REGES BOTÃO BOEGERSHAUSEN

Aplicação de uma Proposta de Protocolo de Segurança para um Sistema Seguro de Atendimento ao Cliente

Joinville, Novembro de 2006

UNIVERSIDADE DO ESTADO DE SANTA CATARINA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Reges Botão Boegershausen

Aplicação de uma Proposta de Protocolo de Segurança para um Sistema Seguro de Atendimento ao Cliente

Trabalho de conclusão de curso submetido à Universidade do Estado de Santa Catarina como parte dos requisitos para obtenção do grau de Bacharel em Ciência da computação Integral (TCC-II).

Orientadora: Luciana Rita Guedes

Joinville, Novembro de 2006

Aplicação de uma Proposta de Protocolo de Segurança para um Sistema Seguro de Atendimento ao Cliente

Reges Botão Boegershausen

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Bacharel em Ciência da Computação – Sistemas de Computação e aprovada em sua forma final pelo Curso de Ciência da Computação Integral do CCT/UDESC. TCC-II

_____
Prof. Roberto Silvio Ubertino Rosso Jr, Dr.

BANCA EXAMINADORA:

_____
Profa. Luciana Rita Guedes, M.Sc.
Orientadora

_____
Prof. Charles Christian Miers, M.Sc.

_____
Prof. Maurício Aronne Pillon, Dr.

_____
Prof. Kariston Pereira, M.Sc.

“É melhor tentar e falhar,
que preocupar-se e ver a vida passar,
é melhor tentar, ainda que em vão,
que sentar-se fazendo nada até o final.”
Martin Luther King

DEDICATÓRIA

Ao pai Renato e minha mãe Geralda (in memorian), por me incentivarem e acreditarem que essa etapa seria vencida.
A minha noiva Angela, pela paciência, compreensão e amor.

AGRADECIMENTOS

Agradecemos a Deus por ter-nos dotado de inteligência e através dela, nos permitido o acesso ao conhecimento.

A minha Orientadora Professora Luciana Rita Guedes, por estimular nosso desenvolvimento e aperfeiçoamento.

Ao meu grande amigo Giovani que teve uma contribuição fundamental para a elaboração do *layout* da aplicação.

Aos meus amigos que me ajudaram a espairecer e a manter a calma nos momentos de estresse e nervosismo no desenvolvimento deste trabalho.

A todos aqueles que, de alguma forma, contribuíram e apoiaram desde a pesquisa até a conclusão deste trabalho.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

SAC – Sistemas de Atendimento ao Cliente

AD – Autoridade de Datação

AF – Autoridade Fiscalizadora

AC – Autoridade Certificadora

AA – Autoridade de Aviso

E – Cifrar

FEC – Fechamento

D – Decifrar

K – Chave

KR – Chave Privada

KU – Chave Pública

REQ – Requisição

RESUMO_REQ – Resumo da Requisição

PHP - PHP: *Hypertext Preprocessor*

OS – Ordem de Serviço

RR – Recibo da Requisição

RF – Recibo de Fechamento

SGBD – Sistema Gerenciador de Banco de Dados

W3C – *World Wide Web Consortium*

RPC – *Remote Procedure Call*s

XML – *eXtensible Markup Language*

ODBC – *Open Data Base Connectivity*

ADO – *ActiveX Data Objects*

LGPL – *Lesser General Public License*

SUMÁRIO

RESUMO

O trabalho proposto nesse documento trata do estudo e da aplicação do protocolo criptográfico proposto por Amauri Sant'Anna Ghisleri para o desenvolvimento de Sistemas Seguros de Atendimento ao Cliente, em sua dissertação de mestrado pela Universidade Federal de Santa Catarina.

Palavras-chave: Sistema de Atendimento ao Cliente, Segurança Computacional, Protocolo Criptográfico.

ABSTRACT

The work considered in this document is to the study and the application of the cryptographic protocol developed by Amauri Sant'Anna Ghisleri for the development of Safe Systems of Attendance the Customer, in its dissertation by the Federal University of Santa Catarina.

Key Words: Security, cryptographic protocols, costumer services.

INTRODUÇÃO

As redes de computadores e a Internet têm sido cada vez mais utilizadas devido à sua evolução e necessidade. Em todos os níveis, desde grandes empresas a usuários domésticos, as redes de computadores e a Internet são úteis para diversos fins, seja para um simples acesso a uma página de informações até transações bancárias. Confrontando com esse crescimento de volume de dados e de usuários a segurança desse meio torna-se cada vez mais necessária (SACERDOTE, 2005).

Por conseqüência, ocorreu a necessidade de redefinir a formalização do conceito de segurança, para que este se adequasse à “Era da Informação”. Sendo assim, descreveu-se formalmente os requisitos de segurança como sendo: confidencialidade, integridade, disponibilidade e autenticidade (MACHADO, et al. 2000), (SCHNEIER,2001).

Para que estas premissas fossem de fato respeitadas, pesquisadores do meio acadêmico, profissionais e órgãos ligados à área de segurança das redes desenvolveram normas, padrões e ferramentas que devem ser utilizados para que todo o projeto possa ser definido usando critérios de segurança (MACHADO, et al. 2000), (BERNARDES, 1999).

Dentre as ferramentas utilizadas, têm-se os algoritmos criptográficos, que se fundamentam na criptografia. A criptografia tem como principal objetivo “embaralhar” o conteúdo de informações para evitar que pessoas não-autorizadas possam compreender esse conteúdo, ou seja, pessoas que não tenham conhecimento da chave de criptografia não podem decifrar tal mensagem (TERADA, 2000).

Para FLEURY (2002) a criptografia tem sua utilidade muito restrita se for utilizada isoladamente. Porém, sua utilidade aparece realmente quando usada para resolver determinado problema como, por exemplo, autenticação de usuários. Partindo desse princípio surgiram os protocolos criptográficos.

Protocolos criptográficos são aqueles que utilizam criptografia em pelo menos um de seus passos, geralmente com alguma aplicação mais complexa que o simples ato de cifrar e decifrar textos. Alguns exemplos de aplicações de

protocolos criptográficos são: assinatura digital, troca de mensagens, autenticação, votação eletrônica e sistemas de atendimento ao cliente, sendo este último o foco do presente trabalho (FLEURY, 2002), (TERADA,2000).

O uso da criptografia em sistema de atendimento ao cliente visa garantir a integridade e confidenciabilidade das informações trocadas durante a sessão entre cliente e empresa.

Existem vários tipos de sistemas de atendimento e relacionamento com clientes: esclarecimento de dúvidas de produto/serviço, sistemas para ouvir reclamações de pós-venda de produtos/serviços e os próprios sistemas de venda e contratação de produtos/serviços. Estes tipos de sistemas auxiliam uma empresa a garantir a satisfação total do cliente tentando manter sua fidelidade (NAKAMURA, 2001).

Um Sistema de Atendimento ao Cliente (SAC) visa abrir um canal direto de comunicação entre uma empresa e o cliente, possibilitando assim que o mesmo possa fazer sugestões ou emitir opiniões sobre os produtos ou serviços que ele adquiriu (SEBRAE, 2005). Esse serviço pode ser disponibilizado através de várias maneiras de modo que facilite o contato do cliente com a empresa como, por exemplo, via telefone ou ainda através do *websites* da empresa (NAKAMURA, 2001). Dentre os tipos de SAC existentes deve ser citado o *helpdesk*, que é um SAC de suporte técnico, onde o cliente "conversa" com agentes especializados em vários níveis do produto ou serviço e para também diferentes níveis de clientes, fornecendo orientações para sanar eventuais dúvidas ou problemas do cliente (GHISLERI, 2003).

Para sistemas de atendimento ao cliente disponíveis em *websites* surge a especial necessidade de assegurar confiabilidade ao processo de atendimento de forma que o cliente receba o atendimento de qualidade o mais breve possível. O protocolo criptográfico proposto por Ghisleri (GHISLERI, 2002) tem o objetivo de garantir a segurança no processo de atendimento ao cliente usando a internet como meio de comunicação.

Este trabalho, que visa implementar o protocolo criptográfico citado, está organizado da seguinte maneira:

- Capítulo 1: Trata sobre os conceitos de atendimento ao cliente, canais de atendimento e alguns tipos de sistemas de atendimento;

- Capítulo 2: Explora os conceitos sobre segurança computacional enfatizando a criptografia;
- Capítulo 3: É o estudo sobre o protocolo proposto por Amauri Sant' Anna Ghisleri;
- Capítulo 4: Trata sobre a implementação da aplicação do protocolo bem como as tecnologias utilizadas, discussão sobre a implementação, teste e resultados obtidos.

A metodologia utilizada para o desenvolvimento deste trabalho baseia-se em pesquisas e levantamentos bibliográficos sobre os conceitos de segurança computacional, criptografia e atendimento ao cliente, juntamente com a proposta de Sistema Seguro de Atendimento ao Cliente presente em uma dissertação de mestrado. Após a realização destes estudos serão pesquisadas tecnologias para que possa ser implementada a aplicação com as características de um Sistema Seguro de Atendimento ao Cliente. Ao final, será apresentada a implementação da aplicação.

1 – ATENDIMENTO AO CLIENTE

Existem na literatura vários conceitos extremamente amplos para definir clientes. NICKELS & WOOD (1999) defendem que cliente é: “indivíduo ou organização que compra ou troca algo de valor pelos serviços e/ou produtos da empresa”. Sendo assim, cliente não deve ser visto apenas como o consumidor final do produto e/ou serviço da organização, mas os elementos internos da mesma também são potenciais clientes já que estes em certos casos fazem esse “escambo” com a empresa (PINTAUD, 2002).

A partir disso, as organizações começam a vislumbrar que o cliente, e não mais seus produtos, pesquisas ou outras competências, deve ser o foco das estratégias do negócio para que esse atinja os desejados níveis de lucratividade, vendas, crescimento, etc. (NEZZE, 2002). Para colocar em prática o que essa nova visão diz, as empresas estão reconhecendo a verdadeira importância e dando a devida atenção aos clientes e aos clientes em potencial.

Para que as empresas possam sobreviver ao mercado cada vez mais competitivo o cliente deve ser considerado como o início e o fim de todas as suas atividades. Com isso, além das empresas conquistarem novos clientes, não só com produtos ou serviços de qualidade e preços atrativos, devem apresentar canais de comunicação com a empresa para que sejam esclarecidas as diversas dúvidas que possam surgir dos clientes sobre produtos ou serviços oferecidos pela empresa (ANDERSEN & KERR, 2003).

A satisfação do cliente é responsabilidade da organização como um todo, bem como manter canais de comunicação permanentes com o cliente que coletem seus dados de modo a contribuir com a avaliação que este faz sobre a empresa e assim orientar suas ações, além da melhoria da produtividade e da qualidade de serviços oferecidos (GADELHA, 2002).

No conceito de CRM (*Customer Relationship Manager*), o qual é um processo amplo e multitarefado, não é difícil criar produtos e serviços que pessoas não queiram comprar ou contratar pelo menos uma vez. O que realmente é difícil é gerar uma organização ou sistema onde seja possível manter os consumidores voltando sempre, já que o mercado atual oferece

inúmeras empresas que oferecem produtos e serviços numa paridade crescente. Para tentar ganhar esses clientes é preciso que toda a organização trabalhe vislumbrando essa meta (ELSENPETER & VELTE, 2002).

Através do CRM a empresa pode adotar uma estratégia mais eficaz para aumentar a satisfação do cliente em relação ao atendimento, adotando comportamentos organizacionais voltados para o cliente, implementando processos e tecnologias que suportem interações com os clientes através de canais de atendimento e relacionamento (SCHWEITZER, 2004). A solução CRM para atender esta estratégia deve conter: suporte a vendas, serviço pós e pré-vendas para o cliente, suporte a *marketing* personalizado ao cliente (PINTAUD,2002). Enfim, CRM não é um apenas um software ou conjunto de softwares, mas sim toda uma filosofia que envolve diversas áreas, pessoas, processos e tecnologia da informação que auxiliam para melhor se relacionar com os clientes.

O primeiro passo para fidelização de cliente é a sua satisfação com os aspectos de comunicação oferecidos pela empresa, o segundo passo que é pauta de inúmeras reuniões da alta diretoria das empresas e que deve fazer parte da qualidade do serviço e que deve se tornar uma prioridade é a questão do atendimento ao cliente (GHISLERI, 2003).

O conceito de qualidade no atendimento ao cliente não é novo, surgiu no início do século XX, proposto por economistas ingleses através de pesquisas econômicas da teoria do comportamento do consumidor (PINTAUD, 2002). E as empresas que pretendem alcançar maior competitividade no mercado vêm se mostrando cada vez mais interessadas neste tema, para que haja sucesso de qualquer empreendimento, já hoje os clientes são mais difíceis de agradar. Eles estão mais inteligentes, perdoam cada vez menos as falhas cometidas, estão mais conscientes em relação a preços e a concorrência que tem ofertas sempre muito próximas ou melhores (SCHEUER, 2001).

Para atender bem os clientes a organização deve entender o que o cliente quer, e disponibilizar serviços de atendimento pré-venda, que devem conter boa quantidade de informações de ordem técnica e não-técnica sobre o produto, tais como: descrição do produto através de páginas de conteúdo (similar a folhetos publicitários); apresentações e demonstrações do produto utilizando animações gráficas e vídeos; publicações como o manual do

produto, o qual dará mais informações sobre o mesmo para o cliente se certificar da sua escolha. O serviço de atendimento pós-venda pode disponibilizar formas de o consumidor sanar suas dúvidas mais comuns como, por exemplo, páginas com dúvidas e perguntas mais freqüentes, dicas de manutenção e utilização do produto adquirido, entre outros (ANDERSEN & KERR, 2003).

Uma maneira eficaz para atender o cliente é a sistematização de sua voz, permitindo o estabelecimento de um canal contínuo e constante entre a empresa e o cliente. Mas, para que isso aconteça deve haver uma excelente coordenação entre a coleta e uso de dados para atingir um atendimento de qualidade (GADELHA, 2002). A partir dessa abordagem começou-se a dar mais atenção aos canais de comunicação cliente-empresa já existentes e também foram surgindo novos canais aliados ao avanço da tecnologia. Alguns dos principais canais de comunicação serão explicados na próxima seção deste trabalho.

1.1 - Canais de Atendimento

São várias formas utilizadas para o atendimento ao cliente, seja para ouvir reclamações pós-venda, esclarecer dúvidas sobre produtos ou serviços que desejam e até mesmo para a própria contratação e compra de serviços e produtos respectivamente. A seguir serão explicados alguns canais de atendimento ao cliente tais como: Caixa de Coleta de Sugestões, Ligações Gratuitas 0800 e Sistemas de Ajuda (*Helpdesk*).

1.1.1 - Caixa de Coleta de Sugestões

O mais simples e barato método de atendimento, devido à sua popularidade é muito eficiente. Consiste numa urna disponibilizada para coleta de sugestões, críticas ou reclamações sobre o serviço prestado. Na maioria

dos casos, o cliente recebe um formulário apropriado no qual vai manifestar suas impressões no final do atendimento, este formulário pode servir em outros momentos, como cupom para sorteios (SHIBA, et al. 1997).

Esta é uma das mais populares e utilizadas formas de atendimento ao cliente, pois é muito simples, fornece resultados muito satisfatórios e pode ser utilizados para várias finalidades, como realizar uma pesquisa com clientes sobre a estrutura da empresa, por exemplo.

1.1.2 – Ligações Gratuitas 0800

Outra forma que se tornou muito popular para o atendimento ao cliente, é este serviço de chamadas telefônicas gratuitas. Muitas empresas estruturaram-se para que a única forma de atendimento seja via telefone. É o caso das empresas de “disk-pizza”, “tele-mensagens”, “disk-flores” que trabalham basicamente com sistema de tele-entrega. Este modelo de atendimento tornou-se popular graças à popularização das linhas telefônicas nas últimas décadas no país (GHISLERI, 2003). Outro fator que ajudou nesse crescimento é o fato desse tipo de ligação ser pago pela empresa que o disponibilizou e não pelo cliente que efetuou a ligação (NAKAMURA, 2001).

Este tipo de sistema tem outras vantagens, os serviços inteligentes, tais como (NAKAMURA, 2001):

- Área limitada atendida pelo serviço: Por se tratar de um recurso gratuito é comum que algumas empresas forneçam para a sua cidade um número convencional para as ligações desse local e um número no formato 0800 para as demais localidades, já que estes têm de pagar uma tarifa interurbana que é mais cara.
- Desvio de ligações para diferentes números conforme a área geradora da ligação: este serviço além de trazer economia nos custos de interurbano melhora muito o tráfego nas centrais telefônicas.

- Manutenção do 0800 mesmo com a mudança da sede da empresa: da mesma forma como se registra o Domínio para endereços Internet o número 0800 pode ser registrado, assim evitam grandes gastos caso a empresa precise mudar sua sede.

O formato de ligações 0800 fornece benefícios tanto para os clientes quanto para as empresas. As empresas podem utilizar este formato para efetuar vendas de seus produtos/serviços, sanar dúvidas de clientes em relação ao produto/serviço, realização de reclamações pós-venda do produto/serviço, entre outras, todas essas funcionalidades numa mesma estrutura interna. Já os clientes têm a comodidade de não precisarem se deslocar até a empresa para qualquer tipo de atendimento além de que, como já foi citado, os clientes não são tarifados por essa ligação.

1.1.3 – Sistemas de Ajuda (*Helpdesk*)

Helpdesks são sistemas de ajuda e suporte sobre o produto ou serviço que a empresa oferece ao usuário. A grande maioria desses sistemas estão localizados nos sítios das empresas como opção do tipo "*fale conosco"*. Estas opções geralmente levam a um formulário padrão da empresa onde o cliente é obrigado a preenchê-lo com alguns dados pessoais (exemplo: nome, telefone, *e-mail* e com o número do CPF ou RG, etc.) e alguns dados fornecidos pela empresa (exemplo: nome de usuário, senha, número do pedido ou produto) e tipo da requisição (dúvida, esclarecimento, reclamação, etc.) (SCHWEITZER, 2004).

Em alguns casos, estes formulários apenas postam as informações para um *e-mail* da empresa, que nem sempre dá o devido tratamento para a requisição, freqüentemente demora em dar retorno e, em alguns casos, simplesmente ignora esses pedidos de atendimento. Vale salientar que existem empresas que tratam de forma digna as requisições de clientes (SEBRAE, 2005).

1.2. Problemas com os Sistemas de Ajuda (Helpdesk)

Os sistemas de atendimento ao cliente do tipo *helpdesk* disponíveis nos sítios das empresas são o foco do protocolo desenvolvido por Amauri Ghisleri e, conseqüentemente, objeto de estudo do trabalho aqui proposto.

Vários problemas relacionados a este tipo de atendimento podem ser citados, principalmente no aspecto de garantia de segurança, seja por parte do cliente atendido ou por parte da empresa que o atende.

Por exemplo, caso aconteça de o cliente não ter seu pedido atendido da forma desejada, é necessário que ele tenha um canal direto e mais eficaz de comunicação com a empresa, onde o profissional que o atenda possa sanar seu problema (ELSENPETER & VELTE, 2002). Mesmo que este sistema funcione corretamente, a sua segurança pode ser questionada tanto pela empresa quanto pelo cliente, já que ambos desejam que as informações trocadas nessa transação sejam confiáveis (SCHWEITZER, 2004). Surgem também outras questões quando se fala sobre atendimento via Internet, tais como: O cliente é realmente quem ele diz ser? A empresa que está fornecendo o atendimento é também quem ela diz ser? Há formas de garantir que o atendimento seja efetuado no prazo estipulado? Pode-se garantir a integridade e o sigilo das informações trocadas entre as partes? Como garantir os direitos e deveres de ambas as partes no atendimento? (GHISLERI, 2003).

Com essas questões em mente (GHISLERI, 2003), propôs um protocolo (conjunto de regras) que une conceitos de atendimento ao cliente com ferramentas de criptografia, que será explicada no capítulo 2 deste trabalho. Caso este protocolo seja adotado pela empresa, vai ajudá-la a responder às questões feitas anteriormente. Obviamente existem outros problemas envolvidos quando se fala de atendimento ao cliente, porém este não é o foco deste trabalho.

1.2.1 Trabalhos correlatos em Sistemas de Ajuda (Helpdesk)

Os sistemas de ajuda mais utilizados nos sítios das empresas são os sistemas baseados em *chat on-line* e sistema por *ticket*.

Um dos sistemas de *chat on-line* pesquisado foi desenvolvido pela empresa *Webmedia Group, que* é fornecedora do software denominado *Cerberus LiveHelp*. O sistema deve ser instalado no sítio da empresa que pretende oferecer o serviço de helpdesk. Para fazer uso deste *helpdesk*, o cliente deve acessar o sítio da empresa e clicar no *link* que leva ao sistema. Em seguida, ele preenche um pequeno formulário com seu nome, endereço de *e-mail* e escolhe qual o departamento se encaixa com seu tipo de dúvida como, por exemplo, departamento de vendas, suporte técnico, entre outros. Caso alguma pessoa responsável pelo atendimento do departamento escolhido esteja disponível naquele momento, a conversa é iniciada e o usuário pode resolver seu caso imediatamente. . Se não houver ninguém disponível, o sistema gera um formulário para o usuário enviar um *e-mail* com a sua dúvida para a empresa. Esse e-mail é enviado para a caixa postal do departamento escolhido. Neste caso, o cliente deve esperar o responsável pelo departamento responder o *e-mail* para realizar o atendimento. Um detalhe importante é que nenhum tipo de comprovante é gerado em nenhuma das duas situações. A rapidez no atendimento é um ponto forte do sistema já que o usuário não precisa esperar muito tempo para obter respostas a respeito de suas dúvidas, especialmente nos casos em que recebe atendimento *online*. (WEBMEDIA, 2006).

Pôde-se perceber que neste sistema não existe nenhum tipo de preocupação com veracidade nos dados fornecidos pelo cliente (nome, *e-mail*, etc.), pois nenhum processo de autenticação é utilizado. Sendo assim, seria possível que uma pessoa mal-intencionada tentasse personificar a identidade de um cliente podendo, eventualmente, receber informações sigilosas com relação ao produto/serviço adquirido.

O sistema baseado em *tickets Craft Syntax Live Help* permite à empresa que o adquirir fazer algumas *customizações*. É o caso do formulário de solicitação de atendimento. Ao instalar o software para ser usado como sistema

de *helpdesk*, a empresa pode escolher os campos desejados para esse formulário, conforme sua necessidade.

Para utilizar o sistema, o cliente deve acessar o *link* disponível no sítio da empresa e preencher um formulário com a solicitação de atendimento. Após o envio, o cliente recebe um *ticket*, ou seja, um comprovante de que ele solicitou o atendimento. Esta solicitação fica registrada nos banco de dados do sistema para que possa ser realizado o procedimento interno. As solicitações de atendimento tornam-se disponíveis para os responsáveis por seus respectivos departamentos. O atendimento realizado é enviado para o e-mail que o usuário preencheu na requisição. Se o cliente não estiver satisfeito com a resposta dada pela empresa ele pode responder essa mensagem enviada para que o processo de atendimento possa continuar (CSLH, 2006).

