

Tiago Estevão De Rolt

*Sistema Gerenciador de Ciclo de vida de
Certificados de Atributos*

Santa Catarina - SC, Brasil

28 de Abril de 2010

Tiago Estevão De Rolt

*Sistema Gerenciador de Ciclo de vida de
Certificados de Atributos*

Monografia apresentada para obtenção do
Grau de Bacharel em Sistemas de Informa-
ção pela Universidade Federal de Santa Ca-
tarina.

Orientador:
Prof. Dr. Ricardo Felipe Custódio

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DE SANTA CATARINA

Santa Catarina - SC, Brasil

28 de Abril de 2010

Monografia de Projeto Final de Graduação sob o título “*Sistema Gerenciador de Ciclo de vida de Certificados de Atributos*”, defendida por Tiago Estevão De Rolt e aprovada em 28 de Abril de 2010, em Santa Catarina, Estado de Santa Catarina, pela banca examinadora constituída pelos senhores:

Prof. Dr. Ricardo Felipe Custódio
Orientador

Msc. Marcelo Brocardo
Universidade Federal de Santa Catarina

Msc. Fabiano Castro Pereira
Serviço Federal de Processamento de Dados

Jonathan Gehard Kohler

Resumo

A utilização cada vez maior de documentos eletrônicos traz consigo a necessidade de ferramentas e procedimentos para proteção e segurança desta massa de documentos.

Uma destas tecnologias é a assinatura digital e cifragem de dados proporcionadas por certificados digitais. Certificados digitais são estrutura de dados que possuem informações de chave, uma espécie de senha, e informações de seu portador, os certificados de chave pública.

Com o estudo e uso cada vez maior sobre os certificados novas tecnologias são conhecidas, seja em função de deficiências do certificado de chave pública, seja por alternativas ao próprio certificado de chaves públicas.

O objeto de estudo deste trabalho é o certificado de atributos, um certificado que possui a funcionalidade de gerenciar com maior eficiência os atributos de seu portador. Deficiência esta encontrada em certificados de chaves públicas.

Foi gerado um protótipo de Autoridade de Atributos, uma espécie de Autoridade Certificadora para a ICP-Brasil. Com este protótipo foi possível estudar com maiores detalhes o porque do certificado de atributos possuir estas características que o diferenciam do certificado de chaves públicas.

Mais que uma simples comparação foi possível visualizar como as tecnologias de atributos e de chaves públicas se complementam.

Palavras-chave: Certificado digital, atributos, certificado de atributos, Autoridade Certificadora.

Abstract

The increasing use of electronic documents brings the tools and procedures necessary for safety and security of this mass of documents.

One of these technologies is the digital signature and encryption of data provided by digital certificates. Digital certificates are data structures that have key information, a kind of password, and information from your carrier, the public key certificates.

With the study and increasing use of certificates, new technologies are known, either because of deficiencies in the public key certificate, or alternatives to their own public key certificate.

The object of this paper is the attribute certificate, a certificate that has the capability to manage more efficiently the attributes of its wearer. Deficiency found in this public key certificates.

Generated a prototype of Attribute Authority, a type of Certificate Authority for the ICP-Brazil. With this prototype was able to study in more detail why the certificate of attributes they possess characteristics that differentiate it from the public key certificate.

More than a simple comparison was possible to visualize how technology of public-key and attributes are complementary.

Keywords: Digital certificate, attributes, attribute certificate, Certificate Authority.

Dedicatória

Dedico este trabalho aos meus familiares que muito me apoiaram durante não somente este período de faculdade, mas por toda a minha vida. Conto sempre com vocês! Com certeza todas as minhas conquistas também são suas conquistas. Dedico também ao Heavy Metal e a música. Hell awaits!!!

Agradecimentos

Agradeço a Deus e seus enviados. Meus amigos que tornaram sempre mais agradáveis os momentos vividos nesta vida universitária. Ao Megadeth por ter lançado o cd "Rust in Peace", CDs como este trazem um momento de calma e reflexão. Agradeço a BRy Tecnologia empresa onde trabalho pelos anos de convivência e de grandes conquistas.

Sumário

Lista de Figuras

1	Introdução	p. 12
1.1	Justificativa	p. 14
1.2	Motivação	p. 15
1.3	Objetivo	p. 15
1.3.1	Objetivo Geral	p. 15
1.3.2	Objetivos Específicos	p. 15
1.4	Estrutura do Trabalho	p. 16
2	Fundamentação Teórica Geral	p. 17
2.1	Conceitos de Criptografia	p. 17
2.2	Criptografia Simétrica	p. 18
2.3	Criptografia Assimétrica	p. 19
2.4	Infraestrutura de Chaves Públicas	p. 20
2.4.1	Autoridade Certificadora Raiz	p. 22
2.4.2	Autoridades Certificadoras	p. 22
2.5	Certificado Digital	p. 22
2.6	Lista de Certificados Revogados	p. 24
2.7	Conclusão	p. 25
3	Fundamentação Teórica Certificado de Atributos	p. 26
3.1	Certificado de Atributos	p. 26

3.2	Infraestrutura de Certificado de Atributos	p. 31
3.3	Conclusão	p. 33
4	Ferramentas e Tecnologias de Desenvolvimento	p. 34
4.1	Plataforma Java Enterprise Edition	p. 34
4.2	Webservices	p. 35
4.2.1	Arquitetura dos Webservices	p. 35
4.2.2	Linguagem de marcação XML	p. 36
4.2.3	Protocolo de transmissão SOAP	p. 36
4.2.4	Linguagem de descrição de Serviço WSDL	p. 38
4.3	Bouncy Castle API de Criptografia para Java	p. 38
4.4	Conclusão	p. 39
5	Visão Geral da Aplicação	p. 40
5.1	Levantamento e Análise de Requisitos	p. 40
5.1.1	Processo de registro e emissão de certificados	p. 41
5.1.2	Processo de Emissão de lista de Certificados Revogados	p. 42
5.2	Requisitos de Negócio	p. 42
5.3	Requisitos Funcionais	p. 42
5.3.1	Requisitos funcionais da Autoridade de Atributos	p. 43
5.3.2	Requisitos funcionais de administração	p. 43
5.3.3	Requisitos funcionais de módulo público	p. 43
5.4	Casos de Uso	p. 44
5.4.1	Atores do Sistema	p. 45
5.4.2	Detalhamento dos casos de uso	p. 45
5.5	Requisitos Não Funcionais	p. 46
5.6	Módulos da Aplicação	p. 51

5.7	Conclusão	p. 53
6	Implementação da Autoridade de Atributos	p. 54
6.1	Módulo Principal e Webservices	p. 54
6.2	Módulo Público	p. 61
6.3	Módulo de Administração	p. 67
6.4	Resultados Obtidos	p. 69
6.5	Conclusão	p. 70
7	Considerações Finais	p. 71
7.1	Trabalhos Futuros	p. 72
	Referências	p. 73
	Apêndice A – Códigos Fontes	p. 75
A.1	Módulo Principal	p. 75
A.1.1	Módulo Logica de Negócio	p. 75
A.1.2	Módulo Persistência de Dados	p. 110
A.1.3	Módulo Webservice	p. 142
A.2	Módulo Público	p. 149
A.3	Módulo Administrativo	p. 169

Lista de Figuras

1	Crescimento das entidades credenciadas a ICP-Brasil	p. 13
2	Certificado Digital emitido para ACs ICP-Brasil Nível 1 e 2 (incluindo renovações: total de 65)	p. 13
3	Normas jurídicas emitidas sobre certificação digital	p. 14
4	Criptografia simétrica	p. 19
5	Criptografia assimétrica - Beto enviando uma mensagem assinada para Ana	p. 20
6	Estrutura básica de uma ICP	p. 21
7	Estrutura ASN.1 do certificado X.509	p. 23
8	Estrutura ASN.1 que representa a extensão do certificado X.509	p. 24
9	Estrutura ASN.1 de um certificado de atributo	p. 28
10	Holder de um certificado de atributo	p. 29
11	Estrutura ASN.1 do campo attribute de um certificado de atributo	p. 30
12	Hierarquia em uma ICP e em uma IGP	p. 32
13	Estrutura do pacote de transmissão SOAP	p. 37
14	Diagrama de caso de usos	p. 44
15	Módulos da Autoridade de Atributos	p. 52
16	IDE Eclipse utilizada no desenvolvimento da Autoridade de Atributos	p. 55
17	Diagrama de sequência ilustrando a requisição e persistência de um certificado de atributo	p. 57
18	Diagrama de sequência ilustrando a adição de atributos na requisição	p. 58
19	Diagrama de sequência ilustrando a emissão de um certificado de atributos	p. 59

20	Ponto de acesso aos serviços disponibilizados pela Autoridade de Atributos	p.61
21	Página inicial do módulo público	p.62
22	Applet para validação do certificado de chaves públicas	p.62
23	Número de requisição gerado após validação do certificado	p.63
24	Inserção do número de requisição para emissão do certificado de atributos	p.63
25	Inserção dos atributos a serem emitidos pela Autoridade de Atributos .	p.64
26	Download do certificado de atributos após emissão	p.64
27	Visualização dos certificados emitidos pela Autoridade de Atributos . .	p.65
28	Requisição para revogação do certificado de atributos	p.65
29	Motivo para revogação do certificado de atributos	p.66
30	Consulta de certificados revogados	p.66
31	Interface gráfica do módulo de administração	p.68
32	Interface gráfica de configuração do módulo de administração	p.68
33	Certificado de atributo emitido utilizando a Autoridade de Atributos .	p.69
34	Lista de certificados revogados emitida utilizando a Autoridade de Atributos	p.70

1 *Introdução*

Atualmente devido à intensificação do uso de computadores e sistemas de informação o documento vai deixando o ambiente físico passando a ser um documento eletrônico. Devido à popularização do meio eletrônico vem a tona um importante tema, como proteger essa massa de documentos de forma segura e confiável. Um documento eletrônico para ser dito confiável deve conferir autenticidade, tempestividade e integridade que são encontradas no meio físico. Quando um documento confere autenticidade tem-se a certeza de qual a pessoa foi responsável pela criação do documento e é responsável pelas informações de seu conteúdo. Na tempestividade é criada uma âncora temporal, garantindo a existência do documento a partir da data contida na âncora temporal e também a garantia de que o documento não foi alterado sem que haja vestígios desta alteração. Na integridade garante-se que o documento não sofreu nenhuma adulteração após sua criação. Mecanismos e protocolos criptográficos como a assinatura digital garantem os requisitos de autenticidade e integridade, a tempestividade é garantida através do uso do carimbo do tempo.

A utilização dos mecanismos de proteção como a assinatura digital, acabou popularizando o uso de certificados digitais, certificados digitais são arquivos que possuem informações de seu portador e provêm insumos para a geração de uma assinatura digital. No período de 2002 a 2008 subiu de 8 para 26 o número de entidades credenciadas junto a ICP-Brasil, sendo que no período de 2001 a 2008 o número de normas jurídicas sobre certificação chegou a mais de 80 publicações[1]. Nos gráficos abaixo é possível ter uma idéia da expansão.

Com o tempo os certificados foram evoluindo e muitas aplicações foram sendo geradas utilizando a sua tecnologia. Decorrente desta evolução as informações contidas nestes certificados também foram se atualizando e aumentando. Muitas destas informações adicionais são características de escopo da vida social do portador do certificado digital, estas características sendo pessoais e diferentes em cada indivíduo causam dificuldades às autoridades certificadoras que precisam de alguma forma obtê-las a fim de adicioná-las

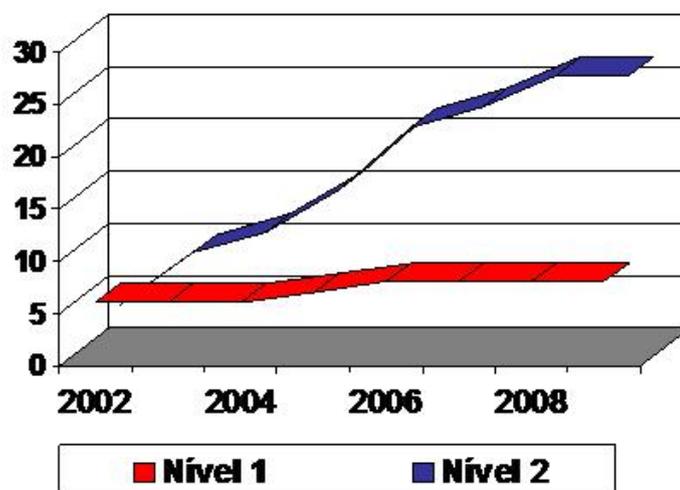


Figura 1: Crescimento das entidades credenciadas a ICP-Brasil

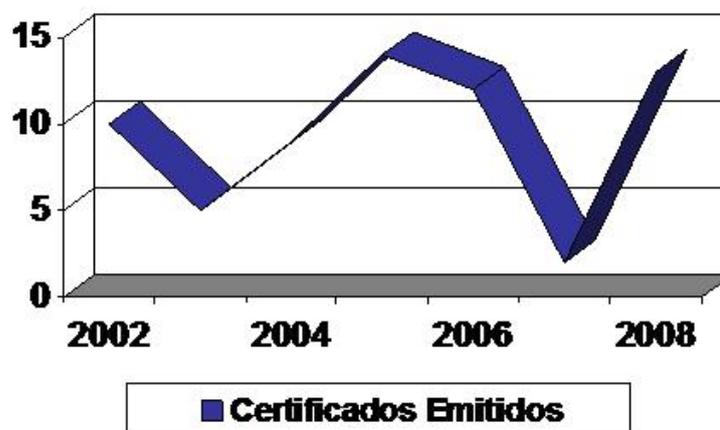


Figura 2: Certificado Digital emitido para ACs ICP-Brasil Nível 1 e 2 (incluindo renovações: total de 65)

aos certificados digitais.

Devido a estas dificuldades um novo modelo de certificado foi proposto com o objetivo de portar características de seu portador de forma mais fácil e segura. Trata-se do Certificado de Atributo (CA), que possui características que facilitam a adição e manutenção de informações.

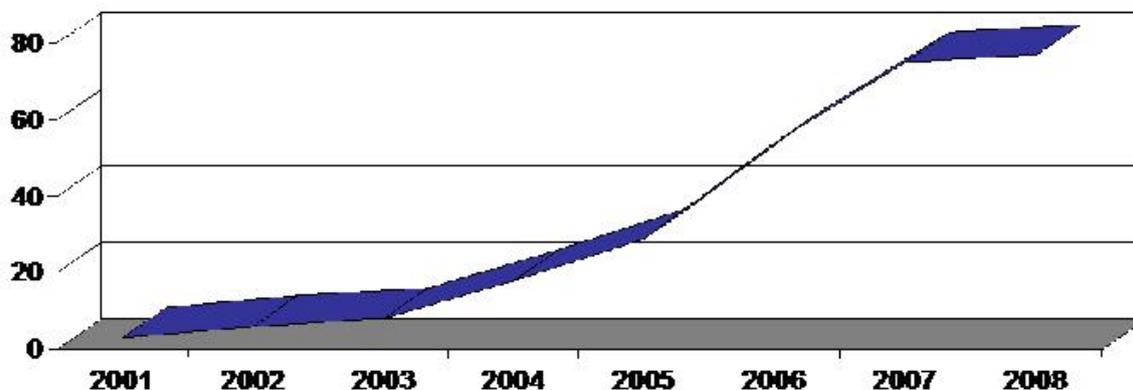


Figura 3: Normas jurídicas emitidas sobre certificação digital

1.1 Justificativa

Atualmente a Infraestrutura de Chaves Pública Brasileira (ICP-Brasil) possui políticas de emissão de certificados para assinatura digital, sendo que essas políticas determinam um tempo limite mínimo de um (1) ano e no máximo três (3) anos de validade do certificado até sua expiração.

As características do portador do certificado podem mudar ao longo do tempo de vida do certificado digital de chaves públicas e devido a esta característica dos certificados de chaves públicas, de possuírem um tempo de vida extenso, pode-se afirmar que qualquer alteração sobre as características do portador acaba afetando o certificado digital de chaves públicas, podendo até levar a uma revogação prematura ou expiração prematura. Devido a este problema sentiu-se a necessidade de emissão de certificados de curto prazo, certificados que possuem informações de seu portador tais que estas informações possuam um tempo de vida menor que a duração dos certificados atuais.

Além do problema temporal, ao agregar informações ao certificado digital deve-se garantir que estas informações sejam válidas durante todo o período de vida do certificado digital, caso contrário um novo certificado deve ser emitido com as informações atualizadas causando a revogação precoce do certificado anterior.

Os certificados de atributos são semelhantes aos certificados de chave pública [2], mas possuem um diferencial, possuem a única e exclusiva finalidade de publicar determinadas in-

formações de seu portador. Devido a esta característica possuem mecanismos que tornam fácil a manutenção destas informações e evita os problemas acima.

1.2 Motivação

A crescente utilização dos certificados digitais nos mais diferentes contextos e aplicações traz consigo a responsabilidade de uso desta tecnologia e é um fator motivador para futuras pesquisas em sua área.

O uso do certificado de chave pública é apenas um artefato de toda a tecnologia e estudo que existe na área de certificação digital. Além do próprio certificado de chaves públicas outros modelos e arquiteturas existem e podem ser de utilidade o que traz a necessidade de pesquisas neste área.

Na infraestrutura de chaves pública brasileira e órgãos gestores o certificado de chaves públicas é o de maior aceitação e discutido, não existem grandes informações sobre outros modelos de certificados, expandir conhecimento e estudar outras tecnologias é preciso.

1.3 Objetivo

1.3.1 Objetivo Geral

Construção de um Sistema funcional Gerenciador de Ciclo de Vida de Certificados de Atributos.

1.3.2 Objetivos Específicos

- Demonstrar as vantagens da utilização do certificado de atributos em relação ao uso dos certificados atuais;
- Conhecer o processo de emissão dos certificados de atributos;
- Verificar como se dá a coexistência do certificado de atributos e o certificado de chaves públicas;
- Aplicar os conhecimentos adquiridos no curso de Sistemas de Informação;
- Agregar conhecimento na área de certificação digital e criptografia;

- Adquirir conhecimento sobre projeto e desenvolvimento de software;
- Compreensão de desenvolvimento de sistemas que utilizem Webservice;

1.4 Estrutura do Trabalho

No presente trabalho será detalhado o modelo e estrutura do certificado de atributos de forma que seja fácil sua compreensão, provendo uma base teórica para seu entendimento e se necessário o reforço dos conceitos através de figuras, tabelas e de exemplos. A seguir um é demonstrado como o trabalho será dividido e quais informações você encontrará em cada uma de suas seções.

Estrutura organizacional do trabalho:

- Capítulo 2: neste capítulo é apresentado um pequeno resumo sobre os conceitos de criptografia e certificação digital encontrados na literatura;
- Capítulo 3: capítulo de fundamental importância para o trabalho. Neste capítulo são apresentados os principais conceitos que permeiam o certificado de atributos e a estrutura utilizada para sua emissão;
- Capítulo 4: neste capítulo são apresentadas as tecnologias utilizadas no projeto do protótipo de autoridade certificadora. Será apresentado um texto sobre cada uma das ferramentas utilizadas para confecção do mesmo;
- Capítulo 5: Uma apresentação geral do funcionamento da aplicação. Seu projeto conceitual com diagramas e requisitos utilizados no desenvolvimento do protótipo;
- Capítulo 6: A implementação do projeto, sua arquitetura, funcionamento e dificuldades de implementação;
- Capítulo 7: As considerações gerais sobre o protótipo e a certificação de atributos e conclusão sobre o trabalho desenvolvido;

2 Fundamentação Teórica Geral

No presente capítulo será apresentada uma breve revisão bibliográfica contendo fundamentação e conceitos básicos de criptografia e certificação digital, tendo como objetivo situar o leitor do trabalho aos conceitos básicos e de fundamental importância para compreensão dos capítulos futuros.

Na primeira parte do capítulo serão apresentados conceitos de criptografia simétrica e assimétrica, que são à base de todas as tecnologias posteriores utilizadas na área de segurança da informação.

Na segunda parte serão apresentados os conceitos de infraestrutura de chaves públicas. Através da infraestrutura de chaves públicas é possível obter certificados digitais.

Na terceira parte será apresentado o conceito de lista de certificados revogados, um mecanismo de proteção provido pela Autoridade Certificadora contra eventos catastróficos acontecidos a certificados por ela emitidos.

Na última parte uma conclusão sobre os fundamentos gerais apresentados, qual o objetivo dos mesmos com a continuidade do trabalho.

2.1 Conceitos de Criptografia

Criptografia é uma área do conhecimento, especificamente da matemática e computação, que aborda problemas relacionados à segurança de dados.

O conceito de criptografia vem do Grego *kryptós*, "enigma", e *graphía*, "escrita" e é a ciência que trata de codificar dados de forma que pareçam não possuir um sentido, objetivando esconder informações.

Segundo Willian Stallings [3] "a criptografia é a área de estudo onde uma mensagem original ou texto plano passa por um processo de codificação de dados chamado cifragem, tendo como resultado dessa codificação um texto cifrado. O processo inverso, o processo

de recuperar o texto plano a partir de um texto cifrado é chamado decifragem". As tecnologias de criptografia atuais são empregadas no sentido de garantir cinco requisitos [3]:

- Sigilo;
- Integridade;
- Autenticidade;
- Não Repúdio;
- Tempestividade;

Os requisitos acima são garantidos utilizando algoritmos empregados na criptografia. Os algoritmos de criptografia são divididos em algoritmos simétricos e algoritmos assimétricos. O conhecimento desses algoritmos é de fundamental importância sendo então apresentados nos tópicos a seguir.

2.2 Criptografia Simétrica

A criptografia simétrica é tradicionalmente conhecida como criptografia convencional. A sua principal característica é a chave utilizada para codificar os dados e a mesma chave utilizada para decodificar os dados [4]

Uma chave é um conjunto de bits utilizados pelos algoritmos criptográficos, como uma senha.

O remetente e o destinatário utilizam a mesma chave para codificar e decodificar mensagens. O remetente de posse da chave utiliza algum algoritmo de cifragem simétrica sobre o texto a ser codificado e envia o texto embaralhado ao destinatário. O destinatário que já possui a mesma chave e o conhecimento do algoritmo os utiliza para a extração do texto plano, ver figura abaixo.

A criptografia simétrica possui como característica fundamental o compartilhamento da chave entre remetente e destinatário. Reside neste compartilhamento de chaves o grande problema no uso da criptografia simétrica. Caso a chave seja compartilhada em um canal aberto nada impede que um intruso obtenha uma cópia da chave e acesso ao texto plano.

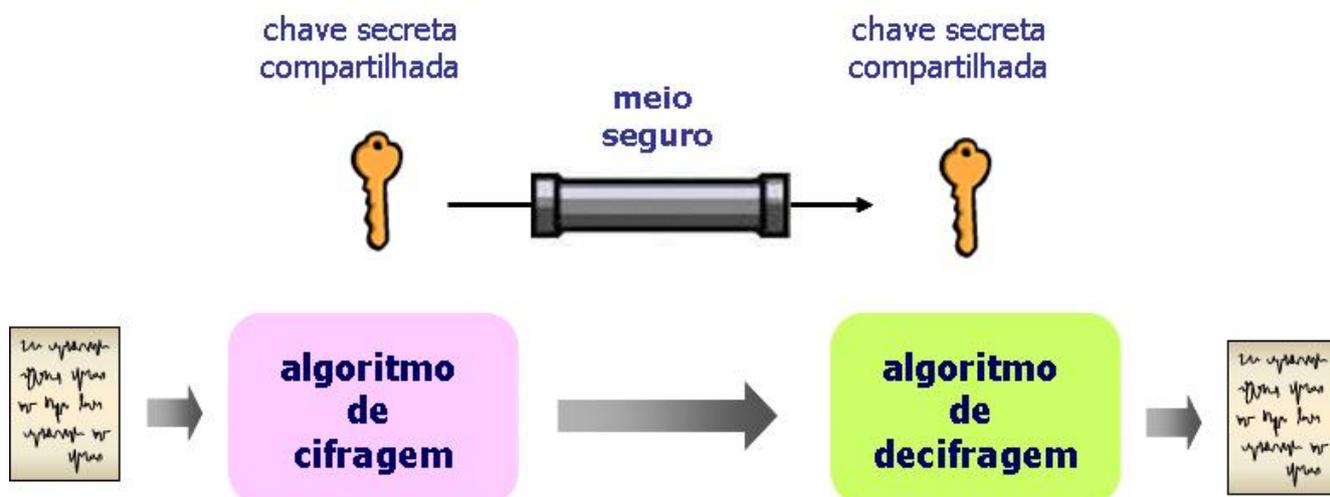


Figura 4: Criptografia simétrica

Devido ao problema acima foi criado outro tipo de cifragem, a cifragem assimétrica ou de chaves públicas, assunto abordado no próximo capítulo.

2.3 Criptografia Assimétrica

O algoritmo de cifragem assimétrica ou de chave pública possui como premissa que a chave utilizada no momento da cifragem do texto plano é diferente da chave utilizada no momento da decifragem do texto codificado [4].

Neste modelo de cifragem existem duas chaves a pública e a privada, sendo que a chave pública deve ser de alguma forma publicada, tornando-se um artefato de livre distribuição, e a chave privada deve ser mantida sob sigilo.

O modelo de chaves pública foi proposto por Whitfield Diffie e Martin Hellman no ano de 1976 e cerca de um ano depois R. L Rivest, A. Shamir e L. M Adelman publicam o primeiro algoritmo a utilizar o modelo, o RSA. O RSA é o algoritmo assimétrico utilizado em grande escala quando se trata de criptografia assimétrica até hoje.

A utilização do modelo de chaves públicas acaba diminuindo muito o problema de compartilhamento da chave, mesmo em ambientes inseguros como a internet. Mas o benefício da segurança acaba trazendo consigo um malefício. A criptografia assimétrica devido as suas características acaba gerando um maior custo computacional, um maior tempo de execução de suas funções.

Para solucionar o problema computacional normalmente procura-se utilizar a criptografia assimétrica em conjunto com a simétrica melhorando o seu desempenho, ver figura.



Figura 5: Criptografia assimétrica - Beto enviando uma mensagem assinada para Ana

A chave pública nada mais é que um conjunto de bits, o que dificulta a sua identificação e a qual respectivo portador ela pertence. Visando o objetivo de atrelar uma chave pública a seu portador foi criado o modelo de infraestrutura de chaves públicas, tema do próximo capítulo.

2.4 Infraestrutura de Chaves Públicas

A infraestrutura de chaves públicas ou ICP é uma infraestrutura de segurança pervasiva, tem como objetivo melhorar a utilização humana dos conceitos que permeiam a criptografia de chaves públicas [5].

Na infra-estrutura de chave pública os seguintes conceitos são apresentados [6]:

- **Autoridade Certificadora:** Entidades que passam por uma série de auditorias e após isso estão habilitadas a emitir certificados para quaisquer entidades, desde pessoas físicas, órgãos do governo e empresas privadas;
- **Autoridade de Registro:** Entidades responsáveis por identificar o portador da chave pública, de forma a registrar o indivíduo junto a Autoridade Certificadora. Somente com sua autorização um certificado é emitido a uma entidade;
- **Repositório:** Sistema responsável por distribuir os certificados e as lista de revogação dos certificados;
- **Lista de Revogação de Certificados:** Lista pública onde constam certificados que a Autoridade Certificadora deixou de confiar por algum motivo;

O certificado digital é o artefato de suma importância em uma ICP, é uma estrutura de dados que possui informações sobre o detentor do par de chaves. Ele amarra a chave pública e conseqüentemente a chave privada a um indivíduo.

No momento anterior a uma comunicação entre indivíduos utilizando certificado digital, é executada uma verificação sobre o certificado digital e a chave pública do mesmo, essa verificação consiste em verificar se a Autoridade Certificadora ou AC que emitiu o certificado é confiável. Dessa forma a AC acaba se tornando uma terceira parte de confiança entre as partes.

Para uma entidade emitir um certificado junto a uma Autoridade Certificadora é necessário um ato de registro. O ato de registro é o momento que a entidade deve se identificar junto a Autoridade Certificadora firmando seu vínculo a chave pública.

Para ser registrada a uma Autoridade Certificadora a entidade deve estar de acordo com as regras impostas pela própria autoridade. Estas regras estão descritas no documento "Declaração de Práticas de Certificação" disponibilizado pela Infraestrutura de Chaves Públicas. Este documento deve ser seguido com rigor pela Autoridade de Certificação.

Caso a demanda por registro junto a Autoridade Certificadora seja intensa, a mesma pode delegar a tarefa de registro a uma terceira parte, as chamadas Autoridades de Registro [5]. Na figura abaixo pode ser visualizado a estrutura de uma ICP.

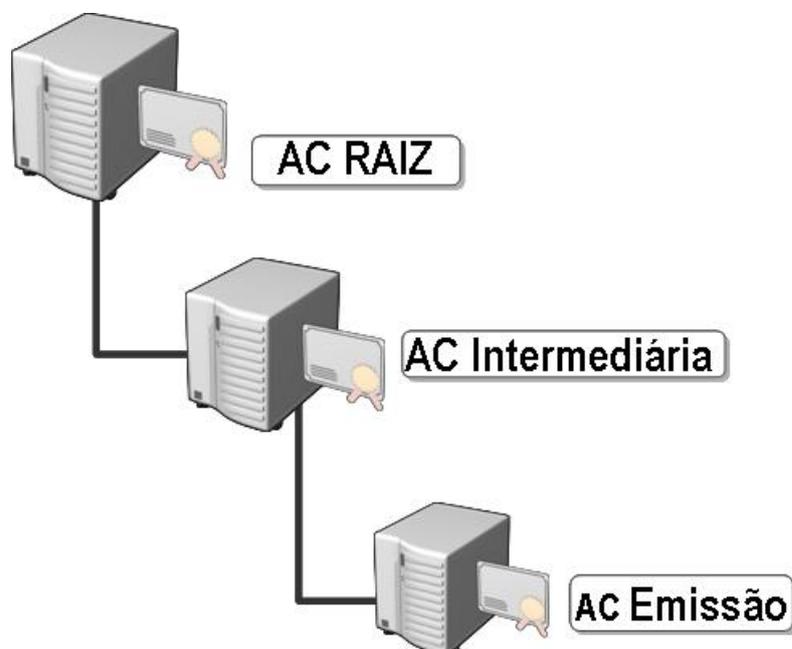


Figura 6: Estrutura básica de uma ICP

2.4.1 Autoridade Certificadora Raiz

A AC raiz é a Autoridade Certificadora que está estabelecida no topo da hierarquia da Infraestrutura de chaves públicas. A AC raiz é o núcleo de confiança em uma infraestrutura, todos os membros intermediários e de emissão confiam nela. Possui atribuições de gerenciar as autoridades certificadoras intermediárias, emitindo, distraindo e revogando certificados. Também possui a atribuição de fiscalizar e auditar as ACs por ela habilitados.

A AC-Raiz na ICP-Brasil é o Instituto Nacional de Tecnologia da Informação - ITI, autarquia federal vinculada a Casa Civil da Presidência da República[7].

2.4.2 Autoridades Certificadoras

São entidades posicionadas um nível abaixo na hierarquia de uma ICP, tendo seus certificados digitais emitidos pela autoridade raiz.

As Autoridades Certificadoras são entidades públicas ou pessoas jurídicas de direito privado credenciadas à AC-Raiz e que emitem certificados digitais vinculando pares de chaves criptográficas ao respectivo titular. Nos termos do art. 60 da MP 2.200/01, competem-lhes "emitir, expedir, distribuir, revogar e gerenciar os certificados, bem como colocar à disposição dos usuários listas de certificados revogados e outras informações pertinentes e manter registro de suas operações"[7].

Nos próximos capítulos é abordado com maior aprofundamento o conceito de certificado digital e o seu modelo mais utilizado o X.509, bem como o conceito de Lista de Certificados Revogados.

2.5 Certificado Digital

O certificado digital é um arquivo assinado digitalmente por uma entidade confiável com o objetivo de associar uma chave pública a uma entidade[4]. O certificado serve então como um mecanismo de publicação da chave pública.

No momento de efetiva geração de um certificado digital, a AC realiza uma assinatura digital sobre o mesmo. Bastando para verificar a integridade e autenticidade do certificado emitido, utilizar o certificado digital da AC e verificar sua assinatura.

Dentre as várias propostas de certificados digitais o modelo mais reconhecido e aceito como padrão é o ITU-T X509 atualmente em sua versão 3 [5]. No certificado digital

X509v3 consta uma série de informações dentre elas a chave pública, nome do emissor do certificado, data de validade e a assinatura digital da AC.

Na infraestrutura de chaves pública brasileira constam além das informações padrão do certificado X509v3, informações adicionais como CPF, RG, título de eleitor. Essas informações são adicionadas na estrutura do certificado X509 no campo "Subject Alternative Name", são os chamados campos ICP-Brasil.

Os certificados digitais possuem uma codificação especial, chamada ASN.1, o uso do ASN.1 prove a interoperabilidade de sistemas para obter as informações do certificado digital, a codificação ASN.1 do certificado x509v3 está melhor demonstrada na figura abaixo.

```
Certificate ::= SEQUENCE {
    tbsCertificate      TBSCertificate,
    signatureAlgorithm  AlgorithmIdentifier,
    signature           BIT STRING }

TBSCertificate ::= SEQUENCE {
    version             [0] Version DEFAULT v1,
    serialNumber        CertificateSerialNumber,
    signature           AlgorithmIdentifier,
    issuer              Name,
    validity            Validity,
    subject             Name,
    subjectPublicKeyInfo SubjectPublicKeyInfo,
    issuerUniqueID     [1] IMPLICIT UniqueIdentifier OPTIONAL,
                      -- If present, version MUST be v2 or v3
    subjectUniqueID    [2] IMPLICIT UniqueIdentifier OPTIONAL,
                      -- If present, version MUST be v2 or v3
    extensions         [3] Extensions OPTIONAL
                      -- If present, version MUST be v3 -- }
```

Figura 7: Estrutura ASN.1 do certificado X.509

A versão três do certificado digital X.509 propõe o uso de extensões. Extensões são utilizadas para adicionar maiores informações sobre a entidade portadora do certificado, a autoridade que emitiu o certificado ou sobre a própria chave pública [6]. A figura abaixo demonstra a estrutura de uma extensão segundo a RFC.

```
Extension ::= SEQUENCE {
    extnID      OBJECT IDENTIFIER,
    critical    BOOLEAN DEFAULT FALSE,
    extnValue   OCTET STRING }
```

Figura 8: Estrutura ASN.1 que representa a extensão do certificado X.509

A estrutura demonstrada na figura acima consiste dos seguintes componentes:

- extnID: Identificador de objeto(OID). Demonstra o tipo da extensão;
- critical: Se a flag estiver ativa demonstra que a extensão possui informações importantes. A extensão deve ser reconhecida caso contrario o certificado deve ser considerado inválido;
- extnValue: As informações contidas na extensão;

O certificado digital X509 em sua recomendação expõe os diversos propósitos de uso de um certificado digital X509 v3[6]. Nas extensões "KeyUsage" e "ExtendedKeyUsage" constam as informações sobre o propósito de uso da chave pública do certificado, dentre os propósitos podemos citar, assinatura digital, não repúdio e codificação de dados.

Em algumas situações de adversidade como roubo do certificado ou comprometimento da chave privada o certificado deve ser revogado. Desta forma quando uma entidade for utilizar um certificado digital ele deve primeiro verificar se o mesmo não está revogado. Para realizar esta verificação é utilizado a Lista de Certificados Revogados (LCR) uma lista divulgada pela AC.

2.6 Lista de Certificados Revogados

Certificados possuem características importantes, uma vez distribuídos é praticamente impossível tomá-los de volta. De fato este é um dos problemas ao utilizar certificados, uma vez em posse da entidade, a mesma pode replicá-lo e distribuí-lo o que dificulta determinar quantas copias existem [8].

Devido a essa característica, um certificado ser público e de livre distribuição pela entidade que o tem em posse, mecanismos foram criados para casos que o certificado perca sua validade antes do período de término de validade, no caso o mecanismo é a revogação.

Um certificado revogado é um certificado que a Autoridade Certificadora deixou de confiar por algum motivo, roubo, perda da chave privada entre outros.

O mecanismo mais comum entre as Autoridades Certificadoras para atestar a revogação de um certificado é a emissão de lista de certificados revogados. As listas de certificados revogados contêm o número de série dos certificados que a autoridade deixou de confiar. Esta lista é pública e pode ser utilizada por qualquer entidade para verificar o estado de qualquer certificado emitido por aquela Autoridade. Caso um certificado conste na lista, o mesmo deve ser ignorado e não utilizado [8]. Um certificado digital pode constar em uma lista de certificados até o momento anterior a seu término de validade, após isso, o certificado deixa de ser válido e também deixa de ser confiável pela autoridade certificadora.

Autoridades Certificadoras emitem as listas de certificados revogados em determinados intervalos de tempo. Na ICP-Brasil o intervalo de tempo consta no documento "Declaração de Práticas de Certificação".

2.7 Conclusão

Após a revisão e estudo dos conceitos relacionados ao trabalho, conclui-se que os conceitos de criptografia simétrica e assimétrica são a base para as tecnologias relacionadas à emissão e uso de certificados digitais.

A infraestrutura de chaves públicas é uma solução amplamente utilizada e aceita para emissão de certificados digitais, motivando o desenvolvimento de soluções baseadas em seu modelo. Os certificados digitais solucionam o problema de autenticação do portador da chave pública, sendo amplamente utilizados em aplicações que utilizam os conceitos de criptografia e autenticação seguras.

Os conceitos apresentados neste capítulo são precursores de vários estudos na área de chaves públicas, novas tecnologias e estudos foram surgindo utilizando como base os conceitos acima, um desses estudos é o objetivo deste trabalho, o certificado de atributo, que será apresentado no próximo capítulo.

3 Fundamentação Teórica Certificado de Atributos

No capítulo anterior foram introduzidos os conceitos básicos de criptografia, certificados digitais e a infraestrutura utilizada para sua emissão. Foi demonstrado o porquê da criação dos certificados digitais de chave pública e o porquê da sua utilização. Neste capítulo será descrito a evolução do certificado de chave pública até o problema que levou a criação do objeto de estudo deste trabalho os certificados de atributos.

A primeira seção descreve o certificado de atributo em si, as informações que contém sua estrutura e qual a sua principal função dentro da infraestrutura de chaves públicas.

A penúltima seção apresenta informações sobre a infraestrutura necessária para realizar a emissão de certificados de atributos com segurança.

A última seção apresenta uma conclusão sobre o certificado de atributos, quais benefícios trazem a infraestrutura de chaves públicas e uso em geral.

3.1 Certificado de Atributos

O certificado digital de chaves públicas possui o objetivo de prover autenticação a chave pública de uma entidade. Utilizando o conceito de emissão junto a uma Autoridade Certificadora é possível atestar a identificação da entidade portadora como possuidora da chave pública e respectiva chave privada.

O uso da ICP levou à necessidade de armazenar outros tipos de informações além da chave pública e identidade do seu titular. Devido a isso as versões mais atuais do padrão X.509 definem uma série de extensões de certificado. A adição de extensões em um certificado digital expande as suas aplicações, trazendo maior utilidade a seu uso, como exemplo de informações adicionadas normalmente pode-se citar papéis que o usuário representa, permissões e informações de autorização. Porém a adição constante

de extensões em um certificado pode distanciá-lo de seu principal objetivo. objetivo que consta em sua recomendação padrão, o de autenticação[9]. A utilização das extensões dos certificados de autenticação para armazenamento de atributos possui efeitos negativos[10], dentre estes os mais importantes são:

- Informações de atributos normalmente possuem tempo de vida mais curto que a validade da identidade e chave pública dos certificados de chave pública. No âmbito ICP-Brasil certificados de chave pública podem chegar a ter três anos de validade. Desta forma se tais informações voláteis são adicionadas em extensões de um certificado de chave pública o tempo de validade do mesmo é diminuído, pois se as informações dos atributos forem alteradas irão requerer a emissão de um novo certificado digital e revogação do certificado antigo. Usando a comparação entre certificado de chaves públicas e atributos podemos comparar o certificado de chaves públicas a um RG que identifica o titular e é valido durante um grande período de tempo e um atributo seria como a permissão para dirigir que possui um tempo de validade e precisa ser renovado em um periodo curto de tempo;
- A Autoridade Certificadora que emite os certificados de chave pública não possui autoridade sobre as informações de atributos a serem adicionados uma vez que esse tipo de informação está muito ligada ao contexto onde se encontra o titular do certificado. Como resultado, as autoridades certificadoras acabam ficando complexas à medida que precisam adicionar informações sobre um conjunto de atributos para adicionar ao certificado de chave pública;

Devido aos fatores acima e a outros problemas de ICP [11], foi desenvolvida uma abordagem alternativa para lidar com informações de autorização. A infraestrutura de gerenciamento de privilégios tem como principal objetivo a autorização independente do processo de autenticação e tem como artefato principal os certificados de atributos X.509 gerados por uma autoridade de atributos.

