MIDDLEWARE PARA A COMUNICAÇÃO DE DADOS ENTRE SISTEMAS DISTRIBUÍDOS COM WS SECURITY

CAIO RENAN HOBUS

Orientador: Jhony Alceu Pereira

ROTEIRO

- Introdução
- Objetivos do trabalho
- Fundamentação teórica
- Desenvolvimento
- Requisitos principais
- Especificação
- Implementação
- Técnicas e ferramentas utilizadas
- Operacionalidade da implementação
- Resultados e discussão
- Conclusões
- Extensões

INTRODUÇÃO

- Este trabalho apresenta o desenvolvimento de um middleware com a finalidade de possibilitar as organizações trocarem informações importantes de forma segura através da rede, disponibilizando um serviço que possibilita a integração entre sistemas, na qual abstrai os detalhes da implementação das camadas de comunicação e os detalhes do WS-Security na qual é utilizado para realização da segurança dos dados na comunicação.
- Com o desenvolvimento do middleware, será possível ao usuário configurar regras de acesso que servirão de base para a comunicação de dados.

- O middleware será o responsável por implementar as questões de segurança e enviar os dados através de web services utilizando o WS-Security para a outra aplicação na qual receberá os dados através do middleware, que a entregará novamente a aplicação com base nas regras cadastradas, garantindo a segurança dos dados e a transparência na questão da implementação de todas as funcionalidades para suportar o WS-Security e demais camadas de rede.

OBJETIVOS DO TRABALHO

- Este trabalho tem como objetivo disponibilizar um middleware para as aplicações, que abstrai o WS-Security na comunicação de dados entre as organizações utilizando web services de forma segura e transparente na questão da implementação de todas as funcionalidades para suportar o WS-Security, garantindo a confidencialidade, integridade e autenticidade dos dados.
- Os objetivos específicos do trabalho podem ser detalhados da seguinte maneira:

- disponibilizar uma funcionalidade que permite abstrair o WS-Security, web service e demais camadas de rede;
- disponibilizar uma interface web onde será feito a manutenção das organizações e das regras de comunicação;
- disponibilizar uma interface onde a aplicação solicitará dados com base em regras definidas para o middleware;
- disponibilizar uma interface onde o middleware receberá os dados e retornará a aplicação destinatária.

FUNDAMENTAÇÃO TEÓRICA

- Na fundamentação teórica do trabalho é explicado os conceitos de sistemas distribuídos, middleware, segurança dos dados e WS-Security.

SISTEMAS DISTRIBUÍDOS E SEGURANÇA DOS DADOS

- A segurança dos dados deve estar presente em todas as aplicações, independentemente do seu tipo, que pode ser sistema web, desktop ou móvel, pois torna-se imprescindível para o processo de gerência das informações.

- Com a conscientização das empresas de que suas informações tem um valor fundamental, aumenta cada vez mais o número de tecnologias disponíveis para a implantação dos modelos de segurança dentro dos sistemas. Porém, muitas das técnicas utilizadas são fundamentadas e voltadas para as aplicações comerciais atuais, não visando o novo modelo de sistemas baseados na web, tornando estas aplicações mais vulneráveis no que diz respeito à segurança. Desta forma, torna-se cada vez mais necessário o conhecimento do funcionamento das tecnologias web para que se possa desenvolver novas técnicas de troca segura de informações no ambiente da internet.

- Um dos principais modelos utilizados na troca de informações são os sistemas distribuídos. Atualmente muitos sistemas são implementados de forma distribuída, ou seja, operam utilizando a troca de mensagens via web.
- Os sistemas distribuídos removem as conexões fixas entre aplicações, servidores, bases de dados, máquinas, armazenamento, entre outros, tratando tudo como um serviço virtualizado.
- Quando as organizações trabalham com a troca de informações via web utilizando serviços como web services, que na maioria das vezes são informações sigilosas e valiosas, deve-se analisar e ter estratégia de gerenciamento dos riscos que podem ser causados em relação ao vazamento de informações importantes.

- Segundo Stallings (2005, p. 380), a aplicação desenvolvida deve levar em consideração alguns requisitos, como:
- autenticidade: garantia de que o serviço é capaz de identificar o usuário;
- privacidade: garante que o acesso à informação somente é obtido a quem é autorizado;
- integridade: garantia de que a informação e o meio como é processada não estejam corrompidos;
- disponibilidade: garantia de que as pessoas autorizadas a acessar determinada informação tenham acesso à ela sempre que necessário.

MIDDLEWARE

- Segundo Ikematu et al., (2003), “middleware é uma categoria de produtos ou módulos de software que são utilizados por aplicações cliente, para acessar aplicações servidoras, e tentam esconder da aplicação a rede, a comunicação e plataformas específicas”, ou seja, os middlewares trabalham disponibilizando determinadas funções já desenvolvidas a uma aplicação cliente, abstraindo sua forma de funcionamento, plataformas e tecnologias utilizadas, fazendo com que estas funções não tenham que ser desenvolvidas novamente.

