UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

TRABALHO DE LICENCIATURA

TEMA

DESDOBRAMENTO DE SEGURANÇA EM SOA NA PROTOTIPAGEM DO SISTEMA DE MONITORIA E EMISSÃO DE DADOS ESTATÍSTICOS DE FUNCIONÁRIOS DA UEM NO QUADRO DE FORMAÇÃO

Autor: Jonnathan Papel Filipe Guambe

Supervisor: dr. José Nhavoto
Có-Supervisor: Prof. Doutor Emilio Mosse

Maputo, 2009

DECLARAÇÃO SOB PALAVRA DE HONRA

Eu **Jonnathan Papel Filipe Guambe**, declaro por minha honra que este trabalho é fruto de minha investigação e de conhecimentos por mim adquiridos no processo de aprendizagem ao longo da carreira estudantil e que o mesmo foi realizado apenas como Trabalho de Licenciatura em Informática na Universidade Eduardo Mondlane.

Maputo, Fevereiro de 2009

Jonnathan P. F. Guambe

(Jonnathan Papel Filipe Guambe)

DEDICATÓRIA

A meus pais e irmãos, dedico.

AGRADECIMENTOS

A concretização do presente trabalho contou com o apoio, disponibilidade e simpatia de muitas pessoas, os quais quero agora agradecer por tudo o que com elas aprendi. Neste sentido terei, necessariamente, de agradecer aos professores do Departamento de Matemática e Informática (DMI) que, ao longo destes quatro anos, me facultaram inúmeros ensinamentos teóricos e práticos, sem os quais não poderia estar a realizar esta tese de licenciatura.

Agradeço especialmente

- Aos meus pais Filipe Jossia e Leia Vita Jonatas Deve por me terem dado a vida, cuidados, educação e terem me ajudado a me tornar o que sou.
- Aos meus irmãos que estiveram sempre presentes nos momentos de alegria e tristeza a dar o máximo apoio.
- A minha namorada e companheira Amália Dalila Matsimbe que sempre esteve ao meu lado motivando-me cada vez mais.
- A meus colegas pioneiros (2005) do curso de informática pós-laboral, que sempre foram meus companheiros e amigos.
- Ao meu supervisor e có-supervisor dr. José Nhavoto e Prof. Dr. Emílio Mosse respectivamente, que contribuíram para que este trabalho se tornasse realidade.
- A meus primos, tios, avós, amigos, professores e outros colegas de faculdade por me terem dado apoio, contribuindo para o sucesso deste trabalho.
- A todos que directa ou indirectamente contribuíram para que o presente trabalho se realizasse com êxito.

⋆ E também a você que está lendo este trabalho.

"Há homens que lutam um dia e são bons.
Há outros que lutam um ano e são melhores.
Há os que lutam muitos anos e são muito bons.
Porém, há os que lutam toda a vida.
Esses são os imprescindíveis."

Bertolt Brecht.

RESUMO

Nos últimos dias, as transformações nas organizações, tecnologias e nos processos de negócios, alteram de certa forma a vida das pessoas. Nesse contexto a Universidade Eduardo Mondlane não é uma excepção. O desenvolvimento de aplicações baseadas em Arquitectura Orientada a Serviço (SOA) garante que haja interoperabilidade entre ambientes e plataformas heterógeneas, maximiza a partilha de funcionalidades entre as aplicações que consequentemente eleva o índice de reutilização. Este facto possibilita as organizações a reagirem as necessidades do mercado de uma forma mais ágil e eficiente.

Entretanto, junto a essas vantagens surgem os desafios de segurança existentes em sistemas baseados em SOA e Web Services, sendo a última a tecnologia mais comum e recomendada para implementação de uma SOA. Devido ao dinamismo da abordagem SOA e a extrema aproximação a realidade do ambiente de negócios é muito comum que existam serviços intermediários em vários processos, deste modo o mecanismo de segurança que melhor se adequa é segurança a nível da mensagem, que foi incorporada na presente tese usando o padrão de segurança WS-Security de modo a garantir autenticação, confidencialidade, integridade e não repúdio das mensagens. WS-Security foi implementado usando a tecnologia Mutual Certificates Security (MCS).

É neste âmbito que aparece o trabalho e espera-se que o trabalho venha a facilitar o processo de monitoria dos funcionários da UEM no quadro de formação e principalmente permitir que o processo de emissão de dados estatísticos seja feito de uma forma rápida, dinâmica e segura. O protótipo prevé também que os parceiros ou orgãos com certificados digitais válidos possam consumir os dados estatísticos relativos aos funcionários da UEM no quadro de formação directamente das suas aplicações, independentemente da plataforma ou linguagem usadas por estes e sem intervenção humana.

Palavras Chave: Arquitectura Orientada a Serviço, Segurança, Serviços Web, Certificado digital, Criptografia, Autenticação, Integração, Padrão WS-Security, Monitoria.

Conteúdo

Lista de Abreviaturas

DRH	Direcção de Recursos Humanos
UEM	Universidade Eduardo Mondlane
TIC	Tecnologias de Informação e Comunicação
CTA	Corpo Técnico Administrativo
CDI	Corpo Docente e Investigador
SOA	Service Oriented Architecture
MS-Excel	Microsoft Office Excel
XML	eXtensible Markup Language
REST	Representational State Transfer
SOAP	Simple Object Acess Protocol
UDDI	Universal Description, Discovery, and Integration
SAML	Security Assertion Markup Language
HTML	Hypertext Markup Language
DCOM	Distributed Component Object Model
CORBA	Common Object Request Broker Architecture
SSL	Secure Socket Layer
TLS	Transport Layer Security
HTTP(s)	Hypertext transfer Protocol (Secure)
DNS	Domain Name System
EAI	Enterprise Application Integrator
RPC	Remote Procedure Call
ICP	Infraestrutura de Chaves Públicas
PKCS	Public Key Cryptography Standards
UML	Unified Modeling Language
ESB	Enterprise Service Bus
WS-Security	Web Services security
SQL	Structured Query Language
DoS	Denial of Service
DDoS	Distributed Denial of Service
OWASP	Open Web Application Security Project

Lista de Figuras

Lista de Tabelas

CAPÍTULO

1

INTRODUÇÃO

Neste século novas transformações se apresentam, consolidam-se novas tecnologias e novas formas de actuar das organizações, alterando profundamente a vida das pessoas. Dentre as transformações que ocorrem na tecnologia, na informação, no conhecimento e nas fontes de recursos, estão as mudanças no cenário educacional, com principal enfoque nas instituições públicas de educação que possuem cursos superiores e de ensino médio profissional. As transformações sempre fizeram parte da sociedade como um todo, mas ultimamente o interessante é a velocidade com que elas ocorrem. A mudança de paradigmas de negócio, aumentou a necessidade de modernização dos processos produtivos, da abordagem de mercado e principalmente da forma de gestão de recursos materiais e humanos.

No contexto em que estas mudanças estão a acontecer, principalmente em função da globalização, as pessoas estão a sofrer alterações de direccionamento profissional. As empresas e instituições de ensino precisam preparar-se para estas mudanças, nas quais o cuidado com a *"reciclagem profissional"* e os programas estratégicos de desenvolvimento de recursos humanos são o diferencial competitivo.

No ambiente em que se encontram as instituições públicas de educação, elas precisam urgentemente de reconhecer as mudanças em curso e adaptar-se às novas condições sociais e tecnológicas, na velocidade em que elas se transformam, para não comprometer sua capacidade de sobrevivência e expansão.

Em instituições de ensino, a formação para reforçar a componente técnica, o nível de conhecimento organizacional e a qualidade de ensino nos cursos ministrados são alguns factores críticos de sucesso e de extrema relevância, surgindo assim a necessidade de mo-

nitoria e gestão do período pré, durante e pós formação dos funcionários dessa instituição. As opções estratégicas da instituição no que diz respeito as opções tecnológicas, ao seu perfil e âmbito de actuação e o posicionamento local, nacional, regional e até internacional é uma das tarefas que as instituições de ensino tentam fazer para aceitação e reconhecimento da qualidade de ensino e dos diplomas atribuídos por estas.

A Universidade Eduardo Mondlane (UEM) como uma instituição de ensino é também atingida por este tipo de visão e com o crescimento organizacional, acredita-se que o número de funcionários no quadro de formação tende a aumentar e, consequentemente, aumenta a complexidade de monitoria e gestão de recursos humanos que resulta em formações, especializações, reciclagens, entre outras actividades que possam contribuir para o incremento do nível de conhecimento ou qualidade de ensino desta.

Junto a este aumento de complexidade, vem acompanhada a necessidade de interoperabilidade entre a Direcção de Recursos Humanos (DRH) da UEM com alguns órgãos dentro e fora desta para troca de informação relativa aos funcionários no quadro de formação de forma eficiente e segura. Porém, as plataformas de implementação de sistemas de informação existentes nestes órgãos são relativamente distintas, sendo que, a diversidade existente entre elas, faz com que os sistemas sejam heterogéneos. Devido a tal heterogeneidade surge a necessidade de ter-se atenção em como garantir a integridade, autenticação, confidenciabilidade e interoperabilidade entre os diversos órgãos (com plataformas heterogéneas). Deste modo, torna-se indispensável a adopção de abordagens e tecnologias que irão satisfazer tais necessidades.

Para satisfação das tais necessidades, podem-se usar várias abordagens e tecnologias, desde abordagens através de base de dados, ficheiros, RPC, EAI, Arquitectura Orientada a Serviços (SOA), entre outros. Dentre essas abordagens e tecnologias, a abordagem SOA destaca-se como sendo uma das melhores opções para a definição de um padrão de interoperabilidade entre sistemas heterogéneos, visto que apresenta como enfoque principal o desenvolvimento de sistemas composto por serviços[1] reutilizáveis e que permitam fácil integração com outros sistemas. A tecnologia Web Services é maioritariamente recomendada e utilizada para implementação de uma SOA.

Visto que sistemas baseados em SOA podem representar vulnerabilidades para a organização devido a abertura de determinadas fronteiras para outros sistemas, torna-se indispensável a implementação do estudo baseado em segurança em SOA e os principais

[1] funcionalidade de determinado sistema, que é disponibilizado e acessível a partir de um outro sistema.

mecanismos de segurança implementáveis nela usando Web Services.

1.1 Definição do problema

A Direcção dos Recursos Humanos (DRH) possui um sistema informatizado para gestão de Recursos Humanos (RH) constituído por vários módulos. No concernente ao módulo de formação, o sistema não consegue satisfazer às necessidades críticas deste departamento relativamente aos funcionários no quadro de formação. Às necessidades que se podem considerar críticas à DRH, relativos aos funcionários no quadro de formação, tem-se a destacar a elaboração e submissão de relatórios a tempo e hora, com dados estatísticos realísticos e características específicas de acordo com os órgãos que os requerem, bem como a interoperabilidade entre as faculdades e a DRH para actualização de informações relativas aos funcionários no quadro de formação.

O sistema actual permite que se crie um plano de formação com dados incompletos relativos a um determinado funcionário no quadro de formação, dados que são considerados de extrema importância para elaboração de relatórios precisos e monitoria eficaz e eficiente dos funcionários no quadro de formação. Além disso, o sistema actual somente possui capacidade de filtragem a partir de um único campo, sendo que se houver necessidade de filtragem a partir de múltiplos campos é necessário filtrar cada campo e passar para uma folha de cálculos MS-Excel para efectuar a junção de centenas de instâncias dos dados filtrados para produção de um único conjunto de valores, que posteriormente são usados para elaboração de cálculos estatísticos.

Certos órgãos, de modo a actualizarem seus sistemas, têm solicitado por meio de cartas à DRH relatórios com dados estatísticos de funcionários no quadro de formação, especificando os dados que devem ser reflectidos no relatório e a data limite de resposta à solicitação.

No acto da resposta às referidas solicitações, a DRH tem excedido a data limite, definida primordialmente e tem emitido relatórios que não reflectem à total realidade dos funcionários no quadro de formação, devido aos factores descritos anteriormente, que resultam na redundância, inconsistência, ambiguidade e correcção dos relatórios elaborados.

A redundância e inconsistência ocorrem quando a mesma informação aparecer por vezes

duplicada devido a forma de criação dos relatórios. Tomemos como exemplo um relatório produzido em Junho de 2008, em que um funcionário tinha duas categorias e fazia no mesmo período dois cursos na mesma instituição de formação. Esta redundância é uma fonte potencial para inconsistência da informação. A inconsistência pode verificar-se, por exemplo, se for prolongada a data prévia do término da formação. Essa alteração só se irá reflectir num dos cursos que o funcionário frequenta.

No concernente a ambiguidade, podemos tomar como exemplo, funcionários em que o curso a obter após a formação é simplesmente "Gestão", ficando-se sem saber a que gestão se refere.

Acerca da correcção dos relatórios podemos tomar como exemplo um registo de instituição de formação diferente da que o funcionário se está a formar, bem como a omissão da data de início e data prévia de término.

A interoperabilidade e sincronização entre as faculdades e a DRH carecem também de atenção, visto que, há funcionários que terminam o período de formação e retornam aos seus postos de trabalho nas faculdades, ou permanecem mais tempo do que o previsto fora das suas actividades laborais, e mesmo assim continuam a receber o mesmo tratamento em relação ao salário por um determinado período, e entretanto, as faculdades não reportam estes factos à DRH. Deste modo, a fraca monitoria dos funcionários no quadro de formação pode trazer consequências transversais, desde a escassez de docentes, implicações financeiras, bem como a falta de Corpo Técnico Administrativo (CTA) para execução de determinadas actividades cruciais para o funcionamento eficiente da UEM.

1.2 Objectivos

1.2.1 Objectivos Gerais

- Aplicar a abordagem de SOA com enfoque a Segurança para prototipagem do sistema de monitoria de funcionários da UEM no quadro de formação e partilha de dados estatísticos destes.

1.2.2 Objectivos Específicos

- Analisar o processo de monitoria dos funcionários da UEM no quadro de formação e identificar os inconvenientes existentes no sistema actual;
- Avaliar a aplicabilidade da Arquitectura Orientada a Serviços e os mecanismos de segurança implementáveis;
- Analisar e seleccionar as tecnologias de implementação da Arquitectura Orientada a Serviços;
- Analisar e seleccionar os mecanismos de segurança implementáveis numa Arquitectura Orientada a Serviços;
- Conceber e testar o protótipo.

1.3 Metodologia

Para a pesquisa, adoptou-se uma abordagem de estudo de caso, e esta escolha se deu em função da necessidade de conhecer, de forma mais detalhada, o processo de gestão da informação. De acordo com PAULA (2004), um estudo de caso é um "estudo profundo e exaustivo de um ou poucos objectos, de maneira a permitir conhecimento amplo e detalhado sobre o mesmo". PAULA (2004) complementa dizendo que tais estudos correspondem a pesquisas que colectam e registram dados de um caso particular ou de vários casos a fim de organizar um relatório ordenado e crítico de uma experiência. Para PAULA (2004), um caso pode ser uma organização, pessoa, processos ou um projecto específico. Tomarei como estudo do caso o departamento de recursos humanos da UEM.

Para análise do processo de monitoria dos funcionários no quadro de formação e identificação dos inconvenientes existentes no sistema actual, recorreu-se a:

- Análise do sistema actual com base na documentação disponível na DRH;
- Observações não participativas e análises de documentos;
- Entrevistas ao pessoal da DRH da UEM com objectivo de obter informação que contribuisse para a construção de um protótipo mais próximo do que realmente se precisa.

Para avaliação da aplicabilidade, dos mecanismos de segurança, análise e selecção das tecnologias de implementação e os mecanismos de segurança implementáveis numa Arquitectura Orientada a Serviços, recorreu-se a:

- Revisão bibliográfica através de visita a páginas de internet, leitura de literatura relacionada a pesquisa, de forma a avaliar se as ferramentas, tecnologias e técnicas escolhidas para concepção do protótipo respondem num todo as necessidades tecnológicas da DRH em relação a monitoria do pessoal no quadro de formação e a interoperabilidade segura entre os orgãos;

Para conceber e testar o protótipo do sistema proposto, recorreu-se a:

- Especificação de linguagens de programação, plataformas e ambientes heterogéneos de modo a testar a interoperabilidade. As linguagens de programação utilizadas foram Java e Visual Basic 2008, as plataformas de desenvolvimento foram Netbeans 6.5 e Visual Studio 2008, foi utilizado também o servidor de aplicações Glassfish V2 para alojar o serviço Web.

1.4 Organização do Trabalho

No capítulo 1, são abordados os conceitos introdutórios de modo a contextualizar o leitor sobre o tema do presente trabalho, no capítulo 2 é feita a revisão bibliográfica com principal enfoque na segurança e implementação de uma arquitectura orientada a serviços, no capítulo 3 são apresentadas as considerações finais que se fizeram sentir ao longo da elaboração do trabalho e finalmente, no capítulo 4 são apresentadas as referências bibliográficas usadas no presente trabalho.

CAPÍTULO

2

REVISÃO BIBLIOGRÁFICA

Este capítulo tem como finalidade fazer a revisão dos conceitos que servirão de base para o acompanhamento da presente trabalho.

2.1 Monitoria

O objectivo do presente sub-capítulo é a elucidação do conceito "monitoria".

No geral, pessoas, organizações entre outros, tem objectivos a alcançar. De modo a alcançar tais objectivos é importante que se elabore um plano com detalhes (datas, actividades, etc.) de como os objectivos serão alcançados. E para sabermos o nível de realização dos objectivos é necessário que se verifique ou analise o grau de realização do plano seguindo uma periodicidade estratégica, sendo exactamente neste acto que esta sendo feita a monitoria, ou seja, com base num plano esta se monitorando (análises em períodos estratégicos) até que nível os objectivos desejados estão sendo realizados.

2.1.1 Definição

Segundo a INDEPENDENT EVALUTION GROUP[1], monitoria pode ser definido como: *"uma função contínua que usa uma colecção sistemática ou estratégica de dados baseados em certos indicadores de modo a fornecer à gerência e aos principais investidores de uma*

[1]Uma unidade independente dentro do Grupo do Banco Mundial

visão de desenvolvimento crescente com indicações da extensão de progresso e realização dos objectivos pré-estabelecidos com base nos fundos alocados".

No contexto de formação de funcionários da UEM, a monitoria pode ser considerada como um instrumento para assegurar a interacção entre o planeamento e a execução, possibilitando a correcção de desvios e a realimentação permanente de todo o processo de planeamento. No acto de monitoria, além de efectuar-se um simples acompanhamento, efectua-se também uma documentação sistemática do processo de implementação de um plano, bem como a avaliação dos desvios na execução das actividades propostas, antecipando e prognosticando as possibilidades de alcance dos objectivos e recomendando acções correctivas para ajuste ou replaneamento.

