada para implementação do mesmo.

Palavras-Chave: Auditoria, Auditoria de Sistemas, Desenvolvimento de Sistemas (Engenharia de *Software*), Criptografia de Dados e Algoritmo RSA.

ABSTRACT

The present study presents a research about an audit performed in software engineering in the development of systems. For the security of the data collected and handled in this activity the encryption mechanism was used. The research performed shows a detailed study of audit in general coming to the area of auditing systems, a division of the audit that is the main focus of this research. The audit system is divided into two main parts: the first is widely used and decimated is the audit systems (systems ready, in operation) and the second is the audit of the development of systems, the latter is going to be implemented in this research. Furthermore this work presents a study of security mechanisms and indicates the encryption of data, along with the algorithm Rivest – Shamir – Adleman (RSA), as a means to achieve protection and secrecy of data to implement the project. Finally, we present the stages of development of this implementation, showing how it was projected and which methodology was used for its implementation.

Keywords: Audit, Audit Systems, Development of Systems (Software Engineering), Data Encryption and Algorithm RSA.

LISTA DE ILUSTRAÇÕES

LISTA DE SIGLAS

AAF	*Audit Automation Facilities*
AIS	*Audit Information System*
ACL	*Audit Command Language*
BCSE	*Base Case System Evaluation*
CAAT	*Computer Assisted Audit Techniques*
CASE	*Computer Aided Software Engineering*
DES	*Data Encryption Standard*
DS	Desenvolvimento do Sistema
ERP	*Enterprise Resource Planning*
ES	Engenharia de *Software*
IDEA	*International Data Encryption Algorithm*
ITF	*Integrated Test Facility*
MD	*Message Digest*
ON	Operação Normal
PC	Ponto de Controle
RACF	*Resource Access Control Facility*
RSA	*Rivest - Shamir - Adleman*
SAP	*Systeme, Anwendungand Programme*
SCARF	*System Control Audit Review File*
SI	Sistemas de Informação
TI	Tecnologia de Informação
UML	*Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO

Com um número cada vez maior de sistemas computacionais desenvolvidos e em desenvolvimento com o propósito de controlar operações de grande relevância no contexto das organizações, visto que empresas estão cada vez mais informatizadas, torna-se indispensável o trabalho de auditoria de sistemas neste âmbito (BRASIL, 1998).

A utilização da tecnologia da informação para a manipulação e armazenamento de dados nos órgãos, introduz novos riscos para o controle, acrescentando assim outras variáveis às questões relacionadas ao planejamento e execução de atividades de fiscalização. Dessa forma, constata-se a dificuldade de auditar entidades com alto grau de informatização (BRASIL, 1998).

Para isto esta pesquisa se propõe a desenvolver e implementar a aplicação da tecnologia da informação às atividades de fiscalização, ou seja, a Auditoria de Sistemas e seu desenvolvimento.

A auditoria de sistemas é uma atividade que engloba o exame das operações, processos, sistemas e responsabilidades de uma determinada entidade, com intuito de verificar sua conformidade com certos objetivos e políticas institucionais, orçamentos, regras, normas ou padrões (DIAS, 2000).

O auditor de sistemas, pessoa que trabalha nesta área, é responsável pela busca de falhas e por propor sugestões de melhorias relacionadas com a segurança física de equipamentos, suprimentos e instalações, segurança lógica e confidencialidade de sistemas, arquivos e informações (GIL, 2000).

A eficácia dos resultados gerados por uma auditoria e a eficiência dos processos concluídos necessitam ser validadas e avaliadas, sendo a auditoria de sistemas o campo de ação para a obtenção do alcance da qualidade de computação necessária.

Por meio da eficácia trazida pela auditoria tem-se a possibilidade de abranger corretamente elementos fundamentais na engenharia de *software*, como: métodos, ferramentas e procedimentos, que possibilitam ao gerente o controle do processo de desenvolvimento do sistema e oferece ao profissional uma base para a construção e manutenção de um *software* produtivo de alta qualidade (PRESSMAN, 1995).

Na atividade de auditoria não são conhecidas ferramentas que auxiliem no exame do desenvolvimento de sistemas, na engenharia de *software*. Atualmente, toda esta atividade é realizada manualmente e documentado com ferramentas e aplicativos impróprios, ou seja, *softwares* inadequados de edição de texto, planilhas eletrônicas, o que pode ocasionar perdas e danos provocados pelo grande volume de informações e dados.

Por isso neste projeto de pesquisa, implementou-se uma ferramenta, uma aplicação para auditoria em desenvolvimento de sistemas, que analisa e examina cada divisão da engenharia do *software* proposta a ser analisado e avaliado pela auditoria. Consequentemente se terá a possibilidade de extrair resultados e relatórios, como pontos fracos e fortes, melhorias, sugestões e ao final poder concluir se o sistema e seu desenvolvimento são ou não adequados, utilitários e confiáveis.

Isso se fará por meio dos papéis de trabalho, responsável por registros que evidenciam atos e fatos observados pelo auditor. Esses registros podem estar sob a forma de documentos, tabelas, planilhas, listas de verificações, entre outros. Tais documentos dão suporte ao relatório de auditoria, pois contêm o registro da metodologia

adotada, procedimentos, verificações, fontes de informação, testes, e demais informações relacionadas ao trabalho de auditoria executado (DIAS, 2000).

Outro problema relevante é a falta de segurança que ocorre no processo de auditagem, pelas mesmas razões citadas anteriormente, ou seja, como as informações são registradas manualmente e em ferramentas inadequadas, usuários não autorizados podem ter acesso, prejudicando seriamente os resultados da auditoria.

A proteção por criptografia é uma alternativa para preservar informações sigilosas. Independente do algoritmo criptográfico utilizado, sempre ocorrerá transformação de um texto legível em um ilegível. Mesmo que o invasor obtenha o conteúdo de um arquivo, dificilmente terá acesso aos dados (MORENO; PEREIRA; CHIARAMONTE, 2005).

O *Rivest, Shamir - Adleman* (RSA) é um sistema ou algoritmo de criptografia de chave assimétrica, onde se gera uma chave pública, geralmente utilizada para cifrar informações, e uma outra privada, utilizada para decifrar os dados (MORENO; PEREIRA; CHIARAMONTE, 2005). Dessa forma, é utilizado na aplicação desenvolvida o algoritmo de criptografia RSA, para maior segurança, proteção e sigilo, assegurando que dados não sejam violados e que apenas usuários com permissão consigam ter acesso a informações coletadas e disponibilizadas pela ferramenta de auditoria.

1.1 OBJETIVO GERAL

Desenvolver uma aplicação para auditoria em desenvolvimento de sistemas, utilizando criptografia nos dados coletados e manipulados nesta atividade.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa são:

a) compreender o conceito e a atividade de auditoria e auditoria de sistemas;
b) entender sobre a tecnologia da informação e sistemas de informação;
c) aprender a engenharia de *software e* as etapas utilizadas no desenvolvimento de um sistema;
d) utilizar, fiscalizar e aplicar técnicas, ferramentas e controles de auditoria nas etapas e métodos da engenharia de *software*, no desenvolvimento de sistemas;
e) compreender a segurança da informação através de mecanismos de segurança;
f) entender a criptografia de dados como mecanismo de segurança;
g) utilizar o algoritmo RSA de criptografia para maior segurança dos dados coletados e disponibilizados na ferramenta de auditoria;
h) construção de uma ferramenta para auditar o desenvolvimento de sistemas utilizando criptografia.

1.3 JUSTIFICATIVA

Atualmente, a disseminação e popularidade dos computadores e a alta exigência do mercado, têm causado um aumento significativo no número de *softwares* implementados, porém, o mesmo não acontece com ferramentas na área de auditoria de sistemas.

Com o aumento da complexidade dos sistemas computacionais desenvolvidos, o uso de *software* como ferramentas de apoio à auditoria e o desenvolvimento da área de segurança de informação tornou-se necessidade. Assim, esses *softwares* são de grande importância, até mesmo para a capacitação de profissionais de informática na área de auditoria (DIAS, 2000).

A auditoria de sistemas dentro de uma organização é de vital importância, pois aumentam consideravelmente o controle sobre a integridade, operacionalidade e segurança das informações. Evitam também transtornos, sendo os principais, as fraudes e erros, além de garantir o bom desempenho e funcionalidade dos sistemas examinados pela atividade.

Os objetivos da auditoria são metas de controle a serem alcançadas, ou efeitos negativos a serem evitados, para cada tipo de transação, atividade ou função fiscalizada (DIAS, 2000).

Os poucos *softwares* disponíveis em nosso mercado para acompanhamento de projetos e aplicações não dão suporte as especificidades e atividades necessárias ao trabalho de auditoria, sendo assim será desenvolvida uma aplicação voltada exclusivamente a auditoria de sistemas em desenvolvimento.

A excelência em *software p*ara ser alcançada tem-se definir e implementar um sistema de tal forma que ele: cumpra seus objetivos, seja gerenciável, passível de manutenção, com longa vida e de fácil aprendizagem (SHILLER, 1992). Desta forma, a auditoria servirá como uma ferramenta de aperfeiçoamento e facilitação de todos estes objetivos almejados para a excelência do *software*, do desenvolvimento de sistema.

Porém, esta pesquisa também propõe uma solução para outro problema encontrado, a falta de segurança nos dados e resultados obtidos pela auditoria. A fim de

garantir a segurança será usada a criptografia, com intuito de firmar o serviço de confidencialidade, proteção e sigilo dos dados (DIAS, 2000).

O algoritmo de criptografia a ser utilizado é o RSA, um algoritmo muito usado e testado. Ele se mostra consideravelmente forte, se adequadamente usado (CARVALHO, 2001). Outro fator a ser considerado, é a maior compreensão e domínio sobre este algoritmo e também pela sua grande difusão.

Conseqüentemente, este projeto também servirá para contribuir com as disciplinas de Engenharia de *Software* I e II e Auditoria de Sistemas. Nesta última, é realizada uma auditoria no sistema desenvolvido e implementado nas disciplinas de engenharia de *software*. Com essa ferramenta se terá maior facilidade e melhor execução desta atividade, além de dar mais aptidão aos acadêmicos.

Outra questão é a falta de uma aplicação que facilite as atividades de um auditor em serviço para auxiliar em auditoria, abordado por um profissional da área, Paracelso de Oliveira Caldas.

1.4 ESTRUTURA DO TRABALHO

A seguir será apresentado um breve resumo dos capítulos que compõem este trabalho.

O primeiro capítulo apresenta um aspecto geral do trabalho, seus objetivos gerais e específicos e a importância de sua realização para a área de Auditoria e Engenharia de *Software*, definindo o problema existente e a sua justificativa.

O segundo capítulo aborda sobre a Auditoria num todo, mostrando suas fases, seus princípios, conceitos e sua natureza.

No terceiro capítulo têm-se uma abordagem mais específica sobre a Auditoria de Sistemas, suas ferramentas e técnicas.

O quarto capítulo trata da Tecnologia da Informação e sua Auditoria de uma forma resumida focando mais para os Sistemas de Informação e o Desenvolvimento de Sistemas (Engenharia de *Software*) e sua Auditoria.

O quinto capítulo descreve sobre a Segurança da Informação, mostrando seus objetivos, mecanismos e apontando os sistemas criptográficos como meio para alcançar a segurança.

O sexto capítulo apresenta a Criptografia de Dados, seus tipos (Simétrica e Assimétrica) e o Algoritmo RSA que será utilizado na implementação do projeto.

O sétimo capítulo apresenta alguns Trabalhos Correlatos que tem alguma relação com a pesquisa.

E por fim, o oitavo capítulo trata especificamente do trabalho desenvolvido, onde se explica o seu funcionamento, modelagem, composição e resultados obtidos. E finalmente são apresentadas as conclusões e sugestões para trabalhos futuros.

Diante do exposto acima, a partir do próximo capítulo tem-se toda a fundamentação necessária para a realização deste trabalho de pesquisa.

2 AUDITORIA

A auditoria é uma atividade que engloba o exame de operações, processos, sistemas e responsabilidades gerenciais de uma determinada entidade, com o intuito de verificar sua conformidade com certos objetivos e políticas institucionais, orçamentos, regras, normas ou padrões (DIAS, 2000).

As atividades de fiscalização da auditoria é um instrumento de direção da entidade, dos acionistas, do ambiente externo à organização, do usuário para, independentemente, opinar, validar e avaliar a qualidade em termos de segurança, eficiência dos trabalhos desenvolvidos com a tecnologia dos computadores.

A auditoria visa descobrir irregularidades, erros e fraudes no tratamento das informações da organização e também identifica os pontos que irão desagradar à alta administração para que estes possam ser corrigidos. Motivos que levam a auditoria a possuir um grande papel e um alto escalão no ambiente empresarial de uma organização, conforme a Figura 1.

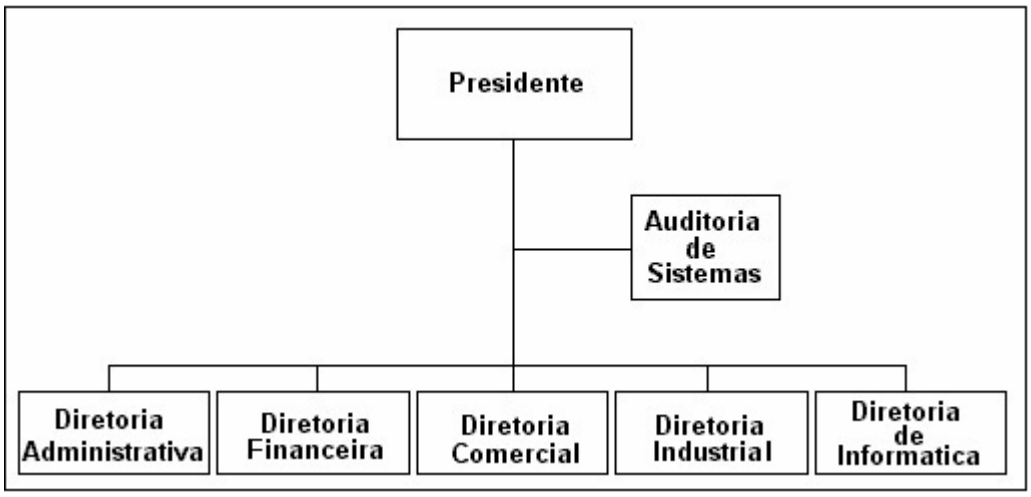


Figura 1. Organização e situação da área de auditoria de sistemas
Fonte: GIL, A. (2000)

2.1 FASES DA AUDITORIA

A atividade de auditoria pode ser dividida em três fases: planejamento; execução e relatórios. Esta divisão e explicação de cada fase estão conforme Dias (2000).

2.1.1 Planejamento

Identifica os instrumentos indispensáveis para a realização de uma auditoria. Estabelecem os recursos necessários para a execução do trabalho, área de verificação, as metodologias, os objetivos de controle e os procedimentos a serem adotados e uma fonte de informação completa do objeto a ser auditado.

2.1.2 Execução

Processo em atividade da auditoria, reunido de evidências suficientemente confiáveis, relevantes e úteis para a consecução dos objetivos da auditoria. Os achados da auditoria e as conclusões devem ser suportados pela correta interpretação e análise dessas evidências. Toda essa documentação, geralmente organizada em papeis de trabalho, deve estar disponível para auxiliar a equipe de auditoria na elaboração dos resultados, ou seja, dos relatórios.

2.1.3 Relatórios

O trabalho, seus achados e conclusões do auditor e sua equipe são normalmente apresentados em forma de relatórios, os quais incluem fatos sobre a entidade avaliada, comprovações, conclusões e, eventualmente, recomendações e/ou determinações. Estes são encaminhados principalmente para a diretoria da organização, ou ao organismo que financia a entidade auditada ou ao responsável pelo controle e auditoria geral da entidade.

2.2 PRINCÍPIOS E CONCEITOS DE AUDITORIA

O campo da auditoria compõe-se de aspectos, como: objetivo a ser fiscalizado; período e natureza da auditoria. O objetivo pode ser uma entidade completa (instituição pública ou privada), uma parte selecionada ou uma função dessa entidade, e o período a ser fiscalizado pode ser um mês, um ano ou até mesmo corresponder ao período completo da gestão de um administrador da instituição (DIAS, 2000).

O âmbito da auditoria estabelece a amplitude e exaustão dos processos da auditoria, definindo até que ponto serão aprofundadas as tarefas e seu grau de abrangência. Desta forma, distingue-se a área de verificação, conjunto formado por campo e âmbito de auditoria, delimitando de modo preciso os temas da auditoria, em função da entidade a ser fiscalizada e da natureza da auditoria (DIAS, 2000). Esta afirmação fica mais claramente evidenciada na Figura 2.

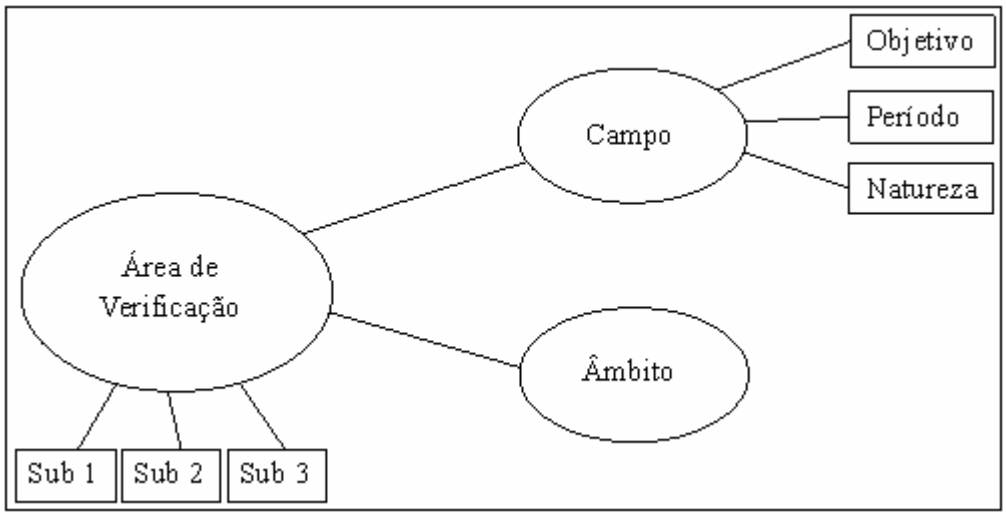


Figura 2. Conceitos básicos: área de verificação
Fonte: DIAS, C. (2000)

A seguir serão citados detalhadamente os termos mais importantes e comumente relacionados com auditoria, segundo Dias (2000), como: os controles; objetivos de controle; procedimentos; achados de auditoria; papéis de trabalho e recomendações de auditoria (relatórios).

2.2.1 Controle

Controle é a fiscalização exercida sobre as atividades de pessoas, órgãos, departamentos ou sobre produtos, para que tais atividades ou produtos não se desviem das normas preestabelecidas.

Os controles podem ser classificados em três tipos:

a) preventivos: utilizados para prevenir erros, omissões ou atos fraudulentos, como por exemplo, senha de acesso a um determinado sistema desenvolvido;

b) detectivos: usados para detectar erros, omissões ou atos fraudulentos e ainda relatar sua ocorrência, como por exemplo, *softwares* de controle de

acesso e relatórios de tentativas de acesso não autorizado a um determinado sistema;

c) corretivos: servem para reduzir impactos ou corrigir erros uma vez detectados.

2.2.2 Objetivos de Controle

Os objetivos de controle são metas de controle a serem alcançadas, ou efeitos negativos a serem evitados, para cada tipo de transação, atividade ou função fiscalizada. Esses objetivos de controle para serem alcançados na prática, são traduzidos em diversos procedimentos de auditoria.

2.2.3 Procedimentos da Auditoria

Os procedimentos formam um conjunto de verificações e averiguações que permitem obter e analisar as informações necessárias a formulação e conclusão da opinião e visão do auditor. Em geral, são listas de pontos a serem verificados durante a auditoria que proporcionam um aumento de produtividade e de qualidade do trabalho do auditor.

2.2.4 Achados da Auditoria

Os achados de auditoria são fatos significativos e relevantes baseado em fatos e evidências observados pelo auditor e sua equipe durante a execução da auditoria,

onde na maioria das vezes, são associados a falhas e irregularidades, porém os achados podem também indicar pontos fortes da instituição ou sistema auditado.

2.2.5 Papéis de Trabalho

Segundo Dias (2000, p.10) os papéis de trabalhos são descritos da seguinte forma:

> Os papéis de trabalho são registros que evidenciam atos e fatos observados pelo auditor. Esses registros podem estar sob a forma de documentos, tabelas, planilhas, listas de verificações, arquivos informatizados, etc. Esses documentos dão suporte ao relatório de auditoria, pois contêm o registro da metodologia adotada, procedimentos, verificações, fontes de alimentação, teste, enfim, todas as informações relacionadas ao trabalho de auditoria executado.

A documentação dos papéis de trabalhos constitui um conjunto de formulários preenchidos logicamente no processo de auditoria de sistemas, com anexos e provas documentais que evidenciam fatos relatados. Estes documentos contêm informações coletadas durante os testes, os procedimentos executados e as opiniões formadas sobre o objetivo da auditoria. As bases de papéis de trabalhos de auditoria de sistemas constituem informações de planejamento, execução, monitoramento e revisões, controles dos usuários do sistema e senhas e alguns recursos de auxilio ao usuário (IMONIANA, 2005).

Conforme Brasil (1998) os papéis de trabalho devem ser documentados de forma a deixarem claro:

a) os objetivos de auditoria que serão fundamentados por dados processados por computador;
b) o grau de importância desses dados para os objetivos de auditoria, e as fontes adicionais que podem confirmá-los;

c) as informações coletadas sobre os dados, o sistema que os processa e seus controles.

2.2.6 Recomendações da Auditoria (Relatórios)

Na fase de relatórios, são feitas as recomendações de auditoria. Essas recomendações são medidas corretivas possíveis, sugeridas pela instituição fiscalizadora ou pelo auditor e sua equipe em seu relatório, para corrigir as deficiências detectadas durante a auditoria. Essas recomendações podem se transformar em determinações a serem cumpridas, dependendo da competência ou posição hierárquica do órgão de controle em relação à entidade ou sistema auditado.

Os objetivos de auditoria para serem alcançados, o relatório deve assegurar que a informação em que se baseia é confiável. Segundo Brasil (1998) deve ser informado nos relatórios:

a) a abrangência da avaliação dos controles, quando a confiança atribuída aos controles do sistema é utilizada para reduzir o teste de dados;
b) os tipos de teste de dados executados, seu propósito, e as taxas de erro encontradas nas três áreas de operação (entrada, processamento e saída de dados);
c) qualquer fator conhecido que limite a confiabilidade dos dados, e o tipo de influência que essa limitação teria sobre os resultados e conclusões apresentados.

2.3 NATUREZA DA AUDITORIA

Nesta pesquisa se apresentará alguns dos tipos mais comuns de Auditoria segundo Dias (2000), pois segundo ele não existe padrão classificatório (tipologia) dos diversos tipos de auditoria existentes. Estes tipos de Auditorias são classificados de acordo com os seguintes aspectos: órgão fiscalizador; forma de abordagem do tema e tipo ou área envolvida, como a auditoria no desenvolvimento de sistemas (engenharia de *software*), foco principal desta pesquisa.

2.3.1 Órgão Fiscalizador

Quanto ao órgão fiscalizador, estes podem ser classificados segundo Dias (2000) em:

a) auditoria interna: realizada por órgão interno da entidade, tem como objetivo reduzir as probabilidades de fraudes, erros, práticas ineficientes ou ineficazes. Deve ser independente e prestar contas diretamente à direção da instituição;

b) auditoria externa: executada por instituição externa e independente da entidade fiscalizada, com objetivo de emitir parecer sobre gestão de recursos, situação financeira, a legalidade de suas operações;

c) auditoria articulada: trabalho conjunto de auditorias internas e externas caracterizada pelo uso de recursos e comunicações recíprocas dos resultados.

2.3.2 Forma de Abordagem do Tema

Em relação à forma de abordagem do tema, a auditoria pode ser rotulada segundo Dias (2000) em:

a) auditoria horizontal: aborda tema específico, realizada em várias entidades ou serviços paralelamente;
b) auditoria orientada: focada em uma atividade específica qualquer ou em atividade com fortes indícios de erros ou fraudes.

2.3.3 Tipo ou Área Envolvida

A respeito do tipo de auditoria exercida ou da área envolvida na fiscalização, as principais classificações segundo Dias (2000) são:

a) auditoria de programas do governo: acompanhamento, exame e avaliação da execução de programas e projetos governamentais específicos (efetividade das medidas governamentais);
b) auditoria do planejamento estratégico: verifica se os principais objetivos da entidade são atingidos e se as políticas e estratégias de aquisição, utilização e alienação de recursos são respeitadas;
c) auditoria administrativa: engloba o plano da organização, seus procedimentos e documentos de suporte à tomada de decisão;
d) auditoria contábil: tem objetivo de garantir a correção das contas da instituição, conforme as devidas autorizações;
e) auditoria financeira: análise das contas, da situação financeira, da legalidade e regularidade das operações e aspectos contábeis financeiros,

orçamentários e patrimoniais, verificando se todas as operações foram corretamente autorizadas, liquidadas, ordenadas, pagas e registradas;

f) auditoria de legalidade: analise da legalidade ou regularidade das atividades, funções, operações ou gestão de recursos, verificando se estão em conformidade com a legislação em vigor;

g) auditoria operacional: analisa todos os níveis de gestão, nas fases de programação, execução e supervisão, sob o ponto de vista da economia, eficiência e eficácia. Também conhecida como auditoria de eficiência, de gestão, de resultados ou de práticas de gestão. São auditados todos os sistemas e métodos utilizados pelo gestor para a tomada de decisão, analisa a execução das decisões e aprecia até que ponto os resultados pretendidos foram atingidos;

h) auditoria integrada: inclui auditoria financeira e a operacional;

i) auditoria da tecnologia da informação: analisa os sistemas de informação, o ambiente computacional, a segurança de informações, e o controle interno da entidade, identificando deficiências e pontos fortes. É essencialmente operacional, conhecida como auditoria informática, computacional ou de sistemas;

j) auditoria do desenvolvimento de sistemas (engenharia de *software*): objetiva avaliar o ajustamento das metodologias e procedimentos de projeto, desenvolvimento, implantação e revisão dos sistemas produzidos, abrangendo o ambiente e o processo de desenvolvimento de um sistema específico, ainda na fase de planejamento, já em andamento ou após conclusão.