WS-SECURITY

- WS-Security é um conjunto de especificações de segurança para serviços web, proposto em conjunto pela Microsoft Corporation, IBM Corporation e VeriSign, com o objetivo de que as empresas pudessem criar e construir aplicações de serviços web, utilizando web services com uma ampla interoperabilidade
- O WS-Security possui um sistema rico para prover segurança, tanto em termos de confidencialidade, quanto em termos de autenticação/autorização. O sistema de autenticação funciona com diversos mecanismos, sendo os mais conhecidos: autenticação via token Security Assertion Markup Language (SAML), via ticket Kerberos, via fornecimento de usuário e senha (tando com senha em texto puro quanto com hash) e via cerificado X.509.

- O WS-Security funciona em conjunto com as especificações WS-Policy e WS-SecurityPolicy. Estas duas especificações têm por objetivo, respectivamente, estabelecer políticas gerais a respeito de sugurança, Service level Agreement (SLA), qualidade de serviço, confiabilidade, etc.; e estabelecer, especificamente, quais são as políticas de segurança aplicáveis a um determinado serviço.
- Para assinatura da mensagem em nídel de XML é utilizado o padrão XML Digital Signature.
- Já para criptografia da mensagem em nível de XML é utilizado o padrão XML Encryption.

DESENVOLVIMENTO

REQUISITOS PRINCIPAIS

- RF01: O sistema web deverá manter o registro de sistemas.
- RF02: O sistema web deverá manter o registro de regras.
- RF03: O sistema web deverá manter o registro de permissões.
- RF04: O sistema web deverá manter o registro de usuários.
- RF05: O sistema web deverá manter o registro de perfis.
- RF06: O sistema web deverá manter o registro de recursos.
- RF07: O middleware deverá integrar-se as aplicações de terceiros.
- RF08: O middleware deverá enviar requisições para outras aplicações.

- RF09: O middleware deverá receber requisições de outras aplicações.
- RF10: O middleware deverá enviar arquivos para outras aplicações.
- RF11: O middleware deverá receber arquivos de outras aplicações.
- RF12: O web service deverá disponibilizar o serviço ao middleware.
- RF13: O web service deverá autenticar usuários.
- RF14: O web service deverá identificar os sistemas na comunicação.
- RF15: O web service deverá verificar as regras de comunicação.
- RF16: O web service deverá verificar as permissões de cada usuário na comunicação.

- RNF01: O sistema deve ser implementado utilizando web services com a API JAX-WS.
- RNF02: O sistema deve ser implementado utilizando a tecnologia Java.
- RNF03: O sistema deve utilizar banco de dados Oracle 10G.
- RNF04: O sistema deve ser implementado utilizando o ambiente de desenvolvimento NetBeans.
- RNF05: O sistema deve ser implementado utilizando os conceitos definidos pelo conjunto de especificações WS-Security.
- RNF06: O middleware deve garantir a confidencialidade, integridade e disponibilidade dos dados que serão utilizados na comunicação.