2.1.2 Objectivos

O acto de monitoria tem como objectivos os seguintes:

- Medir o desempenho actual em relação as bases definidas no plano primordial;
- Verificar, através dos indicadores e metas estabelecidos para cada artefacto, o desempenho obtido, as causas do mesmo e as medidas correctivas tomadas, se necessário;
- Identificar os desvios ocorridos entre o planeado e o realizado.

2.2 Recursos Humanos

Este sub-capítulo aborda alguns conceitos relacionados a recursos humanos com ênfase na formação destes.

A actividade de manutenção e formação de recursos humanos, tem evoluído muito nos últimos anos. Esta evolução deveu-se muito a necessidade de garantir uma competitividade que permitisse a sustentação organizacional em um mercado cada vez mais competitivo.

O factor humano tornou-se um diferencial decisivo para o sucesso das organizações, a partir do momento em que a interacção entre a empresa e o ambiente externo deixou de ser uma relação meramente comercial, revelando a sua natureza sistemática. É certo

que esta constatação não foi abrupta nem casual, pois a convivência da organização com o seu meio foi revelando a inter-relação de forças que há entre o ambiente externo e o interno, mas algumas empresas aprenderam com mais facilidade e rapidez a natureza e a dinâmica das leis naturais, sociais, políticas e económicas que regulam essa transacção (CABREIRA, 2008).

2.2.1 Formação

A formação consiste num processo contínuo e permanente de desenvolvimento e capacitação pessoal e profissional dos trabalhadores de uma organização, actuando não só como factor de qualificação profissional, na medida em que proporciona a aquisição de competências estratégicas, técnicas e relacionais, mas também como agente de inovação organizacional, porque estimula capacidades de liderança, de iniciativa, de participação e de criatividade e permite envolver todos, responsabilizando-os na consecução dos objectivos da instituição.

Visto que cada instituição possui seus planos e critérios para formação de pessoal, abaixo são mencionados alguns objectivos que podem ser relativamente transversais no que diz respeito a formação, a saber:

- Promover continuamente a aprendizagem;
- Melhorar os desempenhos;
- Dinamizar o trabalho;
- Criar novas competências, adequadas às perspectivas de evolução da organização;
- Conciliar as expectativas pessoais e profissionais dos trabalhadores com os objectivos integrados na missão organizacional;
- Predispor/sensibilizar todos os profissionais para processos de mudança;
- Reforçar a cultura organizacional;
- Melhorar a qualidade dos serviços prestados;
- Desenvolver novos serviços;
- Ganhar vantagens competitivas.

A prossecução desses objectivos, irá depender das estratégias adoptadas no processo formativo de cada instituição.

2.3 Arquitectura Orientada a Serviços

Este sub-capítulo retrata de alguns aspectos conceptuais abordados ao longo do trabalho. Em algumas partes, elucidarei alguns pontos através de exemplos.

"A necessidade da integração de dados em ambientes empresariais, apesar de antiga, ainda é um problema crucial a ser resolvido para a maioria das empresas, permitindo integração entre clientes, parceiros e fornecedores." (DEGAN, 2005).

Como um dos assuntos com maior popularidade actualmente no mercado de software, a Arquitectura Orientada a Serviços vem sendo considerada como um marco na evolução de software (ROCHA, 2007). Essa popularidade foi alcançada devido a abordagem de SOA, que enfatiza o desenvolvimento de sistemas que permitam que organizações ou empresas acompanhem facilmente as inevitáveis mudanças necessárias para suporte aos seus negócios e tecnologias.

As previsões acerca de SOA por entidades conceituadas como a GARTNER GROUP são positivas, pois, segundo uma reportagem da GARTNER GROUP que fez analise de 5 tópicos mais discutidos no ano de 2005, apontava que em 2008 80% dos novos projectos de desenvolvimento de Software seriam baseados na Arquitectura Orientada a Serviço (ROCHA, 2007).

De modo a clarificar o descrito acima, suponhamos que se deseja atender às demandas do mercado e das áreas de negócios com maior flexibilidade e agilidade. Tendo como cenário um ambiente baseado em uma arquitectura de software modular, onde todas aplicações são acedidas por uma única interface web e os sistemas utilizam dados uns dos outros e comunicam-se indiscriminadamente. Isto retrata nada mais, nada menos que uma Arquitectura Orientada a Serviços. Com base neste novo modelo arquitectural, já não há necessidade de implementação de uma interface, uma base de dados e um sistema de integração para cada aplicação.

No cenário ilustrado na Figura 2.1, despende-se muito tempo e dinheiro para reutilização de funcionalidades entre aplicações, limitando a capacidade de combinação de diferentes aplicações para criação de uma nova aplicação necessária de modo a atender necessidades de negócios.

A figura 2.1 reflecte uma visão centrada na aplicação, ou seja, reflete como a integração ou reutilização é comummente ou tradicionalmente feita entre aplicações empresariais, sendo esta feita na maioria das vezes de uma forma *ad hoc*[2] através de bases de dados, remote procedure call (RPC), ficheiros, entre outros.

[2]processo em que nenhuma técnica reconhecida é empregada e/ou cujas fases variam em cada aplicação do processo. Algo feito ad hoc ocorre ou é feito somente quando a situação assim o exige, ou o torna desejável, nunca é planeado ou preparado antecipadamente.

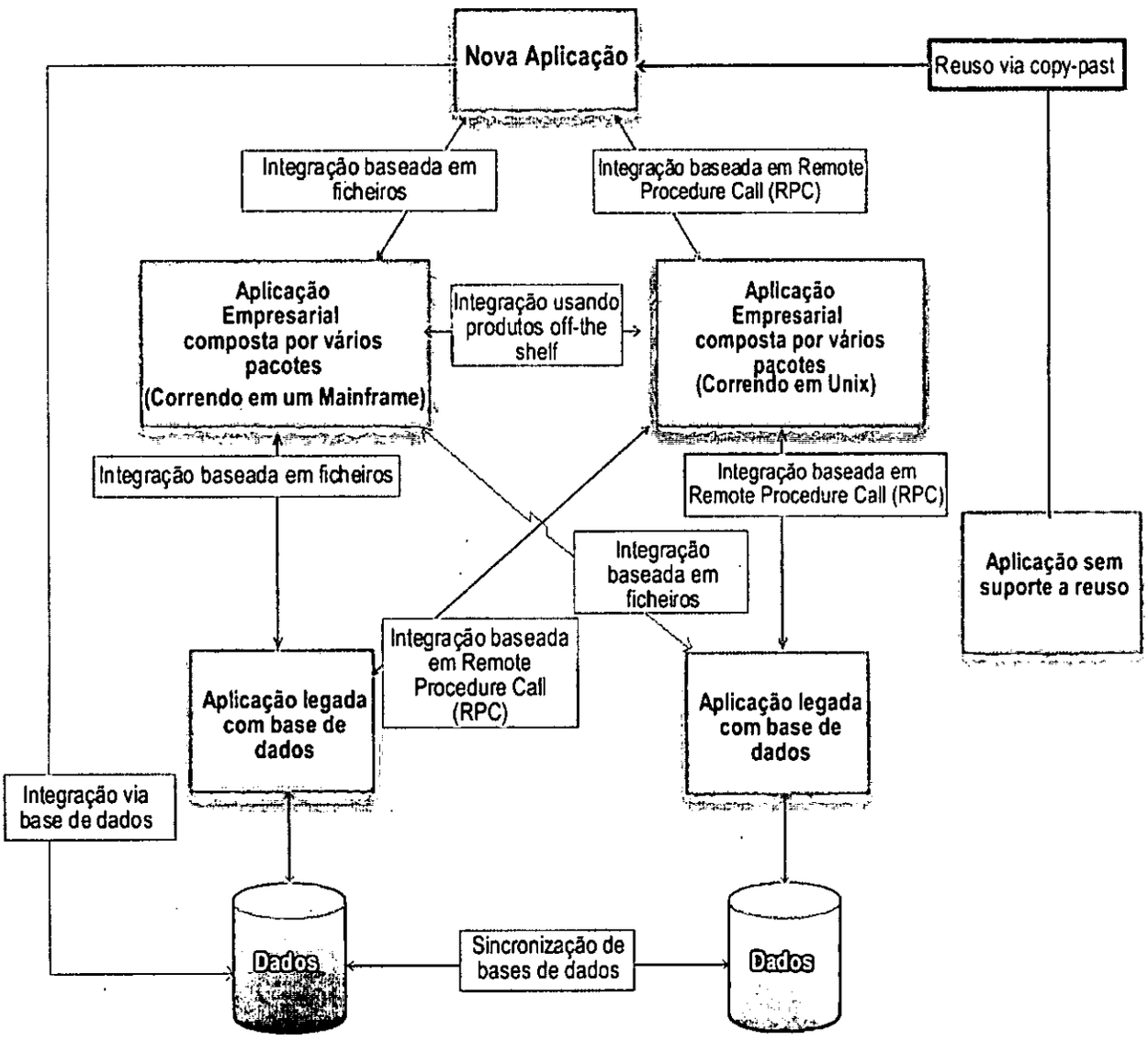


Figura. 2.1: Interoperabilidade em um ambiente empresarial usando aplicações tradicionais (adaptado de KANNEGANT, 2008).

A abordagem SOA resolve esses inconvenientes olhando para sistemas de TIC (Tecnologias de Informação e Comunicação) como colecções de unidades chamadas serviços, não como colecções de aplicações. Contrariamente a visão centrada na aplicação, onde a interoperabilidade entre aplicações é uma ideia que surge depois, em SOA, aplicações são concebidas com uma ideia de como expor serviços e que serviços serão expostos. Isso permite que sejam construídas novas aplicações a partir da combinação de serviços providos

por outras aplicações, a figura 2.2 ilustra este cenário.

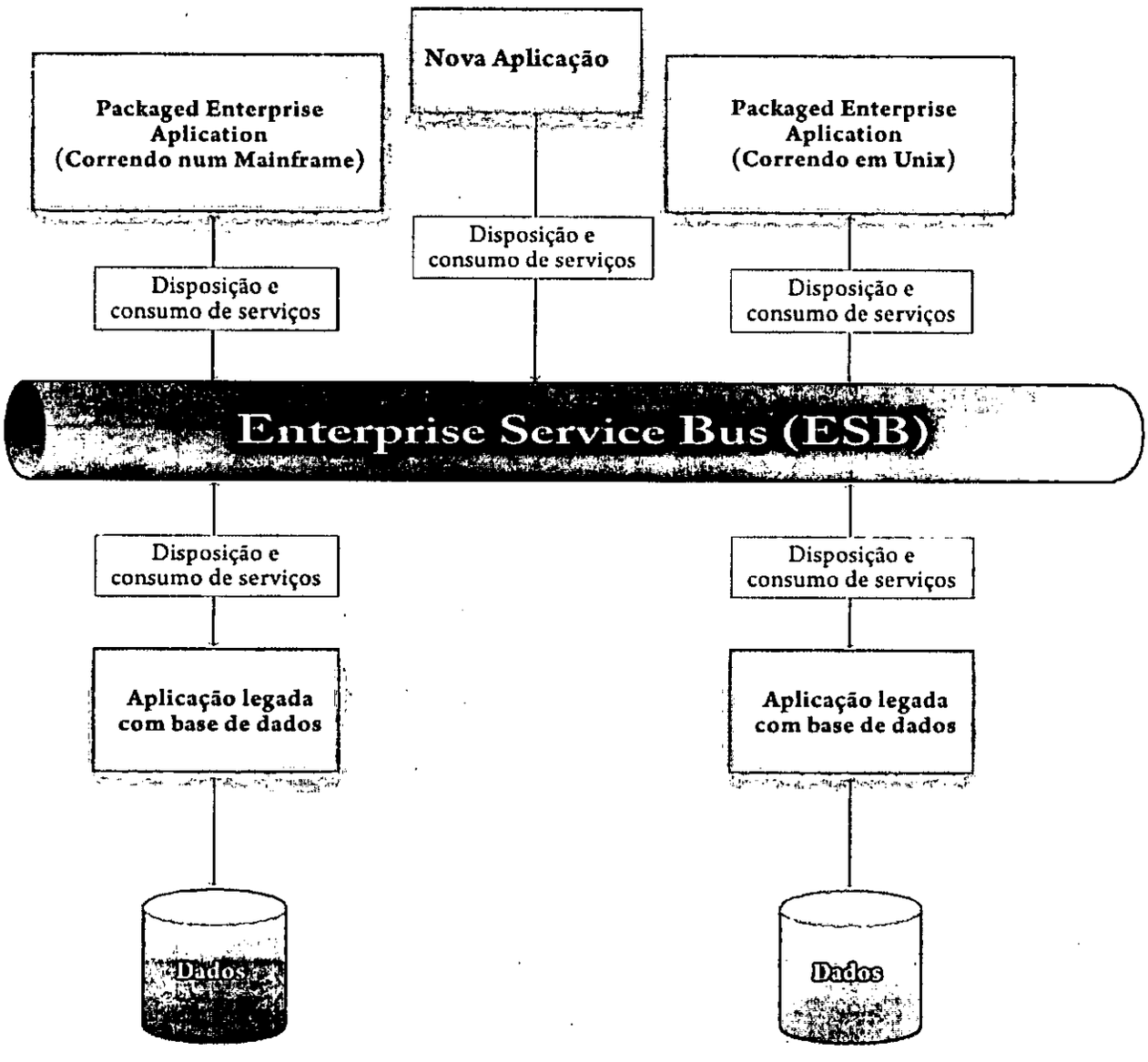


Figura. 2.2: Estado de implementações SOA em empreendimentos actuais (adaptado de KANNEGANT, 2008).

Em vez de mecanismos de reutilização *ad hoc*, em SOA, aplicações dispõem serviços para outras aplicações. As aplicações podem ser consumidoras, disponibilizadoras ou

ambos. Os Serviços são unificados e geridos por um ESB (Enterprise Service Bus) (KANNEGANT, 2008). Deste modo, torna-se evidente que aplicações com implementações baseadas em SOA, podem ser rapidamente desenvolvidas através do uso de funcionalidades disponibilizadas por várias aplicações, flexibilizando necessidades de mudanças que ocorrem no mercado.

2.3.1 Definição e principais termos

Este sub-capítulo tem como objectivo a definição de SOA, abordagem de alguns termos mais usados no "mundo" SOA.

De modo a enquadrar-mo-nos em discursos, artigos, workshops, relacionados com SOA, é importante conhecer os termos mais comuns acerca desse assunto. Começamos por definir Arquitectura Orientada a Serviços como sendo uma composição de um conjunto de conceitos e regras que proporciona a base para arquitectar, desenvolver sistemas e aplicações orientadas a serviços visando obter o máximo de desacoplamento[3] (loose coupling) entre serviços (ROCHA, 2007). E ROCHA (2007) realça também que muitos desses conceitos e regras existentes em SOA, foram baseados em modelos já existentes como, por exemplo, processamentos distribuídos, orientação a objectos entre outros.

Para KOBIELUS (2004), Arquitectura Orientada a Serviço é um modelo de desenho para maximização de partilha de serviços, reusabilidade e interoperabilidade em ambientes distribuídos.

[3]minimização do conjunto de factores que determinado sistema deve levar em consideração de modo a consumir funcionalidades ou serviços fornecidos por um outro sistema.

Termo	Definição/Comentário
Serviço	É uma função independente, sem estado (stateless) que aceita uma ou mais requisições e devolve uma ou mais respostas através de uma interface padronizada e bem definida. Serviços podem também realizar partes discretas de um processo tal como editar ou processar uma transacção. Serviços não devem depender do estado de outras funções ou processos. A tecnologia utilizada para prover o serviço, tal como uma linguagem de programação, não pode fazer parte da definição do serviço.
Orquestração	É processo de sequenciação ou composição de serviços de modo a possibilitar a criação de um novo serviço ou resolver uma tarefa de um processo de negócio. Para tal, há sempre uma figura de um ponto central, podendo ser um serviço ou uma actividade de negócio que coordena a chamada de outros serviços para compor uma função de maior granularidade[4]. A orquestração de serviços é análoga a um método da orientação a objectos que faz chamadas de outros métodos.
Stateless	Não depende de nenhuma condição pré-existente. Os serviços não devem depender de condições de outros serviços. Eles recebem todas as informações necessárias para prover uma resposta consistente. O objectivo de buscar a característica de stateless dos serviços é possibilitar que o consumidor do serviço possa sequenciá-lo, ou seja, orquestrá-los em vários fluxos (algumas vezes chamados de pipelines) para executar a lógica de uma aplicação.
Contracto	Especificação da forma como um consumidor de serviço irá interagir com o provedor do serviço. Um contracto de serviço pode requerer um conjunto de pré-condições e pós-condições, que servem para especificar o estado em que um serviço precisa estar de modo a executar uma função especifica.
Provedor	O recurso que executa o serviço em resposta a uma requisição de um consumidor.
Cliente	É quem consome ou efectua o pedido do resultado de um serviço fornecido por um provedor.
Binding	Define como dois programas podem conectar-se ou relacionar-se entre si. Em SOA, a relação entre os serviços do provedor e do consumidor deve ser dinâmica, sendo que ela é estabelecida em tempo de execução através de um mecanismo de binding.

Tabela 2.1: Termos mais comuns em SOA

Deste modo, pode se definir SOA como sendo uma abordagem que preconiza que funcionalidades implementadas em aplicações sejam desacopladas, interoperáveis e reusáveis de modo a agilizar os processos de negócios empresariais.

2.3.2 Princípios básicos de SOA

Como visto anteriormente, a abordagem SOA tem como conceito nuclear o *serviço*. Contudo, segundo BIACHI (2007), as ideias relacionadas a um serviço são:

- A competência de executar o trabalho para outro;
- A especificação do trabalho oferecido para outro;
- A oferta para executar um trabalho para outro.