Desta forma, finaliza-se este capítulo que aborda a Auditoria num todo, de uma forma geral, mostrando sua definição, sua amplitude e extensão, onde a mesma pode seguir várias áreas e segmentos. Por isso, a partir do próximo capítulo se dá início mais especificamente a Auditoria na área computacional, ou seja, a Auditoria de Sistemas, junto com suas técnicas e ferramentas.

3 AUDITORIA DE SISTEMAS

A auditoria de sistemas avalia o ambiente de processamento de dados para identificar e avaliar os possíveis riscos (acesso indevido, erros, falhas, irregularidades, ineficiências, etc.) que estejam ocorrendo, ou que possam ocorrer, e faz recomendações para correção e melhoria dos controles para a diminuição do grau dos riscos levantados (SILVA JÚNIOR, 2001).

A atividade de auditoria de sistemas é voltada à avaliação dos procedimentos de controle e segurança vinculados ao processamento eletrônico das informações. Tem como funções: documentar, avaliar e monitorar sistemas de controles legais, gerenciais de aplicação e operacionais. Os instrumentos para desempenhar tais funções podem variar do uso da auditoria de *software* ao uso habilidoso de técnicas (SILVA JÚNIOR, 2001).

Segundo Silva Júnior (2001) a auditoria de sistemas objetiva certificar-se de que:

a) as informações são corretas e oportunas;
b) existe um processamento adequado e correto das operações;
c) as informações são bem protegidas, por exemplo, contra fraudes;
d) tem proteção das instalações e dos equipamentos;
e) existe a proteção contra situações de emergências (paralisação de processamento, perda de arquivos, inundações, incêndios, entre outros).

3.1 FERRAMENTAS E TÉCNICAS DE AUDITORIA DE SISTEMAS

As técnicas de auditoria com o auxílio do computador, conhecidas como *Computer Assisted Audit Techniques* (CAAT), são procedimentos efetuados pelo auditor, no processo de auditoria, com a utilização de ferramentas automatizadas que o auxiliam na análise de saldos e testes do controle em sistemas (SILVA JÚNIOR, 2001).

O principal objetivo do uso destas técnicas é auxiliar o profissional da área a auditar totalmente a população da área ou transação revidada, considerando o limite de tempo que possui, aproveitando os recurso de *softwares* e as técnicas de auditoria em ambiente de computação. Essas técnicas são importantes, pois auxiliam na avaliação de ambientes e sistemas, que geralmente processam um grande volume de dados e transações (IMONIANA, 2005).

A utilização de técnicas de auditoria auxiliadas por computador, como em qualquer trabalho de auditoria, deve ser efetuada por profissionais com adequado conhecimento e treinamento técnico e supervisão apropriada. As ferramentas computadorizadas que executam as tarefas têm que ser confiáveis, assim como se pode precisar do envolvimento do pessoal com conhecimento em sistemas. Dessa forma, é recomendável a utilização de um *software* específico. O uso de planilhas eletrônicas, *softwares* inadequados ou interrogações diretamente na base das informações, normalmente, não são métodos suficientemente seguros e confiáveis (SILVA JÚNIOR, 2001).

3.1.1 Ferramentas

As ferramentas auxiliam na extração, sorteio, seleção de dados e transações, com o intuito de apontar com atenção para as discordâncias e desvios existentes. A seguir serão relatadas as ferramentas mais utilizadas atualmente segundo Imoniana (2005), como: *softwares* generalistas de auditoria de sistemas; *softwares* especializados em auditoria e programas utilitários.

3.1.1.1 *Software* Generalista de Auditoria de Sistemas

Envolve o uso de *software* aplicativo (um conjunto de programas) em ambiente *batch*, que pode processar, além de simulação paralela, uma variedade de funções de auditoria e nos formatos que o auditor deseja. As funções são tais como extração de dados de amostra, testes globais, geração de dados estatísticos para análise, sumarização, composição de um outro arquivo a partir de um arquivo mestre de dados, apontamento de duplicidade de registros ou seqüência incorreta entre outras.

As vantagens do uso dessa ferramenta é que o *software* pode processar vários arquivos ao mesmo tempo, pode processar vários tipos de arquivos com formatos diferentes e poderia fazer integração sistêmica com vários tipos de *software* e *hardware*.

Porém existem desvantagens, como: o processamento das aplicações envolve gravação de dados (arquivos) em separado para serem analisados em ambientes distintos, poucas aplicações poderiam ser feitas em ambiente *on-line* e esta ferramenta evita aprofundar as lógicas e matemáticas muito complexas.

3.1.1.2 *Softwares* Especializados de Auditoria

Consiste no programa desenvolvido especificamente para executar certas tarefas numa circunstância definida. O programa pode ser desenvolvido pelo próprio auditor, pelos especialistas da empresa auditada ou por um contratado pelo auditor.

Seus pontos fortes estão em atender aos sistemas ou transações incomuns que não tem contemplados nos *softwares* generalistas, o auditor, quando consegue desenvolver *softwares* específicos numa área muito complexa, pode utilizar isso como vantagem competitiva.

Entretanto existem pontos negativos, como: pode ser muito caro, uma vez que seu uso será limitado e normalmente restrito somente a um cliente e a atualização desses *softwares* pode ser problemática por falta de recursos que acompanhem as novas tecnologias.

3.1.1.3 Programas Utilitários

Utilizados para executar algumas funções muito comuns de processamento, como por exemplo: sortear arquivo, sumarizar, concatenar, gerar relatórios. Porém, vale ressaltar que esses programas não foram desenvolvidos e nem tem recursos para executar as funções de auditoria.

A sua vantagem é que o *software* pode ser utilizado como arranjo, uma alternativa simples na ausência de outros recursos, e sua desvantagem é que sempre necessitará do auxílio do funcionário da empresa auditada para operar a ferramenta.

3.1.2 Técnicas

As técnicas de auditoria, também chamadas de metodologias, proporcionam aos usuários varias vantagens se usadas corretamente. Segundo Imoniana (2005) são elas:

a) produtividade: com a melhoria no processo de planejamento, ajuda na redução do ciclo operacional de auditoria, focalizando o exercício nas funções mais importantes, eliminando tarefas repetitivas;

b) custo: reduz custos relacionados com auditoria, pois não necessita de geração de relatórios e listagem para análise. O auditor tem acesso remotamente, eliminando a necessidade de deslocamento e poupando custo de viagens;

c) qualidade assegurada: com o uso de *softwares* que têm padrões devidamente testados, o auditor aproveita para adequar seus trabalhos aos padrões internacionais, aumentando a qualidade dos serviços prestados. Além disso, aumenta a cobertura dos riscos da auditoria, pois todos os dados podem ser testados;

d) valor agregado: disponibiliza rapidamente os resultados para a tomada de decisões que necessitam de mudanças de rumos mais urgentes, facilitando a correção dos desvios ou irregularidades em tempo hábil;

e) benefícios corporativos: proporcionam as empresas auditoras:

- eficiência nos trabalhos,
- eficácia,
- otimização de recursos disponíveis, compartilhamento de ambientes entre vários auditores em múltiplas localidades,

- melhoria na imagem do auditor, por estar utilizando tecnologia mais apropriada;

f) benefícios para o auditor: proporcionam aos auditores:

- independência, por exemplo na geração de relatórios,
- renovação do foco de auditoria, visando atender as expectativas e tendências do mercado,
- eliminação de tarefas repetitivas, que geralmente podem ser automatizadas,
- mais tempo para pensar e ser criativo nas sugestões para seus clientes,
- redução do risco de auditoria, uma vez que, tudo sendo programado, nada passará despercebido.

Na área da auditoria existem inúmeras técnicas com sua lógica, seus objetivos de aplicação, como também as características de seu conteúdo a cada momento da aplicação (GIL, 2000).

Atualmente existem varias técnicas de Auditoria de Sistemas segundo Gil (2000), como: programa de computador; questionários; simulação de dados (*test-deck*); visita *in loco*; mapeamento estatístico (*mapping*); rastreamento-programas (*tracing*); entrevistas; análise de relatórios/telas; simulação paralela; análise *log*/*accounting*; análise do programa fonte e exibição parcial da memória *snap shot.*

Algumas dessas técnicas de auditoria citadas acima quase nunca são usadas e outras impraticáveis pela maioria dos auditores, portanto serão apresentadas e explicadas nesta pesquisa apenas as técnicas mais utilizadas atualmente conforme Imoniana (2005), como: dados de teste; facilidade de teste integrado; simulação paralela; lógica de auditoria embutida nos sistemas; rastreamento e mapeamento e análise da lógica de programação.

3.1.2.1 Dados de Teste

A técnica de dados de teste, também conhecida por *test data* ou *test deck*, envolve o uso de conjunto de dados de entrada especialmente preparado com o objetivo de testar os controles programados e os controles de sistemas aplicativos. Para que isto seja feito, roda-se uma gama de transações, e comparam-se os resultados obtidos com aqueles predeterminados (IMONIANA, 2005).

O *test deck* é uma técnica aplicada para teste de processos computacionais. Corresponde à elaboração de um conjunto de dados de teste a ser submetido ao programa de computador ou a determinada rotina que o compõe, que necessita ser verificada em sua lógica de processamento (GIL, 2000).

A seguir, serão citadas algumas vantagens e desvantagens da aplicação desta técnica segundo Imoniana (2005):

Vantagem:

a) pode-se usar *software* de gerador de dados normalmente utilizado para gerar massa de dados nos processos de testes de sistemas em desenvolvimento. Os dados podem ser elaborados por pessoas que possuem um mínimo conhecimento técnico de informática.

Desvantagem:

a) na aplicação desta técnica é difícil planejar e antecipar todas as combinações de transações que possam acontecer em ambiente de negócios de empresas.

3.1.2.2 Facilidade de Teste Integrado

A execução técnica, conhecida por *Itegrated Test Facility* (ITF), envolve a aplicação de entidades fictícias, tais como funcionários nulos na folha de pagamento ou cliente inexistente nas contas a receber. São confrontados os dados no processamento de transações reais com esse dados e os resultados comparados com aqueles predeterminados, com isso evita-se que sejam atualizadas as bases reais da organização com dados fictícios (IMONIANA, 2005).

A seguir, serão citadas algumas vantagens e desvantagens da aplicação desta técnica segundo Imoniana (2005):

Vantagem:

a) possibilita o teste de facilidades para a identificação da eficácia das operações em ambiente rotineiro e testa os controles programados nos sistemas. Esta técnica roda num ambiente normal das empresas.

Desvantagens:

a) os efeitos das transações devem ser retirados e dissolvidos, gerando assim mais trabalhos adicionais;
b) as quantidades de dados fictícios incluídos num ambiente podem ser limitados;
c) há possibilidade de se contaminar dados reais com dados fictícios.

3.1.2.3 Simulação Paralela

Envolve o uso de um programa especialmente desenvolvido pelo auditor que atenda a todas as lógicas necessárias para um aplicativo devidamente testado e que

atua como um sistema que tem função de processar transações e dados anteriormente executados numa rotina normal e operacional da empresa com o objetivo de verificar se os resultados são idênticos. Ademais, simula operações normais com o objetivo de estimular a verificação de resultados recorrentes que são inconsistentes (IMONIANA, 2005).

A simulação paralela é a elaboração de um programa de computador para simular as funções de rotina do sistema sob auditoria, onde esta técnica utiliza-se dos dados rotineiros alimentados à rotina do sistema sob auditoria como entrada do programa de computador para auditoria, simulado e elaborado pelo auditor (GIL, 2000).

A seguir, serão citadas algumas vantagens e desvantagens da aplicação desta técnica segundo Imoniana (2005):

Vantagens:

a) custos relacionados com tempo não existem, visto que o programa opera em ambiente real;
b) pode-se processar um grande volume de dados auditados;
c) o teste é mais detalhado e mais representativo, dando maior segurança para o auditor.

Desvantagens:

a) o custo de desenvolvimento de uma simulação paralela pode ser muito alto;
b) o auditor necessitaria de uma habilidade específica para executar uma operação paralela;
c) a simulação paralela tem escopo de teste muito restrito.

3.1.2.4 Lógica de Auditoria Embutida nos Sistemas

Envolve a inclusão de lógicas de auditoria nos sistemas quando são desenvolvidos. Periodicamente, os relatórios de auditoria são emitidos para a revisão e o acompanhamento dos procedimentos operacionais. Como exemplo pode-se citar o *Resource Access Control Facility* (RACF) Auditor, um recurso de auditoria construído junto com o sistema de segurança RACF que possibilita o monitoramento do próprio sistema de segurança dos auditores (IMONIANA, 2005).

A seguir, serão citadas algumas vantagens e desvantagens da aplicação desta técnica segundo Imoniana (2005):

Vantagens:

a) todas as atividades do sistema onde é construída a lógica de auditoria podem ser monitoradas permanentemente com simples acessos;
b) pode ser usada com sistemas processados de forma *on-line*;
c) não apresenta restrições quanto à entrada de dados.

Desvantagens:

a) implantação da lógica de auditoria embutida nos sistemas exige custo adicional da utilização de máquinas;
b) as empresas podem ter dificuldades em implementa-lo se não for concebido e desenvolvido com o sistema.

3.1.2.5 Rastreamento e Mapeamento

Envolve desenvolvimento e implementação de uma trilha de auditoria para acompanhar certos pontos da lógica do processamento de algumas transações. O

mapeamento da execução de transações em programas dá-se com o apontamento de alguns dados estatísticos, tais como algumas funções não executadas, tempo de máquinas utilizado, as funções executadas e quantas vezes, entre outros registros que devem ser documentado (IMONIANA, 2005).

Rastreamento é uma técnica que possibilita seguir o caminho de uma transação durante o processamento do programa. Seu intuito é de identificar as inadequações e ineficiência na lógica de um programa, além de viabilizar a identificação de rotinas fraudulentas pela alimentação de transações particulares. Por sua vez, mapeamento é a técnica de computação que pode ser utilizada pelo auditor para efetuar verificações durante o processamento dos programas, flagrando situações como: rotinas não atualizadas e a quantidade de vezes que cada rotina foi utilizada quando submetida a processamento de uma quantidade de dados (GIL, 2000).

3.1.2.6 Análise da Lógica de Programação

A análise da lógica de programação envolve a verificação da lógica de programação para certificar que as instruções dadas ao computador são as mesmas já identificadas nas documentações dos sistemas aplicativos. Essa técnica pode ser feita manualmente e possibilita a conformação da efetividade dos controles programados (IMONIANA, 2005).

A técnica de avaliação da lógica (programa-fonte) implica em analisar o visual do código-fonte (linguagem em que o usuário ou programador desenvolveu a ferramenta) do programa de computador componente do sistema sob auditoria (GIL, 2000). Este técnica utilizada na auditoria de sistemas e aplicativos também será de muita importância e utilização na auditoria de desenvolvimento de sistemas.

Com isso, constatou-se o foco da Auditoria de sistemas, junto com suas técnicas e ferramentas. A Auditoria de Sistemas é o ramo da Auditoria mais conhecido e trabalhado atualmente e visa à fiscalização e o exame de sistemas e aplicações prontas, porém a mesma servirá de alicerce para aplicarmos a Auditoria no Desenvolvimento de Sistemas (Engenharia de *Software*), pois suas teorias e técnicas poderão ser aplicadas em ambas.

Deste modo, o próximo capítulo abordará os Sistemas de Informações junto com o seu Desenvolvimento (etapas, processos), sua engenharia de *software*, e como a Auditoria se enquadrará com suas técnicas e teorias nesta área.

4 TECNOLOGIA DA INFORMAÇÃO

A grande maioria de órgãos, entidades e usuários já utilizam maciçamente a tecnologia da informação (TI) para automatizar suas operações e registrar, processar, manter e apresentar informações (BRASIL, 1998).

Não se deve partir do princípio de que dados e informações extraídos de computadores são confiáveis, muito menos no desenvolvimento de um sistema, na construção de *softwares* e menos ainda em sistemas prontos utilizados por usuários. Embora ofereçam vantagens para as organizações, os sistemas informatizados podem também representar grandes riscos.

É possível que erros e fraudes não sejam detectados por causa da enorme quantidade de dados, informações e operações controlados pelos sistemas, da possível desorganização entre o que está armazenado e o que é efetivamente apresentada em relatórios de saída, e da mínima necessidade de intervenção humana nos processos (BRASIL, 1998).

4.1 AUDITORIA DA TECNOLOGIA DA INFORMAÇÃO

A auditoria em ambiente de tecnologia da informação entende que as informações disponíveis em forma de papel são agora guardadas em forma eletrônica e que o enfoque de auditoria é de assegurar de que essas informações em forma eletrônica sejam confiáveis (IMONIANA, 2005).

Segundo Imoniana (2005) as atividades da auditoria de tecnologia de informação visam utilizar os recursos de informática para auditar o próprio computador e automatizar todos os processos de auditoria. Entre os objetivos e os benefícios desta

automação de auditoria, destacam-se: melhorar a eficiência e reduzir custos; aumentar a qualidade do trabalho de auditoria; atender as expectativas dos clientes; preparar-se para a globalização; treinamento de pessoal e superação de resistências a tecnologia; avaliação, escolha e implantação de *softwares* e *hardwares*; dispositivos de segurança e backup; independências de limitações impostas pelos arquivos de auditoria em papel; economia de tempo de atualizações; melhor qualidade na apresentação; fluxo de informações mais rápido e maior satisfação, respeito e produtividade.

A auditoria de tecnologia de informação é um tipo de auditoria operacional, ou seja, que analisa a gestão de recursos, abordando os aspectos de eficiência, eficácia, economia e efetividade. Esse tipo de auditoria pode abranger o ambiente de informática como um todo (como segurança física e lógica e planejamento de contingências), ou a organização do departamento de informática (analisando aspectos administrativos da organização, tais como, políticas, padrões e procedimentos) e pode ainda envolver controles sobre os aplicativos (desenvolvimento de sistemas, entrada, processamento e saída de dados) (DIAS, 2000).

De acordo com Dias (2000) não existem subáreas definidas da auditoria da TI, as principais são:

a) auditoria da tecnologia da informação: é abrangente, engloba todos os controles que podem influenciar a segurança de informação e o correto funcionamento dos sistemas de toda a organização:
   - controles organizacionais,
   - de mudança,
   - de operação de sistemas,
   - sobre banco de dados,
   - sobre microcomputadores,

- sobre ambiente cliente-servidor;

b) auditoria da segurança de informações: determina a postura da organização com relação à segurança. Avalia a política de segurança e controles relacionados com aspectos de segurança institucionais mais globais, faz parte da auditoria da TI. Seu escopo envolve:
   - avaliação da política de segurança,
   - controles de acesso lógico,
   - controles de acesso físicos,
   - controles ambientais,
   - planos de contingência e continuidade de serviços;

c) auditoria de aplicativos: segurança e controle de aplicativos específicos, incluindo aspectos intrínsecos à área a que o aplicativo atende:
   - controles sobre o desenvolvimento de sistemas aplicativos,
   - controles de entradas, processamento e saída de dados,
   - controle sobre conteúdo e funcionamento do aplicativo, com relação à área por ele atendida.

4.2 SISTEMAS DE INFORMAÇÃO

O sistema é um conjunto de elementos inter-relacionados com o objetivo de produzir relatórios que coordenam a tomada de decisões gerenciais. Neste percurso, identificam-se o processo que transforma dados de entrada, agregados aos comandos gerenciais, em saídas, conforme pode ser visto na Figura 3. Dessa forma, a realimentação (*feedback*) do sistema faz com que sejam ativadas novas estratégias

empresariais visando à geração de informações qualitativas ou quantitativas para o alcance do sucesso (IMONIANA, 2005).

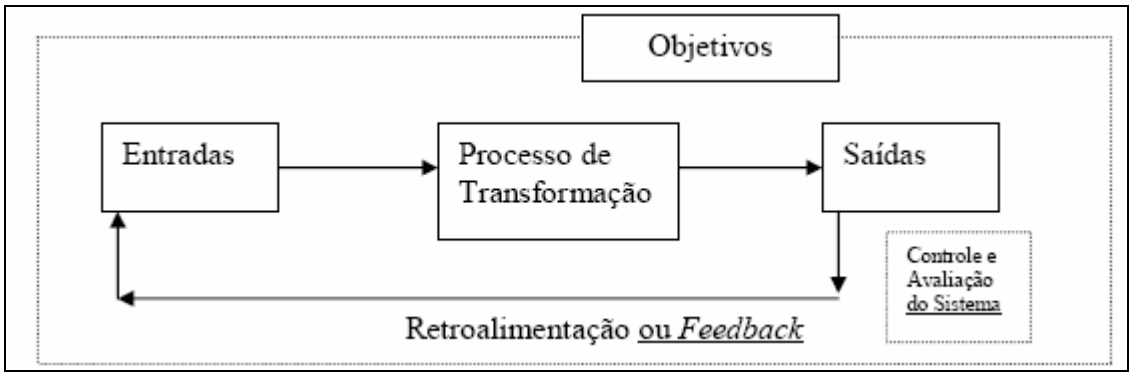


Figura 3. Fluxograma de um sistema
Fonte: MONTANHEIRO, P. (2006)

Sistemas de informação (SI) compreendem um conjunto de recursos humanos, materiais, tecnológicos e financeiros combinados segundo uma seqüência lógica para transformar dados em informações (GIL, 2000).

De acordo com Montanheiro (2002 apud MOSKOVE, 2006), define sistemas de informação como um conjunto de subsistemas inter-relacionados que funcionam em conjunto para coletar, processar, armazenar, transformar e distribuir informações para fins de planejamento, tomada de decisão e controle.

4.3 DESENVOLVIMENTO DE SISTEMAS E SUA AUDITORIA

A auditoria do desenvolvimento de sistemas, foco principal desta pesquisa, objetiva avaliar a adequação das metodologias e procedimentos de projeto, desenvolvimento, implantação e revisão dos sistemas produzidos dentro da organização auditada. Essa avaliação pode abranger apenas o ambiente de desenvolvimento da organização ou prever também a análise do processo de desenvolvimento de um sistema

específico, ainda na fase de planejamento, já em andamento ou após conclusão (BRASIL, 1998).

O desenvolvimento de um sistema de informação representa um investimento que não pode ser assumido sem dados confiáveis e precisos sobre o custo do projeto, seus benefícios e os riscos envolvidos. Todos os projetos de desenvolvimento de sistemas necessitam ser avaliados em profundidade, devendo ser procedidos de análises de custo/benefícios, capacidade de satisfação dos usuários e de atendimento aos objetivos da organização, custos de desenvolvimento, medidas de desempenho, planos de implementação, previsão de recursos humanos, entre outros (BRASIL, 1998).

Conforme Gil (2000) um sistema de informação possui um ciclo de vida caracterizado por:

a) ciclo de desenvolvimento: inicialização do projeto; estudo de viabilidade; análise da situação atual; projeto lógico; projeto físico; desenvolvimento e testes; implantação; administração do projeto e manutenção;
b) ciclo de operação (transformação do dado em informação): captação e registro de dados; conversão dos dados; consistência de dados; atualização de arquivos; armazenamento e recuperação de dados; apresentação das informações e utilização das informações.

O ciclo de vida de um sistema computacional, junto com o seu ciclo de desenvolvimento e operação, e a participação e o envolvimento que a auditoria possui, fica mais evidente na Figura 4.

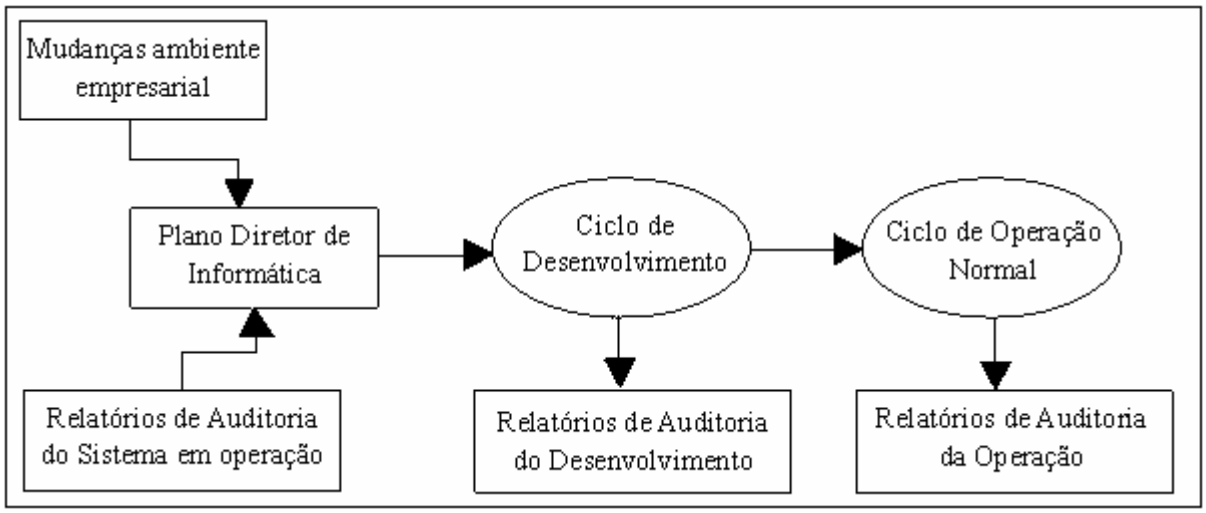


Figura 4. Participação da auditoria de sistemas no ciclo de vida do sistema
Fonte: GIL, A. (2000)

4.3.1 Ciclo de Desenvolvimento de Sistemas de Informação

A seguir, será mostrado o ciclo de desenvolvimento de um sistema de informação junto com algumas fases de evolução que podem ser separadas no desenvolvimento de sistemas e como a auditoria irá permitir a avaliação do ambiente e do processo nestas organizações auditadas. Esta abordagem e as fases a seguir estão de acordo com Brasil (1998).