ESPECIFICAÇÃO

CASOS DE USO

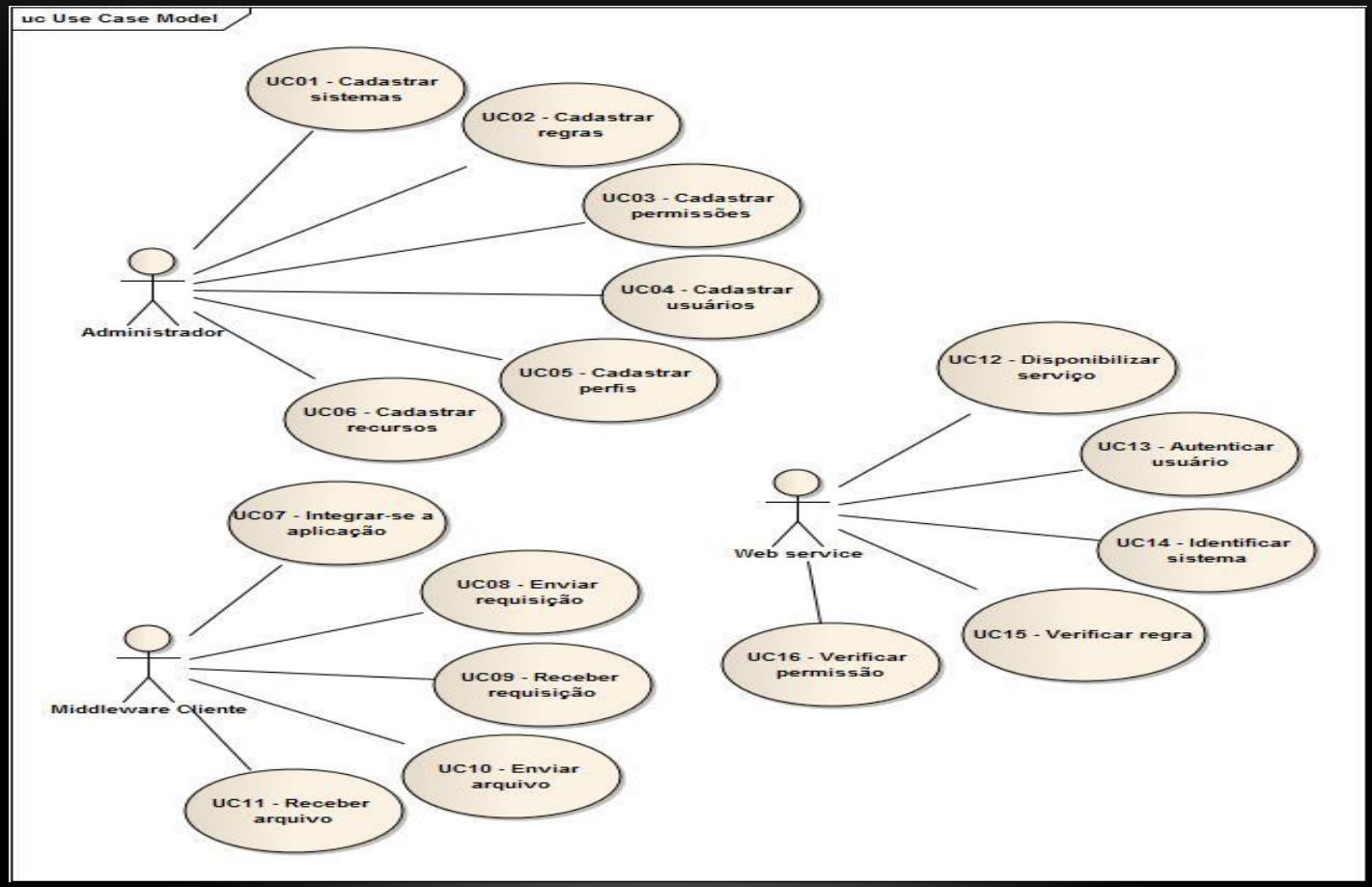

uc Use Case Model
UC01 - Cadastrar sistemas
UC02 - Cadastrar regras
UC03 - Cadastrar permissões
UC04 - Cadastrar usuários
UC05 - Cadastrar perfis
UC06 - Cadastrar recursos
Administrador
UC07 - Integrar-se a aplicação
UC08 - Enviar requisição
UC09 - Receber requisição
UC10 - Enviar arquivo
UC11 - Receber arquivo
Middleware Cliente
UC12 - Disponibilizar serviço
UC13 - Autenticar usuário
UC14 - Identificar sistema
UC15 - Verificar regra
UC16 - Verificar permissão
Web service