No contexto de SOA, serviço é uma função independente, sem estado (stateless), que aceita uma ou mais requisições e retorna uma ou mais respostas através de uma interface padronizada e bem definida (BIACHI, 2007). Segundo KANNEGANT (2008), temos os seguintes princípios relacionados a SOA que são usadas para resolver questões relacionadas com a visão centrada em aplicações em TIC:

- Aplicações tem que abrir capacidades para seu uso por outras aplicações existentes ou novas. Tem que ser possível combinar os serviços oferecidos por diferentes aplicações de modo a criar serviços de alto nível ou aplicações compostas (composit aplications).
- A Diferença de tecnologias não deve ser importante e a interoperabilidade deve ser a meta principal.
- Devem ser adoptados padrões abertos para possibilitar integração entre empresas. Deve ser possível efectuar orquestração de processos de negócios entre múltiplos fornecedores, parceiros, e clientes.
- Deve se prestar atenção para governança e exequibilidade de modo a garantir que a flexibilidade provida pelos 3 primeiros princípios não se tornem um caos.

A SOA é composta por várias funcionalidades conceptuais e fundamentais, que quando implementadas num mesmo ambiente resultam no paradigma "find-bind-execute".

A Figura 2.3 ilustra a SOA e suas entidades básicas e conceptuais, realçando que cada entidade da arquitectura pode assumir um ou mais papeis, entretanto não é obrigatório que estas estejam relacionadas exactamente como ilustrado na Figura 2.3 ou que se encontrem todas entidades ao mesmo tempo numa arquitectura SOA. Por exemplo, a entidade de Registo de Serviços, onde são armazenados a localização dos serviços e dos

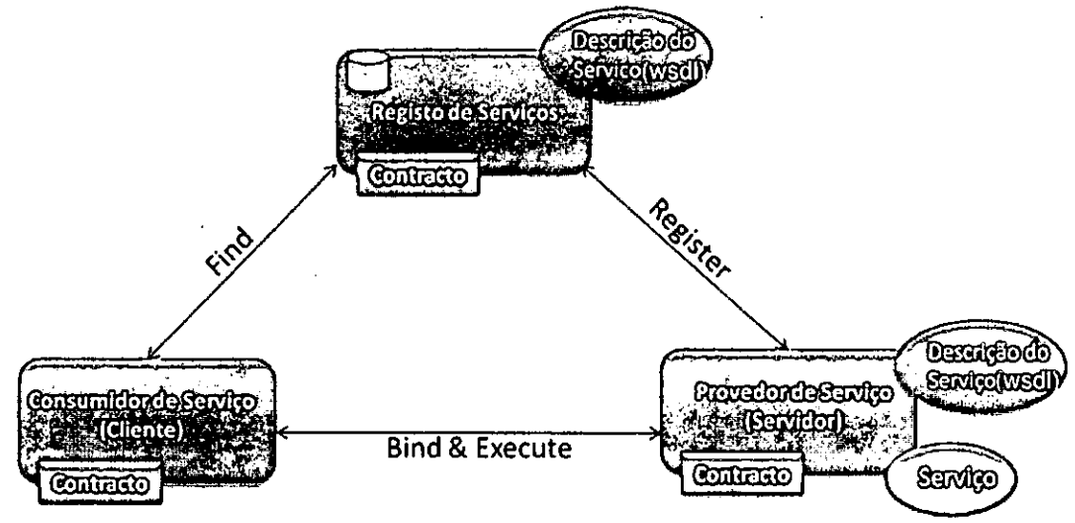


Figura. 2.3: Paradigma "find-bind-execute".

respectivos contractos, comummente não é usado na arquitectura SOA como ilustrado na figura.

2.3.3 Benefícios de SOA

No presente sub-capítulo, serão discutidos alguns aspectos relacionados com os benefícios que advém da adopção de Arquitectura Orientada a Serviço.

Existem inúmeros vantagens que se podem alcançar ao implementar uma SOA. Segundo BIANCHI (2007), os seguintes benefícios são alcançados:

- **Permite a reutilização dos activos de TIC:** Este é o primeiro e o mais importante benefício de implementar SOA. Pode-se construir um serviço de negócio como uma agregação de componentes existentes. Para utilizar este novo serviço é exigido somente saber seu nome e interface. A implementação de um serviço específico, sua arquitectura, bem como as complexidades do fluxo de dados que compõe este serviço, são transparentes para quem o chama. Isto permite às organizações aproveitar investimentos actuais, construir serviços de um conglomerado de componentes construídos em diferentes plataformas, que rodam em diferentes sistemas operacionais e desenvolvidos em diferentes linguagens de programação. Sistemas legados podem

ser encapsulados e acessados utilizando as interfaces Web Services.

- **Diminui o tempo de desenvolvimento e reduz os custos de desenvolvimento e manutenção:** como as demandas do negócio crescem e novos requisitos são introduzidos a todo momento, o custo de desenvolver e criar novos serviços através do SOA é muito reduzido. A curva de aprendizagem de uma equipe de desenvolvimento é também reduzida, pois já podem estar familiarizados com os componentes existentes.

- **Diminuição do risco:** reusar componentes existentes reduzem o risco de introduzir novas funcionalidades dentro de processos de melhoria ou criar novos serviços do negócio. O risco de manutenção e gestão da infra-estrutura dos serviços de suporte também será reduzida.

- **Protecção do investimento:** protege o investimento do cliente na gestão da informação que é altamente volátil num mercado onde as fusões e aquisições são frequentes.

2.3.4 Desafios de SOA

"Actualmente Organizações e pessoas enfatizam: reutilização, custo x beneficio e flexibilidade como pontos cruciais abordados por empreendedores de negócios."

SOA pode resolver alguns dos mais significantes problemas históricos em TIC, que é a necessidade de envolvimento de negócio na solução tecnológica, sendo que para tal é necessário um forte comprometimento na evolução organizacional, na consciencialização da necessidade de evoluir e mudar, no estabelecimento de uma gestão de recursos humanos eficaz, orientada a conhecer a potencialidade, objectivos e desejo das pessoas em detrimento dos objectivos e metas da organização, orientando a gestão dos serviços de acordo com o desempenho individual. Tornar os desafios da organização aderentes as pessoas é um dos maiores desafios da Gestão de mudança organizacional. Deste modo, SOA possibilita a criação de novas aplicações com custos e tempo de desenvolvimento reduzidos e com excelente aproveitamento dos sistemas legados.

É importante que estes inconvenientes sejam levados em consideração logo a prior, de modo que se criem condições para que sejam superados ao longo do tempo e eficiente-

mente.

2.3.5 SOA comparada com a Internet

A Arquitectura Orientada a Serviços e basicamente constituída pela troca de mensagens entre serviços através da utilização de protocolos como XML, REST, SOAP, UDDI, SAML, entre outros. Para melhor entendimento de SOA será feita uma comparação entre a analogia desta e da rede mundial de internet segundo a visão de ROCHA (2007):

- Na Internet temos os Web Servers (provedores de páginas html e serviços) e os Browsers/usuários (consumidores de páginas html e serviços). Os Web Servers (provedores de páginas html e de serviços) e Web Browsers/usuários (consumidores de páginas html e de serviços) não estão necessariamente conectados entre si todo momento. Os serviços providos pelos Web Servers também não são previamente conhecidos ou dependentes de plataforma hardware ou software para estabelecer uma conexão. Sendo assim, podemos notar que na rede mundial de Internet encontramos parte da característica básica e fundamental da arquitectura SOA, o acoplamento fraco de serviços (Loose Coupling) que na prática significa que qualquer modificação nas páginas e serviços oferecidos não afectará o acesso.

- A troca de informação na rede mundial Internet é feita basicamente através dos protocolos HTTP e páginas HTML. A troca de informação na arquitectura SOA é feita através das mensagens XML/SOAP sobre o protocolo HTTP. Para garantir a segurança dos dados trafegados entre os serviços na rede mundial Internet, são utilizados os protocolos SSL/TLS que garantem segurança do transporte, e dependendo do requisito de segurança exigido na troca dessas informações, podem-se utilizar certificados digitais para assinar digitalmente ou cifrar os dados trafegados. Entretanto, na SOA existem outros mecanismos de segurança como por exemplo, XML digital signature e XML encryption, que garantem a segurança da mensagem.

2.3.6 SOA vs Web Services

Este sub-capítulo visa clarificar alguns aspectos relacionados a ambiguidades que se fazem em relação a Arquitectura Orientada a Serviços e Web Services.

Com a difusão de SOA, vários programadores ou mesmo empresas de desenvolvimento de software pensam que ao fazer Web Services estão fazendo SOA. WILNER (2006) ostenta que ao fazer Web Services há quem pense que está fazendo SOA – e para fazer SOA se deve fazer Web Services, afirmando deste modo que esta ambiguidade realmente existe. Segundo WILNER (2006), Web Services é uma tecnologia para troca de informações, de conexão entre aplicações com foco em expor funcionalidades como tecnologia de acesso, enquanto que SOA tem como grande valor a reutilização, e o foco nas funcionalidades de negócio.

Pode se implementar uma Arquitectura Orientada a Serviços usando uma ou varias tecnologias ou formas de comunicação entre aplicações, tendo algumas como destaque: Web Services baseadas em SOAP ou REST, DCOM, RPC, CORBA, etc.. A chave para implementação de sistemas baseados em SOA é a independência de serviços com interfaces definidas de modo que possam ser acedidas para efectuar suas tarefas de uma forma padronizada, sem que o serviço provedor preveja a aplicação consumidora, e sem que a aplicação consumidora precise ou tenha que saber como é que o serviço provedor efectua sua tarefas.

Nota importante:
Web Services não é requisito para se obter, implementar ou estabelecer uma arquitectura SOA, porém a maioria de provedores de serviços preferem implementar seus serviços como serviços Web.

Visto que para implementação da Arquitectura Orientada a Serviços no presente trabalho será utilizada a tecnologia Web Services, na sub secção a seguir será feita uma breve visão das tecnologias usadas em Web Services.

2.3.7 Visão geral da Tecnologia de Web Services

2.3.7.1 XML

A CISCO (2005) define XML como sendo uma linguagem de marcação para documentos que contém informação estruturada. Onde, uma linguagem de marcação é considerada como um mecanismo de identificação de estruturas num documento, e a especificação XML define uma forma padronizada de criação ou adição de marcações em documentos.

XML pode ser uma definição conduzida por uso de DTDs e Schemas que permitem a manipulação de informações entre aplicações. Podem ser combinadas Tags, podem ser definidas interfaces e processamentos podem ser padronizados.

XML está também se tornando um importante padrão para a troca de uma vasta variedade de dados na web, estando rapidamente se tornando a linguagem de negócio para transacções efectuadas entre sistemas legados, parceiros, empresas, clientes etc. XML provê uma abstracção e representação lógica dos dados que a torna flexível e universal possibilitando assim a troca de dados entre sistemas e plataformas diferentes.

Actualmente a linguagem XML é de facto um padrão para inúmeros processos de negócios efectuados via Web Services na Internet. Web Services são componentes de programas reutilizáveis que utilizam XML como padrão de intercâmbio de informações entre aplicações. Com o surgimento da Arquitectura Orientada a Serviços e a perspectiva de sua adopção em larga escala pelo mercado, fez com que a linguagem XML se tornasse ainda mais expressiva no cenário mundial de negócios (ROCHA, 2007).

2.3.7.2 SOAP

O protocolo SOAP foi criado para transportar mensagens XML de um computador para outro, via vários protocolos padrões de transporte. HTTP é o mais comum destes protocolos, obviamente por predominante na Web.

SOAP se define usando XML, que proporciona com simplicidade e coerência uma maneira de uma aplicação enviar mensagem XML para outra. SOAP é o que faz a integração entre aplicações ser possível, pois após a definição do conteúdo do XML, é o SOAP que transfere os dados de um lugar para outro pela rede. Permite enviar e receber documentos baseados em XML que suportam um protocolo comum de transferência de dados. Além disso, SOAP permite tratar mensagens XML retornadas de um serviço remoto e seu modelo

possibilita de forma clara a separação entre os dados de processamento de infra-estrutura e processamento de mensagens de aplicação.

A estrutura básica de uma mensagem SOAP e apresentada da seguinte maneira:

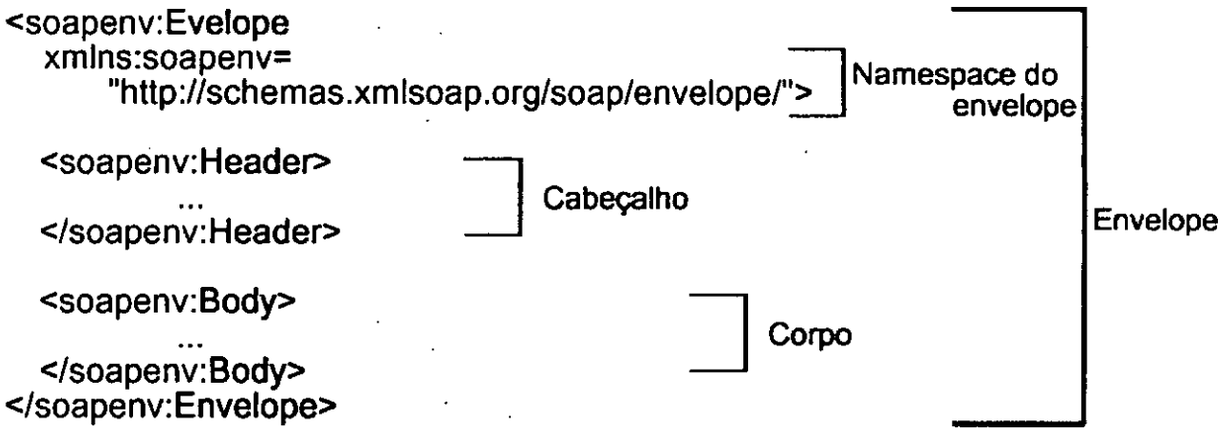


Figura. 2.4: Estruturada básica de uma mensagem SOAP adaptado de (KANNEGANT e CHODAVARAPU, 2008).

O elemento raíz de uma mensagem SOAP é o Envelop, sendo este elemento indispensável para qualquer mensagem SOAP. A notação designada como Namespace do envelope (xmlns:soapenv= "http://schemas.xmlsoap.org/soap/envelop/") serve para definir que o documento XML é uma mensagem SOAP.

Por sua vez, o envelope SOAP é composto por um elemento opcional que é o cabeçalho seguido do corpo sendo este um elemento obrigatário.

Cabeçalho SOAP (Header) – pode conter uma ou mais entradas que podem ser usadas para extender SOAP e/ou expressar semânticas adicionais de aplicações. Estas entradas podem ser credenciais de usuários entre outros, de modo a extender segurança para SOAP.

Corpo SOAP (body) – contém a mensagem SOAP propriamente dita (informação útil). A mensagem poderia ser um documento com uma ordem de pagamento ou uma entrega, poderia ser uma descrição de uma classe com métodos remotos e seus parâmetros. Esta mensagem deve ser sempre transportada como um elemento XML, não sendo possível transportar mensagens completamente de texto, ou seja, obrigatoriamente as mensagens devem estar encapsuladas dentro dum elemento XML (KANNEGANT, 2008).

SOAP precisa ser seguro. A mensagem transportada deve ser somente de conhecimento dos seus receptores. O serviço remoto deve conhecer quem está requisitando seu serviço e se está autorizado. SOAP é um mecanismo de pacote para mensagens XML e documentos. Muitos pacotes necessitam descrever importantes informações sobre o que o pacote todo contém, por exemplo: o remetente, como o receptor vai validar o remetente, quais permissões o remetente possui e assim por diante. Isto basicamente representa a implementação de segurança na própria mensagem SOAP (ROSENBERG, 2004).

2.3.7.3 WSDL

WSDL é uma linguagem XML que define as operações que um serviço Web provê e a estrutura dessas operações em relação às mensagens SOAP. Ou seja, define as estruturas de entrada e saída de um Web Service, associação entre os parâmetros e tipos de dados.

WSDL contém informações de um serviço para que outros possam interagir com este serviço, como a localização do serviço, as possíveis operações que o serviço pode fazer e de que maneira é possível chamá-lo.

Um WSDL possui três secções: o quê, como e onde. Na primeira secção, um ficheiro WSDL especifica as mensagens de entrada e saída, representando o que o serviço faz. A segunda secção define como as mensagens devem se empacotadas na mensagem SOAP e como devem ser transportadas. Além disso, define que informação deve estar no cabeçalho do SOAP. A terceira e última secção descreve a implementação específica de um serviço Web e onde poderá ser encontrado (ROSENBERG, 2004).

2.3.7.4 UDDI

SOA baseia-se na capacidade de identificar serviços e suas características. Consequentemente, esta arquitectura depende de um directório que descreva quais os serviços disponíveis dentro de um domínio. Pode-se considerar as "paginas amarelas" como um exemplo com a mesma filosofia, onde pode-se procurar por determinada organização, serviços oferecidos por esta, ou mesmo contactar a organização para obtenção de mais informações.

A UDDI provê um método padronizado para a publicação e descoberta de informações sobre Web Services. Deste modo UDDI é uma framework independente de plataforma para descrição de serviços, descoberta de negócios, e integração de serviços de negócios a partir da internet (W3SCHOOLS).

2.4 Segurança em Arquitectura Orientada a Serviços

O sub-capítulo que se segue, aborda acerca dos padrões e boas praticas de segurança em Arquitectura Orientada a Serviços, com principal enfoque a Web Services, que será a tecnologia de implementação de SOA abordado no presente trabalho.

SOA promove através de suas regras e conceitos inovadores a facilidade de comunicação dinâmica entre serviços conectados por redes como a Internet.

Basicamente, essa comunicação ocorre entre serviços (por exemplo, Web Services) provenientes de transacções geradas por processos de negócios entre clientes, fornecedores, parceiros de negócios e até mesmo entre serviços mal intencionados. Considerando as facilidades de comunicação e as inovações nos processos de negócios, podemos afirmar que mecanismos de segurança são extremamente necessários para atender aos novos requisitos de segurança que surgem motivados pela arquitectura SOA (ROCHA, 2007).

WAGNER (2004), director de pesquisa do departamento de segurança estratégica do Gartner, afirma que o factor determinante da adopção em massa de Web Services depende do sucesso da utilização de tecnologias de segurança baseados em padrões. Atendendo e considerando que Web Services estão se tornando a tecnologia preferida para implementação da arquitetura SOA, torna-se evidente que é importante que surjam novos e actuais mecanismos de segurança.

2.4.1 Complexidade da Segurança em Arquitectura Orientada a Serviços

Web Services são a encarnação mais nova de middleware[5] da computação distribuída, diferenciando-se de todas as formas de midleware passadas, sendo que, Web Services são mais simples, baseadas em padrões e com maior desacoplamento para conexão de dados, sistemas e organizações, entretanto exige que o comprometimento em relação a segurança tenha maior ênfase do que qualquer outra tecnologia de midleware antecedente (ROSENBERG, 2004).