Os certificados de atributos são certificados associados a entidades para propósitos diferentes ao de autenticação e possuem uma estrutura de codificação própria separada ao certificado de autenticação por convenção e criada com o propósito de conter informações e atributos[9]. Esta separação entre a autenticação e a autorização possui várias vantagens:

- Gerenciamento distribuído de privilégios e atributos. Os atributos podem ser emitidos por autoridades de atributos diferentes, cada uma responsável por informações

de diferentes contextos. Por exemplo, na infraestrutura brasileira o certificado de chaves públicas possui uma raiz central. Mas atributos podem ser emitidos por diferentes autoridades dependendo da diversidade de contextos sociais. Uma autoridade de atributos de um banco pode emitir um certificado de atributo para acessar seus serviços, uma autoridade de atributos de um plano de saúde pode emitir um certificado de atributos com o número do plano, e esta regra se aplica a muitas outras situações que é necessário um artefato para identificar um sujeito ou informação do sujeito;

- Promovem a separação de responsabilidade, já que autoridades de atributos possuem realmente as informações para emissão de seus atributos não sendo então de responsabilidade de autoridades certificadoras obterem estas informações;
- Tempo de vida dos certificados de atributos podem ser menores que os dos certificados de chave pública e possuem mecanismos de revogação separados. Esta propriedade está intimamente ligada à remoção das informações dos atributos das extensões do certificado digital de chaves públicas;

A principal diferença entre os modelos de certificado de atributo e chaves públicas se encontra em sua estrutura ASN.1. Além de prover uma associação entre o certificado de autenticação o certificado de atributo possui em sua estrutura uma sequência de atributos, ver figura.

```
AttributeCertificate ::= SEQUENCE {
    acinfo           AttributeCertificateInfo,
    signatureAlgorithm AlgorithmIdentifier,
    signatureValue   BIT STRING
}

AttributeCertificateInfo ::= SEQUENCE {
    version           AttCertVersion -- version is v2,
    holder            Holder,
    issuer            AttCertIssuer,
    signature         AlgorithmIdentifier,
    serialNumber      CertificateSerialNumber,
    attrCertValidityPeriod AttCertValidityPeriod,
    attributes        SEQUENCE OF Attribute,
    issuerUniqueID    UniqueIdentifier OPTIONAL,
    extensions        Extensions OPTIONAL
}
```

Figura 9: Estrutura ASN.1 de um certificado de atributo

Abaixo é apresentada uma descrição de cada um destes campos:

- **Holder:** Identifica o portador do certificado de atributos. No campo é possível adicionar três tipos de informações distintas sobre o portador PKC[12], exibidas na figura abaixo.

```

Holder ::= SEQUENCE {
    baseCertificateID  [0] IssuerSerial OPTIONAL,
        -- the issuer and serial number of
        -- the holder's Public Key Certificate
    entityName         [1] GeneralNames OPTIONAL,
        -- the name of the claimant or role
    objectDigestInfo   [2] ObjectDigestInfo OPTIONAL
        -- used to directly authenticate the holder,
        -- for example, an executable
}

```

Figura 10: Holder de um certificado de atributo

A opção **baseCertificateID** deve ser utilizada quando o certificado de atributos é utilizado em conjunto com um certificado de chaves públicas. A autoridade certificadora que emite certificados de chaves públicas também faz o papel de autoridade de atributos. Neste caso os campos **serialNumber** e **issuer** devem ser os campos idênticos tanto no certificado de chaves públicas quanto no certificado de atributo.

A opção **entityName** é utilizada quando a autoridade de certificação e a autoridade de atributos são entidades separadas, neste caso o campo **entityName** deve conter o mesmo valor que o campo **subject** do certificado de chaves públicas. Caso contrário deve conter um **distinguished name** que identifique o portador do certificado.

A última opção, **objectDigestInfo** tem como objetivo a associação do certificado de atributo a um hash de um objeto, uma classe Java por exemplo.

Com alguma dessas informações é possível identificar o portador do certificado ICP correspondente ao certificado de atributo.

Pelo menos uma opção deve ser utilizada e deve estar presente. Apesar de ser possível utilizar mais de uma opção a especificação X.509 não recomenda pois não existe uma regra que determine qual opção é mais importante que outra o que pode ocasionar problemas de interpretação do conteúdo do campo.

- **Issuer:** Identifica a autoridade de atributos que emitiu o certificado de atributos;
- **signature:** o campo signature contém o algoritmo de assinatura digital utilizado para assinar digitalmente o certificado;

- **serialNumber:** Número inteiro positivo gerado no momento de emissão do certificado de atributo. Não é necessário que esses números sejam ordenados;
- **attrCertValidityPeriod:** o campo possui a informação quanto ao período de validade do certificado. Especifica quanto tempo a autoridade de atributo garante a associação entre o requerente do certificado e seus atributos;
- **attributes:** campo de principal e de maior importância em um certificado de atributos. O campo é constituído de uma sequência de **Attributes**, sendo os atributos constituídos de um número identificador do atributo, responsável por definir o tipo de atributo e um valor do atributo, que é um ANY ASN.1, podendo então ser qualquer tipo de informação definida pelo número identificador. A figura abaixo demonstra esta estrutura;

```
Attribute ::= SEQUENCE {
    type      AttributeType,
    values    SET OF AttributeValue
    -- at least one value is required
}

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY DEFINED BY AttributeType
```

Figura 11: Estrutura ASN.1 do campo attribute de um certificado de atributo

A especificação do certificado de atributo não impõe restrições quanto a quais atributos serão incorporados à estrutura, sendo responsabilidade da Autoridade de Atributos quais atributos serão emitidos por ela.

- **extensions:** as autoridades de atributos normalmente utilizam extensões para adicionar informações de serviços a serem utilizados para a validação de certificados quanto a revogação, mas também podem utilizar outros tipos de extensão a seu critério.

O certificado de atributo pode ser utilizado em conjunto com o certificado de chaves públicas. O certificado de chaves públicas é utilizado para autenticar o portador da chave pública, após autenticação o certificado de atributos é utilizado para associar atributos à entidade autenticada. Para garantir esta funcionalidade o sistema que irá realizar a verificação deve ter acesso ao certificado de chaves públicas juntamente com o certificado de atributos para garantir que a associação entre os dois é válida através do campo holder.

O certificado de atributos devido à sua característica possui algumas diferenças se comparada à uma infraestrutura de chaves públicas para sua emissão. O próximo capítulo demonstra a infraestrutura necessária para a emissão de certificados de atributos.

3.2 Infraestrutura de Certificado de Atributos

É conhecido que a infraestrutura de chaves públicas é o padrão para emissão de certificados digitais de chaves públicas e o seu padrão mais utilizado é o X.509. O propósito primário de uma infraestrutura de chaves públicas é de prover autenticação a comunicações entre partes através do uso de assinatura digital.

A infraestrutura de chaves públicas é a infra estrutura utilizada para emissão de certificados digitais de chave pública, a IGP ou infraestrutura de gerenciamento de privilégios é a infra estrutura citada na recomendação do padrão X.509 quando se trata de manutenção de certificados de atributos[2].

A IGP é uma infraestrutura muito semelhante à ICP, enquanto que na infraestrutura de chaves públicas existem as ACs, na infraestrutura de autorização existe as Autoridades de Atributos(AA). A raiz de confiança na ICP é a Autoridade Certificadora Raiz, na IGP a raiz de confiança é a Souce of Authority (SOA). Na tabela é demonstrado a comparação entre ICP e IGP.

Conceito	Entidade na ICP	Entidade na IGP
Certificado	Certificado de chave pública	Certificado de atributo
Emissor do certificado	Autoridade Certificadora	Autoridade de Atributo
Portador do certificado	Subject	Holder
Ligação entre certificados	Subject à chave pública	Holder à atributos de privilégio
Revogação	Lista de Certificados Revogados	Lista de certificados de atributos revogados
Raiz de confiança	Autoridade certificadora	Source of Authority
Autoridade Subordinada	Autoridade certificadora intermediária	Autoridade de Atributo

Na infraestrutura de chaves públicas a autoridade raiz de confiança delega a autoridades certificadoras intermediárias o poder para emitir os certificados de chave pública, na infraestrutura de manutenção de privilégio a Source of Authority delega os poderes de autorização a intermediárias chamadas Autoridades de Atributos, responsáveis pela emissão de certificados de atributos.

No caso de necessidade de revogação de um certificado, na infraestrutura de chaves públicas a autoridade certificadora emite lista de certificados revogados. No caso da infraestrutura de manutenção de privilégios caso uma entidade precise ter seus atributos revogados uma lista de certificados de atributos revogados é emitida.

A figura abaixo ilustra o caminho de confiança em uma ICP e em uma IGP.

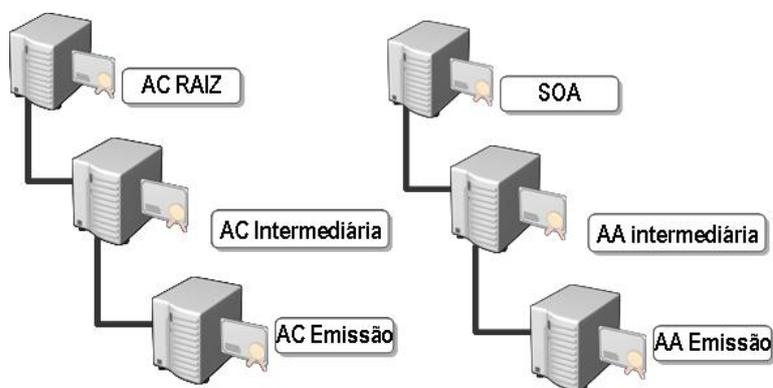


Figura 12: Hierarquia em uma ICP e em uma IGP

Existem atualmente duas formas de emitir Certificados de Atributos[12]:

- Modelo *pull*: No modelo *pull* os certificados são publicados em repositório acessíveis para consulta por partes que desejem "puxar" os certificados. Este modelo geralmente é utilizado por Autoridades de Atributos que não emitem diretamente a usuários, mas sim para que outros softwares acessem os certificados emitidos.
- Modelo *push*: No modelo *push* os certificados são emitidos diretamente aos requerentes do certificado, ficando o certificado sob sua responsabilidade.

3.3 Conclusão

Após o levantamento e revisão dos conceitos sobre certificados de atributos conclui-se que as extensões dos certificados de chaves pública por mais que aumentem em muito o uso dos certificados, não são a melhor solução para portar informações voláteis sobre o seu portador.

Os certificados de atributos por possuir uma infraestrutura a parte da autoridade de certificadora de chaves públicas são de alta escalabilidade, podendo funcionar nos mais diversos escopos sociais de uso. Qualquer organização pode implementar a sua autoridade de atributos sem a necessidade de alterar o funcionamento de sua autoridade de chaves públicas e tudo que a tange. Bem como possuir uma autoridade de atributos sem a necessidade de possuir uma autoridade de chaves pública.

A não necessidade de um par de chaves é uma característica importante de um certificado de atributos, devido a esta característica não existe necessidade de senha, ficando a cargo do portador a sua publicação ou uso.

É quando utilizados em conjunto, um certificado de chaves pública e certificados de atributos, que é possível verificar as maiores contribuições de ambas as tecnologias. O certificado de chaves públicas não corre risco de uma revogação prematura devido a alterações de informações de seu portador, o que garante a segurança de seu par de chaves quanto a um futuro indeterminado, deixando livre para o certificado de atributos a atualização e gestão das informações de seu portador. Esta característica garante segurança ao usuário e a autoridade de atributos a livre atualização de atributos.

4 *Ferramentas e Tecnologias de Desenvolvimento*

Nesta parte do trabalho, serão descritas as ferramentas e tecnologias utilizadas bem como as justificativas para sua utilização. Tal seção tem por objetivo dar fundamentação para o entendimento da aplicação desenvolvida apresentada nos próximos capítulos.

Nos próximos capítulos será descrita as tecnologias e bibliotecas utilizadas para o desenvolvimento da aplicação exemplo.

4.1 **Plataforma Java Enterprise Edition**

JavaEE (Java Platform, Enterprise Edition) é um padrão de indústria para o desenvolvimento portátil, robusto e seguro de aplicações no lado do servidor. JavaEE prove WebServices, modelos de componentes, gerenciamento e APIS de comunicação para produção e implementação de SOA (Arquitetura Orientada a Serviços)[13].

A arquitetura JavaEE facilita o desenvolvimento web. O Enterprise Java Beans (EJB), não exige definições de interface, pois a própria linguagem java já possui interface, eliminando as complexidades de mapeamento e imperfeições entre uma linguagem de programação e uma sintaxe de interface "neutra". Não existe a necessidade de configuração ou inicialização de um middleware [14].

Inicialmente, o maior problema do J2EE, mais especificamente com o EJB, era o de não ser neutro em relação a linguagens de programação. Tanto CORBA como COM permitem a interoperabilidade entre C, C++, java, VB, etc, já o EJB inicialmente não tinha interoperabilidade com outras linguagens, pois utilizava o RMI para realizar a comunicação, um formato muito ligado ao java.

Com a introdução do RMI/IIOP uma implementação alternativa do RMI o EJB passou a se utilizar do IIOP, o protocolo de transporte do CORBA, permitindo assim a

interoperabilidade com aplicações CORBA escritas em outras linguagens.

Grande parte a infra-estrutura do EJB foi baseada no padrão CORBA, o EJB depende de um ORB completo para se comunicar, a grande inovação do JavaEE foi ser uma plataforma totalmente gerenciada graças a máquina virtual java, com o byte code portátil e a GC (Garbage Collection) que realiza o gerenciamento automático da memória , bem como os princípios de segurança da máquina virtual.

4.2 Webservices

Webservices são o que há de mais recente em desenvolvimento de aplicações que efetuam chamadas de procedimentos remotos, inicialmente especificado pela microsoft, vêm atraindo interesse de vários desenvolvedores e pesquisadores em todas as plataformas de programação. Os webservices possuem um conceito muito simples, compor RPCs (Chamadas de procedimentos Remotos) pela internet ou por uma rede. Os webservices não são a primeira arquitetura a permitir isto, mas possuem uma diferença em relação as tecnologias existentes atualmente pois trabalham com padrões neutros de plataforma como o XML e HTML, o cliente só precisa saber o endereço onde está sendo servido o serviço e os tipos de dados utilizados nas chamadas de procedimentos remotos, não importando a linguagem de programação utilizada ou tipo de plataforma em que o mesmo está escrito, como Java em linux, ou um serviço Asp.net no Windows[15].

Como os Webservices são totalmente baseados em XML, e o XML é facilmente transportado pelo protocolo padrão da Web, o http, se resolvem vários problemas do RPC tradicional, a interoperabilidade. São utilizados ainda outros padrões como WSDL.

4.2.1 Arquitetura dos Webservices

Webservices estende o conceito de WWW (World wide Web) provendo meios de um software conectar outro software. Para prover esta interoperabilidade webservices utilizam em sua arquitetura formatos como HTTP, XML e SOAP[16].

Para entender mais sobre a arquitetura webservices é necessário conhecer cada um dos formatos utilizados.

4.2.2 Linguagem de marcação XML

O XML, eXtensible Markup Language, é uma linguagem de marcação padronizada, muito utilizada atualmente para facilitar a interoperabilidade entre sistemas em arquiteturas distintas.

As linguagens de marcação têm como maior característica o uso de tags (marcadores) para estruturar as informações e dados. Com seu uso as aplicações que interpretam as tags podem executar e demonstrar as informações aos clientes, como os navegadores web, que interpretam as tags HTML dos documentos e as exibem para o usuário.

As linguagens de marcação tiveram a sua origem com o SGML (Standard Generalized Markup), que surgiu na década de 70, o SGML é um padrão internacional desde 1986. Ela é muito bem servida em termos de suporte e documentação. É uma linguagem poderosa, mas muito complexa, tornando-se árdua a tarefa de incorporá-la a uma aplicação, só para termos uma idéia a especificação do SGML possui mais de 500 páginas.

O Como o SGML, o HTML (HyperText Transfer Protocol) possui um número fixo de tags. Em função da necessidade cada vez maior de recursos mais complexos surgiu então o XML. No XML cada aplicação pode elaborar suas próprias tags, tendo assim uma flexibilidade muito maior que no HTML. O XML, também torna possível que o documento seja processado automaticamente por aplicações diferentes, não só por navegadores, no caso do HTML.

Um documento XML, num primeiro momento, pode parecer muito semelhante a um documento HTML, pois ambos possuem as tags procedidas do símbolo «"e finalizadas com »". Porém, a maior diferença é que com o XML tem se a liberdade de criar tags, já as tags HTML são pré-definidas.

4.2.3 Protocolo de transmissão SOAP

O Simple Object Access Protocol (SOAP) é um mecanismo para comunicação que permite trocar informações em um ambiente descentralizado e distribuído. O SOAP possui mecanismo de codificação dos tipos de dados inclusos nos pacotes sendo transportados, bem como o empacotamento de informações. No SOAP os dados são transportados em forma de texto codificados no formato XML, o que facilita a interação entre sistemas heterogêneos. Os dados codificados em texto são transportados utilizando os protocolos da Web como o HTTP, SMTP, ou FTP.

Existem vantagens do SOAP em relação aos outros protocolos. Como é transportado pelo protocolo HTTP, acaba sendo um protocolo que navega sem precisar de configurações e portas específicas entre os servidores e firewalls.

A estrutura do SOAP é melhor visualizada na figura abaixo:

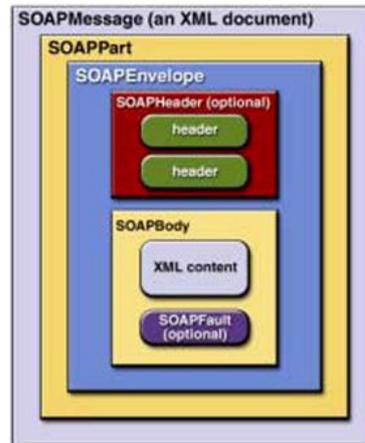


Figura 13: Estrutura do pacote de transmissão SOAP

4.2.4 Linguagem de descrição de Serviço WSDL

A WSDL (Webservices Description Language) é uma linguagem baseada em XML cuja função é descrever Webservices, definindo o que o serviço pode realizar, como localizá-lo e como invocá-lo [17]. É constituída de duas partes, uma mais abstrata que identifica um conjunto de operações e mensagens, e uma parte mais concreta contendo informações para localização na rede e um protocolo de integração das operações [18].

"Definições de Webservices podem ser mapeadas para qualquer linguagem, plataforma, modelo de objetos, ou sistema de mensagens. Simples extensões à infra-estrutura existente da Internet podem implementar Webservices para interação via browsers ou diretamente dentro de uma aplicação. A aplicação poderia ser implementada usando COM, JMS, CORBA, COBOL ou uma variedade de soluções de integração proprietárias. Desde que o solicitante e o provedor concordem na descrição do serviço (o arquivo WSDL), as implementações por trás dos Webservices podem ser qualquer coisa"[18].

4.3 Bouncy Castle API de Criptografia para Java

Java possui um vasto leque de tecnologias e APIs de apoio, uma dessas tecnologias é o Java SE Security ou Java Security. Java Security inclui uma grande quantidade de APIs, ferramentas e implementação de algoritmos, mecanismos e protocolos mais comuns. Java Security possui ferramentas em diversas áreas da segurança como criptografia, infra-estrutura de chaves públicas, comunicação, autenticação, etc [19].

Um dos subcomponentes do pacote de segurança do Java, especificamente do pacote de criptografia é o JCA, "Java Cryptography Architecture". O JCA contém provedores de criptografia e APIs para assinatura digital, extração de resumo criptográfico, certificados, verificação de certificados e outros [20].

A arquitetura JCA é desenvolvida seguinte 3 princípios fundamentais[20]:

- Implementação independente: aplicações não precisam implementar algoritmos seguros. Basta requisitar aos serviços de criptografia da plataforma Java;
- Interoperabilidade de implementação - Provedores são interoperáveis com quaisquer aplicações. Provedores são independentes de aplicações;
- Extensão de algoritmos - A plataforma de segurança Java prove suporte a instalação de provedores de desenvolvedores independentes;

The Legion of Bouncy Castle é uma iniciativa código aberta que prove um provedor de criptografia JCA, podendo então ser utilizado em conjunto com a plataforma Java.

A api de criptografia da Bouncy Castle consiste dos seguintes pacotes principais[21]:

- Provedor de criptografia JCA;
- Biblioteca de leitura e escrita ASN.1;
- Geradores de certificados X509 versão 3, CRLs versão 2 e arquivos PKCS#12;
- Geradores de certificados de atributos X509 versão 2;

4.4 Conclusão

A plataforma Java EE é uma plataforma robusta, inteligente e organizada, prove com segurança tecnologias para interoperabilidade de sistemas, no caso EJB e Webservices quanto APIs de criptografia, o que a torna a escolha certa para o desenvolvimento de uma aplicação robusta e expansível.

Webservices é uma tecnologia de chamada de procedimentos remotos sofisticada e independente de plataforma o que torna seu uso conveniente em arquiteturas em que existe a necessidade de diversos sistemas interoperarem. No caso de uma Autoridade de Atributos poder vir a ser acessada por diferentes arquiteturas e sistemas, o Web service certamente é a melhor escolha.

A biblioteca de apoio Bouncy Castle possui dentre seus pacotes geradores de certificados de atributos e certificados de chave pública, o que diminui significativamente o tempo de desenvolvimento da aplicação.

No próximo capítulo será demonstrada uma visão geral sobre a arquitetura da aplicação protótipo e como a mesma utiliza as tecnologias.

5 *Visão Geral da Aplicação*

No capítulo anterior foram identificadas as ferramentas e tecnologias utilizadas e a justificativa que levaram a escolha de tais ferramentas no desenvolvimento da aplicação protótipo.

Este capítulo tem como objetivo definir que funcionalidades a aplicação protótipo deve prover, serão discutidos e especificados em detalhes cada uma das suas funcionalidades, dando assim uma visão geral de como irá se portar a aplicação e quais são os seus objetivos.

A primeira parte descreve os requisitos de negócio envolvidos com a geração e revogação de certificados de atributos. Nesta parte identificam-se os processos realizados pelo sistema, bem como os atores envolvidos nestes processos.

Na segunda parte os processos realizados pelo sistema são mapeados em casos de usos.

Na terceira e quarta parte é demonstrado os requisitos de sistema e os requisitos não funcionais, identificados conforme os casos de uso previamente identificados.

5.1 **Levantamento e Análise de Requisitos**

Uma das primeiras fases de engenharia de um software consiste no levantamento de requisitos. Nesta etapa, o engenheiro de software busca compreender as necessidades do usuário e o que ele deseja que o sistema a ser desenvolvido realize[22].

Os requisitos de negócios correspondem aos objetivos que o sistema deve prover para poder satisfazer aos negócios do cliente. O usuário ou cliente possui um problema que deve ser atendido pelo sistema.

Logo após o levantamento dos requisitos, passa-se à fase em que as necessidades apresentadas pelo cliente são analisadas; esta etapa é conhecida como Análise de Requisitos, onde o engenheiro examina os requisitos enunciados pelos usuários, verificando se estes foram bem compreendidos. A partir da etapa de Análise de Requisitos são determinadas

as reais necessidades do sistema de informação [22].

Na infra-estrutura de chaves públicas é comum identificar claramente dois processos de negocio, registro e emissão de certificados e emissão de Lista de Certificados Revogados.

No primeiro processo é comum identificar duas entidades, o cliente a quem será emitido o certificado e a Autoridade de Atributos. Em alguns casos as Autoridades Atributos podem ser substituídas por Autoridades de Registro, na execução de uma primeira função de registro do cliente junto à própria autoridade.

No segundo processo a própria Autoridade Atributos é responsável e possui autonomia para emissão das listas de certificados revogados em área pública para consulta.

Nos próximos capítulos será apresentado com maiores detalhes cada um dos processos.

5.1.1 Processo de registro e emissão de certificados

Uma Autoridade de Atributos é responsável por emitir corretamente certificados e lista de certificados revogados [8].

O processo de registro e emissão de certificados tem como objetivo atestar que a entidade requerente possui as qualificações necessárias que possibilite a emissão de um certificado de atributo.

As qualificações exigidas são dependentes de cada uma das Autoridades de Atributos, de suas práticas e políticas para emissão destes atributos.

Na ausência de uma autoridade de registro a própria Autoridade de Atributos realiza a autenticação da entidade e verifica se as qualificações exigidas foram cumpridas. Caso as obrigações tenham sido cumpridas a Autoridade Atributos dá prosseguimento a emissão do certificado digital a entidade requerente.

No fluxo onde há uma autoridade de registro, a mesma realiza as verificações sobre a entidade, caso as obrigações tenham sido cumpridas corretamente envia então um atestado a Autoridade de Atributos, permitindo que a entidade possa então emitir um certificado na mesma.

No escopo do presente trabalho assume-se que as obrigações que a entidade deve cumprir é possuir um certificado de chaves públicas emitidos por uma autoridade de certificação confiável no âmbito ICP-Brasil.

Como é possível a um certificado de atributo possuir qualquer estrutura como atributo,

foi restringida a termo deste protótipo a emissão de atributos semelhantes aos atributos ICP-Brasil encontrados nos certificados de chave pública, constituindo de um número identificador OID e um valor, sendo o valor um array de caracteres ou binários.

5.1.2 Processo de Emissão de lista de Certificados Revogados

Além de emitir certificados digitais uma Autoridade de Atributos deve emitir Lista de Certificados Revogados, onde constam quais certificados que a Autoridade deixou de confiar.

Para que um certificado conste em uma LCR, a entidade o qual o certificado é de posse deve procurar a autoridade que o emitiu e gerar uma requisição de revogação de certificado, nesta requisição deve constar o motivo para a revogação do certificado. Comumente a Autoridade Certificadora fornece um serviço on-line para requerer a revogação de um certificado, bastando utilizar o serviço para realizar a revogação do certificado.

5.2 Requisitos de Negócio

Os requisitos de negócios correspondem aos objetivos que o sistema deve prover para poder satisfazer aos negócios do cliente. O usuário ou cliente possui um problema que deve ser atendido pelo sistema.

Identificados os processos de negócio é possível verificar que o protótipo deve emitir certificados de atributos e lista de certificados revogados e prover uma interface para que o usuário utilize destes serviços e uma interface de administração da autoridade de atributos.

De posse dos requisitos de negócio, o que deve ser efetivamente entregue ao cliente, é gerado os requisitos de sistema. Requisitos de sistema têm um caráter mais técnico e consiste da descrição detalhada de cada uma das ações que o sistema deve executar para cumprir os requisitos de negócio.

5.3 Requisitos Funcionais

Os requisitos funcionais foram divididos em três partes de acordo com os requisitos de negócio levantados. São as seguintes partes: requisitos funcionais da autoridade de atributos, requisitos funcionais de administração e requisitos de módulo público.

5.3.1 Requisitos funcionais da Autoridade de Atributos

São os seguintes requisitos:

- RF1 Emitir certificados de atributos de acordo com o padrão X.509v2 para certificados de atributos utilizando o modo *push*;
- RF2 Emitir lista de certificados de atributos revogados padrão X.509.
- RF3 Verificar se o certificado do requisitante é válido;
- RF4 Verificar se a autoridade emissora do certificado do requisitante é confiável;
- RF5 Construir a cadeia de certificação do certificado do requisitante;
- RF6 Persistir todas as requisições enviadas a autoridade
- RF7 Persistir todos os procedimentos de emissão de certificados de atributos para fins de auditoria;
- RF8 Publicar informações de certificados emitidos
- RF9 Publicar Lista de certificados revogados

5.3.2 Requisitos funcionais de administração

São os seguintes requisitos:

- RF10 Alterar as configurações dos procedimentos de emissão de certificado, como data de validade, algoritmo de assinatura, etc.;
- RF11 Adicionar autoridades certificadoras de chave pública confiáveis;
- RF12 Adicionar atributos a serem emitidos
- RF13 Alterar as configurações de emissão de LCR;

5.3.3 Requisitos funcionais de módulo público

São os seguintes requisitos:

- RF14 Permitir emitir requisições de certificado à autoridade de atributos

- RF15 Permitir emitir certificado de atributo junto à autoridade de atributos
- RF16 Permitir revogar certificado de atributo junto à autoridade de atributos
- RF17 Permitir consultar LCRs emitidas pela autoridade de atributos
- RF18 Permitir consultar certificados emitidos pela autoridade de atributos

5.4 Casos de Uso

O diagrama de casos de uso é o diagrama mais geral e informal da UML, apresenta uma linguagem bem simples e procura identificar os atores (usuários, outros sistemas e hardwares), que utilizarão o sistema. O diagrama de caso de usos é base para outros diagramas e é consultado durante toda a fase de modelagem de um sistema [22].

A figura abaixo ilustra o diagrama de casos de uso gerado.

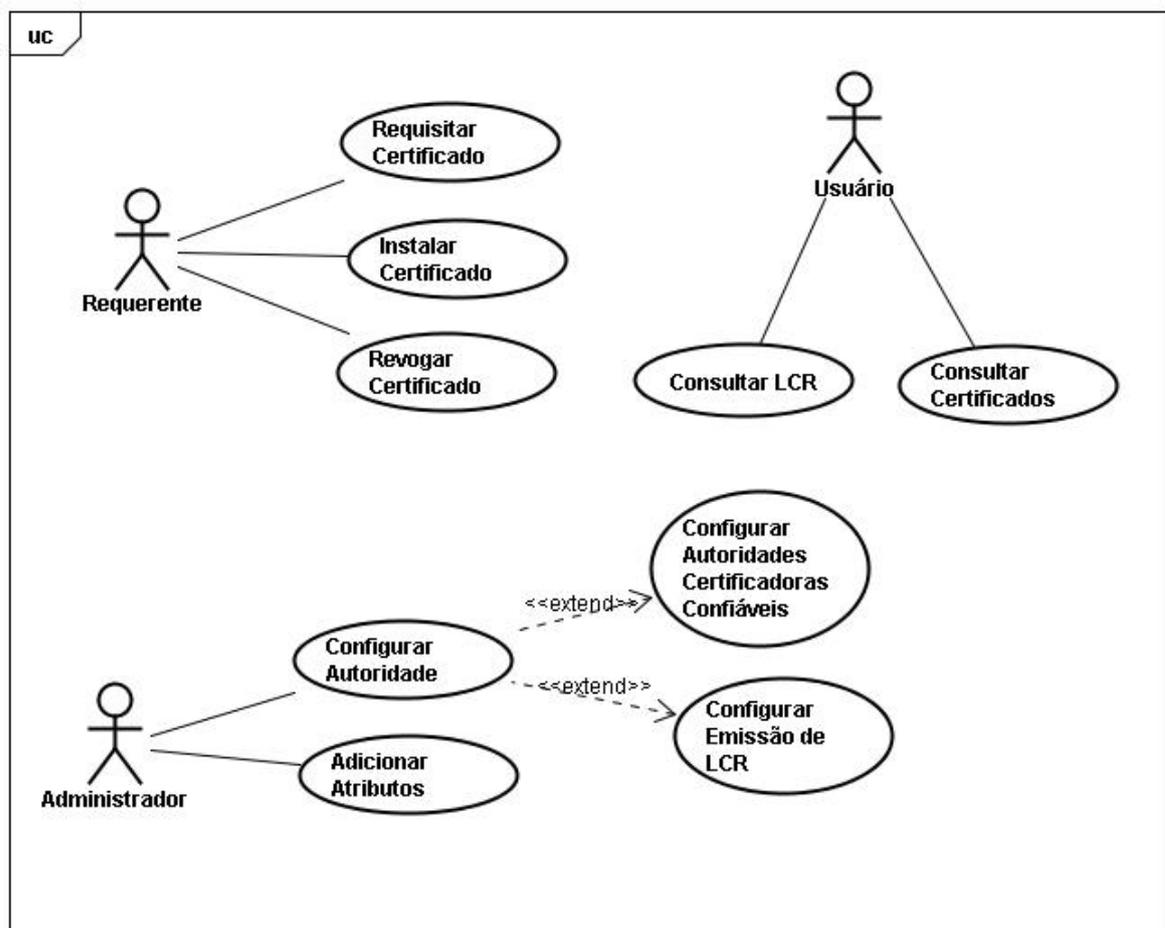


Figura 14: Diagrama de caso de usos

5.4.1 Atores do Sistema

Através dos requisitos levantados e dos casos de uso foram identificados três tipos de atores que irão consumir os serviços disponibilizados pela Autoridade de Atributos. São cada um deles:

- **Requerente:** Entidade que consome os serviços de emissão de certificados de atributos disponibilizados pela autoridade de atributos. Realiza o registro na autoridade, instalação do certificado e revogação de certificado.
- **Administrador:** Entidade responsável pela manutenção das configurações da autoridade certificadora. É responsável por cadastrar atributos, certificados de autoridades de chaves públicas confiáveis, e configuração de emissão de LCRs.
- **Usuário:** Consome serviços de consulta à autoridade certificadora, acessa o repositório público em busca de informações de revogação e de certificados emitidos. Não consome serviços de emissão. Procura a autoridade de atributos em busca de informações.

5.4.2 Detalhamento dos casos de uso

Com os requisitos levantados foram identificados nove casos de uso detalhados a seguir.

UC1	Requisitar certificado
Requisitos funcionais	RF14, RF3, RF4, RF5, RF6
Atores	Requerente
Objetivo	Enviar requisição de certificado de atributo para a autoridade de atributos
Pré-requisito	possuir um certificado de chaves públicas
Fluxo Principal	<p>Requisitar certificado:</p> <ol style="list-style-type: none"> 1. O requerente inicia o caso de uso selecionando a opção de requisição de certificado na tela do módulo público do sistema. 2. Uma nova tela é apresentada onde o requerente deve selecionar o certificado de chave pública. 3. O requerente após seleção do certificado seleciona a opção de enviar. 4. O requerente recebe após o envio em caso de aceitação pelo servidor da requisição um número de identificação da requisição para posterior emissão.
Fluxos Alternativos	<ul style="list-style-type: none"> • Certificado não selecionado pelo usuário • Autoridade certificadora emissora do certificado selecionado não é confiável pela autoridade de atributos

5.5 Requisitos Não Funcionais

Os requisitos funcionais são requisitos relacionados com o ambiente da aplicação, podendo ou não influir diretamente nos requisitos funcionais.

O requisitos não funcionais são os seguintes:

- Persistencia de dados utilizar banco de dados MySQL.
- Liguagem Java como linguagem de pragramação utilizada para construção dos Web-Services.
- Linguagem VB.Net para implementação de interface de administração;
- Asp.Net como plataforma de desenvolvimento do módulo público;
- Sistema operacional Windows;
- servidor web Tomcat versão 5 ou superior para módulo Webservices
- servidor web IIS 5 ou superior para módulo público
- JavaEE 5 ou superior

UC2	Instalar certificado
Requisitos funcionais	RF15, RF7, RF1
Atores	Requerente
Objetivo	Enviar número de identificação para a autoridade certificadora e emitir o certificado de atributos
Pré-requisito	possuir o número de identificação de requisição emitido pela autoridade de atributos
Fluxo Principal	<p>Instalar certificado:</p> <ol style="list-style-type: none"> 1. O requerente inicia o caso de uso selecionando a opção de emitir certificado na tela do módulo público do sistema. 2. Uma nova tela é apresentada onde o requerente deve adicionar o número de identificação exigido. 3. O requerente envia o número para a autoridade selecionando a opção enviar. 4. O requerente recebe após o envio um link para a instalação do certificado.
Fluxos Alternativos	<ul style="list-style-type: none"> • Requerente possui um número de identificação inválido

UC3	Revogar certificado
Requisitos funcionais	RF16, RF2
Atores	Requerente
Objetivo	Enviar número de identificação para a autoridade certificadora e revogar o certificado de atributos
Pré-requisito	possuir o número de identificação de requisição emitido pela autoridade de atributos
Fluxo Principal	<p>Revogar certificado:</p> <ol style="list-style-type: none"> 1. O requerente inicia o caso de uso selecionando a opção de revogar certificado na tela do módulo público do sistema. 2. Uma nova tela é apresentada onde o requerente deve adicionar o número de identificação exigido e descreve o motivo da revogação do certificado. 3. O requerente envia a requisição de revogação para a autoridade selecionando a opção enviar. 4. O requerente recebe após o envio um aviso de revogação do certificado.
Fluxos Alternativos	<ul style="list-style-type: none"> • Requerente possui um número de identificação inválido

UC4	Consultar LCR
Requisitos funcionais	RF17, RF9
Atores	Usuário
Objetivo	Consultar a autoridade de atributos em busca de LCRs por ela emitida
Pré-requisito	N.A
Fluxo Principal	<p>Consultar LCR:</p> <ol style="list-style-type: none"> 1. O usuário inicia o caso de uso selecionando a opção de pesquisa de LCR. 2. Uma nova tela é apresentada onde o usuário pode aplicar filtros de pesquisa. 3. O usuário envia as opções de busca para a autoridade certificadora selecionando a opção enviar. 4. A busca retorna ou não LCRs encontrados e exibe os mesmos para o usuário.
Fluxos Alternativos	<ul style="list-style-type: none"> • Usuário seleciona opções de filtro incorretas.

UC5	Consultar certificados
Requisitos funcionais	RF18, RF8
Atores	Usuário
Objetivo	Consultar a autoridade de atributos em busca dos certificados por ela emitida
Pré-requisito	N.A
Fluxo Principal	<p>Consultar certificados:</p> <ol style="list-style-type: none"> 1. O usuário inicia o caso de uso selecionando a opção de pesquisa de certificados. 2. Uma nova tela é apresentada onde o usuário pode aplicar filtros de pesquisa. 3. O usuário envia as opções de busca para a autoridade certificadora selecionando a opção enviar. 4. A busca retorna ou não certificados encontrados e exibe os mesmos para o usuário.
Fluxos Alternativos	<ul style="list-style-type: none"> • Usuário seleciona opções de filtro incorretas.

UC6	Configurar autoridades certificadoras confiáveis
Requisitos funcionais	RF11
Atores	Administrador
Objetivo	Configurar a Autoridade de atributos a somente aceitar os certificados de chaves pública emitidos por autoridades previamente cadastradas
Pré-requisito	Possuir o arquivo dos certificados da autoridade a ser cadastrada
Fluxo Principal	<p>Configurar autoridades certificadoras confiáveis:</p> <ol style="list-style-type: none"> 1. O administrador inicia o caso de uso executando a aplicação de administração da Autoridade Certificadora. 2. Dentro da tela da aplicação o administrador aciona o menu de configurações e a opção de configurações de certificados. 3. O administrador seleciona o certificado a ser cadastrado e submete o mesmo a autoridade certificadora.
Fluxos Alternativos	<ul style="list-style-type: none"> • Administrador seleciona um certificado codificado incorretamente. • Administrador seleciona um certificado não ICP-Brasil.

UC7	Configurar emissão de LCR
Requisitos funcionais	RF13
Atores	Administrador
Objetivo	Configurar na autoridade de atributos o período de tempo para emissão da LCR e também o local em disco onde será salva a nova LCR emitida
Pré-requisito	N.A
Fluxo Principal	<p>Configurar emissão de LCR:</p> <ol style="list-style-type: none"> 1. O administrador inicia o caso de uso executando a aplicação de administração da Autoridade Certificadora. 2. Dentro da tela da aplicação o administrador aciona o menu de configurações e a opção de configurações de LCR. 3. Na aba LCR o administrador pode executar duas alterações distintas ou em conjunto: <ul style="list-style-type: none"> • Alterar as configurações de período de emissão de LCR • Alterar o local em disco onde será salva a LCR emitida
Fluxos Alternativos	<ul style="list-style-type: none"> • Administrador seleciona um local em disco inválido.

UC8	Configurar autoridade
Requisitos funcionais	RF10
Atores	Administrador
Objetivo	Alterar configurações gerais da autoridade de atributos
Pré-requisito	N.A
Fluxo Principal	<p>Configurar autoridade:</p> <ol style="list-style-type: none"> 1. O administrador inicia o caso de uso executando a aplicação de administração da Autoridade Certificadora. 2. Dentro da tela da aplicação o administrador aciona o menu de configurações. 3. O administrador seleciona uma opção de configuração desejada. 4. O administrador altera as configurações e submete as alterações para a autoridade certificadora
Fluxos Alternativos	<ul style="list-style-type: none"> • Administrador cancela alterações.

UC9	Adicionar atributos
Requisitos funcionais	RF12
Atores	Administrador
Objetivo	Adicionar novos atributos a serem emitidos pela autoridade de atributos
Pré-requisito	N.A
Fluxo Principal	<p>Configurar autoridade:</p> <ol style="list-style-type: none"> 1. O administrador inicia o caso de uso executando a aplicação de administração da Autoridade Certificadora. 2. Dentro da tela da aplicação o administrador aciona o menu de configurações. 3. O administrador seleciona a opção de configuração de atributos. 4. O administrador altera as configurações da aba e submete as alterações para a autoridade certificadora
Fluxos Alternativos	<ul style="list-style-type: none"> • Administrador cancela alterações. • Administrador submete atributos incorretos.

- .Net Framework 2.0 ou superior

5.6 Módulos da Aplicação

A Autoridade de Atributos pode ser dividida nos seguintes módulos de acordo com as funcionalidades:

- Módulo Público
- Módulo Administração
- Módulo Principal
 - Submódulo Persistencia de dados
 - Submódulo lógica e criptografia
 - Submódulo Webservices

Melhor visualizado no diagramas de componentes acima:

O módulo público constitui-se de uma interface gráfica e lógicas de negócio programadas utilizando a linguagem vb.net com o principal objetivo de auxiliar o requerente a requisitar um certificado e emitir um certificado junto à autoridade de atributo.

O módulo de administração constitui-se de uma interface web e lógica de negócio programada utilizando a linguagem ASP.net com o objetivo de auxiliar o administrador a configurar o funcionamento da autoridade certificadora.