Neste sistema um usuário mal-intencionado pode preencher o formulário com alguns dados falsos, por exemplo, e-mail de contato falso e com os outros dados verdadeiros (nome, endereço, etc.) e receber informações no lugar do verdadeiro cliente.

Também neste sistema não existe preocupação com a segurança dos dados e, tal como no sistema citado anteriormente, não são usadas conexões seguras para a troca de informações, não existe preocupação com a autenticação dos clientes. Por parte da empresa também existem falhas em ambos os casos: os sistemas não prevêem a utilização de certificados digitais para comprovar a sua identidade perante os clientes. Outro ponto que não é abordado é a preocupação de que os prazos de atendimento estão sendo respeitados.

Outro sistema baseado em *ticket* é o sistema de suporte a usuários da Coordenadoria de Informática (COINF) do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina (UDESC). O sistema, a exemplo daquele descrito anteriormente, baseia-se no preenchimento de um formulário pelo usuário que, após confirmar o envio, recebe um código denominado “ID do chamado”. Este código deve ser anotado pelo usuário pois através dele poderá ser visualizado o andamento da solicitação realizada. A exemplo dos demais sistemas, não há uso de critérios de segurança e nem garantia de atendimento ou de estabelecimento de prazos para atendimento da solicitação.

Outro sistema encontrado foi um trabalho de conclusão de curso da Universidade Federal de Santa Catarina – UFSC, desenvolvido juntamente com o trabalho de Mestrado de Amauri Sant'Anna Ghisleri. O trabalho é intitulado "Sistema Seguro de Atendimento ao Cliente via WEB" e seus autores são: Fauze Valério Polpeta e Handerson Koerich (POLPETA & KOERICH, 2002).

Neste trabalho, os autores utilizam uma biblioteca para criptografia proprietária da Microsoft e de funcionamento exclusivo no sistema operacional Microsoft Windows® denominada CAPICOM. A linguagem de programação Delphi, que foi utilizada no desenvolvimento de um dos módulos da aplicação, também é destinada ao sistema operacional Microsoft Windows®.

O presente trabalho pretende propor uma solução multi-plataforma e, com isso a aplicação desenvolvida por POLPETA & KOERICH (2002) torna-se diferente da aplicação que será aqui proposta. Além disso, o trabalho aqui proposto pretende demonstrar, através de testes que serão descritos, a viabilidade da aplicação construída.

Inúmeras propostas de protocolos criptográficos podem ser encontradas em trabalhos científicos. Alguns destes trabalhos foram desenvolvidos por pesquisadores ligados ao LabSec (Laboratório de Segurança em Computação). Entre estes trabalhos podemos citar o protocolo criptográfico para votação digital, desenvolvido por Augusto Jun Devegili. Há também o protocolo criptográfico para denúncia anônima segura desenvolvido por Fernando Cézar de Oliveira Lopes. Outro caso é o protocolo protocolo criptográfico para análise segura de crédito de Marcelo Luiz Brocardo. Todas estas propostas podem ser encontradas na sítio da internte do LabSec (LABSEC, 2006). Há muitos outros casos de propostas de protocolos criptográticos para inúmeras aplicações. No entanto, para o protocolo criado por Amauri Ghisleri, que é objeto de estudo deste trabalho, não foi encontrada nenhuma nova proposta além daquela supracitada.

2. SEGURANÇA COMPUTACIONAL E CRIPTOGRAFIA

A segurança tornou-se um aspecto fundamental diante da quantidade de informação que nos cerca e que é transmitida continuamente por meio eletrônico. Por conseqüência, ocorreu a necessidade de redefinir a formalização do conceito de segurança, para que este se adequasse a “Era da Informação”. Assim, o conceito de segurança aponta para a necessidade de garantir os seguintes aspectos: confidencialidade, integridade, disponibilidade e autenticidade (MACHADO, et al. 2000).

Esta nova concepção mostra a necessidade da criação e utilização de novas ferramentas que ajudem a alcançar este quatro aspectos citados. Dentre estas ferramentas está a criptografia.

A criptografia, de forma simples, visa “embaralhar” as informações de modo a torná-las incompreensíveis para pessoas não-autorizadas (TERADA, 2000). De uma maneira abrangente a criptografia divide-se em duas categorias: criptografia simétrica e criptografia assimétrica (SCHNEIER, 2001).

Neste capítulo será apresentada uma noção geral sobre os aspectos de segurança computacional, focando a criptografia, por tratar-se da principal ferramenta utilizada pelo protocolo que é objeto de estudo deste trabalho.

2.1 Conceitos de Segurança

No início, as redes de computadores foram utilizadas quase que exclusivamente por pesquisadores em universidades, para trocar informações através de correio eletrônico e nas empresas, para o compartilhamento de impressoras por seus funcionários. Nesse panorama de utilização das redes de computadores, a segurança não consistia em um elemento essencial. Atualmente, milhões de empresas e pessoas utilizam redes de computadores e

Internet com diversas finalidades, que vão de um simples acesso a uma página na Internet até transações bancárias, com isso um imenso número de informações trafegam mundialmente pelas redes de computadores, a segurança dessas redes tornou-se um requisito básico (SACERDOTE, 2005).

De uma forma simples a segurança baseia-se em quatro premissas (MACHADO et. al, 2000):

- **Confidenciabilidade:** Parte do princípio de que os dados só podem ser acessados por quem comprovadamente tem direitos e/ou privilégios de acesso, ou seja, esses dados necessitam de proteção para que não possam ser divulgados para pessoas com acesso não-autorizado (interna ou externamente);
- **Integridade:** Parte do princípio que os dados devem ser preservados fiéis ao autor, ou seja, deve-se proteger os dados contra todo tipo de modificações sem a explícita autorização do proprietário ou autor daquele dado;
- **Disponibilidade ou Não-Repúdio:** Parte do princípio que os dados devem estar sempre disponíveis para seus usuários legítimos, ou seja, os dados não devem ser degradados nem se tornarem indisponíveis sem devida autorização;
- **Autenticidade:** Parte do princípio que se deve comprovar sua identidade para poder ter acesso a determinados dados ou serviços, ou seja, deve-se fazer uma identificação correta de um usuário ou computador como medida de proteção de determinados serviços e/ou dados.

Na Norma NBR ISO/IEC 17799, os conceitos das premissas citadas anteriormente são muito semelhantes. Para os padrões descritos nessa norma a confidenciabilidade é ter garantia de que a informação só deve ser acessada por pessoas autorizadas. Já a integridade é a precisão e completude da informação para quem possui privilégios de acesso. Por sua vez, disponibilidade é a garantia de que os usuários autorizados possuam acesso à informação sempre que requisitado (ISO,2005).

Para que as premissas citadas anteriormente sejam de fato respeitadas foram desenvolvidas por pesquisadores, do meio acadêmico e por profissionais que trabalham com segurança, normas e padrões a serem seguidos. Inúmeras ferramentas também foram desenvolvidas para complementarem as normas, como é o caso dos *firewalls* e suas variantes, ferramentas de antivírus, ferramentas de detecção de intrusão, entre outras. Além dessas ferramentas baseadas em produtos de software e hardware que foram definidas, aliam-se outras técnicas, como por exemplo definição de políticas de segurança internas a uma organização e uso de criptografia a qual será explicada com maiores detalhes adiante, pois é o foco principal deste capítulo. Para que estas técnicas obtenham melhores resultados deve-se conhecer quais os possíveis ataques e ameaças que podem ser investidos contra uma organização.

2.1.1 Ameaças e Ataques

Existem inúmeras ameaças na segurança às quais redes de computadores ou sistemas de informação podem ser expostas. Segundo Stallings (2005) esses tipos de ameaças podem ser classificados da seguinte maneira:

- Interrupção: Ocorre quando o fluxo normal da mensagem é interrompido entre o emissor e o receptor da mensagem, isso pode ocorrer se alguma peça de hardware for destruída, exemplo disco rígido. Este é um de ataque de disponibilidade, pois torna a mensagem indisponível para o receptor;
- Interceptação: Ocorre quando uma entidade (pessoa ou programa de computador) consegue acesso não-autorizado à rede, fazendo cópias ilícitas de arquivos ou dados. Este é um tipo de ataque de confidenciabilidade;
- Modificação: Ocorre quando uma entidade (pessoa ou programa de computador) consegue acesso não-autorizado à

rede alterando conteúdo de arquivos, programas ou mensagens de forma a se comportarem diferentemente. Este é um tipo de ataque de integridade;

- Fabricação: Ocorre quando uma entidade (pessoa ou programa de computador) consegue acesso não-autorizado à rede e insere programas que geram novas informações ou que insiram informações não-autorizadas nos arquivos dispostos na rede. Este é um tipo de ataque de autenticidade.

As ameaças descritas anteriormente estão ilustradas na Figura 1. Também é demonstrado como o fluxo normal de uma mensagem deve proceder.

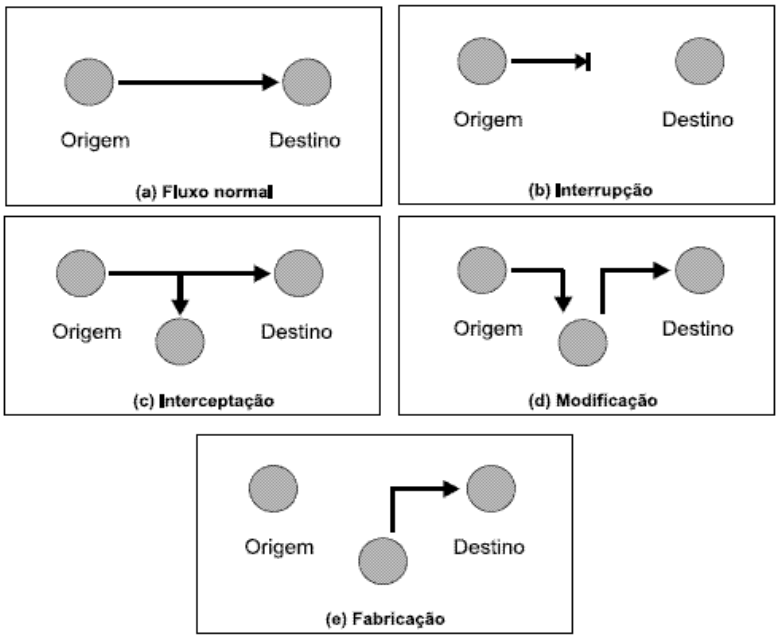

Figura 1 - Detalha o fluxo normal das informações (a) as ameaças a um sistema de computacional – Interrupção(b), Interceptação(c), Modificação (d) e Fabricação(e). Fonte (STALLINGS,1999).

Conforme Stallings (2005) os ataques são as caracterizações das ameaças, quando esses ocorrem são divididos em duas principais categorias: Ataques Passivos e Ataques Ativos.

Ataque passivo é caracterizado, por não alterar o conteúdo das informações, apenas “ouve” ou monitora as suas transmissões. Este ataque também é divido em duas subcategorias (STALLINGS,2005):

- Leitura do conteúdo da mensagem: Neste ataque o objetivo do é xeretar o conteúdo as informações trocadas entre origem e destino;
- Análise de trafego: Diferentemente do ataque anterior, caso o conteúdo da mensagem esteja protegido com o uso de alguma técnica preventiva, o atacante pode observar informações tanto do emissor e do receptor quanto das mensagens, como por exemplo, localização e identidade, freqüência e tamanho respectivamente.

Já os ataques ativos agem de forma diferente, estes podem gerar alterações nas informações como podem também criar novas informações na troca de mensagens, sendo esse dividido em quatro subcategorias (SCHNEIER,2001):

- Repetição: quando uma mensagem, ou parte dela, é capturada durante a transmissão entre duas entidades e após isso é retransmitida,causando um efeito não-autorizado;
- Personificação: um indivíduo dentro de um sistema computacional com privilégio menor, se passa por outra com privilégios maiores para garantir benefícios;
- Modificação de Mensagens: uma mensagem é capturada e seu conteúdo é alterado ou reordenado, e após é enviado para seu destino;
- Negação de Serviço**:** o objetivo deste ataque é causar queda brusca no desempenho do alvo, geralmente um atacante dispara várias mensagens para o destino e este ataque tem sempre um alvo específico.

A diferença fundamental entre os tipos dos ataques é a sua natureza, já que o ataque passivo é mais difícil de detectar por não alterar o conteúdo das informações e, por isso, não deixa rastros. Esses ataques podem ser prevenidos se a organização aliar conceitos descritos anteriormente com ferramentas e técnicas como, por exemplo, a criptografia.

2.2 - Criptografia

Segundo Dicionário Aurélio, “*criptografia é a arte de escrever em cifra ou em código*”. O objetivo da criptografia é “embaralhar” um texto através de fórmulas matemáticas de modo que o torne incompreensível para pessoas que não têm autorização para ler seu conteúdo.

A criptografia é uma arte muito antiga, segundo relatos o primeiro uso documentado da criptografia ocorreu por volta de 1900 a.C. no Egito, um escriba fez uso dos hieróglifos fora do padrão convencional para escrever seu texto. Existem também outros exemplos: na Mesopotâmia foi encontrado um texto de 1500 a.C. , a cifra hebraica ATBASH de 500-600 a.C. e a cifra de Júlio César de 50-60 a.C. que é uma substituição simples (SCHNEIER, 2001).

A idéia principal da criptografia é permitir a um grupo de pessoas que troquem informações de modo que seu conteúdo seja secreto. De uma maneira abrangente, a criptografia é dividida em duas categorias: criptografia simétrica e criptografia assimétrica, as duas serão descritas a seguir.

2.2.1 - Criptografia Simétrica

A criptografia simétrica, também conhecida como criptografia convencional, é de fácil entendimento. Ela consiste basicamente de uma única chave secreta (K) entre as duas entidades que desejam trocar informações. Esta chave única serve tanto para cifrar quanto para decifrar o texto. Para isso, essa chave deve ser de conhecimento de ambas as partes envolvidas (GHISLERI, 2002).

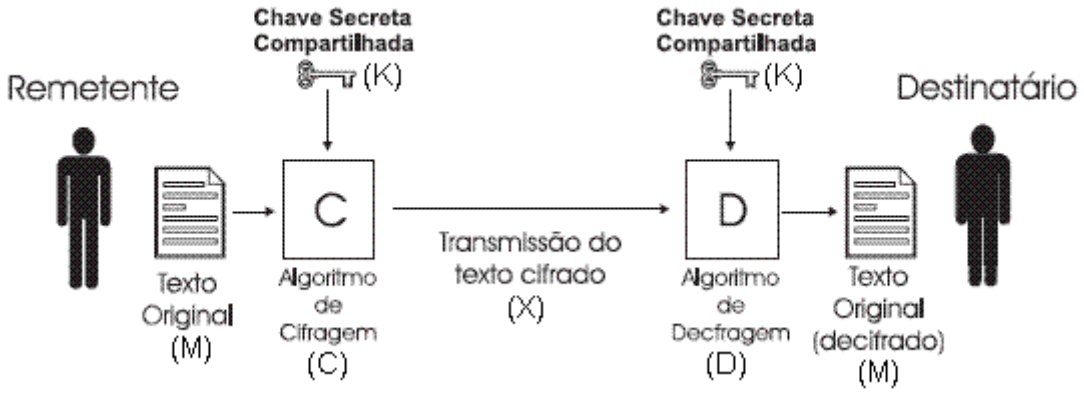

Figura 2 - Esquema de Transmissão de Mensagem Utilizando Criptografia Simétrica. Fonte (GHISLERI, 2003).

Conforme a Figura 2 os passos para o processo de cifrar e emitir a mensagem são (COBB, 2004):

1) O Remetente, de posse do texto original (M), que deseja enviar para o Destinatário, escolhe um algoritmo de criptografia (C) junto da chave (K) definida e compartilhada com o destinatário para cifrar a mensagem. Assim, o texto cifrado (X) é obtido através da fórmula:

$X = C_K(M)$.

2) A mensagem cifrada (X) é enviada para o Destinatário pela rede.

3) O Destinatário recebe a mensagem cifrada e utiliza o algoritmo (D) e chave usados (K) pelo Remetente para decifrar o texto e assim o texto original (M) é recuperado:

$M = D_K(X)$.

Dentre alguns dos algoritmos mais conhecidos da criptografia simétrica estão:

- **DES:** (*Data Encryption Standard*), junto de suas variantes o *Simple* DES, *Double* DES e *Triple* DES é um dos algoritmos mais difundidos no mundo. Criado pela IBM em 1977, por ter o tamanho da chave pequeno (apenas 56 *bits*), foi quebrado por ataques de “força bruta” em 1997.
- **IDEA:** (*International Data Encryption Algorithm*) foi criado em 1991 por Xuejia Lai e James Massey, patenteado pela empresa suíça ASCOM Systec. Possui as mesmas características gerais do DES, porém nos processadores mais modernos ele ganha em desempenho do DES.
- **BLOWFISH:** Algoritmo criado por Bruce Schneier, permite que o usuário escolha entre maior desempenho ou maior segurança com tamanhos de chaves que variam entre 32 a 448 *bits*.
- **RC5:** Desenvolvido por Ron Rivest, possui tamanho de chave variável e em seu processo de crifagem o usuário pode escolhe o número de rodadas de acordo com o tamanho do texto a ser cifrado.
- **AES:** (*Advanced Encryption Standard*) Foi desenvolvido para substituir o DES no concurso promovido pelo NIST (*National Institute of Standards and Technology*), criado Vicent Rijmen e Joan Daemen. Baseado no cifrador Square. É extremamente resistente à ataques conhecidos de criptoanálise.

O emprego da criptografia simétrica, principalmente em protocolos criptográficos para transações seguras realizadas via Web, é muito bem aceito pois seu desempenho é melhor que o desempenho de protocolos que fazem uso da criptografia assimétrica, que será explicada a seguir. Porém, a maior desvantagem deste tipo de criptografia está na distribuição das chaves, já que é necessário pré-estabelecer um canal de comunicação seguro para a combinação desta chave única entre as partes. Em alguns casos, as pessoas envolvidas encontram-se pessoalmente, combinam por telefone, usam

correspondência em correio comum ou qualquer outro meio que considerem seguro para sua necessidade.

2.2.2 - Criptografia Assimétrica

A criptografia assimétrica também é conhecida como criptografia de chave pública. De acordo com Stallings (2005) a melhor e talvez a única evolução real na história da criptografia. Com ela, foram substituídos os algoritmos que eram possíveis ser calculados à mão. Isso se deu também graças ao avanço tecnológico dos computadores que possibilitou a criação de algoritmos/técnicas mais avançadas com cálculos matemáticos também mais elaborados.

Cada uma das partes que participa da troca de informações possui um par de chaves: uma pública e uma privada. A primeira é distribuída para todos através de um repositório e/ou pode ser enviada junto com a mensagem cifrada e poder ser usada pelo destinatário para decifrá-la. A segunda como o próprio nome diz, privada, só seu dono conhece e esta é usada para cifrar as mensagens que serão enviadas. A mensagem que é cifrada com a chave privada só pode ser decifrada pelo seu par, a chave pública e vice-versa. Este tipo de criptografia, entre outras possibilidades, pode garantir a autenticidade de uma mensagem. Isto ocorre quando a mensagem é cifrada através da chave privada de uma entidade e, portanto, só poderá ser decifrada pelo seu par, a chave pública. Como a chave privada está de posse exclusivamnente de seu proprietário, ele é o único que pode usá-la, o que assegura a autenticidade (ARDIGO, 2004).

A Figura 3 demonstra a troca de mensagens que utiliza criptografia assimétrica. Os passos desta transação são:

1) O Remetente com o texto da mensagem (M) em mãos utiliza um algoritmo de criptografia assimétrica (C) com sua chave privada (KR) para cifrar o texto da mensagem, sendo X o texto cifrado da mensagem.

$X = C_{KR}(M)$.

2) A mensagem cifrada (X) é enviada para o Destinatário. Este, de posse da chave pública (KU) do Remetente, decifra a mensagem através do mesmo algoritmo utilizado pelo remetente (D). Obtendo assim o texto original (M).

$M = D_{KU}(X)$.

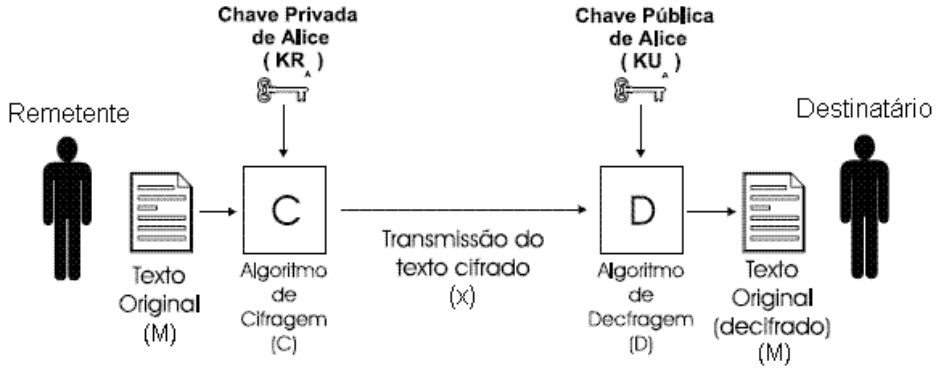

Figura 3 - Funcionamento da Criptografia Assimétrica. Fonte (GHISLERI, 2003)

Esta técnica garante ao destinatário a autenticidade da mensagem enviada a ele, pois uma mensagem que é cifrada com a chave privada de um indivíduo só pode ser decifrada pelo seu par. Porém não garante a confidenciabilidade das mensagens trocadas, já que é fácil chegar ao texto original da mensagem. Basta que a entidade que interceptou conheça a chave pública do Remetente (FLEURY, 2002). Para resolver esse problema basta o Remetente fazer uma dupla encriptação da mensagem, como ilustrado na figura 4 onde os passos são (MOREIRA, 2002):

1) O Remetente utiliza a sua chave privada (KRr) para cifrar a mensagem original (M) que deseja enviar para o Destinatário. Sendo X a mensagem cifrada e C o algoritmo.

$X = C_{KRr}(M)$.

2) Feito isso, a mensagem resultante do processo anterior (X) é novamente cifrada, mas agora utilizando a chave pública do Destinatário (KUd). Isso garante que somente ele possa decifrar a mensagem. Y é a mensagem cifrada chave do Destinatário.

$Y = C_{KUd}(X)$.

3) A mensagem cifrada duas vezes é enviada normalmente para o Destinatário.

4) Após o recebimento da mensagem, o Destinatário decifra (D) a mensagem com o mesmo algoritmo com sua chave privada (KRd). Sendo X a mensagem resultante.

$X = D_{KRd}(Y)$.

5) Agora o destinatário decifra (D) a mensagem com a chave pública do Remetente (KUr) e chega ao texto original (M).