4.3.1.1 Planejamento

Nesta fase a organização define:

a) as necessidades de informação ainda não atendidas e estabelece um plano de ação para o desenvolvimento dos sistemas de maior prioridade, estabelecendo a prioridade dos sistemas a serem desenvolvidos de acordo com sua importância para o cumprimento da missão institucional;

b) estabelece e documenta as metodologias de desenvolvimento a serem adotadas, que:
   - fornece uma abordagem estruturada de desenvolvimento,
   - o envolvimento ativo dos usuários no processo de desenvolvimento de sistemas,
   - prevê o uso de técnicas atuais, tais como tecnologia de banco de dados, redes de comunicação de dados, ferramentas *CASE* ou de reengenharia, linguagens de quarta geração, analise de requisitos, modelagem, dicionário de dados, ferramentas visuais, orientação a objetos, entre outras,
   - é suficientemente documentada, sendo capaz de oferecer orientação a funcionários com diversos níveis de conhecimento e experiência,
   - oferece meios de controlar mudanças nos requisitos de projeto que ocorram durante a vida do sistema,
   - inclui requisitos de programação, documentação, padrões para usuários, programadores, desenvolvedores de sistemas e operadores do centro de processamento de dados,
   - estabelece mecanismos de reavaliação, acompanhamento e controle em todas as fases do processo, pela equipe e as gerências envolvidas, permitindo redirecionar os trabalhos ou abandonar o projeto, quando se concluir que o novo sistema não irá atender às necessidades da organização;
c) define e documenta as responsabilidades de todas as pessoas envolvidas no desenvolvimento de sistemas, como indicações das responsabilidades

de cada parte envolvida (usuários, analistas de sistema, programadores, auditores, controle de qualidade, entre outros).

4.3.1.2 Plano de Desenvolvimento e Início do Projeto

O projeto é avaliado mais minuciosamente e detalhadamente quanto às análises de viabilidade técnica, custo/benefício, importância da informação, custos de desenvolvimento, planos de implementação, entre outros.

Se for confirmada a conveniência do desenvolvimento do projeto, a organização estabelece e aprova um plano de desenvolvimento, onde este possui elementos suficientes para permitir o seu projeto físico e lógico e tem por objetivo atacar deficiências reconhecidas ou problemas sistêmicos da organização.

Dessa forma é feita uma seleção para equipe do projeto, que deve ter em conjunto habilidades e conhecimentos suficientes para conduzir com sucesso as atividades de elaboração do projeto físico e lógico, desenvolvimento e implantação do sistema.

4.3.1.3 Organização do Projeto

Com base no plano de desenvolvimento aprovado, a equipe de projeto cria e submete à gerência um plano de trabalho, informando a abrangência e conteúdo do projeto, bem como os mecanismos de acompanhamento e controle (cronograma, datas-limite, processo de supervisão e acompanhamento das etapas, medidas de desempenho, entre outros), de acordo com o estabelecido na metodologia de desenvolvimento de sistemas.

A equipe do projeto definindo claramente a abrangência do projeto e o conteúdo do sistema de forma coerente com os objetivos previstos no plano de desenvolvimento consegue a concordância da gerência de planejamento e consequentemente a dos usuários.

4.3.1.4 Elaboração do Projeto do Sistema

A fase de elaboração do projeto do sistema consiste na produção dos seus projetos físico e lógico, ou seja, a equipe de projeto define detalhadamente as especificações técnicas e funcionais do sistema, da forma mais completa possível, e elabora o projeto de sistema de acordo com essas especificações, levando em conta o ambiente de operação existente.

No projeto de sistema, apresentam-se as seguintes especificações:

a) características do sistema: físicas (plataforma, recursos exigidos) e funcionais (atividades a serem executadas);
b) restrições ou impedimentos que possam limitar sua implementação;
c) tecnologias envolvidas (arquitetura do sistema, sistemas de segurança, questões de integração, tais como interconectividade e interoperabilidade);
d) abordagem adotada para o projeto, seus componentes mais importantes e como eles deverão operar em conjunto para atingir os objetivos organizacionais;
e) relatórios de viabilidade técnica, análise de riscos e custo/benefício do projeto;

f) controles preventivos e corretivos, e trilhas de auditoria para os pontos críticos do sistema.

4.3.1.5 Revisão e Aprovação dos Dirigentes

Os departamentos envolvidos e superiores, como o gerente de TI e a área usuária, revisam todos os documentos produzidos para determinar se o projeto é adequado para atender às necessidades organizacionais e/ou de usuários. Confirmada a execução do projeto dos pontos de vista tecnológico e orçamentário, analisando o risco de atrasos ou extrapolação do orçamento é concebida a aprovação para seguir a fase de desenvolvimento e implantação do sistema.

4.3.1.6 Desenvolvimento e Implantação

Nesta fase, a equipe desenvolve, codifica, integra e testa o sistema e avalia sua qualidade, repassando aos usuários para que o testem e consequentemente aprovem ou não sua implantação. O sistema é produzido de acordo com a metodologia de desenvolvimento adotada pela organização e atende às especificações de projeto. Se o sistema for aprovado é implantado, de acordo com planos detalhados de testes, transição, implantação e operação.

A documentação do sistema desenvolvido é apropriada, estando dentro dos padrões adotados pela organização no que diz respeito a *hardware*, *software*, sistema operacional e linguagem de programação.

Esta documentação contém também a descrição lógica, arquivos de Linguagem de Modelagem Unificada, ou seja, *Unified Modeling Language* (UML),

descrição de arquivos, modelo de relatórios e outros itens considerados relevantes para o seu bom entendimento e controle do sistema. Nesta parte, também são preparados manuais de operação, de usuário e de manutenção do sistema, bem como um plano de treinamento dos futuros usuários.

4.3.1.6.1 Teste do Sistema

O teste do sistema é um plano detalhado, compatível com os padrões de teste estabelecidos pela organização e respeitando as responsabilidades definidas para cada parte envolvida (usuários, analistas de sistema, programadores, auditores, controle de qualidade, entre outros).

Os testes são compostos pela a utilização de um número suficiente de condições válidas e inválidas, amostras suficientes de transações e dados para representar as várias atividades e condições que serão encontradas no processamento real. Feito isto, os testes são revistos, documentados e seus resultados analisados e aprovados pela organização e pela área usuária do sistema. As deficiências encontradas são devidamente corrigidas antes de o sistema ser considerado em condições de entrar em operação.

4.3.1.6.2 Implantação

O sistema desenvolvido é colocado em uso somente após a aprovação formal dos usuários e da gerência de desenvolvimento de sistemas. Quando o sistema novo é posto em operação, a documentação é atualizada em relação ao *software*, *hardware*, pessoal de operação e usuários.

4.3.1.7 Revisão de Pós-Implantação

A gerência de desenvolvimento de sistemas verifica o grau de satisfação dos usuários com o sistema implantado e conferi se os requisitos iniciais do usuário foram ou não satisfeitos. Estas revisões feitas após o desenvolvimento e implantação do sistema em um usuário e/ou organização tem por objetivo avaliar o resultado do sistema desenvolvido, do atendimento das necessidades e requisitos dos usuários e do seu grau de satisfação.

4.3.2 Processos na Auditoria de Desenvolvimento de Sistemas

O ponto de controle (PC) tanto pode ser um processo como um resultado dentro da auditoria. Segundo Gil (2000) esta abordagem do PC pode ser exemplificado em: PC - Desenvolvimento do Sistema(DS) e PC- Operação Normal(ON).

4.3.2.1 PC-DS

Este é um ponto de controle de vigência durante o ciclo de desenvolvimento do sistema (GIL, 2000). Pontos de controle existentes nas diversas fases da metodologia de desenvolvimento de sistemas que podem ser validados segundo os enfoques:

a) validação do processo de geração das especificações nas respectivas fases da metodologia, examinando e acompanhando as técnicas aplicadas e os procedimentos seguidos;

b) validação dos resultados gerados em cada fase da metodologia, no tocante ao cumprimento das normas e da qualidade.

4.3.2.2 PC-ON

Caracteriza-se o ponto de controle durante o ciclo de desenvolvimento, mas cujo interesse da validação dar-se-á quando o sistema estiver em operação normal (GIL, 2000). Portanto, o PC-ON é de vigência durante o ciclo de vida do sistema e são PC componentes do sistema aplicativo, segundo suas diversas fases de operacionalização, que vão desde a captação e registro de dados até a apresentação e utilização de informação, validados segundo os enfoques:

a) validação do processo de transformação de dados em informações através da identificação de rotinas operacionais e de controle componentes dos sistemas aplicativos;
b) validação do resultado gerado em cada processamento do sistema aplicativo, pela análise de arquivos gravados e de relatórios/telas emitidos.

4.3.3 Técnicas de Auditoria no Desenvolvimento de Sistemas

A validação dos pontos de controle requer técnicas que agilizem e viabilizem o processo de auditagem. A seguir serão descritas especificamente algumas técnicas conforme Gil (2000), como: análise da metodologia de desenvolvimento de sistema, análise da documentação do desenvolvimento do sistema, a *Base Case System*

Evaluation (BCSE), *Integrated Test Facility* (ITF) e a *System Control Audit Review File* (SCARF).

4.3.3.1 Análise da Metodologia de Desenvolvimento de Sistema

Corresponde à avaliação da metodologia adotada pelo centro de computação e que será utilizado pela equipe de projeto do sistema aplicativo em desenvolvimento. Os procedimentos a serem seguidos para aplicação desta técnica são:

a) entendimento da metodologia através da leitura seus manuais componentes e via esclarecimento junto à gerência de desenvolvedores de sistemas;
b) identificar pontos de controle na metodologia de desenvolvimento, tais como: encadeamento lógico das etapas; objetivos de cada etapa; produtos gerados, responsabilidades, documentação exigida em cada etapa, entre outros;
c) avaliar a adequacidade desses pontos de controles à cultura de informática da empresa em termos de *hardware*, de *software* e de pessoal de computação e usuário;
d) emitir opinião, debater com a equipe de computação e continuar a fazer avaliações na continuidade da aplicação da metodologia a novos sistemas em desenvolvimento.

Esta técnica exige do auditor de sistemas conhecimentos de análise de sistemas e de uma metodologia de desenvolvimento de sistemas.

4.3.3.2 Análise da Documentação do Desenvolvimento do Sistema

O método consiste na avaliação das especificações e/ou construções geradas para o sistema, ao final de cada etapa da metodologia de desenvolvimento de sistemas. As etapas de aplicação da técnica implicam:

a) entendimento das especificações e/ou construções através da leitura da documentação gerada;
b) identificação dos pontos fracos das especificações, a respeito de: objetivos do sistema, análise custo/benefício do novo sistema; levantamento e conclusões do sistemas atual; anteprojeto do sistema; projeto lógico; projeto físico; testes isolados e integrados; programação; implantação; prototipagem; treinamento para institucionalização do novo sistema e documentação geral do projeto;
c) analisar e avaliar os resultados obtidos emitindo o relatório de fraquezas de controle interno, para de imediato, reciclar o projeto, permitindo ao coordenador do projeto atender a requisitos mínimos de controles e eficiência de sistemas.

O auditor de sistemas deverá ter fortes conhecimentos de controles de entrada processamento e saída em sistemas computacionais.

4.3.3.3 Base Case System Evaluation

A *Base Case System Evaluation* (BCSE) é uma técnica semelhante ao *test-deck*, quando uma massa de testes, feita pelo auditor, é submetida ao novo sistema computadorizado, antes de sua implantação, permitindo a avaliação do controle interno

e tendo como conseqüência a emissão de um relatório de fraquezas de controle interno, quando inconveniências são identificadas.

A aplicação desta técnica traz as seguintes vantagens ao novo sistema:

a) a documentação do sistema é geralmente superior;

b) correção de falhas e realização de ajustes são antecipadas;

c) o auditor integra-se com os usuários e os profissionais de computação.

4.3.3.4 Integrated Test Facility

Integrated Test Facility (ITF) corresponde ao desenvolvimento de rotinas de auditoria, dentro dos programas de computador a serem auditados, com o objetivo de separar os dados de teste da auditoria, quando estes forem submetidos aos programas sob auditoria.

Os dados de teste da auditoria submetidos em conjunto com os dados normais são processados pelo programa sob auditoria.

4.3.3.5 System Control Audit Review File

O método *System Control Audit Review File* (SCARF) implica a criação de rotinas específicas de auditoria dentro dos programas de sistema, para selecionar transações reais, segundo condições pré-estabelecidas, gravando um arquivo específico para efeito da auditoria.

O SCARF é uma técnica usada intensamente em sistemas *real-time*, por permitir analisar/auditar a tempo de processamento transações com características particulares, tais como:

a) movimentação de item de estoque ou conta corrente paralisada há determinado período de tempo;
b) saque em conta, com valor muito maior que o saque médio.

Desta forma, finaliza-se a parte que trata especificamente do foco da pesquisa, a Auditoria no Desenvolvimento Sistemas, explicado as etapas, processos e técnicas que podem ser utilizadas. Entretanto, pouco importa os resultados e relatórios obtidos através da auditoria sobre o desenvolvimento de um sistema, se não houver segurança nas informações importantes e confidenciais a respeito de um sistema, seu desenvolvimento e sua equipe.

A partir deste momento entra-se em uma outra área estudada no projeto, a Segurança de Dados. Deste modo, o próximo capítulo abordará sobre a Segurança da Informação, seus objetivos e os mecanismos utilizados para gerar segurança nos dados e informações.

5 SEGURANÇA DA INFORMAÇÃO

Os incidentes de segurança são ocorridos por vários motivos e causas, como: ex-funcionário insatisfeito, vírus de computadores, falhas de *hardware*, sobrecarga elétrica, desastres naturais (incêndio, terremoto, enchente), falhas estruturais, sabotagem, fraudes, acessos não autorizados (*hackers*, espionagem industrial, venda de informações confidenciais para a concorrência), entre outros (DIAS, 2000).

Por fim, a segurança do ambiente computacional deve ser encarada como proteção ao patrimônio da organização e aos investimentos feitos de equipamentos, *software* e pessoal. Os recursos computacionais e as informações sobre a organização, por terem alto valor de mercado, podem e são atrativos para ladrões e espiões. Segurança, portanto, é a proteção de informações, sistemas, recursos e serviços contra desastres, erros e manipulação não autorizadas (invasões), de forma a reduzir a probabilidade e o impacto de incidentes de segurança (DIAS, 2000).

5.1 OBJETIVOS DE SEGURANÇA

Os objetivos de segurança variam de acordo com o tipo do ambiente computacional e a natureza do sistema (administrativos, financeiros, militar). Os objetivos mais almejados por profissionais de auditoria, segundo Dias (2000), são os seguintes:

a) confidencialidade: apenas as pessoas autorizadas podem ter acesso à informação e seus processos. Medidas de controle de acesso e criptografia são utilizadas para garantir esse objetivo;

b) integridade: diz respeito das características da informação: completa, sem alterações, confiável. Evita que dados sejam apagados ou de alguma forma alterados, sem a permissão do proprietário da informação;
c) disponibilidade: o princípio da disponibilidade é respeitado quando a informação está acessível, por pessoas autorizadas, sempre que necessário;
d) consistência: certifica-se de que o sistema atua de acordo com as expectativas dos usuários autorizados;
e) isolamento: regular o acesso ao sistema, o uso legítimo para que não haja o acesso não autorizado;
f) auditoria: proteger os sistemas contra erros e atos maliciosos cometidos por usuários não autorizados. Para identificar os autores e sua ações, são utilizados trilhas de auditoria e *logs*, que registram tudo o que foi executado no sistema, por quem e quando;
g) confiabilidade: garantir que, mesmo em condições adversas, o sistema atuará conforme o esperado.

5.2 MECANISMOS DE SEGURANÇA

Mecanismo de segurança é o meio mais utilizado para atender a um serviço de segurança, isto é, para prover e suportar serviços de segurança. A criptografia é um exemplo de mecanismo que pode ser usado para garantir o serviço de confidencialidade de dados e informações (DIAS, 2000).

Segundo Imoniana (2005) para facilitar a implementação dos controles e mecanismos de segurança, é interessante mapear o ambiente de segurança, ou seja,

ambiente onde os recursos de segurança estão instalados. Isto pode ser presenciado e analisado na Figura 5.

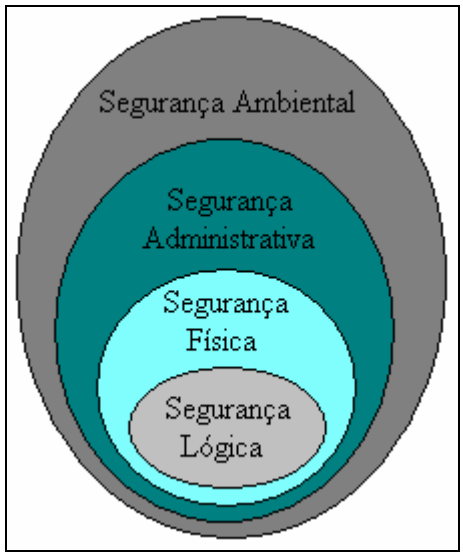


Figura 5. Organização e ambiente de segurança
Fonte: IMONIANA, J. (2005)

Existem vários mecanismos de segurança conforme Dias (2000), como: sistemas criptográficos; assinatura digital; mecanismos de controle de acesso; mecanismos de integridade de dados; mecanismos de disponibilidade; trocas de autenticações; enchimento de tráfego e controles de roteamento.

Porém, nesta pesquisa será abordado apenas o mecanismo de segurança de sistemas criptográficos que será implementada na ferramenta proposta a ser desenvolvida.

5.2.1 Sistemas Criptográficos

Utilizam criptografia ou algoritmos cifrados para proporcionar confidencialidade de dados e de informações de fluxo de dados. Sua vantagem é que, mesmo que outros métodos de proteção de dados (como listas de controle de acesso, permissões de arquivos e senhas, por exemplo) falhem os dados ainda não serão

ininteligíveis ao invasor. Para compreendê-los, o invasor terá de descobrir a chave e o algoritmo utilizados no processo de criptografia (DIAS, 2000).

A criptografia é estudo de métodos para esconder o conteúdo de mensagens ou dados armazenados. O processo de cifragem ou criptação é a transformação da mensagem original em algo ininteligível, utilizando um código secreto – a chave criptográfica. A decifragem ou decriptação, por sua vez, é o processo inverso, onde se tem a recuperação da mensagem original a partir de sua forma cifrada. Os componentes básicos de um sistema criptográfico, mostrado na Figura 6, são: texto claro, algoritmo, chave e texto cifrado (DIAS, 2000).

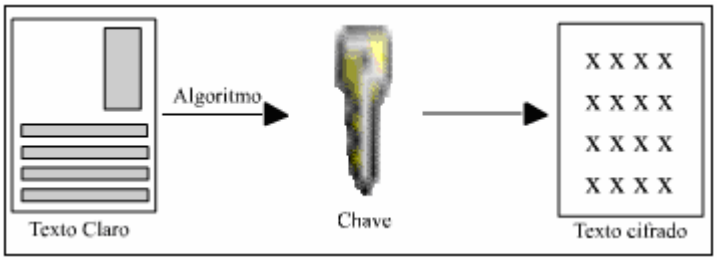


Figura 6. Componentes básicos de um sistema criptográfico
Fonte: DIAS, C. (2000)

O texto claro corresponde á mensagem original. O texto cifrado, também chamado de criptograma, corresponde ao texto claro após ter sido submetido ao processo de criptografia executado por determinado algoritmo. A Figura 7 representa um exemplo de texto claro e um texto cifrado.

Prezado Sr. Presidente,

Em atendimento á sua solicitação, aqui estão relacionados os maiores clientes de nossa empresa e suas respectivias contribuições para nosso excelente faturamento no ano de 1999.

Indústria ABC	R$ 30.000.000,00
Sr. Magnata da Silva	R$ 10.000.000,00

Πρεζαδο Σρ. Πρεσιδεντε,

Εμ ατενδιμεντο (συα σολιχιτα|©ο, αθυι εστ©ο ρελαχιοναδοσ οσ μαιορεσ χλιεντεσ δε νοσσα εμπρεσα ε συασ ρεσπεχτιωασ χοντριβυι|εσ παρα νοσσο εξχελεντε φατυραμεντο νο ανο δε .

Ινδ ⊢τρια

Σρ. Μαγνατα δα Σιλϖα

Figura 7. Texto claro e texto cifrado
Fonte: DIAS, C. (2000)

A chave criptográfica é uma chave secreta utilizada no processo de cifragem e decifragem de mensagens. O algoritmo é uma seqüência de passos e operações matemáticas que transforma o texto claro em texto cifrado, e vice-versa. Os algoritmos de criptografia mais conhecidos são o *Data Encryption Standard* (DES), o *International Data Encryption Algorithm* (IDEA), a família *Message Digest* (MD) e o *Rivest, Shamir - Adleman* (RSA) (DIAS, 2000). Este último algoritmo será detalhado mais a frente, devido a escolha para implementação na ferramenta de auditoria do desenvolvimento de sistemas, ferramenta proposta a ser desenvolvida nesta pesquisa.

Consequentemente, o mecanismo de segurança de sistemas criptográficos proporciona confidencialidade, ao ocultar informações por meio de textos cifrados, e também garante a integridade de dados, pois o conteúdo da mensagem fica inalterado desde a cifragem até a decifragem (DIAS, 2000).

A partir deste momento conclui-se sobre a Segurança da Informação, seus objetivos e mecanismos de segurança e contata-se que os sistemas criptográficos é uma ótima alternativa para garantir a segurança nos dados e informações coletadas e manipuladas na ferramenta deste projeto.

O próximo capítulo tratará dos sistemas criptográficos, ou seja, da Criptografia de Dados, seus tipos e o algoritmo RSA, utilizado para cifrar e decifrar os dados.

6 CRIPTOGRAFIA DE DADOS

Criptografia lida com maneiras pelas qual o significado de mensagens pode ser escondido, de maneira que somente certas pessoas possam entendê-las, bem como com mecanismos que asseguram que o conteúdo dessas mensagens permaneça inalterado (SILVA JÚNIOR, 2001).

Desde seus primórdios, a criptografia baseia-se em dois elementos fundamentais: o algoritmo e a chave. Um algoritmo é uma função matemática que combina texto comum ou outro tipo de informação com uma cadeia de dígitos, chamada de chave, fazendo com que a informação se torne um texto cifrado e, por conseqüência, não legível (SILVA JÚNIOR, 2001).

6.1 CRIPTOGRAFIA SIMÉTRICA

A chave assume papel fundamental neste tipo de criptografia, por ser ela a responsável por cifrar e decifrar uma mensagem. A mensagem quando é cifrada utiliza uma chave, após alcançar o seu destino tem o seu conteúdo decifrado com a mesma chave, tanto o remetente quanto o destinatário compartilham da mesma chave, essa criptografia é chamada de criptografia simétrica. A chave dessa criptografia é única e privada, pois a mesma deve ser mantida em segredo para que continue eficaz (SILVA JÚNIOR, 2001). A Figura 8 representa exatamente o funcionamento da criptografia simétrica.

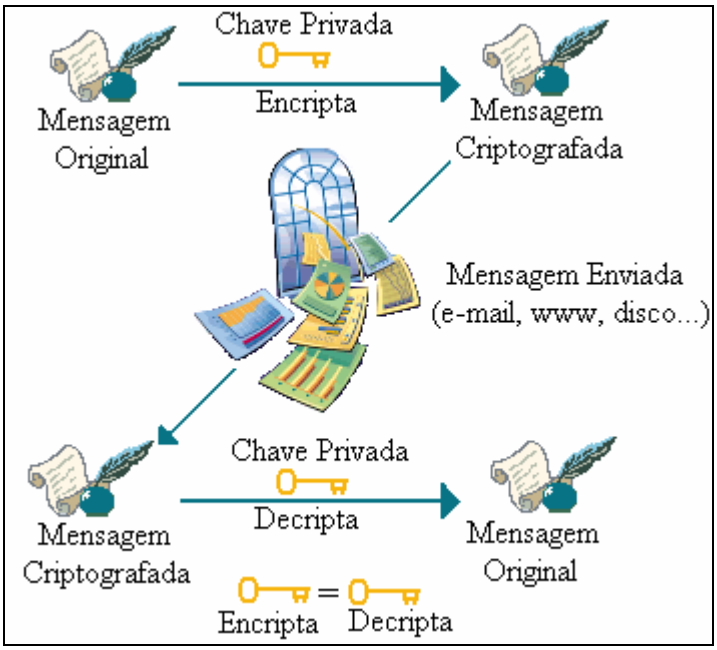


Figura 8. Criptografia Simétrica
Fonte: SILVA JUNIOR, J. (2001)

Este tipo de criptografia tem de se manter uma chave para cada indivíduo com quem se precise ter comunicações codificadas, o que faz com que a manutenção de tais chaves seja bastante trabalhosa e difícil, por conseqüência, não prática.

6.2 CRIPTOGRAFIA ASSIMÉTRICA

No tipo de criptografia assimétrica ou de chave publica um par de chaves é emitido para um individuo, onde somente uma das chaves pode decifrar o que a outra cifrou. Uma chave do par chama-se chave privada e deve ser mantida em segredo (tal como a chave privada da criptografia simétrica), enquanto a outra chave, chamada de chave pública, é divulgada para o público em geral, sempre mantendo sua associação com o individuo para o qual aquele par de chaves foi gerado (SILVA JÚNIOR, 2001).

A vantagem deste modelo de criptografia de chave pública é que ela pode ser utilizada para assegurar confidencialidade e a autoria da mensagem. O funcionamento da criptografia assimétrica é simulado na Figura 9.

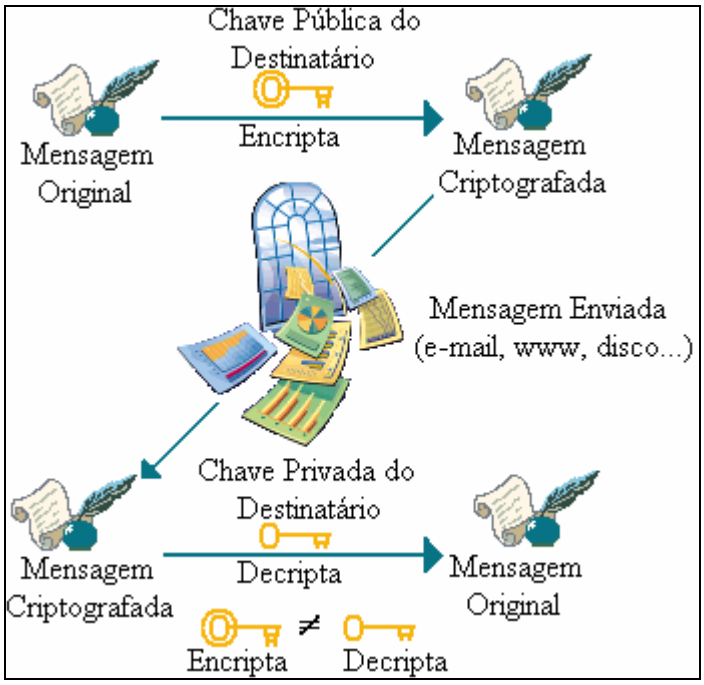


Figura 9. Criptografia Assimétrica
Fonte: SILVA JUNIOR, J. (2001)

6.3 ALGORITMO RSA

O RSA é um sistema de criptografia de chave assimétrica, ou criptografia de chave pública, inventado por volta de 1977 pelos professores *Ronald Rivest, Adi Shamir* e *Leonard Adleman*.