O módulo principal é programado em Java e condensa toda a lógica de emissão de atributos pela autoridade de atributos. O módulo é subdividido em três submódulos especializados e com características próprias. O submódulo de persistência é responsável por registrar em banco de dados, todas as ações que acontecem na autoridade de atributos, entre elas, as informações de emissão, requisição e configurações da autoridade de atributos. O submódulo de persistência é acessado diretamente pelo submódulo de lógica e criptografia. O submódulo de lógica e criptografia é como o controle da autoridade de atributos e possui toda a lógica de controle do sistema, como rotinas de validação das requisições encaminhadas a autoridade, rotinas de acesso a banco de dados, rotinas de configurações e rotinas de emissão dos certificados de atributos, é o submódulo da inteligência da autoridade certificadora. O submódulo de lógica e criptografia é acessado

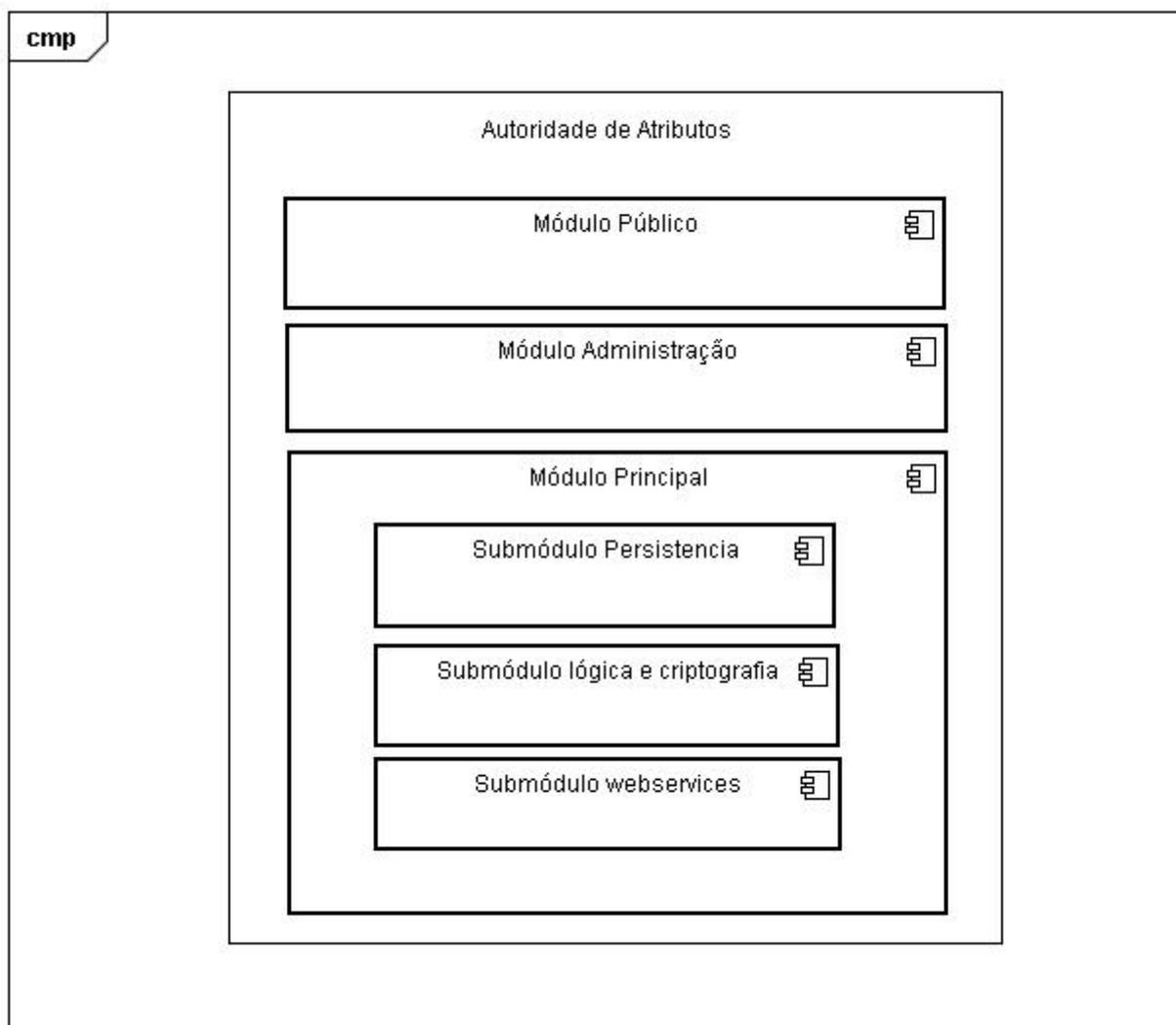


Figura 15: Módulos da Autoridade de Atributos

diretamente pelo submódulo de Webservices. O submódulo de Webservices disponibiliza uma interface para acesso as funcionalidades providas pelo submódulo lógica e criptografia e pela autoridade certificadora, somente pelos serviços publicados é possível interagir com as funcionalidades inteligentes da autoridade certificadora.

É importante ressaltar que o uso de webservices como protocolo de troca de informações implica no uso de um canal seguro evitando assim ataques as informações que trafegam entre os módulos. Embora este trabalho não apresente nenhum mecanismo de segurança, sugere-se o uso de canal SSL para transmissão de dados de forma mais segura.

Definidos todos os requisitos e módulos do sistema o próximo passo é a apresentação da aplicação desenvolvida e sua implementação que será detalhado no próximo capítulo.

5.7 Conclusão

Através das definições dos requisitos de negócio do sistema fica muito mais fácil o levantamento dos requisitos funcionais que o sistema deve atender. Com os requisitos de negócio é possível obter uma abstração de maior nível de que funcionalidades específicas deve-se esperar de um sistema.

Os requisitos funcionais juntamente com os casos de uso detalhados norteiam todo o desenvolvimento de um software, eles possuem uma linguagem simples e são capazes de suprir dúvidas que possam aparecer durante o seu desenvolvimento. Nos casos de usos e requisitos estão detalhados e especificados os requisitos de negócio. Com os requisitos e casos de uso atendidos os requisitos de negócio são atendidos em consequência.

A subdivisão em módulos facilita o desenvolvimento do software podendo então o software ser construído por etapas ou iterações e essas iterações serem responsáveis pelo desenvolvimento de cada um dos módulos do sistema.

6 Implementação da Autoridade de Atributos

Este capítulo tem por objetivo apresentar o desenvolvimento do protótipo, desenvolvimento este orientado pelos requisitos e casos de uso levantados no capítulo anterior. Será demonstrado o desenvolvimento de cada um dos módulos do sistema e como as ferramentas e tecnologias detalhadas no capítulo 4 foram utilizadas neste desenvolvimento.

Neste capítulo também serão demonstrados certificados de atributos emitidos com a autoridade de atributos totalmente concluída. Para isto será utilizada uma ferramenta de visualização ASN.1, esta ferramenta torna muito mais amigável as visualizações de estruturas ASN.1

6.1 Módulo Principal e Webservices

O módulo principal foi escolhido como o primeiro a ser desenvolvido devido a sua maior importância dentro do desenvolvimento da Autoridade de Atributos em si. O módulo principal possui toda a lógica de emissão e persistência e também os webservices que provêm acesso as funcionalidades da Autidade de Atributos.

O módulo principal foi desenvolvido em java utilizando a ferramenta ou IDE de desenvolvimento Eclipse. Na figura abaixo pode-se ver o Eclipse juntamente com o módulo principal e seus submódulos.

O módulo principal foi dividido dentro do ambiente Eclipse em três projetos paralelos, sendo que cada um destes projetos contempla cada um dos submódulos do módulo principal, estes projetos são:

- **Projeto ACA-BL** contempla classes responsáveis por toda a parte inteligente do sistema. Possui classes para a manipulação e geração dos certificados, cadastros de atributos, configuração da autoridade e emissão de lista de certificados revogados.

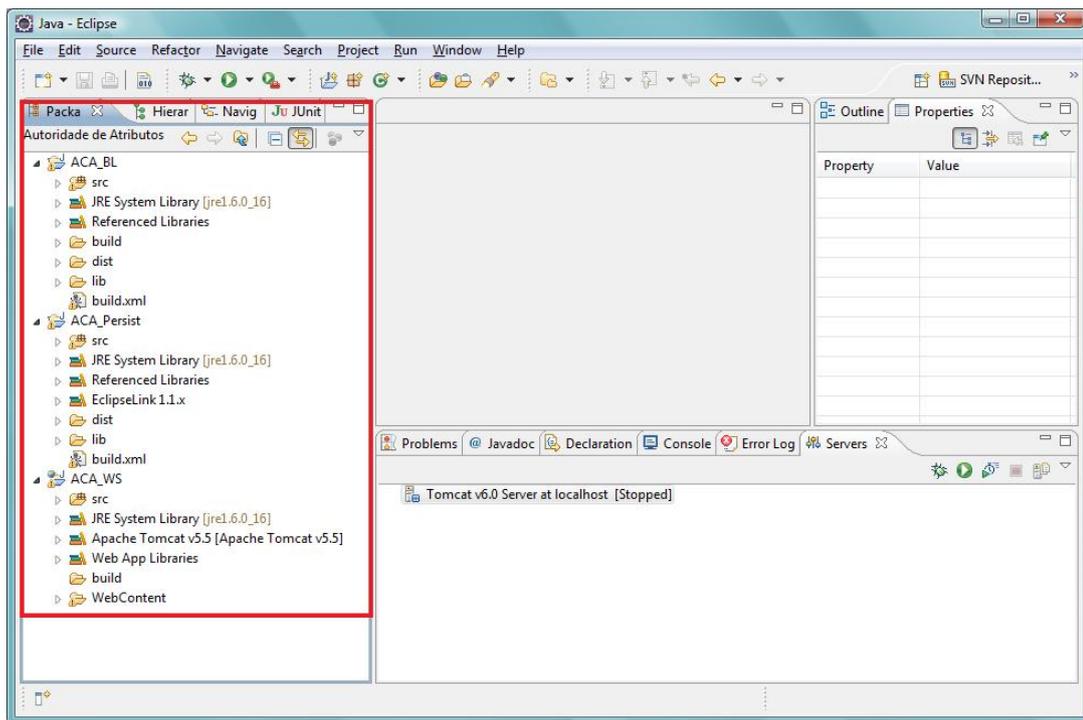


Figura 16: IDE Eclipse utilizada no desenvolvimento da Autoridade de Atributos

Este projeto acessa diretamente as classes de persistência.

- **Projeto ACA-Persist** contempla todas as classes de acesso direto a banco de dados. Possui classes que salvam, atualizam e recuperam informações do banco de dados.
- **Projeto ACA-WS** projeto web que disponibiliza webservices para acesso direto as funcionalidades da Autoridade de Atributos. Os webservices são a ligação entre o mundo externo e a Autoridade de Atributos.

O primeiro projeto criado foi o ACA-BL, isto foi devido a necessidade de desenvolvimento e implementação das funções básicas de emissão de certificado de atributos. Para obter o conhecimento necessário para o desenvolvimento das funções de emissão foi utilizada a documentação oficial da biblioteca de criptografia e certificação digital Bouncy Castle[23]. A documentação completa da Bouncy Castle possui tutoriais, exemplos e a API de desenvolvimento em formato HTML com uma descrição completa da assinatura de classes e métodos disponibilizados por sua API.

Após a obtenção dos conhecimentos básicos necessários ao desenvolvimento foi criada a infraestrutura mínima necessária para o desenvolvimento do projeto e sua codificação isto é a criação propriamente dita das infraestruturas de pacotes, bibliotecas de dependência e

arquivos de configuração. Para isto foi criado um novo projeto na IDE Eclipse. Em termos simplificados basta acessar os menus da ferramenta e criar um novo projeto, informando dados mínimos e solicitar a geração do projeto e o próprio Eclipse cria a infraestrutura básica para início do desenvolvimento do projeto.

Com o projeto no eclipse iniciado foi adicionado o JAR¹ da biblioteca de dependência do projeto, o jar "bcprov-jdk6-141.jar"[24] da Bouncy Castle.

Configurado o Eclipse e adicionado a dependência do projeto estava tudo pronto para o início do desenvolvimento do projeto. Foram então criados os primeiros pacotes e classes, sendo as de maior importância as classes **CertificadoAtributos** e **EmissaoCertificado**, a primeira possui os métodos e atributos que identificam um certificado de atributos e a segunda responsável diretamente pela emissão de um certificado de atributos. De posse de ambas as classes foi possível gerar testes para a emissão dos primeiros certificados de atributos, no caso certificados de testes.

Após os primeiros testes de emissão de certificados iniciou-se o desenvolvimento das classes que gerenciam os atributos e a sua ligação com as classes de geração dos certificados de atributos.

Em meio ao desenvolvimento do módulo de lógica de negócio iniciou-se o desenvolvimento do módulo de persistência. O módulo de persistência foi iniciado exatamente igual ao anterior, primeiramente foi criado um novo projeto no Eclipse, mas este projeto diferente do anterior, foi gerado um projeto de persistência de dados. A diferença de um projeto simples e um projeto de persistência de dados utilizando o Eclipse ficou por conta da geração de um arquivo XML chamado **Persistence.xml**. Este XML é responsável por todas as configurações de acesso ao banco de dados como usuário de acesso ao banco, tipo de sistema gerenciador de banco de dados (SGBD) utilizado, e mapeamentos entre as entidades de persistência. O Eclipse gera este arquivo xml ao solicitar a geração do projeto.

Mesmo com a geração automática do projeto pelo Eclipse foi necessário baixar as dependências JARS utilizadas para acessar o banco de dados, no caso do projeto o driver de conexão MySQL[25] e os JARS do framework de persistência Java, Hibernate[26]. Realizado o download dos jars os mesmos foram adicionados como bibliotecas de dependências do projeto.

Na programação Java de persistência são criadas entidades que representam as tabelas

¹O JAR ou Java Archive é um formato de empacotamento de classes e informações de metadados de forma a facilitar a distribuição de classes Java e executáveis

e os relacionamentos no banco de dados. Essas entidades são criadas utilizando o mecanismo de anotação Java, facilitando o processo de geração das entidades e os dados que elas representam no banco de dados. Cada entidade gerada deve ser incluída no arquivo **Persistence.xml** como classe utilizada pelo framework de persistência.

Com as classes de banco de dados geradas foi possível realizar a emissão em dois processos separados. O primeiro foi à geração de uma requisição e sua persistência em banco e o segundo processo de emissão do certificado através da requisição previamente gerada, conforme os diagramas de sequência abaixo.

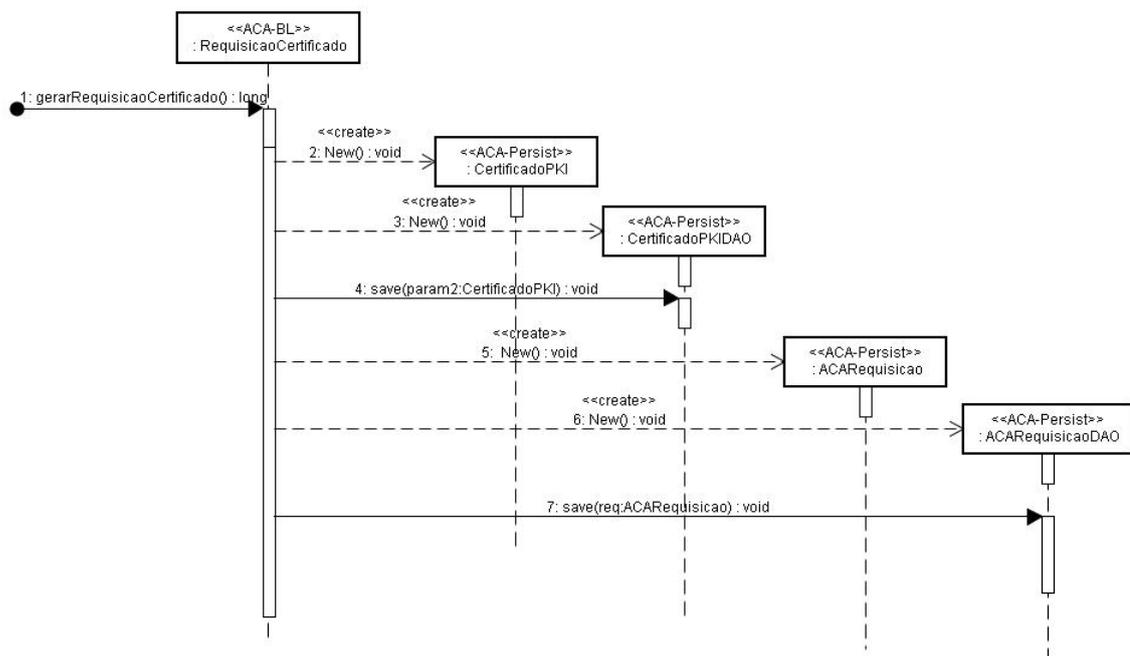


Figura 17: Diagrama de sequência ilustrando a requisição e persistência de um certificado de atributo

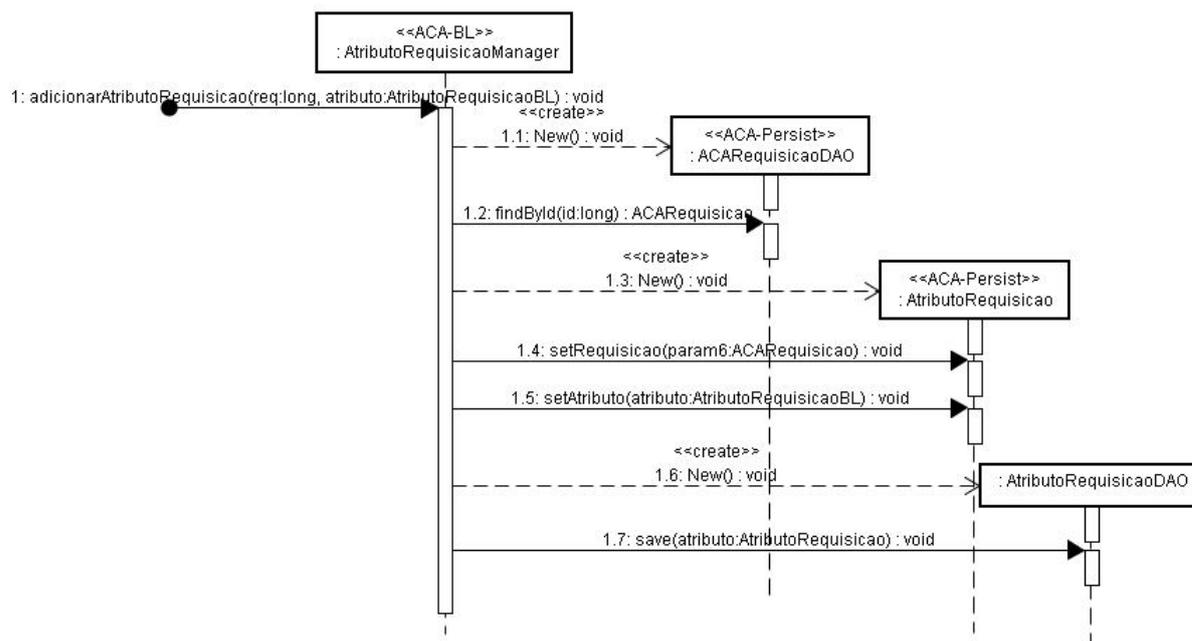


Figura 18: Diagrama de sequência ilustrando a adição de atributos na requisição

No fluxo de geração de requisição é necessária a obtenção do certificado de chave pública do requisitor, de posse do certificado de chaves públicas algumas informações sobre o requerente são colhidas e persistidas em banco. Após a coleta de informações do requerente uma nova requisição é gerada e persistida em banco, sendo gerado então um número de requisição enviado ao requisitor como resultado das operações.

O processo de geração utiliza um número de requisição para realizar o processo de emissão do certificado de atributo. Para emitir um certificado de atributos é necessário primeiramente adicionar quais atributos será emitido juntamente com o certificado e por consequência emitir o certificado com estes atributos, esta sequência é mais bem ilustrada nos diagramas abaixo.

Segundo o diagrama e os requisitos nada impede a adição de vários atributos desde que os atributos estejam previamente cadastrados na Autoridade de Atributos. Após a adição dos atributos na requisição é possível emitir o certificado de atributo.

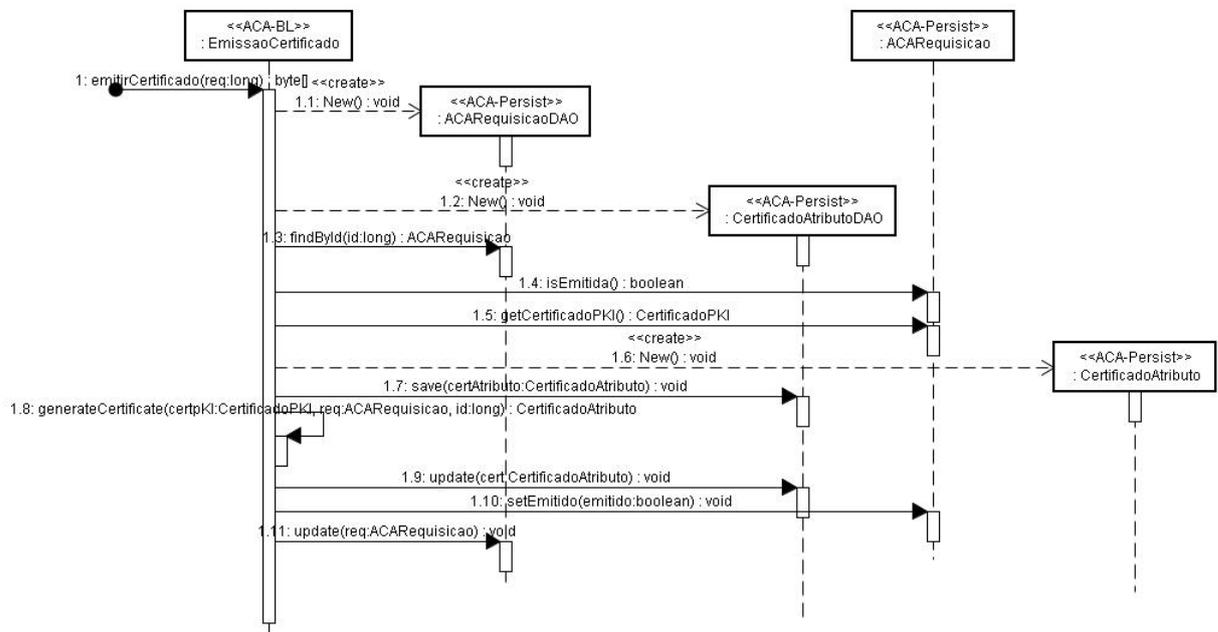


Figura 19: Diagrama de sequência ilustrando a emissão de um certificado de atributos

Para emitir um certificado o requerente necessita possuir um número de requisição válido, esta requisição é enviada a autoridade certificadora que inicia o procedimento de emissão do certificado. O procedimento possui etapas onde são resgatados do banco as informações do certificado de chaves públicas do requerente e os atributos adicionados a requisição do certificado, logo após é gerado o certificado de atributos.

Com os projetos de lógica de negócio e de persistência concluídos e testados foi criado o projeto do submódulo webservices como camada de acesso a lógica de negócio. O projeto webservice foi criado utilizando o Eclipse nos mesmo moldes dos anteriores através de telas providas pela ferramenta.

O projeto webservice segue o padrão JAX-WS[27] distribuído juntamente com a última versão do Java. Foi utilizada a documentação disponibilizada no site do framework para adquirir o conhecimento necessário para o desenvolvimento dos webservices.

O desenvolvimento de um webservice utilizando o framework JAX-WS basicamente consiste no uso de anotações java utilizadas em conjunto com o framework e de arquivos XML de configuração, são eles **web.xml** e **sun-jaxws.xml**. No arquivo **web.xml** reside as configurações do projeto web, neste arquivo foram configuradas algumas informações

para que o servidor web possa disponibilizar os webservices com sucesso, como a URL de acesso aos webservices. No arquivo **sun-jaxws.xml** foi configurado as classes Java onde os webservices estão desenvolvidos. É através deste arquivo de configuração que o motor do framework consegue disponibilizar os webservices.

No total de quatro serviços foram disponibilizados no final do desenvolvimento dos webservices, são eles:

- **Configuracao:** Disponibiliza acesso ao métodos de configuração da Autoridade de Atributos. São configurações gerais da Autoridade de Atributos, entre elas, configurações de certificado da Autoridade de Atributos, cadastro de atributos, emissão de LCR, etc.
- **Emissao:** Disponibiliza acesso aos métodos de emissão do certificado de atributos.
- **Requisicao:** Disponibiliza acesso aos métodos de requisição dos certificados de atributos.
- **Revogacao:** Disponibiliza métodos para um requerente revogar um certificado emitido.

A figura abaixo ilustra os pontos de acesso para cada um dos webservices publicados em um servidor web.

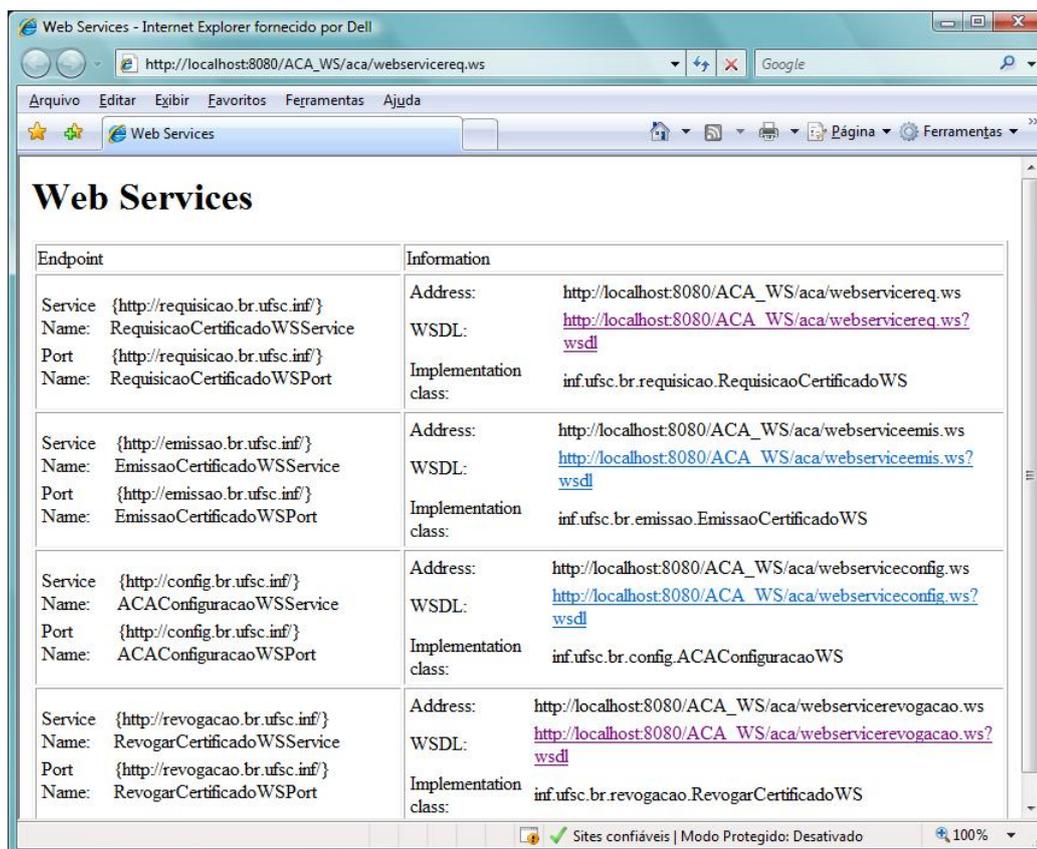


Figura 20: Ponto de acesso aos serviços disponibilizados pela Autoridade de Atributos

Com o módulo principal concluído é possível iniciar o desenvolvimento do módulo público e do módulo de administração da Autoridade de Atributos.

6.2 Módulo Público

O módulo público é caracterizado por uma aplicação web desenvolvida com a tecnologia .NET. O módulo público tem por objetivo constituir uma interface amigável de comunicação entre o módulo principal e seus serviços de emissão, requisição e revogação de certificados. Através do módulo público deve ser possível gerar requisições e emitir os certificados de atributos.

A infraestrutura do módulo público foi gerada acessando a interface gráfica do ambiente Visual Studio e acessando a opção de geração de projeto web. Finalizando as configurações do projeto o ambiente cria toda uma infraestrutura de desenvolvimento.

Para através do módulo público em .NET acessar os webservices disponibilizados foi utilizada a ferramenta **WSDL**, a ferramenta gera automaticamente o código para acesso a webservices bastando apenas informar como parâmetro a URL onde esta a WSDL provida

pelo webservice. Esta ferramenta é distribuída em conjunto com o próprio Visual Studio.

A figura abaixo ilustra o página inicial do módulo público.

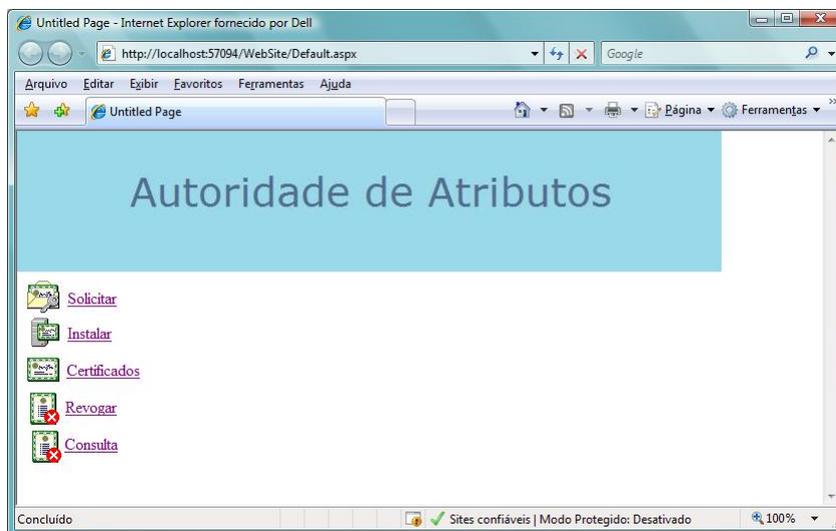


Figura 21: Página inicial do módulo público

O módulo público constitui-se de modo simplista de cinco menus ou links, sendo eles:

- Menu **Solicitar**: O menu Solicitar dispõe das etapas de preenchimento de dados para a obtenção de uma requisição de certificados de atributos. Dentre os dados a serem preenchidos encontra-se a seleção do certificado de chaves públicas, para isto foi desenvolvido um applet como ferramenta de auxílio, este applet acessa os certificados em repositórios da máquina do requerente bem como de cartões inteligentes. Nas figuras abaixo é possível visualizar estas etapas.

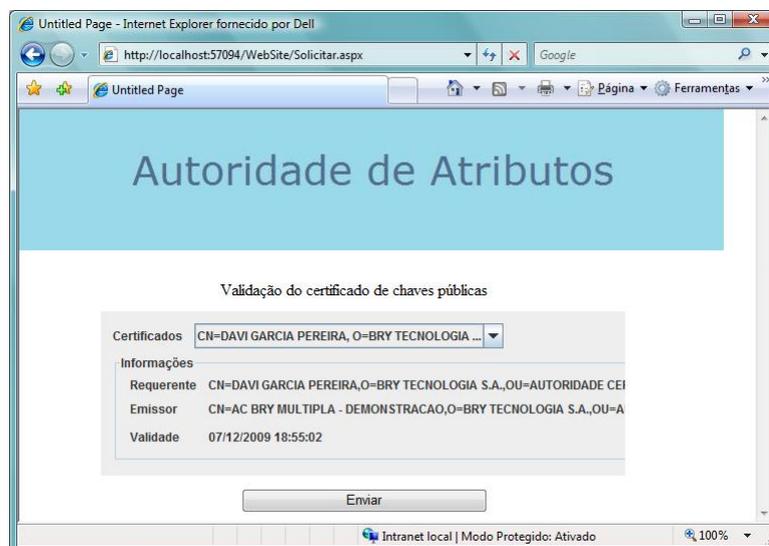


Figura 22: Applet para validação do certificado de chaves públicas

Após a validação um número de requisição é gerado e enviado ao cliente.

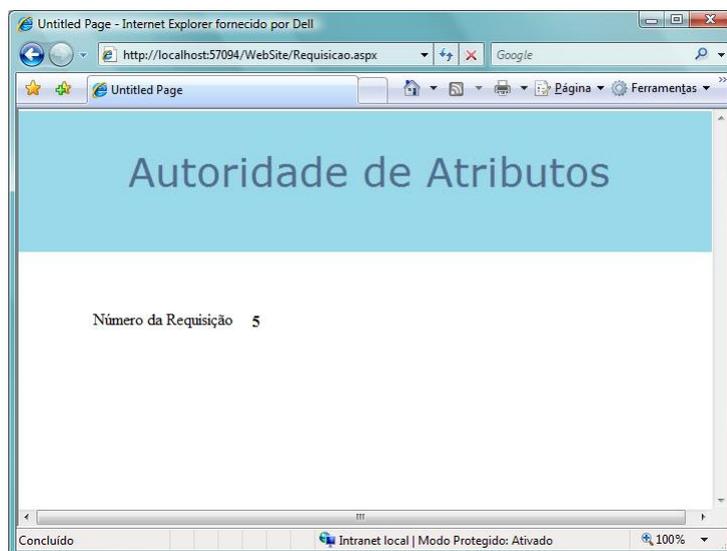


Figura 23: Número de requisição gerado após validação do certificado

- Menu **Instalar**: O menu Instalar constitui a etapa de emissão do certificado de atributo. As etapas para emissão são constituídas do envio e validação da requisição do certificado, a inserção dos atributos a serem emitidos e o download do certificado de atributos emitido. Nas figuras abaixo é possível visualizar estas etapas.

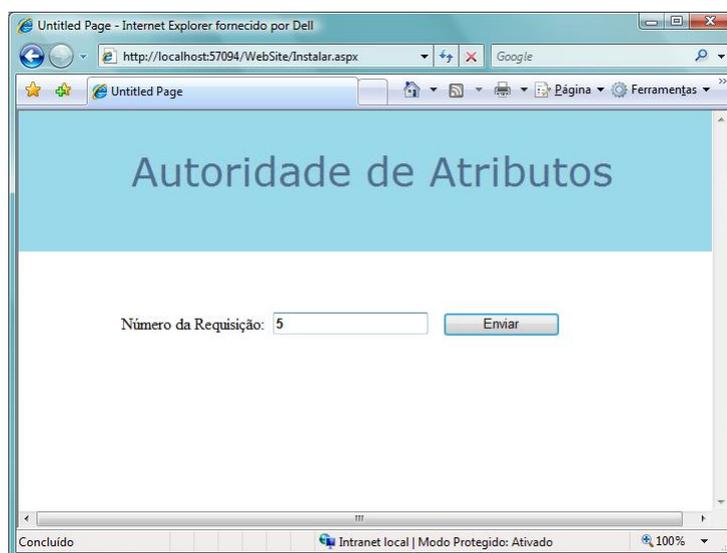


Figura 24: Inserção do número de requisição para emissão do certificado de atributos

Após a validação do número de requisição é necessário informar à Autoridade de Atributos quais atributos devem ser emitidos para o certificado.

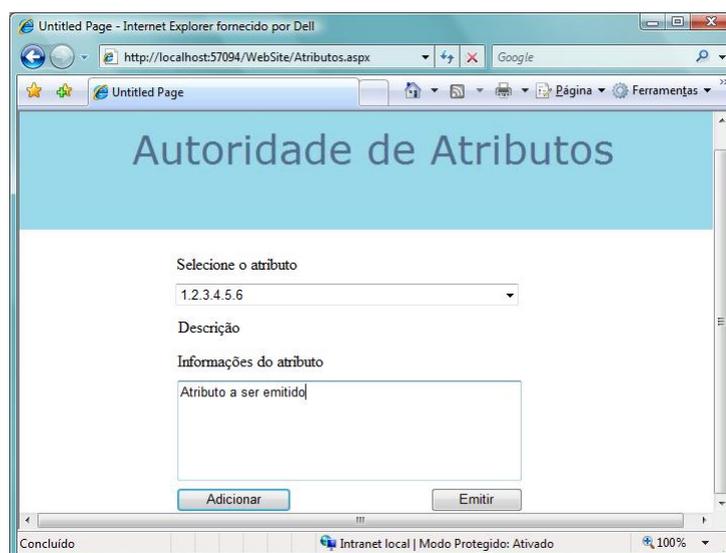


Figura 25: Inserção dos atributos a serem emitidos pela Autoridade de Atributos

Clicando em emitir o certificado de atributo é emitido e seu download pode ser iniciado.

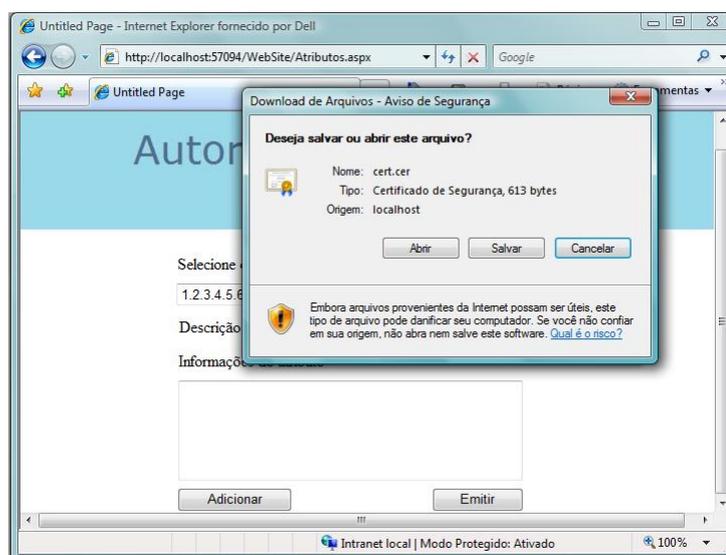


Figura 26: Download do certificado de atributos após emissão

- Menu **Certificados**: Utilizando o menu certificados é possível acessar todos os certificados de atributos emitidos pela Autoridade de Atributos. Na figura abaixo é possível visualizar esta funcionalidade.

Requerente	Data Emissao	Data Validade	Hash do Certificado	Certificado
CN=DAVI GARCIA PEREIRA,O=BRY TECNOLOGIA S.A.,OU=AUTORIDADE CERTIFICADORA DEMONSTRACAO,L=FLORIANOPOLIS,ST=SC,C=BR	06/05/2010 00:00:00	06/07/2010 00:00:00	54032876148bc77b9089772a1c25978b963710	Certificado
CN=DAVI GARCIA PEREIRA,O=BRY TECNOLOGIA S.A.,OU=AUTORIDADE CERTIFICADORA DEMONSTRACAO,L=FLORIANOPOLIS,ST=SC,C=BR	06/05/2010 00:00:00	06/07/2010 00:00:00	79b20b04a28741543491a2b13ee8bc485a2c8b	Certificado
CN=DAVI GARCIA PEREIRA,O=BRY TECNOLOGIA S.A.,OU=AUTORIDADE CERTIFICADORA DEMONSTRACAO,L=FLORIANOPOLIS,ST=SC,C=BR	06/05/2010 00:00:00	06/07/2010 00:00:00	0c8c376245e9e457403dec4d043d3a23338190	Certificado
CN=EVERTON FERNANDES,O=BRY TECNOLOGIA S.A.,OU=AUTORIDADE CERTIFICADORA DEMONSTRACAO,L=FLORIANOPOLIS,ST=SC,C=BR	06/05/2010 00:00:00	06/07/2010 00:00:00	5546a57ed91edba947a08be12b89488338817b2	Certificado
CN=DAVI GARCIA PEREIRA,O=BRY TECNOLOGIA S.A.,OU=AUTORIDADE CERTIFICADORA DEMONSTRACAO,L=FLORIANOPOLIS,ST=SC,C=BR	13/06/2010 15:09:49	13/07/2010 15:09:49	e7913b5d06d169483e263e3ab1c5cb0a0fcd5d	Certificado

Figura 27: Visualização dos certificados emitidos pela Autoridade de Atributos

- Menu **Revogar**: O menu Revogar dispõe as etapas para revogar um certificado de atributos. Para revogar um certificado de atributos deve-se informar a razão para executar tal ato. Nas figuras abaixo é possível visualizar estas etapas.

Número da Requisição:

Figura 28: Requisição para revogação do certificado de atributos

Após a validação da requisição uma nova tela é apresentada onde deve-se informar o motivo da revogação.

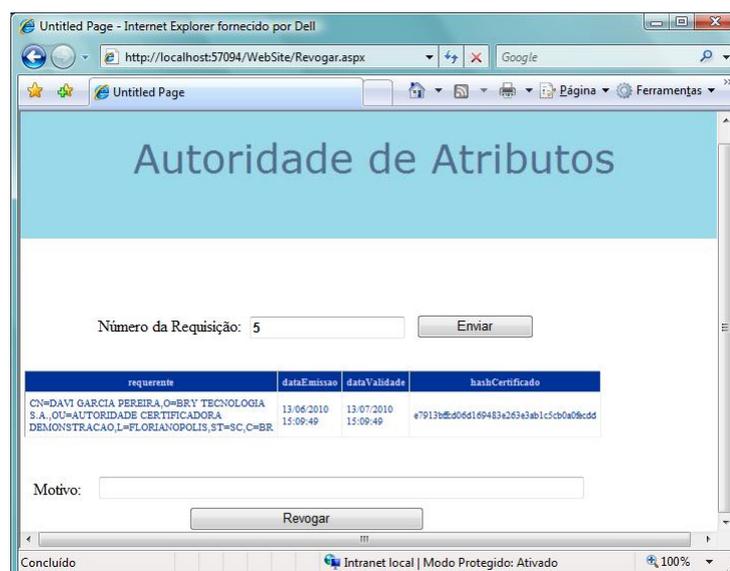


Figura 29: Motivo para revogação do certificado de atributos

Após a inserção do motivo o certificado é revogado clicando-se no botão "Revogar".

- Menu **Consultar**: Utilizando o menu Consultar e possível verificar se o certificado de atributos foi revogado e qual a razão para a sua revogação. Na figura abaixo é possível visualizar esta funcionalidade.

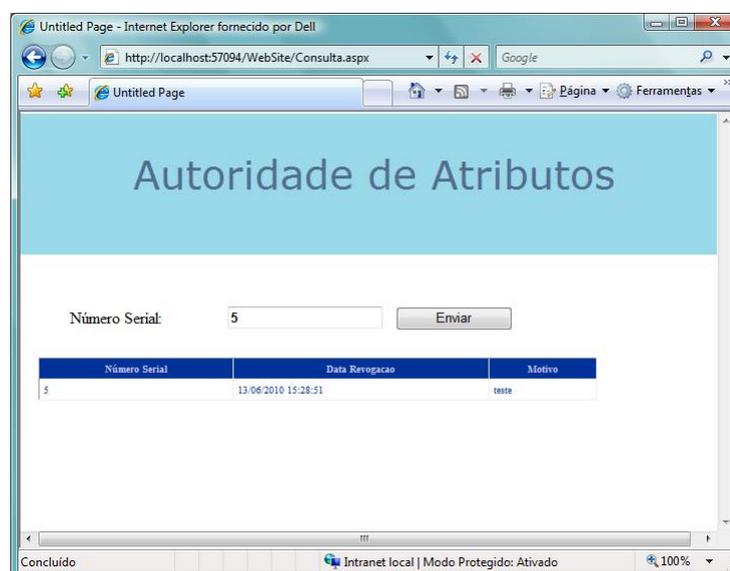


Figura 30: Consulta de certificados revogados

Finalizado o módulo público foi iniciado o desenvolvimento do módulo de administração.