[5]mediador entre aplicações, ou seja, designa camadas de software que não constituem directamente a aplicações, mas que facilitam a comunicação entre estas.

Algumas organizações receiam a implementação de SOA devido aos desafios de segurança que possam encarar, entretanto outras organizações menos atentas não tem percepção ou entendimento básico das principais diferenças e desafios dos aspectos de segurança existentes em SOA e obcecados pelas vantagens e benefícios que esperam alcançar implementando a SOA, a preocupação com a segurança é deixada como segundo plano.

Alguns motivos que fazem com que a Segurança em Web Services seja mais complexas do que em qualquer outro midleware passado, segundo ROSENBERG (2004), são descritos abaixo:

- As integrações a partir de Web Services são desacopladas (loose coupled);
- Não são somente usados para integração de sistemas internos, porém são usados também para integração de fontes de dados provenientes de fora da organização;
- São baseados na passagem ou transmissão de mensagens de negócios legíveis e autodescritivos representados em XML;
- São baseados em tecnologias Web subjacentes (XML, SOAP, UDDI, WSDL, etc.) que por sua vez tem seus próprios desafios de segurança.

A segurança para negócios baseados em Web Services pode ser implementada em dois diferentes níveis, sendo o primeiro o nível de transporte e o segundo o nível de mensagens. Os mecanismos de segurança utilizados no transporte e nas mensagens, podem ser implementados separadamente ou em conjunto. Cada um desses mecanismos de segurança dispõe de padrões e protocolos específicos para atender os diferentes requisitos de seguranças de cada negócio proporcionados pela arquitectura SOA (ROCHA, 2007).

2.4.2 Segurança no transporte

Neste sub-capítulo, abordar-se-a de uma forma rápida, os mecanismos de segurança tradicionais e avaliá-las no contexto das mudanças que SOA pode trazer às TIC.

Actualmente existe uma grande variedade de serviços electrónicos oferecidos via Internet, a maioria desses serviços conhecemos a bastante tempo e seria complicado viver sem eles. Estamos falando de serviços bancários oferecidos via Web, lojas virtuais on-line,

servidores de web-mail etc. Praticamente, esses serviços já fazem parte de nossa rotina diária o que nos faz bastante confortáveis em utilizá-los sem o menor problema, embora exista uma forte preocupação relacionada à segurança da informação.

O que a grande maioria dos usuários desses serviços desconhece ou acha que o assunto é muito técnico para entender mesmo os utilizando no dia a dia, são os inúmeros mecanismos de segurança existentes e utilizados nesses serviços. O mecanismo de segurança para esses tipos de serviços electrónicos que já estamos acostumados a utilizar, garante o sigilo dos dados na comunicação entre as nossas máquinas e os servidores que estamos acessando, ou seja, os dados ou mensagens são transmitidas de maneira que seja extremamente complexo e difícil que pessoas alheias interceptem e leiam.

Os mecanismos e protocolos utilizados para garantir segurança dos dados trafegados entre empresas e parceiros, são praticamente os mesmos utilizados nas transacções electrónicas que efectuamos quando pagamos uma conta on-line na Internet.

Geralmente, as empresas por sentirem necessidade de adequação dos mecanismos de segurança nos seus negócios, acabam exigindo um pouco mais nos requisitos de segurança dos processos de negócios efectuados electronicamente. Os mecanismos adicionais que podem ser inclusos são a autenticação mútua dos provedores e consumidores de serviço utilizando certificados digitais. Além disso, também existe a possibilidade de utilizar assinatura digital nos processos, cifrar os dados ou mesmo combinar diversos mecanismos de segurança de modo a obter maior confidencialidade, confiabilidade, interoperabilidade entre outros.

Todos esses cenários de comunicações mencionados estão presentes no nosso dia a dia e a tendência é de crescimento de utilizadores desses serviços. Tudo isso em função do baixo custo de comunicação que a Internet proporciona e o alto nível de segurança que atingimos ao longo do tempo. Os mecanismos responsáveis pela segurança e sigilo dos dados em canais de comunicação ponto a ponto (cliente/servidor) são baseados nos protocolos SSL (Security Socker Layer) e/ou TLS (Transport Layer Security). Esses protocolos se tornaram padrão no mercado e estão presentes na grande maioria das transacções seguras efectuadas actualmente na Internet.

A segurança no transporte garante-nos sigilo dos dados transmitidos de uma máquina (cliente/servidor) a outra máquina (servidor/cliente), isso significa que a comunicação entre um consumidor de serviço e um provedor de serviço estará totalmente segura contra violação de sigilo (ROCHA, 2007). Considero a afirmação citada anteriormente relativa,

visto que não concordo com a totalidade do sigilo, mas sim com a altíssima complexidade e probabilidade quase nula para invasão de privacidade. A Figura 5. ilustra a arquitectura de segurança a nível de transporte, onde a aplicação gere a segurança interna desta e confia em canais seguros para protecção dos dados intercambeados com a aplicação cliente.

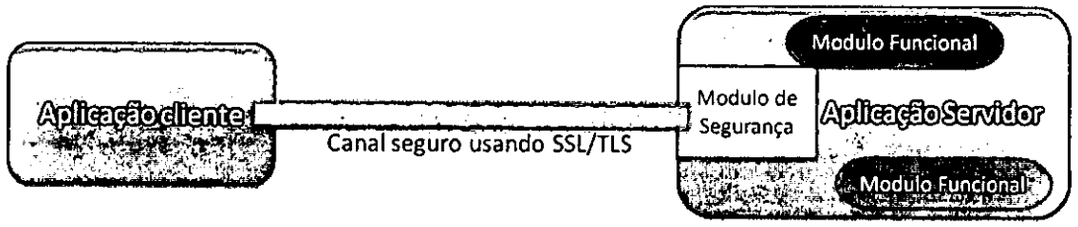


Figura. 2.5: Arquitectura de segurança no transporte (adaptado de KANNEGANT, 2008).

No cenário da Figura. 2.5, um único servidor pode ter varias funcionalidades independentes para oferecer aos clientes, mas tem somente um e único módulo de segurança, ou seja, módulo responsável pela cifragem ou decifragem de dados enviados ou recebidos pela aplicação, garantindo deste modo o sigilo na comunicação.

Pode estar a questionar-se, mas qual é o problema deste cenário? A resposta é nenhum! Pelo menos no contexto de algumas aplicações ou serviços tradicionais como web-mail, compras online, entre outros, este tipo de segurança funciona perfeitamente.

Entretanto, vão surgindo cada vez mais novos modelos de negócios proporcionando uma demanda crescente de serviços intermediários entre o consumidor e o provedor de serviços. Sendo assim, os mecanismos tradicionais de confidencialidade de dados usando somente SSL/TLS não são viáveis.

Apesar de crescente a demanda na utilização de serviços intermediários, é notório que os requisitos e os mecanismos de segurança permaneceram praticamente os mesmos. Isso significa que muitos modelos de negócios mesmo utilizando serviços intermediários no processo de negócio continuam a utilizar mecanismo de segurança como SSL/TLS para garantir a segurança da comunicação, não se importando (talvez devido a complexidade ou dificuldade de implementação) com os novos requisitos de segurança que o modelo utilizando middlewares ou interfaces intermediárias proporciona.

No que se assemelha ao visto anteriormente, segundo ROCHA (2007) quando utilizamos mecanismo de segurança no transporte proporcionado pelos protocolos SSL/TLS,

estamos garantindo apenas que a comunicação entre dois serviços seja segura e com total sigilo. Isso ocorre porque a comunicação é estabelecida e cifrada na camada de transporte apenas, o que significa de maneira bem simples receber os dados enviados cifrados na camada de transporte, efectuar o deciframento dos dados e envia-los para a camada de aplicação processar os dados (a explicação das 7 camadas OSI não faz parte do escopo desta trabalho). Todavia, se existirem interfaces intermediárias entre provedor e consumidor, o processo de criptografia e decriptografia na camada de transporte irá ocorrer toda vez que os dados trafegarem por um serviço intermediário, resultando na quebra do sigilo e segurança em cada serviço intermediário.

Consideremos o cenário ilustrado na Figura 2.6, em que o requisitante precisa de um serviço disponibilizado pelo provedor do serviço, mas por alguma viabilidade de negócio é necessário que se tenha um serviço intermediário.

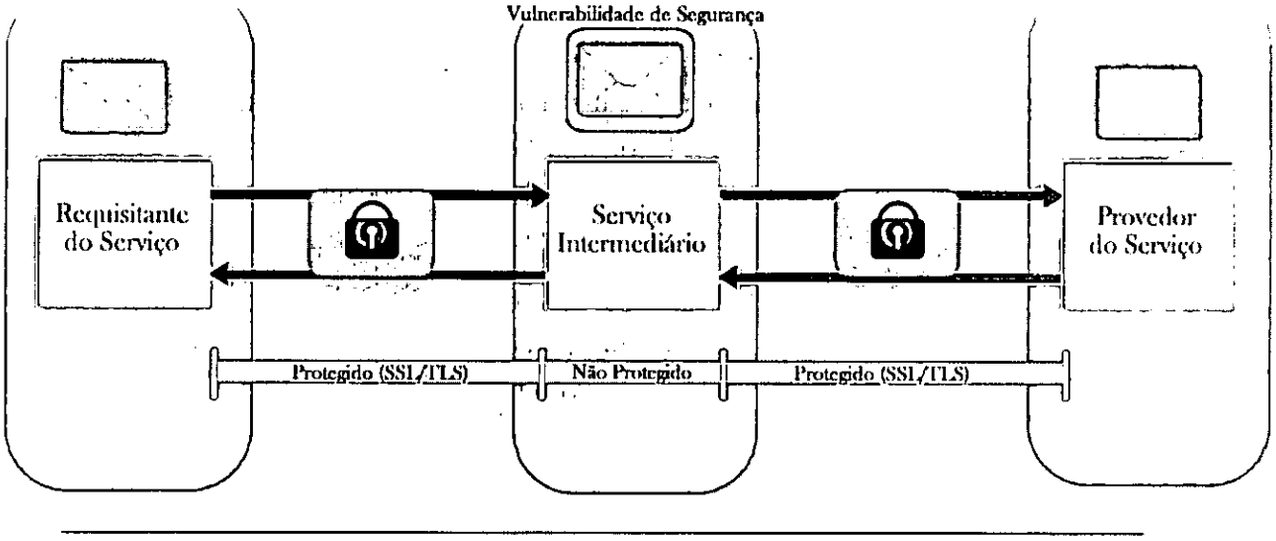


Figura. 2.6: Arquitectura de segurança no transporte com incorporação de serviço intermediário.

É notável que o mecanismo de segurança oferecido pela camada de transporte não proporciona confidencialidade dos dados, visto que o serviço intermediário recebe os mesmos, ou seja, SSL/TLS pode proteger a mensagem do requisitante no canal de comunicação, mas logo que a mensagem chega a aplicação do serviço intermediário, a responsabilidade de SSL/TLS termina e a aplicação esta livre de ler e usar os dados na mensagem. Perante este cenário, claramente precisamos de melhores técnicas para confidencialidade dos dados, um dos destaques seria usar os novos protocolos para garantir a segurança das

mensagens para complementar os mecanismos de segurança já oferecidos pelos protocolos SSL/TLS.

2.4.3 Segurança na mensagem

A necessidade de prover segurança na mensagem surgiu paralelamente com a preocupação de prover segurança no transporte devido a algumas limitações desta. Entretanto, a crescente adopção de SOA, com maior enfoque as características inovadoras de negócios proporcionados por ela, fez com que segurança de mensagens se tornasse um conceito relevante.

Uma das características da SOA que proporciona o requisito de utilização de mecanismos de segurança de mensagens, é a possibilidade de utilização de serviços intermediários no processo de troca de dados via mensagens SOAP/XML (ROCHA, 2007).

Suponhamos que temos um processo de pagamento electrónico por cartão de crédito utilizando uma SOA em que temos um serviço intermediário entre o requisitante e o provedor do serviço (seguindo a abordagem de segurança da Figura 2.6.), neste contexto um administrador do serviço intermediário pode ter acesso aos dados confidenciais do cliente visto que dentro do serviço intermediário os dados não estão criptografados ou não possuem nenhum mecanismo de segurança. De modo a superar esses inconvenientes existem padrões de segurança que podem ser implementados a nível da mensagem (SOAP/XML) além da confidencialidade a nível de transporte (SSL/TLS), garantindo desta forma a confidencialidade da mensagem trafegada pelos possíveis intermediários de serviços que possam surgir.

Consideremos o comentário seguinte, descrito por ROCHA (2007):

"Actualmente é muito comum fazermos o pagamento electrónico de compras efectuadas on-line na Internet utilizando cartão de crédito. No processo de pagamento electrónico geralmente temos uma página Web onde digitamos os dados confidenciais de cartão de crédito para serem validados junto a entidade administradora do cartão, que consequentemente recebe os dados processa os mesmos e efectua a aprovação/recusa do processo de pagamento da compra. Certamente ficamos preocupados com a segurança neste tipo de comunicação, no entanto quase a maioria desse tipo de processo de pagamento on-line uti-

liza os protocolos SSL/TLS para garantir que os dados enviados estejam seguros quando trafegados pela Internet.

O interessante é que pouca gente sabe que os dados confidenciais do cartão de crédito são geralmente expostos no processo de compra on-line, e que essa exposição de dados poderia ser evitada simplesmente deixando de passar em texto aberto pelo serviço intermediário (loja virtual). Isso ocorre em função da particularidade dos protocolos de segurança utilizados na camada de transporte, SSL/TLS, ou seja, os dados enviados são cifrados no serviço consumidor e decifrados no serviço intermediário (loja virtual). Depois eles são novamente criptografados e enviados a outro serviço disponibilizado pelo administrador do cartão de crédito para validação e aprovação do pagamento da compra."

Segundo o comentário acima, é exactamente no momento em que os serviços intermediários decifram os dados que a **vulnerabilidade** de segurança ocorre, deste modo, no serviço intermediário, um administrador malicioso pode usufruir dos dados do cartão de crédito. O desafio existente em SOA é garantir que mesmo o administrador do serviço intermediário não tenha acesso aos dados do requisitante do serviço garantindo assim a confidencialidade da mensagem a partir do requisitante do serviço até ao provedor do serviço, sendo lá onde os dados serão totalmente decifrados.

A alternativa de segurança para a arquitectura SOA é recente e nasceu com a demanda crescente da utilização do XML. Com uma arquitectura formada pela troca de mensagens SOAP/XML, se fez necessário a criação de mecanismos de segurança baseados em XML que possibilitasse a segurança e sigilo dos dados trocados. Assim nasceu XML Digital Signature e XML encryption, protocolos que garantem a segurança e sigilo das mensagens. Esses protocolos são basicamente um complemento dos protocolos SSL/TLS, e eles atendem aos exigentes requisitos de segurança proporcionados pela arquitectura SOA (ROCHA, 2007). A figura 2.7 ilustra o cenário em que temos serviços intermediários e a confidencialidade da mensagem é garantida em todo o caminho entre os serviços da extremidade (do requisitante ao provedor).

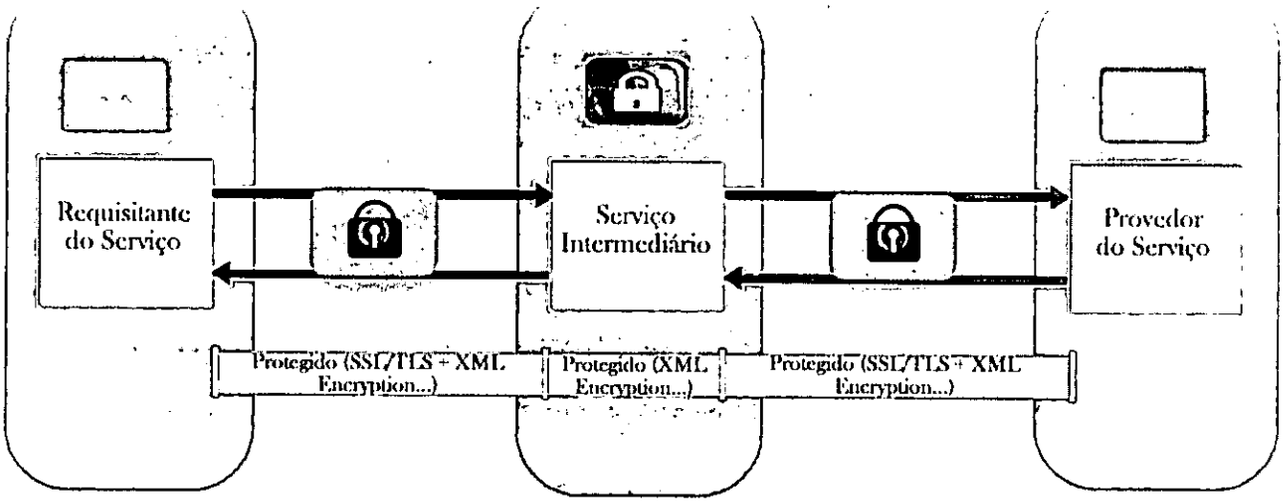


Figura. 2.7: Arquitectura de segurança no transporte e na mensagem com incorporação de serviço intermediário.

No cenário acima, a mensagem é cifrada baseado num dos vários algoritmos de criptografia existentes e posteriormente enviada por um canal seguro (SSL/TLS), sendo assim, a mensagem além de estar livre de possíveis ataques durante a transmissão, está livre também de ser acessada pelo serviço intermediário visto que está cifrada.

Este cenário representa uma abordagem mais flexível e poderosa comparada aos níveis de rede e transporte, usada principalmente pela especificação WS-Security (MICROSOFT, 2004).

2.5 Mecanismos de Segurança

Neste sub-capítulo serão retratados os mecanismos de segurança implementadas no presente trabalho.

Existem actualmente no mercado diferentes tipos de protocolos e mecanismos de segurança que visam proporcionar e garantir segurança e sigilo na troca de informações em que cada um desses mecanismos tem funcionalidade e aplicabilidade específica de acordo com o tipo de negócio ou requisito de segurança exigido. Tendo como exemplo, os protocolos SSL/TLS juntamente com a Infraestrutura de Chaves Públicas (ICP) são de facto

os mecanismos de segurança mais utilizados mundialmente na Internet para prover segurança na troca de informações de negócios entre empresas, parceiros e clientes (Comercio electrónico, EAI, Bancos etc.) (ROCHA, 2007).

O mecanismo de segurança proporcionado pelos protocolos TLS/SSL juntamente com ICP não se adequam a ambientes de negócios em que temos serviços intermediários visto que estes mecanismos de segurança somente garantem segurança e sigilo das informações a nível de transporte entre o serviço consumidor e o serviço provedor.