O sistema consiste em gerar uma chave pública (geralmente utilizada para cifrar os dados) e uma chave privada (utilizada para decifrar) por meio de números primos grandes, o que dificulta a obtenção de uma chave a partir da outra (MORENO; PEREIRA; CHIARAMONTE, 2005).

O Algoritmo RSA é muito utilizado e testado, mostrando-se extremamente forte se for adequadamente usado. Isto devido ao fato de que é extremamente difícil fatorar números muito grandes (CARVALHO, 2001).

Quanto maiores os números primos utilizados para a criação da chave, maior a segurança proporcionada por esse algoritmo. A tendência é que o comprimento

da chave aumente cada vez mais, devido, em grande parte, pelo avanço nos sistemas computacionais que acompanham o surgimento de computadores que são capazes de fatorar chaves cada vez maiores em pouquíssimo tempo (MORENO; PEREIRA; CHIARAMONTE, 2005).

6.3.1 Chaves Privada e Pública

Segundo Buchmann (2002) a escolha das chaves começa escolhendo dois números primos aleatórios *p* e *q* e calcula-se o produto. Sejam então:

$$n = p * q$$

n = produto dos números primos p e q

$$\phi = (p - 1) (q - 1)$$

ϕ = produto dos números primos p-1 e q-1

Escolhe-se agora um número aleatório $e > 1$ tal que o máximo divisor comum dos dois inteiros positivos – MDC (e , ϕ) = 1, e e ϕ devem ser primos entre si. Calcula-se então o número d ($1 < d < \phi$) tal que:

$$(e * d) \text{ Módulo } \phi = 1$$

e = número primo menor que ϕ e maior que 1 de modo que mdc(e, ϕ) = 1

d = número menor que ϕ e d($1<d<\phi$) tal que (e*d) mod ϕ 1

Dessa forma é feito o produto de *e* com *d*, em seguida o modulo do resultado.

A chave pública se constitui dos números [*e* , *n*], e a chave privada dos números [*d* , *n*]. Para garantir a força do algoritmo, os números *p* e *q* devem ser aleatórios e, tipicamente, com mais de 100 algarismos decimais (CARVALHO,2001).

6.3.2 Cifrando e Decifrando

Cifrar uma mensagem é o processo de transformação do texto claro da mensagem em texto cifrado, ou seja, texto criptografado. Isto se faz possível através da chave pública utilizada para cifrar dados (BUCHMANN, 2002). Dessa forma, uma mensagem *w*, para se cifrar *w* em *c* faz-se*:*

$$c = w^{e} \text{ Módulo } n$$

$$c = (w \text{ exp } e) \text{ mod } n$$

Entretanto a decifragem se torna o processo inverso, ou seja, é o processo de transformação do texto cifrado ou criptografado em texto claro da mensagem. Isto se faz admissível por meio da chave privada empregada para decifrar dados (BUCHMANN, 2002). Para decifrar faz-se:

$$w = c^{d} \text{ Módulo } n$$

$$w = (c \text{ exp } d) \text{ mod } n$$

Com o término desse capítulo de Criptografia de Dados, mostrando os tipos de criptografia existentes – Assimétrica e Simétrica, e o Algoritmo RSA – Chave Pública e Privada, Cifrando e Decifrando, está pendente apenas os trabalhos Correlatos e o Desenvolvido para a conclusão da fundamentação téorica existente neste projeto.

O próximo capítulo apresenta alguns Trabalhos Correlatos que tenham alguma relação com a pesquisa, porém são poucas ou inexistentes as ferramentas ou projetos específicos que trabalhem exatamente na área de Auditoria no Desenvolvimento de Sistemas, na Engenharia de *Software*.

7 TRABALHOS CORRELATOS

O aumento desenfreado das aplicações de computadores em todas as áreas operacionais de negócios resulta numa cobrança para agilidade nos setores de serviços da qual não se pode excluir a auditoria, gerando a necessidade de se implementarem as metodologias de auditoria que apliquem tecnologia de informações (IMONIANA, 2005).

O *software* de auditoria consiste em programas aplicativos cujo objetivo é incrementar a qualidade do trabalho do auditor e conseqüentemente dos sistemas em análise, tornando assim o serviço de auditoria interativo com os recursos (IMONIANA, 2005).

Atualmente os pacotes disponíveis no mercado atendem quase que exclusivamente as necessidades de auditoria de sistemas desenvolvidos ou em operação, de aplicativos prontos. São poucas, quase desconhecidas as ferramentas que tratam exatamente da auditoria no desenvolvimento de sistemas de informação.

O que existem nos dias de hoje para auxiliarem no desenvolvimento e construção de sistemas, na engenharia de *software* são ferramentas *Computer Aided Software Engineering* (*CASE)* e *Softwares* de Reengenharia.

As ferramentas *CASE* e os *softwares* de Reengenharia apresentam objetivos e vantagens de uso semelhantes entre si, que poderia ser globalizado como ferramentas e *softwares* que têm por objetivo aumentar a capacidade das pessoas de garantir a qualidade e a facilidade com a continua melhoria do processo de desenvolvimento.

CASE é uma ferramenta ou conjunto de ferramentas que automatizam tarefas que compõem o processo de desenvolvimento de *software*, ou seja, um sistema computacional composto de ferramentas que suportam a automação do ciclo de vida do

software e permite o uso efetivo dos princípios e práticas gerais do desenvolvimento de sistemas.

Reengenharia de s*oftware* é qualquer atividade que prepare ou melhore o entendimento e o *software* em si, aumentando sua manutenção, seu reuso e sua extensão. A reengenharia pode ser definida como o exame e a alteração de um sistema para reconstituí-lo de uma nova forma, seguida pela sua implementação ou também pelo processo de modificação de mecanismos internos ou estruturas de dados de um sistema ou programa sem mudar sua funcionalidade (YAMADA, 1997).

7.1 PACOTES E FERRAMENTAS DE AUDITORIA DE SISTEMAS

Destacam-se segundo Imoniana (2005) entre os sistemas e ferramentas existentes no mercado que auxiliam o auditor de sistemas nos atendimentos as necessidades: *Audit Automation Facililities* (AAF); Sistema de Auditoria Interna – Audin; Sistema AUDITAR e o Sistema SAP.

Entretanto é importante enfatizar que os *softwares* mencionados a cima possuem funções de gerenciamento de auditoria, como planejamento e controle, e com isso auditam resultados de sistemas de informações, de aplicativos já desenvolvidos. Dessa forma, torna-se desconhecido algum *software* ou ferramenta, produzida por uma empresa, trabalho acadêmico, entre outros, que seja direcionada a auditorar do desenvolvimento de sistemas.

A seguir, os sistemas e ferramentas existentes atualmente que auxiliam o trabalho da Auditoria de Sistemas, segundo Imoniana (2005):

7.1.1 Audit Automation Facililities

Audit Automation Facilities (AAF), de propriedade da WJ informática, é um programa de gestão de auditoria interna, que visa o aumento da eficiência e da produtividade dos processos com redução de custos.

O AAF padroniza ações, integra equipes, automatiza processos de cobranças, gera gráficos e relatórios, disponibiliza os dados com total segurança, permitindo inserção, documentação, verificação e acompanhamento dos controles internos.

Suas principais vantagens segundo Imoniana (2005) são:

a) padroniza o processo de auditoria, uniformizando programas, testes, relatórios, etc.;
b) diminui o tempo de adaptação dos novos auditores;
c) integra a equipe, possibilitando que um trabalho seja partilhado entre os demais componentes;
d) diminui significativamente o tempo no preparo dos programas de trabalho, testes, papéis de trabalho padrão;
e) permite a revisão interativa dos trabalhos em andamento;
f) elimina a necessidade do papel;
g) controla custos de horas diretas e indiretas por auditoria
h) permite acesso aos dados relevantes das auditorias encerradas;
i) disponibiliza aos auditados acesso exclusivo aos seus relatórios com total seguramça;
j) disponibiliza diariamente informações sumarizadas de todas as auditorias em andamento;

k) geração de gráficos resultantes do cruzamento de informações, como trabalho realizados com planejados, pontos com unidades auditadas, cobertura com riscos, etc.

7.1.2 Sistema de Auditoria Interna – Audin

Software desenvolvido pela Empresa de Processamento de Dados da Previdência Social – Dataprev, objetivando a melhoria dos seus trabalhos de auditoria interna. O Audin auxilia as equipes envolvidas nas auditorias a realizarem os planejamentos e executarem suas tarefas, permitindo aos gerentes acompanhar os trabalhos, emitir comentários e aprovar ou não os documentos gerados pela equipe.

O sistema pode funcionar isoladamente ou em um computador com uma rede local com as seguintes características segundo Imoniana (2005):

a) um conjunto de tabelas;

b) um conjunto de modelos de programas de auditorias;

c) manual operacional;

d) os documentos de cada auditoria específica, reunidas na base de auditoria.

O sistema Audin foi desenvolvido no *software Lotus Notes*, o qual apóia a sua utilização, sendo usado como mecanismo de comunicação entre a equipe e a gerencia através de correio eletrônico. Os trabalhos podem ser executados remotamente em computadores portáteis, fazendo-se a replicação ao computador central onde ficam armazenadas as diversas bases de documentos.

7.1.3 Sistema AUDITAR

Sistema Integrado de Auditoria da Gestão Pública constituído por um *software* operacional voltado para as atividades típicas de auditoria no Sistema de Controle Interno do Poder Executivo Federal.

O aplicativo cobre rigorosamente todas as fases básicas do processo auditorial segundo Imoniana (2005), como:

a) pré-auditoria: banco de dados que compões os elementos referenciais de qualidades, desempenhos e criticidades;
b) planejamento: módulo destinado à composição dos papéis de planejamento;
c) execução dos exames: módulo destinado ao registro dos exames de campo;
d) comunicação de resultados: módulo-produto de onde são extraídos os relatórios de auditoria;
e) monitoramento dos resultados: módulo destinado ao registro das auto-avaliações dos trabalhos realizados, assim como das avaliações de resultados das auditorias.

Entre os objetivos do sistema segundo Imoniana (2005), destacam-se:

a) otimização das ações de auditoria da gestão pública: assumir a informatização do conjunto complexo de ações auditorias da Gestão Pública Federal, tendo em conta as inconstâncias, incertezas e descontinuidades;

b) integrar as ações de auditoria: devido as atividades típicas de auditoria estarem disseminadas, em suas diferentes etapas, pelos diversos setores componentes de uma unidade de controle;

c) estar presente nas ações típicas dos profissionais de auditoria de gestão pública: o sistema se propõe a participar nas atitudes técnicas, que devem ser assumidas antes, durante e após os exames e as verificações de campo.

7.1.4 Sistema SAP

A empresa alemã *Systeme, Anwendungand Programme* (SAP), desenvolveu um pacote de gestão que representa um conjunto de módulos com diversas aplicações de negócio, inserida no *Enterprise Resource Planning* (ERP), conhecido como planejamento dos recursos internos das empresas.

Os módulos são integrados e contêm a maior parte das funcionalidades necessárias às grandes corporações, incluindo manufatura, finanças, vendas e distribuição e recursos humanos.

O sistema possui ferramenta específica de auditoria chamada *Audit Information System* (AIS) que proporciona ao auditor relatórios e transações. Estes permitem ao auditor realizar relatórios e trabalhos de auditoria em sistemas, auditoria financeira e auditoria de processos.

8 SISTEMA DESENVOLVIDO - AUDISOFT

Neste projeto foi desenvolvida uma aplicação, chamada de AudiSoft, com o propósito de auxiliar e facilitar a Auditoria no Desenvolvimento de Sistemas, com a utilização de criptografia de dados (RSA) para maior segurança nos dados coletados manipulados pelo Auditor.

A ferramenta AudiSoft serve para analisar passo a passo as etapas e processos existentes na engenharia de software, no desenvolvimento de sistemas, sendo possível anotar evidências e achados em cada etapa e também propor um solução, uma recomendação sobre o que foi encontrado. A segurança de dados entra no momento em que o auditor vai gerar o relatório com todos os pontos abordados no AudiSoft a respeito sobre o desenvolvimento do sistema analisado, pois neste relatório estarão informações importantes e que não podem ter livre acesso. Para isso a criptografia juntamente com o algoritmo RSA é usada para cifrar os dados e somente ser decifrado por pessoas autorizadas, como auditores e usuários autorizados por eles, como o responsável ou proprietário da engenharia de software que foi analisada.

A linguagem empregada para a implementação do AudiSoft foi o C++ Orientado a Objetos com o auxilio da ferramenta de desenvolvimento *Borland C++Builder 6*. Essa linguagem foi escolhida por ser estudada em meio acadêmico, podendo assim ter seu código fonte estudado ou até mesmo modificado por outros acadêmicos.

O Banco de Dados utilizado foi *Firebird 1.5 Server Manager*, um banco de dados gratuito e que correspondeu perfeitamente com as necessidades da ferramenta implementada neste projeto e o *EMS QuickDesk 2.0*, ferramenta para administração *InterBase/FireBird* e que permite editar e criar todos objetos de um banco. Além disso,

foi necessária a instalação do *BDE Administrator*, que atua como interface para o BDE (fornecedor da capacidade de acesso padronizado a banco de dados para ambientes de programação da *Borland*) e é através deste aplicativo que serão configuradas todas as propriedades para conectar o *C++ Builder 6* com uma base de dados qualquer. Também foi necessário o *Firebird ODBC Driver* que contém o *driver* do *Firebird* para acessá-lo via BDE.

Na criação do arquivo de Ajuda do AudiSoft foi empregado o uso da ferramenta usada para editar e compilar arquivos de ajuda, o utilitário - *Microsoft Help WorkShop*. Para a criação do arquivo somente com texto foi utilizado o *Help Workshop* e para inserção de figuras o *Dialog Box Help Editor*.

Para modelagem da ferramenta AudiSoft foi utilizado o *Pacestar UML Diagrammer*, um software que ajuda na criação de vários diagramas UML e o *DBDesigner4*, um programa que auxilia e cria modelagens de banco de dados.

Por fim, para a criação do instalador do AudiSoft foi utilizado o *Inno Setup*, um software que cria pacotes de instalação para programas juntamente com o *ISTool*, um programa que auxilia na criação de scripts para o *Inno Setup Compiler*.

8.1 METODOLOGIA

Foram realizadas as seguintes etapas para a elaboração do projeto de pesquisa:

a) **levantamento bibliográfico:** foram pesquisados materiais na área de auditoria, engenharia e desenvolvimento de softwares e segurança de dados;

b) **estudo da auditoria:** capítulo 2, foi explicado a auditoria em âmbito geral, sobre suas fases, princípios, conceitos e natureza;

c) **estudo da auditoria de sistemas:** capítulo 3, foi relatado da auditoria de sistemas, suas ferramentas e técnicas;

d) **estudo do desenvolvimento de sistemas e sua auditoria:** capítulo 4, foi estudado sobre a tecnologia e sistemas da informação juntamente com as etapas e processos de desenvolvimento de sistemas (engenharia de software) e sua auditoria;

e) **estudo da segurança da informação:** capítulo 5, foi explicado sobre segurança de dados juntamente com os objetivos e mecanismos de segurança;

f) **estudo da criptografia de dados:** capítulo 6, compreensão sobre sistemas criptográficos, tipos de criptografias e o algoritmo RSA;

g) **modelagem da interface:** para modelar os processos da ferramenta AudiSoft foi utilizado os diagramas de *Use Case*, de atividades e de classes feito com o auxilio do *Pacestar UML Diagrammer* e diagramas de entidade e relacionamento feitos com o *DBDesigner4*;

h) **implementação da interface:** a implementação foi desenvolvida em linguagem C++ orientada a objetos com o auxilio do ambiente de desenvolvimento *Borland C++Builder 6*;

i) **criação de banco de dados:** a aplicação foi desenvolvida utilizando o banco de dados *Firebird 1.5 Server Manager* e o *EMS QuickDesk 2.0* para editar e criar os objetos do banco;

j) **relacionamento do banco de dados com implementação:** através do aplicativo *BDE Administrator*, são configuradas todas as propriedades

para conectar o *C++ Builder 6* com uma base de dados qualquer. Além disso, foi necessário o *Firebird ODBC Driver* que contém o *driver* do Banco *Firebird* para acessá-lo via *BDE Administrator*;

k) **arquivo de ajuda:** o AudiSoft possui um arquivo de ajuda feito com *Help Workshop* e *Dialog Box Help Editor* ambos da *Microsoft Help WorkShop*;

l) **testes do sistema desenvolvido:** ao final da implementação, foi necessário fazer vários testes;

m) **correções na implementação:** após os testes, algumas correções foram feitas, assim como melhorias no *software*;

n) **instalador do projeto:** após os testes e correções finais foi criado um instalador para a ferramenta AudiSoft com o auxílio do software *Inno Setup Compiler* e o *ISTool*.

8.2 MODELAGEM

Com a finalidade de atingir os objetivos deste projeto de pesquisa, foram necessárias algumas etapas metodológicas, que foram desenvolvidas ao longo do mesmo.

Durante toda pesquisa se fez necessário um levantamento bibliográfico, sobre os assuntos abordados na Metodologia, como auditoria, auditoria de sistemas e do desenvolvimento de sistemas, engenharia de software, segurança de dados, criptografia de dados e o algoritmo RSA. Nessa etapa também foram realizados estudos sobre modelagem de sistemas, principalmente em UML, e desenvolvimento de software em ambiente de programação *C++ Builder 6* com Banco de Dados *Firebird*. O

levantamento bibliográfico foi realizado com pesquisas em livros, artigos, trabalhos de conclusão de curso, entre outros.

A implementação foi iniciada com a modelagem da ferramenta AudiSoft, principalmente UML. Nessa etapa, a implementação deste projeto foi modelado em Linguagem Universal de Modelagem, onde foram obtidos alguns diagramas a seguir.

8.2.1 Diagramas de Caso de Uso

Na Figura 10 pode-se observar um diagrama de caso de uso do ponto de vista do Auditor, administrador que comanda e tem todas as atividades do AudiSoft ao seu dispor.

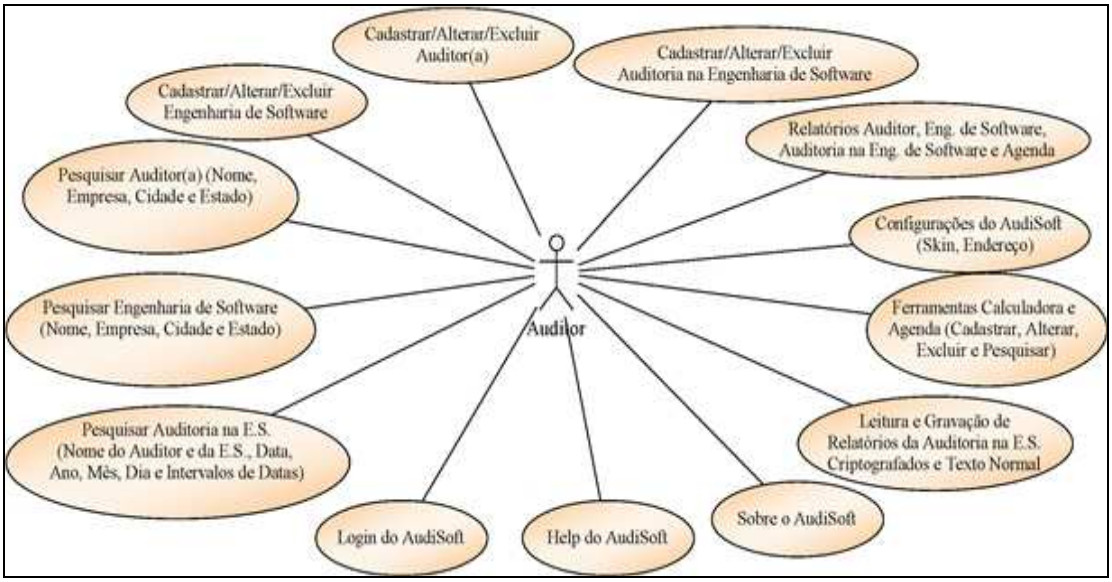


Figura 10. Diagrama de Caso de Uso (Auditor)

Entretanto, na Figura 11 pode-se ver o diagrama de caso de uso do ponto de vista do cliente - proprietário ou responsável pela engenharia de software analisado pelo AudiSoft. Este têm apenas as opções de acesso ao ajuda (*help)* e o sobre do AudiSoft, e de receber o relatório cifrado do auditor junto com a sua chave particular, para

conseguir ter acesso a leitura e gravação dos dados cifrados e decifrados da auditoria na sua engenharia de software.

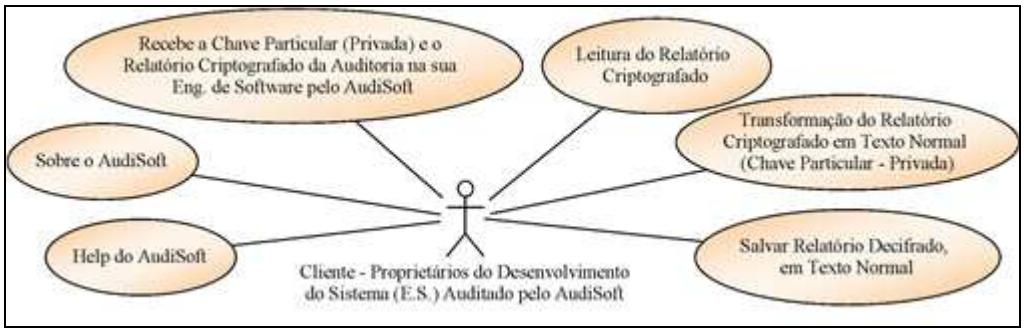


Figura 11. Diagrama de Caso de Uso (Cliente – proprietário da ES)

8.2.2 Diagramas de Atividades

A seguir, na Figura 12, é mostrado um modelo de diagrama de atividades que representa o cadastro de uma auditoria na engenha de software pela ferramenta AudiSoft.

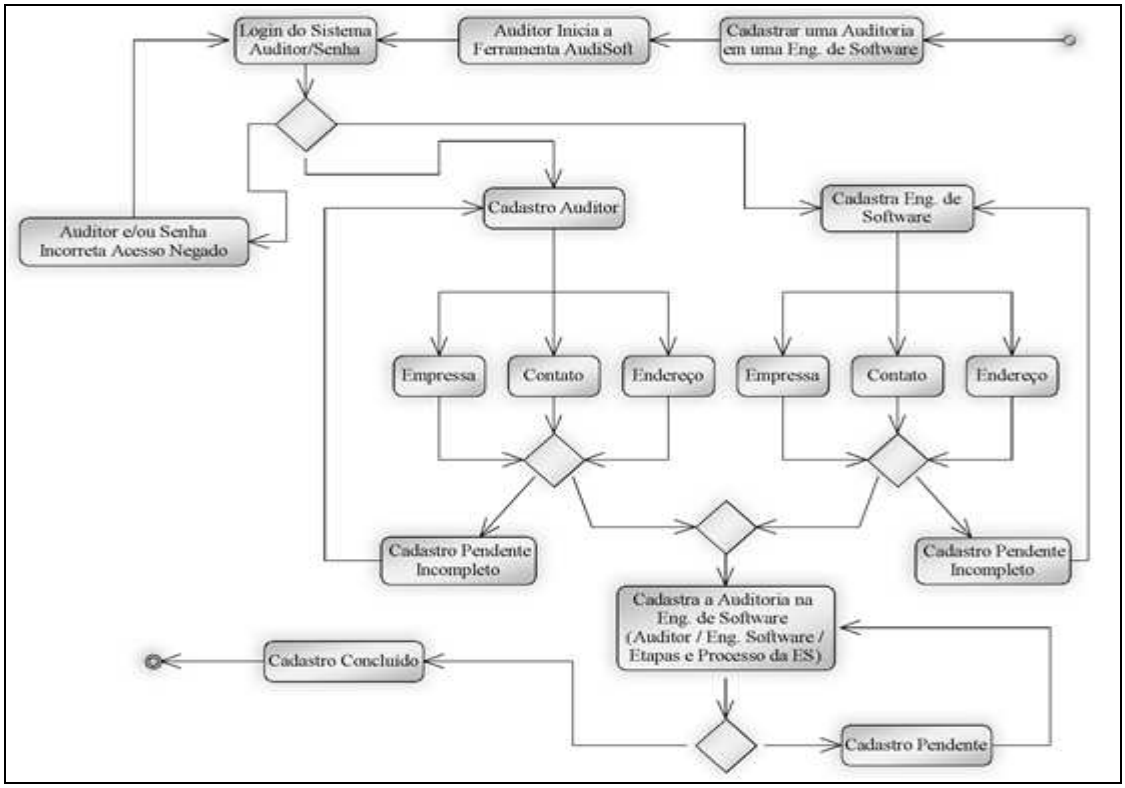


Figura 12. Diagrama de Atividade (Cadastrar uma auditoria na ES)

Na Figura 13, é apontado um outro diagrama de atividades que simula o cliente, proprietário ou responsável, decifrando o resultado (relatório criptografado) da Auditoria na sua Eng. de Software pela ferramenta AudiSoft.