6.3 Módulo de Administração

O módulo administrativo é caracterizado por uma aplicação desktop que deve ser utilizada na máquina onde está localizada a Autoridade de Atributos, isto devido a características de segurança. O módulo de administração disponibiliza uma interface de acesso as funções de administração provida pela Autoridade de Atributos e por seus serviços. O módulo de administração utiliza em sua maioria o webservice de Configuração.

A infraestrutura do módulo de administração foi gerada utilizando o ambiente de desenvolvimento Visual Studio e o framework .NET. A diferença entre o módulo público e o de administração fica por conta de sua geração, o módulo de administração não foi criado como módulo web mas sim configurado como aplicação desktop.

Para acessar o webservice de configuração foi utilizada a ferramenta distribuída com o Visual Studio, trata-se da ferramenta **WSDL** também utilizada na geração do módulo público. A ferramenta gerou as classes para acesso ao webservice de configuração bastando apenas passar como parâmetro a ferramenta o endereço da WSDL do webservice.

Com o módulo de administração é possível realizar as seguintes funções:

- Configurar o certificado utilizado pela Autoridade de Atributos para assinar os certificados de atributos por ela emitidos.
- Configurar os atributos a serem emitidos pela Autoridade de Atributos.
- Configurar a emissão das LCRs
- Realizar o processo de emissão e publicação das LCRs

A imagem abaixo ilustra a interface gráfica principal do módulo de administração.

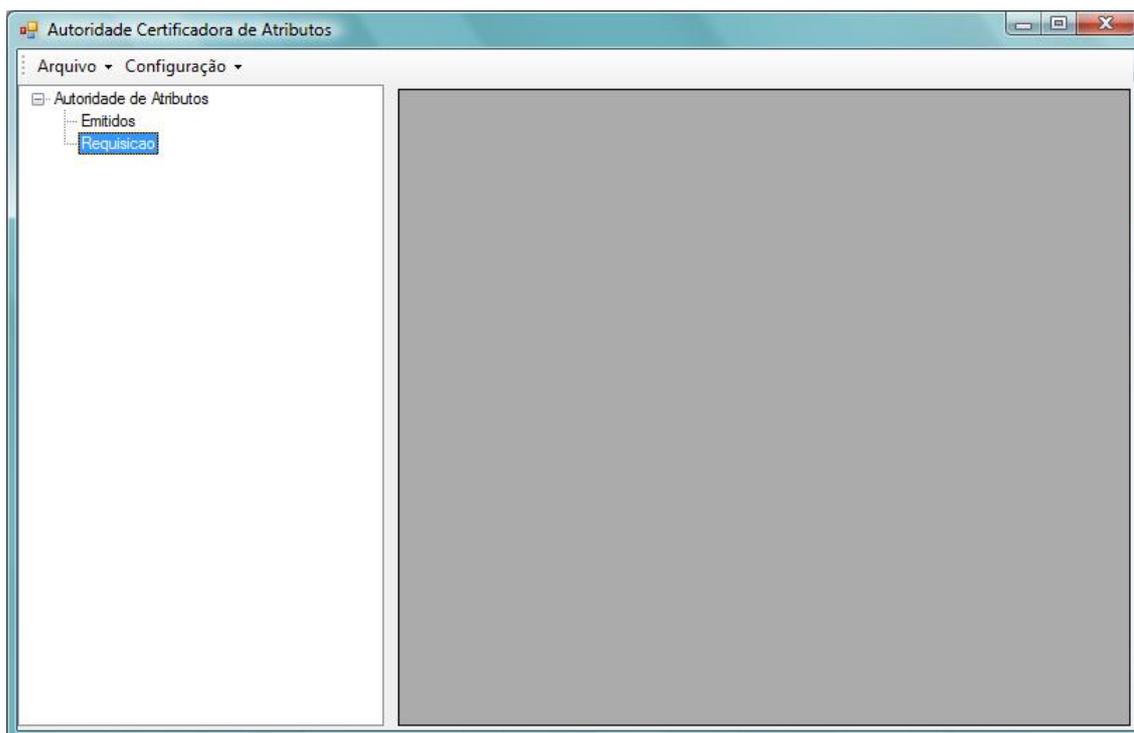


Figura 31: Interface gráfica do módulo de administração

O módulo de administração possui um menu chamando "Configuração", este menu possui as funcionalidades de configuração de atributos e o certificado utilizado pela Autoridade de Atributos para assinar os certificados. Esta funcionalidade pode ser visualizada na figura abaixo.

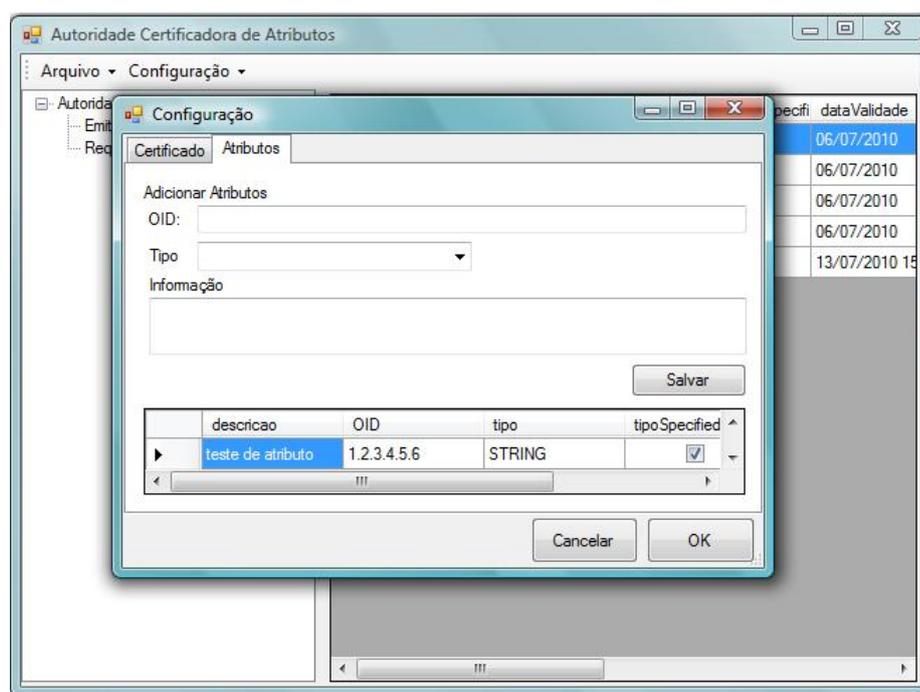


Figura 32: Interface gráfica de configuração do módulo de administração

6.4 Resultados Obtidos

Com a aplicação completa foi possível validar o ciclo completo de acordo com os requisitos e casos de uso levantados. Foi possível realizar o ciclo completo de uso da Autoridade de Atributos, desde sua configuração até a geração de requisição e emissão de certificados de atributos.

Abaixo é ilustrado um certificado de atributos emitido pela Autoridade de Atributos utilizando o software de visualização ASN.1 BerViewer.

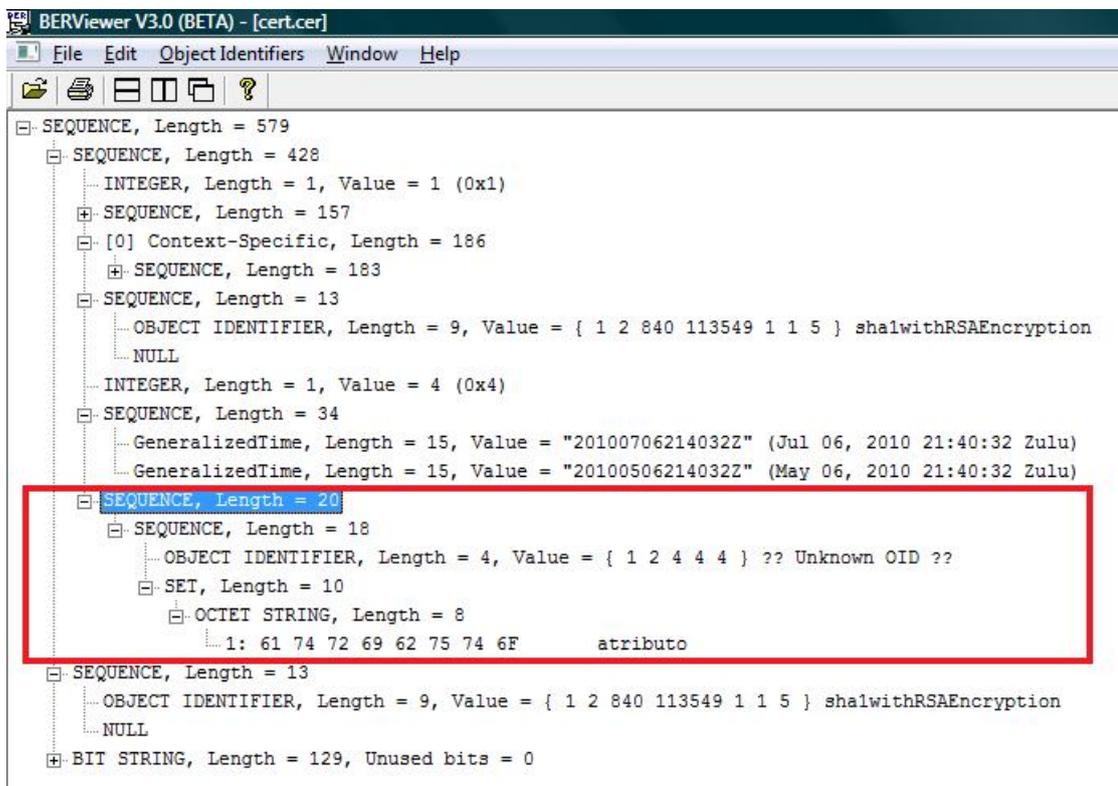


Figura 33: Certificado de atributo emitido utilizando a Autoridade de Atributos

Também foi possível realizar a revogação de certificados de atributos utilizando os mecanismos de revogação e publicação das LCRs disponibilizados pela Autoridade de Atributos.

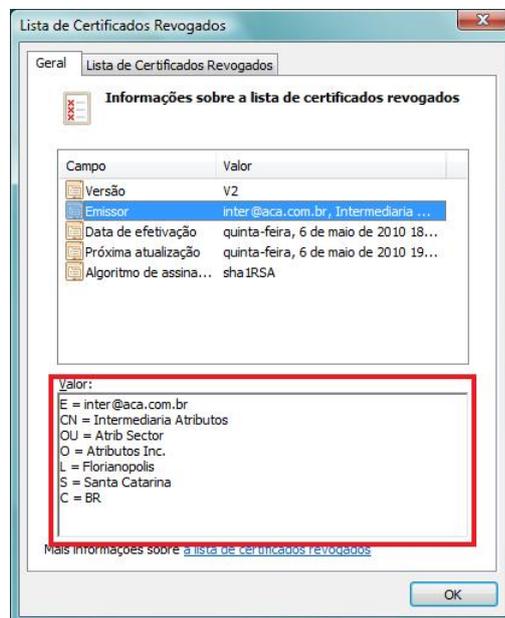


Figura 34: Lista de certificados revogados emitida utilizando a Autoridade de Atributos

6.5 Conclusão

O módulo principal constituiu-se o principal desafio da implementação, mesmo com as facilidades de desenvolvimento com a utilização de ambientes de desenvolvimento como o Eclipse, os frameworks de persistência de dados e de desenvolvimento de webservices geraram grande tempo e esforço para seu conhecimento e desenvolvimento.

Evitando quaisquer comparativos entre linguagens e plataformas de desenvolvimento o .NET e seu ambiente de desenvolvimento Visual Studio constituíram ferramentas mais amigáveis de trabalhar e exigiram menor esforço de desenvolvimento.

Importantes no momento de implementação, todos os diagramas de casos de uso e requisitos bem como os digramas de sequência foram de grande importância para realizar o desenvolvimento do módulo principal. Utilizando estas ferramentas de modelagem foi possível sempre retirar quaisquer dúvidas durante o desenvolvimento.

7 *Considerações Finais*

O protótipo de Autoridade de Atributos pode ser uma obra que contribui para o entendimento do certificado de atributos e suas características, contribuindo de certa forma para um futuro onde esta tecnologia de certificação possa vir a ser utilizada, não somente em frameworks de autorização, mas sim para algo mais que isso, guardar características legais de seu portador, o certificado de atributos possui potencial enorme devido as suas características de portabilidade de atributos.

Em análise sob o ponto de vista de complexidade do projeto, pode-se analisá-lo de forma desafiadora para obtenção de conhecimentos nas diversas áreas desde os aspectos conceituais da autoridade de atributos e certificados de atributos até as tecnologias que permeiam o desenvolvimento do protótipo. A utilização de uma metodologia nos moldes ágeis foi certamente uma das melhores escolhas e de grande valia, considerando prazos curtos, desconhecimento de tecnologias e equipe reduzida, no caso do trabalho uma única pessoa.

Sobre a metodologia e ferramentas utilizadas no desenvolvimento é possível afirmar que a metodologia de análise e projeto focadas no levantamento de requisitos e casos de uso foi de extrema importância durante todo o desenvolvimento do protótipo, os requisitos servem como uma trilha que deve ser seguida com responsabilidade, mesmo que durante o processo de implementação alguma parte do modelo conceitual precise ser ajustado devido a fatores intrínsecos ao desenvolvimento.

As ferramentas de apoio atenderam com eficiência a demanda, principalmente as ferramentas de apoio a codificação no caso deste projeto Eclipse e Visual Studio realmente merecem todos os méritos, seus atalhos e dicas de utilização realmente facilitam e muito a vida do programador.

As dificuldades do desenvolvimento ficaram por conta do aprendizado de novas tecnologias de desenvolvimento. Por ser em projeto um protótipo que utilizou várias tecnologias, várias atividades e horas de estudo e pesquisa foram necessárias para aprender

cada uma das tecnologias empregadas. Em várias situações durante o desenvolvimento a documentação de apoio e exemplos disponibilizados era insuficiente para sanar todas as dúvidas e a alternativa foram pesquisas em sites técnicos e fóruns de discussão sobre a tecnologia.

Quanto ao protótipo por se tratar de um trabalho acadêmico pode ser considerado um software livre podendo ser utilizado para futura pesquisa e apoio a desenvolvimento de forma livre sem restrições.

7.1 Trabalhos Futuros

Este projeto pode ser entendido como artefato introdutório aos certificados de atributos e todo seu escopo e infraestrutura. Várias assuntos relativos a esta tecnologia não foram levantados e precisam ser estudados. Sendo assim, podemos identificar este projeto como o primeiro de outros que podem vir a serem criados. No que tange o planejado para o escopo deste projeto como trabalho acadêmico, os objetivos definidos foram atingidos com pleno êxito, ficando então demonstrado com clareza tecnologias em termos de programação ou codificação de maneira plena para futuros desenvolvimentos.

Dentre os trabalhos futuros identificam-se os seguintes:

- **Framework de Autorização:** Assunto não discutido neste trabalho, mas se trata do primeiro e maior uso dos certificados de atributos. Autoridades de Atributos são utilizadas em conjunto com softwares de autorização para caracterizar quais os limites de ação de usuários. Existem vários frameworks de autorização desenvolvidos e seu estudo são importantes para entender ainda mais os certificados de atributos.
- **Certificados de Atributos e ICP-Brasil:** Estudo sobre a utilização de certificados de atributos em conjunto com a infraestrutura de chaves públicas existente no Brasil. Se o certificado de atributo possui uso e espaço junto a infraestrutura atualmente existente e quais seriam os seus benefícios.
- **Estudo de Autoridade de Atributos existentes no mercado e qual o seu escopo de aplicação.** O objetivo é verificar o uso do certificado de atributos para uma utilização além dos frameworks de autorização.

Referências

- [1] Instituto Nacional de Tecnologia da Informação - ITI. *Certificação Digital*. 2008. URL: <http://www.iti.gov.br/twiki/bin/view/Certificacao/WebHome>.
- [2] IEC 9594-8 - ITU-T Recommendation X.509. *Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks*. [S.l.], 2000.
- [3] William Stallings. *Cryptography and Network Security: Principles and Practice*. [S.l.]: Prentice Hall, 2003.
- [4] Bruce Schneier. *Applied cryptography*. [S.l.]: John Wiley & Sons, Inc, 1996.
- [5] Carlisle Adams; Steve Lloyd. *PKI: Concepts, Standards, and Deployment Considerations*. [S.l.]: Addison Wesley, 2002.
- [6] The Internet Engineering Task Force - IETF. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. 2002. URL: <http://www.ietf.org/rfc/rfc3280.txt>.
- [7] Instituto Nacional de Tecnologia da Informação - ITI. *Estrutura da ICP-Brasil*. 2010. URL: <http://www.iti.gov.br/twiki/bin/view/Certificacao/PerguntaVinteSeis>.
- [8] Russ Housley; Tim Polk. *Planning for PKI, Best Practices Guide for Deploying Public Key Infrastructure*. [S.l.]: John Wiley & Sons, Inc, 2001.
- [9] Jing-Jang Hwang; Kou-Chen Wu; Duen-Ren Liu. *Access control with role attribute certificates*. [S.l.], 1999.
- [10] Wei Zhou; Christoph Meinel. *implement Role-Based Access Control with Attribute Certificates*. [S.l.], 2003.
- [11] GUTMANN, P. *PKI: It's Not Dead, Just Resting*. [S.l.].
- [12] The Internet Engineering Task Force - IETF. *An Internet Attribute Certificate Profile for Authorization*. 2002. URL: <http://www.ietf.org/rfc/rfc3281.txt>.
- [13] SUN Microsystems. *Detalhes Tecnologia JavaEE*. 2010. URL: <http://java.sun.com/javaee/>.
- [14] Doederlin. *Entendendo Web Da RPC a Service Oriented Architecture*. [S.l.], 2006.
- [15] Tiago Estevas De Rolt. *Integracao de Servicos Utilizando WebServices*. [S.l.: s.n.], 2003.

- [16] Mauro Santana. *Informacoes sobre Web Services e seu desenvolvimento na plataforma Microsoft*. 2010. URL: <http://www.microsoft.com/brasil/msdn/webservices/Default.aspx/>.
- [17] UDDI.org. *Especificacao Tecnica UDDI*. 2010. URL: <http://www.uddi.org/about.htm/>.
- [18] W3C - World Wide Web Consortium. *Tutorial W3C sobre Web Services*. 2010. URL: <http://www.w3schools.com/webservices/default.asp>.
- [19] SUN Microsystems. *Java SE Security*. 2010. URL: <http://java.sun.com/javase/technologies/security/>.
- [20] SUN Microsystems. *Java Cryptography Architecture (JCA)*. 2010. URL: <http://java.sun.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html>.
- [21] The Legion of the Bouncy Castle. *Bouncycastle.org*. 2010. URL: <http://www.bouncycastle.org/index.html>.
- [22] Gilleanes Guedes. *UML - Uma Abordagem Pratica*. [S.l.]: Novatec, 2004.
- [23] The Legion of the Bouncy Castle. *Bouncycastle.org Documentation*. 2010. URL: <http://www.bouncycastle.org/documentation.html>.
- [24] The Legion of the Bouncy Castle. *Bouncycastle.org last releases*). 2010. URL: http://www.bouncycastle.org/latest_releases.html.
- [25] MySQL Database. *MySQL connectors*. 2010. URL: <http://dev.mysql.com/downloads/connector/>.
- [26] JBoss Community. *Hibernate Persistence for Java & NET*. 2010. URL: <http://www.hibernate.org/>.
- [27] SUN Microsystems. *Metro Web Services Technologies*. 2010. URL: <http://java.sun.com/webservices/technologies/index.jsp>.

APÊNDICE A – Códigos Fontes

A.1 Módulo Principal

A.1.1 Módulo Logica de Negócio

Listagem A.1: Classe AtributoCadastro

```
package inf.ufsc.br.ac;

import java.io.Serializable;

5 import inf.ufsc.br.atributos.TipoAtributo;

public class AtributoCadastro implements Serializable{

    /**
10     *
    */
    private static final long serialVersionUID = -5714679680639565061L;
    private String OID;
    private TipoAtributo tipo;
15 private byte[] valorPadrao;
    private String descricao;

    public AtributoCadastro() {
        // TODO Auto-generated constructor stub
20    }

    public String getOID() {
        return OID;
    }
25 public void setOID(String oID) {
        OID = oID;
    }
}
```

```
    public TipoAtributo getTipo() {
        return tipo;
30    }
    public void setTipo(TipoAtributo tipo) {
        this.tipo = tipo;
    }
    public byte[] getValorPadrao() {
35        return valorPadrao;
    }
    public void setValorPadrao(byte[] valorPadrao) {
        this.valorPadrao = valorPadrao;
    }
40    public String getDescricao() {
        return descricao;
    }
    public void setDescricao(String descricao) {
        this.descricao = descricao;
45    }
}
}
```

Listagem A.2: Classe AtributoRequisicaoBL

```
package inf.ufsc.br.ac;

import java.io.Serializable;

5 public class AtributoRequisicaoBL implements Serializable{

    /**
     *
     */
10    private static final long serialVersionUID = -2067829003054138221L;
    private String oid;
    private byte[] valor;
    private long idAtributo;

15    public AtributoRequisicaoBL() {
        // TODO Auto-generated constructor stub
    }

    public long getIdAtributo() {
20        return idAtributo;
    }
}
```

```
    }

    public void setIdAtributo(long idAtributo) {
        this.idAtributo = idAtributo;
25    }

    public AtributoRequisicaoBL(String oid, byte[] valor, long
        idAtributo) {
        this.oid = oid;
        this.valor = valor;
30        this.idAtributo = idAtributo;
    }

    public String getOid() {
        return oid;
35    }

    public void setOid(String oid) {
        this.oid = oid;
    }

40    public byte[] getValor() {
        return valor;
    }

    public void setValor(byte[] valor) {
45        this.valor = valor;
    }

}
```

Listagem A.3: Classe GeradorAtributo

```
package inf.ufsc.br.ac;

import java.io.Serializable;

5 import org.bouncycastle.asn1.ASN1Encodable;
import org.bouncycastle.asn1.BERConstructedOctetString;
import org.bouncycastle.asn1.DERPrintableString;
import org.bouncycastle.x509.X509Attribute;

10 public class GeradorAtributo implements Serializable{
```

```
    /**
     *
     */
15  private static final long serialVersionUID = 1L;
    private X509Attribute atributo;

    public GeradorAtributo(String oid, byte[] valor) {
        this.generateAttribute(oid, valor);
20  }

    public GeradorAtributo(String oid, String valor) {
        this.generateAttribute(oid, valor);
    }
25

    private void generateAttribute(String oid, byte[] valor)
    {
        ASN1Encodable asn1 = new BERConstructedOctetString(valor);
        this.atributo = new X509Attribute(oid, asn1);
30  }

    private void generateAttribute(String oid, String valor)
    {
        ASN1Encodable asn1 = new DERPrintableString(valor);
35  this.atributo = new X509Attribute(oid, asn1);
    }

    public X509Attribute getAtributo()
    {
40  return this.atributo;
    }

}
```

Listagem A.4: Classe CadastroAtributo

```
package inf.ufsc.br.cadastro;

import java.util.Collection;

5  import inf.ufsc.br.ac.AtributoCadastro;
import inf.ufsc.br.atributos.Atributo;
import inf.ufsc.br.atributos.AtributoDAO;
```

```
public class CadastroAtributo {
10
    public CadastroAtributo () {
        // TODO Auto-generated constructor stub
    }

15    public void adicionarAtributo(AtributoCadastro cadastro)
    {
        AtributoDAO dao = new AtributoDAO ();
        Atributo atributo = new Atributo ();
        atributo.setDescricao(cadastro.getDescricao());
20        atributo.setOID(cadastro.getOID());
        atributo.setTipo(cadastro.getTipo());
        if (cadastro.getValorPadrao() != null)
            atributo.setValorPadrao(cadastro.getValorPadrao());
        dao.save(atributo );
25    }

    public Collection<Atributo> getAtributos ()
    {
        AtributoDAO dao = new AtributoDAO ();
30        return dao.getAllAtributos ();
    }
}
}
```

Listagem A.5: Classe CadastroAutoridade

```
package inf.ufsc.br.cadastro;

import inf.ufsc.br.certificado.CertificadoACA;
import inf.ufsc.br.certificado.CertificadoACADAO;
5
public class CadastroAutoridade {

    public CadastroAutoridade () {
        // TODO Auto-generated constructor stub
10    }

    public void adicionarCertificadoAutoridade(byte[] pkcs12, String
        senha)
    {
        CertificadoACA aca = new CertificadoACA ();
15        aca.setPkcs12(pkcs12);
```



```
        aca.setSenha(senha);

        CertificadoACADAODao dao = new CertificadoACADAODao();

20         dao.save(aca);
    }
}
```

Listagem A.6: Classe Certificado

```
package inf.ufsc.br.certificado;

import java.io.ByteArrayInputStream;
import java.io.IOException;
5 import java.security.KeyStoreException;
import java.security.cert.Certificate;
import java.security.cert.CertificateException;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;
10

public class Certificado {

15     private Certificate certificado;

    public Certificado(Certificate cert) throws KeyStoreException {
        this.certificado = cert;
    }

20     public Certificado(byte[] valor) throws CertificateException,
        IOException {
        ByteArrayInputStream iStream = new ByteArrayInputStream(
            valor);
        CertificateFactory cf = CertificateFactory.getInstance("X
            .509");
        certificado = cf.generateCertificate(iStream);
25         iStream.close();
    }

    public X509Certificate toX509()
    {
30         return (X509Certificate) this.certificado;
    }
}
```

```
        @Override
        public String toString()
        {
35             return this.toX509().getSubjectDN().getName();
        }
    }
}
```

Listagem A.7: Classe CertificadoAC

```
package inf.ufsc.br.certificado;

import inf.ufsc.br.utils.UtilitariaArquivo;

5 import java.io.ByteArrayInputStream;
import java.io.IOException;
import java.security.KeyStore;
import java.security.KeyStoreException;
import java.security.NoSuchAlgorithmException;
10 import java.security.PrivateKey;
import java.security.UnrecoverableKeyException;
import java.security.cert.CertificateException;
import java.util.Enumeration;

15

public class CertificadoAC {

    private KeyStore keyStore;
20     private String senha;

    public CertificadoAC() {
        try {
            this.keyStore = KeyStore.getInstance("PKCS12");
25
        } catch (KeyStoreException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
30
    }

    public void adicionarPKCS12(String arquivo, String senha) throws
```

```
        NoSuchAlgorithmException, CertificateException, IOException
    {
35         ByteArrayInputStream bStream = new ByteArrayInputStream (
            UtilitariaArquivo.lerArquivo(arquivo));
        this.senha = senha;
        this.keyStore.load(bStream, senha.toCharArray());
    }

40     public void adicionarPKCS12(byte[] pkcs12, String senha) throws
        NoSuchAlgorithmException, CertificateException, IOException
    {
        ByteArrayInputStream bStream = new ByteArrayInputStream (
            pkcs12);
        this.senha = senha;
        this.keyStore.load(bStream, senha.toCharArray());
45     }

    public String getAlias() throws KeyStoreException
    {
        String alias = null;
50         for (Enumeration e = this.keyStore.aliases(); e.
            hasMoreElements();)
        {
            alias = e.nextElement().toString();
        }
        return alias;
55     }

    public Certificado getCertificado() throws KeyStoreException
    {
        return new Certificado(this.keyStore.getCertificate(
            getAlias()));
60     }

    public PrivateKey getChavePrivada() throws
        UnrecoverableKeyException, KeyStoreException,
        NoSuchAlgorithmException
    {
        PrivateKey chavePrivada = (PrivateKey) this.keyStore.getKey
            (getAlias(), this.senha.toCharArray());
65
        return chavePrivada;
    }
}
```

```
    public String getProvider ()
70    {
        return this.keyStore.getProvider ().getName ();
    }
}
```

Listagem A.8: Classe CertificadoAtributos

```
package inf.ufsc.br.certificado;

import java.io.IOException;
import java.security.Principal;
5 import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import org.bouncycastle.x509.AttributeCertificateHolder;
10 import org.bouncycastle.x509.AttributeCertificateIssuer;
import org.bouncycastle.x509.X509Attribute;
import org.bouncycastle.x509.X509AttributeCertificate;
import org.bouncycastle.x509.X509V2AttributeCertificate;

15 public class CertificadoAtributos {

    private X509AttributeCertificate certificate;
    private String holder;
    private String holderCN;
20    private String holderC;
    private String holderO;
    private List<String> listaHolderOU;
    private String holderL;
    private String holderST;
25    private String holderE;
    private String issuer;
    private List<String> listaIssuerOU;
    private String issuerCN;
    private String issuerC;
30    private String issuerO;
    private String issuerL;
    private String issuerST;
    private String issuerE;
```

```
35     public CertificadoAtributos(X509AttributeCertificate certificate) {
        this.certificate = certificate;
        this.listaHolderOU = new ArrayList<String>();
        this.listaIssuerOU = new ArrayList<String>();
40         //this.montarHolder();
        //this.monterIssuer();
    }

    public CertificadoAtributos (byte[] certificado) throws IOException
45     {
        this.certificate = new X509V2AttributeCertificate(
            certificado);
        this.listaHolderOU = new ArrayList<String>();
        this.listaIssuerOU = new ArrayList<String>();
        this.montarHolder();
50         this.monterIssuer();
    }

    public byte[] getEncoded() throws IOException
    {
55         return this.certificate.getEncoded();
    }

    public Date getDataInicio()
    {
60         return this.certificate.getNotAfter();
    }

    public Date getDataFim()
    {
65         return this.certificate.getNotBefore();
    }

    public X509Attribute[] getAtributos()
    {
70         return this.certificate.getAttributes();
    }

    private void montarHolder()
    {
75         AttributeCertificateHolder holder = this.certificate.
            getHolder();
```

```
Principal[] sujeito = holder.getEntityNames();
this.holder = sujeito[0].getName();
String[] splitVirgula = this.holder.split(",");
for (int i = 0; i < splitVirgula.length; i++)
80 {
    String[] splitIgual = splitVirgula[i].split("=");
    if (splitIgual[0].equalsIgnoreCase("CN"))
    {
        holderCN = splitIgual[1];
85 System.out.println(holderCN);
    }

    if (splitIgual[0].equalsIgnoreCase("C"))
    {
90 holderC = splitIgual[1];
        System.out.println(holderC);
    }

    if (splitIgual[0].equalsIgnoreCase("O"))
95 {
        holderO = splitIgual[1];
        System.out.println(holderO);
    }

    if (splitIgual[0].equalsIgnoreCase("OU"))
100 {
        this.listaHolderOU.add(splitIgual[1]);
        System.out.println(this.listaHolderOU);
    }

    if (splitIgual[0].equalsIgnoreCase("L"))
105 {
        holderL = splitIgual[1];
        System.out.println(holderL);
    }

    if (splitIgual[0].equalsIgnoreCase("ST"))
110 {
        holderST = splitIgual[1];
        System.out.println(holderST);
    }

    if (splitIgual[0].equalsIgnoreCase("E"))
```

```

        {
120             holderE = splitIguar[1];
                System.out.println(holderE);
        }
    }
}

125 private void monterIssuer()
{
    AttributeCertificateIssuer issuer = this.certificate.
        getIssuer();
    Principal[] sujeito = issuer.getPrincipals();
130     this.issuer = sujeito[0].getName();
    String[] splitVirgula = this.issuer.split(",");
    for (int i = 0; i < splitVirgula.length; i++)
    {
        String[] splitIguar = splitVirgula[i].split("=");
135         if (splitIguar[0].equalsIgnoreCase("CN"))
            {
                issuerCN = splitIguar[1];
                System.out.println(issuerCN);
            }
140
        if (splitIguar[0].equalsIgnoreCase("C"))
            {
                issuerC = splitIguar[1];
                System.out.println(issuerC);
145            }

        if (splitIguar[0].equalsIgnoreCase("O"))
            {
                issuerO = splitIguar[1];
150                System.out.println(issuerO);
            }

        if (splitIguar[0].equalsIgnoreCase("OU"))
            {
155                this.listaIssuerOU.add(splitIguar[1]);
                System.out.println(this.listaIssuerOU);
            }

        if (splitIguar[0].equalsIgnoreCase("L"))
160            {

```

```
        issuerL = splitIgual[1];
        System.out.println(issuerL);
    }

165     if (splitIgual[0].equalsIgnoreCase("ST"))
        {
            issuerST = splitIgual[1];
            System.out.println(issuerST);
        }

170     if (splitIgual[0].equalsIgnoreCase("E"))
        {
            issuerE = splitIgual[1];
            System.out.println(issuerE);
175     }
    }

    public String getHolder() {
180         return holder;
    }

    public String getHolderCN() {
        return holderCN;
185     }

    public String getHolderC() {
        return holderC;
    }

190     public String getHolderO() {
        return holderO;
    }

    public String getHolderL() {
195         return holderL;
    }

    public String getHolderST() {
200         return holderST;
    }

    public String getHolderE() {
```



```
        return holderE;
205     }

    public String getIssuer () {
        return issuer;
    }

210     public String getIssuerCN () {
        return issuerCN;
    }

215     public String getIssuerC () {
        return issuerC;
    }

    public String getIssuerO () {
220         return issuerO;
    }

    public String getIssuerL () {
225         return issuerL;
    }

    public String getIssuerST () {
        return issuerST;
    }

230     public String getIssuerE () {
        return issuerE;
    }

235 }
```

Listagem A.9: Classe CertificadoRevogado

```
package inf.ufsc.br.certificado;

import java.math.BigInteger;
import java.util.Date;
5

public class CertificadoRevogado {

    private BigInteger numeroSerie;
    private Date dataRevogacao;
```

```
10     private int Razao;

    public CertificadoRevogado(BigInteger numSerie, Date dataRevogacao,
        int razao) {
        this.numeroSerie = numSerie;
        this.dataRevogacao = dataRevogacao;
15     this.Razao = razao;
    }

    public BigInteger getNumeroSerie() {
        return numeroSerie;
20    }

    public Date getDataRevogacao() {
        return dataRevogacao;
    }

25    public int getRazao() {
        return Razao;
    }

30 }
```

Listagem A.10: Classe AtributoRequisicaoManagerBL

```
package inf.ufsc.br.emissao;

import inf.ufsc.br.ac.AtributoRequisicaoBL;
import inf.ufsc.br.requisicao.ACARequisicao;
5 import inf.ufsc.br.requisicao.ACARequisicaoDAO;
import inf.ufsc.br.requisicao.AtributoRequisicaoDAO;
import inf.ufsc.br.requisicao.AtributoRequisicao;

public class AtributoRequisicaoManagerBL {
10
    public AtributoRequisicaoManagerBL() {
        // TODO Auto-generated constructor stub
    }

15    public void adicionarAtributoRequisicao(long req,
        AtributoRequisicaoBL atributo)
    {
        ACARequisicaoDAO reqDao = new ACARequisicaoDAO();
        ACARequisicao requisicao = reqDao.findById(req);
```

```
20         AtributoRequisicao atributoReq = new AtributoRequisicao ();
           atributoReq.setAcReq(requisicao);
           atributoReq.setOid(atributo.getOid());
           atributoReq.setValor(atributo.getValor());

25         AtributoRequisicaoDAO atribReq = new AtributoRequisicaoDAO
           ();

           atribReq.save(atributoReq);

           }

30     }
}
```

Listagem A.11: Classe EmissaoCertificado

```
package inf.ufsc.br.emissao;

import inf.ufsc.br.ac.GeradorAtributo;
import inf.ufsc.br.certificado.Certificado;
5 import inf.ufsc.br.certificado.CertificadoAC;
import inf.ufsc.br.certificado.CertificadoACA;
import inf.ufsc.br.certificado.CertificadoACADAODAO;
import inf.ufsc.br.certificado.CertificadoAtributo;
import inf.ufsc.br.certificado.CertificadoAtributoDAO;
10 import inf.ufsc.br.certificado.CertificadoAtributos;
import inf.ufsc.br.certificado.CertificadoPKI;
import inf.ufsc.br.manager.ACAEntityManager;
import inf.ufsc.br.requisicao.ACARequisicao;
import inf.ufsc.br.requisicao.ACARequisicaoDAO;
15 import inf.ufsc.br.requisicao.AtributoRequisicao;
import inf.ufsc.br.requisicao.AtributoRequisicaoDAO;
import inf.ufsc.br.utils.Hash;

import java.io.IOException;
20 import java.math.BigInteger;
import java.security.InvalidKeyException;
import java.security.KeyStoreException;
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
25 import java.security.PrivateKey;
import java.security.SignatureException;
import java.security.UnrecoverableKeyException;
```

```
import java.security.cert.CertificateEncodingException;
import java.security.cert.CertificateException;
30 import java.security.cert.CertificateParsingException;
import java.security.cert.X509Certificate;
import java.util.Collection;
import java.util.Date;

35 import org.bouncycastle.x509.AttributeCertificateHolder;
import org.bouncycastle.x509.AttributeCertificateIssuer;
import org.bouncycastle.x509.X509AttributeCertificate;
import org.bouncycastle.x509.X509V2AttributeCertificateGenerator;

40 public class EmissaoCertificado {

    protected X509V2AttributeCertificateGenerator x509Generator;
    private CertificadoAC certACA;

45    public EmissaoCertificado() throws NoSuchAlgorithmException,
        CertificateException, IOException {
        x509Generator = new X509V2AttributeCertificateGenerator();
        CertificadoACADAO acaDao = new CertificadoACADAO();
        CertificadoACA certbd = acaDao.findById(1);

50        certACA = new CertificadoAC();
        certACA.adicionarPKCS12(certbd.getPkcs12(), certbd.getSenha
            ());
    }

55    public byte[] emitirCertificado(long requisicao) throws Exception
    {
        ACAREquisicaoDAO reqDao = new ACAREquisicaoDAO();
        CertificadoAtributoDAO atrDao = new CertificadoAtributoDAO
            ();

60        ACAREquisicao acaReq = reqDao.findById(requisicao);

        if (acaReq.isEmitido())
        {
            return acaReq.getCertAtributo().getValorCertificado
                ();
65        }
    }
}
```

```
CertificadoPKI certpki = acaReq.getCertificadoPKI();

Certificado x509Cert = new Certificado(certpki.getValor());
70
ACAEntityManager.beginTransaction();

CertificadoAtributos certatr;

75
try {
    CertificadoAtributo certAtr = new
        CertificadoAtributo();
    atrDao.save(certAtr);
    certatr = generateCertificate(x509Cert.toX509(),
        acaReq, certAtr.getId());
80
    certAtr.setCertificadoPKI(certpki);
    certAtr.setDataEmissao(certatr.getDataInicio());
    certAtr.setDataValidade(certatr.getDataFim());
    certAtr.setNumeroSerial(certAtr.getId());
    certAtr
85
        .setHashCertificado(Hash
            .gerarHashSHA1(
                certatr.
                    getEncoded()));
    certAtr.setRequerente(certpki.getRequerente());
    certAtr.setValorCertificado(certatr.getEncoded());
    atrDao.update(certAtr);
90
    acaReq.setEmitido(true);
    acaReq.setCertAtributo(certAtr);
    reqDao.update(acaReq);
    ACAEntityManager.commit();
} catch (Exception e) {
95
    ACAEntityManager.rollback();
    throw e;
}
return certatr.getEncoded();
}

100
public Collection<CertificadoAtributo> getTodosCertificadosEmitidos
    ()
    {
        CertificadoAtributoDAO atributoDao = new
            CertificadoAtributoDAO();
```

```
        return atributoDao.getAllCertificadoAtributos();
105     }

    public CertificadoAtributo getCertificadoEmitidoById(long id)
    {
        CertificadoAtributoDAO atributoDao = new
            CertificadoAtributoDAO();
110     return atributoDao.findById(id);
    }

    public CertificadoAtributo getCertificadoAtributoPorRequisicao(long
        reqId)
    {
115     ACARequisicaoDAO reqDao = new ACARequisicaoDAO();
        ACARequisicao req = reqDao.findById(reqId);
        return req.getCertAtributo();
    }

120     private CertificadoAtributos generateCertificate(X509Certificate
        userCertificate, ACARequisicao acaReq, long serial) throws
        CertificateParsingException, IOException,
        UnrecoverableKeyException, KeyStoreException,
        NoSuchAlgorithmException, CertificateEncodingException,
        InvalidKeyException, IllegalStateException,
        NoSuchProviderException, SignatureException
    {
        this.x509Generator.setHolder(generateHolder(userCertificate
            ));
        this.x509Generator.setIssuer(generateIssuer(this.certACA.
            getCertificado().toX509()));
        Date dataAtual = new Date();
125     Date dataNotBefore = new Date();
        //dataNotBefore.setMinutes(dataAtual.getMinutes()+this.aca.
            getAcaconfig().getValidadeCertificado());
        dataNotBefore.setMonth(6);
        this.x509Generator.setNotAfter(dataAtual);
        this.x509Generator.setNotBefore(dataNotBefore);
130     this.x509Generator.setSignatureAlgorithm("SHA1WithRSA");
        this.x509Generator.setSerialNumber(new BigInteger(Long.
            toHexString(serial)));

        AtributoRequisicaoDAO reqAtribDao = new
```

```
135     AtributoRequisicaoDAO ();
    Collection<AtributoRequisicao> atribs = reqAtribDao.
        findAtributosRequisicaoByRequisicao (acaReq.getId ());
    for (AtributoRequisicao atributo : atribs) {
        GeradorAtributo gerAtributo = new GeradorAtributo(
            atributo.getOid (), atributo.getValor ());
        this.x509Generator.addAttribute (gerAtributo.
            getAtributo ());
    }
140
    PrivateKey chavePrivada = this.certACA.getChavePrivada ();
    X509AttributeCertificate certificadoAtributo = this.
        x509Generator.generate (chavePrivada, this.certACA.
            getProvider ());
145    CertificadoAtributos certAtributo = new
        CertificadoAtributos (certificadoAtributo);

    return certAtributo;
}

150 private AttributeCertificateHolder generateHolder (X509Certificate
    userCertificate) throws CertificateParsingException
{
    AttributeCertificateHolder holder = new
        AttributeCertificateHolder (userCertificate);
    return holder;
}

155 private AttributeCertificateIssuer generateIssuer (X509Certificate
    userCertificate) throws IOException
{
    AttributeCertificateIssuer issuer = new
        AttributeCertificateIssuer (userCertificate.
            getSubjectX500Principal ());
    return issuer;
160 }
}
}
```