Como ilustrado na figura 2.7 as informações transmitidas do requisitante do serviço devem ser cifradas de tal forma que somente o provedor do serviço possa ter acesso e vice-versa, é evidente que os mecanismos de segurança por exemplo, baseados em TLS/SSL não satisfazem tais requisitos, deste modo surgiu a segurança a nível da mensagem para sanar este problema. Com o mecanismo de segurança a nível da mensagem, diferentes partes da mensagem podem ser protegidos de diferentes maneiras, de modo que sejam usáveis ou acedidas somente por entidades previstas durante o percurso (KANNEGANT, 2008). Os mecanismos para segurança a nível de mensagem foram desenvolvidos especialmente para aplicações baseadas em XML[6], sendo este o principal mecanismo de intercâmbio de informações usado em Web Services.

2.5.1 Padrão de Segurança WS-Security

No presente sub-capítulo procurei explicar os conceitos chaves relacionado a segurança de Web Services baseado no padrão ws-security.

A especificação Web Services Security, ou WS-Security e um mecanismo de incorporação de segurança em mensagens SOAP, criadas em abril de 2002, pela Microsoft, IBM e Verisign, e foi aprovada pelo comité técnico da OASIS (Organization for the Advancement of Structured Information Standards) como um padrão em 2004.

WS-Security suporta, integra e unifica vários modelos, mecanismos e tecnologias de segurança em uso no mercado, permitindo que vários sistemas possam interoperar em plataformas e linguagens heterogéneas.

[6]é uma especificação com uma abordagem generalizada que permite a criação de linguagens de marcação personalizados.

WS-Security consiste basicamente em adicionar informações sobre assinaturas digitais, criptografia, timestamps e tokens[7] de segurança dentro do elemento <wsse:Security>, localizado no cabeçalho da mensagem SOAP (soap:header), assunto que será detalhado posteriormente.

As novas especificações de segurança definem um conjunto de padrões para extensões SOAP, utilizados para oferecer maior integridade, não repúdio, confidencialidade e autenticação de mensagens SOAP transmitidas entre aplicações.

A *autenticação* como abordado anteriormente, está relacionada a identificação do remetente. O WS-Security usa tokens de segurança para manter essas informações com um cabeçalho de segurança da mensagem SOAP.

A *integridade* da mensagem é obtida com assinaturas digitais XML[8](XML Digital Signature), garantindo deste modo que a mensagem não tenha sido alterada após a assinatura do remetente.

A *confidencialidade* da mensagem é baseada na especificação de criptografia XML[9] (XML Encryption) e garante que esta, só possa ser compreendida ou acedida pelo(s) destinatário(s) desejado(s).

[7]segmento de texto ou símbolo que pode ser manipulado por um parser
[8]especifica uma forma de criar assinaturas digitais para o uso em transacções de XML.
[9]processo de cifrar e decifrar partes de um documento XML.

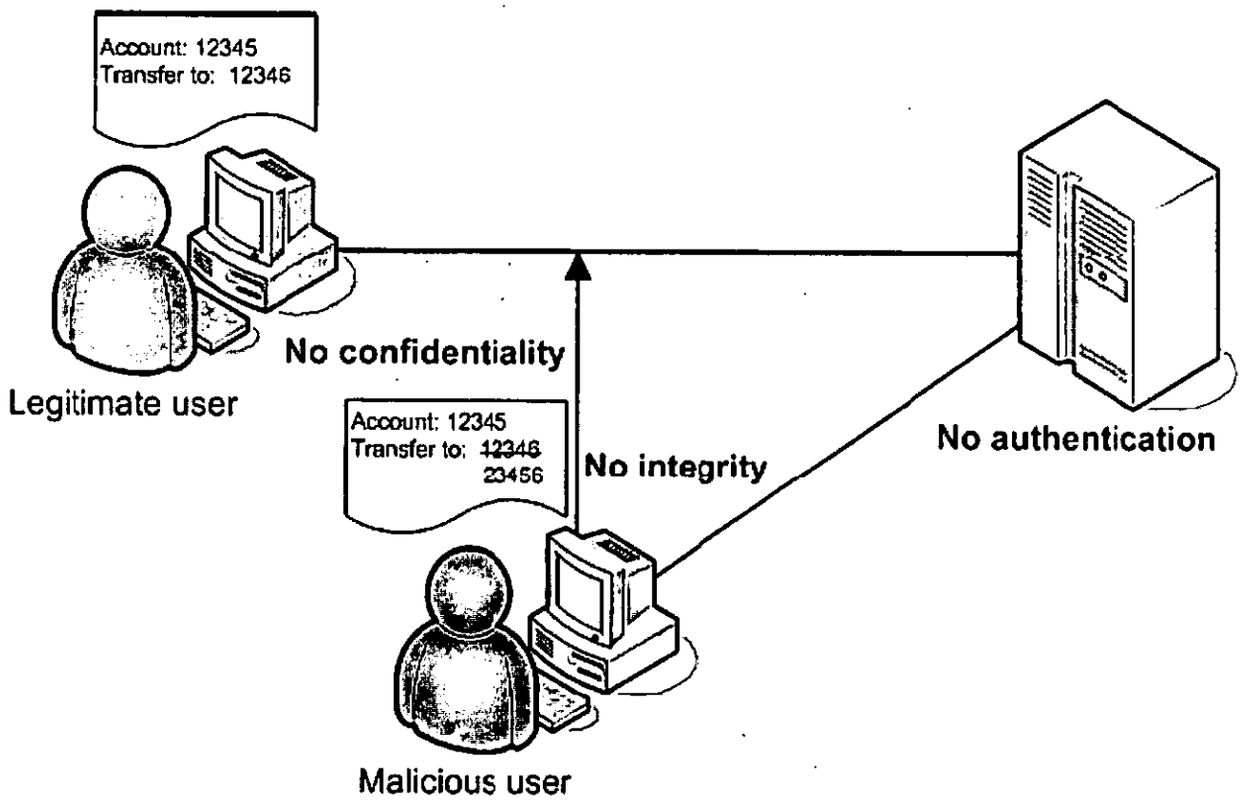


Figura. 2.8: Cenário ilustrativo de um ambiente com falta de autenticação, confidencialidade e integridade. (IBM, 2006)

O *não repúdio* é aplicado de modo a garantir que uma entidade que realizou determinada operação não possa negá-la posteriormente, e é importante para determinar a responsabilidade pelos resultados das operações realizadas, pelas partes envolvidas bem como para realização de auditorias no sistema.

A Figura 2.9 ilustra uma mensagem que implementa segurança baseada em tokens, assinaturas digitais e criptografia.

(001) <?xml version="1.0" encoding="utf-8"?>
(002) <S11:Envelope xmlns:S11="..." xmlns:wsse="..." xmlns:wsu="..."
xmlns:xenc="..." xmlns:ds="...">
(003) <S11:Header>
(004) <wsse:Security>
(005) <wsu:Timestamp wsu:Id="T0">

(006) <wsu:Created>
(007) 2004-09-01T18:27:09Z</wsu:Created>
(008) </wsu:Timestamp>
(009)
(010) <wsse:BinarySecurityToken ValueType="...#X509v3"
wsu:Id="X509Token" EncodingType="...#Base64Binary">
(011) MIIEZzCCA9CgAwIBAgIQEmtJZc0rqrKh5i...
(012) </wsse:BinarySecurityToken>
(013) <xenc:EncryptedKey>
(014) <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
(015) <ds:KeyInfo>
(016) <wsse:KeyIdentifier EncodingType="...#Base64Binary"
ValueType="...#X509v3">MIGfMa0GCSq...
(017) </wsse:KeyIdentifier>
(018) </ds:KeyInfo>
(019) <xenc:CipherData>
(020) <xenc:CipherValue>d2FpbmdvbGRfE0lm4byV0...
(021) </xenc:CipherValue>
(022) </xenc:CipherData>
(023) <xenc:ReferenceList>
(024) <xenc:DataReference URI="#enc1"/>
(025) </xenc:ReferenceList>
(026) </xenc:EncryptedKey>
(027) <ds:Signature>
(028) <ds:SignedInfo>
(029) <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
(030) <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
(031) <ds:Reference URI="#T0">
(032) <ds:Transforms>
(033) <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
(034) </ds:Transforms>
(035) <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
(036) <ds:DigestValue>LyLsF094hPi4wPU...
(037) </ds:DigestValue>
(038) </ds:Reference>
(039) <ds:Reference URI="#body">
(040) <ds:Transforms>

```
(041) <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
(042) </ds:Transforms>
(043) <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
(044) <ds:DigestValue>LyLsF094hPi4wPU...
(045) </ds:DigestValue>
(046) </ds:Reference>
(047) </ds:SignedInfo>
(048) <ds:SignatureValue>
(049) Hp1ZkmFZ/2kQLXDJbchm5gK...
(050) </ds:SignatureValue>
(051) <ds:KeyInfo>
(052) <wsse:SecurityTokenReference>
(053) <wsse:Reference URI="#X509Token"/>
(054) </wsse:SecurityTokenReference>
(055) </ds:KeyInfo>
(056) </ds:Signature>
(057) </wsse:Security>
(058) </S11:Header>
(059) <S11:Body wsu:Id="body">
(060) <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
wsu:Id="enc1">
(061) <xenc:EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#tripledescbc"/>
(062) <xenc:CipherData>
(063) <xenc:CipherValue>d2FpbmdvbGRfE0lm4byV0...
(064) </xenc:CipherValue>
(065) </xenc:CipherData>
(066) </xenc:EncryptedData>
(067) </S11:Body>
(068) </S11:Envelope>
```

Figura. 2.9: Mensagem SOAP baseada no padrão WS-Security.

Abaixo será explicado a função dos blocos existentes na mensagem acima (Figura 2.9) ilustrada.

As linhas [003-058] representam o cabeçalho da mensagem SOAP.

As linhas [004-057] representam o bloco <wsse:Security> do cabeçalho. Este contém a segurança da informação para esta mensagem.
As linhas [005-008] especificam o momento da criação da mensagem.
As linhas [010-012] especificam o token de segurança associado à mensagem. Neste caso é especificado um certificado X.509 que está codificado como Base64.
As linhas [013-026] especificam a chave usada para cifrar o corpo da mensagem.
As linhas [027-056] especificam a assinatura digital. Neste exemplo, a assinatura é baseada num certificado X.509.
A linha [039] referencia o corpo da mensagem.
As linhas [048-050] indicam o actual valor da assinatura que foi especificada na linha (043).
As linhas [052-054] indicam a chave usada na assinatura. O corpo da mensagem está representado pelas linhas [059-067].
As linhas [060-066] representam os metadados da criptografia e formas do corpo usando XML Encryption.
As linhas [063-064] contêm o corpo cifrado (IBM, 2003).

2.5.1.1 XML Digital Signature

O padrão XML digital signature foi desenvolvida pela parceria das organizações World Wide Web Consortium (W3C) e Internet Engineering Task Force (IETF). O padrão proporciona regras e sintaxes para criar e representar assinatura digital para transacções baseadas em XML garantindo o não repúdio, integridade, autenticidade (da mensagem e/ou do remetente). Uma das características fundamentais do XML digital signature é a habilidade e flexibilidade de assinar digitalmente somente parte da mensagem XML (ROCHA, 2007), sendo esta flexibilidade muito importante para modelos de negócios baseados nas aplicações em XML em que as mensagens podem passar por vários serviços ou intermediários de serviços, clarificando assim a sua importância numa arquitectura SOA.

XML digital signature depende necessariamente de uma infra-estrutura de ICP para viabilizar o processo de não repúdio, integridade e autenticação. A integração com uma infra-estrutura de ICP é requisito fundamental para o processo de verificação e validação da assinatura digital. Os novos modelos de negócios baseados na arquitectura SOA utilizam o padrão XML digital signature para garantir os requisitos de segurança conforme exigência do negócio (ROCHA, 2007).

A estrutura básica de uma assinatura XML segundo (ROSENBERG, 2004) é listada

na figura abaixo:

```
<Signature>
<SignedInfo>
(CanonicalizationMethod)
(SignatureMethod)
(<Reference(URI=)?>
(Transforms)?
(DigestMethod)
(DigestValue)
</Reference>)+
</SignedInfo>
(SignatureValue)
(KeyInfo)?
(Object)*
</Signature>
```

Figura. 2.10: Estrutura básica de uma assinatura XML.

A informação assinada aparece dentro do elemento <SignedInfo>. O algoritmo usado no cálculo do elemento (SignatureValue) é mencionado dentro da secção assinada. O elemento (SignatureMethod) especifica o algoritmo usado para converter o <SignedInfo> canonizado[10] no (SignatureValue). Esta é uma combinação de um algoritmo que depende da chave e de um algoritmo de resumo. O elemento (KeyInfo) indica a chave que é usada para validar a assinatura. Possíveis formas de identificação ou validação são: certificados, nomes de chaves, algoritmos de aceitação de chaves e informação (SIMON, 2001).

Cada recurso por assinar, deve ter seu próprio elemento <Reference>, identificado pelo atributo URI. O elemento (Transform) especifica uma lista ordenada de processos que são aplicados ao conteúdo do recurso especificado antes de ser aplicada a função hash[11]. O (DigestValue) é o elemento que recebe o resultado da função hash aplicada ao recurso. O elemento (Object) é opcional e pode albergar informação diversa não incluída nos ele-

[10]algoritmo de verificação de inconsistência em mensagens XML antes de extrair a representação em bits para posterior processamento de uma assinatura.

[11]sequência de letras ou números geradas por um algoritmo de dispersão.

mentos anteriores.

2.5.1.2 XML Encryption

Como referenciado anteriormente, XML Encryption provê segurança ao nível da mensagem através da cifragem destas, evitando que terceiros leiam ou vejam seu conteúdo. O padrão XML Encryption proporciona regras e sintaxes para criar e representar criptografia para garantir o sigilo em transacções baseadas em XML, tendo flexibilidade e habilidade de criptografar somente certas partes de determinada mensagem XML.

O mecanismo de segurança SSL/TSL é de facto o padrão de comunicação segura commumente usada na internet. O padrão XML Encryption não pretende substituir SSL/TSL, mas sim prover um mecanismo para requisitos de segurança que não é coberto pelo SSL/TSL, tal como o estabelecimento de uma sessão segura entre dois ou mais pontos e a criptografia de partes de uma mensagem podendo usar algoritmos de criptografia diferentes para cada parte.

Tomemos como exemplo, transacções de compras efectuadas via Internet em que os dados confidencias são transmitidos entre cliente, loja electrónica e uma entidade administradora do cartão de crédito do cliente. No caso apresentado, os dados do cliente passam por mais de um serviço receptor de dados, e consequentemente de modo a garantir sigilo e confiabilidade da transacção do cliente, a loja electrónica não precisa conhecer por exemplo o número de cartão de crédito do cliente, bastando que a entidade administradora aprove o cartão possibilitando a efectivação da venda ao cliente pela loja.

Para esse tipo de transacção de negócio, que possibilita a validação de cartão de crédito baseado em aplicações XML, poderíamos simplesmente utilizar o mecanismo de criptografia para XML, ou seja, os dados confidenciais do cartão de crédito (número do cartão, senha, etc.) ficariam criptografados entre o cliente e a operadora do cartão. Esse mecanismo de segurança proporcionado pelo XML Encryption, beneficia a loja electrónica que não necessita se preocupar com a segurança dos dados confidenciais bem como a operadora juntamente com o cliente visto que as informações do cartão de crédito trocados permanece em sigilo.

Abaixo são ilustrados alguns exemplos de criptografia de mensagens XML usando o padrão XML Encryption, que serão feitas de diversas formas (criptografia do ficheiro XML

inteiro, criptografia de um elemento do XML e criptografia do conteúdo de um elemento).

- Exemplo de um ficheiro XML a ser cifrado, SIDDIQUI (2002).

  <purchaseOrder>
  <Order>
  <Item>book</Item>
  <Id>123-958-74598</Id>
  <Quantity>12</Quantity>
  </Order>
  <Payment>
  <CardId>123654-8988889-9996874</CardId>
  <CardName>visa</CardName>
  <ValidDate>12-10-2004</ValidDate>
  </Payment>
  </purchaseOrder>

- Criptografia de um ficheiro XML inteiro, SIDDIQUI (2002).

  <?xml version='1.0' ?>
  ...
  <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
  Type='http://www.isi.edu/in-notes/iana/assignments/mediatypes/text/xml'>
  <CipherData>
  <CipherValue>A23B45C56</CipherValue>
  </CipherData>
  </EncryptedData>...

- Criptografia somente do elemento <payment> , SIDDIQUI (2002).

  <?xml version='1.0' ?>
  ...
  <PurchaseOrder>
  <Order>
  <Item>book</Item>
  <Id>123-958-74598</Id>
  <Quantity>12</Quantity>
  </Order>
  <EncryptedData Type='http://www.w3.org/2001/04/xmlenc#Element'

xmlns='http://www.w3.org/2001/04/xmlenc#'> <CipherData>
<CipherValue>A23B45C564587</CipherValue>
</CipherData>
</EncryptedData>
</PurchaseOrder>...

- Criptografia somente do conteúdo dentro do elemento <CardId>, SIDDIQUI (2002).
  <?xml version='1.0' ?>
  ...
  <PurchaseOrder>
  <Order>
  <Item>book</Item>
  <Id>123-958-74598</Id>
  <Quantity>12</Quantity>
  </Order>
  <Payment>
  <CardId>
  <EncryptedData
  Type='http://www.w3.org/2001/04/xmlenc#Content'
  xmlns='http://www.w3.org/2001/04/xmlenc#'>
  <CipherData>
  <CipherValue>A23B45C564587</CipherValue>
  </CipherData>
  </EncryptedData></CardId>
  <CardName>visa</CardName>
  <ValidDate>12-10-2004</CardName>
  </Payment>
  </PurchaseOrder>...

Com base nos exemplos ilustrados anteriormente, creio que ficou relativamente mais clara a visão acerca dos mecanismos de implementação de segurança a nível da mensagem usando XML Encryption.

2.5.2 Vulnerabilidades Comuns

Este sub-capítulo visa abordar algumas vulnerabilidades que estão no top 10 de vulnerabilidades de segurança mais críticas em aplicações WEB e noutros tipos de aplicações.