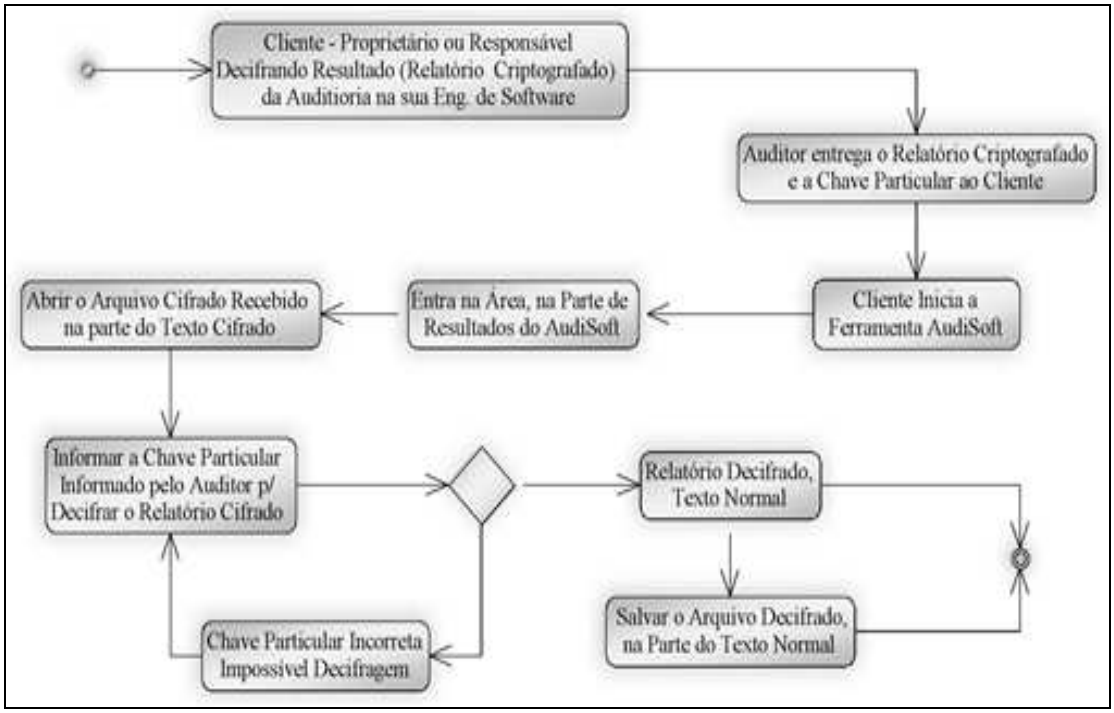


Figura 13. Diagrama de Atividade (Cliente decifrando relatório com chave)

8.2.3 Diagramas de Entidade e Relacionamento

Na Figura 14, é representado um diagrama de entidade e relacionamento, onde as entidades Auditor e Eng. de Software se relacionam com demais entidades, como Endereço (País, Estado, Cidade, Bairro e Endereço), Empresa e Contato, pertencente ao sistema AudiSoft.

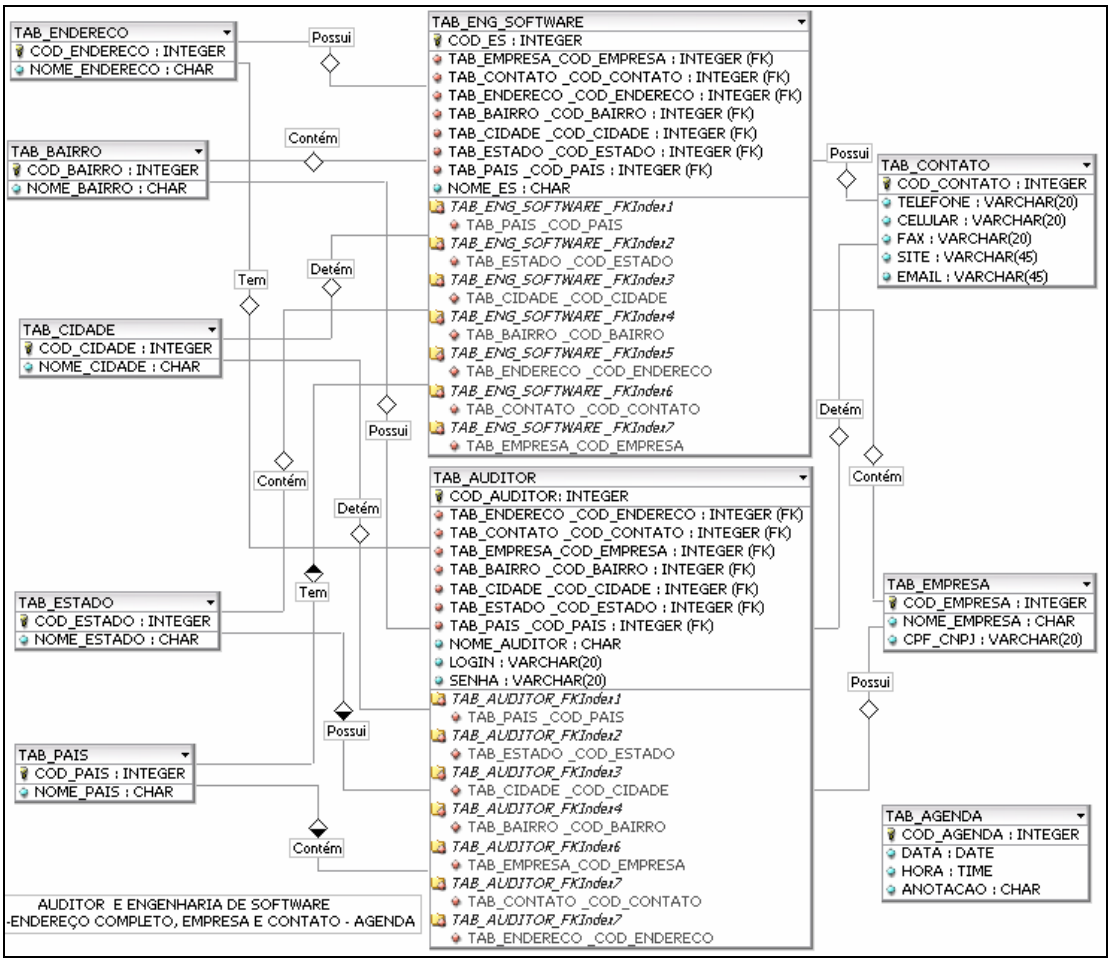


Figura 14. Entidade e Relacionamento (Entidades Auditor e ES e relacionamentos)

O diagrama de entidade e relacionamento representado na Figura 15, trata da entidade Auditoria na Eng. de Software (ES) se relacionando com as demais entidades, que são as etapas e processos contidos no desenvolvimento de sistemas, como: Viabilidade, Levantamento de Dados, Planejamento Estratégico, Análise de Requisitos, Diagramas, Dicionário de Dados, Implementação, Testes, Interfaces, Implantação, Treinamento, Avaliação e Projeto de Custo.

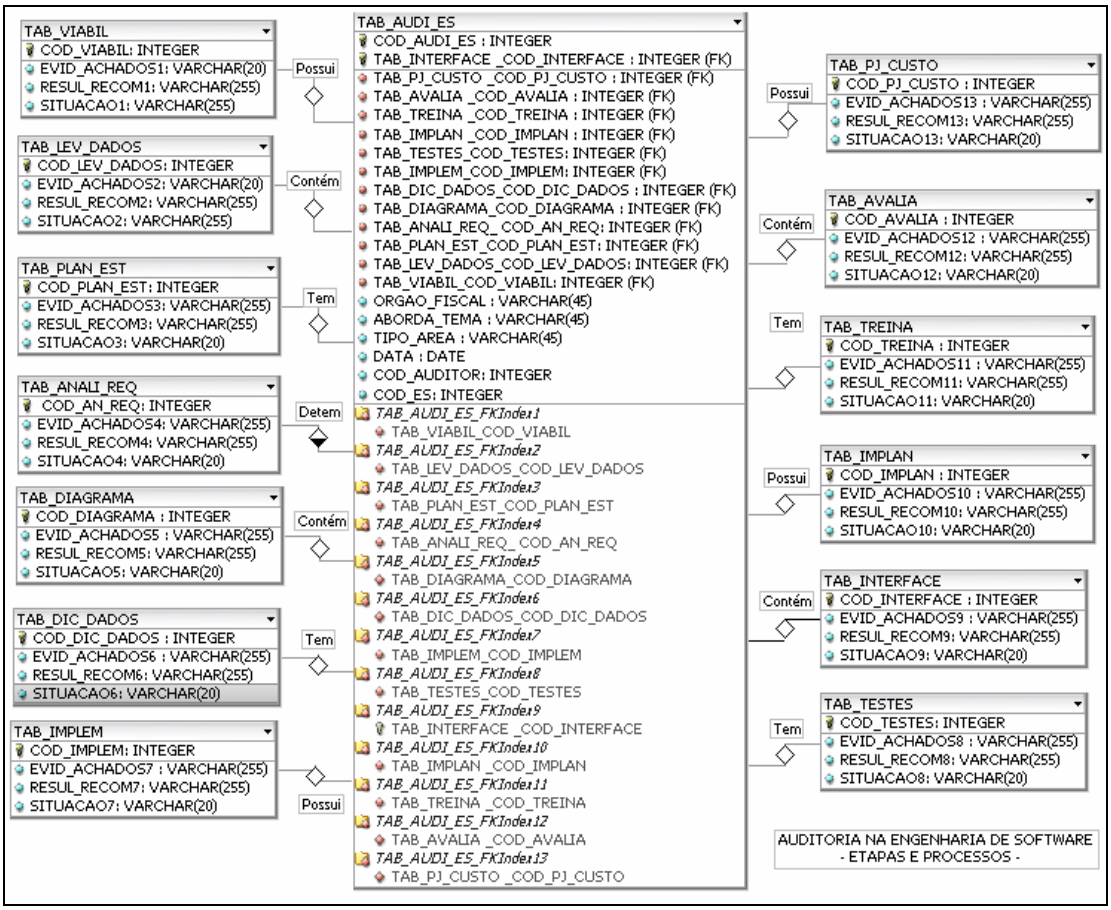


Figura 15. Entidade e Relacionamento (Auditoria na ES e relacionamentos)

Por fim, na Figura 16, representa-se o diagrama de entidade e relacionamento entre as entidades Auditor, Eng. de Software e a Auditoria na ES e o relacionamento entre ambas.

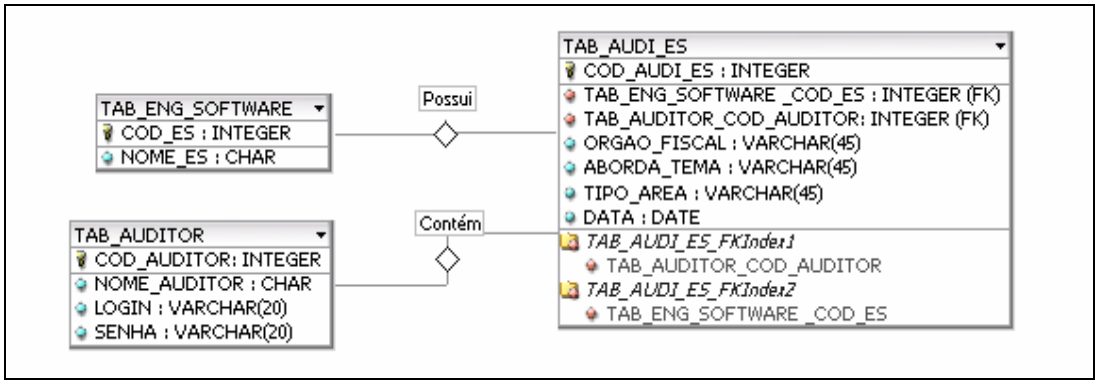


Figura 16. Entidade e Relacionamento (Auditor e ES relacionamento com Auditoria)

8.2.4 Diagrama de Classes

O diagrama de classes exibido na Figura 17 ilustra todas as classes, métodos e relacionamentos existente no banco de dados da ferramenta AudiSoft.

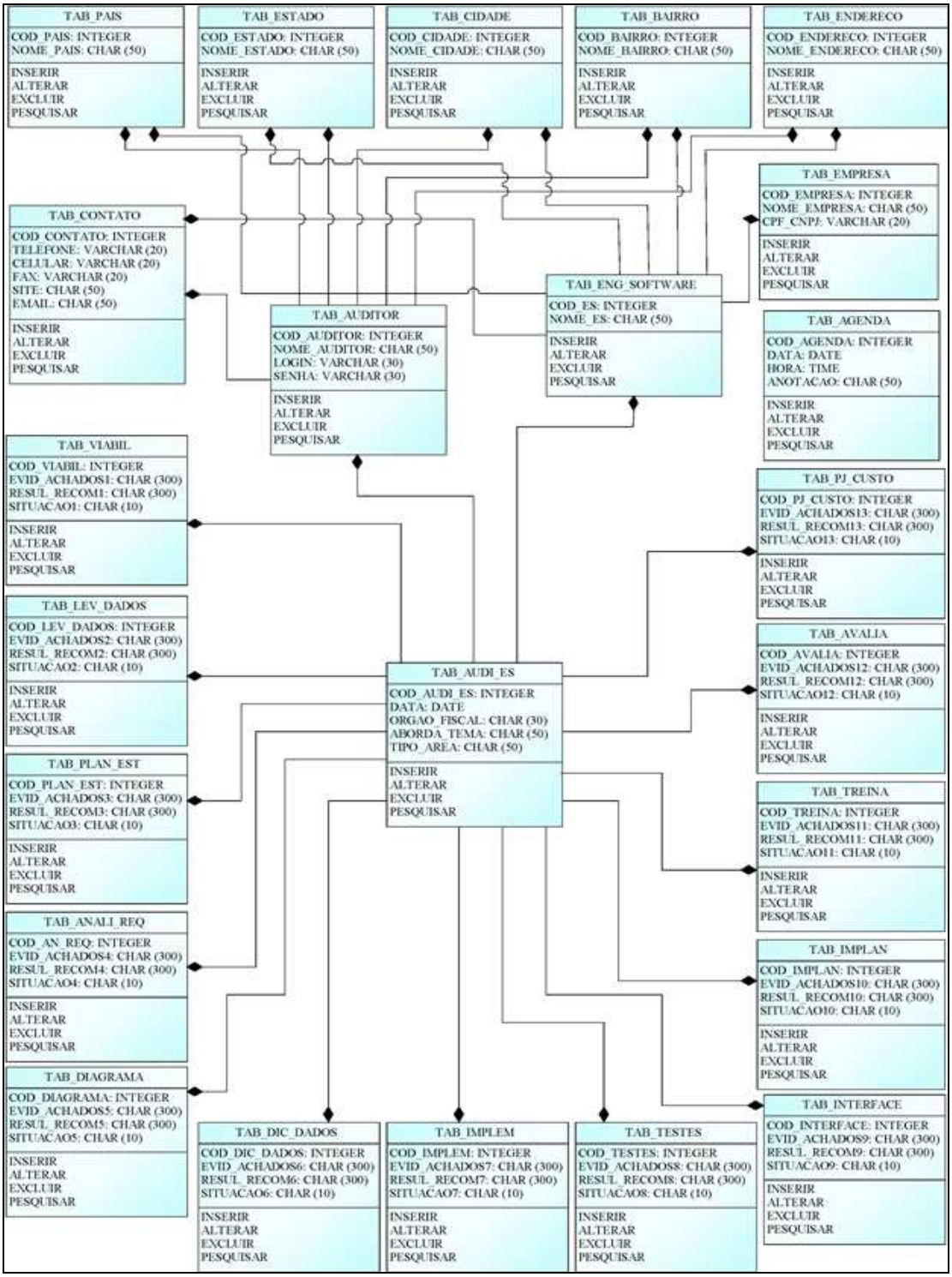


Figura 17. Diagrama de Classes do AudiSoft

8.3 COMPOSIÇÃO E FUNCIONAMENTO DO AUDISOFT

O AudiSoft, ferramenta para auditoria em engenharia de software com criptografia de dados, tem seu funcionamento e composição dividido conforme Figura 18, em:

a) **inicio:** *login* do AudiSoft, configurações, ajuda e sair;

b) **cadastro:** cadastro do auditor e da engenharia de software (desenvolvimento do sistema);

c) **ação:** cadastro da auditoria na engenharia de software;

d) **pesquisa:** consulta auditor, engenharia de software e auditoria na ES;

e) **relatório:** relatórios sobre o auditor, a engenheria de software, a auditoria na ES e as anotações da agenda do auditor;

f) **resultados**: geração dos resultados da auditoria na ES, podendo ser feita a cifragem e decifragem e a leitura e gravação dos relatórios;

g) **ferramentas:** calculadora do Windows e agenda de anotações do auditor;

h) **sobre e sair:** o primeiro contém dados do AudiSoft, do desenvolvedor, informações do computador onde está instalada a ferramenta e créditos, o segundo fecha a ferramenta AudiSoft.

Figura 18. Interface de Entrada do AudiSoft

8.3.1 Inicialização do AudiSoft

A ferramenta de auditoria AudiSoft foi desenvolvida para atender a dois tipos de usuários: o cliente e o auditor.

O primeiro, o cliente é o proprietário ou responsável pela engenharia de software que teve seu conteúdo auditado pelo AudiSoft. Este usuário terá acesso limitado, pois o mesmo não conseguirá fazer *login* da ferramenta, tendo acesso apenas ao arquivo de ajuda, aos resultados, para conseguir decifrar o relatório de sua ES com sua chave particular fornecida pelo auditor responsável e no sobre da ferramenta.

Entretanto, o segundo usuário é o auditor, onde este tem acesso ilimitado a ferramenta, através de seu *login* e senha, conforme Figura 19, conseguindo ter autorização a todos os recursos da implementação.

Figura 19. Interface de *Login* (Auditor e Senha)

Após a realização do *login* do sistema novas opções serão ativadas para o auditor, como a opção de configurações, conforme Figura 20, onde podem ser realizadas trocas de *skins* e aparências da ferramenta AudiSoft e também ter acesso a cadastros, como o de país, estado, cidade, bairro e endereço utilizados no cadastro do auditor e da engenharia de software e pode-los inserir, alterar, remover e consultar.

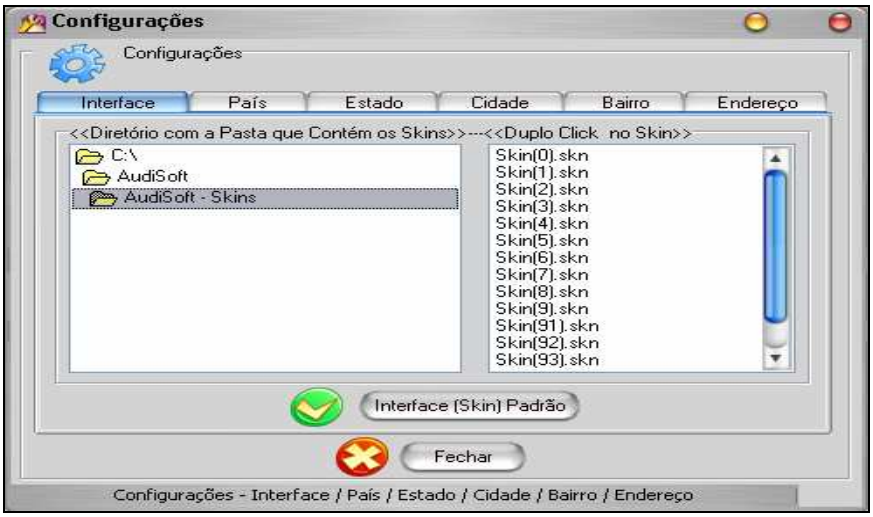


Figura 20. Interface de Configurações

Ainda na parte inicial do programa existe um atalho para a chamada do arquivo de ajuda feito para o AudiSoft, mais especificamente para esclarecer palavras e termos técnicos utilizados na ferramenta e na área de auditoria. Este arquivo de ajuda, visualizado na Figura 21, ainda contém uma lista de etapas e processos envolvidos na engenharia de software, com objetivos e palavras chaves, detalhado no APÊNDICE A, para facilitar o trabalho do auditor no momento de aplicar a auditoria na ES.

Figura 21. Interface do Arquivo de Ajuda

8.3.2 Auditor e Engenharia de Software

A atividade relacionada ao cadastro do auditor e da engenharia de software, permitida apenas ao auditor depois de efetuar o *login* do AudiSoft, é o primeiro passo para se conseguir realizar uma auditoria na ES, ou seja, necessita-se de um auditor responsável e apto a auditar e uma engenharia de software ou sistema em desenvolvimento pronto a ser auditada.

Ambos os cadastros, tanto do auditor como o da engenharia de software, são similares entre si, possuindo as opções de inserir, alterar, excluir e pesquisar informações referentes a dados pessoais, de endereço (localização), profissionais e de contato, conforme a Figura 22. A única diferença é que somente o cadastro do auditor possui informações referentes a dados de acesso (*login* e senha), para que o mesmo daquele momento em diante possa fazer o login da ferramenta AudiSoft e ter autorização a todos os recursos da implementação.

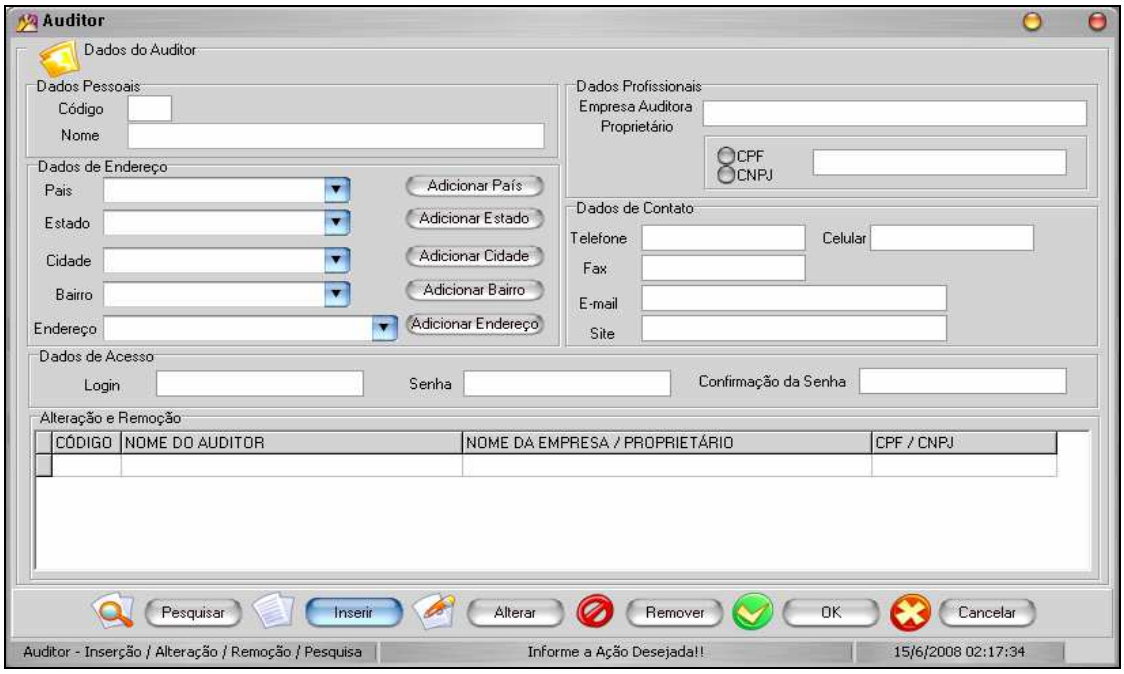


Figura 22. Interface do Auditor

8.3.3 Ação da Auditoria na Engenharia de Software

A ação de auditar a engenharia de software, permitida apenas ao auditor, é o foco principal desta pesquisa. Neste cadastro, são identificados dados, como os principais: a engenharia de software a ser auditada e o auditor responsável, e em seguida vão sendo preenchidos os papéis de trabalho da auditoria, com evidências, achados, controles, irregularidades, resultados, recomendações, pontos fortes e fracos, melhorias, sugestões e situação de cada etapa e processo envolvido na engenharia de software.

As etapas e processos, visíveis no APÊNDICE A, da engenharia de software (viabilidade, levantamento de dados, planejamento estratégico, análise de requisitos, diagramas, dicionário de dados, implementação, testes, interfaces, implantação, treinamento, avaliação e projeto de custo) e os termos próprios da auditoria (órgão fiscalizador, abordagem do tema, tipo ou área envolvida, evidência e achados da auditoria, os tipos de controles, irregularidades e resultados e recomendações da auditoria) são explicados passo a passo no arquivo de ajuda, dessa forma existem vários botões de ajuda que ao serem pressionados trazem todas as informações necessárias ao auditor sobre o tópico relacionado, deixando o trabalho do auditor dessa forma mais eficaz, tranqüilo e completo.

O cadastro é iniciado informando o auditor e a engenharia de software a ser avaliada, em seguida de algumas informações complementares de auditoria, até o ponto de chegar a primeira etapa da engenharia de software. Posteriormente, toda a etapa ou processo possui um botão ajuda e avançar, onde este último vai passando de uma etapa para outra. Se em algum momento for encontrada alguma evidência ou achado que comprometa a engenharia de software, a auditoria pode ser finalizada e concluída, não precisando auditar todas as etapas e processos do desenvolvimento de sistemas.

Este cadastro possui as operações normais de inserir, alterar, excluir e pesquisar informações referentes a auditoria executada em uma engenharia de software por um determinado auditor, conforme visualizado na Figura 23.

Contudo, percebe-se que é um trabalho árduo, que exige muita atenção, raciocínio e estudo por parte do auditor que está executando a auditoria.

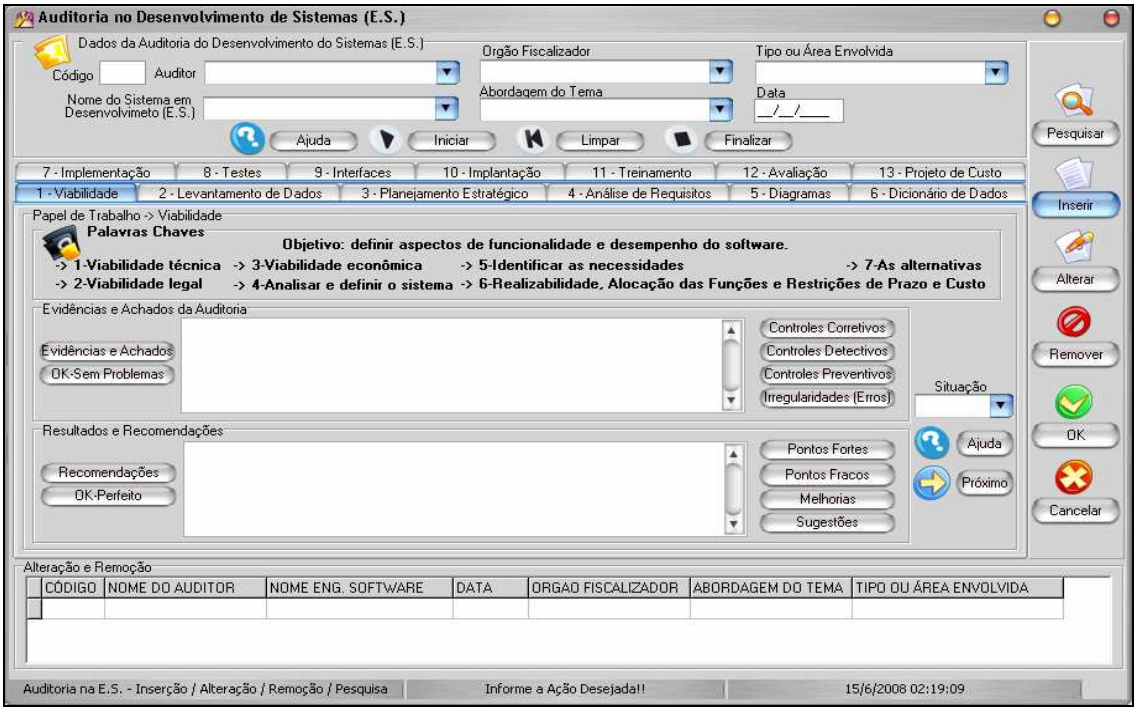


Figura 23. Interface da Auditoria na Engenharia de Software (ES)

8.3.4 Pesquisas e Relatórios no AudiSoft

A ferramenta de auditoria AudiSoft possui pesquisas e relatórios voltadas somente ao auditor, a fim de descobrir ou estabelecer fatos ou princípios relativos a um campo qualquer do conhecimento, indagando a busca minuciosa para averiguação da realidade que está ocorrendo no sistema, e dessa forma deixar o auditor a par de tudo.