Listagem A.12: Classe LCRGenerator

```
package inf.ufsc.br.lcr;
```

```
import inf.ufsc.br.certificado.Certificado;
import inf.ufsc.br.certificado.CertificadoAC;
5 import inf.ufsc.br.certificado.CertificadoRevogado;
import inf.ufsc.br.model.config.LCRConfig;

import java.security.InvalidKeyException;
import java.security.KeyStoreException;
10 import java.security.NoSuchAlgorithmException;
import java.security.SignatureException;
import java.security.UnrecoverableKeyException;
import java.security.cert.CRLException;
import java.security.cert.X509CRL;
15 import java.util.Date;

import org.bouncycastle.x509.X509V2CRLGenerator;

public class LCRGenerator {
20
    private X509V2CRLGenerator crlGenerator;
    private LCRConfig config;

    public LCRGenerator(LCRConfig config) {
25         this.config = config;
        this.crlGenerator = new X509V2CRLGenerator();
        this.crlGenerator.setSignatureAlgorithm(config.getAlgAssinatura());
    }

30     public void addCertificadoRevogado(CertificadoRevogado revogado)
    {
        this.crlGenerator.addCRLEntry(revogado.getNumeroSerie(),
        revogado.getDataRevogacao(),
        revogado.getRazao());
35     }

    public X509CRL generate(CertificadoAC certificadoACA) throws
        KeyStoreException, InvalidKeyException,
        UnrecoverableKeyException, CRLException, IllegalStateException,
        NoSuchAlgorithmException, SignatureException
    {
40         Date dataAtual = new Date();
        this.crlGenerator.setThisUpdate(dataAtual);
```



```

        Date dataPublicacao = new Date();
        dataPublicacao.setMinutes(dataAtual.getMinutes() + this.
            config.getTempoPublicacao());
        this.crlGenerator.setNextUpdate(dataPublicacao);
        Certificado certCA = certificadoACA.getCertificado();
45    this.crlGenerator.setIssuerDN(certCA.toX509().
            getIssuerX500Principal());
        return this.crlGenerator.generate(certificadoACA.
            getChavePrivada());
    }

50

}

```

Listagem A.13: Classe LCRManager

```

package inf.ufsc.br.lcr;

import inf.ufsc.br.certificado.CertificadoAC;
import inf.ufsc.br.certificado.CertificadoACA;
5  import inf.ufsc.br.certificado.CertificadoACADA0;
import inf.ufsc.br.certificado.CertificadoAtributoRevogado;
import inf.ufsc.br.certificado.CertificadoAtributoRevogadoDAO;
import inf.ufsc.br.certificado.CertificadoRevogado;
import inf.ufsc.br.manager.ACAEntityManager;
10 import inf.ufsc.br.model.config.LCRConfig;
import inf.ufsc.br.model.config.LCRConfigDAO;
import inf.ufsc.br.utils.Hash;

import java.io.FileOutputStream;
15 import java.math.BigInteger;
import java.security.cert.X509CRL;
import java.util.Date;
import java.util.List;

20 public class LCRManager {

    public LCRManager() {
        // TODO Auto-generated constructor stub
    }

25

    public void alterarConfiguracaoLCR(LCRConfig config)

```

```
{
    ACAEntityManager.beginTransaction();
    LCRConfigDAO lcrDAO = new LCRConfigDAO();
30    lcrDAO.save(config);
    ACAEntityManager.commit();
}

public void emitirLCR() throws Exception
35 {
    try {
        ACAEntityManager.beginTransaction();

        LCRConfigDAO lcrCfgDAO = new LCRConfigDAO();
40    LCRConfig lcrconfig = lcrCfgDAO.findById(1);
        LCRGenerator lcrGen = new LCRGenerator(lcrconfig);
        CertificadoAtributoRevogadoDAO atribDAO = new
            CertificadoAtributoRevogadoDAO();
        List<CertificadoAtributoRevogado> listaAtribs =
            atribDAO
                .getAllCertificadoAtributoRevogados
                    ();
45    for (CertificadoAtributoRevogado
        certificadoAtributoRevogado : listaAtribs) {
            CertificadoRevogado revogado = new
                CertificadoRevogado(
                    new BigInteger(Long.toHexString(
                        certificadoAtributoRevogado.
                            getNumeroSerial())),
                    certificadoAtributoRevogado.
                        getDatarevogacao(), 1);
                    lcrGen.addCertificadoRevogado(revogado);
50    }
        CertificadoACADAO acaDAO = new CertificadoACADAO();
        CertificadoACA aca = acaDAO.findById(1);
        CertificadoAC certAC = new CertificadoAC();

60    certAC.adicionarPKCS12(aca.getPkcs12(), aca.
        getSenha());
        X509CRL crl = lcrGen.generate(certAC);

        Lcr lcr = new Lcr();

        lcr.setDataEmissao(new Date());
```

```

        lcr.setHash(Hash.gerarHashSHA1(crl.getEncoded()));
        lcr.setValor(crl.getEncoded());

        LcrDAO lcrDao = new LcrDAO();
65         lcrDao.save(lcr);

        ACAEntityManager.commit();
    } catch (Exception e) {

70         ACAEntityManager.rollback();
        throw e;
    }
}
75 }

```

Listagem A.14: Classe RepositorioWindows

```

package inf.ufsc.br.repositorio;

import inf.ufsc.br.certificado.Certificado;
import inf.ufsc.br.certificado.CertificadoAC;
5
import java.io.IOException;
import java.security.KeyStore;
import java.security.KeyStoreException;
import java.security.NoSuchAlgorithmException;
10 import java.security.cert.CertificateException;
import java.util.ArrayList;
import java.util.Enumeration;
import java.util.List;

15 public class RepositorioWindows {

    private KeyStore windows;

    public RepositorioWindows() throws KeyStoreException,
        NoSuchAlgorithmException, CertificateException, IOException {
20         windows = KeyStore.getInstance("Windows-MY");
        windows.load(null);
    }

    public int getCountCertificados() throws KeyStoreException
25 {

```

```
        return this.windows.size();
    }

    public List<Certificado> getCertificados() throws KeyStoreException
30 {
        List<Certificado> listaCertificados = new ArrayList<
            Certificado>();
        Enumeration<String> aliases = this.windows.aliases();
        while(aliases.hasMoreElements())
        {
35            String alias = aliases.nextElement();
            Certificado cert = new Certificado(this.windows.
                getCertificate(alias));
            listaCertificados.add(cert);
        }
        return listaCertificados;
40    }
}
}
```

Listagem A.15: Classe RequisicaoBL

```
package inf.ufsc.br.requisicao;

import inf.ufsc.br.ac.AtributoRequisicaoBL;
import inf.ufsc.br.certificado.CertificadoAtributo;
5 import inf.ufsc.br.certificado.CertificadoPKI;

import java.util.Date;
import java.util.HashSet;
import java.util.Set;
10

public class RequisicaoBL {

    private long id;
    private CertificadoPKI certificadoPKI;
15 private Date dataRequisicao;
    private boolean emitido;
    private CertificadoAtributo certAtr;

    public RequisicaoBL() {
20        // TODO Auto-generated constructor stub
    }
}
```

```
    public long getId () {  
25         return id;  
    }  
  
    public void setId(long id) {  
30         this.id = id;  
    }  
  
    public CertificadoPKI getCertificadoPKI () {  
35         return certificadoPKI;  
    }  
  
    public void setCertificadoPKI(CertificadoPKI certificadoPKI) {  
40         this.certificadoPKI = certificadoPKI;  
    }  
  
    public Date getDataRequisicao () {  
45         return dataRequisicao;  
    }  
  
    public void setDataRequisicao(Date dataRequisicao) {  
50         this.dataRequisicao = dataRequisicao;  
    }  
  
    public boolean isEmitido () {  
55         return emitido;  
    }  
  
    public void setEmitido(boolean emitido) {  
60         this.emitido = emitido;  
    }  
  
    public CertificadoAtributo getCertAtr () {  
65         return certAtr;  
    }  
  
    public void setCertAtr(CertificadoAtributo certAtr) {  
70         this.certAtr = certAtr;  
    }  
  
}
```

Listagem A.16: Classe RequisicaoCertificado

```
package inf.ufsc.br.requisicao;

import java.io.IOException;
import java.security.NoSuchAlgorithmException;
5 import java.security.cert.CertificateEncodingException;
import java.util.Collection;
import java.util.Date;
import java.util.HashSet;
import java.util.Set;

10
import inf.ufsc.br.certificado.Certificado;
import inf.ufsc.br.certificado.CertificadoPKI;
import inf.ufsc.br.certificado.CertificadoPKIDAO;
import inf.ufsc.br.utils.Hash;

15

public class RequisicaoCertificado {

    private Certificado certificado;

20

    public RequisicaoCertificado(Certificado cert) {
        this.certificado = cert;
    }

25

    public RequisicaoCertificado() {
        // TODO Auto-generated constructor stub
    }

    public long gerarRequisicaoCertificado() throws
        CertificateEncodingException, NoSuchAlgorithmException,
        IOException

30
    {
        CertificadoPKI pki = new CertificadoPKI();
        pki.setEmissao(this.certificado.toX509().getNotAfter());
        pki.setEmissor(this.certificado.toX509().
            getIssuerX500Principal().getName());
        pki.setFinalValidade(this.certificado.toX509().getNotBefore
            ());

35
        pki.setHash(Hash.gerarHashSHA1(this.certificado.toX509().
            getEncoded()));
    }
}
```

```
        pki.setRequerente(this.certificado.toX509().
            getSubjectX500Principal().getName());
        pki.setValor(this.certificado.toX509().getEncoded());

        CertificadoPKIDAO pkiDao = new CertificadoPKIDAO();
40    pkiDao.save(pki);

        ACARequisicao requisicao = new ACARequisicao();
        requisicao.setCertificadoPKI(pki);
        requisicao.setDataRequisicao(new Date());
45

        ACARequisicaoDAO reqDao = new ACARequisicaoDAO();

        reqDao.save(requisicao);

50    return requisicao.getId();
    }

    public boolean isEmitida(Long reqId)
    {
55        ACARequisicaoDAO reqDao = new ACARequisicaoDAO();
        ACARequisicao req = reqDao.findById(reqId);

        return req.isEmitido();
    }

60

    public RequisicaoBL getRequisicao(Long reqId)
    {
        ACARequisicaoDAO reqDao = new ACARequisicaoDAO();
        ACARequisicao req = reqDao.findById(reqId);
65

        RequisicaoBL reqbl = new RequisicaoBL();
        reqbl.setCertAtr(req.getCertAtributo());
        reqbl.setCertificadoPKI(req.getCertificadoPKI());
        reqbl.setDataRequisicao(req.getDataRequisicao());
70        reqbl.setEmitido(req.isEmitido());
        reqbl.setId(req.getId());

        return reqbl;
    }

75

    public void setRequisicaoEmitida(ACARequisicao req)
    {
```

```

        ACARequisicaoDAO reqDao = new ACARequisicaoDAO();
        req.setEmitido(true);
80      reqDao.update(req);
    }

    public Set<RequisicaoBL> getTodasRequisicao()
    {
85      ACARequisicaoDAO dao = new ACARequisicaoDAO();
        Collection<ACARequisicao> req = dao.getAllRequisicao();

        HashSet<RequisicaoBL> listaReqBL = new HashSet<RequisicaoBL
            >();
90      for (ACARequisicao acaRequisicao : req) {
            RequisicaoBL reqbl = new RequisicaoBL();
            reqbl.setCertAtr(acaRequisicao.getCertAtributo());
            reqbl.setCertificadoPKI(acaRequisicao.
                getCertificadoPKI());
            reqbl.setDataRequisicao(acaRequisicao.
                getDataRequisicao());
            reqbl.setEmitido(acaRequisicao.isEmitido());
95      reqbl.setId(acaRequisicao.getId());

            listaReqBL.add(reqbl);
        }
100     return listaReqBL;
    }
}

```

Listagem A.17: Classe RevogacaoCertificado

```

package inf.ufsc.br.revogacao;

public class RevogacaoCertificado {
5     private String motivo;
    private long requisicao;

    public RevogacaoCertificado() {
        // TODO Auto-generated constructor stub
10    }

    public String getMotivo() {

```



```
        return motivo;
    }
15
    public void setMotivo(String motivo) {
        this.motivo = motivo;
    }

20    public long getRequisicao() {
        return requisicao;
    }

    public void setRequisicao(long requisicao) {
25        this.requisicao = requisicao;
    }

30 }
```

Listagem A.18: Classe RevogacaoCertificadoManager

```
package inf.ufsc.br.revogacao;

import java.util.Date;

5 import inf.ufsc.br.certificado.CertificadoAtributo;
import inf.ufsc.br.certificado.CertificadoAtributoDAO;
import inf.ufsc.br.certificado.CertificadoAtributoRevogado;
import inf.ufsc.br.certificado.CertificadoAtributoRevogadoDAO;
import inf.ufsc.br.manager.ACAEntityManager;
10 import inf.ufsc.br.requisicao.ACARequisicao;
import inf.ufsc.br.requisicao.ACARequisicaoDAO;

public class RevogacaoCertificadoManager {

15    public RevogacaoCertificadoManager() {
        // TODO Auto-generated constructor stub
    }

    public void revogarCertificadoAtributo(RevogacaoCertificado
        revogacao) throws Exception
20    {
        ACARequisicaoDAO reqDao = new ACARequisicaoDAO();
        ACARequisicao req = reqDao.findById(revogacao.getRequisicao
```

```
        ());

        CertificadoAtributo certAtributo = req.getCertAtributo();

25
        try {
            ACAEntityManager.beginTransaction();
            CertificadoAtributoDAO certAtributoDao = new
                CertificadoAtributoDAO();
            certAtributoDao.revogarCertificadoById(certAtributo
                .getId());
30
            CertificadoAtributoRevogado certRevogado = new
                CertificadoAtributoRevogado();
            certRevogado.setDataRevogacao(new Date());
            certRevogado.setDescricaoMotivo(revogacao.getMotivo
                ());
            certRevogado.setNumeroSerial(certAtributo.
                getNumeroSerial());
            certRevogado.setValor(certAtributo.
                getValorCertificado());
35
            CertificadoAtributoRevogadoDAO rev = new
                CertificadoAtributoRevogadoDAO();
            rev.save(certRevogado);
            ACAEntityManager.commit();
        } catch (Exception e) {
            ACAEntityManager.rollback();
40
            throw e;
        }
    }

    public CertificadoAtributoRevogado
        getCertificadoRevogadoPorNumeroSerial(long numeroSerial)
45
    {
        CertificadoAtributoRevogadoDAO certRevDao = new
            CertificadoAtributoRevogadoDAO();
        return certRevDao.findByNumeroSerie(numeroSerial);
    }

50 }
```

Listagem A.19: Classe Hash

```
package inf.ufsc.br.utils;

import java.io.ByteArrayInputStream;
```

```
import java.io.ByteArrayOutputStream;
5 import java.io.IOException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

import org.bouncycastle.util.encoders.HexEncoder;
10
public class Hash {

    public static String gerarHashSHA1(byte[] valor) throws
        NoSuchAlgorithmException, IOException
    {
15
        MessageDigest dsg = MessageDigest.getInstance("SHA
            -1");
        byte[] b = dsg.digest(valor);

        ByteArrayOutputStream out = new
            ByteArrayOutputStream();
20
        HexEncoder enc = new HexEncoder();
        enc.encode(b, 0, b.length, out);

        return out.toString();
25
    }
}
```

Listagem A.20: Classe UtilitariaArquivo

```
package inf.ufsc.br.utils;

import java.io.File;
import java.io.FileInputStream;
5 import java.io.FileOutputStream;
import java.io.IOException;

public class UtilitariaArquivo {

10
    public static byte[] lerArquivo(String caminho) throws IOException
    {
        byte[] conteudo = null;

        try
```

```
15         {
                File arquivo = new java.io.File(caminho);
                if (arquivo.exists())
                {
                    conteudo = new byte[(int) arquivo.length()
20                    ];
                    FileInputStream fStream = new
                        FileInputStream(arquivo);
                    fStream.read(conteudo, 0, (int) arquivo.
                        length());
                    fStream.close();
                }
            } catch (IOException ex)
25         {
                throw ex;
            }
            return conteudo;
        }
30
    public static void escreverArquivo(byte[] conteudo, String caminho)
        throws IOException
    {
        File arquivo = new File(caminho);
        FileOutputStream fStream = new FileOutputStream(arquivo);
35        fStream.write(conteudo, 0, conteudo.length);
        fStream.close();
    }
40 }
```

Listagem A.21: Classe UtilitariaCodificacao

```
package inf.ufsc.br.utils;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
5
import org.bouncycastle.util.encoders.Base64Encoder;
import org.bouncycastle.util.encoders.HexEncoder;

public class UtilitariaCodificacao {
10
    public static String codificarBase64(byte[] dado) throws
```

```
        IOException
    {
        Base64Encoder encoder64 = new Base64Encoder();
        ByteArrayOutputStream stream = new ByteArrayOutputStream();
15         encoder64.encode(dado, 0, dado.length, stream);
        return stream.toString();
    }

    public static String codificarHexadecimal(byte[] dado) throws
        IOException
20     {
        HexEncoder encoderHex = new HexEncoder();
        ByteArrayOutputStream stream = new ByteArrayOutputStream();
        encoderHex.encode(dado, 0, dado.length, stream);
        return stream.toString();
25     }

    public static byte[] DecodificarBase64(byte[] dado) throws
        IOException
    {
        Base64Encoder encoder64 = new Base64Encoder();
30         ByteArrayOutputStream stream = new ByteArrayOutputStream();
        encoder64.decode(dado, 0, dado.length, stream);
        return stream.toByteArray();
    }

35     public static byte[] DecodificarHexadecimal(byte[] dado) throws
        IOException
    {
        HexEncoder encoderHex = new HexEncoder();
        ByteArrayOutputStream stream = new ByteArrayOutputStream();
        encoderHex.encode(dado, 0, dado.length, stream);
40         return stream.toByteArray();
    }
}
}
```

Listagem A.22: build

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="MyProject" default="dist" basedir=".">
    <description>
        Build for ACA_BL
5    </description>
```

```

<!-- set global properties for this build -->
<property name="src" location="src"/>
<property name="build" location="build"/>
<property name="dist" location="dist"/>
10 <property name="build.lib" location="build/lib"/>
<property name="lib" location="lib"/>

    <path id="master-classpath">
        <fileset dir="${lib}"/>
15         <include name="*.jar"/>
        </fileset>
        <pathelement path="${build}"/>
    </path>

20 <target name="init">
    <!-- Create the time stamp -->
    <tstamp/>
    <!-- Create the build directory structure used by compile -->
    <mkdir dir="${build}"/>
25     <mkdir dir="${build.lib}"/>
    <!--<copydir dest="${build.lib}" src="${lib}"></copydir-->
</target>

<target name="compile" depends="init" description="compile the source " >
30 <!-- Compile the java code from ${src} into ${build} -->
    <javac srcdir="${src}" destdir="${build}" debug="true">
        <classpath refid="master-classpath"/>
    </javac>
</target>

35 <target name="dist" depends="compile" description="generate the
    distribution" >
    <!-- Create the distribution directory -->
    <mkdir dir="${dist}/lib"/>

40 <!-- Put everything in ${build} into the MyProject-${DSTAMP}.jar file
    -->
    <jar jarfile="${dist}/lib/ACA_BL-${DSTAMP}.jar" basedir="${build}"/>
</target>

<target name="clean" description="clean up" >
45 <!-- Delete the ${build} and ${dist} directory trees -->
    <delete dir="${build}"/>

```

```
<delete dir="${dist}"/>
</target>
</project>
```

A.1.2 Módulo Persistência de Dados

Listagem A.23: Classe Atributo

```
package inf.ufsc.br.tributos;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
5 import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "Atributos")
10 public class Atributo {

    @Id
    @GeneratedValue
    private long id;
15 private String OID;
    private TipoAtributo tipo;
    private byte[] valorPadrao;
    private String descricao;

20

    public String getDescricao() {
        return descricao;
    }
    public void setDescricao(String descricao) {
25         this.descricao = descricao;
    }
    public byte[] getValorPadrao() {
        return valorPadrao;
    }
30 public void setValorPadrao(byte[] valorPadrao) {
        this.valorPadrao = valorPadrao;
    }
    public String getOID() {
        return OID;
    }
}
```

```
35     }
    public void setOID(String oID) {
        OID = oID;
    }
    public TipoAtributo getTipo() {
40         return tipo;
    }
    public void setTipo(TipoAtributo tipo) {
        this.tipo = tipo;
    }
45     public long getId() {
        return id;
    }
}
}
```

Listagem A.24: Classe AtributoDAO

```
package inf.ufsc.br.atributos;

import java.util.List;

5 import inf.ufsc.br.manager.ACAEntityManager;

import javax.persistence.EntityManager;
import javax.persistence.Query;

10 public class AtributoDAO {

    public AtributoDAO() {
        // TODO Auto-generated constructor stub
    }

15     public EntityManager getEntityManager()
    {
        return ACAEntityManager.getEntityManager();
    }

20     public void save(Atributo atributo)
    {
        try
        {
25             ACAEntityManager.beginTransaction();
            getEntityManager().persist(atributo);
        }
    }
}
```



```
        ACAEntityManager.commit();
    } catch (RuntimeException e)
    {
30         throw e;
    }
}

public Atributo findById(long id)
35 {
    try {
        Atributo atributo = ACAEntityManager.
            getEntityManager().find(Atributo.class, id);

        return atributo;
40    } catch (RuntimeException e) {
        throw e;
    }
}

public void delete(Atributo atributo)
45 {
    try{
        atributo = ACAEntityManager.getEntityManager().
            getReference(Atributo.class, atributo.getId());
        ACAEntityManager.getEntityManager().remove(atributo
50    );
    } catch (RuntimeException e) {
        throw e;
    }
}

public List<Atributo> getAllAtributos()
55 {
    Query q = ACAEntityManager.getEntityManager().createQuery("
        select a from Atributo a");

    List<Atributo> atributos = q.getResultList();
60    return atributos;
}
}
```

Listagem A.25: Classe TipoAtributo

```
package inf.ufsc.br.atributos;
```

```
public enum TipoAtributo {  
    NUMERO,  
5    STRING,  
    BYTEARRAY  
}
```

Listagem A.26: Classe CertificadoACA

```
package inf.ufsc.br.certificado;  
  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
5 import javax.persistence.Id;  
import javax.persistence.Lob;  
  
@Entity  
public class CertificadoACA {  
10  
    @Id  
    @GeneratedValue  
    private long id;  
  
15    @Lob  
    private byte[] pkcs12;  
  
    private String senha;  
  
20    public byte[] getPkcs12() {  
        return pkcs12;  
    }  
  
    public void setPkcs12(byte[] pkcs12) {  
25        this.pkcs12 = pkcs12;  
    }  
  
    public String getSenha() {  
        return senha;  
30    }  
  
    public void setSenha(String senha) {  
        this.senha = senha;  
    }  
}
```

```
        public long getId () {
            return id ;
        }
40
    }
}

```

Listagem A.27: Classe CertificadoACADAO

```
package inf.ufsc.br.certificado;

import inf.ufsc.br.manager.ACAEntityManager;
import inf.ufsc.br.requisicao.ACARequisicao;
5
import javax.persistence.EntityManager;

public class CertificadoACADAO {

10    public CertificadoACADAO () {
        // TODO Auto-generated constructor stub
    }

    public EntityManager getEntityManager ()
15    {
        return ACAEntityManager.getEntityManager ();
    }

    public void save(CertificadoACA certificado)
20    {
        try
        {
            ACAEntityManager.beginTransaction ();
            getEntityManager ().persist (certificado);
25            ACAEntityManager.commit ();
        }catch (RuntimeException e)
        {
            throw e;
        }
30    }

    public CertificadoACA findById (long id)
    {
        try {
```

```
35         CertificadoACA aca = ACAEntityManager.  
            getEntityManager().find(CertificadoACA.class, id  
            );  
  
            return aca;  
        } catch (RuntimeException e) {  
            throw e;  
40     }  
    }  
  
    public void delete(CertificadoACA certificado)  
    {  
45         try{  
            certificado = ACAEntityManager.getEntityManager().  
                getReference(CertificadoACA.class, certificado.  
                    getId());  
            ACAEntityManager.getEntityManager().remove(  
                certificado);  
        } catch (RuntimeException e) {  
            throw e;  
50     }  
    }  
  
}
```

Listagem A.28: Classe CertificadoAtributo

```
package inf.ufsc.br.certificado;  
  
import java.util.Date;  
  
5 import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import javax.persistence.Id;  
import javax.persistence.Lob;  
10 import javax.persistence.OneToOne;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
@Entity  
15 public class CertificadoAtributo {
```

```
    @Id
```

```
    @GeneratedValue
    private long id;
20 private String requerente;
    @Temporal(TemporalType.DATE)
    private Date dataEmissao;
    @Temporal(TemporalType.DATE)
    private Date dataValidade;
25 private String hashCertificado;

    private boolean revogado;

    private long numeroSerial;
30

    public boolean isRevogado() {
        return revogado;
    }

35 public void setRevogado(boolean revogado) {
    this.revogado = revogado;
}

    @Lob
40 private byte[] valorCertificado;

    @OneToOne(cascade = {}, fetch = FetchType.LAZY )
    private CertificadoPKI certificadoPKI;
45

    public String getRequerente() {
        return requerente;
    }

50 public void setRequerente(String requerente) {
    this.requerente = requerente;
}

    public Date getDataEmissao() {
55     return dataEmissao;
}

    public void setDataEmissao(Date dataEmissao) {
60     this.dataEmissao = dataEmissao;
}
```

```
public Date getDataValidade() {  
    return dataValidade;  
}  
65  
public void setDataValidade(Date dataValidade) {  
    this.dataValidade = dataValidade;  
}  
70  
public String getHashCertificado() {  
    return hashCertificado;  
}  
75  
public void setHashCertificado(String hashCertificado) {  
    this.hashCertificado = hashCertificado;  
}  
80  
public byte[] getValorCertificado() {  
    return valorCertificado;  
}  
85  
public void setValorCertificado(byte[] valorCertificado) {  
    this.valorCertificado = valorCertificado;  
}  
90  
public CertificadoPKI getCertificadoPKI() {  
    return certificadoPKI;  
}  
95  
public void setCertificadoPKI(CertificadoPKI certificadoPKI) {  
    this.certificadoPKI = certificadoPKI;  
}  
100  
public long getId() {  
    return id;  
}  
public long getNumeroSerial() {  
    return numeroSerial;  
}  
public void setNumeroSerial(long numeroSerial) {
```

```
        this.numeroSerial = numeroSerial;
105     }
}

```

Listagem A.29: Classe CertificadoAtributoDAO

```
package inf.ufsc.br.certificado;

import inf.ufsc.br.manager.ACAEntityManager;
import inf.ufsc.br.model.config.ACAConfig;
5 import inf.ufsc.br.requisicao.ACAREquisicao;

import java.util.List;

import javax.persistence.EntityManager;
10 import javax.persistence.Query;

public class CertificadoAtributoDAO {

    public CertificadoAtributoDAO () {
15         // TODO Auto-generated constructor stub
    }

    public EntityManager getEntityManager ()
    {
20         return ACAEntityManager.getEntityManager ();
    }

    public void save(CertificadoAtributo certificado)
    {
25         try
        {
            getEntityManager().persist(certificado);

        }catch(RuntimeException e)
30         {
            throw e;
        }
    }

35 public void delete(CertificadoAtributo certificado)
    {
        try{
```

```
        certificado = ACAEntityManager.getEntityManager().
            getReference(CertificadoAtributo.class,
                certificado.getId());
        ACAEntityManager.getEntityManager().remove(
            certificado);
40    } catch (RuntimeException e) {
        throw e;
    }
}

45    public List<CertificadoAtributo> getAllCertificadoAtributos()
    {
        Query q = ACAEntityManager.getEntityManager().createQuery("
            select a from CertificadoAtributo a");

        List<CertificadoAtributo> certAtributos = q.getResultList()
            ;
50    return certAtributos;
    }

    public CertificadoAtributo findById(long id)
    {
55        try {
            CertificadoAtributo cert = ACAEntityManager.
                getEntityManager().find(CertificadoAtributo.
                    class, id);

            return cert;
        } catch (RuntimeException e) {
60            throw e;
        }
    }

    public CertificadoAtributo update(CertificadoAtributo certificado)
65    {
        try {
            CertificadoAtributo merge = ACAEntityManager.
                getEntityManager().merge(certificado);

            return merge;
70
        } catch (RuntimeException e) {
            throw e;
        }
    }
}
```



```

        }
    }
75
    public CertificadoAtributo revogarCertificadoById(long id)
    {
        try {
            CertificadoAtributo cert = ACAEntityManager.
                getEntityManager().find(CertificadoAtributo.
80
                    class, id);

            cert.setRevogado(true);

            update(cert);

85
            return cert;
        } catch (RuntimeException e) {
            ACAEntityManager.rollback();
            throw e;
        }
90
    }
}

```

Listagem A.30: Classe CertificadoAtributoRevogado

```

package inf.ufsc.br.certificado;

import java.util.Date;

5 import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Lob;
import javax.persistence.Temporal;
10 import javax.persistence.TemporalType;

@Entity
public class CertificadoAtributoRevogado {

15     @Id
    @GeneratedValue
    private long id;

    @Temporal(TemporalType.DATE)

```

```
20     private Date datarevogacao;

    private long numeroSerial;

    private String descricaoMotivo;
25
    @Lob
    private byte[] valor;

    public Date getDatarevogacao() {
30         return datarevogacao;
    }

    public void setDatarevogacao(Date datarevogacao) {
        this.datarevogacao = datarevogacao;
35    }

    public long getNumeroSerial() {
        return numeroSerial;
    }
40

    public void setNumeroSerial(long numeroSerial) {
        this.numeroSerial = numeroSerial;
    }

    public String getDescricaoMotivo() {
45         return descricaoMotivo;
    }

    public void setDescricaoMotivo(String descricaoMotivo) {
50         this.descricaoMotivo = descricaoMotivo;
    }

    public byte[] getValor() {
        return valor;
55    }

    public void setValor(byte[] valor) {
        this.valor = valor;
    }

60

    public long getId() {
        return id;
    }
```

```
    }  
65 }  


---



Listagem A.31: Classe CertificadoAtributoRevogadoDAO



---



```
package inf.ufsc.br.certificado;

import inf.ufsc.br.manager.ACAEntityManager;

5 import java.util.List;

import javax.persistence.EntityManager;
import javax.persistence.Query;

10 public class CertificadoAtributoRevogadoDAO {

 public CertificadoAtributoRevogadoDAO () {
 // TODO Auto-generated constructor stub
 }

15 public EntityManager getEntityManager ()
 {
 return ACAEntityManager.getEntityManager ();
 }

20 public void save(CertificadoAtributoRevogado certificado)
 {
 try
 {
25 getEntityManager().persist(certificado);
 } catch (RuntimeException e)
 {
 throw e;
 }
 }

30 }

public void delete(CertificadoAtributoRevogado certificado)
 {
 try{
35 certificado = ACAEntityManager.getEntityManager().
 getReference(CertificadoAtributoRevogado.class,
 certificado.getId());
 ACAEntityManager.getEntityManager().remove(

```


```

```
                certificado);
            }catch (RuntimeException e) {
                throw e;
            }
40     }

    public CertificadoAtributoRevogado FindByNumeroSerie(long numero)
    {
        CertificadoAtributoRevogado retorno = null;
45     Query q = ACAEntityManager.getEntityManager().createQuery("
        select a from CertificadoAtributoRevogado a where
        numeroSerial = " +numero);

        List<CertificadoAtributoRevogado> certAtributos = q.
            getResultList();

        if (certAtributos.size() > 0 )
50     {
            retorno = certAtributos.get(0);
        }
        return retorno;
    }

55     public List<CertificadoAtributoRevogado>
        getAllCertificadoAtributoRevogados()
    {
        Query q = ACAEntityManager.getEntityManager().createQuery("
        select a from CertificadoAtributoRevogado a");

60     List<CertificadoAtributoRevogado> certAtributos = q.
        getResultList();
        return certAtributos;
    }
}
}
```

Listagem A.32: Classe CertificadoPKI

```
package inf.ufsc.br.certificado;

import java.io.Serializable;
import java.util.Date;
5
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Lob;
10 import javax.persistence.Temporal;
import javax.persistence.TemporalType;

@Entity
public class CertificadoPKI implements Serializable{
15

    /**
     *
     */
20 private static final long serialVersionUID = 3610519160252789765L;
@Id
@GeneratedValue
private long id;
private String requerente;
25 private String emissor;
@Temporal(TemporalType.DATE)
private Date emissao;
@Temporal(TemporalType.DATE)
private Date finalValidade;
30 private String hash;

@Lob
private byte[] valor;

35 public byte[] getValor() {
    return valor;
}
public void setValor(byte[] valor) {
    this.valor = valor;
40 }
public String getRequerente() {
    return requerente;
}
public void setRequerente(String requerente) {
45     this.requerente = requerente;
}
public String getEmissor() {
    return emissor;
}
```

```
50     public void setEmissor(String emissor) {
           this.emissor = emissor;
       }
       public Date getEmissao() {
           return emissao;
55     }
       public void setEmissao(Date emissao) {
           this.emissao = emissao;
       }
       public Date getFinalValidade() {
60           return finalValidade;
       }
       public void setFinalValidade(Date finalValidade) {
           this.finalValidade = finalValidade;
       }
65     public String getHash() {
           return hash;
       }
       public void setHash(String hash) {
           this.hash = hash;
70     }
       public long getId() {
           return id;
       }
75 }
```

Listagem A.33: Classe CertificadoPKIDAO

```
package inf.ufsc.br.certificado;

import inf.ufsc.br.manager.ACAEntityManager;

5 import javax.persistence.EntityManager;

public class CertificadoPKIDAO {

       public CertificadoPKIDAO() {
10           // TODO Auto-generated constructor stub
       }

       public EntityManager getEntityManager()
       {
15           return ACAEntityManager.getEntityManager();
       }
```

```
    }

    public void save(CertificadoPKI certificado)
    {
20         try
            {
                ACAEntityManager.beginTransaction();
                getEntityManager().persist(certificado);
                ACAEntityManager.commit();
25         } catch (RuntimeException e)
            {
                ACAEntityManager.rollback();
                throw e;
            }
30     }

    public void delete(CertificadoPKI certificado)
    {
        try{
35             certificado = ACAEntityManager.getEntityManager().
                getReference(CertificadoPKI.class, certificado.
                    getId());
                ACAEntityManager.getEntityManager().remove(
                    certificado);
        } catch (RuntimeException e) {
            throw e;
        }
40     }
}
}
```

Listagem A.34: Classe Lcr

```
package inf.ufsc.br.lcr;

import java.util.Date;

5 import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Lob;
import javax.persistence.Temporal;
10 import javax.persistence.TemporalType;
import javax.persistence.Transient;
```

```
@ Entity
public class Lcr {
15
    @Id
    @GeneratedValue
    private long id;

20
    @Temporal(TemporalType.DATE)
    private Date dataEmissao;

    private String hash;

25
    @Lob
    private byte[] valor;

    public Date getDataEmissao() {
        return dataEmissao;
30
    }

    public void setDataEmissao(Date dataEmissao) {
        this.dataEmissao = dataEmissao;
    }

35
    public String getHash() {
        return hash;
    }

40
    public void setHash(String hash) {
        this.hash = hash;
    }

    public byte[] getValor() {
45
        return valor;
    }

    public void setValor(byte[] valor) {
50
        this.valor = valor;
    }

    public long getId() {
        return id;
    }
}
```


55

}

Listagem A.35: Classe LcrDAO

```
package inf.ufsc.br.lcr;  
  
import inf.ufsc.br.manager.ACAEntityManager;  
import inf.ufsc.br.model.config.LCRConfig;  
5  
public class LcrDAO {  
  
    public LcrDAO() {  
        // TODO Auto-generated constructor stub  
10    }  
  
    public void save(Lcr lcr)  
    {  
        try {  
15            ACAEntityManager.getEntityManager().persist(lcr);  
        } catch (RuntimeException e) {  
            throw e;  
        }  
    }  
20  
    public Lcr findById(long id)  
    {  
        try {  
            Lcr lcr = ACAEntityManager.getEntityManager().find(  
                Lcr.class, id);  
25  
            return lcr;  
        } catch (RuntimeException e) {  
            throw e;  
        }  
30    }  
}
```

Listagem A.36: Classe ACAEntityManager

```
package inf.ufsc.br.manager;
```

```
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
5 import javax.persistence.Persistence;

public class ACAEntityManager {

    private static EntityManager manager;
10    private static EntityManagerFactory fac;

    public static void initManager()
    {
15        if (fac == null)
        {
            fac = Persistence.createEntityManagerFactory("
                ACA_Persist");
        }

        if (manager == null)
20        {
            manager = fac.createEntityManager();
        }
    }

25    public static EntityManager getEntityManager()
    {
        initManager();
        return manager;
30    }

    public static void closeEntityManager()
    {
        fac.close();
35        manager.close();
    }

    public static void beginTransaction()
    {
40        initManager();
        manager.getTransaction().begin();
    }

    public static void commit()
```

```
45     {
        initManager ();
        manager.getTransaction().commit ();
    }

50     public static void rollback ()
    {
        initManager ();
        manager.getTransaction().rollback ();
    }

55 }
}
```

Listagem A.37: Classe ACAConfig

```
package inf.ufsc.br.model.config;

import javax.persistence.Column;
import javax.persistence.Entity;
5 import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
10 @Table(name = "ACACONFIG")
public class ACAConfig {

    @Id
    @GeneratedValue
15     private long id;
    @Column(name="VALCERTIFICADO")
    private int validadeCertificado;

    private String AlgAssinatura;

20     public String getAlgAssinatura() {
        return AlgAssinatura;
    }
    public void setAlgAssinatura(String algAssinatura) {
25         AlgAssinatura = algAssinatura;
    }
    public long getId() {
        return id;
    }
}
```

```
30     public int getValidadeCertificado () {
           return validadeCertificado;
       }
       public void setValidadeCertificado(int validadeCertificado) {
           this.validadeCertificado = validadeCertificado;
35     }
}
}
```

Listagem A.38: Classe ACAConfigDAO

```
package inf.ufsc.br.model.config;

import inf.ufsc.br.manager.ACAEntityManager;

5 public class ACAConfigDAO {

       public ACAConfigDAO () {
           // TODO Auto-generated constructor stub
10     }

       public void save(ACAConfig config)
       {
           try {
15                 ACAEntityManager.getEntityManager().persist(config)
                           ;
           } catch (RuntimeException e) {
               throw e;
           }
       }

20     public void delete(ACAConfig config)
       {
           try {
               config = ACAEntityManager.getEntityManager().
                   getReference(ACAConfig.class, config.getId());
25                 ACAEntityManager.getEntityManager().remove(config);

           } catch (RuntimeException e) {
               throw e;
           }
30     }
}
```

```
public ACAConfig update(ACAConfig config)
{
    try {
35         ACAConfig merge = ACAEntityManager.getEntityManager
            ().merge(config);

        return merge;

    } catch (RuntimeException e) {
40         throw e;
    }
}

public ACAConfig findById(long id)
45 {
    try {
        ACAConfig config = ACAEntityManager.
            getEntityManager().find(ACAConfig.class, id);

        return config;
50     } catch (RuntimeException e) {
        throw e;
    }
}
55 }
```

Listagem A.39: Classe LCRConfig

```
package inf.ufsc.br.model.config;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
5 import javax.persistence.Id;

@Entity
public class LCRConfig {

10     @Id
    @GeneratedValue
    private long id;

    private int tempoPublicacao;
15     private String diretorioPublicacao;
```

```
    private String urlPublicacao;
    private String algAssinatura;

    public String getAlgAssinatura() {
20         return algAssinatura;
    }
    public void setAlgAssinatura(String algAssinatura) {
        this.algAssinatura = algAssinatura;
    }
25    public int getTempoPublicacao() {
        return tempoPublicacao;
    }
    public void setTempoPublicacao(int tempoPublicacao) {
        this.tempoPublicacao = tempoPublicacao;
30    }
    public String getDiretorioPublicacao() {
        return diretorioPublicacao;
    }
    public void setDiretorioPublicacao(String diretorioPublicacao) {
35         this.diretorioPublicacao = diretorioPublicacao;
    }
    public String getUrlPublicacao() {
        return urlPublicacao;
    }
40    public void setUrlPublicacao(String urlPublicacao) {
        this.urlPublicacao = urlPublicacao;
    }
    public long getId() {
        return id;
45    }
}
```

Listagem A.40: Classe LCRConfigDAO

```
package inf.ufsc.br.model.config;

import inf.ufsc.br.manager.ACAEntityManager;

5 public class LCRConfigDAO {

    public LCRConfigDAO() {
        // TODO Auto-generated constructor stub
    }
10 }
```

```
    public void save(LCRConfig config)
    {
        try {
            ACAEntityManager.getEntityManager().persist(config)
                ;
15     } catch (RuntimeException e) {
            throw e;
        }
    }

20     public LCRConfig findById(long id)
    {
        try {
            LCRConfig config = ACAEntityManager.
                getEntityManager().find(LCRConfig.class, id);

25         return config;
        } catch (RuntimeException e) {
            throw e;
        }
    }