A criação de serviços Web seguros exige uma análise cuidadosa das ameaças e ataques possíveis a partir do instante em que o serviço Web é publicado. Em seguida, cada ameaça ou ataque identificado, necessita de um estudo para aplicar mecanismos de segurança com o objectivo de eliminar ou reduzir possíveis pontos de insegurança passíveis de serem explorados pelos invasores (WATHIER, 2004).

Dependendo do ponto e do modo em que o invasor poderá atacar pode existir um tipo de vulnerabilidade, onde esta pode surgir no consumidor do serviço Web, no próprio serviço Web ou mesmo em vários pontos do caminho entre o consumidor e o serviço Web.

Para cada tipo de ameaça ou ataque deverá haver uma contramedida para resolver a questão da insegurança (MICROSOFT, 2005).

Segundo OWAST(2007) ataques de Buffer Overflow, SQL Injection, DoS, Cross Site Scripting, Broken authentication and session management, entre outros estão no top 10 de vulnerabilidades de segurança mais críticas em aplicações WEB. Abaixo serão abordados alguns dos ataques mais críticos segundo OWAST(2007) e mecanismos de prevenção a estes.

2.5.2.1 Buffer Overflow

Buffer overflow é provavelmente a vulnerabilidade de segurança mais comum explorada. O ataque de Buffer overflow acontece quando um programa ou processo tenta armazenar mais dados no buffer (área temporária de armazenamento) do que era previsto armazenar (SEARCHSECURITY, 2007).

Este tipo de falhas acontece quando não é feita uma revisão adequada dos dados de input numa determinada aplicação e ocorre comummente em linguagens de programação que não efectuam verificação de limite ou alteração dinâmica do tamanho do buffer.

Uma das formas de prevenir-se deste tipo de ataques é usando plataformas de programação modernas. Um dos exemplos seria a linguagem de programação Java que interna-

mente verifica todos acessos a arrays para detectar violação de limites (KANNEGANT, 2008).

Estas verificações certamente causam uma degradação de desempenho, mas mesmo assim muitas aplicações não se importam com isso. As aplicações que a violação de limites não é feita automaticamente costumam ter alto desempenho, mas a única forma de evitar este tipo de situação é aplicando praticas fortes de programação, procurando usar bibliotecas que ajudam a mitigar este tipo de ataques.

Supondo que determinada aplicação possa sofrer algum ataque de buffer overflow e deseja minimizar os efeitos do ataque, é recomendado que nunca conceda privilégios mais do que esta realmente precisa de modo a reduzir os danos que o ataque possa causar (KANNEGANT, 2008).

2.5.2.2 SQL Injection

A maioria dos serviços de qualquer organização tem suporte de uma base de dados. Se determinado serviço executa uma consulta SQL fornecida por um requisitante sem validação desta, corre-se um risco de um ataque de SQL Injection.

SQL Injection ocorre quando o atacante consegue inserir uma série de instruções SQL dentro de uma consulta através da manipulação das entrada de dados de uma aplicação (KANNEGANT, 2008).

Tomemos como exemplo a seguinte consulta:

"SELECT balancoConta FROM contas WHERE IdConta = '"+IdConta+"';"

O implementador do serviço espera que o requisitante passe o IdConta da seguinte forma:

<balancoConta>
<IdConta>18234</IdConta>
</balancoConta>

Onde *18234* é o valor de IdConta. Um requisitante malicioso pode atacar a aplicação e tentar submeter o valor de IdConta da seguinte maneira:

<balancoConta>
<IdConta>
18234';
UPDATE TABLE contas SET balancoConta=10000000 WHERE IdConta='18234';
SELECT balancoConta FROM contas WHERE IdConta = '18234';
</IdConta>
</balancoConta>

Se o implementador do serviço nao efectua uma verificação do valor do parâmetro IdConta fornecido pelo requisitante, ou seja, usa o valor do parâmetro IdConta exactamente como foi passado, a consulta SQL irá consistir de 3 elementos SQL seguintes:

- *SELECT balancoConta FROM contas WHERE IdConta = '18234';*
- *UPDATE TABLE contas SET balancoConta=10000000 WHERE IdConta='18234';*
- *SELECT balancoConta FROM contas WHERE IdConta = '18234';*

Como pode notar, o atacante pode actualizar o balanço da conta de alguém para dez milhões de meticais por exemplo. Os ataques de SQL Injection tiram vantagem de aplicações que não validam dados de input antes de usá-los. Para evitar este ataque é necessário que sempre sejam validados os dados de input enviados antes de usá-los.

2.5.2.3 Distributed Denial of Service (DDoS)

Ataques DoS (Denial of Service) ocorrem através do envio indiscriminado de requisições a um computador alvo, e visam causar a indisponibilidade dos serviços oferecidos por ele. Onde indisponibilizar pode significar retirar totalmente o computador alvo de operação ou apenas deixa-lo lento, ao ponto do cliente abandonar o serviço devido ao tempo de resposta.

Fazendo uma analogia simples, DoS é o que ocorre com as companhias de telefonia móvel nas noites de natal e ano novo, quando milhares de pessoas comunicam-se simultaneamente.

Ao longo do último ano, uma categoria de ataques de rede tem-se tornado bastante conhecida: a intrusão distribuída. Neste novo enfoque, os ataques não são baseados no uso de um único computador para iniciar um ataque, no lugar são utilizados centenas ou até

milhares de computadores desprotegidos e ligados na Internet para lançar coordenadamente o ataque (SOLHA, 2008).

Os ataques DDoS pode ser considerada como conjugação entre DoS e intrusão distribuída, ou seja, ataques DDoS podem ser definidos como ataques DoS diferentes partindo de várias origens, disparados simultânea e coordenadamente sobre um ou mais alvos.

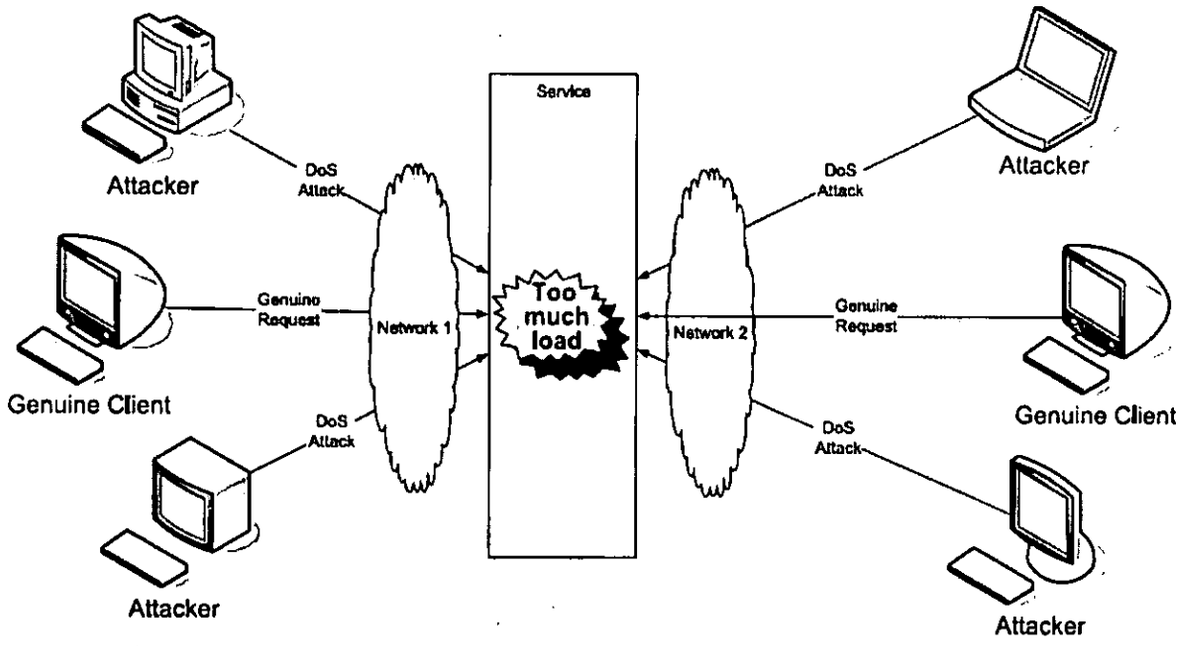


Figura. 2.11: Ataque DDoS saturando um serviço através de pedidos de um número elevado de clientes (KANNEGANT, 2008).

De modo a cobrir este tipo de ataques, é necessário que aplicações tenham capacidade de rapidamente distinguirem entre pedidos legítimos e ilegítimos filtrando o tráfego de atacantes o mais rápido possível de modo a eliminar o impacto deste tipo de ataque.

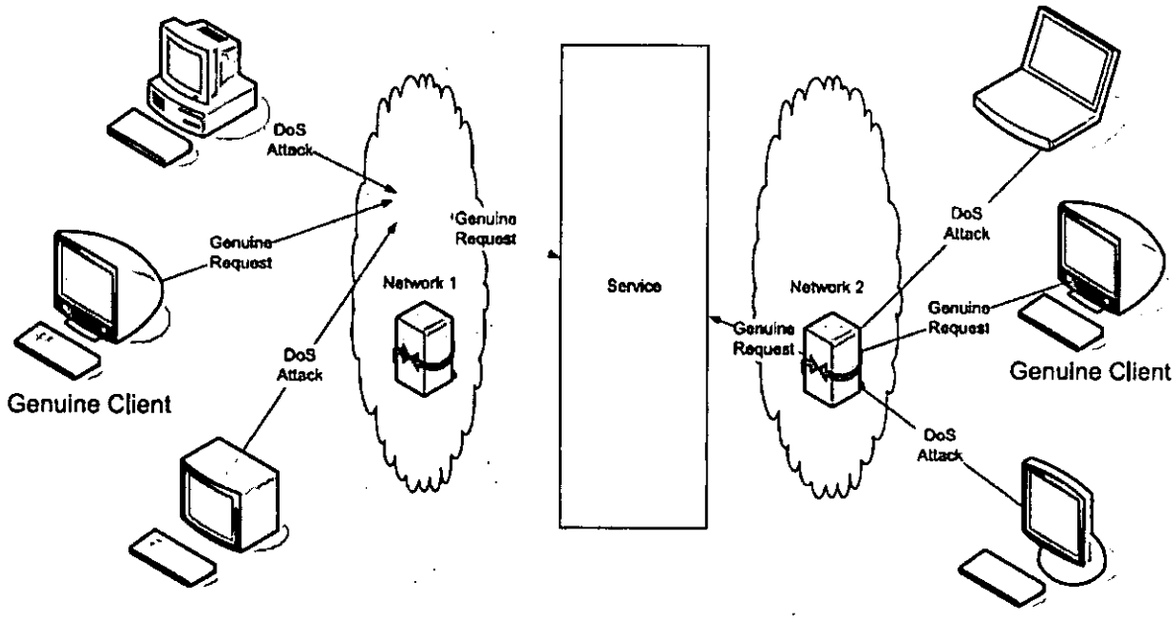


Figura. 2.12: Estratégia de prevenção contra ataques DDoS (KANNEGANT, 2008).

CAPÍTULO

3

CASO DE ESTUDO

3.1 Sistema actual

A Direcção dos Recursos Humanos (DRH) possui um sistema informatizado para gestão de Recursos Humanos (RH) constituído por vários módulos, incluindo o modulo de gestão dos funcionários da UEM no quadro de formação.

Após os funcionários retornarem da sua formação às respectivas faculdades, estas devem enviar à Direcção de Recursos Humanos, o relatório da formação do funcionário de modo que a DRH actualize os dados de progresso do funcionário no sistema actual, podendo haver possibilidade de omissão de alguns dados relevantes no processo de actualização dos dados visto que neste sistema, não há rigorosidade de preenchimento de campos indispensáveis para uma gestão e monitoria eficiente.

Porém, o processo de envio dos relatórios de formação por parte das faculdades e actualização dos dados relativos aos funcionários no quadro de formação por parte do DRH não funciona devidamente, visto que as faculdades não são regulares no envio dos dados e por outro lado, a DRH não dispõe de meios para monitoria eficiente e que rapidamente possam alertar acerca dos funcionários cuja data de finalização da formação prevista esteja próxima ou tenha sido ultrapassada.

Certas entidades como o Gabinete de planificação da UEM, Ministério de Ciências e Tecnologias, Direcção Científica, Ministério de Educação e Cultura, Registo Académico, entre outros, regularmente necessitam de relatórios de dados estatísticas relativos aos funcionários da UEM no quadro de formação, em que para tal efeito, estes enviam cartas

que geralmente incluem detalhes dos campos que precisam, prazo de resposta a carta, e por vezes um modelo específico a partir do qual o relatório deve ser enviado ao Gabinete de Planificação (GP) da Universidade Eduardo Mondlane. O GP por sua vez encaminha estes pedidos ou solicitações a DRH para processamento e elaboração dos relatórios solicitados.

Uma vez recebida a carta encaminhada pelo Gabinete de Planificação, a DRH analisa o tipo de dados a buscar e posteriormente prepara os dados necessários para efectuar os cálculos estatísticos por reflectir nos relatórios no sistema actual. A preparação dos dados e feita a partir filtragens a partir de um ou dois campos no máximo (devido a limitação do sistema actual) de alguns dados ou instâncias de cada funcionário existentes no sistema informatizado actual.

O resultado da preparação inicial dos dados e por volta de centenas de instâncias que posteriormente são passados para uma folha de cálculos MS-Excel, e a partir daí efectuar algumas transformações manuais, ou seja, continuar com o processo de filtragem das centenas de instâncias uma por uma de modo a reunir os requisitos reais para efectuação dos cálculos estatísticos para elaboração do relatório. Devido a quantidade maciça dos dados por filtrar manualmente, há maior probabilidade de ocorrência de redundâncias, inconsistências e morosidade na elaboração do relatório podendo comprometer o prazo de elaboração, disponibilização dos relatórios bem como a própria veracidade destes. Posteriormente os resultados obtidos são agrupados e se necessário mapeados sob o modelo proposto pelo orgão requisitante e encaminhados ao Gabinete de Planificação, que por sua vez opcionalmente informa ao orgão solicitante da disponibilidade do relatório ou aguarda até que o solicitante venha levantar o relatório que requisitou.

3.2 Sistema Proposto (Protótipo)

O protótipo está subdividido em 2 partes:

1ª parte:

É composta pelo Serviço Web e pela Aplicação Desktop desenvolvidos e implementados em Java, onde:

- O Serviço Web implementado estará alojado no servidor de aplicações Glassfish[1] V2.

[1]https://glassfish.dev.java.net/.

O Serviço Web sera responsável pela autenticação e autorização dos consumidores dos serviços, bem como pela geração e envio dos dados estatísticos após passagem de parâmetros para modelação dinâmica dos dados.

- A Aplicação Desktop servirá basicamente para demonstração do processo de filtragem e monitoria dos funcionários no quadro de formação a partir de alertas gerados com base nos prazos de finalização das formações por parte dos CDIs e CTAs. A tela que mostra o processo de filtragem e monitoria poderá ser encontrada no apêndice A.

2ª parte:

É composta por uma aplicação cliente desenvolvida em Visual Basic 2008 sob o sistema operativo Windows XP, que servirá para demonstrar a interoperabilidade e o mecanismo em que os parceiros da UEM poderam consultar os planos de formação de uma forma rápida, eficiente e segura. Foi usado o mecanismo de autenticação mútua através da troca de certificados digitais X509-v3 de modo a garantir autenticidade, confidencialidade, integridade e não repúdio entre as partes envolvidas, sendo estes tópicos discutidos no sub-capítulo 2.5.1. A tela que mostra o resultado do consumo do Serviço Web (consulta de dados estatísticos dos planos de formação) por esta aplicação poderá ser encontrada no apêndice B, entretanto, no apêndice C estão listados os dados que trafegaram entre esta e o serviço Web no processo de requisição e resposta dos dados estatísticos.

3.3 Ferramentas usadas

Para implementar o protótipo, utilizou-se a IDE (Integrated Development Enviroment) Netbeans[2] 6.5 para implementação do web service em java e desenvolvimento da aplicação desktop em java para suporte a monitoria, utilizou-se Visual Studio[3] 2008 para desenvolvimento da aplicação Desktop para consumo do web service. Utilizou-se também o servidor de aplicações Glassfish compatível com as especificações J2EE e WS-Security, viabilizando a implementação desejada, bem como Keytool[4] e Openssl[5] para criação de certificados digitais e suas chaves privadas e publicas respectivamente.

[2]http://www.netbeans.org
[3]http://www.microsoft.com/emea/msdn/visualstudio/
[4]http://java.sun.com/j2se/1.3/docs/tooldocs/win32/keytool.html
[5]http://www.openssl.org

3.4 Protótipo do Sistema proposto

Devido a natureza do trabalho (protótipo), preferiu-se não abordar-se com incidência acerca da modelação do sistema, por exemplo, apresentação do leque de diagramas de UML cruciais (Classes, Sequência de Eventos, Estado, entre outros), visto que o enfoque do presente trabalho é segurança em SOA adoptando alguns padrões e tecnologias de implementação desta. Contudo, será ilustrado somente um diagrama de casos de uso de modo a perceber-se quais os possíveis actores e suas respectivas iterações com o sistema.

3.4.1 Diagrama de Casos de Uso

O Diagrama de casos de uso desenhado é um conjunto de cenários que descrevem a iteração entre o usuário e o sistema, mostrando a relação existente entre os casos de uso e os actores permitindo que se elucide as funcionalidades do sistema. Entretanto, o cenário da figura 3.1 servirá de base para as funcionalidades do protótipo do sistema.