$M = D_{KUr}(X)$.

Para uma melhor compreensão esses passos são demonstrados na Figura 2.4.

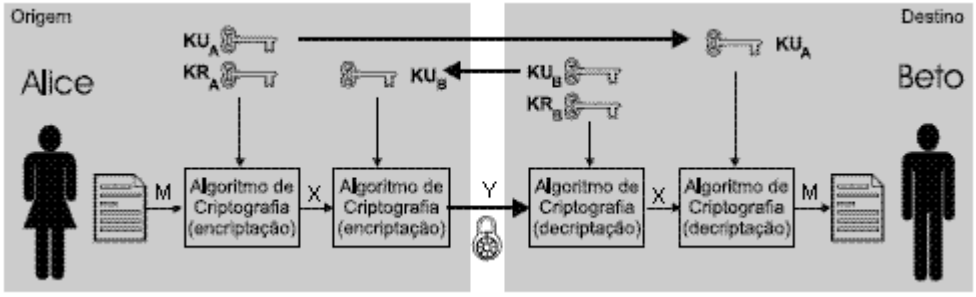

Figura 4 - Cifrando a Mensagem com a Chave Privada do Remetente e com a Chave Pública do Destinatário. Garantia de Confidenciabilidade e Autenticidade. Fonte (GHISLERI, 2002)

Alguns algoritmos mais conhecidos da criptografia assimétrica são os seguintes:

- **RSA:** É denominado dessa forma por causa dos nomes dos seus idealizadores Ron **R**ivest, Adi **S**hamir e Len **A**dleman. É o mais utilizado nessa categoria. Utiliza o esquema de fatoração de números primos. Uma chave do RSA de 512 *bits* foi quebrada em 1999 pelo instituto de pesquisa nacional da Holanda.
- **ElGamal:** Este algoritmo é utilizado para o gerenciamento de chaves públicas. A sua segurança provém da grande dificuldade de calcular logaritmos discretos.
- **Diffie-Hellman:** O propósito deste algoritmo é o compartilhamento de chaves através de canais não-seguros.
- **Curvas Elípticas:** Um dos mais recentes algoritmos criados. Usa o esquema de curvas elípticas sobre corpos finitos.

Obviamente existem outros algoritmos e estes aqui citados são temas de estudos profundos, esta foi apenas uma pequena explanação sobre eles para o conhecimento dos mesmos. Os algoritmos de criptografia assimétrica são utilizados, dentre outros fins, para assinaturas digitais e no protocolo SSL. Isso para garantir a autenticidade das chaves utilizadas para cifrar as informações trocadas entre duas entidades.

2.2.3 – Protocolo SSL (*Secure Sockets Layer*)

É um protocolo de comunicação que visa criar um caminho seguro para troca de informações entre aplicações na Internet independente de plataforma, de forma transparente e sobretudo de forma segura (ARAÚJO, 2006).

Este protocolo foi desenvolvido pela empresa *Netscape Communications* e em julho de 1994 foi lançada sua primeira versão. Em abril do ano seguinte foi lançada a referência para a implementação da segunda versão do protocolo que foi distribuída com os navegadores Internet Explorer e Netscape e com os servidores web mais comuns no mercado, por exemplo, IIS, Apache, httpd, etc., esta versão do SSL foi submetida ao IETF denominado TLS (*Transport Layer Security*) (SCHNEIER, 2001).

A terceira versão do SSL foi lançada em novembro de 1995 As melhorias mais significativas foram a diminuição no número de rodadas de negociação. O servidor passou a definir o algoritmo para cifrar as informações e compressão, separação das chaves usadas para cifrar os dados e para autenticação, entre outras.

O protocolo SSL visa garantir algumas premissas de segurança como confidenciabilidade dos dados trocados numa sessão de comunicação. Para que isso aconteça o protocolo faz uso de algoritmos de criptografia simétrica, como por exemplo, DES ou RC4. Com SSL é possível realizar autenticação de servidores ou cliente utilizando algoritmo de criptografia assimétrica, por exemplo o RSA (GOOTS, et al. 2003).

O SSL trabalha entre as camadas de transporte e aplicação. Este pode funcionar sobre os protocolos: HTTP, FTP, SMTP, Telnet, entre outros, de

forma transparente. O SSL é um conjunto de três protocolos, dois deles situados na camada de aplicação e o terceiro entre o protocolo da aplicação e protocolo de transporte. Os protocolos que fazem parte desse conjunto denominam-se:

- *SSL HandShake Protocol:* Sua função é estabelecer um identificador para a sessão criada, um método de compressão e um conjunto de algoritmos criptográficos que serão adotados nessa sessão.
- *SSL Alert Protocol:* Projetado para enviar e receber mensagens de erro e interromper a sessão de comunicação, caso seja necessário.
- *SSL Record Protocol:* Faz o encapsulamento das camadas de nível mais alto, fornece os serviços de compressão, fragmentação, autenticação de mensagens e etc.

Com isso pode-se notar que o SSL foi projetado para garantir a confiabilidade e integridade das mensagens trocadas numa sessão de comunicação através da criptografia desses dados, autenticação de servidores e clientes através de certificados e assinaturas digitais.

2.2.3 – Assinatura Digital

A assinatura digital foi criada para o mesmo propósito de uma assinatura manual em um documento em papel, ou seja, ela visa garantir a autenticidade do autor de um documento digital, mas considerando os meios eletrônicos para transmissão documentos digitais assinados (BRUNORI, 1999).

Alguns sistemas de assinaturas digitais utilizam esquemas de criptografia simétrica para assinar os documentos, com funções chamadas *one-way hash function*, conhecida também como *fingerprint*, *cryptographic checksum*, entre outros. Essa função com esquema de criptografia simétrica gera uma cadeia de caracteres sobre um documento. Se o valor for o mesmo tanto no Remetente quanto no Destinatário, significa que essa informação não foi alterada (ARDIGO, 2004).

Porém, este tipo de assinatura não é predominante por não garantir a integridade total do documento, onde seu conteúdo pode ter sido alterado e um novo valor da função pode ser calculado.

Para sanar esse problema, são utilizados algoritmos de criptografia assimétrica com a função calculando o valor das chaves num sentido inverso, ou seja, primeiramente é calculado o *hash* da mensagem e após é cifrado com a chave privada remetente. Assim é enviada a mensagem junto com o *hash* cifrado da mensagem para o destinatário. O destinatário ao receber a mensagem para verificá-la, segue os seguintes passos: calcula o *hash* da mensagem, decifra a mensagem com a chave pública do remetente e compara. Se ambos forem iguais, significa que a mensagem assinada está integra e autêntica (SCHNEIER, 2001), (STALLINGS, 2005). Para visualizar melhor a Figura 5 ilustra esse processo.

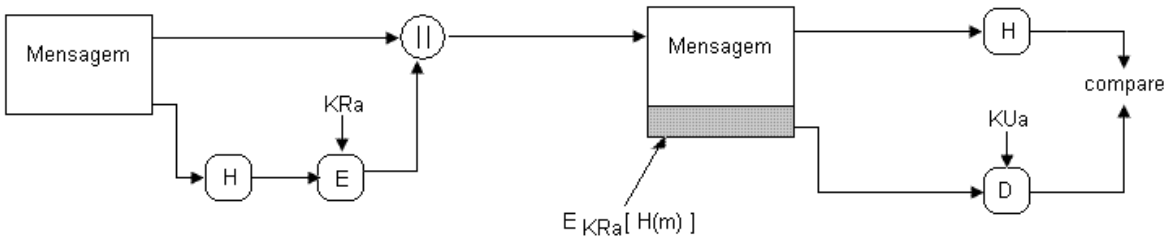

Figura 5 - Passos da Assinatura Digital. Fonte (STALLINGS,1999)

2.2.4 – Considerações

Os conceitos criptografia, SSL, certificado digital e assinatura digital vistos neste capítulo serão utilizados na implementação do protocolo proposto por Amauri Sant´Anna Ghisleri para um sistema seguro de atendimento ao cliente. Esses conceitos são utilizados na prática em vários passos de seu funcionamento. O capitulo 3 deste trabalho descreve o protocolo.

3. PROTOCOLO PROPOSTO POR AMAURI SANT'ANNA GHISLERI

O sistema proposto por Amauri Sant'Anna Ghisleri em sua dissertação de mestrado (GHISLERI,2002) visa a garantia da qualidade do serviço no atendimento ao cliente, onde os principais participantes são: a empresa que oferece serviços e o cliente, que faz requisições de serviços para esta empresa. Para isto, todas as requisições feitas pelo cliente deverão ser tratadas igualmente pela empresa garantindo que seus direitos (como consumidor) sejam assegurados. Caso isto não venha a acontecer, uma autoridade certificadora com credibilidade sobre as partes pode intervir. Esta garantia ocorre através da intervenção de vários participantes do sistema que possuem credibilidade pública tais como: autoridade fiscalizadora, autoridade de aviso e autoridade de datação. Estes participantes serão descritos juntamente com o protocolo, no decorrer deste capítulo.

O uso da criptografia e o acordo entre partes envolvidas em uma comunicação que troque informações originam os protocolos criptográficos, ou seja, uma maneira específica que utilizada criptografia em seus passos para a troca de informações. Um protocolo criptográfico é uma série de passos que envolve duas ou mais partes para realizarem suas tarefas.

Esse tipo de protocolo é utilizado para a comunicação segura em redes públicas objetivando garantir a autenticidade, confidenciabilidade e integridade dos dados trocados. O protocolo proposto deseja proporcionar para ambas as partes a legitimidade das informações trocadas na comunicação entre as partes.

Neste capítulo será apresentado o protocolo, com suas fases, seus detalhes, enfim, seus elementos e também as hipóteses consideradas para a aplicação.

3.1 Fases do Protocolo

O protocolo é composto por quatro fases: Fase de Requisição, Fase de Atendimento, Fase de Fechamento da Requisição e Litígio. Estas fases são descritas na ordem em que ocorrem durante sua execução, para que haja um melhor entendimento do mesmo e para facilitar o tratamento de questões relacionadas a elas (GHISLERI, 2002).

3.1.1 Fase de Requisição

A requisição feita pelo cliente é o ponto inicial do protocolo. Para efetuar essa requisição o cliente acessa o formulário hospedado no sítio da empresa, preenche com seus dados fazendo a requisição de um atendimento, que pode ser tanto para contratação de um serviço ou para uma eventual reclamação. Nesta etapa o cliente precisa ser identificado com sua assinatura digital para comprovar para o protocolo a fonte e a autoria da requisição. Com isso, é possível atender as premissas de segurança requisitadas pelo protocolo (GHISLERI, 2002).

Feito isto, o cliente deve receber um comprovante que lhe assegure que a sua requisição foi aceita e assim começa a contagem do prazo para o seu atendimento. Esse comprovante emitido pela empresa deve conter a requisição do cliente e o protocolo feito pela autoridade de datação[1] (GHISLERI, 2002).

A autoridade fiscalizadora definida no protocolo, a qual pode ser um setor da empresa ou algum órgão do governo ou qualquer outro órgão que visa a qualidade do atendimento ao cliente como Procon, Frebraban, etc., também deve receber uma cópia desse comprovante para acompanhar o processo e fazer uma possível intervenção. Ao término dessa fase o tempo para o atendimento passa a contar, iniciando a fase posterior (GHISLERI, 2002).

[1] Autoridade de Datação: protocola a requisição do cliente, dando a essa requisição uma referência temporal.

Nessa fase os pedidos de atendimentos são gerados pelo cliente e dão início ao funcionamento do protocolo. Visto que o protocolo interage com entidades externas trocando informações para gerar os comprovantes devidamente carimbados visando à segurança do cliente garantindo o cumprimento dos prazos e garantindo a identidade da empresa e atestando que o cliente é quem realmente diz ser.

3.1.2 Fase de Atendimento

Nesta fase, é realizado o atendimento propriamente dito, ou seja, a requisição que o cliente fez passa a ser tratada pela empresa com a abertura de uma ordem de serviço. Quando concluído o atendimento, uma notificação de conclusão da ordem de serviço é enviada para o cliente e para a autoridade fiscalizadora. Para comprovar se o prazo de atendimento ao cliente foi respeitado, esta notificação é protocolada pela autoridade de datação. A autoridade fiscalizadora guarda os registros desses eventos em seu banco de dados para possível uso da própria empresa em caso de litígio com o cliente (GHISLERI, 2002).

A fase de atendimento é uma atividade que ocorre extra-protocolo, ou seja, a empresa, de posse da requisição gerada pelo cliente, inicia os procedimentos cabíveis para sanar a solicitação de atendimento durante o prazo estipulado pelo protocolo. Depois do atendimento ter sido realizado, a empresa comunica o cliente avisando que seu requerimento foi atendido pela empresa.

3.1.3 Fase de Fechamento da Requisição

O encerramento de uma transação entre a empresa e o cliente é denominado fechamento da requisição. Esta fase pode ocorrer de 2 formas diferentes (GHISLERI, 2002):

- **Fechamento natural pelo cliente:** Quando o cliente recebe uma notificação de que a sua requisição foi atendida pela empresa, ele deve verificar se isso ocorreu e deve encerrá-la junto à empresa, enviando essa notificação de volta para a empresa assinada digitalmente.
- **Fechamento forçado por decurso de prazo:** Se o cliente não se manifestar a respeito da notificação, este é acionado a partir de uma autoridade de aviso. A função da autoridade de aviso é terceirizar a notificação de conclusão da ordem de serviço pela empresa. Esta terceirização é de grande interesse para as empresas que adotam o protocolo, pois essas autoridades possuem meios de comunicação mais eficientes para contatar o cliente.

O fechamento da requisição ocorre após o cliente ser notificado sobre a conclusão do atendimento pela empresa. Esse fechamento pode ser efetuado pelo cliente caso ele entenda que sua requisição foi totalmente atendida de modo satisfatório ou pela empresa caso o cliente não se manifeste depois de um período estipulado após o envio da notificação, nesse processo o protocolo faz uso dos serviços da autoridade de aviso que dispões de mecanismos para contatar o cliente.

3.1.4 Litígio

Caso ocorra algum litígio entre a empresa e o cliente, a autoridade fiscalizadora pode assumir o papel de mediadora dos interesses das partes envolvidas, pois, durante as transações, essa autoridade recebe e documenta diversas informações trocadas nessas transações. Logo, os recibos de contratação de serviço ou reclamação, notificações de encerramento de ordem de serviço, são considerados documentos oficiais do protocolo proposto e

podem ser usados como prova para quaisquer ações que venham a ser tomadas adiante (GHISLERI, 2002).

O litígio ocorre quando o client,e não feliz com o resultado do atendimento de sua requisição, utiliza-se de mecanismos jurídicos para contestá-lo perante a empresa. Quando a situação litigiosa ocorre, a autoridade fiscalizadora faz um papel importante, pois ela possui todos os comprovantes gerados durante aquela requisição e com isso pode atuar como mediadora ou como testemunha para uma das partes.

3.2 O Protocolo proposto

Após descritas as fases que compõem o protocolo, pode-se dar continuidade ao estudo. O funcionamento do protocolo é descrito na Figura 6. Deve-se observar os passos numerados nesta figura, (GHISLERI, 2002):

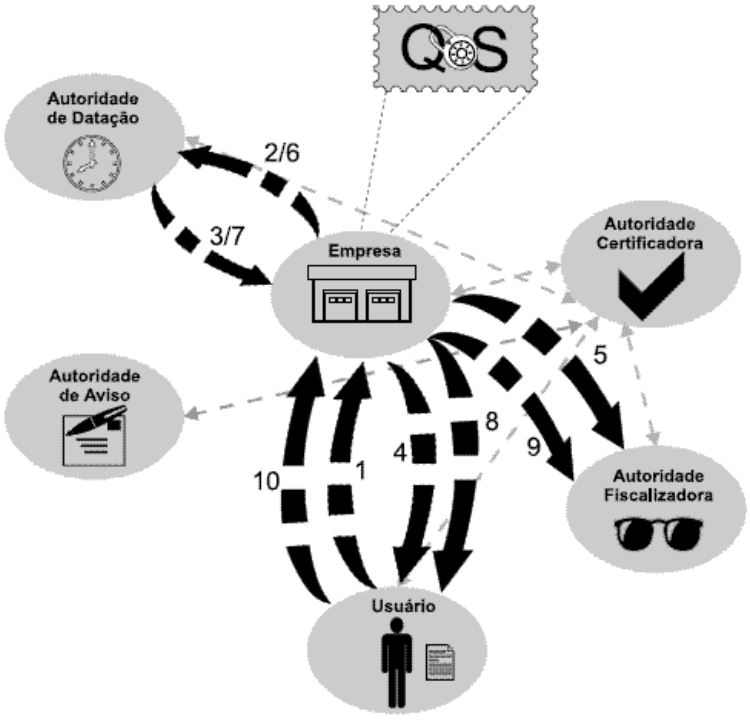

Figura 1 - Protocolo Proposto. Fonte (GHISLERI,2002).

1. O cliente é ponto inicial do protocolo. Este visita o sítio da empresa para contratar um serviço ou solicitar um atendimento,

gerando uma requisição que é assinada digitalmente pelo cliente para garantir sua identidade;

2. A empresa recebe essa requisição, gera um resumo dessa mensagem com um algoritmo de *hash*, assina digitalmente e envia essa mensagem para a autoridade de datação;
3. A autoridade de datação recebe o resumo da mensagem enviada pela empresa, protocola concatenando data e hora na mensagem, assina digitalmente a mensagem protocolada e envia de volta para empresa;
4. A empresa recebe a requisição protocolada pela autoridade de datação e gera o recibo da requisição para o cliente, oferecendo a ele a garantia do atendimento. Esse recibo é composto pela requisição do cliente acrescido do protocolo fornecido pela autoridade de datação;
5. Uma cópia do recibo é enviado para a autoridade fiscalizadora; que armazena esse recibo em seu banco de dados para o caso de existir litígio entre as partes ou mesmo para auditorias internas. Encerrando assim a fase de requisição do protocolo;
6. A empresa posteriormente ao atendimento e conclusão da ordem de serviço gerada pela requisição do cliente, efetua o fechamento dessa ordem de serviço, gera um resumo (*hash*) e envia para a autoridade de datação;
7. A autoridade de datação em posse do resumo da ordem de serviço, protocola e devolve para a empresa. O fato de protocolar o fechamento da ordem de serviço fornece para a empresa a comprovação do atendimento ao cliente no prazo determinado;
8. A empresa recebe o fechamento da ordem de serviço protocolado e gera um comprovante de fechamento, envia esse comprovante junto com uma mensagem para o cliente. Essa mensagem solicita para o cliente efetuar o encerramento da transação junto à empresa;
9. A empresa envia uma cópia do fechamento da ordem de serviço para a autoridade fiscalizadora, isso permite que seja

feita uma comparação com o recibo da requisição para se verificar o prazo no atendimento;

10. O cliente após verificar o atendimento recebido, encerra a transação com sua assinatura digital na mensagem de fechamento enviada pela empresa. Com isso, é gerado o recibo de fechamento que é a garantia da empresa perante o protocolo.

Essa forma descrita é o comportamento ideal de uma transação de atendimento do protocolo proposto, considerando que nenhuma das partes envolvidas esteja num ambiente que não possui falhas dos canais de comunicação com a Internet, porém isso na prática pode acontecer. Em caso de falhas na comunicação a autoridade de aviso tem os recursos necessários para contatar tanto a empresa quanto o cliente, desta forma as partes podem suspender o atendimento até que a comunicação direta torne-se disponível.

Se após um determinado período que o cliente fez a requisição, este não receber o recibo de sua requisição ele pode solicitar à autoridade de aviso que re-encaminhe sua requisição para a empresa. Caso a autoridade de aviso não obtenha sucesso, poderá entrar em contato com a autoridade fiscalizadora para que essa possa imediatamente intervir junto à empresa.

Porém, se a empresa responder ao comunicado transmitido pela autoridade de aviso, ela deverá providenciar o recibo de requisição para o cliente, como deveria ser feito normalmente como mostra a Figura 7 (GHISLERI, 2002).

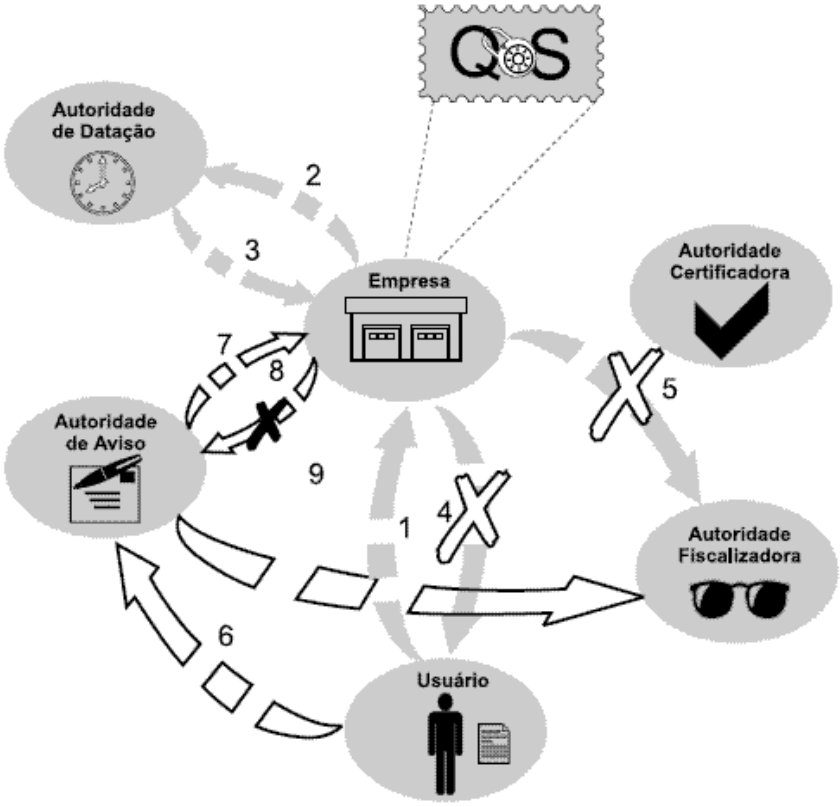

Figura 2 - Comportamento do Protocolo quando a empresa não responde. Fonte (GHISLERI,2002).

Existe a possibilidade de o cliente não confirmar o fechamento do pedido até um determinado prazo que é contado a partir da data de fechamento da ordem de serviço. Quando isso ocorre, a empresa entra em contato com a autoridade de aviso que deve entrar em contato com o cliente para notificá-lo que sua requisição foi atendida. Se nenhuma das tentativas de contato for bem sucedida a empresa pode fechar a requisição por decurso de prazo. Esses passos são descritos na Figura 8 (GHISLERI, 2002).