DIAGRAMA DE CLASSE MIDDLEWARE

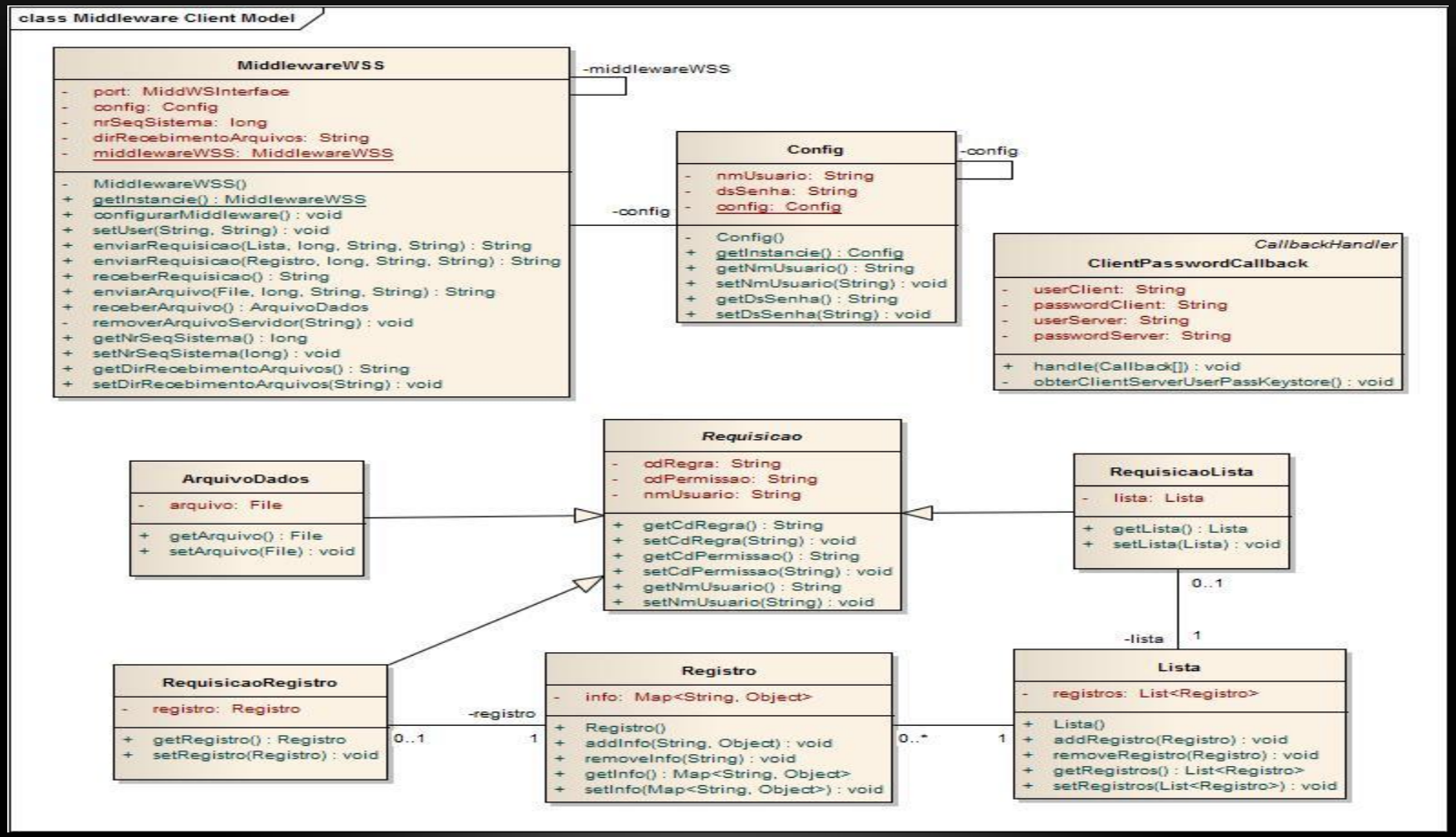


DIAGRAMA DE CLASSES SERVIDOR WEB SERVICE