Na área de pesquisas, existem buscas e consultas no auditor, na engenharia de software e na auditoria da ES. Os tipos de pesquisas são:

a) **auditor:** busca por nome do auditor, empresa ou proprietário, por estado e cidade, conforme a figura 24;

Figura 24. Interface de Pesquisa do Auditor

b) **engenharia de software:** busca por nome da engenharia de software, empresa ou proprietário, por estado e cidade, semelhante a pesquisa do auditor;

c) **auditoria na engenharia de software:** consulta por nome do auditor, por nome da engenharia de software, por data, intervalos de data, ano, mês e dia, conforme a figura 25.

Figura 25. Interface de Pesquisa da Auditoria na ES

Os relatórios são exposições e relações dos principais fatos, dados e informações colhidas pela ferramenta AudiSoft administradas pelos auditores.

No sistema existem relatórios sobre auditor, engenharia de software, auditoria na ES e anotações da agenda do auditor, visíveis na Figura 26. Os tipos de relatórios são:

a) **auditor:** listar todos os auditores cadastrados com empresa, com endereço e com contatos;

b) **engenharia de software:** listar todas as engenharias de softwares cadastradas com empresa, com endereço e com contatos;

c) **auditoria na engenharia de software:** listar todas as auditorias da engenharia de software em geral e em detalhes;

d) **agenda do auditor:** listar todas as anotações agendadas.

Figura 26. Interface do Menu Relatórios do AudiSoft

8.3.5 Resultados – Relatórios Cifrados e Decifrados

Na área de Resultados da ferramenta AudiSoft encontra-se a utilização da criptografia de dados juntamente com algoritmo RSA. Esta questão é muito importante e tem um grande destaque junto com auditoria neste projeto de pesquisa, por isso ela foi estudada e implementada na ferramenta AudiSoft.

Nesta parte do programa são gerados os relatórios de auditoria de uma determinada engenharia de software feitos pelo auditor. Os relatórios criados podem ser cifrados e decifrados respectivamente através da chave pública e particular gerada na ferramenta.

Esta atividade pode ser usados pelos dois usuários, auditor e cliente, do AudiSoft, porém no momento em que o auditor realiza o *login* do sistema ele ganha mais uma opção, a de gerar o relatórios pesquisando e escolhendo qual auditoria realizada escolher e desta forma poder cifrar e decifrar esta auditoria e ainda poder salvar e abrir arquivos de relatórios cifrados e decifrados.

Entretanto, o cliente, responsável pela engenharia de software tem o direito apenas de abrir seu relatório cifrado e junto com a sua chave particular, ambos repassados pelo auditor responsável, ter a possibilidade de decifrar os resultados (relatório) encontrados e visualizar quais questões foram levantadas sobre o seu desenvolvimento segundo o auditor e desta forma poder salvar o arquivo decifrado (texto normal).

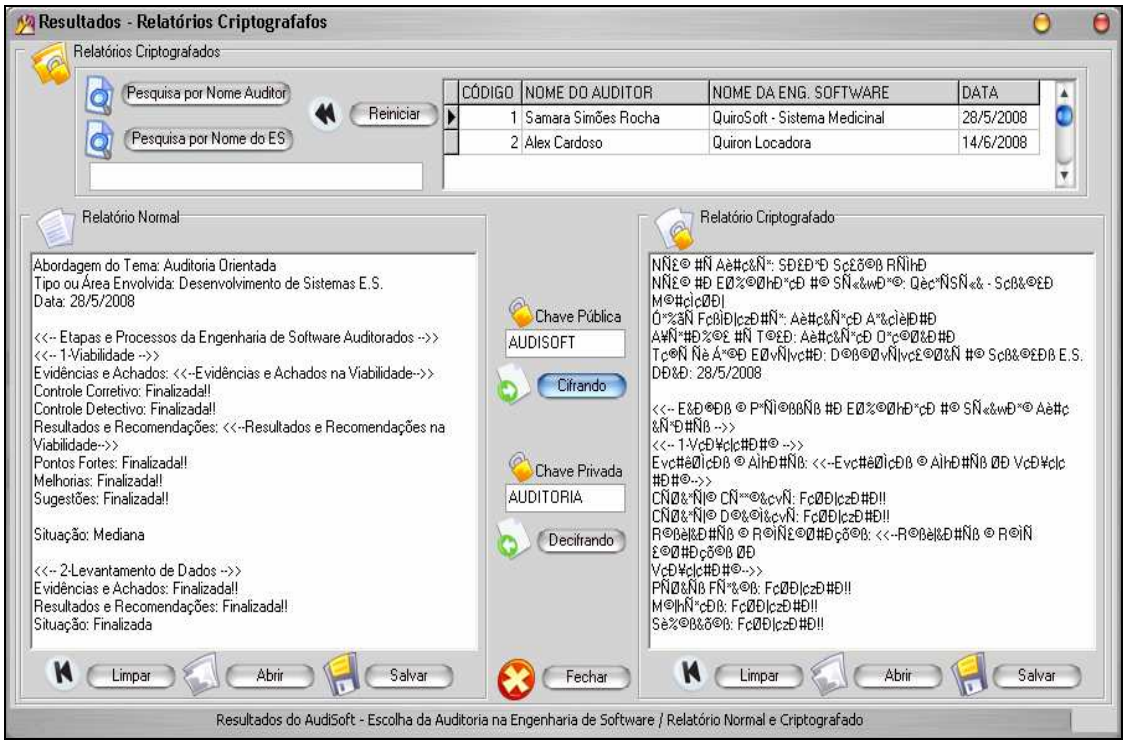


Figura 27. Interface dos Resultados – Relatórios Cifrados e Decifrados

A criptografia de dados foi implementada através do algoritmo RSA pelo motivo de se estar tratando de informações importantes e sigilosas sobre um determinado desenvolvimento de sistemas. Desta forma, fica inevitável a utilização de mecanismos de segurança nas informações e dados manipulados e coletados pela ferramenta AudiSoft. Com isso, consegue-se garantir a segurança, a confiabilidade e o sigilo das informações, de maneira que nenhuma pessoa não autorizada tenha acesso aos resultados reais (relatório) da auditoria, ou seja, se conseguir ter acesso terá apenas um relatório todo cifrado e de muito difícil interpretação e compreensão.

Uma outra questão importante nesta área que merece destaque é a implementação do algoritmo RSA, pois o mesmo requer um maior domínio e conhecimento matemático. Desta forma, a Figura 28 mostra os valores e as fórmulas utilizadas no algoritmo RSA e como foi achada a chave primária (pública), usada para cifrar, e a chave secundária (particular) usado para decifrar.

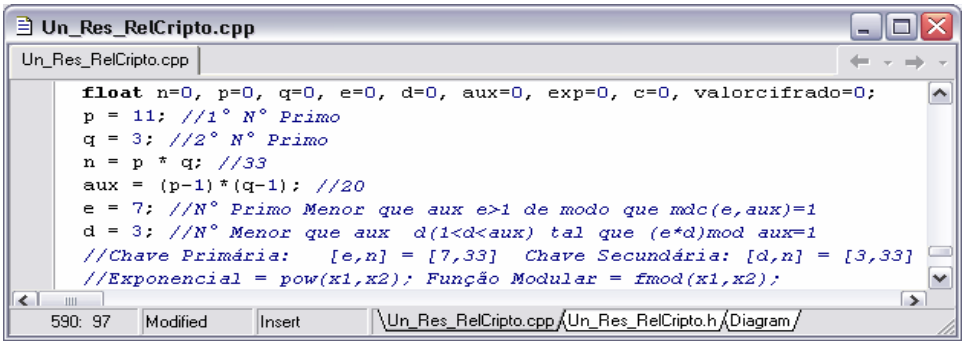


Figura 28. Código Fonte – Valores para o Algoritmo RSA

Na Figura 29 é comentado detalhadamente o processo de cifragem da informação, onde um relatório de texto normal e transformado em um texto cifrado e ilegível.

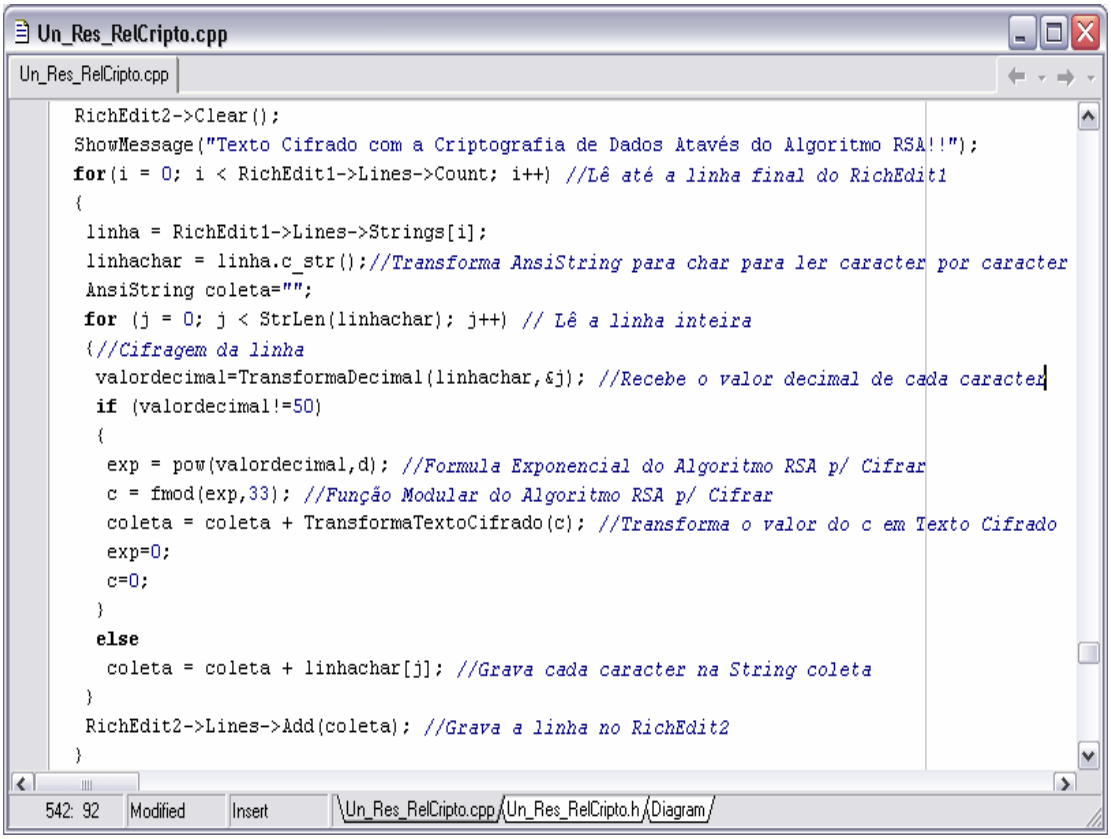


Figura 29. Código Fonte – Cifragem do Relatório com Algoritmo RSA

Contudo, na Figura 30 é comentado detalhadamente o processo inverso, que é o da decifragem da informação, onde um texto cifrado e ilegível e transformado em um relatório de texto normal e compreensível.

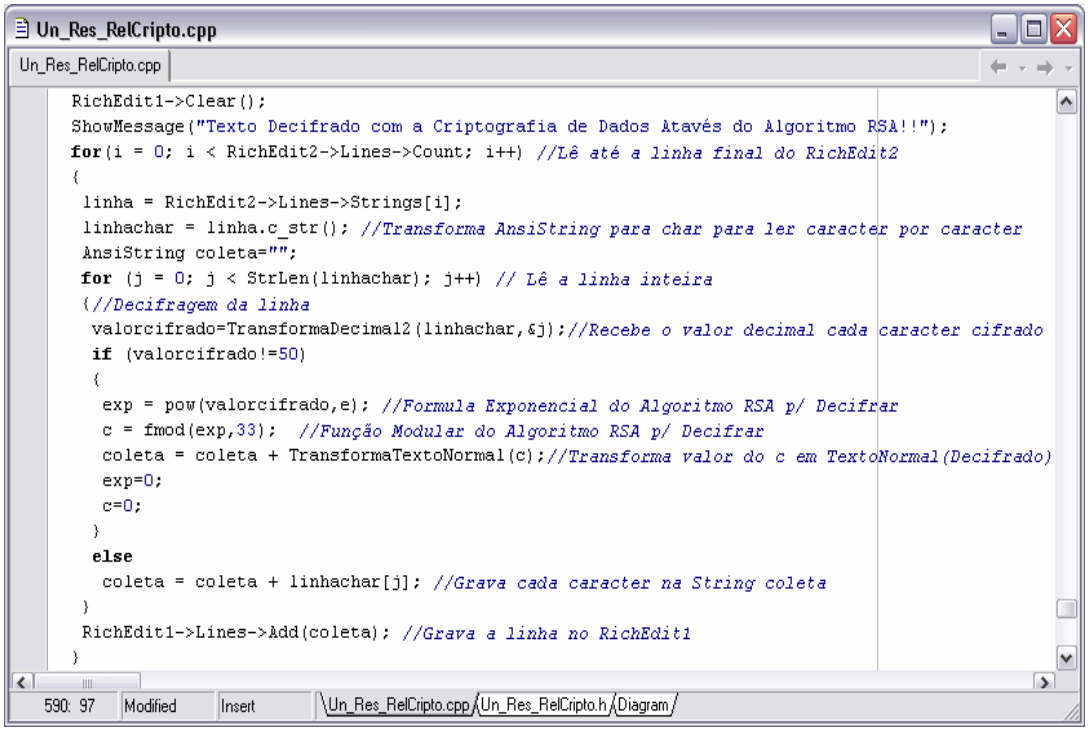


Figura 30. Código Fonte – Decifragem do Relatório com Algoritmo RSA

8.3.6 Ferramentas Auxiliares do AudiSoft

Entre as ferramentas auxiliares disponíveis no AudiSoft estão a calculadora do Windows (através de um atalho), para possíveis cálculos necessários no processo de auditoria, e a agenda com calendário para anotações e pesquisas de compromissos do auditor, conforme a Figura 31.

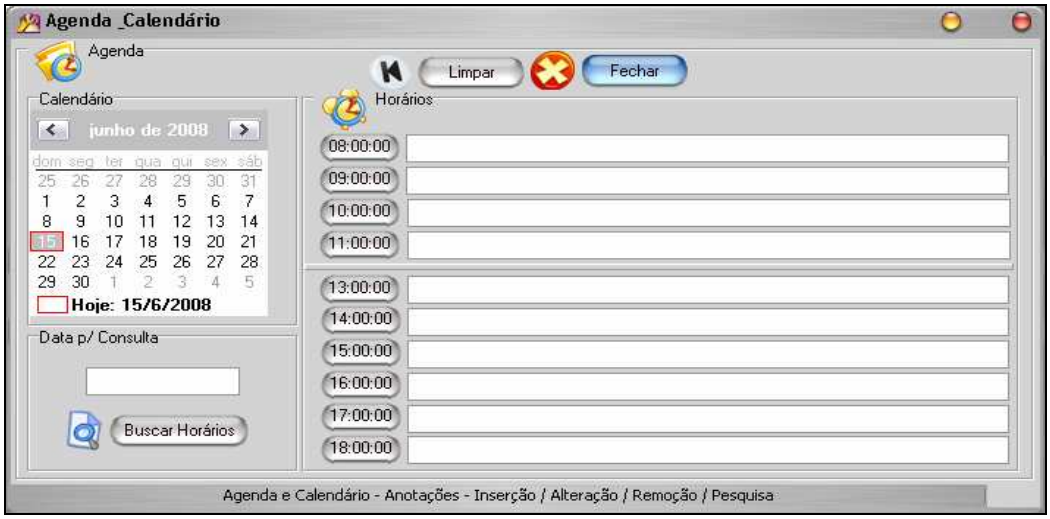


Figura 31. Interface da Ferramenta Agenda

A agenda funciona da seguinte forma: escolhe-se uma data qualquer para consulta, se não houver nenhum compromisso ou anotação marcado no horário de interesse, insere-se a anotação no campo ao lado da hora preferida e pressiona-se no botão com horário. Automaticamente, a agenda saberá que é uma inserção na agenda, se algo for acrescentado ou retirado da anotação e for pressionado no botão com horário novamente a agenda saberá que é uma alteração, e se tudo for apagado saberá que é uma exclusão. A lista com todas as anotações e compromissos do auditor podem ser vista também na parte de Relatórios - Agenda e Calendários

8.4 RESULTADOS OBTIDOS

O principal resultado obtido deste projeto de pesquisa foi a ferramenta de auditoria AudiSoft, onde esta tem o objetivo de auxiliar o auditor nas atividades relacionadas a auditoria e a fiscalização do desenvolvimento de sistemas, da engenharia de software. Para isto são usados técnicas e papéis de trabalho da auditoria com intuito de encontrar e apontar possíveis problemas e suas soluções para o responsável da engenharia de software (cliente).

Além disso, foi obtido um mecanismo de segurança através da criptografia de dados (algoritmo RSA) na implementação que garanti e confirma a proteção e sigilo dos dados e informações encontrados nos resultados (relatórios) emitidos pela ferramenta ao cliente após o termino das atividades da auditoria.

Terminados os testes da ferramenta, constatou-se que a mesma atingiu o objetivo geral de desenvolver uma aplicação para auditoria em desenvolvimento de sistemas, utilizando criptografia nos dados coletados e manipulados nesta atividade, e conseqüentemente, alcançou todos os objetivos específicos determinados no projeto.

CONCLUSÃO

Atualmente existe uma grande disseminação de sistemas computacionais para os mais variados tipos de controles de operação e que cada vez mais tem relevância para as organizações que os utilizam.

Conseqüentemente ocorre um aumento significativo de empresas, organizações ou equipes de desenvolvimento que trabalham com construção e engenharia de software, por isso torna-se indispensável o trabalho de auditoria de sistemas neste âmbito.

Por isso, foi desenvolvida a ferramenta AudiSoft para auxiliar a auditoria a examinar cada divisão da engenharia do software, do desenvolvimento do sistema proposto a ser avaliado, para que se extraia resultados e relatórios, como pontos fracos e fortes, melhorias, sugestões e ao final poder concluir se o sistema e seu desenvolvimento são ou não adequados, utilitários e confiáveis. Todo este resultado ou relatório serão cifrados e decifrados, através do algoritmo RSA, para maior segurança dos dados na ferramenta. Desta forma, conseguiu-se alcançar e atingir o objetivo geral e todos os objetivos específicos traçados no projeto de pesquisa.

Durante toda a elaboração desse projeto de pesquisa se fez necessário um estudo bibliográfico detalhado sobre auditoria, auditoria de sistemas, engenharia de software, segurança da informação, criptografia de dados e algoritmo RSA. Nessa etapa vale ressaltar a dificuldade de encontrar material publicado sobre auditoria na área da Ciência da Computação, principalmente na área de engenharia de software e desenvolvimento de sistemas.

Após ter entendido a fundamentação teórica e os objetivos da ferramenta AudiSoft, iniciou-se a modelagem e implementação do sistema. Durante essa etapa

houve dificuldades específicas na implementação, principalmente na programação da ação de auditorar a engenharia de software, devido ao grande número de etapas e processo envolvidos no desenvolvimento de sistemas, o que resultou em muitas tabelas e informações do banco de dados com a ferramenta.

Porém a maior dificuldade encontrada no AudiSoft foi a implementação do algoritmo RSA, pois este trabalha com exponencial e funções modulares, e a ferramenta de programação, *C++ Buider 6*, não é preparada para tratamento de números grandes e volumosos, o que ocasiona erros matemáticos no momento do cálculo das fórmulas e funções de cifragem e decifragem do algoritmo RSA. A solução encontrada foi usar valores pequenos para cálculo e com isso apenas as letras minúsculas conseguiram ser cifradas e decifradas, o que já dificulta e muito o entendimento e interpretação dos dados contidos nos relatórios e resultados gerados pela ferramenta AudiSoft.

O AudiSoft aplica a auditoria com pensamento e a filosofia de livre acesso a todos os dados e informações contidas nas etapas e processos da engenharia de software. Por isso, um trabalho futuro que merece destaque é a implementação de uma ferramenta de auditoria que aplique a engenharia reversa, ou seja, a partir de uma aplicação pronta, um programa executável consiga-se ter acesso a dados e lógica da programação do software a ser analisado e assim poder aplicar a auditoria no mesmo.

Um outro trabalho futuro seria a implementação da ferramenta AudiSoft ou pelo menos a parte da criptografia em uma outra linguagem ou ambiente de programação que de estrutura para tratamento de números grandes e que também seja multiplataforma, para que possa ser executado em outros sistemas operacionais.

E como último trabalho futuro, a necessidade de testar o sistema desenvolvido AudiSoft, colocando-o em pratica para utilização.

REFERÊNCIAS

ALBERTIN, Alberto Luiz; MOURA, Rosa Maria de. **Tecnologia de informação.** São Paulo: Atlas, 2004.

ARIMA, Carlos Hideo; SANTOS, Jose Luiz dos; SCHMIDT, Paulo. **Fundamentos da Auditoria de Sistemas.** São Paulo: Atlas, 2006.

ARIMA, Carlos Hideo. **Metodologia de auditoria de sistemas.** São Paulo: Érica, 2006.

BARCELOS, Moair. **Auditoria de fraudes:** uma abordagem geral. 2006. 89 f. Monografia (Especialização em Auditoria Integral) - Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2006.

BRASIL. Tribunal de Contas da União. **Manual de Auditoria de Sistemas** / Tribunal de Contas da União. Brasília: TCU, Secretaria de Auditoria e Inspeções, 1998.

BUCHMANN, Johannes A. **Introdução à criptografia.** São Paulo: Berkeley, 2002.

BURNETT, Steve; PAINE, Stephen. **Criptografia e segurança:** o guia oficial RSA. Rio de Janeiro: Elsevier, 2002.

CARVALHO, Daniel Balparda de. **Segurança de Dados com Criptografia:** Métodos e Algoritmos. 2. ed. Rio de Janeiro: Books Express, 2001.

CASSARRO, Antonio Carlos. **Sistemas de informações para tomada de decisões.** 3. ed. São Paulo: Pioneira, 2001.

CONSELHO REGIONAL DE CONTABILIDADE DO ESTADO DE SÃO PAULO. **Auditoria por meios eletrônicos.** São Paulo: Atlas, 1999.

DIAS, Cláudia. **Segurança e Auditoria da Tecnologia da Informação.** Rio de Janeiro: Axcel Books , 2000.

GIL, Antonio de Loureiro. **Auditoria de Computadores.** 5. ed. São Paulo: Atlas, 2000.

GRAEML, Alexandre Reis. **Sistemas de informação: o alinhamento da estratégia de TI com a estratégia corporativa.** São Paulo: Atlas, 2000.

GUSTAFSON, David A. **Teoria e problemas de engenharia de software.** Porto Alegre: Bookman, 2003.

IMONIANA, Joshua Onome. **Auditoria de sistemas de informação.** São Paulo: Atlas, 2005.

MONTANHEIRO, Paulo César. **O Papel da Auditoria da Informação na Gestão Organizacional.** 2006. 125 f. Dissertação (Mestrado) – Pontifícia Universidade Católica de Campinas (PUC), Centro de Ciências Sociais Aplicadas, Pós-Graduação em Ciência da Informação, Campinas, São Paulo, 2006.

MORENO, Edward David; PEREIRA, Fábio Dacêncio; CHIARAMONTE, Rodolfo Barros. **Criptografia em Software e Hardware.** São Paulo: Novatec, 2005.

PFLEEGER, Shari Lawrence. **Engenharia de Software: teoria e prática.** 2. ed. São Paulo: Prentice Hall, 2004

PRESSMAN, Roger S. **Engenharia de Software.** São Paulo: Makron Books, 1995.

SÁ, Antônio Lopes. **Curso de Auditoria.** 10. ed. São Paulo: Atlas, 2002.

SHILLER, Larry. **Excelência em software.** São Paulo: Makron Books, 1992.

SILVA JÚNIOR, José Barbosa da. **Auditoria em ambiente de internet.** São Paulo: Atlas, 2001.

SIMCSIK, Tíbor; POLLONI, Enrico Giulio Franco. **Tecnologia da informação automatizada.** São Paulo: Berkeley, 2002.

SOMMERVILLE, Ian. **Engenharia de software**. São Paulo: Addison-Wesley, 2003.

YAMADA, Edson Noboru. **Uma Ferramenta Para Avaliação Semi-Automática de Programas Exercício.** 1997. 103 f. Dissertação (Mestrado) – Instituto de Matemática e Estatística da Universidade de São Paulo, Mestrado em Matemática Aplicada, São Paulo, São Paulo, 1997.

BIBLIOGRAFIA COMPLEMENTAR

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML.** Rio de Janeiro: Campus, 2003.

BRASIL, Secretaria de Política de Informática e Automação. **Tecnologia da informação:** Programa Brasileiro da Qualidade e Produtividade em Software. 4. ed. Brasília: Ministério da Ciência e Tecnologia, 2006.

CANTÚ, Marco; TORTELLO, João Eduardo Nóbrega. **Dominando o Delphi 6:** a bíblia. São Paulo: Makron Books, 2002.

CHANDOR, Antony; GRAHAM, John; WILLIAMSON, Robin. **Análise de sistemas teoria e prática.** Rio de Janeiro: Livros Técnicos e Científicos, 1977.

DATE, C. J.; Newton Dias de Vasconcellos. **Bancos de dados tópicos avançados.** Rio de Janeiro: Campus, 1988.