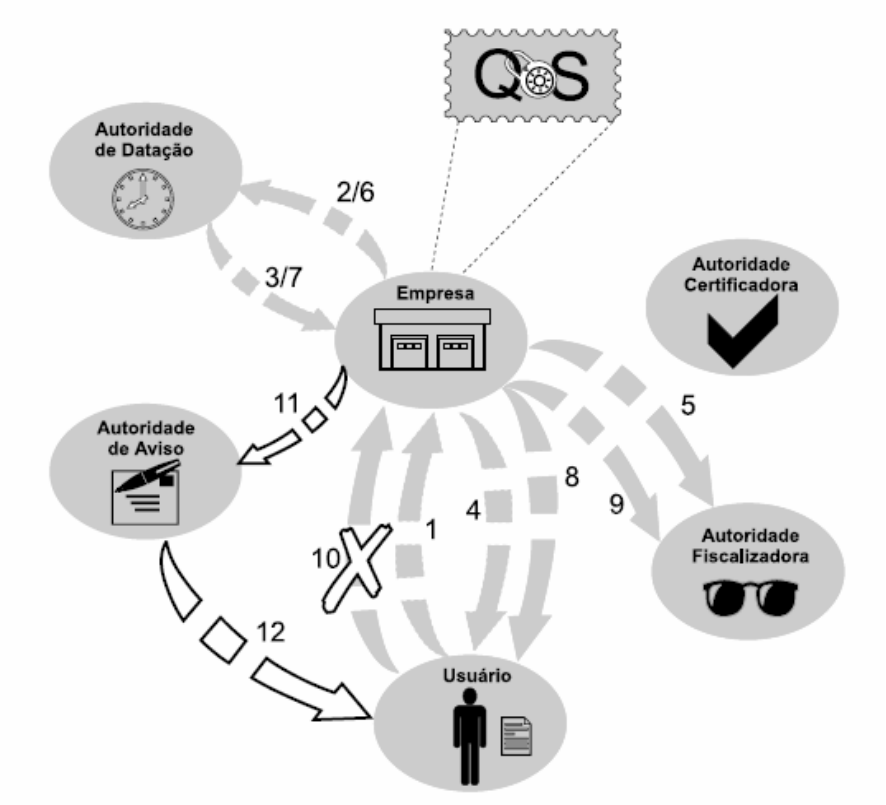

Figura 3 - Comportamento do protocolo quando o cliente não responde a solicitação de fechamento da ordem de serviço. Fonte (GHISLERI,2002)

3.2.1 Considerações

Feito o estudo do protocolo fica claro que é possível sua implementação, pois o mesmo, além de utilizar de forma adequada os dispositivos de segurança computacional apresentados, também foi formalizado e validado segundo (GHISLERI, 2002).

O protocolo proposto usa mecanismos de segurança para que a privacidade dos dados trocados durante as transações realizadas seja mantida. Além disso, através de entidades externas, como por exemplo, autoridade fiscalizadora, de datação e de aviso, o protocolo assegura que os direitos do consumidor e da empresa sejam respeitados seja através do cumprimento dos prazos estabelecidos ou através de comprovantes eletrônicos confiáveis devidamente documentados.

Além da situação ideal de comportamento que o protocolo possui, existem as situações anormais que podem ocorrer. Porém, ele esta provido de recursos para que, nessas situações, a parte que comete a falha seja notificada

da negligência e a outra parte tenha satisfações a respeito do fato ocorrido. Visto que os comportamentos não-ideais do protocolo não fazem parte do escopo do trabalho, somente as situações de comportamento ideal serão implementadas.

4 . PROPOSTA PARA IMPLEMENTAÇÃO DE UMA APLICAÇÃO QUE CONTENHA AS CARACTERÍSTICAS DO PROTOCOLO

Este capítulo apresenta o projeto para a implementação da aplicação, mostrando a metodologia utilizada, os diagramas de casos de uso, a modelagem do banco de dados, os diagramas de estados de navegação e as tecnologias selecionadas para a implementação. Além disso, serão mostradas as propostas para as telas do sistema e a discussão sobre a implementação com os detalhes de como foi construída.

4.1 Metodologia Usada no Projeto para Implementação da Aplicação

Para desenvolver o projeto da aplicação foi utilizada a metodologia Processo Unificado (UP), também conhecido como *Rational Unified Process* (RUP). Consiste de um processo ágil com poucos, porém concisos, artefatos e pouca burocracia. O objetivo é diminuir o tempo de desenvolvimento, pois o cliente está interessado em seu software pronto, não numa documentação enorme que justifica o motivo do software não estar pronto (WAZLAWICK, 2004).

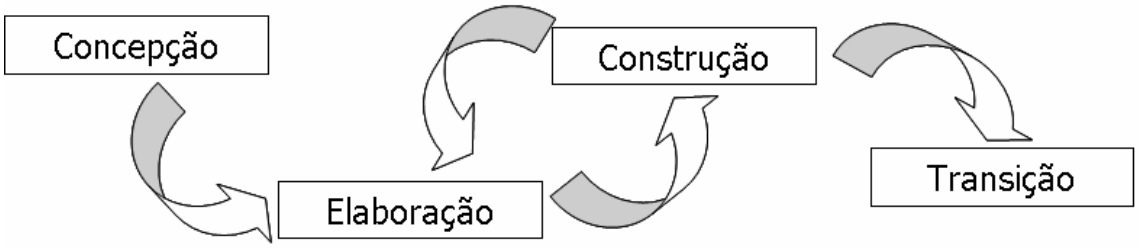

Figura 1 - Fases do processo unificado

O UP é composto de quatro fases (WAZLAWICK, 2004):

- Fase de Concepção: é a primeira fase no desenvolvimento do software. Levantam-se os principais requisitos e é realizado o estudo de viabilidade.

- Fase de Elaboração: Esta fase atua no ciclo de iteração do processo. É constituída do projeto e da análise do software.
- Fase de Construção: Também atua nos ciclos de iteração do processo de desenvolvimento. Corresponde à implementação e aos testes do software.
- Fase de Transição: Ocorre depois dos ciclos iterativos, quando o software está pronto e será implantado no cliente.

A metodologia UP propõe que cada etapa do processo tenha um objetivo claro e preciso, vislumbrando a produção de código que atenda aos requisitos levantados no menor tempo possível, utilizando poucos artefatos e pouca burocracia, auxiliando na redução do tempo de produção do software.

4.1.1 Análise de Requisitos

Uma das etapas mais importantes da fase de análise é a análise de requisitos, por tratar de descobrir o que o cliente quer com o sistema. Um dos artefatos produzidos nesta fase é o “Sumário Executivo” que corresponde a um sumário executivo de um plano de negócios, ou seja, mostra uma visão geral e resumida sobre o projeto (WAZLAWICK, 2004).

Sumário Executivo:

É proposta a implementação de um sistema seguro de atendimento ao cliente fundamentado na proposta de um protocolo criptográfico. Neste sistema, o cliente poderá, através do sítio da empresa na *Web*, fazer solicitações de serviços, sanar dúvidas e receber informações diversas. O cliente terá sido previamente cadastrado pela empresa no momento em que contratou um produto ou serviço. O sistema irá assegurar ao cliente que sua solicitação seja atendida pela empresa no prazo e garantirá à empresa que o serviço prestado terá confirmação de recebimento.

A análise de requisitos é o processo para descobrir quais serão as operações que o sistema realizará e quais as restrições sobre essas

operações. As operações constituem as funcionalidades do sistema e por esse motivo são chamadas de requisitos funcionais. Já as restrições sobre as funcionalidades são denominadas requisitos não-funcionais (WAZLAWICK, 2004). Os requisitos funcionais obtidos a partir da análise feita sobre o protocolo estudado são:

- Abrir OS
- Fechar OS
- Confirmar fechamento da OS pelo cliente
- Confirmar fechamento da OS pela empresa
- Verificar a situação da OS

A seguir serão mostradas as tabelas de requisitos funcionais resultantes dessa análise.

Tabela 1 – Requisito Funcional Abrir OS

F1 – Abrir OS (Ordem de Serviço)				
Descrição: O cliente visita o sítio da empresa, faz sua validação perante o sistema, preenche o formulário para abertura da OS (Ordem de Serviço) e esta fica registrada no sistema.				
Requisitos não-funcionais				
Nome	**Descrição**	**Categoria**	**Desejável**	**Permanente**
Autenticação do Cliente	O cliente entra com seus dados (nome de usuário e sua senha acesso) para que o sistema faça a autenticação do mesmo.	Segurança	( )	(X)
Especificação da OS	O cliente deve preencher os dados relativos a OS: (tipo, data de abertura, título e suas observações.	Interface	( )	(X)
Identificação do tipo da OS	O cliente deve selecionar o tipo da OS: reclamação, esclarecimento de dúvidas ou outros.	Interface	( )	(X)
Identificação temporal da abertura	A empresa deve “solicitar” para autoridade de datação um carimbo (data/hora) que comprove o atendimento no prazo.	Segurança	( )	(X)
Geração do Número da OS	O sistema deverá gerar um número exclusivo para a Ordem de Serviço (OS)	Segurança	( )	(X)
Geração e envio do comprovante de abertura	A empresa envia para o cliente um comprovante indicando que a sua OS foi devidamente registrada.	Segurança	( )	(X)

Tabela 2 – Requisito Funcional Fechar OS

F2 – Fechar OS (Ordem de Serviço)				
Descrição: A empresa, após realizar os procedimentos de atendimento, envia para o cliente um comprovante do fechamento da OS.				
Requisitos não-funcionais				
Nome	**Descrição**	**Categoria**	**Desejável**	**Permanente**
Identificação da OS	A empresa deve buscar em seus bancos de dados a OS que será fechada	Especifica-ção	( )	(X)
Descrição do procedimento realizado	A empresa deve descrever quais medidas foram tomadas para sanar a OS do cliente.	Interface	( )	(X)
Identificação temporal do fechamento	A empresa deve "solicitar" para autoridade de datação um carimbo (data/hora) que comprove o momento em que atendimento foi concluído.	Segurança	( )	(X)
Geração e envio do comprovante de atendimento	A empresa envia para o cliente um comprovante indicando que a sua OS já foi devidamente atendida	Segurança	( )	(X)

Tabela 3 – Requisito Funcional Confirmar Fechamento da OS pelo Cliente

F3 – Confirmar Fechamento da OS (Ordem de Serviço) pelo Cliente				
Descrição: O cliente, após receber o comprovante de fechamento da OS, confirma o fechamento quando verifica a resolução do problema e considera-a satisfatória.				
Requisitos não-funcionais				
Nome	**Descrição**	**Categoria**	**Desejável**	**Permanente**
Autenticação do Cliente	O cliente entra com seus dados para que o sistema faça a autenticação do mesmo.	Segurança	( )	(X)
Identificação da OS	O cliente entra com os dados para identificação da OS	Interface	( )	(X)
Identificação temporal do fechamento	Ao confirmar o fechamento da OS a autoridade de datação deve gerar o carimbo comprovando o atendimento.	Segurança	( )	(X)

Tabela 4 – Requisito Funcional Confirmar Fechamento da OS pela Empresa

F3 – Confirmar Fechamento da OS (Ordem de Serviço) pela Empresa				
Descrição: A empresa pode confirmar o fechamento da OS caso o cliente não se manifeste por um período após o fechamento da OS (Decurso de Prazo).				
Requisitos não-funcionais				
Nome	**Descrição**	**Categoria**	**Desejável**	**Permanente**
Autenticação do Usuário	O cliente entra com seus dados para que o sistema faça a autenticação do mesmo.	Segurança	( )	(X)
Identificação da OS	O cliente entra com os dados para identificação da OS.	Interface	( )	(X)
Identificação temporal do fechamento	Ao confirmar o fechamento da OS a autoridade de datação deve gerar o carimbo comprovando o atendimento.	Segurança	( )	(X)

Tabela 5 – Requisito Funcional Verificar situação da OS

F4 – Verificar situação da OS				
Descrição:. Cliente, Empresa e Autoridade fiscalizadora podem verificar a situação, o andamento da OS e se os prazos estão sendo respeitados.				
Requisitos não-funcionais				
Nome	**Descrição**	**Categoria**	**Desejável**	**Permanente**
Autenticação do Cliente	O usuário* entra com seus dados para que o sistema faça a autenticação do mesmo.	Segurança	( )	(X)
Identificação da OS	O usuário* entra com os dados para identificação da OS	Interface	( )	(X)
Exibição de informações da OS	O sistema deve exibir os dados da OS indicando a situação em que se encontra e suas ocorrências.	Interface	( )	(X)

* O termo usuário aqui utilizado representa qualquer um dos três possíveis usuários do sistema: Cliente, Empresa e Autoridade Fiscalizadora.

Tabela 6 – Requisitos Funcionais

Nome	**Descrição**	**Categoria**	**Desejável**	**Permanente**
Autoridade Certificadora	Deve existir uma autoridade certificadora que forneça os certificados digitais que serão utilizados nos momentos que ocorram assinaturas digitais. As assinaturas digitais serão utilizadas para autenticar as informações oriundas da empresa e da autoridade de datação.	Segurança	( )	(X)
Autoridade de Datação	Será necessária para prover o carimbo de comprovação temporal para algumas informações fornecidas pelo sistema.	Segurança	( )	(X)
Autoridade Fiscalizadora	Trata-se de uma entidade que será utilizada garantir que os prazos tanto de atendimento quanto de fechamento da OS sejam respeitados, “serve” como mediador ou testemunha para eventuais situações de litígio.	Segurança	( )	(X)
Banco de Dados Centralizado	Será utilizado para armazenar as informações do sistema.	Implemen-tação	( )	(X)
Níveis de Usuários	Será feita no momento de acesso ao sistema garantindo a comprovação da identidade do usuário. Usuários podem ser do tipo: Empresa (privilégio total sobre o uso do sistema), Cliente	Segurança	( )	(X)

	(privilégio de incluir, consultar e confirmar fechamento da OS) e Autoridade Fiscalizadora (privilégio para consultar OS).			
Autenticação de Cliente	Supõe-se que todo cliente que usará o sistema já estará previamente cadastrado desde o momento que adquiriu produtos ou serviços da empresa.	Segurança	( )	(X)
Autenticação da Autoridade Fiscalizadora	A autoridade fiscalizadora que usará o sistema é cadastrada pelos administradores do sistemas por se tratar de uma entidade conhecida da empresa.	Segurança	( )	(X)
Tecnologias Voltadas para Web	Essas tecnologias serão utilizadas de modo a fornecer funcionalidades para implementação e funcionamento do sistema.	Implemen-tação	( )	(X)
Garantir que o sistema esteja disponível para os usuários	O sistema deverá permanecer disponível para os usuários sempre que os mesmos desejarem acessa-lo.	Segurança	( )	(X)
Garantir Segurança Contra ataques de injeção de SQL e injeção de PHP	O sistema deverá conter mecanismos capazes de detectar esses tipos de ataques, sem que estes comprometam o banco de dados do sistema e/ou os arquivos que o compõem	Segurança	( )	(X)

4.1.2 Diagrama de Casos de Uso

Os diagramas de casos de uso são utilizados para representar como o sistema deverá se comportar quando estiver pronto e para entender as regras de negócio para a qual o sistema está sendo modelado. Portanto, os diagramas de casos de uso são uma maneira eficiente de entender o ponto de vista do usuário, já que este diagrama modela somente o que os atores irão executar em determinado sistema (MATOS,2002). Os casos de uso da aplicação estão ilustrados na Figura 9.

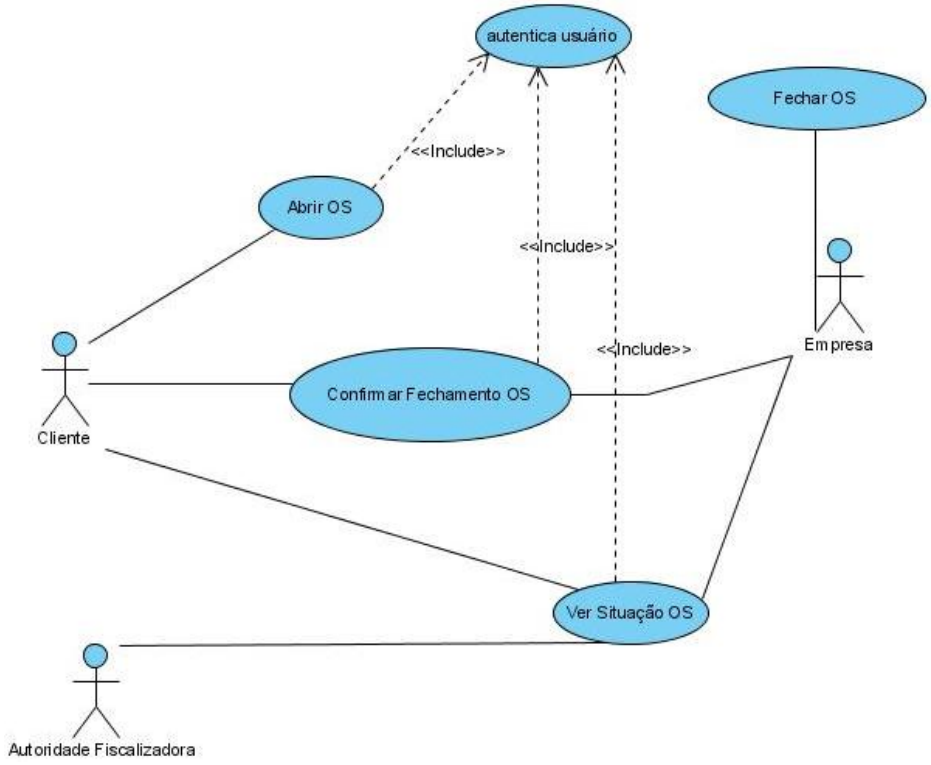

Figura 2 - Diagrama de Caso de Uso do protocolo implementado

4.1.3 Expansões dos Casos de Uso

A metodologia do UP prevê que, após a elaboração dos diagramas de casos de uso sejam feitas as expansões dos casos de uso, que fornecem uma descrição detalhada dos mesmos. Em seguida, projeta-se os diagramas de seqüência (WAZLAWICK, 2004). Sendo assim, a seguir são apresentados os detalhamentos de cada caso de uso identificado no diagrama apresentado na figura 10.

Nesta aplicação temos os seguintes atores: Cliente, Empresa que realiza o “atendimento” referente a OS do cliente e Autoridade Fiscalizadora que fiscaliza se os prazos para o atendimento estão sendo cumpridos.

Caso de Uso “Abrir OS”:

- Descrição: Este caso de uso é o início do funcionamento do protocolo, onde o cliente visita o sítio da empresa, faz sua

validação perante o sistema e abre a OS para que a empresa proceda ao atendimento.

- Atores: Cliente.
- Pré-condições: Sítio da empresa deve estar disponível.
- Fluxo principal: 1º Cliente entra no sítio da empresa. 2º Preenche formulário para autenticação. 3º Preenche formulário com a descrição do seu problema para abrir a OS. 4º Recebe o recibo da requisição protocolado que comprova a abertura da OS.
- Pós-condições: Requisição gravada no banco de dados da empresa; recibo da requisição protocolado e enviado ao cliente.

Caso de Uso “Verificar Situação da OS”:

- Descrição: Neste caso de uso Cliente, Empresa e Autoridade fiscalizadora podem verificar a situação e o andamento da OS. A autoridade fiscalizadora o utiliza para verificar se os prazos de atendimento estão sendo respeitados. O cliente verifica a situação também para constatar os prazos e para tomar conhecimento sobre as possíveis ocorrências sobre a sua OS. A empresa tem a possibilidade de verificar se o setor responsável pela OS está cumprindo com os prazos.
- Atores: Cliente, Empresa e Autoridade Fiscalizadora.
- Pré-condições: A OS deve estar aberta, Sítio da empresa deve estar disponível.
- Fluxo principal: Para o Cliente: 1º Entrar no sítio da empresa. 2º Preencher formulário para autenticação. 3º Através do número da requisição escolher a OS que deseja verificar. 4º Verificar situação e ocorrências. Para Empresa e Autoridade Fiscalizadora: 1º Verificar OS abertas. 2º Verificar prazos. 3º Enviar notificação para Autoridade Fiscalizadora.
- Pós-condições: OS aberta.

Caso de Uso “Fechar OS”:

- Descrição: Após realizar o atendimento requisitado na OS do cliente, a empresa envia para o cliente um comprovante do fechamento da OS.
- Atores: Empresa.
- Pré-condições: A OS deve estar aberta, Sítio da empresa deve estar disponível.
- Fluxo principal: 1º Entrar no sítio da empresa. 2º Preencher formulário para autenticação. 3º Buscar OS no banco de dados. 4º Solicitar fechamento da OS.
- Pós-condições: Aguardando confirmação de fechamento.

Caso de Uso “Confirmar Fechamento da OS”:

- Descrição: Tanto a empresa quanto o cliente podem confirmar o fechamento da OS. O cliente confirma o fechamento da OS quando este verifica a resolução do problema e considera-a satisfatória. A empresa pode efetuar o fechamento da OS se ocorrer o decurso de prazo, ou seja, após o envio do comprovante do fechamento da OS o cliente fica um longo período sem entrar em contato com a empresa, ela então passa a ter o poder de confirmar o fechamento.
- Atores: Cliente e Empresa.
- Pré-condições: A OS deve estar aberta, Sítio da empresa deve estar disponível.
- Fluxo principal: Para o Cliente 1º Entrar no sítio da empresa. 2º Preencher formulário para autenticação. 3º Buscar OS no banco de dados. 4º Confirmar o fechamento da OS. Para a Empresa: 1º Aguardar o decurso de prazo. 2º Confirma o fechamento a OS
- Pós-condições: Cliente on-line, Confirmação da OS gravada no banco de dados.

4.1.4 Diagramas de Seqüência

O diagrama de seqüência é um diagrama muito útil em projetos UML, pois através desse diagrama é possível representar a seqüência dos eventos em um cenário dos casos de uso desenvolvidos no projeto. Os elementos usados no diagrama de seqüência são: os atores e instâncias dos objetos constituintes do sistema. Nessa fase ainda não foram projetados os objetos internos do sistema por esses motivos o sistema é representado com apenas um objeto (caixa preta) (WAZLAWICK, 2004).

Esse diagrama mostra como se dará a seqüência da troca de mensagens da aplicação como um todo. Para cada caso de uso proposto foi desenvolvido um diagrama de seqüência.