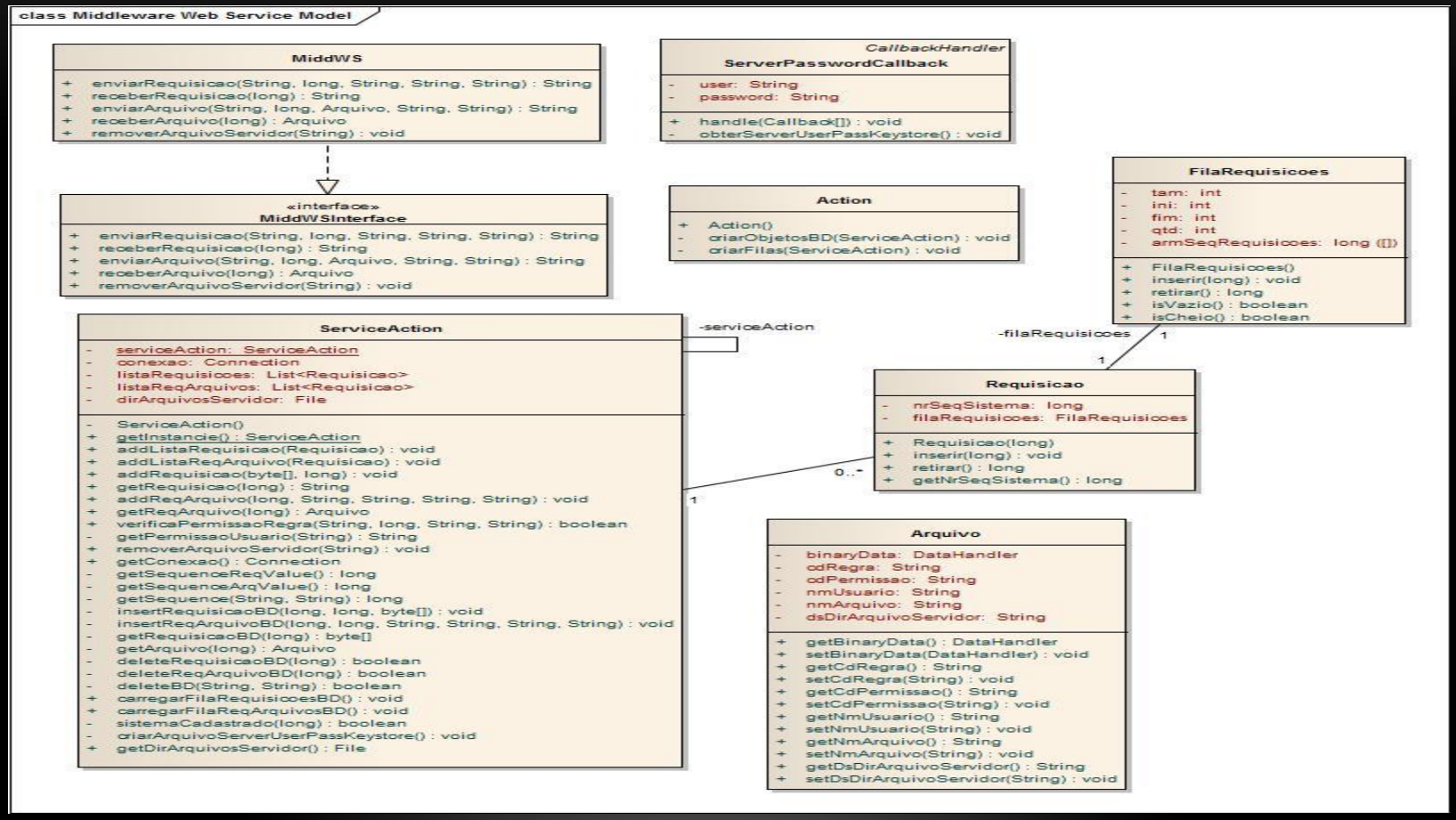


DIAGRAMA DE ATIVIDADES

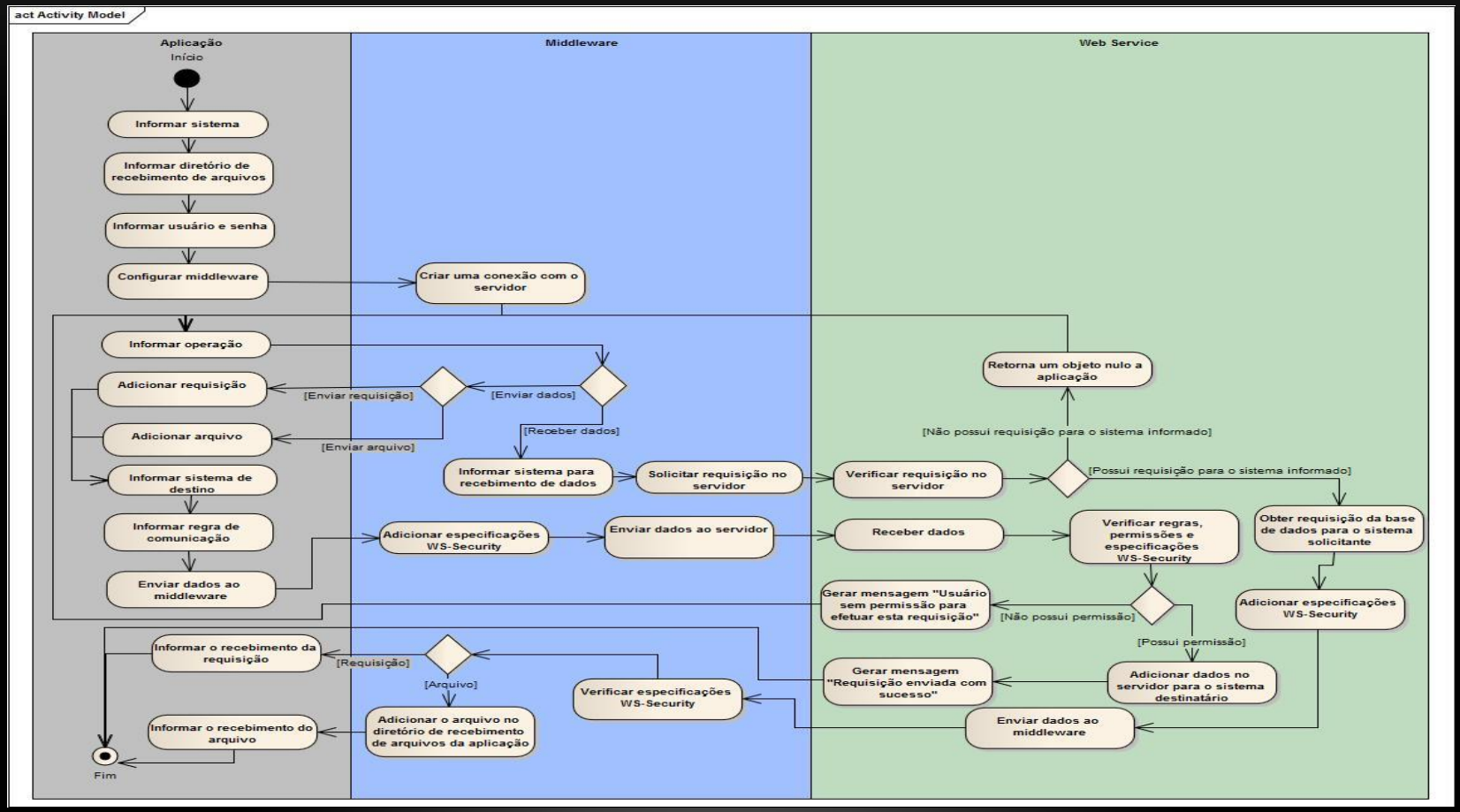