GAMMA, Erich. **Padrões de projeto: soluções reutilizáveis de software orientado a objetos.** Porto Alegre: Bookman, 2000.

GANE, Chris; SARSON, Trish. **Análise estruturada de sistemas.** Rio de Janeiro: LTC, 1983/1999.

GHEZZI, Carlo; JAZAYERI, Mehdi; MANDRIOLI, Dino. **Fundamentals of soltware engineering.** New Jersey: Prentice-Hall, 1991.

GIL, Antonio de Loureiro. **Segurança de Informática.** 2. ed. São Paulo: Atlas, 1998.

INTHURN, Cândida. **Qualidade & teste de software: engenharia de software, qualidade de software, qualidade de produtos de software, teste de software, formalização do processo de teste, aplicação prática dos testes.** Florianópolis: Visual Books, 2001.

LARMAN, Graig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos.** Porto Alegre: Bookman, 2000.

MARTIN, James; MCCLURE, Carma. **Técnicas estruturadas e CASE.** São Paulo: McGraw-Hill, 1991.

MEDEIROS, Ernani. **Desenvolvendo software com UML 2.0:** definitivo. São Paulo: Makron Books, 2004.

MENDES, Antonio. **Arquitetura de software: desenvolvimento orientado para arquitetura.** Rio de Janeiro: Campus, 2002.

OLIVEIRA, Wilson José de. **SQL server 7 com delphi.** Florianópolis: Visual Books, 2001.

PENDER, Tom. **UML, a bíblia.** Rio de Janeiro: Elsevier, 2004.

RAMALHO, José Antonio A. **SQL:** a linguagem dos bancos de dados. São Paulo: Berkeley, 1999.

RUMBAUGH, James. **Modelagem e projetos baseados em objetos.** Rio de Janeiro: Campus, 1994.

RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário.** Rio de Janeiro: Campus, 2000.

SONNINO, Bruno. **Desenvolvimento aplicações com delphi 6.** São Paulo: Market books, 2001.

SOUZA, Adriano Luiz de. **Portando aplicações do sistema gerenciador de bando de dados sybase para firebird.** 2005. 84 f. Monografia (Trabalho de Conclusão de Curso Especialização em Banco de dados) - Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2005.

WAZLAWICK, Raul Sidnei. **Análise e projeto de sistemas de informação orientados a objetos.** Rio de Janeiro: Elsevier, 2004.

WILDEROM, Bastiaan Pieter Marinus; WILDEROM, Stella Martinez. **Firebird/InterBase 6.0: cliente/servidor com Delphi 6:** tópicos avançados. São Paulo: Érica, 2002.

APÊNDICE A – ETAPAS E PROCESSOS DA ENG. DE *SOFTWARE* (ES)

Etapas e Processos relacionados e encontradas no desenvolvimento de um sistema, na engenharia de *software*, que vão ser analisados passo a passo pelo Auditor na Ferramenta AudiSoft. A seguir são descritas as etapas por ordem de desenvolvimento junto com o objetivo e a palavras chaves a serem analisado, tornando-se uma importante ajuda para executar a auditoria na Engenharia de *Software*.

1. VIABILIDADE - **Objetivo:** definir aspectos de funcionalidade e desempenho do *software*. **Palavras Chaves**:
    a) Viabilidade técnica: permite obter informações sobre a possibilidade de se construir um sistema considerando os requisitos funcionais e de desempenho.
    a) Viabilidade legal: implica em verificar se o desenvolvimento do sistema não vai violar limites da lei.
    b) Viabilidade econômica: permite verificar se o sistema pode ser desenvolvido de modo que a relação custo/benefício esteja dentro de padrões aceitáveis de desenvolvimento.
    c) Analisar e definir o sistema: analisar e criar um primeiro nível de definição do sistema que sirva de ponto de partida para as demais tarefas de engenharia relacionadas ao seu desenvolvimento.
    d) Identificar as necessidades: identificar as necessidades do usuário do sistema (resultando no Documento Conceitual do Sistema).
    e) Realizabilidade, alocação das Funções e Restrições de Prazo e Custo: Avaliar a concepção do sistema quanto à realizabilidade, alocar as

funções ao hardware, ao *software* e aos demais elementos do sistema e estabelecer as restrições de prazo e custo.

f) As alternativas: permitem levantar quais seriam as opções com relação ao desenvolvimento do sistema. Parte do documento de Especificação de Sistema.

2. LEVANTAMENTO DE DADOS - **Objetivo:** levantar o maior número de dados e informações necessários para o desenvolvimento do sistema. **Palavras Chaves**:

a) Informativos – Reais, Irreais e Opinativos: dados provenientes de respostas e perguntas formuladas pelo analista. Podem traduzir uma realidade, uma fantasia ou uma opinião.

b) Concretos – Reais e Irreais: dados diretamente constatados pelo analista diante da execução do trabalho. Podem traduzir uma realidade ou uma irrealidade.

c) Questionário: técnica de levantamento de informações que consiste em formular perguntas objetivas e padronizadas, com vistas a obter dados sobre um método ou organização. Para obter poucas informações de um grande numero de pessoas. Quando a fonte de informações está muito distante. Para preparar uma entrevista. Para selecionar pessoas a serem entrevistadas ou setores a serem observados. É a técnica de mais difícil preparação.

d) Observação Direta: consiste na obtenção de informações através da constatação dos fenômenos relativos ao trabalho. Em estudos de tempos e movimentos. Na análise de *layout* físico e do ambiente de trabalho. No estudo de arquivos. Pode ser conjugado com a entrevista.

e) Entrevista: é um diálogo planejado, controlado e com objetivo determinado. Preparação para entrevista. O desenrolar da entrevista. *Check-list* do entrevistador.

f) Tipos de Observação: inspeção Geral. Inspeção dos serviços individuais dos funcionários. Acompanhamento de certas rotinas. Preenchimento e utilização de formulários. Observação do ambiente.

3. PLANEJAMENTO ESTRATÉGICO - **Objetivo:** define como e o que será necessário para a instalação e implantação do novo sistema. **Palavras Chaves**:

    a) Infra-estrutura administrativa: para a implantação do sistema. Treinamento. Especialização. Capacitação. Mão-de-obra.

    b) Recursos Humanos: técnicos, especialistas.

    c) Infra-estrutura técnica: para que o *software* seja executado utilizando todos os recursos oferecidos: hardware e especificações.

    d) Recursos Tecnológicos: tipos de tecnologias e equipamentos de apoio.

4. ANÁLISE DE REQUISITOS - **Objetivo:** é o processo de descobrir, refinar, modelar e especificar os propósitos do cliente. **Palavras Chaves**:

    a) Reconhecimento do problema: analisa a documentação da especificação do sistema e o plano do *software*, para entender seu posicionamento no sistema. O analista reconhece os elementos problemáticos básicos, conforme percebidos pelo cliente.

    b) Avaliação e síntese: definir o problema. Causas que levam ao problema. Conseqüências do problema. Qual a solução para o problema? Qual a solução alternativa?

c) Modelagem: melhor compreensão dos fluxos. Pode ser necessária a criação de um protótipo.

d) Especificação de seus requisitos: produz um documento o qual registra os resultados das tarefas realizadas. Pode desenvolver um manual, pode contribuir para o analista visualizar o *software* sob a ótica do cliente.

e) Revisão: um manual permite a revisão dos requisitos num estágio prematuro, mas alguns problemas no *software* podem ser evitados.

f) Requisitos Funcionais e Não-Funcionais: permite ao engenheiro de *software* refinar a atribuição do *software* e construir modelos dos domínios de dados, funcional e comportamental que serão tratados pelo *software*. Funcionais – o que o sistema deve fazer. Não-funcionais – restrições sobre como o sistema deve desempenhar suas funções.

5. DIAGRAMAS - **Objetivo:** o Diagrama Estruturado do *Software* mostrará sua estrutura hierárquica em módulos, e as informações trocadas entre os mesmos. Deverão ser representados no diagrama, além dos procedimentos lógicos, os módulos de controle e segurança necessários para o *Software*. Ou seja, diferentes formas (diagramas) de representação do sistema e suas ramificações (processos, atores, ações). **Palavras Chaves**:

    a) Completar fluxo de informação, dados, saídas, relação entre os dados, reciclagem de erros.

    b) Definir necessidades de relatórios, volume, freqüência, distribuição.

    c) Lógica geral do sistema e Procedimentos de controle e de auditoria.

    d) Identificar programas de computador e procedimentos manuais.

e) Especificar programas: descrever os programas em termos de Objetivo, Procedimentos Básicos (descrição dos módulos executados), Projeto de Comunicação e Telas.

Tipos de Diagramas:

- *Use Case* (Casos de Uso): diagrama usado para identificar como o sistema se comporta em várias situações que podem ocorrer durante sua operação.
- Classe: diagrama de classes mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos. Descreve os tipos de objetos no sistema e os vários tipos de relacionamento estático que existem entre eles.
- Objeto: diagrama de objetos modela as instâncias das classes contidas no diagrama de classes, o diagrama de objetos mostra um conjunto de objetos e seus relacionamentos no tempo.
- Estado: diagrama de Estado descreve o comportamento de um sistema. Ele descreve todos os estados possíveis em que um objeto particular pode estar e como o estado do objeto muda como resultado de eventos que o atingem.
- Seqüência (interação): diagrama de seqüência, um objeto é mostrado como uma caixa na parte superior de uma linha tracejada vertical. A linha vertical é chamada de linha de vida do objeto. A linha de vida representa a vida do objeto durante a interação.
- Colaboração (interação): diagrama de colaboração os objetos são mostrados como ícones e flechas indicam as mensagens enviadas dentro de um dado caso de uso.

- Atividade: diagrama de atividade combina idéias de várias técnicas. Esses diagramas são particularmente úteis na conexão com *workflow* e na descrição de comportamento que tem muito processamento em paralelo.
- Utilização: mostra as relações físicas entre componentes de *software* e hardware no sistema implementado.
- Componente: mostra os vários componentes em um sistema e suas dependências. Um componente representa um módulo físico do código.

6. DICIONÁRIO DE DADOS - **Objetivo:** descrever classes e atributos pertencentes ao sistema, detalhando as variáveis da classe, seus tipos, descrições, restrições, entre outros. **Palavras Chaves**:
    a) Tabelas Normalizadas: verifica se todas as tabelas do banco de dados do sistema estão normalizadas e corretamente estruturadas.
    b) Tabelas Não-Normalizadas: tabelas do banco de dados mal formuladas e feitas, sem as referencias necessárias, e que podem trazer problemas futuros ao sistema.
    c) Referências: controla as chaves primárias e secundárias entre tabelas do banco de dados do *software*.
    d) Chaves Primárias(PK) e Secundárias(FK): analisa se a classificação dos dados em chaves primárias(PK) e chaves secundárias(FK) nas tabelas são pertinentes e logicamente corretas.
    e) Classificação de Variáveis: organiza e verifica se todas as variáveis estão corretamente classificadas em relação ao seu tipo, descrição e restrição, constatando qualquer anormalidade encontrada.

7. IMPLEMENTAÇÃO - **Objetivo:** o objetivo desta fase é o desenvolvimento e simulação do *software* especificado no Projeto. O resultado são os programas fontes, devidamente testados. Estes, por sua vez, devem ser entregues ao usuário. Neste caso, cabe ao programador dominar as características específicas das linguagens, ferramentas e estruturas de dados para adaptar o código gerado aos requisitos indicados quando necessário. **Palavras Chaves**:
    a) Codificação dos programas, segundo uma lógica.
    b) Construir arquivos.
    c) Fazer testes dos programas.
    d) Testes, através de massa de dados abrangente.
    e) Simulação: o objetivo da simulação é colocar o *Software* em funcionamento, dentro dos requisitos estabelecidos.
    f) Elaboração dos manuais de operação.

8. TESTES - **Objetivo:** a fase de testes envolve os testes de unidade, feitos pelo programador, para verificar se os componentes gerados atendem à especificação do projetista, e aos testes de caso de uso, normalmente efetuados por um analista experiente, que visam verificar a adequação do sistema aos requisitos inicialmente levantados. O processo de realização de testes concentra-se nos aspectos lógicos internos do *software*, garantindo que todas as instruções tenham sido testadas, e concentram-se também nos aspectos funcionais externos, ou seja, realizando testes para descobrir erros e garantir que a entrada definida produza resultados reais que concordem com os resultados exigidos. **Palavras Chaves**:
    a) Garantir que todas as instruções tenham sido testadas.

b) Verificar se os componentes gerados atendem à especificação do projetista.
c) Verificar a adequação do sistema.
d) Descobrir erros.

9. INTERFACES - **Objetivo:** tornou indispensável no desenvolvimento de *software* a preocupação em conseguir projetar sistemas interativos mais usáveis. Desta maneira deve-se testar e avaliar os sistemas para assegurar que os mesmos estejam de acordo com as expectativas dos usuários. Avaliação de Usabilidade de sistemas de *software* pode contribuir para, com custos reduzidos, torná-los mais fáceis de aprender, com menos erros e mais eficientes e usáveis no suporte à tarefa do usuário. **Palavras Chaves**:
    a) Avaliar a extensão das funcionalidades do sistema.
    b) Avaliar os efeitos da interface nos usuários: facilidade de aprendizagem, facilidade e eficiência de uso e efetivo suporte à tarefa.
    c) Identificar algum problema com o sistema.

10. IMPLANTAÇÃO - **Objetivo:** a conversão e inicialização de arquivos e a implantação do *software* para produção. Nesta fase, é elaborado e entregue o Manual do Usuário, assim como o Termo de Encerramento do Desenvolvimento do *Software*, onde o analista ou empresa desenvolvedora declara que o *software*, uma vez implantado, está entregue e considerado, aceito: devendo o mesmo entrar no período de garantia. **Palavras Chaves**:

a) Manual do Usuário: montar o Manual do Usuário conforme modelo do documento, de tal forma que, o usuário por mais leigo que seja, terá um guia que o induzirá a produzir e a operar efetivamente o *software*.
b) Controle da Qualidade Funcional do *Software*: neste estágio de desenvolvimento deverá ser planejada e realizada uma revisão para a avaliação da estrutura e conteúdo do Manual do Usuário, observando a adequada descrição das atividades para a eficaz operação do *software*.
c) Instalação: desenvolver programa/rotina para a instalação do *software*. A rotina em questão deverá entre outras coisas preocupar-se com: Criação de Diretórios, inicialização de arquivos, carga dos programas objetos, entre outros.
d) Encerramento do Desenvolvimento: por ocasião da conclusão desta fase deverá ser providenciado o Termo de Encerramento do Desenvolvimento do *Software*, com a devida aceitação pelo usuário.

11. TREINAMENTO - **Objetivo**: treinamento e capacitação de usuários para a utilização da ferramenta, do *software* desenvolvido. **Palavras Chaves**:
    a) Capacitar o usuário para o uso/operação do *Software* com confiabilidade e segurança.
    b) Conversão de Arquivos.
    c) Converter os arquivos atuais, se necessário, para a nova estrutura projetada.
    d) Controle de Qualidade da Fase.
    e) Deverá se preparado e realizado, em conjunto com o usuário os procedimentos para execução do teste de validação do *Software*.

12. AVALIAÇÃO - **Objetivo:** executar as atividades de produção do *software* pelo usuário, com acompanhamento inicial da execução das rotinas, avaliação do desempenho, pequenos ajustes e análise de resultados. **Palavras Chaves**:
    a) Periodicidade: responsável por manter o sistema funcionando. Disponibilidade de tempo para acompanhamento entre o *software* desenvolvido e o usuário (cliente).
    b) Correção: responsável por mudanças no *software* para mantê-lo viável. Alterações internas e externas (mudanças de legislação, reorganização da empresa). Alteração na configuração do equipamento de processamento de dados (*hardware*/*software*). Correção dos erros encontrados no sistema. Modificação para melhoria de desempenho do sistema, de maneira a atender melhor aos escalões administrativos intermediários e inferiores da empresa.

13. PROJETO DE CUSTO - **Objetivo:** determinar detalhadamente os recursos, materiais e observações necessários para a implantação e operação do projeto, junto com o desembolso e período para a realização do mesmo. **Palavras Chaves**:
    a) Recursos: os recursos equivalem a material e infra-estrutura que o Cliente ainda não possui e que precisa ser adquirido para a implantação do projeto (*software*).
    b) Cronograma de Execução: materiais e períodos para adquirir.
    c) Cronograma de Desembolso (R$ Reais).
    d) Observações Importantes.

APÊNDICE B – EXEMPLO DE EVIDÊNCIAS DE AUDITORIA NA ES

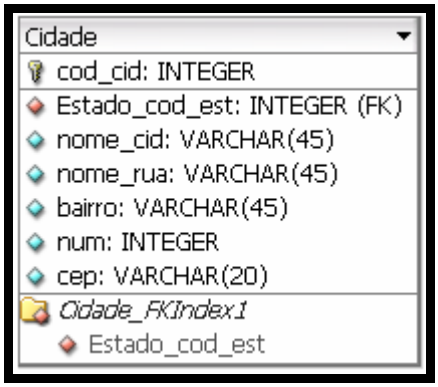
Exemplos ilustrativos de evidências e achados de auditoria que podem ocorrer no desenvolvimento de sistemas com recomendações e opiniões de como solucionar os problemas e questões apontadas.

Projeção dos Resultados

Descrição		nov-2007	dez-2007
1 FATURAMENTO DE VENDAS		$ 3.325,00	$ 5.600,00
2,1 (-) Impostos sobre o faturamento	3,0%	$ (99,75)	$ (168,00)
2,2 (-) Comissões a pagar	0,0%	$ -	$ -
3 FATURAMENTO LÍQUIDO DE VENDAS		$ 3.225,25	$ 5.432,00
4 (-) Despesas de Produção		$ (2.225,00)	$ (3.417,50)
Pessoal da Produção		$ *(480,00)*	$ *(480,00)*
Compras/Matéria-prima		$ *(1425,00)*	$ *(2.412,50)*
Fretes & Embalagens		$ *(320,00)*	$ *(525,00)*
5 MARGEM DE CONTRIBUIÇÃO		$ 1.000,25	$ 2.014,50
6 (-) DESPESAS OPERACIONAIS		$ (6.784,72)	$ (6.784,72)
6,1 (-) Despesas Administrativas		$ (6.511,80)	$ (6.511,80)
Pessoal da Administração & Terceiros		$ *(5.211,80)*	$ *(5.211,80)*
Aluguéis, Condomínios e IPTU		$ *(500,00)*	$ *(500,00)*
Despesas de Vendas e Marketing		$ *(100,00)*	$ *(100,00)*
Despesas Gerais		$ *(700,00)*	$ *(700,00)*
6,2 Depreciação		$ (272,92)	$ (272,92)
7 RESULTADO OPERACIONAL		$ (5.784,47)	$ (4.770,22)
8 Juros de Financiamentos		$ (175,00)	$ (175,00)
9 LUCRO BRUTO		$ (5.959,47)	$ (4.945,22)
10 (-) Impostos sobre os lucros	0%	$ -	$ -
11 LUCRO LÍQUIDO		$ (5.959,47)	$ (4.945,22)
MARGEM DE CONTRIBUIÇÃO		30,1%	36,0%
PONTO DE EQUILÍBRIO		$ 21.646,32	$ 18.101,80

Viabilidade - Viabilidade legal: Desenvolvimento do sistema viola limites da lei. Projeção de Resultados incorretos sem taxas e Impostos. Pode ocasionar emissão de cupons e notas fiscais incorretas (frias)

Viabilidade - Viabilidade econômica: o custo/benefício não está dentro de padrões aceitáveis de desenvolvimento. Custo muito Elevado R$50.000



Dicionário de Dados - Tabelas Não-Normalizadas: Tabelas do banco de dados mal formuladas e feitas. Normaliza-las em: país, estado, cidade, bairro, endereço

APÊNDICE C – ARTIGO CIÊNTIFICO

AUDISOFT – FERRAMENTA PARA AUDITORIA NO DESENVOLVIMENTO DE SISTEMAS UTILIZANDO CRIPTOGRAFIA DE DADOS COM O ALGORITMO RSA

Diego Machado Medeiros[1], Paracelso de Oliveira Caldas[1]

1Curso de Ciência da Computação, Unidade Acadêmica de Ciências, Engenharias e Tecnologia – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma – SC

diegomacmed@gmail.com, poc@unesc.net

Resumo. *Este artigo aborda o estudo sobre a Auditoria executada nas etapas e processos da Engenharia de Software (ES) no Desenvolvimento de Sistemas. Para a segurança dos dados coletados e manipulados nesta atividade foi utilizada a Criptografia por meio do Algoritmo RSA, como Mecanismo de Segurança.*

Palavras Chave: Auditoria, Auditoria de Sistemas, Desenvolvimento de Sistemas (Engenharia de *Software*), Criptografia de Dados e Algoritmo RSA

1. Introdução

Com um número cada vez maior de sistemas computacionais desenvolvidos e em desenvolvimento com o propósito de controlar operações de grande relevância no contexto das organizações, visto que empresas estão cada vez mais informatizadas, torna-se indispensável o trabalho de auditoria de sistemas neste âmbito [Brasil 1998].

A utilização da tecnologia da informação para a manipulação e armazenamento de dados nos órgãos, introduz novos riscos para o controle, acrescentando assim outras variáveis às questões relacionadas ao planejamento e execução de atividades de fiscalização. Dessa forma, constata-se a dificuldade de auditar entidades com alto grau de informatização [Brasil 1998].

Para isto este artigo se propõe a desenvolver e implementar a aplicação da tecnologia da informação às atividades de fiscalização, ou seja, a Auditoria de Sistemas e seu desenvolvimento.

A auditoria de sistemas dentro de uma organização é de vital importância, pois aumentam consideravelmente o controle sobre a integridade, operacionalidade e segurança das informações. Evitam também transtornos, sendo os principais, as fraudes e erros, além de garantir o bom desempenho e funcionalidade dos sistemas examinados pela atividade.

Na atividade de auditoria não são conhecidas ferramentas que auxiliem no exame do desenvolvimento de sistemas, na engenharia de *software*. Atualmente, toda esta atividade é realizada manualmente e documentado com ferramentas e aplicativos impróprios, ou seja, *softwares* inadequados de edição de texto, planilhas eletrônicas, o que pode ocasionar perdas e danos provocados pelo grande volume de informações e dados.

Por esta razão, implementou-se uma ferramenta, uma aplicação para auditoria em desenvolvimento de sistemas, que analisa e examina cada divisão da engenharia do

software proposta a ser analisado e avaliado pela auditoria. Consequentemente se terá a possibilidade de extrair resultados e relatórios, como pontos fracos e fortes, melhorias, sugestões e ao final poder concluir se o sistema e seu desenvolvimento são ou não adequados, utilitários e confiáveis.

Outro problema relevante é a falta de segurança que ocorre no processo de auditagem, pelas mesmas razões citadas anteriormente, ou seja, como as informações são registradas manualmente e em ferramentas inadequadas, usuários não autorizados podem ter acesso, prejudicando seriamente os resultados da auditoria.

A proteção por criptografia é uma alternativa para preservar informações sigilosas. Independente do algoritmo criptográfico utilizado, sempre ocorrerá transformação de um texto legível em um ilegível. Mesmo que o invasor obtenha o conteúdo de um arquivo, dificilmente terá acesso aos dados [Moreno, Pereira e Chiaramonte 2005].

O *Rivest - Shamir - Adleman* (RSA) é um sistema ou algoritmo de criptografia de chave assimétrica, onde se gera uma chave pública, geralmente utilizada para cifrar informações, e uma outra privada, utilizada para decifrar os dados [Moreno, Pereira e Chiaramonte 2005]. Dessa forma, é utilizado na aplicação desenvolvida o algoritmo de criptografia RSA, para maior segurança, proteção e sigilo, assegurando que dados não sejam violados e que apenas usuários com permissão consigam ter acesso a informações coletadas e disponibilizadas pela ferramenta de auditoria.

Assim, este artigo abrange o desenvolvimento da auditoria na engenharia de software utilizando a criptografia de dados com o algoritmo RSA.

2. Metodologia

Na metodologia foi realizado o levantamento teórico, como auditoria, auditoria de sistemas, desenvolvimento de sistemas e a auditoria, segurança da informação, criptografia de dados, o algoritmo RSA, a modelagem do sistema e por fim o desenvolvimento da ferramenta AudiSoft, que agrupa todo o levantamento teórico, finalizando com os resultados obtidos e a conclusão.

2.1. Levantamento Teórico

O Levantamento teórico aborda sobre a Auditoria num todo, mostrando suas fases, seus princípios, conceitos e sua natureza. A seguir uma abordagem mais específica sobre a Auditoria de Sistemas, suas ferramentas e técnicas. Logo depois, Sistemas de Informação e o Desenvolvimento de Sistemas (Engenharia de *Software*) e sua Auditoria. Continuando, a Segurança da Informação, mostrando seus objetivos, mecanismos e apontando os sistemas criptográficos como meio para alcançar a segurança. E por fim a Criptografia de Dados, seus tipos (Simétrica e Assimétrica) e o Algoritmo RSA.

2.1.1. Auditoria

A auditoria é uma atividade que engloba o exame de operações, processos, sistemas e responsabilidades gerenciais de uma determinada entidade, com o intuito de verificar sua conformidade com certos objetivos e políticas institucionais, orçamentos, regras, normas ou padrões [Dias 2000].

A auditoria visa descobrir irregularidades, erros e fraudes no tratamento das informações da organização e também identifica os pontos que irão desagradar à alta administração para que estes possam ser corrigidos. Motivos que levam a auditoria a

possuir um grande papel e um alto escalão no ambiente empresarial de uma organização, conforme a Figura 1.

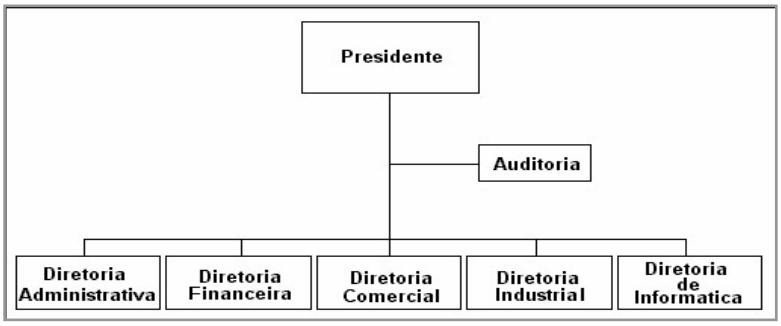


Figura 1. Organização e situação da área de auditoria
Fonte: [Gil, A. 2000]

A atividade de auditoria pode ser dividida, conforme [Dias 2000], em três fases:

- Planejamento: identifica os instrumentos indispensáveis para a realização de uma auditoria.
- Execução: processo em atividade da auditoria, reunido de evidências suficientemente confiáveis, relevantes e úteis para a consecução dos objetivos da auditoria.
- Relatórios: incluem fatos sobre a entidade avaliada, comprovações, conclusões e, eventualmente, recomendações e/ou determinações.