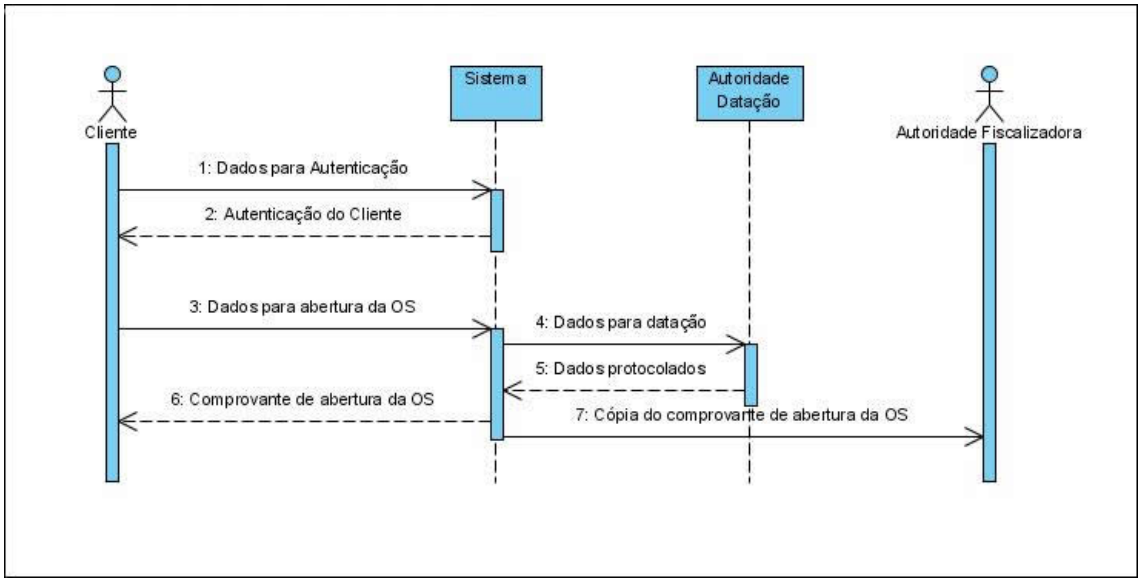

Figura 3 - Diagrama de seqüência Abrir OS

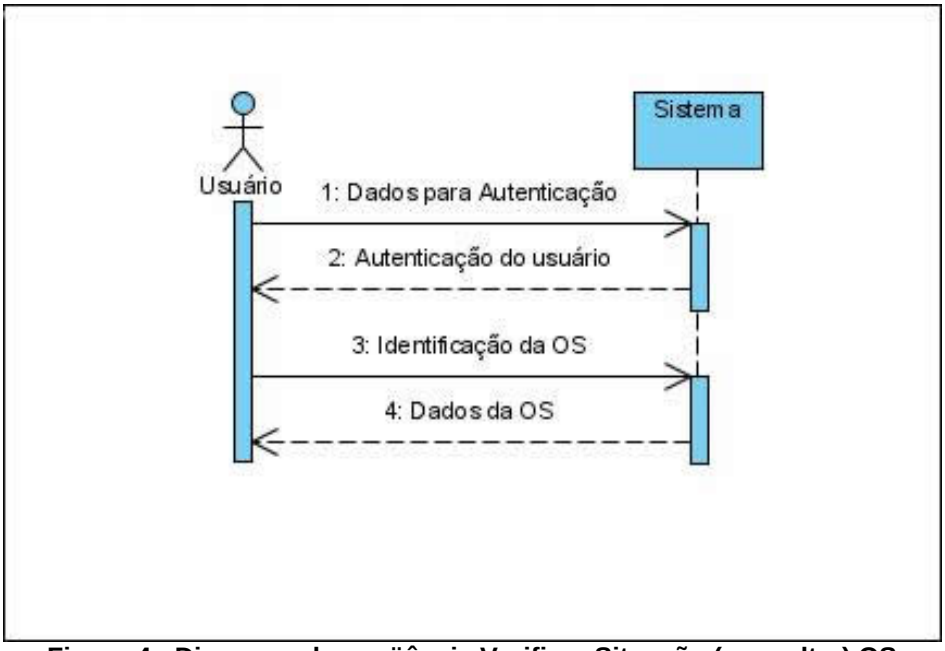

Figura 4 - Diagrama de seqüência Verificar Situação (consultar) OS

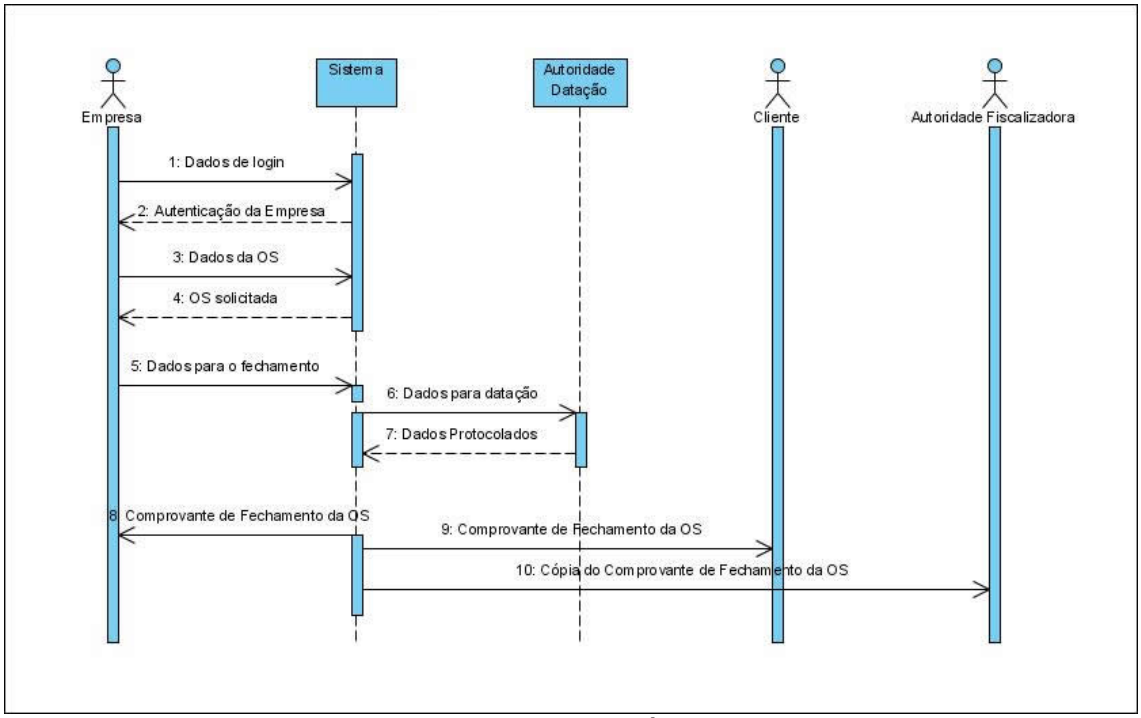

Figura 5 - Diagrama de seqüência Fechar OS

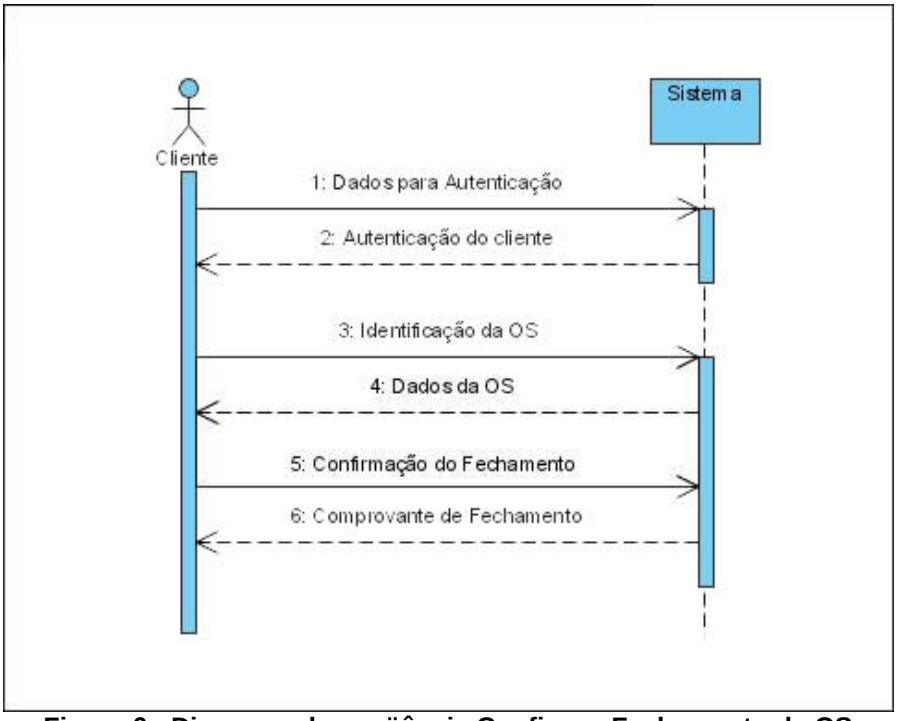

Figura 6 - Diagrama de seqüência Confirmar Fechamento da OS

Visto os diagramas de seqüência das expansões dos casos de uso pode-se passar para o próximo passo do projeto que é o modelo conceitual, que consiste de uma representação através de um Diagrama de Classes.

4.1.5 Diagrama de Classes

O diagrama de classes define as classes que o sistema necessita para ser construído, representa a estrutura e a relação entre elas servindo como modelo para os objetos. A tabela 1 contém as descrições das classes que estão presentes no diagrama de classes.

Tabela 7 – Descrição das Classes

CLASSE	DESCRIÇÃO
OS	Guarda todos os dados principais da OS, tais como, id, descrição, data de abertura e etc.
OCORRENCIA_OS	Define todas as possíveis ocorrências que podem acontecer numa OS. Registro de abertura, registro de datação, registro de fechamento,
EVENTOS	Possui os registros dos possíveis eventos que podem ocorrer em uma ocorrência da OS, tais como, datação de abertura,

	envio de comprovante, etc.
STATUS_OS	Define todos os possíveis status que uma OS pode receber. Os status possíveis que a OS pode adquirir: abertura, em curso, concluída, fechada e pendente.
USUÁRIO	Contém as informações e define os níveis de acesso para cada tipo de usuário do sistema. Os usuários podem ser do tipo: Empresa, Cliente e Autoridade Fiscalizadora.
CLIENTE	Contém os dados cadastrais dos clientes da empresa que utilizam o sistema, tais como nome, endereço, e-mail, cpf e etc.
EMPRESA	Contém os dados dos usuários do tipo Empresa, tais como *login*, senha, etc.
AUTORIDADE FISCALIZADORA	Contém os dados dos usuários do tipo Autoridade Fiscalizadora, tais como *login*, senha, denominação da entidade, responsável e etc.

Após visualizar as descrições das classes a figura 15 ilustra o diagrama de classes desenvolvido.

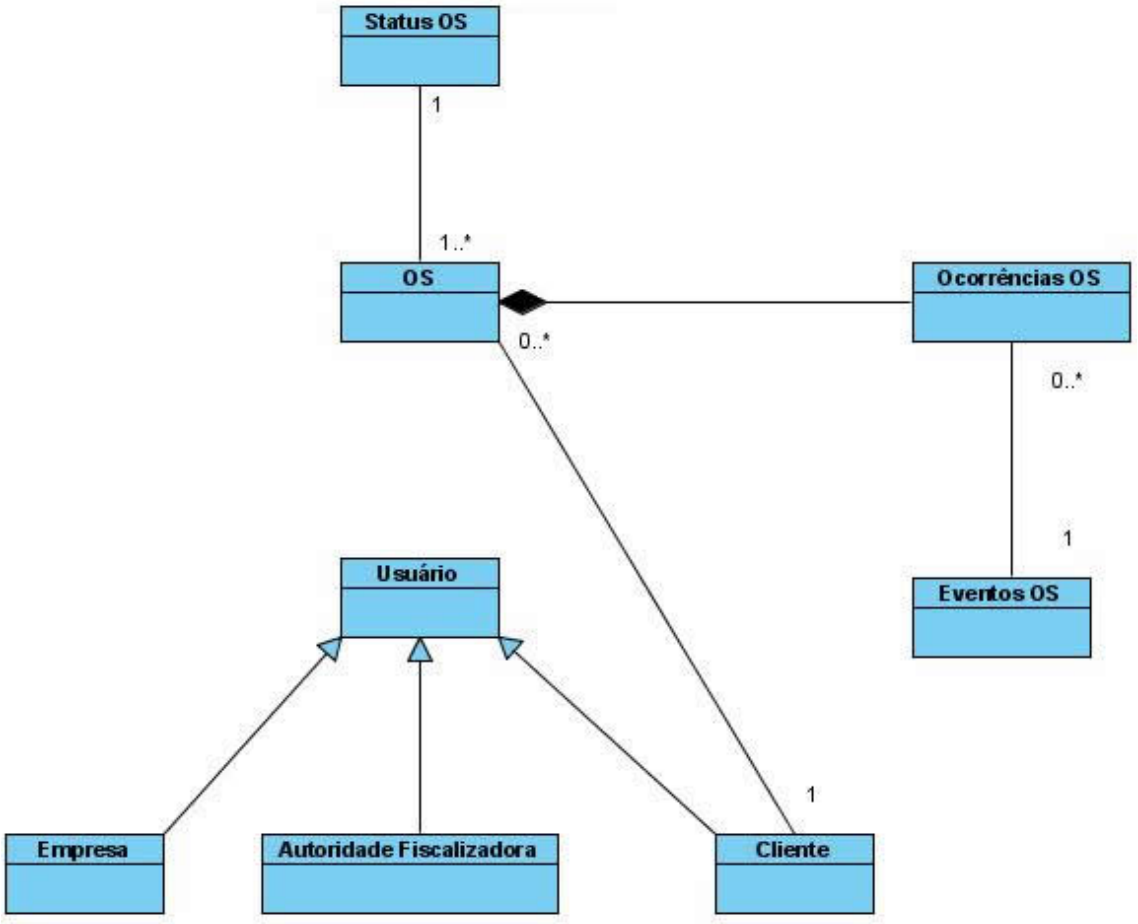

Figura 7 - Diagrama de Classes da Aplicação

Após ser mostrado os diagramas de casos de uso, seqüência e classes o próximo passo é mostrar sobre a modelagem do banco de dados desse sistema. A modelagem é ilustrada através do modelo entidade e relacionamentos situados no item 4.1.6 deste trabalho.

4.1.6 Modelo Entidade-Relacionamento

O modelo entidade e relacionamento é utilizado para a modelagem conceitual do banco de dados independentemente da ferramenta utilizada para gerenciá-lo, auxiliando na manutenção e a migração do modelo e facilita a compreensão da semântica dos dados por leigos e não-leigos. Este modelo é a descrição do banco de dados com abstração em nível de SGBD, ou seja, o modelo é apresentado sem a preocupação com a forma que será implementado. O modelo entidade e relacionamentos esta descrito na figura 16 (WIKIPÉDIA, 2005).

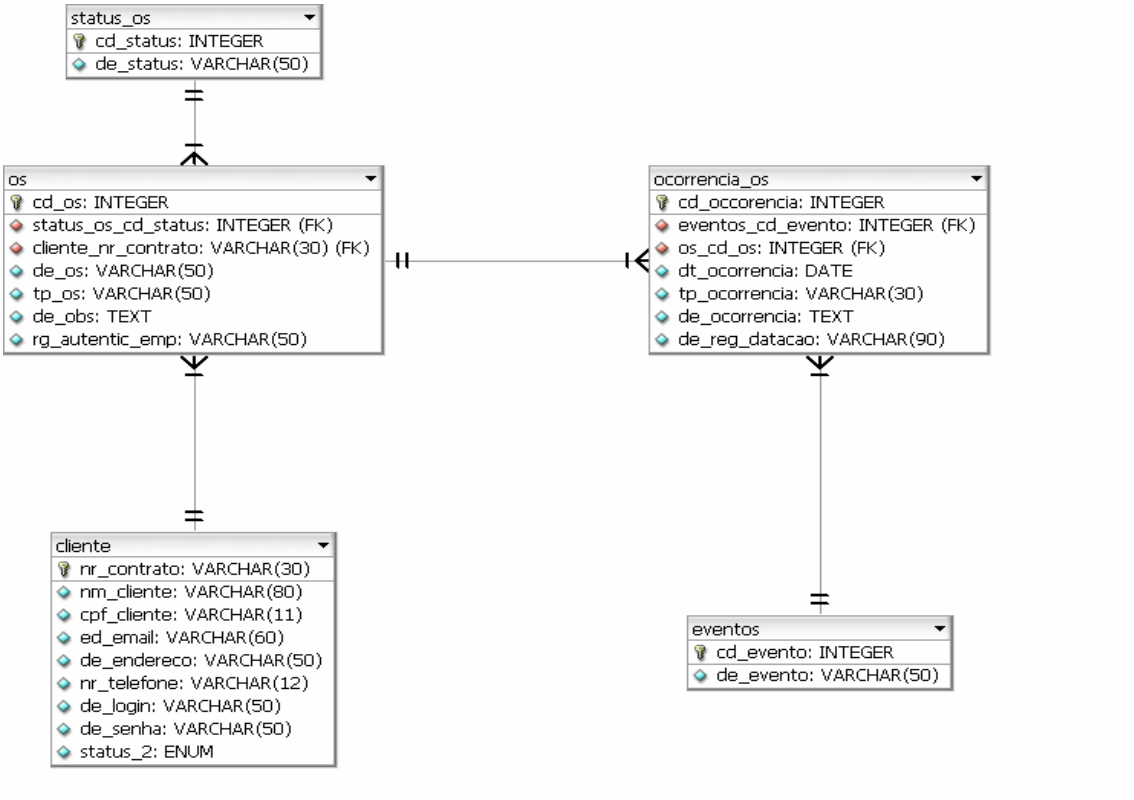

Figura 8 - Modelo Entidade e Relacionamentos (modelo conceitual)

As Tabelas 7 a 11 descrevem com mais detalhes as entidades vista no modelo entidades e relacionamentos da Figura 16. A coluna ID identificam os campos com símbolos: $ para chave primária, # para chave estrangeira e os campos que não são chave não possuem símbolo.

Tabela 7 – Descrição da Entidade OS

OS		
ID	**CÓDIGO**	**DESCRIÇÃO**
$	cd_os	Código da OS
#	cd_status	Código do Status da OS
#	nr_contrato	Número do contrato do cliente com a empresa
	de_os	Descrição da OS
	tp_os	Tipo da OS
	de_obs	Texto que contém as Observações feitas pelo cliente
	rg_autentic_emp	Registro da autenticação da OS

Tabela 8 – Descrição da Entidade Status

STATUS		
ID	**CÓDIGO**	**DESCRIÇÃO**

$	cd_status	Código do Status
	de_status	Descrição do Status

Tabela 9 – Descrição da Entidade Cliente

CLIENTE		
ID	**CÓDIGO**	**DESCRIÇÃO**
$	nr_contrato	Número do Contrato que identifica o cliente
	nm_cliente	Nome do cliente
	cpf_cliente	Número do CPF do cliente
	ed_email	Endereço de e-mail do cliente
	de_endereco	Endereço do cliente
	nr_telefone	Número do telefone para contato
	de_login	Nome de usuário para acessar o sistema
	de_senha	Senha de acesso ao sistema
	ativo	Representa o estado no sistema, ou seja, se está ativo ou inativo.

Tabela 10 – Descrição da Entidade Ocorrencia_os

OCORRENCIA_OS		
ID	**CÓDIGO**	**DESCRIÇÃO**
$	cd_ocorrencia	Código da Ocorrência da OS
$	cd_os	Código da OS
#	cd_evento	Código do evento nessa ocorrência
	dt_ocorrencia	Data da Ocorrência
	tp_ocorrencia	Tipo da Ocorrência
	de_ocorrencia	Descrição da Ocorrência
	de_reg_datacao	Grava o registro da datação feito pela AD

Tabela 11 – Descrição da Entidade Eventos

EVENTOS		
ID	**CÓDIGO**	**DESCRIÇÃO**
$	cd_evento	Código do Evento
	de_evento	Descrição do Evento

4.1.7 Diagrama de Estados de Navegação

Os diagramas de estados de navegação expõem a dinâmica que o sistema irá assumir, ou seja, modelam o sistema de forma mostrar as possíveis

situações que o usuário encontrará ao utilizar o sistema (MATOS, 2002). Na figura 17 está representado o diagrama de estados de navegação da aplicação.

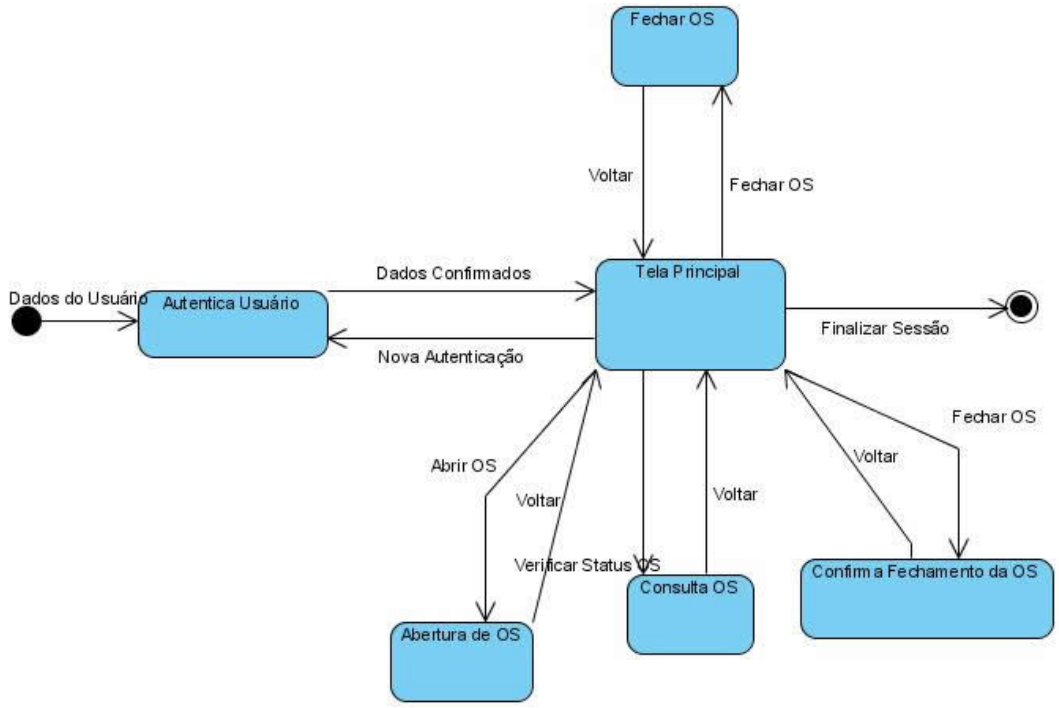

Figura 9 - Diagrama de Estados de Navegação

Com o diagrama de estados de navegação pode-se ter uma clara visão de como se dará a navegação dentro da aplicação quando esta estiver finalizada, isso proporciona a possibilidade dessa estrutura navegacional ser redefinida caso não agrade o cliente.

4.1.8 Telas do Sistema

Neste trecho serão apresentadas as telas com o *layout* proposto para a aplicação. A figura 18 contém a tela para o usuário efetuar a autenticação no sistema.