DIAGRAMA DE SEQUENCIA

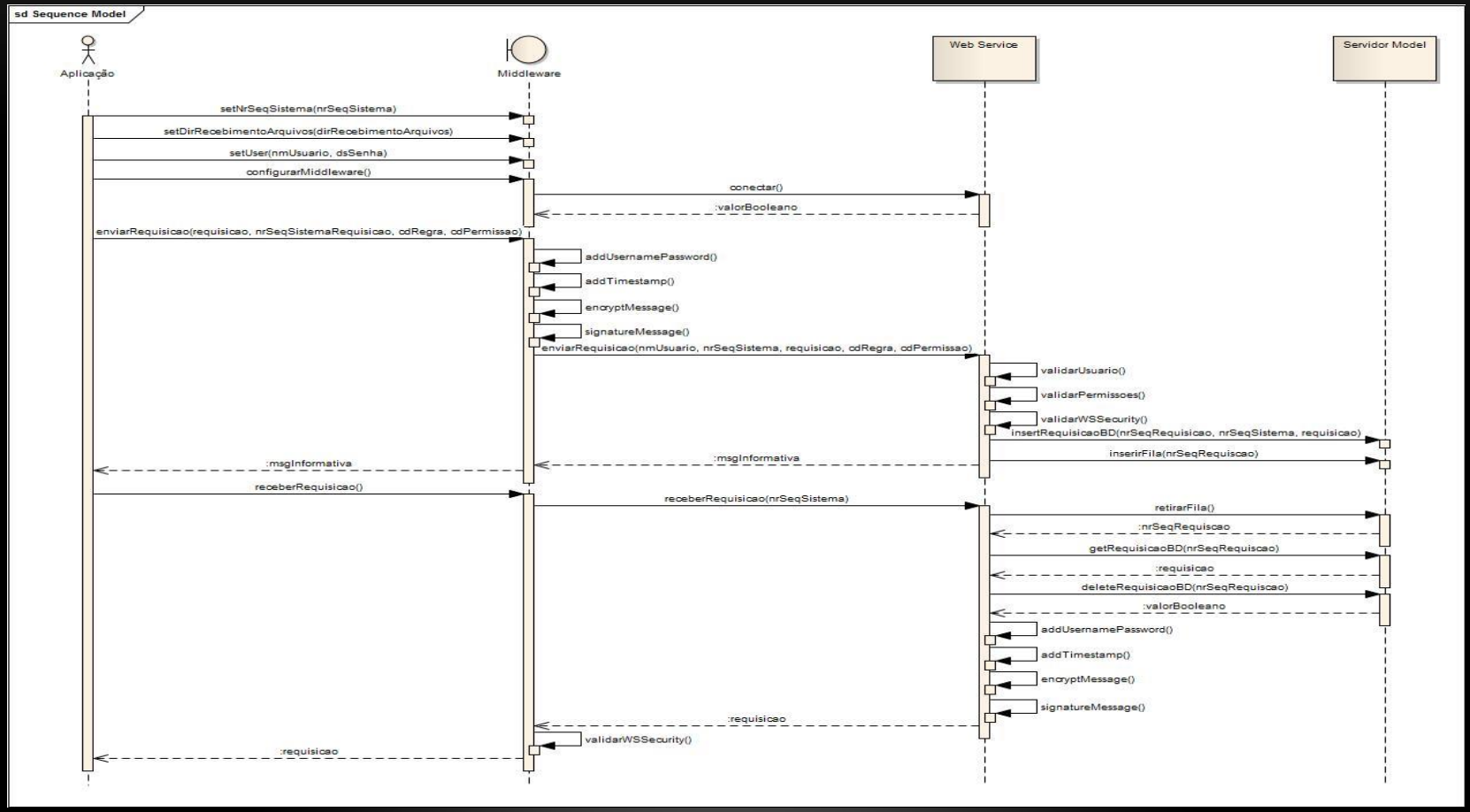


DIAGRAMA DE ENTIDADE