30 }
```

Listagem A.41: Classe ACAREquisicao

```
package inf.ufsc.br.requisicao;

import inf.ufsc.br.certificado.CertificadoAtributo;
import inf.ufsc.br.certificado.CertificadoPKI;
5
import java.util.Date;
import java.util.HashSet;
import java.util.Set;

10 import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.OneToOne;
15 import javax.persistence.Temporal;
import javax.persistence.TemporalType;
```

```
@Entity
20 public class ACARequisicao
{
    @Id
    @GeneratedValue
    private long id;

25
    @OneToOne(cascade = {}, fetch = FetchType.LAZY )
    private CertificadoPKI certificadoPKI;
    @Temporal(TemporalType.DATE)
    private Date dataRequisicao;
30
    private boolean emitido;
    @OneToOne(cascade = {}, fetch = FetchType.LAZY )
    private CertificadoAtributo certAtr;

    @OneToMany(mappedBy = "acReq", fetch = FetchType.LAZY, targetEntity
        = inf.ufsc.br.requisicao.AtributoRequisicao.class)
35
    private Set<AtributoRequisicao> listaAtributos = new HashSet<
        AtributoRequisicao>();

    public Set<AtributoRequisicao> getListaAtributos() {
        return listaAtributos;
    }
40
    public void setListaAtributos(Set<AtributoRequisicao>
        listaAtributos) {
        this.listaAtributos = listaAtributos;
    }
    public CertificadoAtributo getCertAtributo() {
        return certAtr;
45
    }
    public void setCertAtributo(CertificadoAtributo certAtr) {
        this.certAtr = certAtr;
    }
    public boolean isEmitido() {
50
        return emitido;
    }
    public void setEmitido(boolean emitido) {
        this.emitido = emitido;
    }
55
    public CertificadoPKI getCertificadoPKI() {
        return certificadoPKI;
    }
    public void setCertificadoPKI(CertificadoPKI cert) {
```



```
        this.certificadoPKI = cert;
60    }
    public Date getDataRequisicao() {
        return dataRequisicao;
    }
    public void setDataRequisicao(Date dataRequisicao) {
65        this.dataRequisicao = dataRequisicao;
    }
    public long getId() {
        return id;
    }
70 }
}
```

Listagem A.42: Classe ACARequisicaoDAO

```
package inf.ufsc.br.requisicao;

import java.util.Collection;
import java.util.List;
5 import javax.persistence.Query;

import inf.ufsc.br.certificado.CertificadoAtributo;
import inf.ufsc.br.certificado.CertificadoPKI;
10 import inf.ufsc.br.manager.ACAEntityManager;

public class ACARequisicaoDAO {

    public ACARequisicaoDAO() {
15        // TODO Auto-generated constructor stub
    }

    public void save(ACARequisicao requisicao)
    {
20        try {
            ACAEntityManager.beginTransaction();
            ACAEntityManager.getEntityManager().persist(
                requisicao);
            ACAEntityManager.commit();
        } catch (RuntimeException e) {
25            ACAEntityManager.rollback();
            throw e;
        }
    }
}
```

```
    }  
  
30    public void delete(ACARequisicao requisicao)  
    {  
        try {  
            requisicao = ACAEntityManager.getEntityManager().  
                getReference(ACARequisicao.class, requisicao.  
                    getId());  
            ACAEntityManager.getEntityManager().remove(  
35                requisicao);  
  
        } catch (RuntimeException e) {  
            throw e;  
        }  
    }  
  
40    public ACARequisicao update(ACARequisicao requisicao)  
    {  
        try {  
            ACARequisicao merge = ACAEntityManager.  
                getEntityManager().merge(requisicao);  
  
45                return merge;  
  
        } catch (RuntimeException e) {  
            throw e;  
50        }  
    }  
  
    public ACARequisicao findById(long id)  
    {  
55        try {  
            ACARequisicao requisicao = ACAEntityManager.  
                getEntityManager().find(ACARequisicao.class, id)  
                ;  
  
            return requisicao;  
        } catch (RuntimeException e) {  
60            throw e;  
        }  
    }  
  
    public ACARequisicao findByCertificado(CertificadoPKI pki)
```

```
65     {
        try {
            ACARequisicao requisicao = ACAEntityManager.
                getEntityManager().find(ACARequisicao.class, pki
                );

                return requisicao;
70     } catch (RuntimeException e) {
            throw e;
        }
    }

75     public Collection<ACARequisicao> getAllRequisicao()
    {
        Query q = ACAEntityManager.getEntityManager().createQuery("
            select a from ACARequisicao a");

        List<ACARequisicao> requisicoes = q.getResultList();
80     return requisicoes;
    }
}
}
```

Listagem A.43: Classe AtributoRequisicao

```
package inf.ufsc.br.requisicao;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
5 import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.Lob;
import javax.persistence.ManyToOne;
import javax.xml.bind.annotation.XmlTransient;

10 @Entity
public class AtributoRequisicao {

    @Id
15     @GeneratedValue
    private long id;

    private String oid;
```

```
20     @Lob
        private byte[] valor;

        @ManyToOne
        @JoinColumn(name = "ACARRequisicao_ID")
25     @XmlTransient
        private ACARRequisicao acReq;

        public ACARRequisicao getAcReq() {
            return acReq;
30     }
        public void setAcReq(ACARRequisicao acReq) {
            this.acReq = acReq;
        }

35     public String getOid() {
            return oid;
        }
        public void setOid(String oid) {
            this.oid = oid;
40     }

        public byte[] getValor() {
            return valor;
        }

45     public void setValor(byte[] valor) {
            this.valor = valor;
        }

50     public long getId() {
            return id;
        }

55 }
```

Listagem A.44: Classe GenerateDataBase

```
package inf.ufsc.br.util;

import java.util.Properties;

5 import javax.persistence.EntityManager;
```

```

import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

public class GenerateDataBase {
10
    public static void main(String[] args) {

        Properties cfg = new Properties();

        cfg.setProperty("hibernate.hbm2ddl.auto", "create");
15
        EntityManagerFactory emf = Persistence.
            createEntityManagerFactory("ACA_Persist",cfg);
        EntityManager emg = emf.createEntityManager();

        emg.close();
20
        emf.close();
    }
}

```

Listagem A.45: persistence.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0" xmlns="http://java.sun.com/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation
    ="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/
    persistence/persistence_1_0.xsd">
    <persistence-unit name="ACA_Persist">
        <class>inf.ufsc.br.requisicao.AtributoRequisicao</class>
        <class>inf.ufsc.br.requisicao.ACAREquisicao</class>
5
        <class>inf.ufsc.br.model.config.LCRConfig</class>
        <class>inf.ufsc.br.model.config.ACAConfig</class>
        <class>inf.ufsc.br.lcr.Lcr</class>
        <class>inf.ufsc.br.certificado.CertificadoPKI</class>
10
        <class>inf.ufsc.br.certificado.CertificadoAtributoRevogado<
            /class>
        <class>inf.ufsc.br.certificado.CertificadoAtributo</class>
        <class>inf.ufsc.br.certificado.CertificadoACA</class>
        <class>inf.ufsc.br.atributos.Atributo</class>

        <properties>
15
            <property name = "hibernate.hbm2ddl.auto" value = "
                none" />

```

```

        <property name="hibernate.format_sql" value="true"/
        >

        <property name="hibernate.dialect" value="org.
            hibernate.dialect.MySQLDialect"/>
20 <property name="hibernate.connection.driver_class"
        value="com.mysql.jdbc.Driver"/>
        <property name="hibernate.connection.url" value="
            jdbc:mysql://localhost/aca"/>
        <property name="hibernate.connection.username"
            value="root"/>
        <property name="hibernate.connection.password"
            value="123456"/>
        </properties>
25 </persistence-unit>
</persistence>

```

Listagem A.46: build.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<project name="MyProject" default="dist" basedir=".">
    <description>
        Build for ACA_Persist
5    </description>
    <!-- set global properties for this build -->
    <property name="src" location="src"/>
    <property name="build" location="build"/>
    <property name="dist" location="dist"/>
10 <property name="build.lib" location="build/lib"/>
    <property name="lib" location="lib"/>

    <path id="master-classpath">
        <fileset dir="${lib}"/>
15     <include name="*.jar"/>
        </fileset>
        <pathelement path="${build}"/>
    </path>

20 <target name="init">
    <!-- Create the time stamp -->
    <tstamp/>
    <!-- Create the build directory structure used by compile -->
    <mkdir dir="${build}"/>
25     <mkdir dir="${build.lib}"/>

```

```

    <!--<copy todir="{build.lib}" flatten="true">
      <path>
        <pathelement path="{lib}/*.jar"/>
      </path>
30 </copy>-->
      <!--<copydir dest="{build.lib}" src="{lib}"></copydir-->
</target>

<target name="compile" depends="init" description="compile the source " >
35 <!-- Compile the java code from {src} into {build} -->
  <javac srcdir="{src}" destdir="{build}" debug="on">
    <classpath refid="master-classpath"/>
  </javac>
</target>

40 <target name="dist" depends="compile" description="generate the
  distribution" >
  <!-- Create the distribution directory -->
  <mkdir dir="{dist}/lib"/>

45 <!-- Put everything in {build} into the MyProject-{DSTAMP}.jar file
  -->
  <jar jarfile="{dist}/lib/ACA_Persist-{DSTAMP}.jar" basedir="{build}"
  />
</target>

<target name="clean" description="clean up" >
50 <!-- Delete the {build} and {dist} directory trees -->
  <delete dir="{build}"/>
  <delete dir="{dist}"/>
</target>
</project>

```

A.1.3 Módulo Webservice

Listagem A.47: Classe ACAConfiguracaoWS

```

package inf.ufsc.br.config;

import java.util.Collection;

5 import inf.ufsc.br.ac.AtributoCadastro;

```

```
import inf.ufsc.br.tributos.Atributo;
import inf.ufsc.br.cadastro.CadastroAtributo;
import inf.ufsc.br.cadastro.CadastroAutoridade;
import inf.ufsc.br.lcr.LCRManager;
10 import inf.ufsc.br.model.config.LCRConfig;

import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
15 import javax.jws.soap.SOAPBinding;
import javax.jws.soap.SOAPBinding.ParameterStyle;

@WebService
@SOAPBinding(parameterStyle = ParameterStyle.WRAPPED)
20 public class ACAConfiguracaoWS {

    @WebMethod
    public void adicionarCertificadoAutoridade( @WebParam(name="
        pkcs12ACA") byte[] pkcs12, @WebParam(name="senha") String senha)
    {
25         CadastroAutoridade cadastroACA = new CadastroAutoridade();

        cadastroACA.adicionarCertificadoAutoridade(pkcs12, senha);
    }

    @WebMethod
    30 public void adicionarAtributo(@WebParam(name="atributo")
        AtributoCadastro atributo)
    {
        CadastroAtributo cadastra = new CadastroAtributo();
        cadastra.adicionarAtributo(atributo);
35     }

    @WebMethod
    public Collection<Atributo> getAtributosCadastrados()
    {
40         CadastroAtributo cadastra = new CadastroAtributo();
        return cadastra.getAtributos();
    }

    @WebMethod
    45 public void configurarLCR(@WebParam(name="lcrconfig") LCRConfig
        lcrCfg)
```



```

        {
            LCRManager man = new LCRManager ();
            man.alterarConfiguracaoLCR (lcr cfg);
        }
50
    }

```

Listagem A.48: Classe EmissaoCertificadoWS

```

package inf.ufsc.br.emissao;

import inf.ufsc.br.certificado.CertificadoAtributo;

5 import java.io.IOException;
import java.security.NoSuchAlgorithmException;
import java.security.cert.CertificateException;
import java.util.Collection;

10 import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
import javax.jws.soap.SOAPBinding;
import javax.jws.soap.SOAPBinding.ParameterStyle;

15 @WebService
@SOAPBinding(parameterStyle = ParameterStyle.WRAPPED)
public class EmissaoCertificadoWS {

20     @WebMethod
    public byte[] gerarCertificado(@WebParam(name ="id_req") long req)
        throws Exception
    {
        EmissaoCertificado emissao = new EmissaoCertificado ();
        return emissao.emitirCertificado (req);

25     }

    @WebMethod
    public Collection<CertificadoAtributo>
        getTodosCertificadosAtributosEmitidos() throws
        NoSuchAlgorithmException, CertificateException, IOException
    {

30         EmissaoCertificado emissao = new EmissaoCertificado ();
        return emissao.getTodosCertificadosEmitidos ();
    }
}

```

```

    @WebMethod
35    public CertificadoAtributo getCertificadoAtributoEmitidosPorId (long
        id) throws NoSuchAlgorithmException, CertificateException,
        IOException
    {
        EmissaoCertificado emissao = new EmissaoCertificado ();
        return emissao.getCertificadoEmitidoById (id);
    }
40
}

```

Listagem A.49: Classe RequisicaoCertificadoWS

```

package inf.ufsc.br.requisicao;

import inf.ufsc.br.ac.AtributoRequisicaoBL;
import inf.ufsc.br.atributos.Atributo;
5 import inf.ufsc.br.cadastro.CadastroAtributo;
import inf.ufsc.br.certificado.Certificado;
import inf.ufsc.br.certificado.CertificadoAtributo;
import inf.ufsc.br.emissao.AtributoRequisicaoManagerBL;
import inf.ufsc.br.emissao.EmissaoCertificado;
10
import java.io.IOException;
import java.security.NoSuchAlgorithmException;
import java.security.cert.CertificateException;
import java.util.Collection;
15 import java.util.Set;

import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
20 import javax.jws.soap.SOAPBinding;
import javax.jws.soap.SOAPBinding.ParameterStyle;

@WebService
@SOAPBinding(parameterStyle = ParameterStyle.WRAPPED)
25 public class RequisicaoCertificadoWS {

    @WebMethod
    public long gerarRequisicaoCertificado (@WebParam(name="certificado"

```

```
        ) byte[] certificado) throws CertificateException, IOException,
        NoSuchAlgorithmException
    {
30
        Certificado cert = new Certificado(certificado);

        RequisicaoCertificado req = new RequisicaoCertificado(cert)
            ;

35
        return req.gerarRequisicaoCertificado();
    }

    @WebMethod
40
    public void adicionarAtributoRequisicao (@WebParam(name="req") long
        req, @WebParam(name="atributo") AtributoRequisicaoBL atributo)
    {
        AtributoRequisicaoManagerBL atributoReq = new
            AtributoRequisicaoManagerBL();
        atributoReq.adicionarAtributoRequisicao(req, atributo);
    }

45
    @WebMethod
    public CertificadoAtributo getCertificadoAtributoPorRequisicao (
        @WebParam(name="req") long req) throws Exception
    {
        EmissaoCertificado emissao = new EmissaoCertificado();
50
        return emissao.getCertificadoEmitidoById(req);
    }

    @WebMethod
    public Collection<Atributo> getAtributosCadastrados ()
55
    {
        CadastroAtributo cadAtributo = new CadastroAtributo();
        return cadAtributo.getAtributos();
    }

60
    @WebMethod
    public Set<RequisicaoBL> getTodasRequisicoes ()
    {
        RequisicaoCertificado req = new RequisicaoCertificado();
        return req.getTodasRequisicao();
65
    }
```

```

    @WebMethod
    public RequisicaoBL getRequisicaoPorNumeroIdentificador (@WebParam(
        name="req") long reqid)
    {
70         RequisicaoCertificado req = new RequisicaoCertificado ();
        RequisicaoBL requi = req.getRequisicao (reqid);

        return requi;
    }
75
}

```

Listagem A.50: Classe RevogarCertificadoWS

```

package inf.ufsc.br.revogacao;

import inf.ufsc.br.certificado.CertificadoAtributoRevogado;
import inf.ufsc.br.lcr.LCRManager;
5
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
import javax.jws.soap.SOAPBinding;
10 import javax.jws.soap.SOAPBinding.ParameterStyle;

@WebService
@SOAPBinding(parameterStyle = ParameterStyle.WRAPPED)
public class RevogarCertificadoWS {
15
    @WebMethod
    public void revogarCertificado (@WebParam(name="revogacao")
        RevogacaoCertificado revogacao) throws Exception
    {
        RevogacaoCertificadoManager revmanager = new
            RevogacaoCertificadoManager ();
20         revmanager.revogarCertificadoAtributo (revogacao);
    }

    @WebMethod
    public void emitirListaCertificadoRevogado () throws Exception
25
    {
        LCRManager lcrman = new LCRManager ();
        lcrman.emitirLCR ();
    }
}

```

```

    }

30    @WebMethod
    public CertificadoAtributoRevogado
        getCertificadoAtributoRevogadoPorNumeroSerial (@WebParam (name="
            numero") long numero)
    {
        RevogacaoCertificadoManager revmanager = new
            RevogacaoCertificadoManager ();
        return revmanager.getCertificadoRevogadoPorNumeroSerial (
            numero);
35    }
}

```

Listagem A.51: webservicesun-jaxws

```

<?xml version="1.0" encoding="UTF-8"?>
<endpoints xmlns="http://java.sun.com/xml/ns/jax-ws/ri/runtime" version="
    2.0">
    <endpoint name="webserviceRequisicao" implementation="inf.ufsc.br.
        requisicao.RequisicaoCertificadoWS"
        url-pattern="/aca/webservicereq.ws" />
5
    <endpoint name="webserviceEmissao" implementation="inf.ufsc.br.
        emissao.EmissaoCertificadoWS"
        url-pattern="/aca/webserviceemis.ws" />

    <endpoint name="webserviceConfiguracao" implementation="inf.ufsc.br
        .config.ACAConfiguracaoWS"
10    url-pattern="/aca/webserviceconfig.ws" />

        <endpoint name="webserviceRevogacao" implementation="inf.
            ufsc.br.revogacao.RevogarCertificadoWS"
            url-pattern="/aca/webservicerevogacao.ws" />
</endpoints>

```

Listagem A.52: web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4"
    xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.
        org/2001/XMLSchema-instance"

```

```

    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun
        .com/xml/ns/j2ee/web-app_2_4.xsd">
5 <display-name>ACA_WS</display-name>
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
10 <welcome-file>default.html</welcome-file>
        <welcome-file>default.htm</welcome-file>
        <welcome-file>default.jsp</welcome-file>
    </welcome-file-list>
    <listener>
15 <listener-class>com.sun.xml.ws.transport.http.servlet.
        WSServletContextListener</listener-class>
    </listener>
    <servlet>
        <servlet-name>WSACA</servlet-name>
        <servlet-class>com.sun.xml.ws.transport.http.servlet.
            WSServlet</servlet-class>
20 </servlet>
    <servlet-mapping>
        <servlet-name>WSACA</servlet-name>
        <url-pattern>/aca/*</url-pattern>
    </servlet-mapping>
25 </web-app>

```

A.2 Módulo Público

Listagem A.53: Página Default

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Default.aspx.vb"
    Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
    w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
5 <html xmlns="http://www.w3.org/1999/xhtml" >
    <head runat="server">
        <title>Untitled Page</title>
    </head>
    <body>
10 <form id="form1" runat="server">

```


35

```

    </div>
  </form>
</body>
</html>

```

Listagem A.54: Classe Default

```

Partial Class _Default
    Inherits System.Web.UI.Page

```

5 End Class

Listagem A.55: Página Solicitar

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Solicitar.aspx.vb"
    Inherits="Solicitar" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
    w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

5 <html xmlns="http://www.w3.org/1999/xhtml" >
  <head runat="server">
    <title>Untitled Page</title>
  </head>
  <body>
10 
    <form id="form1" runat="server">
      <div>
        &nbsp; &nbsp; &nbsp;
15 <asp:Label ID="Label1" runat="server" Style="z-index: 100; left:
        200px; position: absolute;
            top: 171px" Text="Validação do certificado de chaves públicas">
          </asp:Label>
        &nbsp; &nbsp; &nbsp;
        <applet id ="applet" archive="applet/AppletSelecaoCertificado.
          jar"
            code="inf.ufsc.br.applet.AppletSelecao.class" width="
            521" height="164" style="z-index: 104; left: 81px;
            position: absolute; top: 202px">
20 </hr/>

```


End Sub

End Class

Listagem A.57: Página Salvar

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Salvar.aspx.vb"
    Inherits="Salvar" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
    w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

5 <html xmlns="http://www.w3.org/1999/xhtml" >
    <head runat="server">
        <title>Untitled Page</title>
    </head>
    <body>
10     <form id="form1" runat="server">
        <div>

            </div>
15     </form>
    </body>
</html>

```

Listagem A.58: Classe Salvar

```

Imports Emissao

Imports System.IO
Partial Class Salvar
5     Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.
        EventArgs) Handles Me.Load
        Dim emissao As New EmissaoCertificadoWSService

10     Try
        Response.Clear()
        Dim certName As String = "cert.cer"
        Response.ContentType = "APPLICATION/OCTET-STREAM"
        Dim disHeader As String = "Attachment; Filename="" & certName
            & """"
15     Response.AddHeader("Content-Disposition", disHeader)

```

```

    Dim req As Long = Session.Item("AcaRequisicao")

    Dim cert () As Byte = emissao.gerarCertificado(req)
20
    Response.AddHeader("Content-Length", cert.Length.ToString())
    Response.Flush()

    Response.BinaryWrite(cert)
25
    Catch ex As Exception
        Session.Add("Erro", ex)
        Response.Redirect("Erro.aspx")
    End Try
30
End Sub

```

```
End Class
```

Listagem A.59: Página Revogar

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Revogar.aspx.vb"
    Inherits="Revogar" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
    w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
5 <html xmlns="http://www.w3.org/1999/xhtml" >
    <head runat="server">
        <title>Untitled Page</title>
    </head>
    <body>
10    <form id="form1" runat="server">
        <div>
            
            <asp:Label ID="Label1" runat="server" Style="z-index: 100; left: 77
                px; position: absolute;
15            top: 220px" Text="Número da Requisição:"></asp:Label>
            <asp:TextBox ID="TextBox1" runat="server" Font-Bold="True" Style="z
                -index: 101; left: 228px;
                position: absolute; top: 220px"></asp:TextBox>

```

```
<asp:Button ID="Button1" runat="server" Style="z-index: 102; left:
    394px; position: absolute;
    top: 218px" Text="Enviar" Width="116px" />
20 <asp:Label ID="Label2" runat="server" Style="z-index: 103; left: 96
    px; position: absolute;
    top: 168px"></asp:Label>
    &nbsp;

</div>
25 <asp:ObjectDataSource ID="ObjectDataSource1" runat="server"
    SelectMethod="getCertificadoAtributoEmitidosPorId"
    TypeName="Emissao.EmissaoCertificadoWSService">
    <SelectParameters>
        <asp:ControlParameter ControlID="TextBox1" Name="arg0"
            PropertyName="Text" Type="Int64" />
    </SelectParameters>
30 </asp:ObjectDataSource>
    <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="
        False" BackColor="White"
        BorderColor="#3366CC" BorderStyle="None" BorderWidth="1px"
        CellPadding="4" DataSourceID="ObjectDataSource1"
        Font-Size="X-Small" Height="41px" PageSize="5" Style="z-index:
            104; left: 4px;
        position: absolute; top: 274px" Width="555px">
35 <RowStyle BackColor="White" ForeColor="#003399" />
    <Columns>
        <asp:BoundField DataField="requerente" HeaderText="
            requerente" SortExpression="requerente" />
        <asp:BoundField DataField="dataEmissao" HeaderText="
            dataEmissao" SortExpression="dataEmissao" />
        <asp:BoundField DataField="dataValidade" HeaderText="
            dataValidade" SortExpression="dataValidade" />
40 <asp:BoundField DataField="hashCertificado" HeaderText="
            hashCertificado" SortExpression="hashCertificado" />
    </Columns>
    <FooterStyle BackColor="#99CCCC" ForeColor="#003399" />
    <PagerStyle BackColor="#99CCCC" ForeColor="#003399"
        HorizontalAlign="Left" />
    <SelectedRowStyle BackColor="#009999" Font-Bold="True"
        ForeColor="#CCFF99" />
45 <HeaderStyle BackColor="#003399" Font-Bold="True" ForeColor="#
        CCCCFF" />
</asp:GridView>
```

```

    <asp:Button ID="Button2" runat="server" Style="z-index: 105; left:
        168px; position: absolute;
        top: 410px" Text="Revogar" Visible="False" Width="233px" />
    <asp:Label ID="Label3" runat="server" Style="z-index: 106; left: 13
        px; position: absolute;
50     top: 383px" Text="Motivo:" Visible="False"></asp:Label>
    <asp:TextBox ID="TextBox2" runat="server" Style="z-index: 109;
        left: 78px; position: absolute;
        top: 380px" Visible="False" Width="476px"></asp:TextBox>
    </form>
</body>
55 </html>

```

Listagem A.60: Classe Revogar

```

Imports Emissao
Imports Revogacao
Partial Class Revogar
    Inherits System.Web.UI.Page
5
    Private cert As certificadoAtributo

    Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.
        EventArgs) Handles Button1.Click
        Dim emissao As New EmissaoCertificadoWSService
10     Try
            cert = emissao.getCertificadoAtributoEmitidosPorId(TextBox1.
                Text)
            TextBox2.Visible = True
            Label3.Visible = True
            Button2.Visible = True
15     Catch ex As Exception
            Label2.Text = ex.Message
        End Try

    End Sub
20

    Protected Sub Button2_Click(ByVal sender As Object, ByVal e As System.
        EventArgs) Handles Button2.Click
        Dim revogacao As New revogacaoCertificado
        Dim rev As New RevogarCertificadoWSService
        revogacao.motivo = TextBox2.Text
25     revogacao.requisicao = TextBox1.Text

```

```

    Try
        rev.revogarCertificado (revogacao)

30         TextBox2.Visible = False
           Label3.Visible = False
           Button2.Visible = False

           Label2.Text = "Certificado revogado com sucesso!"
35     Catch ex As Exception
           Label2.Text = ex.Message
    End Try

    End Sub
40 End Class

```

Listagem A.61: Página Requisicao

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Requisicao.aspx.vb
  " Inherits="Requisicao" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
  w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

5 <html xmlns="http://www.w3.org/1999/xhtml" >
  <head runat="server">
    <title>Untitled Page</title>
  </head>
  <body>
10   <form id="form1" runat="server">
    <div>
      
      <asp:Label ID="Label1" runat="server" Style="z-index: 100; left: 74
        px; position: absolute;
15      top: 201px" Text="Número da Requisição"></asp:Label>
      <asp:Label ID="Label2" runat="server" Font-Bold="True" Style="z-
        index: 103; left: 232px;
        position: absolute; top: 203px" Text="Label"></asp:Label>

    </div>
20   </form>
  </body>
</html>

```

Listagem A.62: Classe Requisicao

```

Partial Class Requisicao
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.
        EventArgs) Handles Me.Load
        Dim req As Long = Session.Item("requisicao")

        Label2.Text = req
    End Sub
10 End Class

```

Listagem A.63: Página Instalar

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Instalar.aspx.vb"
    Inherits="Instalar" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
    w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

5 <html xmlns="http://www.w3.org/1999/xhtml" >
    <head runat="server">
        <title>Untitled Page</title>
    </head>
    <body>
10     <form id="form1" runat="server">
        <div>
            
            <asp:Label ID="Label1" runat="server" Style="z-index: 100; left:
                102px; position: absolute;
15             top: 204px" Text="Número da Requisição:"></asp:Label>
            <asp:TextBox ID="TextBox1" runat="server" Font-Bold="True" Style="z
                -index: 101; left: 253px;
                position: absolute; top: 202px"></asp:TextBox>
            <asp:Button ID="Button1" runat="server" Style="z-index: 102; left:
                422px; position: absolute;
                top: 202px" Text="Enviar" Width="116px" />
20     <asp:Label ID="Label2" runat="server" Style="z-index: 105; left:
                178px; position: absolute;

```

```
        top: 268px"></asp:Label>

    </div>
</form>
25 </body>
</html>
```

Listagem A.64: Classe Instalar

```
Imports Emissao
Imports RequisicaoWeb
Partial Class Instalar
    Inherits System.Web.UI.Page
5
    Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.
        EventArgs) Handles Button1.Click
        Dim req As New RequisicaoCertificadoWSService
        Dim acaReq As requisicaoBL
        Dim redic As Boolean
10    Try
        acaReq = req.getRequisicaoPorNumeroIdentificador(TextBox1.Text)

        If (acaReq IsNot Nothing) Then

15            If (acaReq.emitido) Then
                Label2.Text = "Um certificado já foi emitido para essa
                    requisição"
                redic = False
            Else
                Session.Add("AcaRequisicao", TextBox1.Text)
20                redic = True
            End If
        Else
            Label2.Text = "Requisição inexistente"
            redic = False
25        End If
        Catch ex As Exception
            Session.Add("Erro", ex)
            Response.Redirect("Erro.aspx")
        End Try
30

        If (redic = True) Then
            Response.Redirect("Atributos.aspx")
        End If
```


End Sub

35 **End Class**

Listagem A.65: Página Erro

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Erro.aspx.vb"
    Inherits="Erro" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
    w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

5 <html xmlns="http://www.w3.org/1999/xhtml" >
  <head runat="server">
    <title>Untitled Page</title>
  </head>
  <body>
10   <form id="form1" runat="server">
    <div>
      
      <asp:Label ID="Label1" runat="server" Style="z-index: 101; left: 20
        px; position: absolute;
15       top: 172px" Text="Erro:" ForeColor="Red"></asp:Label>

    </div>
  </form>
</body>
20 </html>

```

Listagem A.66: Classe Erro

Partial Class Erro

Inherits **System.Web.UI.Page**

5 Protected **Sub** Page_Load(ByVal sender As Object, ByVal e As **System.**
EventArgs) Handles Me.Load

Dim ex As Exception = Session.Item("Erro")

Label1.Text = ex.Message

End Sub

10 **End Class**

Listagem A.67: Página Consulta

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Consulta.aspx.vb"
    Inherits="Consulta" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
    w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

5 <html xmlns="http://www.w3.org/1999/xhtml" >
  <head runat="server">
    <title>Untitled Page</title>
  </head>
  <body>
10   <form id="form1" runat="server">
    <div>
      
      <asp:Label ID="Label1" runat="server" Style="z-index: 100; left: 49
        px; position: absolute;
15       top: 197px" Text="Número Serial:"></asp:Label>
      <asp:TextBox ID="TextBox1" runat="server" Font-Bold="True" Style="z
        -index: 101; left: 206px;
        position: absolute; top: 195px"></asp:TextBox>
      <asp:Button ID="Button1" runat="server" Style="z-index: 102; left:
        373px; position: absolute;
        top: 195px" Text="Enviar" Width="116px" />
20     <asp:Label ID="Label2" runat="server" Style="z-index: 103; left:
        162px; position: absolute;
        top: 159px"></asp:Label>
      <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="
        False" BackColor="White"
        BorderColor="#3366CC" BorderStyle="None" BorderWidth="1px"
        CellPadding="4" DataSourceID="ObjectDataSource1"
        Font-Size="X-Small" Height="41px" PageSize="5" Style="z-index:
        104; left: 18px;
25       position: absolute; top: 246px" Width="555px">
      <RowStyle BackColor="White" ForeColor="#003399" />
      <Columns>
        <asp:BoundField DataField="numeroSerial" HeaderText="N
          &#250;mero Serial" SortExpression="numeroSerial" />
        <asp:BoundField DataField="datarevogacao" HeaderText="Data
          Revogacao" SortExpression="datarevogacao" />
30       <asp:BoundField DataField="descricaoMotivo" HeaderText="

```

```

        Motivo" SortExpression="descricaoMotivo" />
    </Columns>
    <FooterStyle BackColor="#99CCCC" ForeColor="#003399" />
    <PagerStyle BackColor="#99CCCC" ForeColor="#003399"
        HorizontalAlign="Left" />
    <SelectedRowStyle BackColor="#009999" Font-Bold="True"
        ForeColor="#CCFF99" />
35    <HeaderStyle BackColor="#003399" Font-Bold="True" ForeColor="#
        CCCCFF" />
</asp:GridView>
<asp:ObjectDataSource ID="ObjectDataSource1" runat="server"
    SelectMethod="getCertificadoAtributoRevogadoPorNumeroSerial"
    TypeName="Revogacao.RevogarCertificadoWSService">
    <SelectParameters>
40    <asp:ControlParameter ControlID="TextBox1" Name="numero"
        PropertyName="Text" Type="Int64" />
    </SelectParameters>
</asp:ObjectDataSource>

</div>
45 </form>
</body>
</html>

```

Listagem A.68: Classe Consulta

```

Imports Revogacao

Partial Class Consulta
5    Inherits System.Web.UI.Page

    Private cert As certificadoAtributoRevogado

    Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.
        EventArgs) Handles Button1.Click
10    Dim rev As New RevogarCertificadoWSService
        Try
            cert = rev.getCertificadoAtributoRevogadoPorNumeroSerial(
                TextBox1.Text)
        Catch ex As Exception
            Label2.Text = ex.Message
15    End Try
    End Sub

```

End Class

Listagem A.69: Página CertificadosEmitidos

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="
  CertificadosEmitidos.aspx.vb" Inherits="CertificadosEmitidos" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
  w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

5 <html xmlns="http://www.w3.org/1999/xhtml" >
  <head runat="server">
    <title>Untitled Page</title>
  </head>
  <body>
10   <form id="form1" runat="server">
    <div>
      
      <asp:GridView ID="GridView1" runat="server" AllowPaging="True"
        AutoGenerateColumns="False"
15      BackColor="White" BorderColor="#3366CC" BorderStyle="None"
        BorderWidth="1px"
        CellPadding="4" DataSourceID="AtributosEmitidosDataSource" Font
          -Size="X-Small"
        PageSize="5" Style="z-index: 101; left: 3px; position: absolute
          ; top: 177px"
        Width="567px">
      <RowStyle BackColor="White" ForeColor="#003399" />
20     <Columns>
      <asp:BoundField DataField="requerente" HeaderText="
        Requerente" SortExpression="requerente" />
      <asp:BoundField DataField="dataEmissao" HeaderText="Data
        Emissao" SortExpression="dataEmissao" />
      <asp:BoundField DataField="dataValidade" HeaderText="Data
        Validade" SortExpression="dataValidade" />
      <asp:BoundField DataField="hashCertificado" HeaderText="
        Hash do Certificado" SortExpression="hashCertificado" />
25     <asp:ButtonField ButtonType="Button" DataTextField="
        certificadoPKI" DataTextFormatString="Certificado"
        HeaderText="Certificado" Text="Certificado" CommandName
          = "SalvarICP"/>
    </Columns>
  </form>
  </div>
  </body>
</html>

```

```

        <FooterStyle BackColor="#99CCCC" ForeColor="#003399" />
        <PagerStyle BackColor="#99CCCC" ForeColor="#003399"
            HorizontalAlign="Left" />
30    <SelectedRowStyle BackColor="#009999" Font-Bold="True"
            ForeColor="#CCFF99" />
        <HeaderStyle BackColor="#003399" Font-Bold="True" ForeColor="#
            CCCCFF" />
    </asp:GridView>
    <asp:ObjectDataSource ID="AtributosEmitidosDataSource" runat="
        server" SelectMethod="getTodosCertificadosAtributosEmitidos"
        TypeName="Emissao.EmissaoCertificadoWSService"></
        asp:ObjectDataSource>
35
    </div>
    </form>
</body>
</html>

```

Listagem A.70: Classe CertificadosEmitidos

```

Imports System.data
Imports Emissao
Partial Class CertificadosEmitidos
    Inherits System.Web.UI.Page
5
    Protected Sub GridView1_RowCommand(ByVal sender As Object, ByVal e As
        System.Web.UI.WebControls.GridViewCommandEventArgs) Handles
        GridView1.RowCommand
        If (e.CommandName = "SalvarICP") Then
            Dim row As Integer = -1
            Integer.TryParse(e.CommandArgument, row)
10            If (row = -1) Then
                Return
            End If
        End If
    End Sub
15 End Class

```

Listagem A.71: Página Atributos

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Atributos.aspx.vb"
    Inherits="Atributos" %>

```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
    w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

5 <html xmlns="http://www.w3.org/1999/xhtml" >
  <head runat="server">
    <title>Untitled Page</title>
  </head>
  <body>
10   <form id="form1" runat="server">
    <div>
      
      <asp:DropDownList ID="DropDownList1" runat="server" DataSourceID="
        ObjectDataSource2"
15      DataTextField="0ID" DataValueField="descricao" Style="z-index:
        100; left: 155px; position: absolute;
        top: 195px" Width="341px">
    </asp:DropDownList>
    <asp:ObjectDataSource ID="ObjectDataSource2" runat="server"
      SelectMethod="getAtributosCadastrados"
      TypeName="RequisicaoWeb.RequisicaoCertificadoWSService"></
        asp:ObjectDataSource>
20   <asp:ObjectDataSource ID="ObjectDataSource1" runat="server"
      SelectMethod="getAtributosCadastrados"
      TypeName="Emissao.RequisicaoCertificadoWSService"></
        asp:ObjectDataSource>
    <asp:Label ID="Label1" runat="server" Style="z-index: 101; left:
      156px; position: absolute;
      top: 166px" Text="Selecione o atributo"></asp:Label>
    <asp:TextBox ID="TextBox1" runat="server" Height="94px" Style="z-
      index: 102; left: 157px;
25      position: absolute; top: 292px" Width="336px"></asp:TextBox>
    <asp:Label ID="Label2" runat="server" Style="z-index: 103; left:
      157px; position: absolute;
      top: 263px" Text="Informações do atributo"></asp:Label>
    <asp:Label ID="Label3" runat="server" Style="z-index: 104; left:
      158px; position: absolute;
      top: 229px" Text="Descrição"></asp:Label>
30   <asp:Label ID="Label4" runat="server" Style="z-index: 105; left:
      229px; position: absolute;
      top: 229px"></asp:Label>
    <asp:Button ID="Button1" runat="server" Style="z-index: 106; left:
```

```

        156px; position: absolute;
        top: 399px" Text="Adicionar" Width="114px" />
<asp:Button ID="Button2" runat="server" Style="z-index: 109; left:
        409px; position: absolute;
35         top: 399px" Text="Emitir" Width="91px" />

        </div>
        </form>
</body>
40 </html>

```

Listagem A.72: Classe Atributos

```

Imports Emissao
Imports RequisicaoWeb
Partial Class Atributos
    Inherits System.Web.UI.Page
5
    Protected Sub DropDownList1_SelectedIndexChanged(ByVal sender As Object
        , ByVal e As System.EventArgs) Handles DropDownList1.
        SelectedIndexChanged
        Label4.Text = DropDownList1.SelectedItem.Value
    End Sub

10    Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.
        EventArgs) Handles Button1.Click
        Dim requisicao As New RequisicaoCertificadoWSService

        Try
            Dim atribReq As New atributoRequisicaoBL
15
            atribReq.oid = DropDownList1.SelectedItem.Text
            Dim encoding As New System.Text.ASCIIEncoding()
            Dim valor() As Byte = encoding.GetBytes(TextBox1.Text)

20
            atribReq.valor = valor

            Dim req As Long = Session.Item("AcaRequisicao")
            requisicao.adicionarAtributoRequisicao(req, atribReq)

25
            TextBox1.Text = ""
        Catch ex As Exception
            Session.Add("Erro", ex)
            Response.Redirect("Erro.aspx")

```

```

    End Try
30 End Sub

Protected Sub Button2_Click(ByVal sender As Object, ByVal e As System.
    EventArgs) Handles Button2.Click
    Response.Redirect("Salvar.aspx")
End Sub
35 End Class

```

Listagem A.73: web.config

```

<?xml version="1.0"?>
<!--
    Note: As an alternative to hand editing this file you can use the
    web admin tool to configure settings for your application. Use
5 the Website->Asp.Net Configuration option in Visual Studio.
    A full list of settings and comments can be found in
    machine.config.comments usually located in
    | Windows | Microsoft .Net | Framework | v2.x | Config
-->
10 <configuration>
    <appSettings>
        <add key="Emissao.EmissaoCertificadoWSService" value="http:
            //localhost:8080/ACA_WS/aca/webserviceemis.ws"/>
        <add key="RequisicaoWeb.RequisicaoCertificadoWSService"
            value="http://localhost:8080/ACA_WS/aca/webservicereq.ws
            "/>
        <add key="Revogacao.RevogarCertificadoWSService" value="http://
            localhost:8080/ACA_WS/aca/webservicerevogacao.ws"/>
15 </appSettings>
    <connectionStrings/>
    <system.web>
        <!--
20 Set compilation debug="true" to insert debugging
        symbols into the compiled page. Because this
        affects performance, set this value to true only
        during development.

        Visual Basic options:
25 Set strict="true" to disallow all data type conversions
        where data loss can occur.
        Set explicit="true" to force declaration of all variables.
-->
    <compilation debug="true" strict="false" explicit="true"/>

```



```

30         <pages>
           <namespaces>
             <clear />
             <add namespace="System" />
             <add namespace="System.Collections" />
35         <add namespace="System.Collections.
           Specialized" />
             <add namespace="System.Configuration" />
             <add namespace="System.Text" />
             <add namespace="System.Text.
           RegularExpressions" />
             <add namespace="System.Web" />
40         <add namespace="System.Web.Caching" />
             <add namespace="System.Web.SessionState" />
             <add namespace="System.Web.Security" />
             <add namespace="System.Web.Profile" />
             <add namespace="System.Web.UI" />
45         <add namespace="System.Web.UI.WebControls" /
           >
             <add namespace="System.Web.UI.WebControls.
           WebParts" />
             <add namespace="System.Web.UI.HtmlControls"
           />
           </namespaces>
         </pages>
50         <!--
           The <authentication> section enables configuration
           of the security authentication mode used by
           ASP.NET to identify an incoming user.
           →
55         <authentication mode="Windows" />
           <!--
           The <customErrors> section enables configuration
           of what to do if/when an unhandled error occurs
           during the execution of a request. Specifically ,
60         it enables developers to configure html error pages
           to be displayed in place of a error stack trace.