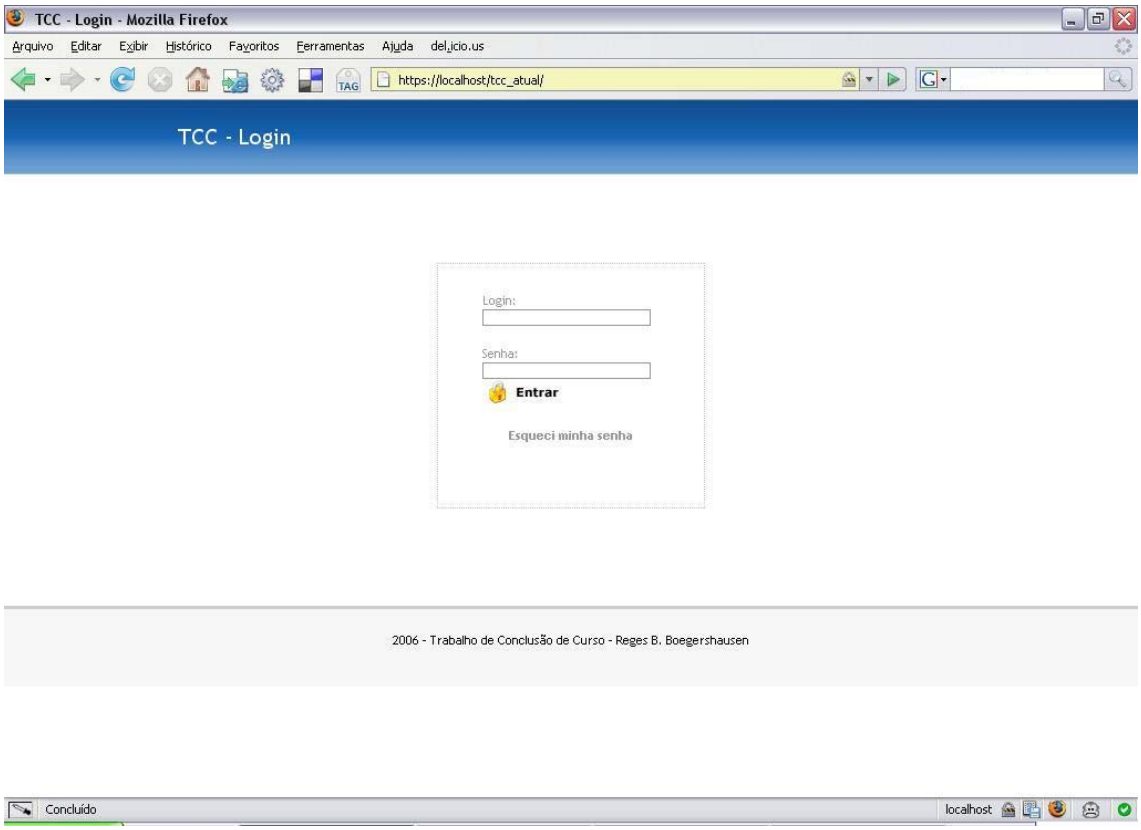

Figura 10 - Tela de Login do Usuário no Sistema

A tela principal do sistema, descrita na figura 19, possui os *links* para as funcionalidades que o sistema oferece para um usuário autenticado. As funcionalidades estão: cadastro ou abertura de OS, consulta de OS, fechamento de OS, listagem de OS's abertas e alteração de senha de acesso. Vale lembrar que os usuários realizam operações apenas nas OS's criadas por si próprio, não é permitido acesso a OS de outros usuários de maneira nenhuma.

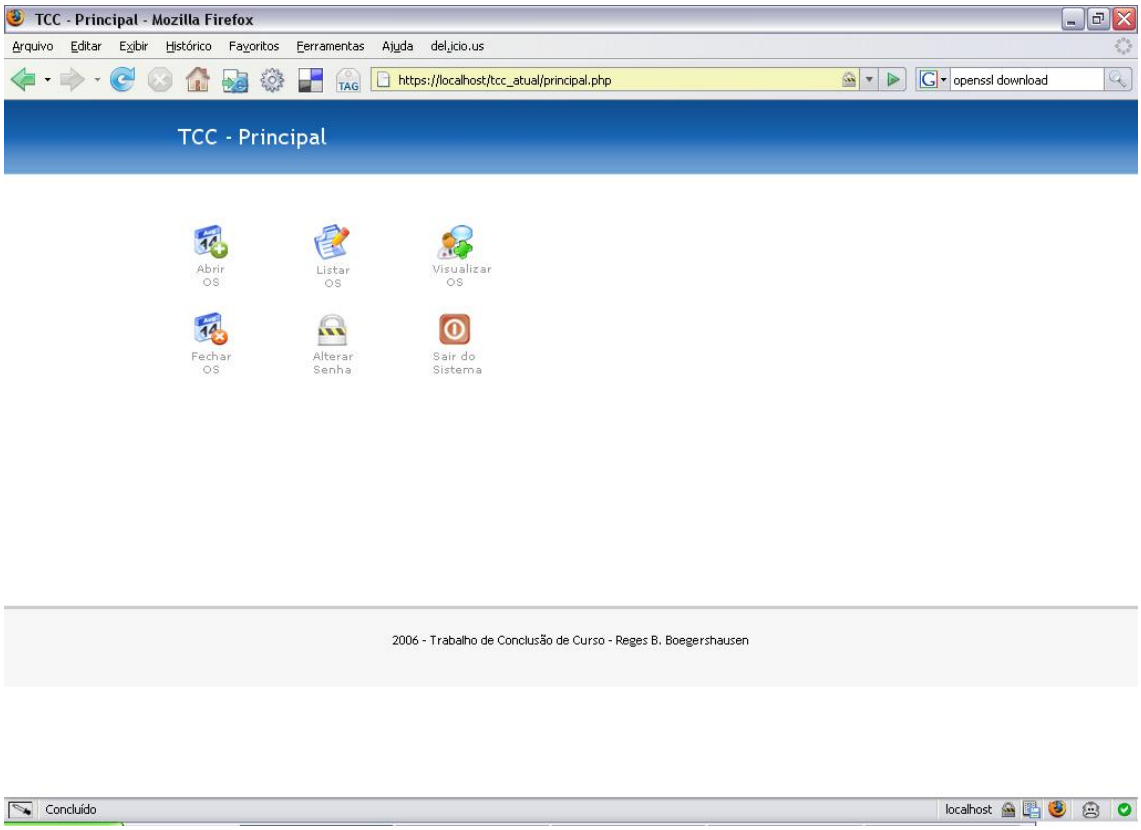

Figura 11 - Tela Principal

A funcionalidade “Abrir OS” é demonstrada na figura 20. Nessa tela o cliente preenche todos os dados obrigatórios relevantes para a abertura da OS. Os processos decorrentes da abertura da OS, descrito no protocolo serão explicados no item 4.2 deste capítulo.

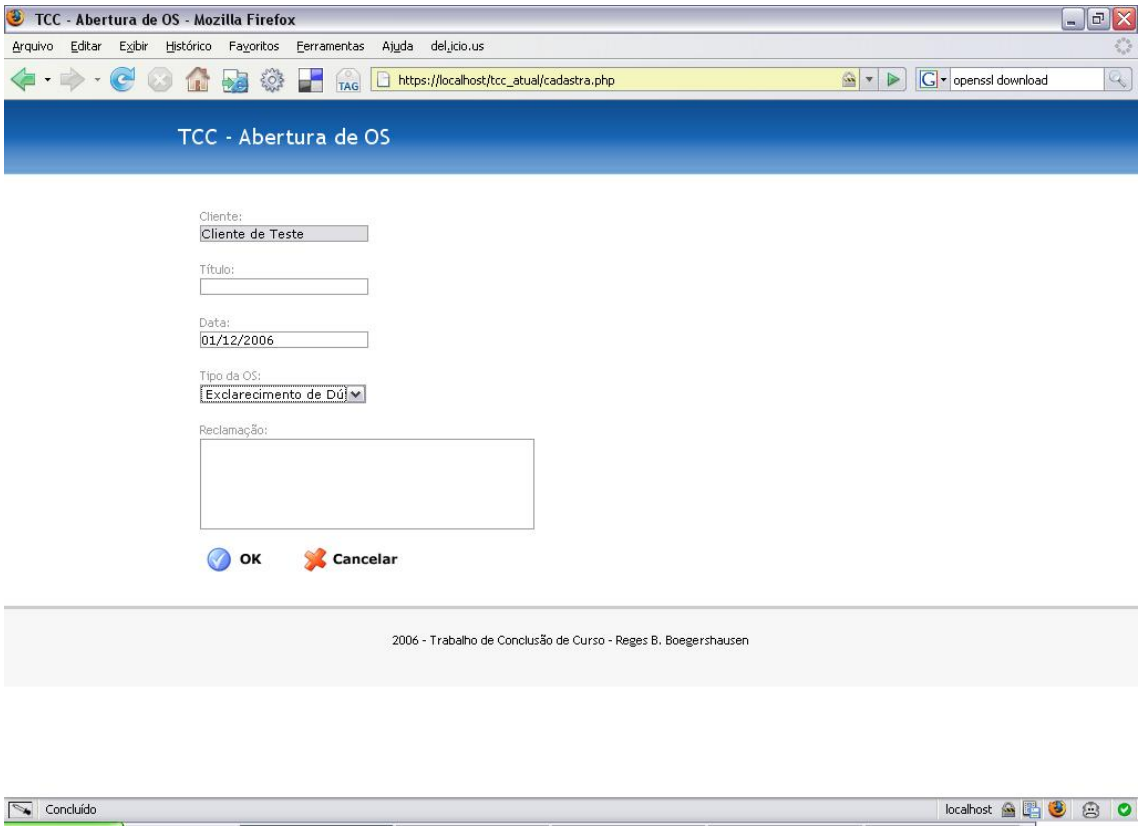

Figura 12 - Abrir OS

No fechamento o cliente verifica se o atendimento da OS está de acordo com a sua requisição, caso ele constate que foi atendido pode fechar a OS, senão pode entrar em litígio com a empresa. A tela de fechamento está ilustrada na figura 21.

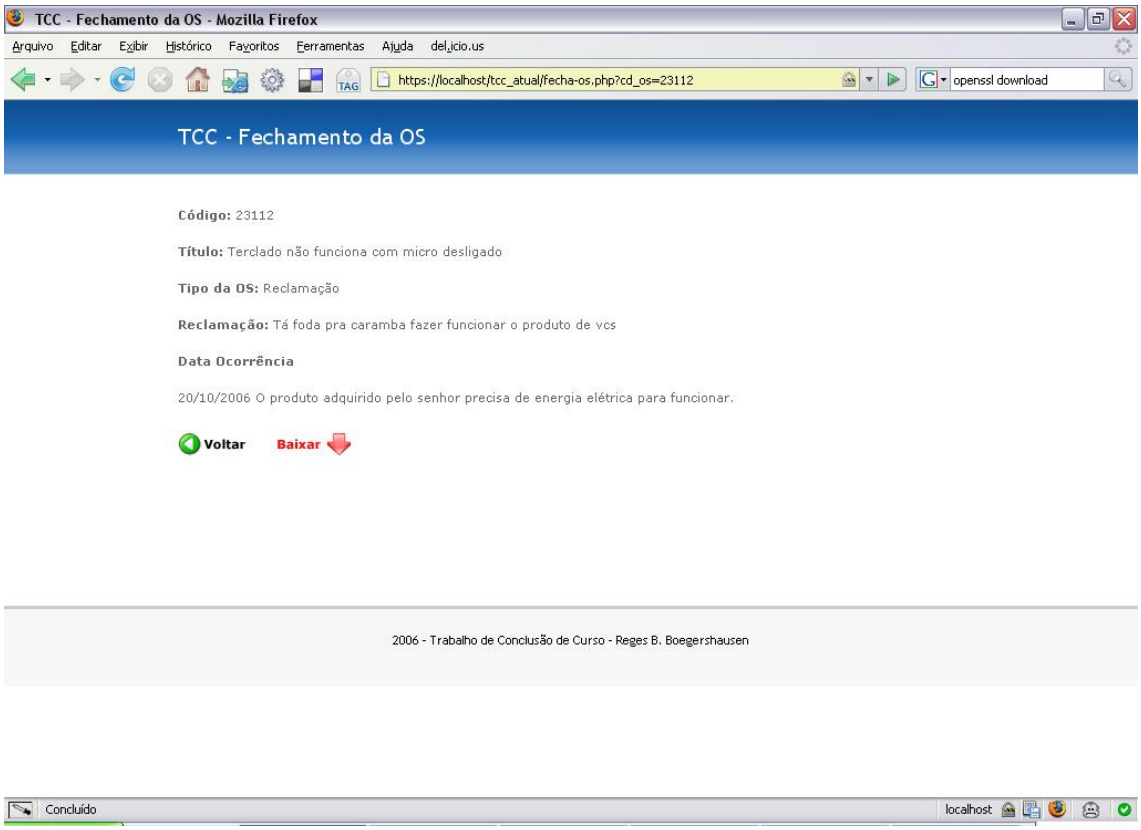

Figura 13 - Fechamento da OS

A tela de alterar a senha de acesso está demonstrada na figura 22, nessa tela o usuário digita a atual e a nova senha, com isso ela atualiza a sua senha de acesso.

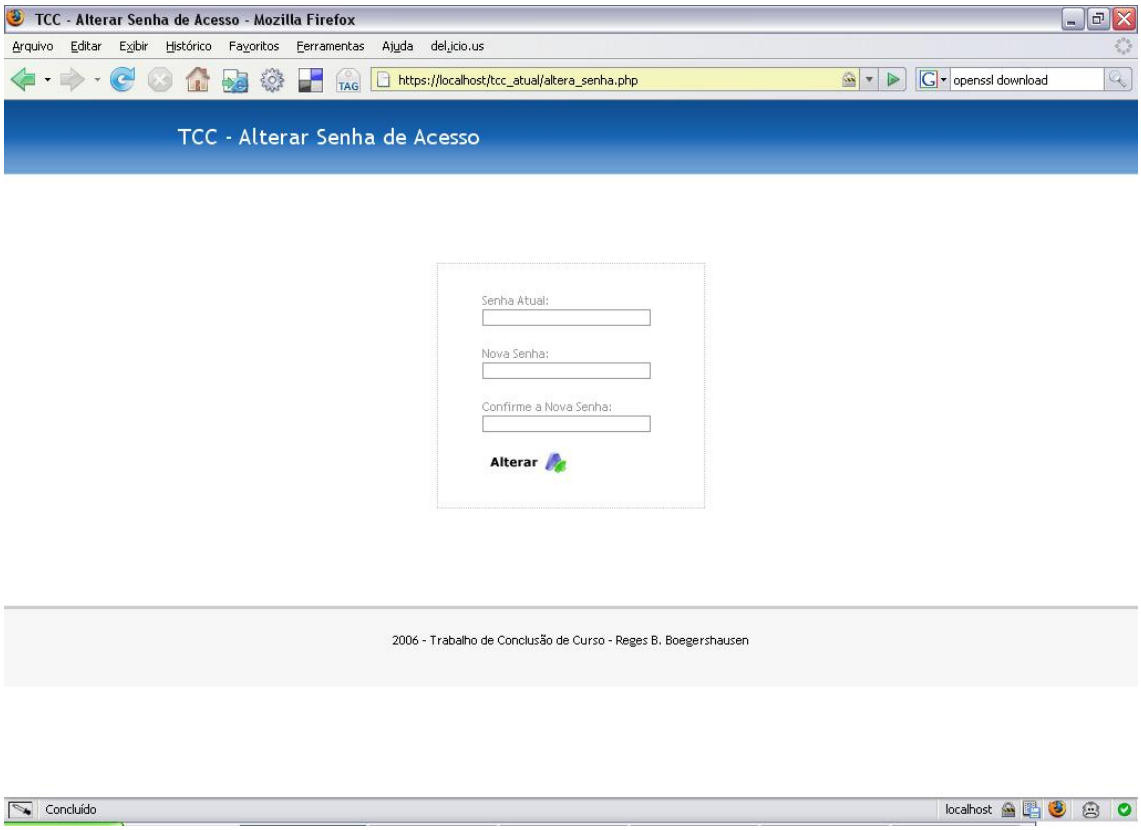

Figura 14 - Alterar Senha de Acesso

Após a confecção do *layout* para a aplicação o passo seguinte é fazer as implementações das funções que correspondem aos passos do protocolo proposto por Amauri Sant'Anna Ghisleri. A discussão da implementação está descrita no item 4.3 deste capítulo. Porém será feita uma breve sobre as tecnologias e ferramentas utilizadas para o desenvolvimento da aplicação no item 4.2.

4.2 Tecnologias Escolhidas para o Desenvolvimento da Aplicação

Após vislumbrar o *layout* da aplicação se faz necessário discorrer sobre as tecnologias e ferramentas utilizadas para a implementação da aplicação,

como por exemplo, linguagem de programação, SGBD e etc. Este tópico do trabalho trata sobre essa explanação.

4.2.1 Linguagem de Programação

O PHP é uma linguagem de programação voltada para a WEB que possibilita a construção de sistemas dinâmicos com uma interface amigável, oferecendo ao usuário do sistema uma maior facilidade na utilização do sistema. Além de ser uma linguagem totalmente gratuita e com código fonte aberto, podendo assim ser utilizada livremente sem custos adicionais para o desenvolvimento de aplicações e modificada dependendo da necessidade do usuário (PHP,2006).

Uma grande vantagem em utilizar o PHP: Portável para diversos sistemas operacionais, possibilidade de conexão com os melhores e mais populares bancos de dados do mercado, exemplos Oracle, SQL Server, MySQL, PostgreSQL etc (MUTO, 2004).

Dentre as características que ajudaram na escolha do PHP para o desenvolvimento da aplicação, destacam-se as grandes variedades de funções nativas para diversos fins sem a necessidade de adquirir ou instalar essas funções de outros desenvolvedores.

4.2.2 Sistema Gerenciador de Banco de Dados MySQL

O *MySQL* é um sistema de gerenciamento de banco de dados relacionais baseado em comandos *SQL* (*Structured Query Language - Linguagem Estruturada para Pesquisas*) que vem ganhando grande popularidade, sendo atualmente um dos bancos de dados mais populares, com mais de 4 milhões de instalações (SOARES, 2001).

Este banco de dados é reconhecido pelo seu desempenho e robustez e também por ser multitarefa e multiusuário. Outra grande vantagem é a de ter código aberto e funcionar na maioria dos sistemas operacionais entre eles GNU/Linux e Windows (MYSQL, 2006).

O MySQL é um banco de dados altamente confiável que possibilita uma total integração com a linguagem de programação PHP. Este bando de dados é distribuído de duas maneiras diferentes: a versão *community*, que é gratuita voltada para aplicações *open source* e a versão *enterprise* que é cobrada uma licença e pode ser utilizada em aplicações de uso comercial.

4.2.3 ADOdb

A ADOdb é uma biblioteca para abstração de acesso a bancos de dados desenvolvida para ser utilizada com as linguagens Python e PHP. Essa biblioteca é distribuída sob a licença LGPL possibilitando que seja empregada em aplicações comerciais sem infligir nenhuma lei (ADODB,2006).

A ADOdb disponibiliza suporte aos principais SGBDs disponíveis atualmente no mercado, como por exemplo, ORACLE, MySQL, SQL Server, PostgreSQL, além de oferecer suportar também conexão via *drivers* ODBC e ADO, suprindo assim as necessidades dos desenvolvedores que necessitam portabilidade de SGBDs (ADODB, 2006).

Outra característica interessante sobre essa biblioteca é a facilidade com que o desenvolvedor pode mudar a conexão de um banco de dados para outro diferente, a figura 4.11 mostra na linha 3 um exemplo de conexão com MySQL e na linha 4 um exemplo com PostgreSQL.

```
1 <?
2 require("adodb/adodb.inc.php");
3 $conn = NewADOConnection("mysql");
4 $conn = NewADOConnection("postgre");
5 $conn->Connect("localhost","root","rootroot","tcc");
6
7 ?>
```

Figura 4.11 - Exemplo de conexão

Vistas estas características é possível notar a simplicidade que o desenvolvedor tem caso haja a necessidade de seu software migrar de SGDB.

4.2.3 OpenSSL

É um conjunto de ferramenta de software livre que implementam os protocolos SSL(*Secure Socket Layer*)v2/v3 e o TLS(*Transport Layer Security*)v1, funções criptográficas de uso comum além de ferramentas para manipulação de assinaturas digitais, manipulação de certificados digitais, funções *hash* e algoritmos de criptografia assimétrica e simétrica (OPENSSL, 2006).

Dentre as funcionalidades providas pelo OpenSSL estão a PKCS(*Public-Key Cryptography Standards*) e o protocolo X509. O PKCS são especificações desenvolvidas pelo laboratório RSA (Rivest, Shamir e Andleman) que utilizam algoritmos de criptografia assimétrica para a troca segura de informações, fazer assinaturas digitais, funções de resumo (*hash*) e entre outras funções. O protocolo X509 é um padrão utilizado para manipulação de certificados digitais baseados em criptografia de chave pública para provar a identidade de alguém ou de algum servidor na Internet (HEINLEIN, 2006).

Outra funcionalidade importante disponível neste conjunto de ferramentas é o *Timestamp*, que é utilizado como “carimbo” que comprova a criação da mensagem data informada e gera uma referência temporal confiável (OPENSSL, 2006).

Esse conjunto de ferramentas tem a vantagem de ser distribuído num tipo de licença na qual o uso para aplicações comerciais não se torna ilegal.

4.2.4 SOAP e Web Services

SOAP (*Simple Object Access Protocol*) é um protocolo especificado pelo consórcio W3C que está se tornando padrão na troca de mensagens entre *web services* e aplicações distribuídas. Este protocolo foi projetado para fazer chamadas a aplicações remotas através de RPC (*Remote Procedure Calls*) em aplicações independentes de plataforma e linguagem de programação (W3C, 2006).

As mensagens trocadas pelo SOAP baseiam-se em documentos XML para trocar informações entre as aplicações e são formadas basicamente pelos seguintes elementos (W3C, 2006):

- Envelope: É o elemento raiz do documento XML, contém a definição do estilo de codificação do documento, declarações de atributos e a representação dos dados no documento.
- *Header*: Este cabeçalho pode ser suprimido por carregar apenas informações adicionais que não são fundamentais para o processamento da mensagem.
- *Body*: Elemento obrigatório no documento, pois essa parte contém a informação transportada para o destino. Esse elemento pode conter um outro elemento adicional que é o "*Fault*" que é encarregado de transportar informações sobre erros e *status* decorrentes do processamento da mensagem.

Web Services são soluções desenvolvidas para a comunicação entre aplicações distintas e para a integração de sistemas, ou seja, esse tipo de solução permite que aplicações desenvolvidas em plataformas diferentes se tornem compatíveis, já que eles utilizam o padrão XML e SOAP para a troca de mensagens (CHAMPION, 2006).

Os *Web Services* utilizam a WSDL (*Web Services Definition Language*) que contém as definições do serviço a ser utilizado, tais como, métodos a serem invocados, tipos de dados corretos, operações e suas interfaces. Ao acessar um serviço o cliente monta a mensagem de acordo com as instruções encontradas no documento WSDL, com isso o serviço sabe como tratar e processar a mensagem recebida e como construir a resposta para quem o requisitou (CHAMPION, 2006).