RELACIONAMENTO

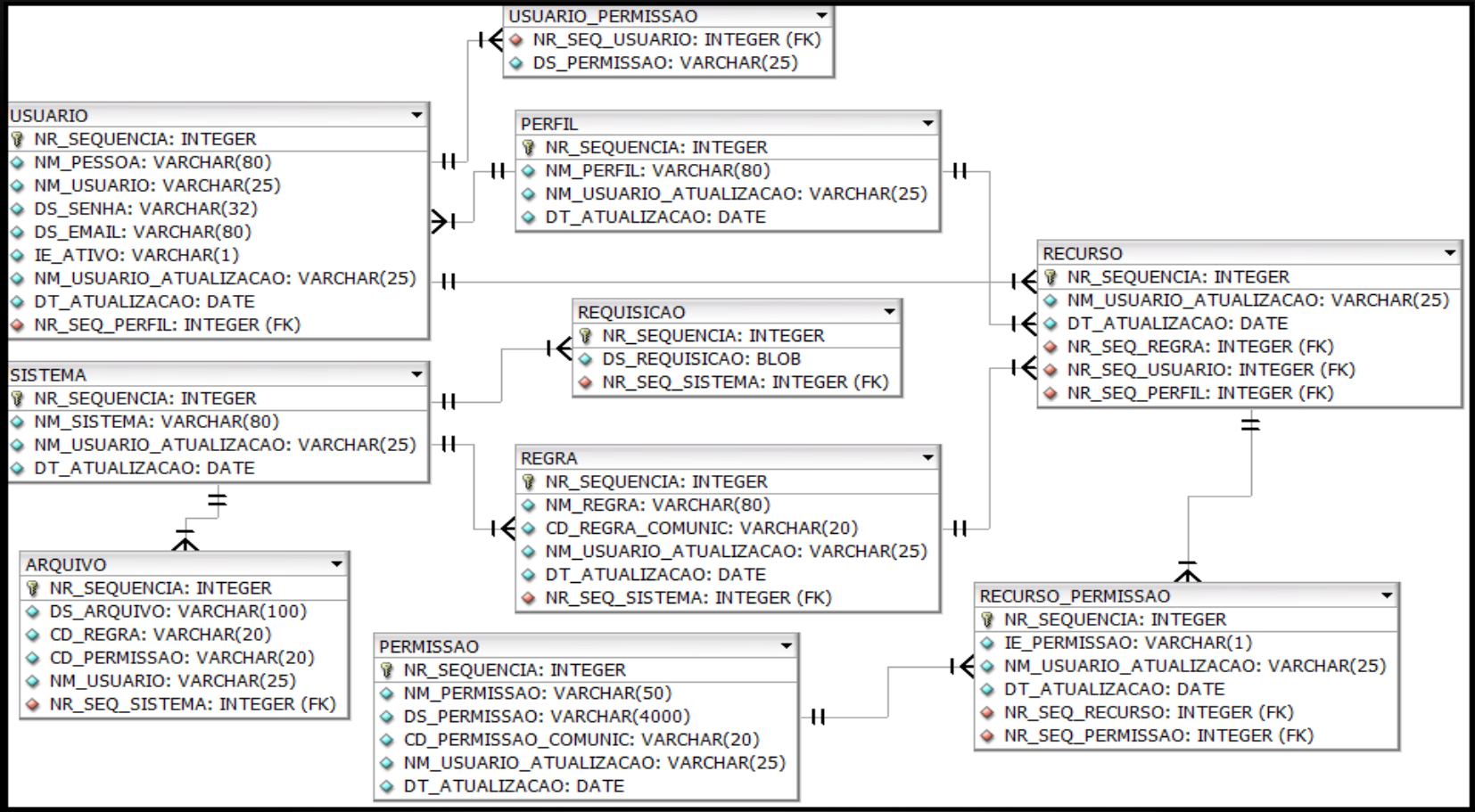
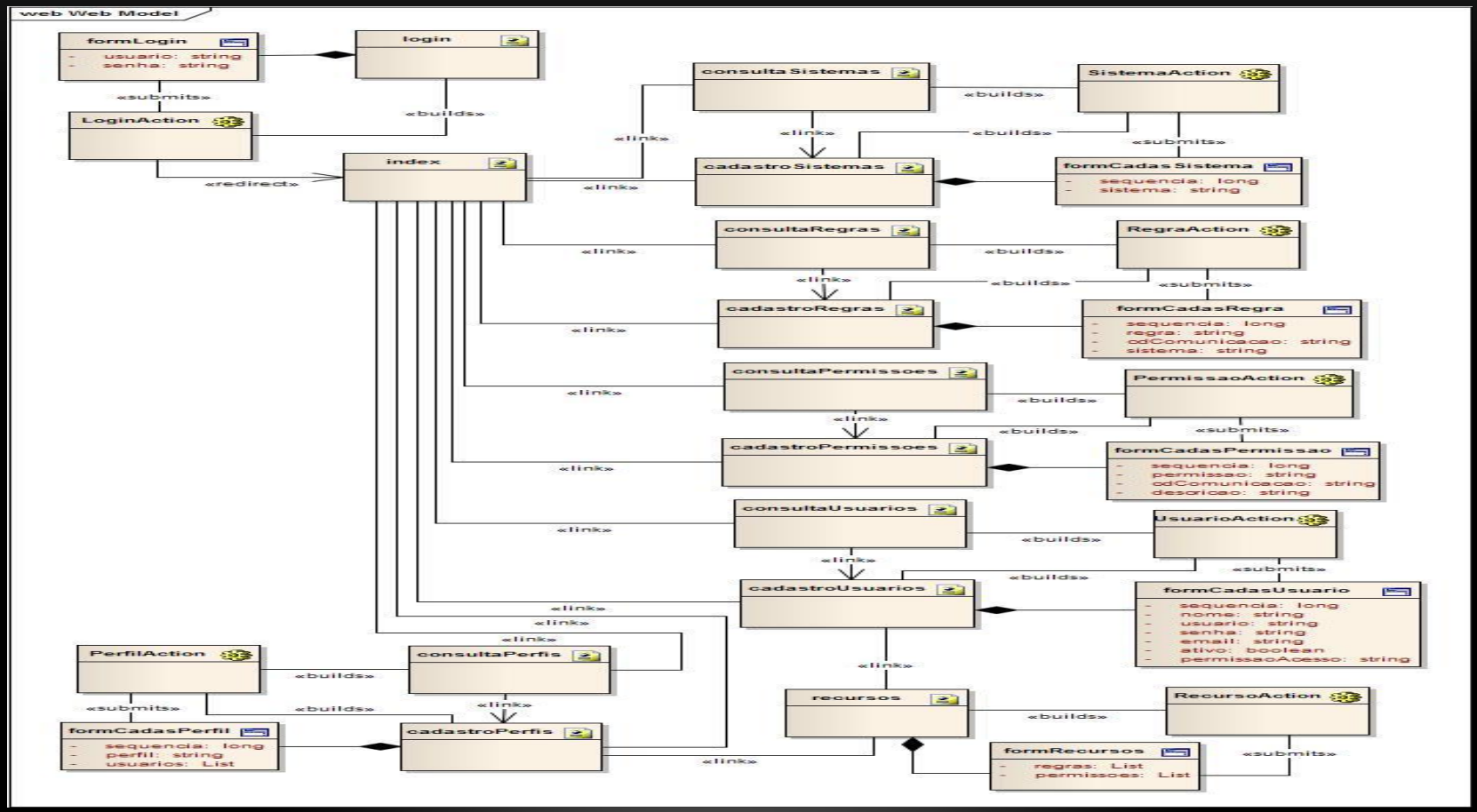