           <customErrors mode="RemoteOnly" defaultRedirect="GenericErrorPage.
           htm">
             <error statusCode="403" redirect="NoAccess.htm" />
65         <error statusCode="404" redirect="FileNotFound.htm" />
           </customErrors>

```

```
    →  
    </system.web>  
</configuration>
```

A.3 Módulo Administrativo

Listagem A.74: Classe Repositorio

```
Imports System.Security.Cryptography.X509Certificates  
  
Public Class Repositorio  
  
5     Private store As X509Store  
  
    Public Sub New()  
        store = New X509Store(StoreName.My, StoreLocation.CurrentUser)  
10     store.Open(OpenFlags.ReadOnly)  
    End Sub  
  
    Public Function GetCertificados()  
15     Return store.Certificates  
    End Function  
  
    Public Sub Atualizar()  
        store.Close()  
20     store = New X509Store(StoreName.My, StoreLocation.CurrentUser)  
        store.Open(OpenFlags.ReadOnly)  
    End Sub  
End Class
```

Listagem A.75: Classe frmRequisicao

```
Imports System.Security.Cryptography.X509Certificates  
  
Public Class frmRequisicao  
  
5     Private rep As Repositorio  
  
    Private Sub frmRequisicao_Load(ByVal sender As System.Object, ByVal e  
        As System.EventArgs) Handles MyBase.Load
```

```
        rep = New Repositorio

10        cboCertificados.DataSource = rep.GetCertificados
End Sub

Private Sub btnVisualizar_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnVisualizar.Click
    Dim cert As X509Certificate2
15    cert = cboCertificados.SelectedItem
        X509Certificate2UI.DisplayCertificate(cert)
End Sub

Private Sub cboCertificados_SelectedIndexChanged(ByVal sender As System
    .Object, ByVal e As System.EventArgs) Handles cboCertificados.
    SelectedIndexChanged
20    Dim cert As X509Certificate2
        cert = cboCertificados.SelectedItem

        txtRequerente.Text = cert.Subject
        txtEmissor.Text = cert.Issuer
25    txtValidade.Text = cert.NotBefore
End Sub

Private Sub btnEnviar_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnEnviar.Click

30    Try
        Dim req As New Requisicao.RequisicaoCertificadoWSService
        Dim cert As X509Certificate2
        cert = cboCertificados.SelectedItem
        txtRequisicao.Text = req.gerarRequisicaoCertificado(cert.
            RawData)
35
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try

40 End Sub

Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnOK.Click
    Me.Close()
End Sub
```

45 **End Class**

Listagem A.76: Classe FrmEmissao

```
Public Class FrmEmissao

    Private requester As Requisicao.requisicaoBL
    Private indice As Integer

5
    Private Sub btnCancelar_Click(ByVal sender As System.Object , ByVal e As
        System.EventArgs) Handles btnCancelar.Click
        Me.Close()
    End Sub

10
    Private Sub btnValidar_Click(ByVal sender As System.Object , ByVal e As
        System.EventArgs) Handles btnValidar.Click
        Dim webReq As New Requisicao.RequisicaoCertificadoWSService
        If (txtRequisicao.Text <> "") Then
            requester = webReq.getRequisicaoPorNumeroIdentificador(
                txtRequisicao.Text)
            If (requester IsNot Nothing) Then
15
                btnNext.Enabled = True
            Else
                btnNext.Enabled = False
            End If
        End If
20
        indice = 0
    End Sub

    Public Sub SetNumeroRequisicao(ByVal numero As Long)
        txtRequisicao.Text = numero
25
    End Sub

    Private Sub btnNext_Click(ByVal sender As System.Object , ByVal e As
        System.EventArgs) Handles btnNext.Click
        Select Case indice
            Case 0

30

        End Select

    End Sub
End Class
```

Listagem A.77: Classe frmConfiguracao

```
Imports System.IO

Public Class frmConfiguracao

5     Private Sub btnSelecionar_Click(ByVal sender As System.Object, ByVal e
        As System.EventArgs) Handles btnSelecionar.Click
        OpenFileDialog1.ShowDialog()
        Dim pfx As String

        pfx = OpenFileDialog1.FileName
10     txtArquivo.Text = pfx
    End Sub

    Private Sub btnSalvar_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles btnSalvar.Click
        Dim config As New Configuracao.ACAConfiguracaoWSService
15     Try
        If (txtArquivo.Text <> "") Then
            Dim st As New IO.FileStream(txtArquivo.Text, FileMode.Open)

            Dim bytes(CInt(st.Length - 1)) As Byte
20
            st.Read(bytes, 0, CInt(st.Length))

            st.Close()

25     config.adicionarCertificadoAutoridade(bytes, txtSenha.Text)
        End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
30     End Sub

    Private Sub btnSalvarAtributo_Click(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles btnSalvarAtributo.Click
        Try
            Dim config As New Configuracao.ACAConfiguracaoWSService
35     Dim atributo As New Configuracao.atributoCadastro
            atributo.OID = txtOID.Text
            atributo.descricao = txtDescricao.Text
            atributo.tipoSpecified = True
            Select Case cboTipoAtributo.SelectedIndex
```

```
40         Case 0
           atributo.tipo = Configuracao.tipoAtributo.NUMERO
         Case 1
           atributo.tipo = Configuracao.tipoAtributo.STRING
         Case 2
45         atributo.tipo = Configuracao.tipoAtributo.BYTEARRAY
       End Select

       config.adicionarAtributo(atributo)
       MessageBox.Show("Adicionado")
50       AtualizarGrid()
     Catch ex As Exception
       MessageBox.Show(ex.Message)
     End Try
End Sub

55 Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnOK.Click
  Me.Close()

End Sub

60 Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnCancel.Click
  Me.Close()

End Sub

65 Private Sub AtualizarGrid()
  Try
    Dim config As New Configuracao.ACAConfiguracaoWSService
    Dim cadastrados() As Configuracao.atributo
    cadastrados = config.getAtributosCadastrados()
70    If (cadastrados IsNot Nothing) Then
      dtgAtributos.DataSource = cadastrados
    End If
    Catch ex As Exception

75    End Try
End Sub

Private Sub tpgAtributos_Enter(ByVal sender As System.Object, ByVal e
  As System.EventArgs) Handles tpgAtributos.Enter
  AtualizarGrid()
```

```
80      End Sub

      Private Sub Button1_Click(ByVal sender As System.Object , ByVal e As
        System.EventArgs) Handles Button1.Click
          FolderBrowserDialog1.ShowDialog()

85          txtDiretorioLCR.Text = FolderBrowserDialog1.SelectedPath
      End Sub

      Private Sub Button2_Click(ByVal sender As System.Object , ByVal e As
        System.EventArgs) Handles Button2.Click
          Dim lcrConfig As New Configuracao.lcrConfig

90          lcrConfig.tempoPublicacao = TextBox1.Text
          lcrConfig.algAssinatura = ComboBox1.Text
          lcrConfig.diretorioPublicacao = txtDiretorioLCR.Text
          lcrConfig.urlPublicacao = ""

95          Dim lcrws As New Configuracao.ACAConfiguracaoWSService

          Try
            lcrws.configurarLCR(lcrConfig)
100            MessageBox.Show("Salvo com sucesso!")
          Catch ex As Exception
            MessageBox.Show(ex.Message)
          End Try

105      End Sub
End Class
```

Listagem A.78: Classe Form

```
Imports System.IO

Public Class frmPrincipal

5      Private Sub treeAutoridade_AfterSelect(ByVal sender As System.Object ,
        ByVal e As System.Windows.Forms.TreeViewEventArgs) Handles
        treeAutoridade.AfterSelect
          Dim nodo As TreeNode = treeAutoridade.SelectedNode
          Try
            If (nodo.Text = "Emitidos") Then
              Dim webEmissor As New emissao.EmissaoCertificadoWSService
```

```
10         Dim certificados () As emissao.certificadoAtributo =
            webEmissor.getTodosCertificadosAtributosEmitidos ()
        If (certificados IsNot Nothing) Then
            CertificadoAtributoBindingSource.DataSource =
                certificados
            DataGridView1.BringToFront ()
        End If
15     End If

    If (nodo.Text = "Requisicao") Then
        Dim webRequisicao As New Requisicao.
            RequisicaoCertificadoWSService
        Dim reqs () As Requisicao.requisicaoBL = webRequisicao.
            getTodasRequisicoes ()
20     If (reqs IsNot Nothing) Then
        BindingSource1.DataSource = reqs
        dtgAutoridade.BringToFront ()
    End If
    End If

25     If (nodo.Text = "LCR") Then
        Dim rev As New LCR.RevogarCertificadoWSService
    End If
    Catch ex As Exception
        MessageBox.Show(ex.Message)
30     End Try

End Sub

Private Sub GeralToolStripMenuItem_Click(ByVal sender As System.Object ,
    ByVal e As System.EventArgs) Handles GeralToolStripMenuItem.Click
35     Dim fConfig As New frmConfiguracao
        fConfig.ShowDialog ()
End Sub

Private Sub treeAutoridade_NodeMouseClicked(ByVal sender As System.Object
    , ByVal e As System.Windows.Forms.TreeNodeMouseClickEventArgs)
    Handles treeAutoridade.NodeMouseClicked
40     If (e.Button = Windows.Forms.MouseButtons.Right) Then
        Dim nodo As TreeNode = e.Node
        If (nodo.Text = "Requisicao") Then
            ctxmnuNovo.Show(sender , e.Location)
        End If
45     If (nodo.Text = "LCR") Then
```



```
        ctxLCR.Show(sender, e.Location)
    End If
End If

50 End Sub

Private Sub NovoToolStripMenuItem_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles NovoToolStripMenuItem.Click
    Dim frmRequisicao As New frmRequisicao

55     frmRequisicao.ShowDialog()
End Sub

Private Sub dtgAutoridade_CellMouseClick(ByVal sender As System.Object,
    ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)
    Handles dtgAutoridade.CellMouseClick
    If (e.Button = Windows.Forms.MouseButtons.Right) Then
60         ctxEmitir.Show(sender, e.Location)
    End If
End Sub

Private Sub EmitirToolStripMenuItem_Click(ByVal sender As System.Object
    , ByVal e As System.EventArgs) Handles EmitirToolStripMenuItem.Click
65     Dim frmEmitir As New FrmEmissao
    Dim req As Requisicao.requisicaoBL
    req = dtgAutoridade.SelectedRows.Item(0).DataBoundItem
    frmEmitir.ShowDialog()
End Sub

70 Private Sub EmitirToolStripMenuItem1_Click(ByVal sender As System.
    Object, ByVal e As System.EventArgs) Handles
    EmitirToolStripMenuItem1.Click
    Dim rev As New LCR.RevogarCertificadoWSService
    Try
        rev.emitirListaCertificadoRevogado()
75     Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try

End Sub

80 End Class
```

Listagem A.79: Classe ConfiguracaoWSCliente

```

i»i '
-----

' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:2.0.50727.3053
5 '
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
'
-----

10
Option Strict Off
Option Explicit On

Imports System
15 Imports System.ComponentModel
Imports System.Diagnostics
Imports System.Web.Services
Imports System.Web.Services.Protocols
Imports System.Xml.Serialization
20
',
'This source code was auto-generated by Microsoft.VSDesigner, Version
  2.0.50727.3053.
',
Namespace Configuracao
25
'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
  "2.0.50727.3053"), _
  System.Diagnostics.DebuggerStepThroughAttribute(), _
  System.ComponentModel.DesignerCategoryAttribute("code"), _
30 System.Web.Services.WebServiceBindingAttribute(Name:="
    ACAConfiguracaoWSPortBinding", [Namespace]:= "http://config.br.ufsc.
    inf/")> _
Partial Public Class ACAConfiguracaoWSService
  Inherits System.Web.Services.Protocols.SoapHttpClientProtocol

  Private getAtributosCadastradosOperationCompleted As System.

```

```
        Threading.SendOrPostCallback
35
    Private adicionarCertificadoAutoridadeOperationCompleted As System.
        Threading.SendOrPostCallback

    Private adicionarAtributoOperationCompleted As System.Threading.
        SendOrPostCallback

40    Private configurarLCROperationCompleted As System.Threading.
        SendOrPostCallback

    Private useDefaultCredentialsSetExplicitly As Boolean

    '''<remarks/>
45    Public Sub New()
        MyBase.New
        Me.Url = Global.WindowsApplication1.My.MySettings.Default.
            WindowsApplication1_Configuracao_ACAConfiguracaoWSService
        If (Me.IsLocalFileSystemWebService(Me.Url) = true) Then
            Me.UseDefaultCredentials = true
50            Me.useDefaultCredentialsSetExplicitly = false
        Else
            Me.useDefaultCredentialsSetExplicitly = true
        End If
    End Sub

55    Public Shadows Property Url() As String
        Get
            Return MyBase.Url
        End Get
60        Set
            If (((Me.IsLocalFileSystemWebService(MyBase.Url) = true) _
                AndAlso (Me.useDefaultCredentialsSetExplicitly
                    = false)) _
                AndAlso (Me.IsLocalFileSystemWebService(value)
                    = false)) Then
                MyBase.UseDefaultCredentials = false
65            End If
            MyBase.Url = value
        End Set
    End Property

70    Public Shadows Property UseDefaultCredentials() As Boolean
```

```

        Get
            Return MyBase.UseDefaultCredentials
        End Get
        Set
75         MyBase.UseDefaultCredentials = value
            Me.useDefaultCredentialsSetExplicitly = true
        End Set
    End Property

80     '''<remarks/>
    Public Event getAtributosCadastradosCompleted As
        getAtributosCadastradosCompletedEventHandler

        '''<remarks/>
    Public Event adicionarCertificadoAutoridadeCompleted As
        adicionarCertificadoAutoridadeCompletedEventHandler

85     '''<remarks/>
    Public Event adicionarAtributoCompleted As
        adicionarAtributoCompletedEventHandler

        '''<remarks/>
90     Public Event configurarLCRCompleted As
        configurarLCRCompletedEventHandler

        '''<remarks/>
    <System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
        RequestNamespace:="http://config.br.ufsc.inf/",
        ResponseNamespace:="http://config.br.ufsc.inf/", Use:=System.Web
        .Services.Description.SoapBindingUse.Literal, ParameterStyle:=
        System.Web.Services.Protocols.SoapParameterStyle.Wrapped)> _
    Public Function getAtributosCadastrados() As <System.Xml.
        Serialization.XmlElementAttribute("return", Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> atributo()
95         Dim results() As Object = Me.Invoke("getAtributosCadastrados",
            New Object(-1) {})
            Return CType(results(0), atributo())
    End Function

        '''<remarks/>
100    Public Overloads Sub getAtributosCadastradosAsync()
        Me.getAtributosCadastradosAsync(Nothing)
    End Sub

```

```

'''<remarks/>
105 Public Overloads Sub getAtributosCadastradosAsync (ByVal userState
    As Object)
    If (Me.getAtributosCadastradosOperationCompleted Is Nothing)
        Then
            Me.getAtributosCadastradosOperationCompleted = AddressOf Me
                .OngetAtributosCadastradosOperationCompleted
        End If
    Me.InvokeAsync ("getAtributosCadastrados", New Object(-1) {}, Me
        .getAtributosCadastradosOperationCompleted, userState)
110 End Sub

Private Sub OngetAtributosCadastradosOperationCompleted (ByVal arg
    As Object)
    If (Not (Me.getAtributosCadastradosCompletedEvent) Is Nothing)
        Then
            Dim invokeArgs As System.Web.Services.Protocols.
                InvokeCompletedEventArgs = CType(arg, System.Web.Services.
                    Protocols.InvokeCompletedEventArgs)
115 RaiseEvent getAtributosCadastradosCompleted (Me, New
                getAtributosCadastradosCompletedEventArgs (invokeArgs.
                    Results, invokeArgs.Error, invokeArgs.Cancelled,
                    invokeArgs.UserState))
        End If
End Sub

'''<remarks/>
120 <System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
    RequestNamespace:="http://config.br.ufsc.inf/",
    ResponseNamespace:="http://config.br.ufsc.inf/", Use:=System.Web.
        Services.Description.SoapBindingUse.Literal, ParameterStyle:=
        System.Web.Services.Protocols.SoapParameterStyle.Wrapped)> _
Public Sub adicionarCertificadoAutoridade(<System.Xml.Serialization.
    XmlElementAttribute (Form:=System.Xml.Schema.XmlSchemaForm.
        Unqualified, DataType:="base64Binary", IsNullable:=true)> ByVal
        pkcs12ACA () As Byte, <System.Xml.Serialization.
        XmlElementAttribute (Form:=System.Xml.Schema.XmlSchemaForm.
        Unqualified)> ByVal senha As String)
    Me.Invoke ("adicionarCertificadoAutoridade", New Object () {
        pkcs12ACA, senha})
End Sub

```

```

125     '''<remarks/>
Public Overloads Sub adicionarCertificadoAutoridadeAsync (ByVal
        pkcs12ACA () As Byte, ByVal senha As String)
        Me.adicionarCertificadoAutoridadeAsync (pkcs12ACA, senha,
            Nothing)
End Sub

130     '''<remarks/>
Public Overloads Sub adicionarCertificadoAutoridadeAsync (ByVal
        pkcs12ACA () As Byte, ByVal senha As String, ByVal userState As
        Object)
        If (Me.adicionarCertificadoAutoridadeOperationCompleted Is
            Nothing) Then
            Me.adicionarCertificadoAutoridadeOperationCompleted =
                AddressOf Me.
                OnadicionarCertificadoAutoridadeOperationCompleted
        End If
135     Me.InvokeAsync ("adicionarCertificadoAutoridade", New Object () {
        pkcs12ACA, senha}, Me.
        adicionarCertificadoAutoridadeOperationCompleted, userState)
End Sub

Private Sub OnadicionarCertificadoAutoridadeOperationCompleted (
        ByVal arg As Object)
        If (Not (Me.adicionarCertificadoAutoridadeCompletedEvent) Is
            Nothing) Then
140         Dim invokeArgs As System.Web.Services.Protocols.
            InvokeCompletedEventArgs = CType(arg, System.Web.Services.
            . Protocols.InvokeCompletedEventArgs)
            RaiseEvent adicionarCertificadoAutoridadeCompleted (Me, New
                System.ComponentModel.AsyncCompletedEventArgs (invokeArgs
                . Error, invokeArgs.Cancelled, invokeArgs.UserState))
        End If
End Sub

145     '''<remarks/>
    <System.Web.Services.Protocols.SoapDocumentMethodAttribute ("",
        RequestNamespace:="http://config.br.ufsc.inf/",
        ResponseNamespace:="http://config.br.ufsc.inf/", Use:=System.Web.
        . Services.Description.SoapBindingUse.Literal, ParameterStyle:=
        System.Web.Services.Protocols.SoapParameterStyle.Wrapped)> _
Public Sub adicionarAtributo (<System.Xml.Serialization.
        XmlElementAttribute (Form:=System.Xml.Schema.XmlSchemaForm.

```

```

        Unqualified)> ByVal atributo As atributoCadastro)
        Me.Invoke("adicionarAtributo", New Object() {atributo})
End Sub
150
    '''<remarks/>
Public Overloads Sub adicionarAtributoAsync(ByVal atributo As
        atributoCadastro)
        Me.adicionarAtributoAsync(atributo, Nothing)
End Sub
155
    '''<remarks/>
Public Overloads Sub adicionarAtributoAsync(ByVal atributo As
        atributoCadastro, ByVal userState As Object)
        If (Me.adicionarAtributoOperationCompleted Is Nothing) Then
            Me.adicionarAtributoOperationCompleted = AddressOf Me.
                OnadicionarAtributoOperationCompleted
160        End If
        Me.InvokeAsync("adicionarAtributo", New Object() {atributo}, Me
            .adicionarAtributoOperationCompleted, userState)
End Sub

Private Sub OnadicionarAtributoOperationCompleted(ByVal arg As
        Object)
165        If (Not (Me.adicionarAtributoCompletedEvent) Is Nothing) Then
            Dim invokeArgs As System.Web.Services.Protocols.
                InvokeCompletedEventArgs = CType(arg, System.Web.Services
                    . Protocols.InvokeCompletedEventArgs)
            RaiseEvent adicionarAtributoCompleted(Me, New System.
                ComponentModel.AsyncCompletedEventArgs(invokeArgs.Error,
                    invokeArgs.Cancelled, invokeArgs.UserState))
        End If
End Sub
170
    '''<remarks/>
    <System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
        RequestNamespace:="http://config.br.ufsc.inf/",
        ResponseNamespace:="http://config.br.ufsc.inf/", Use:=System.Web
            . Services.Description.SoapBindingUse.Literal, ParameterStyle:=
                System.Web.Services.Protocols.SoapParameterStyle.Wrapped)> _
Public Sub configurarLCR(<System.Xml.Serialization.
        XmlElementAttribute(Form:=System.Xml.Schema.XmlSchemaForm.
            Unqualified)> ByVal lcrconfig As lcrConfig)
        Me.Invoke("configurarLCR", New Object() {lcrconfig})

```

```

175      End Sub

      '''<remarks/>
      Public Overloads Sub configurarLCRAsync (ByVal lcrconfig As
          lcrConfig)
          Me.configurarLCRAsync (lcrconfig , Nothing)
180      End Sub

      '''<remarks/>
      Public Overloads Sub configurarLCRAsync (ByVal lcrconfig As
          lcrConfig , ByVal userState As Object)
          If (Me.configurarLCROperationCompleted Is Nothing) Then
185              Me.configurarLCROperationCompleted = AddressOf Me.
                  OnconfigurarLCROperationCompleted
          End If
          Me.InvokeAsync ("configurarLCR", New Object () {lcrconfig}, Me.
              configurarLCROperationCompleted , userState)
      End Sub

190      Private Sub OnconfigurarLCROperationCompleted (ByVal arg As Object)
          If (Not (Me.configurarLCRCompletedEvent) Is Nothing) Then
              Dim invokeArgs As System.Web.Services.Protocols.
                  InvokeCompletedEventArgs = CType (arg , System.Web.Services
                  . Protocols.InvokeCompletedEventArgs)
              RaiseEvent configurarLCRCompleted (Me, New System.
                  ComponentModel.AsyncCompletedEventArgs (invokeArgs.Error ,
                  invokeArgs.Cancelled , invokeArgs.UserState))
          End If
195      End Sub

      '''<remarks/>
      Public Shadows Sub CancelAsync (ByVal userState As Object)
          MyBase.CancelAsync (userState)
200      End Sub

      Private Function IsLocalFileSystemWebService (ByVal url As String)
          As Boolean
          If ((url Is Nothing) _
              OrElse (url Is String.Empty)) Then
205              Return false
          End If
          Dim wsUri As System.Uri = New System.Uri (url)
          If ((wsUri.Port >= 1024) _

```



```

                AndAlso (String.Compare(wsUri.Host, "localhost",
                System.StringComparison.OrdinalIgnoreCase) = 0)
                Then
210             Return true
            End If
            Return false
        End Function
    End Class

215    '''<remarks/>
    <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Xml", "
        2.0.50727.3053"), _
        System.SerializableAttribute(), _
        System.Diagnostics.DebuggerStepThroughAttribute(), _
220    System.ComponentModel.DesignerCategoryAttribute("code"), _
        System.Xml.Serialization.XmlTypeAttribute([Namespace]:="http://config.
            br.ufsc.inf/")> _
    Partial Public Class atributo

        Private descricaoField As String

225        Private oIDField As String

        Private tipoField As tipoAtributo

230        Private tipoFieldSpecified As Boolean

        Private valorPadraoField() As Byte

        '''<remarks/>
235    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
        Public Property descricao() As String
            Get
                Return Me.descricaoField
            End Get
240            Set
                Me.descricaoField = value
            End Set
        End Property

245    '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.

```

```
Schema.XmlSchemaForm.Unqualified)> _
Public Property OID() As String
    Get
        Return Me.oIDField
250    End Get
    Set
        Me.oIDField = value
    End Set
End Property

255    '''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified)> _
Public Property tipo() As tipoAtributo
    Get
260        Return Me.tipoField
    End Get
    Set
        Me.tipoField = value
    End Set
265 End Property

    '''<remarks/>
<System.Xml.Serialization.XmlIgnoreAttribute()> _
Public Property tipoSpecified() As Boolean
270    Get
        Return Me.tipoFieldSpecified
    End Get
    Set
        Me.tipoFieldSpecified = value
275    End Set
End Property

    '''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified, DataType:="base64Binary")> _
280 Public Property valorPadrao() As Byte()
    Get
        Return Me.valorPadraoField
    End Get
    Set
285        Me.valorPadraoField = value
    End Set
```

```

    End Property
End Class

290    '''<remarks/>
    <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Xml", "
        2.0.50727.3053"), _
        System.SerializableAttribute(), _
        System.Xml.Serialization.XmlTypeAttribute([Namespace]:="http://config.
            br.ufsc.inf/")> _
Public Enum tipoAtributo

295    '''<remarks/>
    NUMERO

    '''<remarks/>
300    [STRING]

    '''<remarks/>
    BYTEARRAY
End Enum

305    '''<remarks/>
    <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Xml", "
        2.0.50727.3053"), _
        System.SerializableAttribute(), _
        System.Diagnostics.DebuggerStepThroughAttribute(), _
        System.ComponentModel.DesignerCategoryAttribute("code"), _
310    System.Xml.Serialization.XmlTypeAttribute([Namespace]:="http://config.
        br.ufsc.inf/")> _
Partial Public Class lcrConfig

    Private algAssinaturaField As String

315    Private diretorioPublicacaoField As String

    Private tempoPublicacaoField As Integer

320    Private urlPublicacaoField As String

    '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property algAssinatura() As String

```

```
325         Get
            Return Me.algAssinaturaField
        End Get
        Set
            Me.algAssinaturaField = value
330        End Set
    End Property

    '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
335    Public Property diretorioPublicacao() As String
        Get
            Return Me.diretorioPublicacaoField
        End Get
        Set
340            Me.diretorioPublicacaoField = value
        End Set
    End Property

    '''<remarks/>
345    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property tempoPublicacao() As Integer
        Get
            Return Me.tempoPublicacaoField
        End Get
350        Set
            Me.tempoPublicacaoField = value
        End Set
    End Property

    '''<remarks/>
355    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property urlPublicacao() As String
        Get
            Return Me.urlPublicacaoField
360        End Get
        Set
            Me.urlPublicacaoField = value
        End Set
    End Property
```

```
365     End Class

    '''<remarks/>
    <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Xml", "
        2.0.50727.3053"), _
    System.SerializableAttribute(), _
370     System.Diagnostics.DebuggerStepThroughAttribute(), _
    System.ComponentModel.DesignerCategoryAttribute("code"), _
    System.Xml.Serialization.XmlTypeAttribute([Namespace]="http://config.
        br.ufsc.inf/")> _
    Partial Public Class atributoCadastro

375     Private descricaoField As String

    Private oIDField As String

    Private tipoField As tipoAtributo

380     Private tipoFieldSpecified As Boolean

    Private valorPadraoField() As Byte

385     '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property descricao() As String
        Get
            Return Me.descricaoField
390     End Get
        Set
            Me.descricaoField = value
        End Set
    End Property

395     '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property OID() As String
        Get
400     Return Me.oIDField
        End Get
        Set
            Me.oIDField = value
```

```

    End Set
405 End Property

    '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property tipo() As tipoAtributo
410     Get
        Return Me.tipoField
    End Get
    Set
        Me.tipoField = value
415     End Set
    End Property

    '''<remarks/>
    <System.Xml.Serialization.XmlIgnoreAttribute()> _
420 Public Property tipoSpecified() As Boolean
    Get
        Return Me.tipoFieldSpecified
    End Get
    Set
425     Me.tipoFieldSpecified = value
    End Set
    End Property

    '''<remarks/>
430 <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified, DataType:="base64Binary")> _
    Public Property valorPadrao() As Byte()
    Get
        Return Me.valorPadraoField
    End Get
435     Set
        Me.valorPadraoField = value
    End Set
    End Property
End Class
440

    '''<remarks/>
    <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
        "2.0.50727.3053")> _
    Public Delegate Sub getAtributosCadastradosCompletedEventHandler(ByVal
```

```

        sender As Object , ByVal e As
        getAtributosCadastradosCompletedEventArgs)

445    '''<remarks/>
    <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
        "2.0.50727.3053"), _
        System.Diagnostics.DebuggerStepThroughAttribute(), _
        System.ComponentModel.DesignerCategoryAttribute("code")> _
    Partial Public Class getAtributosCadastradosCompletedEventArgs
450        Inherits System.ComponentModel.AsyncCompletedEventArgs

        Private results() As Object

        Friend Sub New(ByVal results() As Object, ByVal exception As System
            .Exception, ByVal cancelled As Boolean, ByVal userState As
            Object)
455            MyBase.New(exception, cancelled, userState)
            Me.results = results

        End Sub

        '''<remarks/>
460        Public ReadOnly Property Result() As atributo()
            Get
                Me.RaiseExceptionIfNecessary
                Return CType(Me.results(0), atributo())
            End Get
        End Property
465    End Class

    '''<remarks/>
    <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
        "2.0.50727.3053")> _
470    Public Delegate Sub adicionarCertificadoAutoridadeCompletedEventHandler
        (ByVal sender As Object, ByVal e As System.ComponentModel.
        AsyncCompletedEventArgs)

        '''<remarks/>
    <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
        "2.0.50727.3053")> _
    Public Delegate Sub adicionarAtributoCompletedEventHandler(ByVal sender
        As Object, ByVal e As System.ComponentModel.AsyncCompletedEventArgs
        )
475

```

```

'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053")>
    Public Delegate Sub configurarLCRCompletedEventHandler(ByVal sender As
        Object, ByVal e As System.ComponentModel.AsyncCompletedEventArgs)
End Namespace

```

Listagem A.80: Classe EmissaoWSCliente

```

ï»¿

```

```

' <auto-generated>
'     This code was generated by a tool.
'     Runtime Version:2.0.50727.3053
5 '
'     Changes to this file may cause incorrect behavior and will be lost if
'     the code is regenerated.
' </auto-generated>
'

```

```

10
Option Strict Off
Option Explicit On

Imports System
15 Imports System.ComponentModel
Imports System.Diagnostics
Imports System.Web.Services
Imports System.Web.Services.Protocols
Imports System.Xml.Serialization
20
,
' This source code was auto-generated by Microsoft.VSDesigner, Version
    2.0.50727.3053.
,
Namespace emissao
25
'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053"),
    System.Diagnostics.DebuggerStepThroughAttribute(),
    System.ComponentModel.DesignerCategoryAttribute("code"),

```



```

30      System.Web.Services.WebServiceBindingAttribute(Name:="
      EmissaoCertificadoWSPortBinding", [Namespace]:="http://emissao.br.
      ufsc.inf/")> _
Partial Public Class EmissaoCertificadoWSService
      Inherits System.Web.Services.Protocols.SoapHttpClientProtocol

      Private gerarCertificadoOperationCompleted As System.Threading.
      SendOrPostCallback

35      Private getTodosCertificadosAtributosEmitidosOperationCompleted As
      System.Threading.SendOrPostCallback

      Private getCertificadoAtributoEmitidosPorIdOperationCompleted As
      System.Threading.SendOrPostCallback

40      Private useDefaultCredentialsSetExplicitly As Boolean

      '''<remarks/>
Public Sub New()
      MyBase.New
45      Me.Url = Global.WindowsApplication1.My.MySettings.Default.
      WindowsApplication1_emissao_EmissaoCertificadoWSService
      If (Me.IsLocalFileSystemWebService(Me.Url) = true) Then
      Me.UseDefaultCredentials = true
      Me.useDefaultCredentialsSetExplicitly = false
      Else
50      Me.useDefaultCredentialsSetExplicitly = true
      End If
End Sub

      Public Shadows Property Url() As String
55      Get
      Return MyBase.Url
      End Get
      Set
      If (((Me.IsLocalFileSystemWebService(MyBase.Url) = true) _
60      AndAlso (Me.useDefaultCredentialsSetExplicitly
      = false)) _
      AndAlso (Me.IsLocalFileSystemWebService(value)
      = false)) Then
      MyBase.UseDefaultCredentials = false
      End If
      MyBase.Url = value

```

```

65         End Set
End Property

Public Shadows Property UseDefaultCredentials () As Boolean
    Get
70         Return MyBase.UseDefaultCredentials
    End Get
    Set
        MyBase.UseDefaultCredentials = value
        Me.useDefaultCredentialsSetExplicitly = true
75    End Set
End Property

'''<remarks/>
Public Event gerarCertificadoCompleted As
    gerarCertificadoCompletedEventHandler
80

'''<remarks/>
Public Event getTodosCertificadosAtributosEmitidosCompleted As
    getTodosCertificadosAtributosEmitidosCompletedEventHandler

'''<remarks/>
85 Public Event getCertificadoAtributoEmitidosPorIdCompleted As
    getCertificadoAtributoEmitidosPorIdCompletedEventHandler

'''<remarks/>
<System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
    RequestNamespace:="http://emissao.br.ufsc.inf/",
    ResponseNamespace:="http://emissao.br.ufsc.inf/", Use:=System.
    Web.Services.Description.SoapBindingUse.Literal, ParameterStyle
    :=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)> _
Public Function gerarCertificado(<System.Xml.Serialization.
    XmlElementAttribute(Form:=System.Xml.Schema.XmlSchemaForm.
    Unqualified)> ByVal id_req As Long) As <System.Xml.Serialization.
    XmlElementAttribute("return", Form:=System.Xml.Schema.
    XmlSchemaForm.Unqualified, DataType:"base64Binary", IsNullable
    :=true)> Byte()
90    Dim results () As Object = Me.Invoke("gerarCertificado", New
        Object() {id_req})
    Return CType(results(0),Byte())
End Function

'''<remarks/>

```

```

95      Public Overloads Sub gerarCertificadoAsync (ByVal id_req As Long)
          Me.gerarCertificadoAsync (id_req, Nothing)
End Sub

    '''<remarks/>
100  Public Overloads Sub gerarCertificadoAsync (ByVal id_req As Long,
          ByVal userState As Object)
          If (Me.gerarCertificadoOperationCompleted Is Nothing) Then
              Me.gerarCertificadoOperationCompleted = AddressOf Me.
                  OngerarCertificadoOperationCompleted
          End If
          Me.InvokeAsync ("gerarCertificado", New Object () {id_req}, Me.
              gerarCertificadoOperationCompleted, userState)
105  End Sub

Private Sub OngerarCertificadoOperationCompleted (ByVal arg As
          Object)
          If (Not (Me.gerarCertificadoCompletedEvent) Is Nothing) Then
              Dim invokeArgs As System.Web.Services.Protocols.
                  InvokeCompletedEventArgs = CType(arg, System.Web.Services
                  .Protocols.InvokeCompletedEventArgs)
110          RaiseEvent gerarCertificadoCompleted (Me, New
                  gerarCertificadoCompletedEventArgs (invokeArgs.Results,
                  invokeArgs.Error, invokeArgs.Cancelled, invokeArgs.
                  UserState))
          End If
End Sub

    '''<remarks/>
115  <System.Web.Services.Protocols.SoapDocumentMethodAttribute ("",
          RequestNamespace:="http://emissao.br.ufsc.inf/",
          ResponseNamespace:="http://emissao.br.ufsc.inf/", Use:=System.
          Web.Services.Description.SoapBindingUse.Literal, ParameterStyle
          :=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)> _
Public Function getTodosCertificadosAtributosEmitidos () As <System.
          Xml.Serialization.XmlElementAttribute ("return", Form:=System.Xml
          .Schema.XmlSchemaForm.Unqualified)> certificadoAtributo ()
          Dim results () As Object = Me.Invoke ("
              getTodosCertificadosAtributosEmitidos", New Object (-1) {})
          Return CType (results (0), certificadoAtributo ())
End Function

120  '''<remarks/>

```

```

Public Overloads Sub getTodosCertificadosAtributosEmitidosAsync ()
    Me.getTodosCertificadosAtributosEmitidosAsync (Nothing)
End Sub

125
'''<remarks/>
Public Overloads Sub getTodosCertificadosAtributosEmitidosAsync (
    ByVal userState As Object)
    If (Me.getTodosCertificadosAtributosEmitidosOperationCompleted
        Is Nothing) Then
        Me.getTodosCertificadosAtributosEmitidosOperationCompleted
            = AddressOf Me.
            OngetTodosCertificadosAtributosEmitidosOperationCompleted

130
    End If
    Me.InvokeAsync ("getTodosCertificadosAtributosEmitidos", New
        Object(-1) {}, Me.
        getTodosCertificadosAtributosEmitidosOperationCompleted,
        userState)
End Sub

Private Sub
    OngetTodosCertificadosAtributosEmitidosOperationCompleted (ByVal
        arg As Object)
135
    If (Not (Me.getTodosCertificadosAtributosEmitidosCompletedEvent
        ) Is Nothing) Then
        Dim invokeArgs As System.Web.Services.Protocols.
            InvokeCompletedEventArgs = CType(arg, System.Web.Services
                . Protocols.InvokeCompletedEventArgs)
        RaiseEvent getTodosCertificadosAtributosEmitidosCompleted (
            Me, New
            getTodosCertificadosAtributosEmitidosCompletedEventArgs (
                invokeArgs.Results, invokeArgs.Error, invokeArgs.
                Cancelled, invokeArgs.UserState))

    End If
End Sub

140
'''<remarks/>
<System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
    RequestNamespace:="http://emissao.br.ufsc.inf/",
    ResponseNamespace:="http://emissao.br.ufsc.inf/", Use:=System.
    Web.Services.Description.SoapBindingUse.Literal, ParameterStyle
    :=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)> _
Public Function getCertificadoAtributoEmitidosPorId (<System.Xml.

```

```

        Serialization.XmlElementAttribute(Form:=System.Xml.Schema.
        XmlSchemaForm.Unqualified)> ByVal arg0 As Long) As <System.Xml.
        Serialization.XmlElementAttribute("return", Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> certificadoAtributo
        Dim results() As Object = Me.Invoke("
            getCertificadoAtributoEmitidosPorId", New Object() {arg0})
145     Return CType(results(0), certificadoAtributo)
End Function

'''<remarks/>
Public Overloads Sub getCertificadoAtributoEmitidosPorIdAsync (ByVal
    arg0 As Long)
150     Me.getCertificadoAtributoEmitidosPorIdAsync(arg0, Nothing)
End Sub

'''<remarks/>
Public Overloads Sub getCertificadoAtributoEmitidosPorIdAsync (ByVal
    arg0 As Long, ByVal userState As Object)
155     If (Me.getCertificadoAtributoEmitidosPorIdOperationCompleted Is
        Nothing) Then
        Me.getCertificadoAtributoEmitidosPorIdOperationCompleted =
            AddressOf Me.
            OngetCertificadoAtributoEmitidosPorIdOperationCompleted
        End If
        Me.InvokeAsync("getCertificadoAtributoEmitidosPorId", New
            Object() {arg0}, Me.
            getCertificadoAtributoEmitidosPorIdOperationCompleted,
            userState)
End Sub
160

Private Sub OngetCertificadoAtributoEmitidosPorIdOperationCompleted
    (ByVal arg As Object)
    If (Not (Me.getCertificadoAtributoEmitidosPorIdCompletedEvent)
        Is Nothing) Then
        Dim invokeArgs As System.Web.Services.Protocols.
            InvokeCompletedEventArgs = CType(arg, System.Web.Services.
            Protocols.InvokeCompletedEventArgs)
        RaiseEvent getCertificadoAtributoEmitidosPorIdCompleted (Me,
            New
            getCertificadoAtributoEmitidosPorIdCompletedEventArgs (
            invokeArgs.Results, invokeArgs.Error, invokeArgs.
            Cancelled, invokeArgs.UserState))
165     End If

```

```

End Sub

'''<remarks/>
Public Shadows Sub CancelAsync(ByVal userState As Object)
170     MyBase.CancelAsync(userState)
End Sub

Private Function IsLocalFileSystemWebService(ByVal url As String)
    As Boolean
    If ((url Is Nothing) _
175         OrElse (url Is String.Empty)) Then
        Return false
    End If
    Dim wsUri As System.Uri = New System.Uri(url)
    If ((wsUri.Port >= 1024) _
180         AndAlso (String.Compare(wsUri.Host, "localhost",
            System.StringComparison.OrdinalIgnoreCase) = 0))
        Then
            Return true
        End If
        Return false
    End Function
185 End Class

'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Xml", "
    2.0.50727.3053"), _
System.SerializableAttribute(), _
190 System.Diagnostics.DebuggerStepThroughAttribute(), _
System.ComponentModel.DesignerCategoryAttribute("code"), _
System.Xml.Serialization.XmlTypeAttribute([Namespace]:="http://emissao
    .br.ufsc.inf/")> _
Partial Public Class certificadoAtributo

195     Private certificadoPKIField As certificadoPKI

    Private dataEmissaoField As Date

    Private dataEmissaoFieldSpecified As Boolean

200     Private dataValidadeField As Date

    Private dataValidadeFieldSpecified As Boolean

```

```
205     Private hashCertificadoField As String

     Private numeroSerialField As Long

     Private requerenteField As String
210

     Private revogadoField As Boolean

     Private valorCertificadoField () As Byte

215     '''<remarks/>
     <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
         Schema.XmlSchemaForm.Unqualified)> _
     Public Property certificadoPKI () As certificadoPKI
         Get
             Return Me.certificadoPKIField
220         End Get
         Set
             Me.certificadoPKIField = value
         End Set
     End Property

225     '''<remarks/>
     <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
         Schema.XmlSchemaForm.Unqualified)> _
     Public Property dataEmissao () As Date
         Get
230             Return Me.dataEmissaoField
         End Get
         Set
             Me.dataEmissaoField = value
         End Set
     End Property

235     '''<remarks/>
     <System.Xml.Serialization.XmlIgnoreAttribute ()> _
     Public Property dataEmissaoSpecified () As Boolean
240         Get
             Return Me.dataEmissaoFieldSpecified
         End Get
         Set
             Me.dataEmissaoFieldSpecified = value
```

```
245         End Set
End Property

'''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
Schema.XmlSchemaForm.Unqualified)> _
250 Public Property dataValidade() As Date
    Get
        Return Me.dataValidadeField
    End Get
    Set
255         Me.dataValidadeField = value
    End Set
End Property

'''<remarks/>
260 <System.Xml.Serialization.XmlIgnoreAttribute()> _
Public Property dataValidadeSpecified() As Boolean
    Get
        Return Me.dataValidadeFieldSpecified
    End Get
265     Set
        Me.dataValidadeFieldSpecified = value
    End Set
End Property