4.3 Implementação da Aplicação

Neste item será visto como a aplicação foi implementada seguindo as características do protocolo proposto descrito na figura 6, para auxiliar no entendimento será explicado conforme seus passos.

4.3.1 Autoridade de Datação

A principal função da AD de acordo com a especificação do protocolo proposto por Amauri Sant'Anna Ghisleri é protocolar a mensagens que serão trocadas entre cliente e empresa, dando a essas mensagens noção e validade temporal.

Esta protocolização será feita através da função protocolar, essa função recebe o resumo da mensagem da empresa, verifica a assinatura e através de funções oferecidas na classe *openssl* do PHP realizada o "carimbo" na mensagem que lhe foi passada.

O código-fonte do arquivo datacao.php que contém a função protocolar está disponível no anexo A.

4.3.2 – Autoridade Fiscalizadora

A AF tem como principais finalidades receber as cópias das requisições e armazená-las de forma que as tornem disponíveis para eventuais ações litigiosas entre as partes envolvidas. Para realizar sua função a AF recebe uma cópia das OS abertas e dos comprovantes gerados durante a vigência do protocolo e os armazena em seu banco de dados para que estejam disponíveis para eventuais situações de litígio entre a empresa e o cliente.

4.3.3 – Interação Cliente e Empresa

Após o preenchimento do formulário proposto (figura 4.8) a empresa recebe esses dados através do programa cadastra.php disposto no anexo A. Este programa monta a requisição de atendimento, gera um resumo (*hash)* da mensagem e assina utilizando o certificado digital da empresa. Neste processo as funções para resumo e assinatura digital são fornecidas pelo pacote *openssl*, para os resumos é utilizada a algoritmo MD5, para o processo de assinaturas digitais as funções do padrão PKCS e para "carimbar" os recibos gerados as funções de *Timestamp* e do protocolo X509.

Feita a assinatura digital, a empresa envia para a autoridade de datação o resumo que recebe com o programa datacao.php,que também se encontra disponível no anexo A, este verifica a veracidade da assinatura digital. Se a verificação for bem sucedida a requisição é protocolada, o recibo é gerado contendo o “carimbo” e a assinatura digital da AD. De posse do recibo a empresa envia por e-mail para o cliente junto com a requisição de atendimento e uma cópia é enviada para a AF que armazena em seus bancos de dados. A troca de informações entre a aplicação com a AD e com a AF da-se através dos *web services*.

Essas funcionalidades aqui implementados correspondem aos passos 2 a 5 e 6 a 9 do protocolo proposto (figura 3.1).

4.3.4 – Confirmar Fechamento da OS

Ao receber o aviso de que sua requisição foi atendida o cliente recebe também um *link* com o endereço para confirmar o fechamento da OS. O cliente ao visitar esse *link* deve efetuar a autenticação de usuário e assim poderá efetuar a confirmação clicando no botão fechar (figura 4.9).

4.4 Testes e Validação da Aplicação

A fase de validação e testes tem como objetivo minimizar o acontecimento de erros e riscos associados a estes erros, verificar se as técnicas e a metodologia de desenvolvimento estão sendo aplicadas de forma adequadas. Além de assegurar que o software em desenvolvimento está de acordo com os requisitos do cliente (ROCHA, et. al., 2001).

Dentre as atividades que compõem a fase de testes e validação têm-se as atividades denominadas análise estática e análise dinâmica do software em desenvolvimento. Essas análises podem ser realizadas manualmente ou automaticamente de acordo com a disponibilidade de recursos destinados ao projeto (NOGUEIRA, 1991).

Segundo ROCHA, et. al. (2001) a análise estática não envolve a execução do software propriamente dita, ela visa determinar propriedades

válidas para qualquer execução do software final, ou seja, essa análise define um conjunto mínimo de características funcionais definidas na especificação do software para que este funcione de forma adequada.

A análise estática deve ser realizada em qualquer fase do desenvolvimento do software diminuindo o esforço gasto futuramente em manutenção e correção de erros. Um exemplo clássico de análise estática são as revisões feitas através de *checklists* desenvolvidas e agendadas para cada fase do processo de desenvolvimento.

A análise dinâmica também objetiva detectar erros e inconformidades no software, porém essa fase envolve a execução propriamente dita da aplicação em desenvolvimento testando-a com a finalidade de encontrar erros e aumentar a confiança de que o software esteja correto e de acordo com a especificação (NOGUEIRA, 1991).

Nessa fase devem ser avaliadas todas as necessidades e opções que determinam quais comportamentos são permitidos e aceitáveis durante a execução desse software por parte do usuário final.

Em suma as atividades de teste e validação têm como objetivo prevenir e apontar erros e falhas decorrentes do processo de desenvolvimento do software. Os resultados dessas etapas devem ser documentados de acordo com os padrões que a organização adota no seu cotidiano.

As atividades de teste de software possuem três fases distintas: de unidade, de integração e de sistema. O primeiro objetiva identificar erros de lógica e de implementação em cada módulo separadamente nas menores partes funcionais de um programa que podem ser executadas (rotinas, sub-rotinas e etc.). Os testes de integração visam testar os erros associados às interfaces entre os módulos interados, ou seja, testar quais as conseqüências de um erro que ocorre num módulo em relação a outro que depende da sua execução correta. A última fase, teste de sistema visa identificar erros de funções e características de desempenho que não façam parte da especificação do cliente (ROCHA, et. al., 2001).

As atividades de teste podem ser definidas de forma que em cada fase diferente sejam testados diferentes tipos de erros em diferentes escopos do

software sejam descobertos e diferentes aspectos do software sejam testados, dessa forma facilitam a escolha dos critérios e cenários para teste do software.

Tendo em mãos a aplicação implementada e a fundamentação sobre a elaboração e realização dos testes é possível realizá-los e discuti-los.

4.4.1 – Discussão dos Testes

Para a melhor organização e realização dos testes foi adotada uma tabela elaborada pelo DataSUS (departamento de informática doe ministério da saúde), pois essa tabela é de fácil entendimento e boa organização (DATASUS, 2006). A escolha da tabela foi meramente pela praticidade em reunir as informações necessárias para montagem dos casos de testes desenvolvidos. Os casos de testes foram montados a partir das tabelas de requisitos funcionais e também da tabela de requisitos suplementares.

A organização dos testes foi feita da seguinte forma: inicialmente descreveu-se os testes dos requisitos suplementares. Posteriormente, foram montados os cenários de testes dos requisitos funcionais.

Salienta-se que os requisitos suplementares não foram testados em termos de casos de testes pela sua característica peculiar de não referir-se diretamente a funções do sistema, como é o caso dos requisitos funcionais. A única exceção neste aspecto refere-se ao requisito de disponibilidade, como será explicado mais adiante.

4.4.1.1 – Teste do requisito suplementar: Autoridade Fiscalizadora

A autoridade fiscalizadora atua no sistema como um órgão com credibilidade entre as partes envolvidas no sistema, para fiscalizar o cumprimento ou não nos prazos de atendimento e para servir como mediadora numa eventual situação litigiosa. Dependendo da área de atuação da empresa que utilizará o sistema a autoridade fiscalizadora pode ser algum órgão regulador, por exemplo, na área de telefonia a Anatel, na área de energia elétrica a Aneel, entre outros exemplos.

No caso deste trabalho a autoridade fiscalizadora está sendo simulada como um órgão responsável por determinada área, pois se tratando de um trabalho acadêmico o sistema não tem uma área específica de atuação.

Sendo assim, não foi projetado um caso de teste para este requisito, já que o objetivo seria confirmar a existência e confiabilidade pública desta entidade.

4.4.1.2 – Teste do requisito suplementar: Autoridade de Datação

A autoridade de datação utilizada para as simulações e testes foi a openTSA, que fornece tanto o serviço *on-line* gratuito quanto os programas sob a mesma licença que o pacote de ferramentas OpenSSL.

Por tratar-se de uma autoridade de datação publicamente reconhecida e confiável, entende-se que este requisito foi adequadamente atendido.

4.4.1.3 – Teste do requisito suplementar: Tecnologias Voltadas para Web

As tecnologias escolhidas para o desenvolvimento da aplicação foram descritas no capítulo 4.2 do presente trabalho, estas tecnologias fornecem a interface para o usuário final, as funcionalidades para o processamento dos dados. Além disso, com o uso desse tipo de tecnologia o processamento das informações é realizado no servidor tornando mais fácil o gerenciamento e manutenção do sistema.

Deste modo, este requisito está sendo atendido conforme especificado.

4.4.1.4 – Teste do requisito suplementar: Garantir que o sistema esteja disponível para os usuários

Segundo Pressman (2002), confiabilidade de um software é definida em termos estatísticos como a probabilidade de um software operar livre de falhas, num ambiente especificado, durante um tempo especificado. Ou seja, dizer

que um software tem confiabilidade 0,96 é o mesmo que dizer que entre as 100 tentativas de executá-lo 96 dessas tentativas de execução são bem sucedidas.

Vale salientar que a tabela 12, que apresenta os testes deste requisito foi adaptada a partir da tabela-padrão usada para os testes dos requisitos funcionais. Esta adaptação refere-se especialmente aos cenários de testes por conterem características temporais muito particulares.

Tabela 13 – Testes do requisito Garantir que o sistema esteja disponível para os usuários

Requisito	**Garantir que o sistema esteja disponível para os usuários**
Descrição	Para testar esse requisito serão realizados vários acessos ao sistema, para realizar as operações disponíveis num determinado período de tempo.
Tipo de teste	Teste de sistema
Abordagem	Teste de caixa preta
Técnica	Teste realizado de forma manual
Dados do Teste	
Pré-condições	Sistema hospedado da internet e disponível para acesso
Pós-condições	
Observações	Esse teste foi realizado com o sistema hospedado na internet em uma conta do provedor de hospedagem www.maisemconta.com
Passos	
1	Acessar o sistema
2	Autenticar-se no sistema
3	Abrir OS
4	Verificar situação da OS
5	Fechar OS
6	Confirmar Fechamento da OS

Cenário	**Data**	**Hora**	**Tempo de Acesso**	**Tempo de Disponibilidade**	**% do tempo em disponibilidade**
1	23/10/2006	9:00	15 min	15 min	100%
2	23/10/2006	13:00	15 min	15 min	100%
3	23/10/2006	17:00	15 min	15 min	100%
4	23/10/2006	22:00	15 min	15 min	100%
5	24/10/2006	9:00	20 min	20 min	100%
6	24/10/2006	13:00	20 min	20 min	100%
7	24/10/2006	17:00	20 min	20 min	100%
8	24/10/2006	22:00	20 min	20 min	100%
9	25/10/2006	9:00	20 min	20 min	100%
10	25/10/2006	13:00	20 min	20 min	100%
11	25/10/2006	17:00	20 min	20 min	100%
12	25/10/2006	22:00	20 min	20 min	100%
13	26/10/2006	9:00	15 min	15 min	100%
14	26/10/2006	13:00	15 min	15 min	100%
15	26/10/2006	17:00	15 min	15 min	100%
16	26/10/2006	22:00	15 min	15 min	100%
17	26/10/2006	9:00	15 min	15 min	100%
18	26/10/2006	13:00	15 min	15 min	100%
19	26/10/2006	17:00	15 min	15 min	100%

20	26/10/2006	22:00	15 min	15 min	100%
21	27/10/2006	10:00	20 min	20 min	100%
22	27/10/2006	12:30	20 min	20 min	100%
23	27/10/2006	17:30	20 min	20 min	100%
24	27/10/2006	21:00	20 min	20 min	100%
25	28/10/2006	10:00	15 min	15 min	100%
26	28/10/2006	12:30	15 min	15 min	100%
27	28/10/2006	17:30	15 min	15 min	100%
28	28/10/2006	21:00	15 min	15 min	100%
29	29/10/2006	10:00	20 min	20 min	100%
30	29/10/2006	12:30	20 min	20 min	100%
31	29/10/2006	17:30	20 min	20 min	100%
32	29/10/2006	21:00	20 min	20 min	100%
33	30/10/2006	10:00	15 min	15 min	100%
34	30/10/2006	12:30	15 min	15 min	100%
35	30/10/2006	17:30	15 min	15 min	100%
36	30/10/2006	21:00	15 min	15 min	100%
37	01/11/2006	10:00	15 min	15 min	100%
38	01/11/2006	12:30	15 min	15 min	100%
39	01/11/2006	17:30	15 min	15 min	100%
40	01/11/2006	21:00	15 min	15 min	100%

Para que o sistema esteja disponível para os usuários sempre que estes o acessem a empresa deve contratar uma empresa de hospedagem confiável, com boa referência no mercado e com infra-estrutura satisfatória. Caso a empresa decida hospedar o sistema em seus próprios servidores ela deve investir em *links* de conexão que comportem o volume de dados gerado pelo tráfego, estrutura física para evitar que fenômenos naturais interfiram no funcionamento, aparelhos como *no-breaks*, climatização do ambiente, segurança física dos servidores e etc.

O teste desse requisito está, portanto, totalmente dependente das situações supracitadas. O ambiente de simulação não apresentou nenhum problema de indisponibilidade durante os testes realizados.

4.4.1.5 – Teste do requisito suplementar: Garantir segurança contra taques de injeção de SQL e injeção de PHP

Ataques de injeção de SQL são vulnerabilidades que ocorrem em nível de banco de dados numa aplicação, ou seja, é um subconjunto de vulnerabilidades exploradas através de entradas do usuário do sistema. A idéia desse tipo de ataque é a de que a aplicação, ao processar os comandos

programados execute também os comandos digitados pelo atacante (SHIFLETT, 2005). A figura 23 ilustra um exemplo de como pode ser realizado um ataque dessa natureza.

Figura 15 - Exemplo de Ataque de Injeção de SQL

Um usuário mal-intencionado entra com comandos SQL em campos de formulários com intuito de acessar dados não permitidos ou causar sérios danos á integridade da base de dados. Esse ataque também pode ser articulado através de parâmetros que são passados via URL do site, como por exemplo, http://www.seusitio.com.br?id=12'; drop table users --.

Além desse tipo citado, o outro tipo que se deve prever é o ataque de injeção de PHP. Pode-se dizer que os intuitos desse ataque são modificar ou driblar a forma de funcionamento desse programa adicionando comandos externos que não foram programados originalmente no sistema (SHIFLETT, 2006).

Esses tipos de ataques serão bem-sucedidos se ao resgatar as variáveis vindas de um formulário e/ou passadas via parâmetro na URL não for realizado nenhum tipo de filtragem e/ou validação das mesmas. Ou seja, dentro da aplicação deve ser verificado se o tipo de dado é o correto, se existem caracteres inválidos ou palavras maliciosas, ou seja, não utilizar dados entrados pelo usuário sem antes fazer algum tipo de validação.

```php
<?
function anti_injection($sql)
{
    // remove palavras que contenham sintaxe sql
    $sql = preg_replace(sql_regcase("/(from|select|insert|delete|where|drop table|show tables|
    #|\*|--|\\\\)/"),"",$sql);
    $sql = trim($sql);//limpa espaços vazio
    $sql = strip_tags($sql);//tira tags html e php
    $sql = addslashes($sql);//Adiciona barras invertidas a uma string
    return $sql;
}
```

Figura 16 - Função que previne ataques de injeção de SQL e injeção de PHP

A figura 24 possui a função que foi desenvolvida para prevenir os ataques citados, essa função foi desenvolvida usando mecanismos simples disponíveis na própria linguagem PHP sem a necessidade de adicionar componentes ou bibliotecas desenvolvidas por terceiros. A seguir os testes e seus resultados serão mostrados.

Tabela 14 – Testes do requisito Garantir segurança contra ataques de injeção de SQL e injeção de PHP

Requisito	**Garantir segurança contra taques de injeção de SQL e injeção de PHP**	
Descrição	Neste teste o requisito testado sofrerá tentativas de injeção de comandos SQL e comandos PHP para desestabilizar o sistema.	
Tipo de teste	Teste de unidade	
Abordagem	Teste de caixa preta	
Técnica	Teste realizado de forma manual	
Dados do Teste		
Pré-condições	O usuário deve estar em algum ponto do sistema	
Pós-condições		
Passos		
1	Acessar o sistema	
2	Preencher o formulário ou URL com diferentes tipos de ataque	
Cenário	**Saída Esperada**	**Campo**
1	Erro	‘; DROP TABLE usuarios; --
2	Erro	‘ OR 1 = 1; --
3	Erro	‘; GRANT ALL PRIVILEGES ON *.* TO ‘usuario_bd’@’%’ WITH GRANT OPTION; --
4	Erro	‘; DROP DATABASE tcc; --
5	Erro	‘; TRUNCATE TABLE usuarios; --
6	Erro	‘; SELECT * FROM usuarios; --
7	Erro	‘; show tables;--
8	Erro	“; eval(“print_r($_SESSION)”);
9	Erro	“; eval(“unlink(index.php)”);
10	Erro	“; system(“rn *.*”);
11	Erro	“; system(“del *.*”);

Resultados Obtidos:

Este requisito obteve resultados totalmente positivos para os testes realizados, devido a função, que é mostrada na figura 24, onde a linha 5 tem o papel de substituir as palavras mais comumente usadas em instruções SQL por um caractere em branco. Além disso, na linha 8 elimina todos os comandos PHP existentes na variável.

4.4.1.6 – Teste do requisito suplementar: Autenticação do Cliente

Tabela 15 – Testes do requisito Autenticação do Cliente

Requisito	**Autenticação do Cliente**		
Descrição	Neste teste o requisito testado sofrerá tentativas de usuários não cadastrados, usuário com senhas incorretas e usuários utilizando ataque de injeção de SQL se autenticarem.		
Tipo de teste	Teste de unidade		
Abordagem	Teste de caixa preta		
Técnica	Teste realizado de forma manual		
Dados do Teste			
Pré-condições	O usuário deve estar na página de autenticação do cliente		
Pós-condições	Autenticação do cliente no caso de dados válidos ou rejeição em caso de dados inválidos.		
Passos			
1	Acessar o sistema		
2	Preencher o formulário com diferentes tipos de dados		
3	Submeter formulário		
Cenário	**Saída Esperada**	**Campo1(*login*)**	**Campo2(senha)**
1	Erro: Usuário não cadastrado	[a-zA-Z0-9]	[a-zA-Z0-9]
2	Erro: Usuário não cadastrado	[‘ or 1=1]	[a-zA-Z0-9]
3	Erro: Favor preencher todos os campos	[vazio]	[a-zA-Z0-9]
4	Erro: Favor preencher todos os campos	[a-zA-Z0-9]	[vazio]
5	Erro: Favor preencher todos os campos	[vazio]	[vazio]
6	Erro: Senha Incorreta	[válido]	[‘ or 1=1]
7	Erro: Senha Incorreta	[válido]	[a-zA-Z0-9]
8	Autenticação do Cliente	[válido]	[válido]

Resultados obtidos:

Para o requisito “Autentica Cliente” todos os testes realizados obtiveram as saídas esperadas, mesmo com ataques de injeção de SQL (tais como os descritos nos cenários 2 e 6) que representam um dos maiores perigos para sistemas *web*. Isso ocorreu devido à validação dos tipos de entradas que o

cliente digita no sistema e a elaboração/utilização de uma função que previne contra ataques de injeção de SQL; esta função está disponível no anexo A deste trabalho.

4.4.1.7 – Teste do requisito suplementar: Autoridade Certificadora

Tabela 16 – Testes do requisito Autoridade Certificadora

Requisito	**Autoridade Certificadora**
Descrição	Serão testadas as assinaturas digitais geradas nas trocas de mensagens que utilizam esse recurso. O teste consiste em assinar digitalmente algumas mensagens e depois conferir se estas mensagens foram ou não comprometidas da origem até seu destino.
Tipo de teste	Teste de unidade
Abordagem	Teste de caixa preta
Técnica	Manual
Dados do Teste	
Pré-condições	Estar de posse da mensagem que será assinada
Pós-condições	Mensagem assinada e enviada.
Passos	
1	Gerar a mensagem
2	Assinar mensagem
3	Enviar Mensagem via *e-mail*
4	Abrir mensagem assinada
5	Verificar assinatura

A autoridade certificadora tem o intuito de fornecer os certificados digitais usados nas assinaturas digitais de modo que comprovem a identidade da entidade que criou as mensagens. Para isso foi adquirido um certificado digital da empresa Thawte, que é uma divisão da empresa VeriSign e que não cobra pela emissão dos certificados digitais. Outra grande vantagem dos certificados emitidos por essa empresa é o fato dela constar entre as autoridades certificadoras reconhecidas internacionalmente.

Foram feitos testes utilizando o certificado digital acima citado e as mensagens resultantes apareceram adequadamente assinadas nos seguintes clientes de e-mail *Microsoft Outlook Express*, *Microsoft Outlook* 2003, *Mozilla Thunderbird* e *Kmail.*

As assinaturas digitais tiveram resultado totalmente positivos nos testes realizados, pois as funções utilizadas utilizam protocolos e algoritmos validados

por órgãos internacionais que tratam sobre o assunto, além do fato de ter sido usado um certificado digital válido e confiável.

4.4.1.6 – Teste do requisito suplementar: Autoridade Fiscalizadora

Tabela 17 – Testes do requisito Autenticação da Autoridade Fiscalizadora

Requisito	**Autenticação da Autoridade Fiscalizadora**		
Descrição	Neste teste o requisito testado sofrerá tentativas de usuários não cadastrados, usuário com senhas incorretas e usuários utilizando ataque de injeção de SQL se autenticarem.		
Tipo de teste	Teste de unidade		
Abordagem	Teste de caixa preta		
Técnica	Teste realizado de forma manual		
Dados do Teste			
Pré-condições	O usuário estar na página de autenticação do cliente		
Pós-condições	Autenticação do cliente no caso de dados válidos.		
Passos			
1	Acessar o sistema		
2	Preencher o formulário com diferentes tipos de dados		
3	Submeter formulário		
Cenário	**Saída Esperada**	**Campo1**	**Campo2**
1	Erro: Usuário não cadastrado	[a-zA-Z0-9]	[a-zA-Z0-9]
2	Erro: Usuário não cadastrado	[‘ or 1=1]	[a-zA-Z0-9]
3	Erro: Favor preencher todos os campos	[vazio]	[a-zA-Z0-9]
4	Erro: Favor preencher todos os campos	[a-zA-Z0-9]	[vazio]
5	Erro: Senha Incorreta	[válido]	[‘ or 1=1]
6	Erro: Senha Incorreta	[válido]	[a-zA-Z0-9]
7	Autenticação da Autoridade Fiscalizadora	[válido]	[válido]

A autenticação da AF funciona igual à validação do clientes perante o sistema, com isso os resultados também são iguais, o que os diferencia são as permissões de acesso que cada tipo de usuário possui no sistema.