Os princípios e conceitos utilizados da auditoria, conforme [Dias 2000], são:

- Controle: é a fiscalização para que tais atividades ou produtos não se desviem das normas preestabelecidas.
- Controles Preventivos: utilizados para prevenir erros, omissões ou atos fraudulentos.
- Controles Detectivos: usados para detectar erros, omissões ou atos fraudulentos.
- Controles Corretivos: servem para reduzirem impactos ou corrigir erros uma vez detectados.
- Achados da Auditoria: são fatos significativos e relevantes baseado em fatos e evidências observados pelo auditor e sua equipe durante a execução da auditoria.
- Papéis de Trabalho: são registros que evidenciam atos e fatos observados pelo auditor.
- Recomendações da Auditoria (Relatórios): são medidas corretivas possíveis, sugeridas pela instituição fiscalizadora ou pelo auditor e sua equipe em seu relatório, para corrigir as deficiências detectadas durante a auditoria.

A natureza e os tipos mais comuns da auditoria, conforme [Dias 2000], são:
Órgão Fiscalizador:

- Auditoria interna: realizada por órgão interno da entidade.
- Auditoria externa: executada por instituição externa e independente da entidade fiscalizada.
- Auditoria articulada: trabalho conjunto de auditorias internas e externas.

Forma de Abordagem do Tema:

- Auditoria horizontal: aborda tema específico, realizada em várias entidades ou serviços paralelamente.
- Auditoria orientada: focada em uma atividade específica qualquer ou em atividade com fortes indícios de erros ou fraudes.

Tipo ou Área Envolvida:

- Administrativa, contábil, financeira, operacional, sistemas, desenvolvimento de sistemas, entre outras.

2.1.2. Auditoria de Sistemas

A auditoria de sistemas avalia o ambiente de processamento de dados para identificar e avaliar os possíveis riscos (acesso indevido, erros, falhas, irregularidades, ineficiências, etc.) que estejam ocorrendo, ou que possam ocorrer, e faz recomendações para correção e melhoria dos controles para a diminuição do grau dos riscos levantados [Silva Júnior 2001].

A auditoria de sistemas, segundo [Silva Júnior 2001], objetiva certificar-se de que:

- Informações são corretas e oportunas.
- Existe um processamento adequado e correto das operações.
- Informações são bem protegidas, por exemplo, contra fraudes.
- Tem proteção das instalações e dos equipamentos.
- Existe a proteção contra situações de emergências (paralisação de processamento, perda de arquivos, inundações, incêndios, entre outros).

O objetivo do uso de técnicas e ferramentas é o auxilio ao profissional da área a auditar totalmente a população da área ou transação revidada, considerando o limite de tempo que possui, aproveitando os recursos de *softwares* e as técnicas de auditoria em ambiente de computação. Essas técnicas e ferramentas são importantes, pois auxiliam na avaliação de ambientes e sistemas, que geralmente processam um grande volume de dados e transações [Imoniana 2005]. Estas ferramentas e técnicas da auditoria de sistemas podem ser aplicadas na auditoria do desenvolvimento de sistemas, por isso a importância do seu estudo.

2.1.3. Desenvolvimento de Sistemas e a Auditoria

O sistema é um conjunto de elementos inter-relacionados com o objetivo de produzir relatórios que coordenam a tomada de decisões gerenciais [Imoniana 2005].

Sistemas de informação e seu desenvolvimento compreendem um conjunto de recursos humanos, materiais, tecnológicos e financeiros combinados segundo uma seqüência lógica para transformar dados em informações [Gil 2000].

A auditoria do desenvolvimento de sistemas, foco principal deste artigo, objetiva avaliar a adequação das metodologias e procedimentos de projeto, desenvolvimento, implantação e revisão dos sistemas produzidos dentro da organização auditada. Essa avaliação pode abranger apenas o ambiente de desenvolvimento da organização ou prever também a análise do processo de desenvolvimento de um sistema específico, ainda na fase de planejamento, já em andamento ou após conclusão [Brasil 1998].

Para a realização da auditoria na engenharia de software e necessário compreender e fiscalizar o ciclo de desenvolvimento de um sistema, conforme a Figura 2, e para isto utilizar processos, técnicas e ferramentas que venham a auxiliar e facilitar o auditor.

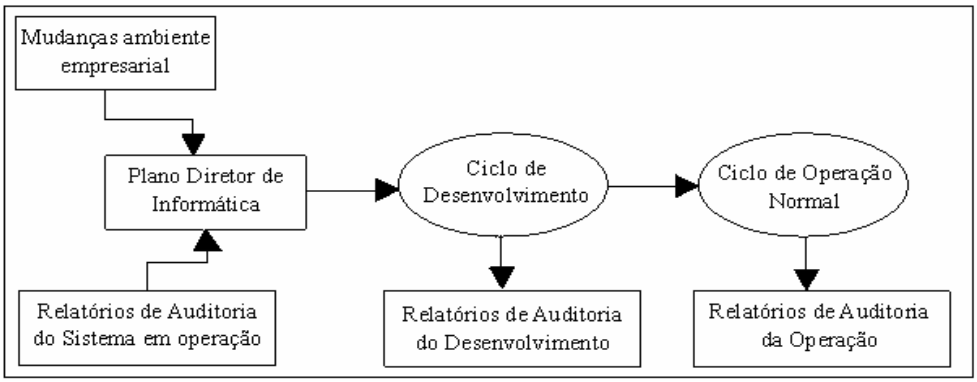


Figura 2. Participação da auditoria no ciclo de vida do sistema
Fonte: [Gil, A. 2000]

2.1.4. Mecanismos de Segurança

Os incidentes de segurança são ocorridos por vários motivos e causas, como: ex-funcionário insatisfeito, vírus de computadores, falhas de *hardware*, sobrecarga elétrica, desastres naturais (incêndio, terremoto, enchente), falhas estruturais, sabotagem, fraudes, acessos não autorizados (*hackers*, espionagem industrial, venda de informações confidenciais para a concorrência), entre outros [Dias 2000].

Por isso, segundo [Dias 2000], a auditoria almeja alguns objetivos de segurança, como: confidencialidade, integridade, disponibilidade, consistência, isolamento, auditoria e confiabilidade:

Para se conseguir alcançar estes objetivos são utilizados mecanismos de segurança, onde este é o meio mais utilizado para atender a um serviço de segurança, isto é, para prover e suportar serviços de segurança.

Existem vários mecanismos de segurança, conforme [Dias 2000], como: sistemas criptográficos; assinatura digital; mecanismos de controle de acesso; mecanismos de integridade de dados; mecanismos de disponibilidade; trocas de autenticações; enchimento de tráfego e controles de roteamento.

Neste artigo será utilizado os sistemas criptográficos como mecanismo de segurança, que utilizam criptografia ou algoritmos cifrados para proporcionar confidencialidade, integridade e isolamento de dados e de informações.

2.1.5. Criptografia de Dados

A criptografia é estudo de métodos para esconder o conteúdo de mensagens ou dados armazenados. O processo de cifragem ou criptação é a transformação da mensagem original em algo ininteligível, utilizando um código secreto – a chave criptográfica. A decifragem ou decriptação, por sua vez, é o processo inverso, onde se tem a recuperação da mensagem original a partir de sua forma cifrada.

Os componentes básicos de um sistema criptográfico, mostrado na Figura 3, são: texto claro, algoritmo, chave e texto cifrado [Dias 2000].

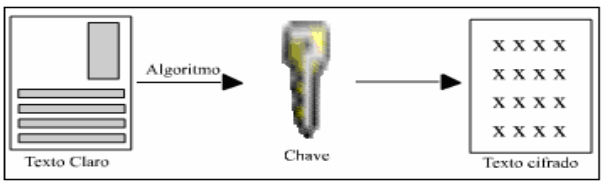


Figura 3. Componentes básicos de um sistema criptográfico
Fonte: [Dias, C. 2000]

O texto claro corresponde á mensagem original. O texto cifrado, também chamado de criptograma, corresponde ao texto claro após ter sido submetido ao processo de criptografia executado por determinado algoritmo. A chave criptográfica é uma chave secreta utilizada no processo de cifragem e decifragem de mensagens. O algoritmo é uma seqüência de passos e operações matemáticas que transforma o texto claro em texto cifrado, e vice-versa.

A criptografia se divide em dois tipos: simétrica: a chave assume papel fundamental neste tipo de criptografia, por ser ela a responsável por cifrar e decifrar uma mensagem, e a assimétrica: um par de chaves é emitido, onde somente uma das chaves pode decifrar o que a outra cifrou [Silva Junior 2001].

Os algoritmos de criptografia mais conhecidos são o *Data Encryption Standard* (DES), o *International Data Encryption Algorithm* (IDEA), a família *Message Digest* (MD) e o *Rivest, Shamir - Adleman* (RSA) [Dias 2000].

Este último algoritmo será detalhado neste artigo, devido a escolha para implementação na ferramenta de auditoria no desenvolvimento de sistemas – AudiSoft.

2.1.6. Algoritmo RSA

O RSA é um sistema de criptografia de chave assimétrica, ou criptografia de chave pública, inventado por volta de 1977 pelos professores *Ronald Rivest*, *Adi Shamir* e *Leonard Adleman*.

O sistema consiste em gerar uma chave pública (geralmente utilizada para cifrar os dados) e uma chave privada (utilizada para decifrar) por meio de números primos grandes, o que dificulta a obtenção de uma chave a partir da outra [Moreno, Pereira e Chiaramonte 2005].

O Algoritmo RSA é muito utilizado e testado, mostrando-se extremamente forte se for adequadamente usado. Isto devido ao fato de que é extremamente difícil fatorar números muito grandes [Carvalho 2001].

Para criação das suas chaves privada e pública, segundo [Buchmann 2002], a escolha das chaves começa escolhendo dois números primos aleatórios p e q e calcula-se o produto. Sejam então:

$$n = p * q$$

$$\phi = (p - 1) (q - 1)$$

Escolhe-se agora um número aleatório $e > 1$ tal que o máximo divisor comum dos dois inteiros positivos – $MDC (e , \phi) = 1$, e e ϕ devem ser primos entre si. Calcula-se então o número $d (1 < d < \phi)$ tal que:

$$(e * d) \text{ Módulo } \phi = 1$$

A chave pública se constitui dos números [e , n], e a chave privada dos números [d , n].

A Cifragem de uma mensagem se faz possível através da chave pública utilizada para cifrar dados [Buchmann 2002]. Dessa forma, uma mensagem w, para se cifrar w em c faz-se:

$$c = w^e \text{ Módulo } n$$

A Decifragem se faz admissível por meio da chave privada empregada para decifrar dados [Buchmann 2002]. Para decifrar faz-se:

$$w = c^{d} \text{ Módulo } n$$

O exemplo ilustrado na Figura 4, mostra como o algoritmo RSA funciona exatamente, desde a criação das chaves pública e privada, até cifragem e decifragem da mensagem.

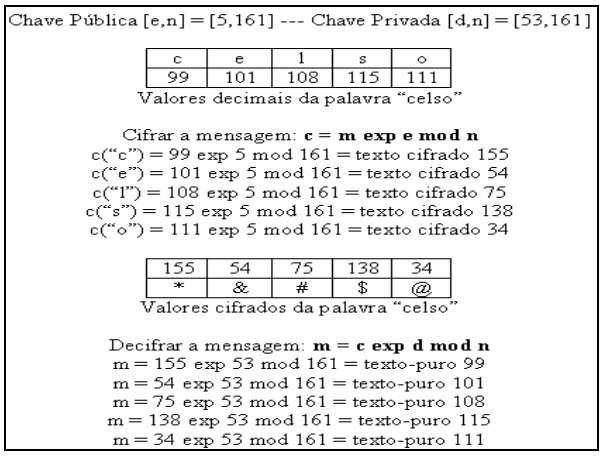


Figura 4. Exemplo da utilização do algoritmo RSA

2.2. Ferramenta AudiSoft

AudiSoft analisa passo a passo as etapas e processos existentes na engenharia de software, sendo possível anotar evidências e achados em cada etapa e também propor um solução, uma recomendação sobre o que foi encontrado.

A segurança de dados, através da criptografia de dados com o algoritmo RSA, entra no momento em que o auditor vai gerar o relatório com todos os pontos abordados no AudiSoft a respeito sobre o desenvolvimento do sistemas analisado, pois neste relatório estarão informações importantes e que não podem ter acesso não autorizado.

2.2.1. Recursos e Ferramentas Necessárias

A linguagem empregada para a implementação do AudiSoft foi o C++ Orientado a Objetos com o auxilio da ferramenta de desenvolvimento *Borland C++Builder 6*.

O Banco de Dados utilizado foi *Firebird 1.5 Server Manager* e o *EMS QuickDesk 2.0* para editar e criar todos objetos de um banco. Além disso, foi necessária a instalação do *BDE Administrator*, para conectar o *C++ Builder 6* com uma base de dados qualquer. Também foi necessário o *Firebird ODBC Driver* que contém o *driver* do *Firebird* para acessá-lo via BDE.

O arquivo de Ajuda do AudiSoft foi empregado o uso da ferramenta *Microsoft Help WorkShop* (*Help Workshop* e o *Dialog Box Help Editor*).

Para modelagem da ferramenta AudiSoft foi utilizado o *Pacestar UML Diagrammer* e o *DBDesigner4.*

Por fim, para a criação do instalador do AudiSoft foi utilizado o *Inno Setup*, juntamente com o *ISTool*, um programa que auxilia na criação de scripts para o *Inno Setup Compiler*.

2.2.2. Modelagem

A ferramenta de auditoria AudiSoft foi desenvolvida para atender a dois tipos de usuários: o cliente e o auditor. O primeiro, o cliente é o proprietário ou responsável pela engenharia de software que teve seu conteúdo auditado pelo AudiSoft. Entretanto, o segundo usuário é o auditor, onde este tem acesso ilimitado a ferramenta, através de seu *login* e senha, conseguindo ter autorização a todos os recursos da implementação.

Os Diagramas de Caso de Uso, ilustrado na Figura 5, demonstram todas as atividades e permissões oferecidas para os usuários auditor e cliente.

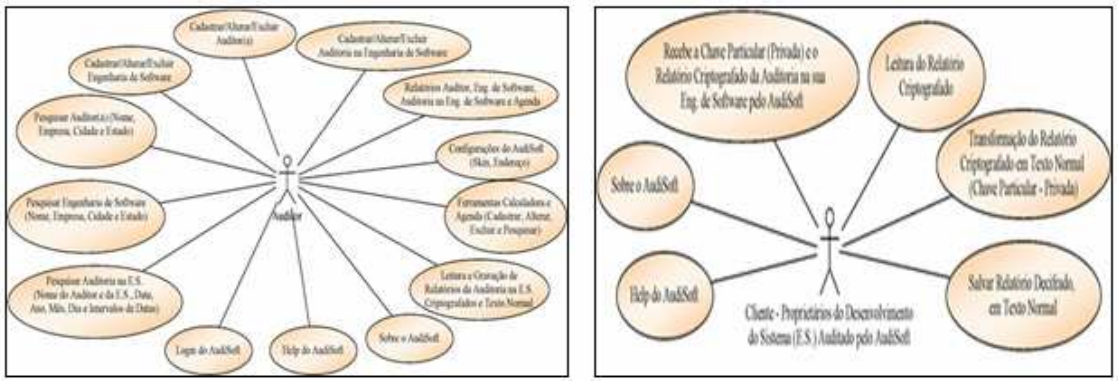


Figura 5. Diagramas de Caso de Uso do Auditor e Cliente – Responsável pelo ES

2.2.3. Implementação AudiSoft

O AudiSoft, ferramenta para auditoria em engenharia de software com criptografia de dados, tem seu funcionamento e composição dividido conforme Figura 6, em:

- inicio: *login* do AudiSoft, configurações, ajuda e sair;
- cadastro: cadastro do auditor e da engenharia de software (desenvolvimento do sistema);
- ação: cadastro da auditoria na engenharia de software;
- pesquisa: consulta auditor, engenharia de software e auditoria na ES;
- relatório: relatórios sobre o auditor, a engenheria de software, a auditoria na ES e as anotações da agenda do auditor;
- resultados: geração dos resultados da auditoria na ES, podendo ser feita a cifragem e decifragem e a leitura e gravação dos relatórios;
- ferramentas: calculadora do Windows e agenda de anotações do auditor;
- sobre e sair: o primeiro contém dados do AudiSoft, do desenvolvedor, informações do computador onde está instalada a ferramenta e créditos, o segundo fecha a ferramenta AudiSoft.

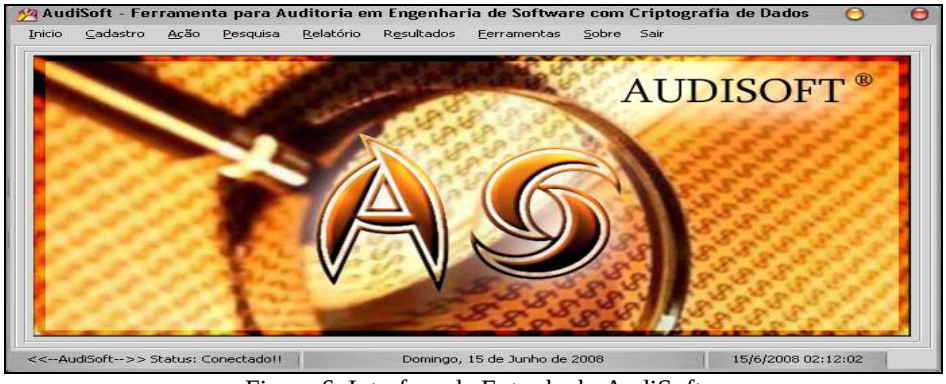


Figura 6. Interface de Entrada do AudiSoft

A ação principal da ferramenta AudiSoft é auditar o desenvolvimento de sistema, conforme a Figura 7, para isto é necessário que o desenvolvimento do sistema é o auditor já estejam cadastrados. Neste formulário são preenchidos os papéis de trabalho da auditoria, com evidências, achados, controles, irregularidades, resultados, recomendações, pontos fortes e fracos, melhorias, sugestões e situação de cada etapa e processo envolvido na engenharia de software. Para facilitar o trabalho de fiscalização a engenharia de software foi dividida em suas etapas e processos e um arquivo de ajuda foi disponibilizado.

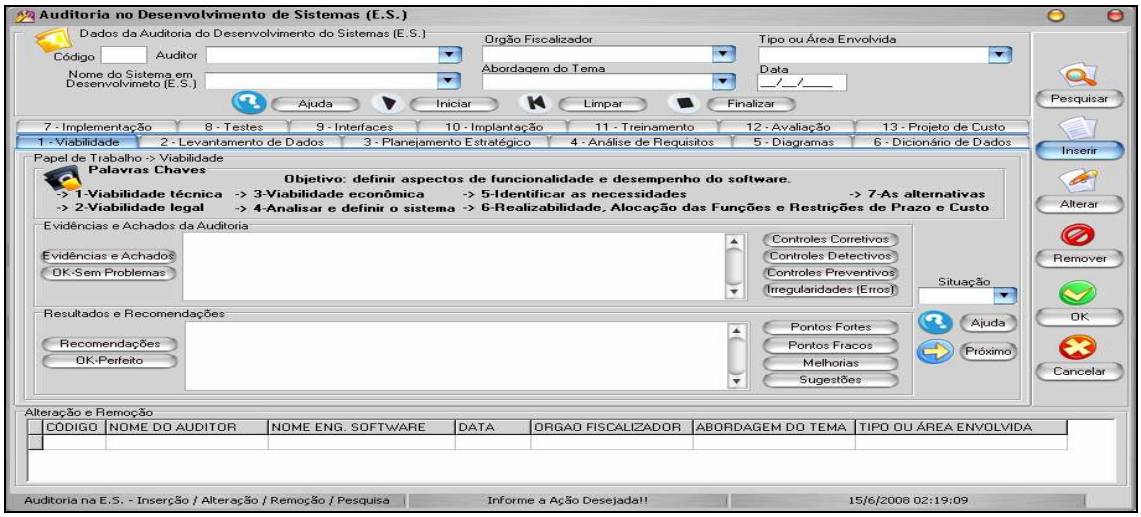


Figura 7. Auditoria na Engenharia de Software (ES)

Na área de Resultados da ferramenta AudiSoft encontra-se a utilização da criptografia de dados juntamente com algoritmo RSA. Esta questão é muito importante e tem um grande destaque junto com auditoria neste artigo.

Nesta parte da ferramenta AudiSoft são gerados os relatórios de auditoria de uma determinada engenharia de software feitos pelo auditor. Os relatórios criados podem ser cifrados e decifrados através do algoritmo RSA (chave pública e particular) e também serem lidos ou gravados, conforme mostrado na Figura 8.

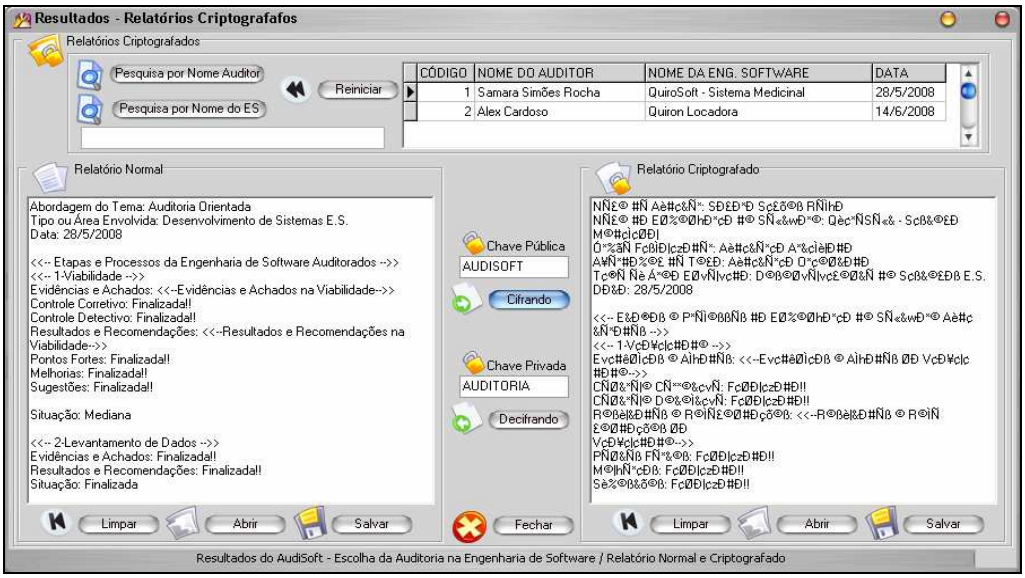


Figura 8. Resultados - Relatórios Cifrados e Decifrados (Algoritmo RSA)

2.3. Resultados Obtidos

O resultados obtidos foram a ferramenta AudiSoft para auxiliar o Auditor no trabalho da Auditoria na Engenharia de Software, através de papéis de trabalho de Auditoria para cada etapa e processo da ES com intuito de mostrar e apontar possíveis problemas e suas soluções.

Outro grande resultado foi a criptografia de dados, através do algoritmo RSA, com mecanismo de segurança do AudiSoft, onde os resultados (relatórios) emitidos pela ferramenta são cifrados e/ou decifrados para maior segurança.

3 – Considerações Finais

Com este artigo conclui-se que o trabalho da auditoria se torna indispensável, pois facilita e agiliza os processos e as etapas de desenvolvimento e engenharia de software. Consequentemente resulta em um produto final (Software) mais adequado, correto e confiável.

Além disso, a ferramenta AudiSoft se mostrou segura e confiável através da criptografia de dados (algoritmo RSA) como mecanismo de segurança da Auditoria, atendendo as necessidades de confidencialidade, proteção e sigilo dos dados manipulados e coletados na Ferramenta.

No desenvolver do artigo algumas dificuldades foram encontradas, como:

- pouco material publicado sobre auditoria na área de engenharia de software e desenvolvimento de sistemas;
- no desenvolvimento do AudiSoft, a ação de Auditorar a ES envolve e relaciona muitas tabelas e informações do banco de dados com a ferramenta;
- implementação do algoritmo RSA, devido aos cálculos matemáticos de exponencial e funções modulares, havia inexatidão da ferramenta de desenvolvimento com cálculos matemáticos devido aos números volumosos.

Nos trabalhos futuros constatam-se os seguintes pontos importantes:

- implementar uma ferramenta de auditoria que aplique a engenharia reversa, ou seja, a partir de uma aplicação pronta, um programa executável, consiga-se ter acesso a dados e a lógica de programação do software para ser auditado;
- a ferramenta AudiSoft ou a parte da criptografia fosse desenvolvida em uma outra linguagem ou ambiente de programação que desse suporte as cálculos matemáticos do algoritmo RSA e que já fosse multiplataforma (pode ser executado em outros sistemas operacionais);
- a necessidade de testar o sistema AudiSoft, para por em prática a sua utilização.

4 – Referências

Brasil, Tribunal de Contas da União (1998). **Manual de Auditoria de Sistemas /** Tribunal de Contas da União. Brasília: TCU, Secretaria de Auditoria e Inspeções.

Buchmann, Johannes A. (2002). **Introdução à criptografia.** São Paulo: Berkeley.

Carvalho, Daniel Balparda de (2001). **Segurança de Dados com Criptografia:** Métodos e Algoritmos. 2. ed. Rio de Janeiro: Books Express.

Dias, Cláudia (2000). **Segurança e Auditoria da Tecnologia da Informação.** Rio de Janeiro: Axcel Books.

Gil, Antonio de Loureiro (2000). **Auditoria de Computadores.** 5. ed. São Paulo: Atlas.

Imoniana, Joshua Onome (2005). **Auditoria de sistemas de informação.** São Paulo: Atlas.

Moreno, Edward David; Pereira, Fábio Dacêncio; Chiaramonte, Rodolfo Barros (2005). **Criptografia em Software e Hardware.** São Paulo: Novatec.

Silva Junior, José Barbosa da (2001). **Auditoria em ambiente de internet.** São Paulo: Atlas.