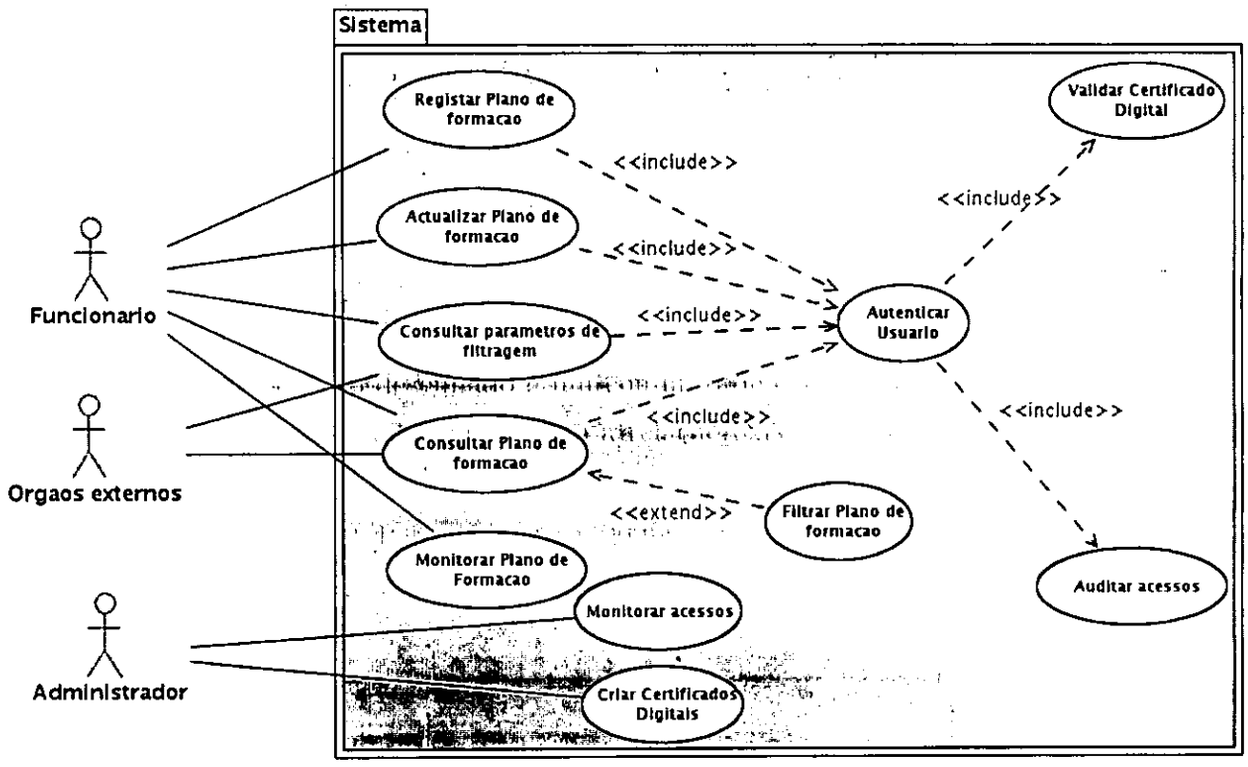


Figura. 3.1: Diagrama de casos de uso.

3.4.1.1 Descrição dos actores

Administrador do Sistema

O Administrador do Sistema é responsável pela criação de certificados digitais de modo a possibilitar o consumo dos Web Services que o sistema dispõe e monitoria dos acessos por parte dos consumidores dos Web Services.

Funcionário

Este é responsável pela gestão (adição, actualização, consulta) e monitoria dos planos de formação.

Orgãos externos

Estes actores representam instituições parceiras da UEM, com certificados de acesso ao sistema oferecidos pela UEM.

CAPÍTULO

4

CONCLUSÕES E RECOMENDAÇÕES

4.1 CONCLUSÕES

Este trabalho teve como meta o levantamento das principais características de Arquitectura Orientada a Serviços, os desafios de segurança que advém da adopção desta e com base num caso de estudo desenvolver um protótipo com base em padrões e tecnologias de implementação e de segurança em SOA.

Arquitecturas tradicionais se tornaram mais complexas, caras, difíceis de gerir e desafiadoras nos aspectos de integração, interoperabilidade e segurança. Como alternativa, surge a arquitectura SOA que promete reduzir custos integrando plataformas heterogéneas e reutilizando funcionalidades já existentes, proporcionando assim agilidade na melhoria e ou desenvolvimento de novas aplicações.

O desenvolvimento de soluções SOA com extensão a segurança não é fácil. É necessário combinar algumas ideias, a partir das abordadas no presente trabalho até as ideias abordadas em livros, artigos, etc. com o ambiente a que se encontra de modo a criar soluções que satisfaçam as necessidades do ecossistema de negócios a que se encontra.

As plataformas, documentação, entre outros recursos para implementação de uma SOA com base nos padrões de segurança WS-* ainda são relativamente escassos, estando a maioria disponíveis para as principais e mais conhecidas plataformas de desenvolvimento existentes actualmente no mercado como Java e .NET. E a escassez destes recursos condiciona a interoperabilidade entre várias linguagens de programação ou plataformas usando os padrões de segurança WS-*, contudo o suporte a estes padrões esta em constante evo-

lução.

Mesmo no desenvolvimento de uma SOA, e necessário levar em consideração os aspectos de segurança a nível da aplicação e as possíveis vulnerabilidades comuns como o vazamento de informações, tratamento de erros inapropriado, SQL Injection, Buffer Overflow, DoS, entre outros, visto que a tecnologia por si só não é capaz de eliminar tais ameaças.

Com base num desenvolvimento contínuo do protótipo pode se esperar que se obtenham bons resultados como a redução dos processos burocráticos e tempo de resposta (de dias para segundos) para obtenção de dados estatísticos, facilidade de monitoria por parte da DRH e contudo, aumento de produtividade por parte de todos que compõem este ciclo.

4.2 RECOMENDAÇÕES

- Invistir em infra-estruturas de integração "multi-protocolo", tais como Enterprise Service Bus (ESB);
- Complementar o estudo efectuado com base noutros padrões de implementação de uma SOA mais segura e flexível não abordados no presente trabalho tais como: WS-Trust, WS-Security Policy, SAML, WS-Federation, entre outros;
- Estudar mais ataques através de vulnerabilidades em aplicações observando seu comportamento em situações de excepção, não limitando-se apenas ao uso de ferramentas ou tecnologias para garantir a segurança, exigindo desta forma que esta seja encarada como um processo de mitigação de riscos;
- Efectuar uma implementação gradual de SOA na DRH de modo a interoperar o sistema de gestao de recursos humanos com os demais sistemas da UEM;
- Criar um centro de competência em integração e segurança como uma "torre de controle" das iniciativas relacionadas a SOA na UEM.

CAPÍTULO

5

REFERÊNCIAS BIBLIOGRÁFICAS

1. BIANCHI, C., **Método para análise, especificação, desenvolvimento e governança de uma SOA**, Dissertação de Mestrado, Universidade de São Paulo, SP, 2007.

2. CABREIRA, L. C. Q., **Administração de Pessoal, Gerência de Recursos Humanos e Gestão Estratégica**, disponível em: <http://www.ResumosConcursos.hpg.com.br>, acessado em 16/07/2008.

3. CISCO, **XML Web Services and SOA Glossary**, 2005, disponível em: <http://www.cisco.com/cdc_content_elements/acquisitions/reactivity/soa/glossary.html>, acessado em 11/10/2008.

4. DEGAN, J. O. C., **Integração de dados corporativos : uma proposta de arquitectura baseada em serviços de dados**, disponível em: <http://libdigi.unicamp.br/document/?code=vtls000378466>, acessado em 01 de Setembro de 2008.

5. GARTNER GROUP, disponível em: <http://www.gartner.com>, acessado em 12/06/2008.

6. IBM, **Getting Started with SOA on System i5**, 2006, disponível em: <http://www.ibm.com/ru/events/presentations/systemi5/WebServicesDeployment.pdf>, acessado em 25/11/2008.

7. IBM, **Web Services Security (WS-Security)**, 2003, disponível em: <http://www-128.ibm.com/developerworks/webservices/library/ws-secure/index.html>, acessado em 17/04/2008.

8. INDEPENDENT EVALUTION GROUP, **What is Monitoring and Evaluation (M&E)?**, disponível em <http://www.worldbank.org/ieg/ecd/what_is_me.html>, acessado em 13/09/2008.

9. KANNEGANT, R., CHODAVARAPU, P., **SOA Security**, 2008, Manning.

10. KOBIELUS, J., **SERVICE ORIENTED ARCHITECTURE: Developing the enterprise Roadmap**, 2004,.

11. MICROSOFT. **Módulo 10 – Segurança de Web Services.**, 2005, disponível em: <http://www.microsoft.com/brasil/security/guidance/topics/devsec/secmod10.mspx>, acessado em 14/12/2008.

12. OWASP, **As 10 vulnerabilidades de segurança mais críticas em aplicações WEB**, 2007, disponível em: www.owasp.org/images/4/42/OWASP_TOP_10_2007_PT-BR.pdf, acessado em 18/12/2008.

13. PAULA, A. C., **Rede organizacional - uma estratégia de crescimento?**, 2004, disponível em: <http://gpo.com.br/tese/metodologia.htm>, acessado em 12 de Maio de 2008.

14. ROCHA, C. A., **Um estudo sobre os desafios de Segurança na adoção da Arquitectura Orientada a Serviços**, Trabalho final de Mestrado, Universidade Estatal de Campinas, SP, 2007.

15. ROSENBERG, J., REMY, D. L., **Securing Web Services with WS-Security**, 2004, Sams Publishing.

16. SEARCHSECURITY, **Buffer Overflow**, 2007, disponível em <http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci549024,00.html>, acessado em 16/12/2008.

17. SIDDIQUI, B., **Exploring XML Encryption**, 2002, disponível em: <http://www-128.ibm.com/developerworks/xml/library/x-encrypt/#code1>, acessado em 14/10/2008.

18. SIMON, E., MADSEN, P., ADAMS, C., **An Introduction to XML Digital Signatures**, 2001, disponível em <http://www.xml.com/pub/a/2001/08/08/xmldsig.html>, acessado em 15/08/2008.

19. SOLHA, L. E. V. A., TEIXEIRA, R. C., PICCOLINI, J. D. B., **Ataque DDoS**, 2008, disponível em <http://www.oficinadanet.com.br/artigo/1026/ataque_ddos>, consultado em 17/01/2009.

20. WAGNER, R., **WS-I Releases Basic Security Profile Version 1.0 Working Group Draft**, 2004, disponível em: <http://xml.coverpages.org/ni2004-05-18-a.html>, acessado em 13/06/2008.

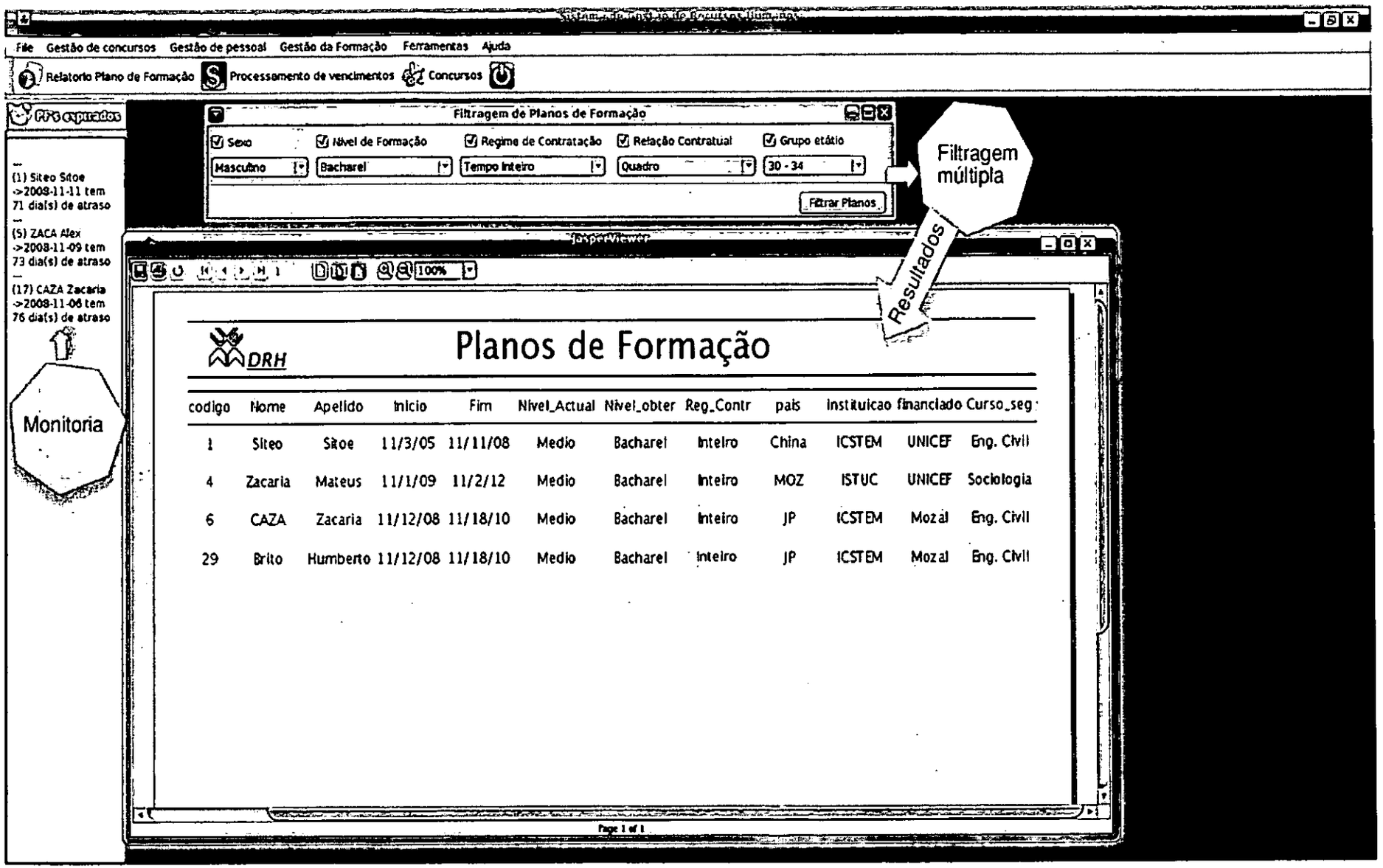
21. WATHIER, A. J., **Segurança em Web Services com WS-Security**, 2004, Trabalho de Especialização, Uiversidade Federal do Rio Grande do Sul, Porto Alegre, 2004.

22. WILNER, K., **Especialista desvenda segredos da SOA**, 2006, Entrevista concedida a Nivaldo Foresti.

APÊNDICE

A

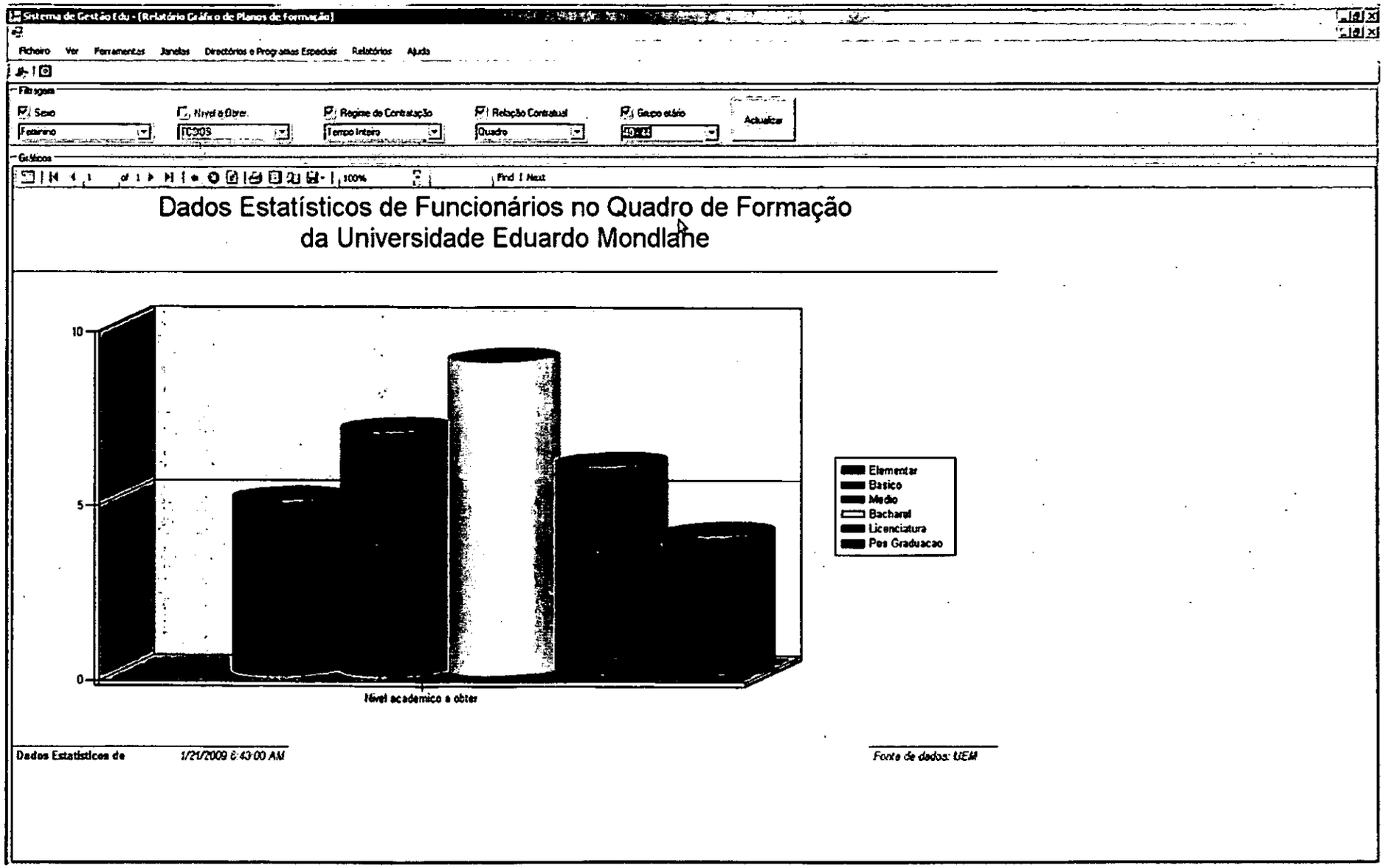
Processo de Monitoria e Filtragem


File Gestão de concursos Gestão de pessoal Gestão da Formação Ferramentas Ajuda
Relatorio Plano de Formação
Processamento de vencimentos
Concursos
RH's expirados
(1) Siteo Sitoe
->2008-11-11 tem
71 dia(s) de atraso
(5) ZACA Alex
->2008-11-09 tem
73 dia(s) de atraso
(17) CAZA Zacaria
->2008-11-06 tem
76 dia(s) de atraso
Monitoria
Filtragem de Planos de Formação
Sexo
Nível de Formação
Regime de Contratação
Relação Contratual
Grupo etário
Masculino
Bacharel
Tempo Inteiro
Quadro
30 - 34
Filtrar Planos
Filtragem múltipla
JasperViewer
100%
Resultados
DRH
Planos de Formação
codigo Nome Apelido Inicio Fim Nivel_Actual Nivel_obter Reg_Contr pais instituicao financiado Curso_seg
1 Siteo Sitoe 11/3/05 11/11/08 Medio Bacharel Inteiro China ICSTEM UNICEF Eng. Civil
4 Zacaria Mateus 11/1/09 11/2/12 Medio Bacharel Inteiro MOZ ISTUC UNICEF Sociologia
6 CAZA Zacaria 11/12/08 11/18/10 Medio Bacharel Inteiro JP ICSTEM Mozal Eng. Civil
29 Brito Humberto 11/12/08 11/18/10 Medio Bacharel Inteiro JP ICSTEM Mozal Eng. Civil