DIAGRAMA DE WAE



IMPLEMENTAÇÃO

TÉCNICAS E FERRAMENTAS UTILIZADAS

- Linguagem de programação Java
- Orientação Objetos
- NetBeans IDE 7.2
- Banco de dados Oracle 10g
- SQL Developer 3.2.2
- Apache Tomcat 7.0.27

- Apache CXF 2.7.6
- JDBC Oracle 10g
- JPA Hibernate 4.2.2
- JSF 2.1.22
- JSTL 1.2.1
- PrimeFaces 3.5
- PrimeFaces Theme Sam 1.0.10
- Spring Security 3.2.0
- Spring Security Facelets TagLib
- XStream 1.4.4

OPERACIONALIDADE DA IMPLEMENTAÇÃO

- Cenário onde possui um sistema distribuído
- Servidor com o serviço web configurado
- Criado duas aplicações para testes do middleware
- Módulo faturamento
- Módulo contábil

RESULTADOS E DISCUSSÃO

- Com o desenvolvimento do projeto de software, alcançou-se os objetivos propostos para o mesmo. Assim, através do serviço de segurança disponibilizado, qualquer aplicação já desenvolvida pode incorporar funções de criptografia, autenticação, assinatura digital e permissoes de acesso aos dados sem a necessidade de maiores implementações, uma vez que a implementação é disponibilizada pelos serviços de segurança, reforçando o conceito inicial de ser um middleware e uma parte de um sistema distribuído.

- Como resultado final, foi criado o middeware para integração com outras aplicações, com a finalidade de disponibilizar serviços de segurança à aplicações que não possuem estas funcionalidades. A maior vantagem deste middleware é fazer com que os desenvolvedores não precisem desenvolver rotinas de segurança em suas aplicações, podendo utilizar as rotinas já definidas.

- Com relação aos trabalhos correlatos, os mesmos apresentam soluções baseadas em web services, as quais foram estudadas para verificar o desenvolvimento do middleware de segurança. Os trabalhos de Hansen e Pinto (2003), Martins, Rocha e Henriques (2003) e Silva (2004) apresentam ainda algumas soluções para segurança em sistemas distribuídos, sendo que, diferentemente do trabalho desenvolvido, limitam-se apenas ao estudo, sem o desenvolvimento ou utilização das técnicas descritas.

DIFERENÇAS ENTRE TRABALHOS CORRELATOS E PROJETO DESENVOLVIDO

	Trabalho 1	Trabalho 2	Trabalho 3	Projeto
Web services	X	X	X	X
Segurança		X	X	X
Regras de comunicação				X

CONCLUSÕES

- Com o desenvolvimento do middleware, tendo como objetivo integrar-se as aplicações de modo geral, de forma transparente na implementação das questões de segurança com web services e dos detalhes da implementação do WS-Security, para efetuar a comunicação via web de forma segura, garantindo a confidencialidade, integridade e disponibilidade dos dados, baseando-se nas regras previamente registradas na interface web, sendo alcançados no desenvolvimento deste projeto todos os objetivos propostos.

- Este trabalho apresentou o desenvolvimento da especificação do WS-Security na comunicação de dados entre as aplicações, utilizando regras e permissões. O framework Apache CXF utilizado na implementação do projeto se demonstrou muito flexível e extensível para diversas situações e necessidades. O projeto possui uma limitação, sendo que na comunicação de dados utilizando arquivos é possível apenas utilizar um arquivo por vez na transfêrencia de dados de uma aplicação a outra, ésta limitação foi implementada devida a performace do middleware e do trafego na rede.

- A API Apache CXF em Java foi escolhida como base para implementação da especificação do WS-Security, se demonstrou bem intuitiva e estrutura. Comparada com outras implementações, a API Apache CXF é bem mais simples e leve, principalmente porque utiliza interceptadores spring em sua implementação, e por estes motivos foi escolhida como base para o desenvolvimento do projeto.

EXTENSÕES

- WS-Policy: definição de recursos e restrições;
- WS-Trust: definição de um modelo de confiança;
- WS-Privacy: define de que forma os web services serão implementados;
- WS-Secure Conversation: define como autenticar e gerenciar troca de mensagens;
- WS-Federation: define o gerenciamento de relacionamentos em ambientes heterogêneos;
- WS-Authorization: define a forma de administração dos dados pelos web services.

DEMONSTRAÇÃO DO PROJETO DE SOFTWARE

OBRIGADO!