4.4.1.8 – Teste do requisito suplementar: Níveis de Usuário

Tabela 18 – Testes do requisito Níveis de usuário

Requisito	**Níveis de usuário**
Descrição	Neste requisito vários usuários com permissões diferentes se autenticarão no sistema e tentarão acessar funcionalidades que não fazem parte seu escopo. Os níveis de acesso estão explicados na tabela 6.

Tipo de teste	Teste de unidade
Abordagem	Teste de caixa preta
Técnica	Teste realizado de forma manual
Dados do Teste	
Pré-condições	O usuário deve ter ser autenticado no sistema.
Pós-condições	
Passos	
1	Acessar o sistema
2	Preencher o formulário com dados para autenticação
3	Submeter formulário
4	Autenticar-se
5	Acessar a página principal do sistema
6	Tentar acessar funcionalidades não permitidas

Cenário	Saída Esperada	Tipo de usuário
1	Acesso negado ao módulo de “Atendimento”	Cliente
2	Acesso negado ao módulo de “Atendimento”	Cliente
3	Acesso total ao sistema	Empresa
4	Acesso apenas a consultas de OS	Autoridade Fiscalizadora
5	Acesso total ao sistema	Empresa
6	Acesso total ao sistema	Empresa
7	Acesso negado ao módulo de “Atendimento”	Cliente
8	Acesso apenas a consultas de OS	Autoridade Fiscalizadora
9	Acesso apenas a consultas de OS	Autoridade Fiscalizadora
10	Acesso apenas a consultas de OS	Autoridade Fiscalizadora
11	Acesso negado ao módulo de “Atendimento”	Cliente
12	Acesso negado ao módulo de “Atendimento”	Cliente
13	Acesso apenas a consultas de OS	Autoridade Fiscalizadora
14	Acesso total ao sistema	Empresa

Resultados obtidos:

O acesso restrito a usuários do tipo Cliente funcionaram corretamente restringindo o acesso aos módulos que esse tipo de usuário tem privilégios.

Para usuários do tipo Empresa não houveram problemas, pois estes têm acesso irrestrito ao sistema.

Inconformidade:

Para os usuários do tipo Autoridade Fiscalizadora foi encontrado uma falha na qual era disponibilizado o acesso irrestrito ao sistema.

Ações de correção:

O problema foi resolvido adicionando mais uma categoria de usuário junto aos seus dados no banco de dados, essa medida não afetou os outros tipos de usuário do sistema.

Após corrigido foi submetido novamente aos testes quando se obteve resultados totalmente positivos.

4.4.1.9 – Teste do requisito funcional: Abrir OS

Tabela 19 – Testes do requisito Abrir OS

Requisito	**Abrir OS**
Descrição	Neste requisito serão abertas várias OS de usuários diferentes para detectar se existe falha.
Tipo de teste	Teste de unidade
Abordagem	Teste de caixa preta
Técnica	Teste realizado de forma manual
Dados do Teste	
Pré-condições	O usuário deve ter ser autenticado no sistema.
Pós-condições	OS aberta e comprovante de aberta deve ser enviado
Passos	
1	Acessar o sistema
2	Preencher o formulário com dados para autenticação
3	Submeter formulário
4	Autenticar-se
5	Acessar a página principal do sistema
6	Acessar a página de abertura de OS
7	Preencher o formulário de abertura
8	Submeter formulário
9	Aguardar mensagem e comprovante de confirmação de abertura

Cenário	Saída Esperada	Campo1 (Nome do Cliente)	Campo2 (título da OS)	Campo3 (data de abertura da OS)	Campo4 (Tipo da OS)	Campo5 (Texto)
1	Erro: Favor preencher todos os campos	[automático]	[vazio]	[automático]	[vazio]	[vazio]
2	Erro: Favor preencher todos os campos	[automático]	[vazio]	[automático]	[válido]	[a-zA-Z0-9]
3	Erro: Favor preencher todos os campos	[automático]	[a-zA-Z0-9]	[automático]	[vazio]	[a-zA-Z0-9]
4	Erro: Favor preencher todos os campos	[automático]	[a-zA-Z0-9]	[automático]	[válido]	[vazio]
5	Erro: Preencher campo somente caracteres válidos	[automático]	[inválido]	[automático]	[inválido]	[inválido]
6	Erro: Preencher	[automático]	[válido]	[automático]	[inválido]	[válido]

	campo somente caracteres válidos					
7	Erro: Preencher campo somente caracteres válidos	[automático]	[válido]	[automático]	[inválido]	[inválido]
8	Comprovante de abertura da OS	[automático]	[válido]	[automático]	[válido]	[válido]
9	Erro: Preencher campo somente caracteres válidos	[automático]	[inválido]	[automático]	[válido]	[válido]
10	Comprovante de abertura da OS	[automático]	[válido]	[automático]	[válido]	[válido]
11	Comprovante de abertura da OS	[automático]	[válido]	[automático]	[válido]	[válido]
12	Comprovante de abertura da OS	[automático]	[válido]	[automático]	[válido]	[válido]
13	Erro: Preencher campo somente caracteres válidos	[automático]	[válido]	[automático]	[inválido]	[vazio]
14	Erro: Preencher campo somente caracteres válidos	[automático]	[válido]	[automático]	[vazio]	[inválido]
15	Comprovante de abertura da OS	[automático]	[válido]	[automático]	[válido]	[válido]
16	Erro: Preencher campo somente caracteres válidos	[automático]	[‘ or 1=1]	[automático]	[válido]	[válido]
17	Erro: Preencher campo somente caracteres válidos	[automático]	[‘ or 1=1]	[automático]	[‘ or 1=1]	[‘ or 1=1]
18	Erro: Preencher	[automático]	[inválido]	[automático]	[válido]	[‘ or 1=1]

	campo somente caracteres válidos					
19	Erro: Preencher campo somente caracteres válidos	[automático]	[inválido]	[automático]	[válido]	[válido]
20	Comprovante de abertura da OS	[automático]	[válido]	[automático]	[válido]	[válido]
21	Comprovante de abertura da OS	[automático]	[válido]	[automático]	[válido]	[válido]
22	Comprovante de abertura da OS	[automático]	[válido]	[automático]	[válido]	[válido]
23	Comprovante de abertura da OS	[automático]	[válido]	[automático]	[válido]	[válido]
24	Comprovante de abertura da OS	[automático]	[válido]	[automático]	[válido]	[válido]
25	Comprovante de abertura da OS	[automático]	[válido]	[automático]	[válido]	[válido]

Resultados obtidos:

Para os cenários em que houveram qualquer tipo de entradas inválidas o testes obtiveram resultados corretos.

Inconformidade:

Nos cenários em que foram utilizadas todas as entradas válidas a aplicação mostrou o erro informando que os campos não estavam preenchidos com dados corretos.

Ações de correção:

Os campos que são preenchidos automaticamente (nome do cliente e data da OS) pela aplicação estavam sendo submetidos com algumas “sujeiras” anexas. Para o campo data de abertura bastou limitar a quantidade de caracteres para seu tamanho e para o campo nome do cliente bastou ser descartado e recuperado seu código de identificação após a submissão do formulário.

Depois de corrigido foi submetido novamente aos testes quando se obteve resultados totalmente positivos.

4.4.1.10 – Teste do requisito funcional: Fechar OS

Tabela 20 – Testes do requisito Fechar OS

Requisito	**Fechar OS**
Descrição	Neste requisito serão abertas várias OS de usuários diferentes para detectar se existe falha.
Tipo de teste	Teste de unidade
Abordagem	Teste de caixa preta
Técnica	Teste realizado de forma manual
Dados do Teste	
Pré-condições	OS Aberta
Pós-condições	OS fechada e envio de comprovante de fechamento
Passos	
1	Acessar o sistema
2	Preencher o formulário com dados para autenticação
3	Submeter formulário
4	Autenticar-se
5	Acessar a página principal do sistema
6	Acessar a página de fechamento da OS
7	Preencher formulário com dados do procedimento de atendimento
8	Submeter formulário
9	Receber mensagem de fechamento e enviar comprovante de fechamento

Cenário	**Saída Esperada**	**Campo1 (Status OS)**	**Campo2 (descrição do procedimento)**
1	OS fechada	[automático]	[válido]
2	OS fechada	[automático]	[válido]
3	Erro: Favor preencher todos os campos	[vazio]	[a-zA-Z0-9]
4	Erro: Favor preencher todos os campos	[automático]	[vazio]
5	Erro: Preencher campo somente caracteres válidos	[válido]	[‘ or 1=1]
6	Erro: Senha Incorreta	[válido]	[a-zA-Z0-9]
7	OS fechada	[válido]	[válido]
8	Erro: Preencher campo somente caracteres válidos	[automático]	[inválido]
10	Erro: Favor preencher todos os campos	[automático]	[vazio]

Resultados obtidos:

Os testes desse requisito foram totalmente positivos. Por ser tratar de uma funcionalidade que somente usuários do tipo Empresa e com poucos campos que devem ser preenchidos tornou-se mais simples o controle dos dados de entrada.

4.4.1.11 – Teste do requisito funcional: Verificar Situação da OS

Tabela 21 – Testes do requisito Verificar situação da OS

Requisito	**Verificar situação da OS**
Descrição	Serão fechadas diferentes OS e enviados os comprovantes para que os respectivos clientes possam confirmar o seu fechamento.Será simulado também o fechamento dessas OS pela Empresa através do decurso de prazo.
Tipo de teste	Teste de unidade
Abordagem	Teste de caixa preta
Técnica	Teste realizado de forma manual
Dados do Teste	
Pré-condições	O usuário deve ter ser autenticado no sistema. Devem existir OS fechadas para confirmar seu fechamento
Pós-condições	
Passos	
1	Acessar o sistema
2	Preencher o formulário com dados para autenticação
3	Submeter formulário
4	Autenticar-se
5	Acessar a página principal do sistema
6	Acessar página de confirmação de fechamento
7	Selecionar a OS desejada
8	Confirmar o fechamento da OS

Resultados obtidos:

Para os usuários do tipo Cliente e Empresa os testes obtiveram resultados positivos. Os primeiros tipos usuários visualizaram apenas as OS abertas por si e o segundo tipo visualizaram todas as OS cadastradas.

Inconformidade:

Para os usuários do tipo Autoridade Fiscalizadora nenhuma OS foi apresentada.

Ações de correção:

Notou-se que o problema ocorria devido a comparação para listar todas as OS era realizada só para os usuários do tipo Empresa. Para corrigir isso bastou adicionar uma condição na qual o tipo Autoridade Fiscalizadora também teria privilégios para visualizar todas as OS.

Depois de corrigido foi submetido novamente aos testes quando se obteve resultados totalmente positivos.

4.4.1.12 – Teste do requisito funcional: Confirmar Fechamento da OS

Tabela 22 – Testes do requisito Confirmar Fechamento da OS

Requisito	**Confirmar Fechamento da OS**
Descrição	Neste requisito os usuários do tipo Cliente consultarão as suas OS. Os usuários do tipo Empresa e Autoridade Fiscalizadora irão consultar OS de vários usuários distintos.
Tipo de teste	Teste de unidade
Abordagem	Teste de caixa preta
Técnica	Teste realizado de forma manual
Dados do Teste	
Pré-condições	O usuário deve ter ser autenticado no sistema. Devem existir OS cadastradas
Pós-condições	
Passos	
1	Acessar o sistema
2	Preencher o formulário com dados para autenticação
3	Submeter formulário
4	Autenticar-se
5	Acessar a página principal do sistema
6	Acessar a página de consulta de OS
7	Selecionar a OS desejada
8	Verificar seu andamento

Resultados obtidos:

Para os usuários do tipo Cliente e Empresa os testes obtiveram resultados positivos, pois o usuário do tipo Cliente, que abriu a OS, recebe no comprovante de fechamento o link com número de identificação da OS e assim já o envia para s OS correta.

CONSIDERAÇÕES FINAIS

Com a finalidade de alcançar completamente os objetivos deste trabalho o TCC-I procurou fornecer o embasamento teórico necessário para que o desenvolvimento do TCC-II fosse possível. No TCC-I foram feitas pesquisas sobre atendimento ao cliente, mostrando seus conceitos, características e alguns sistemas semelhantes à proposta deste trabalho. Também foi pesquisada a segurança computacional com enfoque em criptografia e o protocolo desenvolvido por Ghisleri(2003) que era a base para o desenvolvimento do TCC-II.

A maior dificuldade no TCC-I foi encontrar fontes literárias confiáveis sobre os tipos de sistemas de atendimento disponíveis, visto que grande parte do material citava apenas os conceitos de atendimento, mas não sobre os sistemas em si.

Os sistemas de atendimento ao cliente têm o intuito de abrir um canal direto de comunicação e interação entre cliente e empresa. Porém, a grande maioria dos sistemas pesquisados para o desenvolvimento do presente trabalho não se preocupam com a segurança dos dados trocados em suas transações, com a veracidade da identidade dos clientes ou com os direitos do consumidor.

Com preocupações como estas, Ghisleri(2003) propôs o protocolo para um sistema seguro de atendimento ao cliente. Este protocolo é dividido em quatro fases: requisição, atendimento, fechamento da requisição e litígio. Em todas as fases do protocolo são gerados comprovantes eletrônicos assinados digitalmente garantindo, assim, confiabilidade no processo.

Os objetivos do TCC-II eram a elaboração de um projeto de uma aplicação com as características previstas no protocolo proposto, além da implementação e testes da mesma. As dificuldades encontradas no decorrer do desenvolvimento foram a definição de requisitos concisos e a elaboração dos cenários de testes.

O desenvolvimento deste trabalho colocou em prática conceitos vistos apenas teoricamente como, por exemplo, o uso de criptografia em transações

de dados e na assinatura de documentos, o uso de metodologias da engenharia de software usadas em todo o projeto da aplicação. Além de reforçar os conceitos citados, outros novos foram aprendidos como a questão do atendimento e respeito ao cliente e a aplicação de testes de software.

A sugestão para trabalhos futuros é a implementação das outras duas situações do protocolo: empresa não responde e cliente não responde. Estas situações são consideradas como exceções e não foram inseridas no escopo da presente pesquisa. Sendo assim, a agregação destas situações asseguraria a completa implementação do protocolo apresentado, cujos principais objetivos são a segurança dos dados e respeito aos direitos do consumidor.

REFERÊNCIAS

ADODB. **ADOdb database abstraction library for PHP (and python).** Disponível em < http:// http://adodb.sourceforge.net/#docs >. Acesso em: 30/08/2006.

ANDERSEN, K. KERR, C. **Costumer relationship management.** McGraw-Hill: Chigaco. 2003.

ARAÚJO, G. **Transações seguras via web.** Disponível em: < http://www.rnp.br/newsgen/9803/https.html#ng-certificacoes: >. Acesso em: 27/11/2006.

ARDIGO, J. D. 2004. **Modelo de infra-estrutura de chaves públicas como organização virtual para processos de avaliação somativa à distância.** Tese de Doutorado. Florianópolis. Universidade do Estado de Santa Catarina.

CAMPOS, E. **Information week**. Disponível em < http://www.informationweek.com.br/edicoes.php?ano=1999 >. Acesso em 30/08/2005.

CHAMPION, M. et.al **Web services architecture.** Disponível em: < http://www.w3.org/TR/2002/WD-ws-arch-20021114/ >. Acesso em 15/08/2006.

COBB, C. **Cryptography for dummies.** Wiley Publishing: Indianapolis. 2004.

CSLH. **Crafty syntax live help: open source live support system.** Disponível em < http://craftysyntax.com/ >. Acesso em: 11/11/2006.

CUNHA, D. **Web services, SOAP e aplicações Web.** Disponível em < http://devedge-temp.mozilla.org/viewsource/2002/soap-overview/index_pt_br.html >. Acesso em 15/08/2006.

DATASUS. **Tabela sugerida para utilização em testes de software**. Disponível em < http://w3.datasus.gov.br/datasus/ >. Acesso em 11/11/2006.

ELSENPETER, R. C., VELTE, T. J. **Iniciando em *e-Business*:** guia prático. Makron Books: São Paulo, 2002.

FELIPINI, D. **ABC do *e-Commerce*:** os quatro segredos de um negócio bem-sucedido na internet. Disponível em: < http://www.abc-commerce.com.br >. Acesso em 12/09/2005.

FLEURY, A. M. 2002. **Gerenciamento de segurança:** criptografias e seus conceitos**.** Material de Apoio às disciplinas de Pós-Graduação**.** Universidade Federal do Rio Grande do Sul.

FONTES, E. **Considerações sobre segurança da informação.** ITWeb. Disponível em: < http://www.itweb.com.br/noticias/artigo.asp?id=4929 > Acesso em: 21/06/2005.

GADELHA, J. F. 2002. **Uma avaliação do atendimento ao cliente na prestação de serviços com base na norma nbr iso 9004-2:** um estudo de caso. Dissertação de Mestrado. Florianópolis. Universidade Federal de Santa Catarina.

GHISLERI, A. S. 2002. **Sistema seguro de atendimento ao cliente:** garantia da qualidade de serviço. Dissertação de Mestrado. Florianópolis. Universidade Federal de Santa Catarina.

GHISLERI, L. R. G. 2003. **Proposta de um protocolo criptográfico para auditoria de publicidade na web.** Dissertação de Mestrado. Florianópolis. Universidade Federal de Santa Catarina.

GOOTS, N. et.al. **Modern cryptography:** protect your data wish fast block ciphers. A-List: Wayne. 2003.

HEINLEIN, P. OpenSSL: howto.. Disponível em: < http://www.madboa.com/geek/openssl/ >. Acesso em: 27/11/2006.

ISO - INTERNATIONAL ORGANIZATION FOR STANDARIZATION. **Norma ISO/IEC 17799.** Disponível em < www.iso.org >.

LABSEC. LabSEC: computer security lab. Disponível em: < http://www.labsec.ufsc.br/tiki-index.php >. Acesso em: 27/11/2006.

MACHADO, P. P. et.al. **Segurança da informação:** fundamentos do modelo de segurança da informação. Brasília: Ministério do Planejamento, 2000.

MATOS, A. V. d. **UML:** prático e descomplicado. 2 ed. Érica: São Paulo, 2002.

MYSQL. **Manual de referência do MySQL.** Disponível em: < http://dev.mysql.com/doc/refman/4.1/pt/index.html >. Acesso em: 20/03/2006.

MOLINARI, L. **Testes de software:** produzindo sistemas melhores e confiáveis. Erica: São Paulo. 2003.

MOREIRA, N. S. **Segurança mínima:** uma visão corporativa da segurança de informações. Axcel Books: Rio de Janeiro. 2002.

MUTO, C. A. **PHP & mysql:** guia introdutório. 2 ed. Brasport: Rio de Janeiro. 2004.

NEZZE, M. 2002. **Fidelização de cliente bancário:** o *marketing* de relacionamento como base na estratégia de lealdade. Dissertação de Mestrado. Florianópolis. Universidade Federal de Santa Catarina.

NICKELS, W. G.; WOOD, M. B. ***Marketing*** **:** relacionamentos-qualidade-valor. Rio de Janeiro: Livros Técnicos e Científicos S/A, 1999.

NOGUEIRA, J., M., S. **Protocolos de comunicação:** conceitos, serviços, especificação e teste. Belo Horizonte, UFMG-DCC, 1991.

OPENSSL, **Documentação do projeto OpenSSL.** Disponível em: < http://www.openssl.org/docs/app/openssl.html >. Acesso em: 15/09/2006.

PHP. **Manual do PHP.** Disponível em: < http://www.php.net/manual/pt_BR/index.php >. Acesso em: 20/03/2006.

PHP-OPENSSL. **Manual do PHP: OpenSSL.** Disponível em: < http://www.php.net/manual/pt_BR/ref.openssl.php >. Acesso em: 28/05/2006.

PINTAUD, M. de F. 2002. **A excelência no atendimento a clientes como diferencial competitivo.** Dissertação de Mestrado. Florianópolis. Universidade Federal do Estado de Santa Catarina.

POLPETA, F., V.; KOERICH, H. 2002. **Sistema seguro de atendimento ao cliente via web.** Trabalho de Conclusão de Curso. Universidade Federal do Estado de Santa Catarina.

PRESSMAN, R. S. **Engenharia de software.** 5ed. McGraw-Hill: Rio de Janeiro. 2002.

ROCHA, A., R., C., d. et. al. **Qualidade de software:** teoria e prática. Prentice Hall: São Paulo, 2001.

SACERDOTE, H. C. S. **Segurança da informação.** Disponível em: < http://pesquisa.cbs.unc.br/Seguranca%20da%20Informacao.pdf >. Acesso em 25/04/2005.

SANTOS, M. C. P. 2001. **Proposta e implementação de um modo seguro para http, com nível seletivo de segurança, sem alterações em servidores e navegadores.** Dissertação de Mestrado. Rio de Janeiro. Universidade Federal do Rio de Janeiro.

SCHEUER, L. 2001. **A qualidade do atendimento eletrônico em uma agência bancária segundo a percepção dos clientes.** Dissertação de Mestrado. Florianópolis. Universidade Federal do Estado de Santa Catarina.

SCHNEIER, B. **Segurança.com:** segredos e mentiras sobre a proteção na vida digital**.** Campus: Rio de Janeiro, 2001.

SEBRAE . **Atendimento a clientes / promoção de vendas**. São Paulo:2005.

SHERIF, M., H. **Protocols for secure eletronic commerce**. 2 ed. CRC Press: New Jersey, 2004.

SHIFLETT, C. **Essential PHP security.** O'Reilly Media: Nevada, 2005.

SHIFLETT, C. **PHP security guide.** Disponível em: < http://phpsec.org/projects/guide/ >. Acesso em: 27/11/2006.

STALLINGS, W. **Cryptography and network security.** Principles and Practice.2 ed. Prentice Hall: New York, 1999.

STALLINGS, W. **Cryptography and network security.** Principles and Practice.4 ed. Prentice Hall: New York, 2005.

SCHWEITZER, A. 2004. **Pressupostos para o gerenciamento de soluções CRM (*costumer relationship management*).** Tese de Doutorado. Florianópolis. Universidade Federal de Santa Catarina.

TERADA, R. **Segurança de dados:** criptografia em redes de computador. Edgard Blücher: São Paulo, 2000.

SOARES, W. **MySQL:** conceitos e aplicações. Érica: São Paulo. 2001.

WEBMEDIA. **Webmedia group**. Disponível em: < http://www.cerberusweb.com/index.php >. Acesso em: 11/11/2006.

W3C. **W3C SOAP1.1 documentation.** Disponível em: < http://www.w3.org/TR/2000/NOTE-SOAP-20000508/ >. Acesso em 15/08/2006.

WAZLAWICK, R. S. **Análise e projeto de sistemas de informação orientados a objetos.** Campus: Rio de Janeiro, 2004.

WIKIPÉDIA. **Modelo entidade e relacionamentos**. Disponível em: < http://pt.wikipedia.org/wiki/Modelo_de_Entidades_e_Relacionamentos >. Acesso em 15/04/2006.