APÊNDICE

B

Consumo do Serviço Web pelo cliente


Sistema de Gestão Edu - [Relatório Gráfico de Planos de Formação]
Ficheiro Ver Ferramentas Janelas Directórios e Programas Especiais Relatórios Ajuda
Filtragem
Sexo
Feminino
Nível a Obter
TC203
Regime de Contratação
Tempo Inteiro
Relação Contratual
Quadro
Grupo etário
Actualizar
Gráficos
100%
Find | Next
Dados Estatísticos de Funcionários no Quadro de Formação
da Universidade Eduardo Mondlane
10
5
0
Nível academico a obter
Elementar
Basico
Medio
Bacharel
Licenciatura
Pos Graduacao
Dados Estatisticos de
1/21/2009 8:43:00 AM
Fonte de dados: UEM

APÊNDICE

C

Dados no padrão WS-Security

A listagem abaixo é de mensagens SOAP usando o padrão WS-Security, capturada em tempo de execução da aplicação cliente e o serviço Web.
A mensagem da página 62-65 é uma mensagem de requisição do serviço Web, enquanto que a mensagem da página 66-68 é uma mensagem de resposta a requisição.

```
<?xml version="1.0" ?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
 <s:Header>
  <a:Action s:mustUnderstand="1"
u:Id="_3">http://tese.jp/buscarPlano/buscarTodosRequest</a:Action>
  <a:MessageID u:Id="_4">urn:uuid:bba12429-d14b-4d5c-a6d8-d4da53286c0f</a:MessageID>
  <a:ReplyTo u:Id="_5">
   <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
  </a:ReplyTo>
  <a:To s:mustUnderstand="1"
u:Id="_6">http://localhost:8080/ws_smede/buscarPlanoService</a:To>
  <o:Security xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" s:mustUnderstand="1">
   <u:Timestamp u:Id="uuid-899834bd-aff1-4f6d-8819-9ef421272b62-1">
    <u:Created>2009-01-21T20:44:06.104Z</u:Created>
    <u:Expires>2009-01-21T20:49:06.104Z</u:Expires>
   </u:Timestamp>
   <o:BinarySecurityToken u:Id="uuid-726aa2a3-0d11-4b2f-8156-b9d530dae9e5-2"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-
profile-1.0#X509v3" EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
soap-message-
security-1.0#Base64Binary">MIIDxjCCAy+gAwIBAgIJAO5kOtyiFVM7MA0GCSqGSIb3DQEB
BQUAMIGjMQswCQYDVQQGEwJNWjEPMA0GA1UECBMGTWFwdXRvMQ8wDQYDVQQ
HEwZNYXRvbGExJjAkBgNVBAoTHVVuaXZlcnNpZGFkZSBFZHVhcmRvIE1vbmRsYW5lM
QwwCgYDVQQLEwNETUkxGTAXBgNVBAMTEEpvbm5hdGhhbiBHdWFtYmUxITAfBgkqhki
G9w0BCQEWEmdobGVnc0Bob3RtYWlsLmNvbTAeFw0wODExMDIxMjA3MjZaFw0wOTExM
DIxMjA3MjZaMIGVMQswCQYDVQQGEwJHQjESMBAGA1UECBMJU2hlZmZpZWxkMRIw
EAYDVQQHEwlCZWF1Y2hpZWYxIzAhBgNVBAoTGkVsZWN0cm9uaWMgRGF0YSBQcm9j
ZXNzaW5nMSMwIQYDVQQLExpFbGVjdHJvbmljIERhdGEgUHJvY2Vzc2luZzEUMBIGA1UE
AwwLY2xpZW50X2NlcnQwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAK6hhLF+H
Dtc5LlwE7Nf2tkHuzrnarV4QPbwR4BPqCOCoCC2ZSLX6ci0ppAwHHXp/M7kPTqacxBvY3Cyx
UBTIc6OUr29m6Kf5ynfbnL4602ns18gRTQah7jDQWPC6WzF26/a1f0WlZ3wyiqHDQBpCCaEG
vsHHUTAZog97VGxyqTdAgMBAAGjggEMMIIBCDAdBgNVHQ4EFgQUc8/4wGzH2HuPNKJf
nuHBX4q5Z38wgdgGA1UdIwSB0DCBzYAUZJmmx//yBbraW8ZGZQ/ZlX9uD+WhgamkgaYwg
aMxCzAJBgNVBAYTAk1aMQ8wDQYDVQQIEwZNYXB1dG8xDzANBgNVBAcTBk1hdG9sY
TEmMCQGA1UEChMdVW5pdmVyc2lkYWRlIEVkdWFyZG8gTW9uZGxhbmUxDDAKBgNVB
AsTA0RNSTEZMBcGA1UEAxMQSm9ubmF0aGFuIEd1YW1iZTEhMB8GCSqGSIb3DQEJARY
SZ2hsZWdzQGhvdG1haWwuY29tggkAyAmS8H2Wg2AwDAYDVR0TBAUwAwEB/zANBgkqhk
iG9w0BAQUFAAOBgQB2BA3f3/SsJ4GkdvpmT2VSiFPwukd/DjCsZAenj4ud8QD+tiecCwBtyak
BLzg5DPqIW7iFyN/JRhhCfbjfVNdgEZkZPUun/3izpJ7NiKWxKCMEv8BW9hF6iJhkl/qt7NU9Js
4w20/PEnMY8eJJtCJ+kRKPO4kkjsJ5rO0PXhBsbQ==</o:BinarySecurityToken>
   <e:EncryptedKey xmlns:e="http://www.w3.org/2001/04/xmlenc#" Id="_0">
    <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p">
     <DigestMethod xmlns="http://www.w3.org/2000/09/xmldsig#"
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
    </e:EncryptionMethod>
```


```
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
 <o:SecurityTokenReference>
  <o:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
x509-token-profile-1.0#X509SubjectKeyIdentifier" EncodingType="http://docs.oasis-open.org/wss/
2004/01/oasis-200401-wss-soap-message-
security-1.0#Base64Binary">b+HhvkxYzLyZery+oteRyVZLmkU=</o:KeyIdentifier>
 </o:SecurityTokenReference>
</KeyInfo>
<e:CipherData>
 <e:CipherValue>dnimG31frDkTUHCB5/31qGVhVq6pMOqeTPVBEhN1P5zYHiDqBuQ4y
PT/k9lpyMrRnLlCQFax4JgV+az4LnXLK6URige0CKCOtYZ17G0Nd1TSRHBjlk+7+ZZVCmz3N
E2bR3nBPH4CbaCXwYdas/BxhZHJskdCDogyBF966SsbV48=</e:CipherValue>
</e:CipherData>
<e:ReferenceList>
 <e:DataReference URI="#_2"></e:DataReference>
</e:ReferenceList>
</e:EncryptedKey>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
 <SignedInfo>
  <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"></CanonicalizationMethod>
  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"></SignatureMethod>
  <Reference URI="#_1">
   <Transforms>
    <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"></Transform>
   </Transforms>
   <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
   <DigestValue>1AufdE0V953JdsZzvQGXloMBMQk=</DigestValue>
  </Reference>
  <Reference URI="#_3">
   <Transforms>
    <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"></Transform>
   </Transforms>
   <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
   <DigestValue>sfIH78ORoF2k6khtcqXbXyqvaVM=</DigestValue>
  </Reference>
  <Reference URI="#_4">
   <Transforms>
    <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"></Transform>
   </Transforms>
   <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
   <DigestValue>OGXVbn5O3SKAOjd6Rqf6rk/54Lc=</DigestValue>
  </Reference>
  <Reference URI="#_5">
   <Transforms>
    <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"></Transform>
   </Transforms>
```

```
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
        <DigestValue>k69pykploFPkXhw5ogDHcjcJUI0=</DigestValue>
       </Reference>
       <Reference URI="#_6">
        <Transforms>
         <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"></Transform>
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
        <DigestValue>qtAPAeBxzw3ni1q2MgWRA+XKMQI=</DigestValue>
       </Reference>
       <Reference URI="#uuid-899834bd-aff1-4f6d-8819-9ef421272b62-1">
        <Transforms>
         <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"></Transform>
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
        <DigestValue>n1mud6g3osLKClqY8OYtNE7dWlQ=</DigestValue>
       </Reference>
      </SignedInfo>
      <SignatureValue>lUPwxX4UCzLFJCJEK+T+NVWRCIz7lSIha8wwwVHHaTp+M8jD6z9TH
Oy7D0iNqPwCo5fcjg/kyTkQ3tqhRgFNkuYaOfiphHR6bpJJx5Y3F2tX7IoiNLchJrtlU6b5JT/YdoE
mDSffzPER3vJwpPLcrozokyihPycE8kLeOnfgZrM=</SignatureValue>
      <KeyInfo>
       <o:SecurityTokenReference>
        <o:Reference ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-
token-profile-1.0#X509v3" URI="#uuid-726aa2a3-0d11-4b2f-8156-
b9d530dae9e5-2"></o:Reference>
       </o:SecurityTokenReference>
      </KeyInfo>
     </Signature>
    </o:Security>
   </s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" u:Id="_1">
    <e:EncryptedData xmlns:e="http://www.w3.org/2001/04/xmlenc#" Id="_2"
Type="http://www.w3.org/2001/04/xmlenc#Content">
     <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-
cbc"></e:EncryptionMethod>
     <e:CipherData>
      <e:CipherValue>vMU8PklG1LU+zB8NokXP9Q+D4lcJy0jrHHDgNEVC0FTCCpl9Hhgpizx6l
x9p5qxTYMpbSNZFZC9hZ1cdk6tWJs3jxvzxbWH0KJCTgRws5K3EV8TRe03Dem0JzlfK62Rbd
AUr5jfR1QcNBXRDsyRP7KtBggg6YCQ581LVgl9YshIGTwcOHEdbvI8jW3VWAcqF3lgf2iRlJL
MH7nOpG1M7HFiGI6c4+
+lF3UvzFKmCNwNa/3uc8KKmYGwE/kLrOEidPjaYpJm9uTsRygVYPchi5wUpeu8TT+iTxoYLsc
TLkAkfPBduMwROI8Gcz3ULVbcQeZtzQ56ataJgeBkdVQ2fnj8U6vcxsyNyW4JqrORmFvtln0DG
RlwE0NgZ1IvMXxVvwP3GR/WY885PD3AKooiOxj8j71MXiwYxk3XZ+1x1AJIBkqeJj6lxHoC4l
fwsl+3Rp9j8fh6K3jYf5TsYFBLMD6RlpjejxH4LD8+XusCqHEU=</e:CipherValue>
     </e:CipherData>
```

```
    </e:EncryptedData>
  </s:Body>
</s:Envelope>
```

```
<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:exc14n="http://www.w3.org/2001/10/xml-exc-c14n#" xmlns:xenc="http://www.w3.org/
2001/04/xmlenc#">
 <S:Header>
  <To xmlns="http://www.w3.org/2005/08/addressing"
wsu:Id="_5005">http://www.w3.org/2005/08/addressing/anonymous</To>
  <Action xmlns="http://www.w3.org/2005/08/addressing"
wsu:Id="_5003">http://tese.jp/buscarPlano/buscarTodosResponse</Action>
  <MessageID xmlns="http://www.w3.org/2005/08/addressing" wsu:Id="_5002">uuid:8d8b6ed6-
d26b-49cc-85b0-1827d41aba0c</MessageID>
  <RelatesTo xmlns="http://www.w3.org/2005/08/addressing"
wsu:Id="_5004">urn:uuid:bba12429-d14b-4d5c-a6d8-d4da53286c0f</RelatesTo>
  <wsse:Security S:mustUnderstand="1">
   <wsu:Timestamp xmlns:ns17="http://docs.oasis-open.org/ws-sx/ws-
secureconversation/200512" xmlns:ns16="http://www.w3.org/2003/05/soap-envelope" xmlns=""
wsu:Id="_3">
    <wsu:Created>2009-01-21T20:44:13Z</wsu:Created>
    <wsu:Expires>2009-01-21T20:49:13Z</wsu:Expires>
   </wsu:Timestamp>
   <xenc:EncryptedKey xmlns:ns17="http://docs.oasis-open.org/ws-sx/ws-
secureconversation/200512" xmlns:ns16="http://www.w3.org/2003/05/soap-envelope" xmlns=""
Id="_5007">
    <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-
mgf1p"></xenc:EncryptionMethod>
    <ds:KeyInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="keyInfo">
     <wsse:SecurityTokenReference>
      <wsse:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
x509-token-profile-1.0#X509SubjectKeyIdentifier" EncodingType="http://docs.oasis-open.org/wss/
2004/01/oasis-200401-wss-soap-message-
security-1.0#Base64Binary">c8/4wGzH2HuPNKJfnuHBX4q5Z38=</wsse:KeyIdentifier>
     </wsse:SecurityTokenReference>
    </ds:KeyInfo>
    <xenc:CipherData>
     <xenc:CipherValue>Grg0cNmM7nZIe9ImoxoCPobKFeC+kvpY/vOtLWGhTVi0LtAWkgyjN
MR/lDdrpIIAzreVLURRf2C/aU9qd0dp2oqMKsGBLt/uBmGGXPj2jcB0qcf4uzCjN1oEtr82LRuB
YOHIgCWfcMYjx2mGQqX5UsTF6oYroAe7HY6yE9+UcR4=</xenc:CipherValue>
    </xenc:CipherData>
    <xenc:ReferenceList>
     <xenc:DataReference URI="#_5008"></xenc:DataReference>
    </xenc:ReferenceList>
   </xenc:EncryptedKey>
   <ds:Signature xmlns:ns17="http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512"
xmlns:ns16="http://www.w3.org/2003/05/soap-envelope" xmlns="" Id="_1">
    <ds:SignedInfo>
```

```
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
          <exc14n:InclusiveNamespaces PrefixList="wsse S"></exc14n:InclusiveNamespaces>
        </ds:CanonicalizationMethod>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"></ds:SignatureMethod>
        <ds:Reference URI="#_5002">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
              <exc14n:InclusiveNamespaces PrefixList="S"></exc14n:InclusiveNamespaces>
            </ds:Transform>
          </ds:Transforms>
          <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></ds:DigestMethod>
          <ds:DigestValue>YnGJYWsfU/qd7JbfcHhYhqvEmC0=</ds:DigestValue>
        </ds:Reference>
        <ds:Reference URI="#_5003">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
              <exc14n:InclusiveNamespaces PrefixList="S"></exc14n:InclusiveNamespaces>
            </ds:Transform>
          </ds:Transforms>
          <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></ds:DigestMethod>
          <ds:DigestValue>YD/1ByPd3qs7vnC10K/pfz5uwqc=</ds:DigestValue>
        </ds:Reference>
        <ds:Reference URI="#_5004">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
              <exc14n:InclusiveNamespaces PrefixList="S"></exc14n:InclusiveNamespaces>
            </ds:Transform>
          </ds:Transforms>
          <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></ds:DigestMethod>
          <ds:DigestValue>dkGybz4gmhA9kpOmIAY/PtsOrc0=</ds:DigestValue>
        </ds:Reference>
        <ds:Reference URI="#_5005">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
              <exc14n:InclusiveNamespaces PrefixList="S"></exc14n:InclusiveNamespaces>
            </ds:Transform>
          </ds:Transforms>
          <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></ds:DigestMethod>
          <ds:DigestValue>Nd/8wVmBdLowQKMblBRYK+6xcjA=</ds:DigestValue>
        </ds:Reference>
        <ds:Reference URI="#_5006">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
```

<exc14n:InclusiveNamespaces PrefixList="S"></exc14n:InclusiveNamespaces>
</ds:Transform>
</ds:Transforms>
<ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></ds:DigestMethod>
<ds:DigestValue>iULV+JtInk4KdlaZT5vnqZOdKGU=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#_3">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
<exc14n:InclusiveNamespaces PrefixList="wsu wsse
S"></exc14n:InclusiveNamespaces>
</ds:Transform>
</ds:Transforms>
<ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></ds:DigestMethod>
<ds:DigestValue>oIesNmg9B93XAKEpytcDMZ0de2c=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>UbfmyORAPjZLEYVCFexEFW+w7cpsKi54lNA363Zmv8PsOpz/f1Xriq
miH0BNgNsmkb9+qHcLTqDSUGXRdytqRDFEEH9m85njZF0B5d2SWF0rRtORZqje5WP369SU
3F1ql8MYMAFp6P/pYYPzIJb8CG3SHUZOWKaw5hrWQA95IVI=</ds:SignatureValue>
<ds:KeyInfo>
<wsse:SecurityTokenReference>
<wsse:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
x509-token-profile-1.0#X509SubjectKeyIdentifier" EncodingType="http://docs.oasis-open.org/wss/
2004/01/oasis-200401-wss-soap-message-
security-1.0#Base64Binary">b+HhvkxYzLyZery+oteRyVZLmkU=</wsse:KeyIdentifier>
</wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
</wsse:Security>
</S:Header>
<S:Body wsu:Id="_5006">
<xenc:EncryptedData xmlns:ns17="http://docs.oasis-open.org/ws-sx/ws-
secureconversation/200512" xmlns:ns16="http://www.w3.org/2003/05/soap-envelope" xmlns=""
Type="http://www.w3.org/2001/04/xmlenc#Content" Id="_5008">
<xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-
cbc"></xenc:EncryptionMethod>
<xenc:CipherData>
<xenc:CipherValue>PhD7Ph06AIbcUQPW17qCB69TrYkfdh6rmXVeIfyKXGLZhTBV6JCd7
5erAzAJMIsOD0cPO2QD9a/3rZMidMfgZjblGe0zImYYS3TUQkqBdJWyws5UB9xAg6ppRbnsqe
GT/0wjI9YA9T1qMETebJo8pNDtRqW0Jtkp5RaT9jFr9OE0Axt3uuG8Zp5HjmCPhhe8bgR3cKbx7
oixV8fJyduFeFsjXiz568InafG2WfdeBl2DDZVeE9q4YkidrSBK9pynudNMFBkrlNb6p+nt/k3p/A=
=</xenc:CipherValue>
</xenc:CipherData>
</xenc:EncryptedData>
</S:Body></S:Envelope>