'''<remarks/>
270 <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
Schema.XmlSchemaForm.Unqualified)> _
Public Property hashCertificado() As String
    Get
        Return Me.hashCertificadoField
275     End Get
    Set
        Me.hashCertificadoField = value
    End Set
End Property

280 '''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
Schema.XmlSchemaForm.Unqualified)> _
Public Property numeroSerial() As Long
    Get
```



```
285         Return Me.numeroSerialField
        End Get
        Set
            Me.numeroSerialField = value
        End Set
290 End Property

'''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified)> _
Public Property requerente() As String
295     Get
        Return Me.requerenteField
    End Get
    Set
        Me.requerenteField = value
    End Set
300 End Property

'''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified)> _
305 Public Property revogado() As Boolean
    Get
        Return Me.revogadoField
    End Get
    Set
310         Me.revogadoField = value
    End Set
End Property

'''<remarks/>
315 <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified, DataType:="base64Binary")> _
Public Property valorCertificado() As Byte()
    Get
        Return Me.valorCertificadoField
    End Get
320     Set
        Me.valorCertificadoField = value
    End Set
End Property
End Class
```

325

```

'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Xml", "
    2.0.50727.3053"), _
    System.SerializableAttribute(), _
    System.Diagnostics.DebuggerStepThroughAttribute(), _
330    System.ComponentModel.DesignerCategoryAttribute("code"), _
    System.Xml.Serialization.XmlTypeAttribute([Namespace]:="http://emissao
        .br.ufsc.inf/")> _
Partial Public Class certificadoPKI

```

335

```

    Private emissaoField As Date

```

```

    Private emissaoFieldSpecified As Boolean

```

```

    Private emissorField As String

```

340

```

    Private finalValidadeField As Date

```

```

    Private finalValidadeFieldSpecified As Boolean

```

```

    Private hashField As String

```

345

```

    Private requerenteField As String

```

```

    Private valorField() As Byte

```

350

```

'''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified)> _
    Public Property emissao() As Date

```

```

    Get

```

```

        Return Me.emissaoField

```

355

```

    End Get

```

```

    Set

```

```

        Me.emissaoField = value

```

```

    End Set

```

```

End Property

```

360

```

'''<remarks/>
<System.Xml.Serialization.XmlIgnoreAttribute()> _
    Public Property emissaoSpecified() As Boolean
    Get

```

```
365         Return Me.emissaoFieldSpecified
        End Get
        Set
            Me.emissaoFieldSpecified = value
        End Set
370 End Property

'''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified)> _
Public Property emissor() As String
375     Get
        Return Me.emissorField
    End Get
    Set
        Me.emissorField = value
    End Set
380 End Property

'''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified)> _
385 Public Property finalValidade() As Date
    Get
        Return Me.finalValidadeField
    End Get
    Set
390         Me.finalValidadeField = value
    End Set
End Property

'''<remarks/>
395 <System.Xml.Serialization.XmlIgnoreAttribute()> _
Public Property finalValidadeSpecified() As Boolean
    Get
        Return Me.finalValidadeFieldSpecified
    End Get
400     Set
        Me.finalValidadeFieldSpecified = value
    End Set
End Property

405 '''<remarks/>
```

```
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified)> _
Public Property hash() As String
    Get
        Return Me.hashField
410    End Get
    Set
        Me.hashField = value
    End Set
End Property

'''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified)> _
Public Property requerente() As String
    Get
420    Return Me.requerenteField
    End Get
    Set
        Me.requerenteField = value
    End Set
425 End Property

'''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified, DataType:="base64Binary")> _
Public Property valor() As Byte()
430    Get
        Return Me.valorField
    End Get
    Set
        Me.valorField = value
435    End Set
End Property
End Class

'''<remarks/>
440 <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053")> _
Public Delegate Sub gerarCertificadoCompletedEventHandler(ByVal sender
    As Object, ByVal e As gerarCertificadoCompletedEventArgs)

'''<remarks/>
```

```

<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053"), _
445   System.Diagnostics.DebuggerStepThroughAttribute(), _
    System.ComponentModel.DesignerCategoryAttribute("code")> _
Partial Public Class gerarCertificadoCompletedEventArgs
    Inherits System.ComponentModel.AsyncCompletedEventArgs

450   Private results() As Object

    Friend Sub New(ByVal results() As Object, ByVal exception As System
        .Exception, ByVal cancelled As Boolean, ByVal userState As
        Object)
        MyBase.New(exception, cancelled, userState)
        Me.results = results
455   End Sub

    '''<remarks/>
    Public ReadOnly Property Result() As Byte()
        Get
460            Me.RaiseExceptionIfNecessary
            Return CType(Me.results(0), Byte())
        End Get
    End Property
End Class

465
    '''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053")> _
    Public Delegate Sub
        getTodosCertificadosAtributosEmitidosCompletedEventHandler(ByVal
            sender As Object, ByVal e As
            getTodosCertificadosAtributosEmitidosCompletedEventArgs)

470
    '''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053"), _
    System.Diagnostics.DebuggerStepThroughAttribute(), _
    System.ComponentModel.DesignerCategoryAttribute("code")> _
Partial Public Class
    getTodosCertificadosAtributosEmitidosCompletedEventArgs
475    Inherits System.ComponentModel.AsyncCompletedEventArgs

    Private results() As Object

```

```

Friend Sub New(ByVal results () As Object, ByVal exception As System
    .Exception, ByVal cancelled As Boolean, ByVal userState As
    Object)
480     MyBase.New(exception, cancelled, userState)
        Me.results = results
End Sub

'''<remarks/>
485 Public ReadOnly Property Result () As certificadoAtributo ()
    Get
        Me.RaiseExceptionIfNecessary
        Return CType(Me.results (0), certificadoAtributo ())
    End Get
490 End Property
End Class

'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053")> _
495 Public Delegate Sub
    getCertificadoAtributoEmitidosPorIdCompletedEventHandler (ByVal
    sender As Object, ByVal e As
    getCertificadoAtributoEmitidosPorIdCompletedEventArgs)

'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053"), _
500 System.Diagnostics.DebuggerStepThroughAttribute (), _
System.ComponentModel.DesignerCategoryAttribute("code")> _
Partial Public Class
    getCertificadoAtributoEmitidosPorIdCompletedEventArgs
    Inherits System.ComponentModel.AsyncCompletedEventArgs

    Private results () As Object
505
Friend Sub New(ByVal results () As Object, ByVal exception As System
    .Exception, ByVal cancelled As Boolean, ByVal userState As
    Object)
    MyBase.New(exception, cancelled, userState)
    Me.results = results
End Sub
510

```

```

    '''<remarks/>
    Public ReadOnly Property Result() As certificadoAtributo
        Get
            Me.RaiseExceptionIfNecessary
515         Return CType(Me.results(0), certificadoAtributo)
        End Get
    End Property
End Class
End Namespace

```

Listagem A.81: Classe RevogacaoWSCliente

```

'''
'''<auto-generated>
'''
'''    This code was generated by a tool.
'''    Runtime Version:2.0.50727.3053
5  '''
'''    Changes to this file may cause incorrect behavior and will be lost if
'''    the code is regenerated.
'''</auto-generated>
'''
'''
10
Option Strict Off
Option Explicit On

Imports System
15 Imports System.ComponentModel
Imports System.Diagnostics
Imports System.Web.Services
Imports System.Web.Services.Protocols
Imports System.Xml.Serialization
20
,
'''This source code was auto-generated by Microsoft.VSDesigner, Version
    2.0.50727.3053.
,
Namespace LCR
25
    '''<remarks/>
    <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",

```

```

        "2.0.50727.3053"), _
    System.Diagnostics.DebuggerStepThroughAttribute(), _
    System.ComponentModel.DesignerCategoryAttribute("code"), _
30    System.Web.Services.WebServiceBindingAttribute(Name:=
        RevogarCertificadoWSPortBinding", [Namespace]:= "http://revogacao.br
        .ufsc.inf/")> _
Partial Public Class RevogarCertificadoWSService
    Inherits System.Web.Services.Protocols.SoapHttpClientProtocol

    Private revogarCertificadoOperationCompleted As System.Threading.
        SendOrPostCallback

35    Private emitirListaCertificadoRevogadoOperationCompleted As System.
        Threading.SendOrPostCallback

    Private useDefaultCredentialsSetExplicitly As Boolean

40    '''<remarks/>
    Public Sub New()
        MyBase.New
        Me.Url = Global.WindowsApplication1.My.MySettings.Default.
            WindowsApplication1_LCR_RevogarCertificadoWSService
        If (Me.IsLocalFileSystemWebService(Me.Url) = true) Then
45            Me.UseDefaultCredentials = true
            Me.useDefaultCredentialsSetExplicitly = false
        Else
            Me.useDefaultCredentialsSetExplicitly = true
        End If
50    End Sub

    Public Shadows Property Url() As String
        Get
            Return MyBase.Url
55        End Get
        Set
            If (((Me.IsLocalFileSystemWebService(MyBase.Url) = true) _
                AndAlso (Me.useDefaultCredentialsSetExplicitly
                    = false)) _
                AndAlso (Me.IsLocalFileSystemWebService(value)
                    = false)) Then
60                MyBase.UseDefaultCredentials = false
            End If
            MyBase.Url = value

```



```

        End Set
    End Property

65
    Public Shadows Property UseDefaultCredentials () As Boolean
        Get
            Return MyBase.UseDefaultCredentials
        End Get
70
        Set
            MyBase.UseDefaultCredentials = value
            Me.useDefaultCredentialsSetExplicitly = true
        End Set
    End Property

75
    '''<remarks/>
    Public Event revogarCertificadoCompleted As
        revogarCertificadoCompletedEventHandler

    '''<remarks/>
80
    Public Event emitirListaCertificadoRevogadoCompleted As
        emitirListaCertificadoRevogadoCompletedEventHandler

    '''<remarks/>
    <System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
        RequestNamespace:="http://revogacao.br.ufsc.inf/",
        ResponseNamespace:="http://revogacao.br.ufsc.inf/", Use:=System.
        Web.Services.Description.SoapBindingUse.Literal, ParameterStyle
        :=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)> _
    Public Sub revogarCertificado(<System.Xml.Serialization.
        XmlElementAttribute(Form:=System.Xml.Schema.XmlSchemaForm.
        Unqualified)> ByVal revogacao As revogacaoCertificado)
85
        Me.Invoke("revogarCertificado", New Object() {revogacao})
    End Sub

    '''<remarks/>
    Public Overloads Sub revogarCertificadoAsync(ByVal revogacao As
        revogacaoCertificado)
90
        Me.revogarCertificadoAsync(revogacao, Nothing)
    End Sub

    '''<remarks/>
    Public Overloads Sub revogarCertificadoAsync(ByVal revogacao As
        revogacaoCertificado, ByVal userState As Object)
95
        If (Me.revogarCertificadoOperationCompleted Is Nothing) Then

```

```

        Me.revogarCertificadoOperationCompleted = AddressOf Me.
            OnrevogarCertificadoOperationCompleted
    End If
    Me.InvokeAsync("revogarCertificado", New Object() {revogacao},
        Me.revogarCertificadoOperationCompleted, userState)
End Sub
100
Private Sub OnrevogarCertificadoOperationCompleted(ByVal arg As
    Object)
    If (Not (Me.revogarCertificadoCompletedEvent) Is Nothing) Then
        Dim invokeArgs As System.Web.Services.Protocols.
            InvokeCompletedEventArgs = CType(arg, System.Web.Services
                .Protocols.InvokeCompletedEventArgs)
        RaiseEvent revogarCertificadoCompleted(Me, New System.
            ComponentModel.AsyncCompletedEventArgs(invokeArgs.Error,
                invokeArgs.Cancelled, invokeArgs.UserState))
105    End If
End Sub

'''<remarks/>
<System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
    RequestNamespace:="http://revogacao.br.ufsc.inf/",
    ResponseNamespace:="http://revogacao.br.ufsc.inf/", Use:=System.
    Web.Services.Description.SoapBindingUse.Literal, ParameterStyle
    :=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)> _
110 Public Sub emitirListaCertificadoRevogado()
    Me.Invoke("emitirListaCertificadoRevogado", New Object(-1) {})
End Sub

'''<remarks/>
115 Public Overloads Sub emitirListaCertificadoRevogadoAsync()
    Me.emitirListaCertificadoRevogadoAsync(Nothing)
End Sub

'''<remarks/>
120 Public Overloads Sub emitirListaCertificadoRevogadoAsync(ByVal
    userState As Object)
    If (Me.emitirListaCertificadoRevogadoOperationCompleted Is
        Nothing) Then
        Me.emitirListaCertificadoRevogadoOperationCompleted =
            AddressOf Me.
                OnemitirListaCertificadoRevogadoOperationCompleted
    End If

```

```

        Me.InvokeAsync("emitirListaCertificadoRevogado", New Object(-1)
            {}, Me.emitirListaCertificadoRevogadoOperationCompleted,
            userState)
125     End Sub

    Private Sub OnemitirListaCertificadoRevogadoOperationCompleted(
        ByVal arg As Object)
        If (Not (Me.emitirListaCertificadoRevogadoCompletedEvent) Is
            Nothing) Then
            Dim invokeArgs As System.Web.Services.Protocols.
                InvokeCompletedEventArgs = CType(arg, System.Web.Services
                    . Protocols . InvokeCompletedEventArgs)
130            RaiseEvent emitirListaCertificadoRevogadoCompleted(Me, New
                System.ComponentModel.AsyncCompletedEventArgs(invokeArgs
                    .Error, invokeArgs.Cancelled, invokeArgs.UserState))
            End If
        End Sub

    '''<remarks/>
135     Public Shadows Sub CancelAsync(ByVal userState As Object)
        MyBase.CancelAsync(userState)
    End Sub

    Private Function IsLocalFileSystemWebService(ByVal url As String)
        As Boolean
140        If ((url Is Nothing) _
            OrElse (url Is String.Empty)) Then
            Return false
        End If
        Dim wsUri As System.Uri = New System.Uri(url)
145        If ((wsUri.Port >= 1024) _
            AndAlso (String.Compare(wsUri.Host, "localhost",
                System.StringComparison.OrdinalIgnoreCase) = 0))
            Then
                Return true
            End If
            Return false
150        End Function
    End Class

    '''<remarks/>
    <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Xml", "
        2.0.50727.3053"), _

```

```

155     System.SerializableAttribute(), _
        System.Diagnostics.DebuggerStepThroughAttribute(), _
        System.ComponentModel.DesignerCategoryAttribute("code"), _
        System.Xml.Serialization.XmlTypeAttribute([Namespace]:="http://
            revogacao.br.ufsc.inf/")> _
Partial Public Class revogacaoCertificado
160
    Private motivoField As String

    Private requisicaoField As Long

165     '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property motivo() As String
        Get
            Return Me.motivoField
170     End Get
        Set
            Me.motivoField = value
        End Set
    End Property

175     '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property requisicao() As Long
        Get
180     Return Me.requisicaoField
        End Get
        Set
            Me.requisicaoField = value
        End Set
    End Property
185 End Class

    '''<remarks/>
    <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
        "2.0.50727.3053")> _
190 Public Delegate Sub revogarCertificadoCompletedEventHandler(ByVal
    sender As Object, ByVal e As System.ComponentModel.
    AsyncCompletedEventArgs)

```

```

'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053")> _
    Public Delegate Sub emitirListaCertificadoRevogadoCompletedEventHandler
        (ByVal sender As Object, ByVal e As System.ComponentModel.
            AsyncCompletedEventArgs)
195 End Namespace

```

Listagem A.82: Classe RequisicaoWSCliente

```

ï»¿'
' <auto-generated>
'     This code was generated by a tool.
'     Runtime Version:2.0.50727.3053
5 '
'     Changes to this file may cause incorrect behavior and will be lost if
'     the code is regenerated.
' </auto-generated>
'
'
10
Option Strict Off
Option Explicit On

Imports System
15 Imports System.ComponentModel
Imports System.Diagnostics
Imports System.Web.Services
Imports System.Web.Services.Protocols
Imports System.Xml.Serialization
20
'
' This source code was auto-generated by Microsoft.VSDesigner, Version
'     2.0.50727.3053.
'
Namespace Requisicao
25
'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053"), _
    System.Diagnostics.DebuggerStepThroughAttribute(), _

```

```

System.ComponentModel.DesignerCategoryAttribute("code"), _
30 System.Web.Services.WebServiceBindingAttribute(Name:= "
    RequisicaoCertificadoWSPortBinding", [Namespace]:= "http://
    requisicao.br.ufsc.inf/")> _
Partial Public Class RequisicaoCertificadoWSService
    Inherits System.Web.Services.Protocols.SoapHttpClientProtocol

    Private gerarRequisicaoCertificadoOperationCompleted As System.
        Threading.SendOrPostCallback

35 Private adicionarAtributoRequisicaoOperationCompleted As System.
        Threading.SendOrPostCallback

Private getCertificadoAtributoPorRequisicaoOperationCompleted As
        System.Threading.SendOrPostCallback

40 Private getAtributosCadastradosOperationCompleted As System.
        Threading.SendOrPostCallback

Private getTodasRequisicoesOperationCompleted As System.Threading.
        SendOrPostCallback

Private getRequisicaoPorNumeroIdentificadorOperationCompleted As
        System.Threading.SendOrPostCallback

45 Private useDefaultCredentialsSetExplicitly As Boolean

    '''<remarks/>
Public Sub New()
50     MyBase.New
        Me.Url = Global.WindowsApplication1.My.MySettings.Default.
            WindowsApplication1_Requisicao_RequisicaoCertificadoWSService

        If (Me.IsLocalFileSystemWebService(Me.Url) = true) Then
            Me.UseDefaultCredentials = true
            Me.useDefaultCredentialsSetExplicitly = false
55 Else
            Me.useDefaultCredentialsSetExplicitly = true
        End If
End Sub

60 Public Shadows Property Url() As String
    Get

```

```
        Return MyBase.Url
    End Get
    Set
65         If (((Me.IsLocalFileSystemWebService(MyBase.Url) = true) _
                AndAlso (Me.useDefaultCredentialsSetExplicitly
                        = false)) _
                AndAlso (Me.IsLocalFileSystemWebService(value)
                        = false)) Then
            MyBase.UseDefaultCredentials = false
        End If
70         MyBase.Url = value
    End Set
End Property

Public Shadows Property UseDefaultCredentials() As Boolean
75     Get
        Return MyBase.UseDefaultCredentials
    End Get
    Set
        MyBase.UseDefaultCredentials = value
80         Me.useDefaultCredentialsSetExplicitly = true
    End Set
End Property

'''<remarks/>
85 Public Event gerarRequisicaoCertificadoCompleted As
    gerarRequisicaoCertificadoCompletedEventHandler

'''<remarks/>
Public Event adicionarAtributoRequisicaoCompleted As
    adicionarAtributoRequisicaoCompletedEventHandler

90 '''<remarks/>
Public Event getCertificadoAtributoPorRequisicaoCompleted As
    getCertificadoAtributoPorRequisicaoCompletedEventHandler

'''<remarks/>
Public Event getAtributosCadastradosCompleted As
    getAtributosCadastradosCompletedEventHandler

95 '''<remarks/>
Public Event getTodasRequisicoesCompleted As
    getTodasRequisicoesCompletedEventHandler
```

```

'''<remarks/>
100 Public Event getRequisicaoPorNumeroIdentificadorCompleted As
      getRequisicaoPorNumeroIdentificadorCompletedEventHandler

'''<remarks/>
<System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
      RequestNamespace:="http://requisicao.br.ufsc.inf/",
      ResponseNamespace:="http://requisicao.br.ufsc.inf/", Use:=System
      .Web.Services.Description.SoapBindingUse.Literal, ParameterStyle
      :=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)> _
Public Function gerarRequisicaoCertificado(<System.Xml.
      Serialization.XmlElementAttribute(Form:=System.Xml.Schema.
      XmlSchemaForm.Unqualified, DataType:="base64Binary", IsNullable
      :=true)> ByVal certificado () As Byte) As <System.Xml.
      Serialization.XmlElementAttribute("return", Form:=System.Xml.
      Schema.XmlSchemaForm.Unqualified)> Long
105 Dim results () As Object = Me.Invoke("gerarRequisicaoCertificado
      ", New Object () { certificado })
      Return CType(results(0),Long)
End Function

'''<remarks/>
110 Public Overloads Sub gerarRequisicaoCertificadoAsync (ByVal
      certificado () As Byte)
      Me.gerarRequisicaoCertificadoAsync(certificado, Nothing)
End Sub

'''<remarks/>
115 Public Overloads Sub gerarRequisicaoCertificadoAsync (ByVal
      certificado () As Byte, ByVal userState As Object)
      If (Me.gerarRequisicaoCertificadoOperationCompleted Is Nothing)
      Then
      Me.gerarRequisicaoCertificadoOperationCompleted = AddressOf
      Me.OngerarRequisicaoCertificadoOperationCompleted
      End If
      Me.InvokeAsync("gerarRequisicaoCertificado", New Object () {
      certificado }, Me.
      gerarRequisicaoCertificadoOperationCompleted, userState)
120 End Sub

Private Sub OngerarRequisicaoCertificadoOperationCompleted (ByVal
      arg As Object)

```



```

If (Not (Me.gerarRequisicaoCertificadoCompletedEvent) Is
Nothing) Then
    Dim invokeArgs As System.Web.Services.Protocols.
        InvokeCompletedEventArgs = CType(arg,System.Web.Services
        .Protocols.InvokeCompletedEventArgs)
125    RaiseEvent gerarRequisicaoCertificadoCompleted(Me, New
        gerarRequisicaoCertificadoCompletedEventArgs(invokeArgs.
        Results, invokeArgs.Error, invokeArgs.Cancelled,
        invokeArgs.UserState))
End If
End Sub

'''<remarks/>
130 <System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
    RequestNamespace:="http://requisicao.br.ufsc.inf/",
    ResponseNamespace:="http://requisicao.br.ufsc.inf/", Use:=System
    .Web.Services.Description.SoapBindingUse.Literal, ParameterStyle
    :=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)> _
Public Sub adicionarAtributoRequisicao(<System.Xml.Serialization.
    XmlElementAttribute(Form:=System.Xml.Schema.XmlSchemaForm.
    Unqualified)> ByVal req As Long, <System.Xml.Serialization.
    XmlElementAttribute(Form:=System.Xml.Schema.XmlSchemaForm.
    Unqualified)> ByVal atributo As atributoRequisicaoBL)
    Me.Invoke("adicionarAtributoRequisicao", New Object() {req,
        atributo})
End Sub

135 '''<remarks/>
Public Overloads Sub adicionarAtributoRequisicaoAsync(ByVal req As
    Long, ByVal atributo As atributoRequisicaoBL)
    Me.adicionarAtributoRequisicaoAsync(req, atributo, Nothing)
End Sub

140 '''<remarks/>
Public Overloads Sub adicionarAtributoRequisicaoAsync(ByVal req As
    Long, ByVal atributo As atributoRequisicaoBL, ByVal userState As
    Object)
If (Me.adicionarAtributoRequisicaoOperationCompleted Is Nothing
) Then
    Me.adicionarAtributoRequisicaoOperationCompleted =
        AddressOf Me.
        OnadicionarAtributoRequisicaoOperationCompleted
End If

```

```

145         Me.InvokeAsync("adicionarAtributoRequisicao", New Object() {req
            , atributo}, Me.
            adicionarAtributoRequisicaoOperationCompleted, userState)
End Sub

Private Sub OnadicionarAtributoRequisicaoOperationCompleted (ByVal
    arg As Object)
    If (Not (Me.adicionarAtributoRequisicaoCompletedEvent) Is
        Nothing) Then
150         Dim invokeArgs As System.Web.Services.Protocols.
            InvokeCompletedEventArgs = CType(arg, System.Web.Services
            . Protocols.InvokeCompletedEventArgs)
            RaiseEvent adicionarAtributoRequisicaoCompleted (Me, New
            System.ComponentModel.AsyncCompletedEventArgs (invokeArgs
            . Error, invokeArgs.Cancelled, invokeArgs.UserState))
        End If
End Sub

155     '''<remarks/>
    <System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
        RequestNamespace:="http://requisicao.br.ufsc.inf/",
        ResponseNamespace:="http://requisicao.br.ufsc.inf/", Use:=System
        . Web.Services.Description.SoapBindingUse.Literal, ParameterStyle
        :=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)> _
    Public Function getCertificadoAtributoPorRequisicao(<System.Xml.
        Serialization.XmlElementAttribute(Form:=System.Xml.Schema.
        XmlSchemaForm.Unqualified)> ByVal req As Long) As <System.Xml.
        Serialization.XmlElementAttribute("return", Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> certificadoAtributo
        Dim results() As Object = Me.Invoke("
            getCertificadoAtributoPorRequisicao", New Object() {req})
        Return CType(results(0), certificadoAtributo)
160 End Function

    '''<remarks/>
    Public Overloads Sub getCertificadoAtributoPorRequisicaoAsync (ByVal
        req As Long)
        Me.getCertificadoAtributoPorRequisicaoAsync(req, Nothing)
165 End Sub

    '''<remarks/>
    Public Overloads Sub getCertificadoAtributoPorRequisicaoAsync (ByVal
        req As Long, ByVal userState As Object)

```

```

If (Me.getCertificadoAtributoPorRequisicaoOperationCompleted Is
    Nothing) Then
170     Me.getCertificadoAtributoPorRequisicaoOperationCompleted =
        AddressOf Me.
        OngetCertificadoAtributoPorRequisicaoOperationCompleted
End If
    Me.InvokeAsync("getCertificadoAtributoPorRequisicao", New
        Object() {req}, Me.
        getCertificadoAtributoPorRequisicaoOperationCompleted,
        userState)
End Sub

175 Private Sub OngetCertificadoAtributoPorRequisicaoOperationCompleted
    (ByVal arg As Object)
        If (Not (Me.getCertificadoAtributoPorRequisicaoCompletedEvent)
            Is Nothing) Then
            Dim invokeArgs As System.Web.Services.Protocols.
                InvokeCompletedEventArgs = CType(arg, System.Web.Services
                . Protocols. InvokeCompletedEventArgs)
            RaiseEvent getCertificadoAtributoPorRequisicaoCompleted (Me,
                New
                getCertificadoAtributoPorRequisicaoCompletedEventArgs (
                invokeArgs.Results, invokeArgs.Error, invokeArgs.
                Cancelled, invokeArgs.UserState))
        End If
180 End Sub

    '''<remarks/>
    <System.Web.Services.Protocols. SoapDocumentMethodAttribute("",
        RequestNamespace:="http://requisicao.br.ufsc.inf/",
        ResponseNamespace:="http://requisicao.br.ufsc.inf/", Use:=System
        . Web.Services. Description. SoapBindingUse.Literal, ParameterStyle
        :=System. Web.Services. Protocols. SoapParameterStyle.Wrapped)> _
    Public Function getAtributosCadastrados () As <System.Xml.
        Serialization. XmlElementAttribute("return", Form:=System.Xml.
        Schema. XmlSchemaForm.Unqualified)> atributo ()
185     Dim results () As Object = Me.Invoke("getAtributosCadastrados",
        New Object(-1) {})
        Return CType(results(0), atributo ())
End Function

    '''<remarks/>
190 Public Overloads Sub getAtributosCadastradosAsync ()

```

```

    Me.getAtributosCadastradosAsync(Nothing)
End Sub

    '''<remarks/>
195 Public Overloads Sub getAtributosCadastradosAsync(ByVal userState
    As Object)
    If (Me.getAtributosCadastradosOperationCompleted Is Nothing)
    Then
        Me.getAtributosCadastradosOperationCompleted = AddressOf Me
            .OngetAtributosCadastradosOperationCompleted
    End If
    Me.InvokeAsync("getAtributosCadastrados", New Object(-1) {}, Me
        .getAtributosCadastradosOperationCompleted, userState)
200 End Sub

Private Sub OngetAtributosCadastradosOperationCompleted(ByVal arg
    As Object)
    If (Not (Me.getAtributosCadastradosCompletedEvent) Is Nothing)
    Then
        Dim invokeArgs As System.Web.Services.Protocols.
            InvokeCompletedEventArgs = CType(arg, System.Web.Services
                .Protocols.InvokeCompletedEventArgs)
205        RaiseEvent getAtributosCadastradosCompleted(Me, New
            getAtributosCadastradosCompletedEventArgs(invokeArgs.
                Results, invokeArgs.Error, invokeArgs.Cancelled,
                invokeArgs.UserState))
    End If
End Sub

    '''<remarks/>
210 <System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
    RequestNamespace:="http://requisicao.br.ufsc.inf/",
    ResponseNamespace:="http://requisicao.br.ufsc.inf/", Use:=System
        .Web.Services.Description.SoapBindingUse.Literal, ParameterStyle
            :=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)> _
Public Function getTodasRequisicoes() As <System.Xml.Serialization.
    XmlElementAttribute("return", Form:=System.Xml.Schema.
        XmlSchemaForm.Unqualified)> requisicaoBL()
    Dim results() As Object = Me.Invoke("getTodasRequisicoes", New
        Object(-1) {})
    Return CType(results(0), requisicaoBL())
End Function

```

```

    '''<remarks/>
Public Overloads Sub getTodasRequisicoesAsync ()
    Me.getTodasRequisicoesAsync (Nothing)
End Sub
220
    '''<remarks/>
Public Overloads Sub getTodasRequisicoesAsync (ByVal userState As
    Object)
    If (Me.getTodasRequisicoesOperationCompleted Is Nothing) Then
    Me.getTodasRequisicoesOperationCompleted = AddressOf Me.
    OngetTodasRequisicoesOperationCompleted
225
    End If
    Me.InvokeAsync ("getTodasRequisicoes", New Object(-1) {}, Me.
    getTodasRequisicoesOperationCompleted, userState)
End Sub

Private Sub OngetTodasRequisicoesOperationCompleted (ByVal arg As
    Object)
230
    If (Not (Me.getTodasRequisicoesCompletedEvent) Is Nothing) Then
    Dim invokeArgs As System.Web.Services.Protocols.
    InvokeCompletedEventArgs = CType(arg, System.Web.Services
    . Protocols. InvokeCompletedEventArgs)
    RaiseEvent getTodasRequisicoesCompleted (Me, New
    getTodasRequisicoesCompletedEventArgs (invokeArgs.Results
    , invokeArgs.Error, invokeArgs.Cancelled, invokeArgs.
    UserState))
    End If
End Sub
235
    '''<remarks/>
    <System.Web.Services.Protocols. SoapDocumentMethodAttribute ("",
    RequestNamespace:="http://requisicao.br.ufsc.inf/",
    ResponseNamespace:="http://requisicao.br.ufsc.inf/", Use:=System
    . Web.Services. Description. SoapBindingUse.Literal, ParameterStyle
    :=System. Web.Services. Protocols. SoapParameterStyle.Wrapped)> _
Public Function getRequisicaoPorNumeroIdentificador (<System.Xml.
    Serialization. XmlElementAttribute (Form:=System.Xml. Schema.
    XmlSchemaForm.Unqualified)> ByVal req As Long) As <System.Xml.
    Serialization. XmlElementAttribute ("return", Form:=System.Xml.
    Schema. XmlSchemaForm.Unqualified)> requisicaoBL
    Dim results () As Object = Me.Invoke ("
    getRequisicaoPorNumeroIdentificador", New Object () {req})
240
    Return CType (results (0), requisicaoBL)

```

End Function

'''<remarks/>

Public Overloads Sub getRequisicaoPorNumeroIdentificadorAsync (ByVal
req As Long)

245 Me.getRequisicaoPorNumeroIdentificadorAsync (req , Nothing)

End Sub

'''<remarks/>

Public Overloads Sub getRequisicaoPorNumeroIdentificadorAsync (ByVal
req As Long, ByVal userState As Object)

250 **If** (Me.getRequisicaoPorNumeroIdentificadorOperationCompleted **Is**
Nothing) **Then**

Me.getRequisicaoPorNumeroIdentificadorOperationCompleted =
AddressOf Me.

OngetRequisicaoPorNumeroIdentificadorOperationCompleted

End If

Me.InvokeAsync ("getRequisicaoPorNumeroIdentificador" , New
Object () {req}, Me.

getRequisicaoPorNumeroIdentificadorOperationCompleted ,
userState)

End Sub

255

Private Sub OngetRequisicaoPorNumeroIdentificadorOperationCompleted
(ByVal arg As Object)

If (**Not** (Me.getRequisicaoPorNumeroIdentificadorCompletedEvent)
Is Nothing) **Then**

Dim invokeArgs As **System.Web.Services.Protocols.**

InvokeCompletedEventArgs = CType(arg ,**System.Web.Services**
. Protocols . InvokeCompletedEventArgs)

RaiseEvent getRequisicaoPorNumeroIdentificadorCompleted (Me,
New

getRequisicaoPorNumeroIdentificadorCompletedEventArgs (
invokeArgs.Results , invokeArgs.**Error** , invokeArgs.
Cancelled , invokeArgs.UserState))

260

End If

End Sub

'''<remarks/>

Public Shadows Sub CancelAsync (ByVal userState As Object)

265 MyBase.CancelAsync (userState)

End Sub

```

Private Function IsLocalFileSystemWebService(ByVal url As String)
  As Boolean
  If ((url Is Nothing) _
270         OrElse (url Is String.Empty)) Then
    Return false
  End If
  Dim wsUri As System.Uri = New System.Uri(url)
  If ((wsUri.Port >= 1024) _
275         AndAlso (String.Compare(wsUri.Host, "localhost",
          System.StringComparison.OrdinalIgnoreCase) = 0))
    Then
      Return true
    End If
  Return false
End Function
280 End Class

'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Xml", "
  2.0.50727.3053"), _
  System.SerializableAttribute(), _
285  System.Diagnostics.DebuggerStepThroughAttribute(), _
  System.ComponentModel.DesignerCategoryAttribute("code"), _
  System.Xml.Serialization.XmlTypeAttribute([Namespace]:="http://
    requisicao.br.ufsc.inf/")> _
Partial Public Class atributoRequisicaoBL

290   Private idAtributoField As Long

   Private oidField As String

   Private valorField() As Byte
295

   '''<remarks/>
   <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
     Schema.XmlSchemaForm.Unqualified)> _
   Public Property idAtributo() As Long
     Get
300       Return Me.idAtributoField
     End Get
     Set
       Me.idAtributoField = value
     End Set

```

```

305      End Property

      '''<remarks/>
      <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
          Schema.XmlSchemaForm.Unqualified)> _
      Public Property oid() As String
310          Get
              Return Me.oidField
          End Get
          Set
              Me.oidField = value
315          End Set
      End Property

      '''<remarks/>
      <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
          Schema.XmlSchemaForm.Unqualified, DataType:="base64Binary")> _
320      Public Property valor() As Byte()
          Get
              Return Me.valorField
          End Get
          Set
325              Me.valorField = value
          End Set
      End Property
End Class

330      '''<remarks/>
      <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Xml", "
          2.0.50727.3053"), _
      System.SerializableAttribute(), _
      System.Diagnostics.DebuggerStepThroughAttribute(), _
      System.ComponentModel.DesignerCategoryAttribute("code"), _
335      System.Xml.Serialization.XmlTypeAttribute([Namespace]:="http://
          requisicao.br.ufsc.inf/")> _
      Partial Public Class requisicaoBL

          Private certAtrField As certificadoAtributo

340          Private certificadoPKIField As certificadoPKI

          Private dataRequisicaoField As Date

```



```
345     Private dataRequisicaoFieldSpecified As Boolean

     Private emitidoField As Boolean

     Private idField As Long

350     '''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified)> _
     Public Property certAtr() As certificadoAtributo
         Get
             Return Me.certAtrField
355         End Get
         Set
             Me.certAtrField = value
         End Set
     End Property

360     '''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified)> _
     Public Property certificadoPKI() As certificadoPKI
         Get
365             Return Me.certificadoPKIField
         End Get
         Set
             Me.certificadoPKIField = value
         End Set
     End Property

370     '''<remarks/>
<System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
    Schema.XmlSchemaForm.Unqualified)> _
     Public Property dataRequisicao() As Date
375         Get
             Return Me.dataRequisicaoField
         End Get
         Set
             Me.dataRequisicaoField = value
380         End Set
     End Property

     '''<remarks/>
```

```

385     <System.Xml.Serialization.XmlIgnoreAttribute()> _
Public Property dataRequisicaoSpecified() As Boolean
        Get
            Return Me.dataRequisicaoFieldSpecified
        End Get
        Set
390            Me.dataRequisicaoFieldSpecified = value
        End Set
End Property

    '''<remarks/>
395 <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
Public Property emitido() As Boolean
        Get
            Return Me.emitidoField
        End Get
400        Set
            Me.emitidoField = value
        End Set
End Property

    '''<remarks/>
405 <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
Public Property id() As Long
        Get
            Return Me.idField
410        End Get
        Set
            Me.idField = value
        End Set
End Property
415 End Class

    '''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Xml", "
    2.0.50727.3053"), _
System.SerializableAttribute(), _
420 System.Diagnostics.DebuggerStepThroughAttribute(), _
System.ComponentModel.DesignerCategoryAttribute("code"), _
System.Xml.Serialization.XmlTypeAttribute([Namespace]:= "http://
    requisicao.br.ufsc.inf/")> _

```

```
Partial Public Class certificadoAtributo

425     Private certificadoPKIField As certificadoPKI

        Private dataEmissaoField As Date

        Private dataEmissaoFieldSpecified As Boolean

430     Private dataValidadeField As Date

        Private dataValidadeFieldSpecified As Boolean

435     Private hashCertificadoField As String

        Private numeroSerialField As Long

        Private requerenteField As String

440     Private revogadoField As Boolean

        Private valorCertificadoField () As Byte

445     '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property certificadoPKI() As certificadoPKI
        Get
            Return Me.certificadoPKIField
450     End Get
        Set
            Me.certificadoPKIField = value
        End Set
    End Property

455     '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property dataEmissao() As Date
        Get
460     Return Me.dataEmissaoField
        End Get
        Set
            Me.dataEmissaoField = value
```

```

    End Set
465 End Property

    '''<remarks/>
    <System.Xml.Serialization.XmlIgnoreAttribute(> _
    Public Property dataEmissaoSpecified() As Boolean
470     Get
        Return Me.dataEmissaoFieldSpecified
    End Get
    Set
        Me.dataEmissaoFieldSpecified = value
475 End Set
End Property

    '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
480 Public Property dataValidade() As Date
    Get
        Return Me.dataValidadeField
    End Get
    Set
485     Me.dataValidadeField = value
    End Set
End Property

    '''<remarks/>
490 <System.Xml.Serialization.XmlIgnoreAttribute(> _
    Public Property dataValidadeSpecified() As Boolean
    Get
        Return Me.dataValidadeFieldSpecified
    End Get
495     Set
        Me.dataValidadeFieldSpecified = value
    End Set
End Property

    '''<remarks/>
500 <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property hashCertificado() As String
    Get
        Return Me.hashCertificadoField
```

```
505         End Get
           Set
             Me.hashCertificadoField = value
           End Set
        End Property

510        '''<remarks/>
        <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
          Schema.XmlSchemaForm.Unqualified)> _
        Public Property numeroSerial() As Long
           Get
515             Return Me.numeroSerialField
           End Get
           Set
             Me.numeroSerialField = value
           End Set
        End Property

520        '''<remarks/>
        <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
          Schema.XmlSchemaForm.Unqualified)> _
        Public Property requerente() As String
525           Get
             Return Me.requerenteField
           End Get
           Set
             Me.requerenteField = value
           End Set
        End Property

530        '''<remarks/>
        <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
          Schema.XmlSchemaForm.Unqualified)> _
535        Public Property revogado() As Boolean
           Get
             Return Me.revogadoField
           End Get
           Set
540             Me.revogadoField = value
           End Set
        End Property

        '''<remarks/>
```

```

545     <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified, DataType:="base64Binary")> _
    Public Property valorCertificado() As Byte()
        Get
            Return Me.valorCertificadoField
        End Get
550     Set
        Me.valorCertificadoField = value
    End Set
    End Property
End Class

555     '''<remarks/>
    <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Xml", "
        2.0.50727.3053"), _
    System.SerializableAttribute(), _
    System.Diagnostics.DebuggerStepThroughAttribute(), _
560     System.ComponentModel.DesignerCategoryAttribute("code"), _
    System.Xml.Serialization.XmlTypeAttribute([Namespace]:="http://
        requisicao.br.ufsc.inf/")> _
    Partial Public Class certificadoPKI

        Private emissaoField As Date

565         Private emissaoFieldSpecified As Boolean

        Private emissorField As String

570         Private finalValidadeField As Date

        Private finalValidadeFieldSpecified As Boolean

        Private hashField As String

575         Private requerenteField As String

        Private valorField() As Byte

580     '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property emissao() As Date
        Get

```

```

        Return Me.emissaoField
585     End Get
        Set
            Me.emissaoField = value
        End Set
    End Property

590     '''<remarks/>
    <System.Xml.Serialization.XmlIgnoreAttribute(> _
    Public Property emissaoSpecified() As Boolean
        Get
595             Return Me.emissaoFieldSpecified
        End Get
        Set
            Me.emissaoFieldSpecified = value
        End Set
600    End Property

    '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property emissor() As String
605        Get
            Return Me.emissorField
        End Get
        Set
            Me.emissorField = value
610        End Set
    End Property

    '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
615    Public Property finalValidade() As Date
        Get
            Return Me.finalValidadeField
        End Get
        Set
620            Me.finalValidadeField = value
        End Set
    End Property

    '''<remarks/>
```

```
625     <System.Xml.Serialization.XmlIgnoreAttribute(> _
        Public Property finalValidadeSpecified() As Boolean
            Get
                Return Me.finalValidadeFieldSpecified
            End Get
630     Set
            Me.finalValidadeFieldSpecified = value
        End Set
    End Property

635     '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property hash() As String
        Get
            Return Me.hashField
640     End Get
        Set
            Me.hashField = value
        End Set
    End Property

645     '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property requerente() As String
        Get
650     Return Me.requerenteField
        End Get
        Set
            Me.requerenteField = value
        End Set
655 End Property

        '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified, DataType:="base64Binary")> _
    Public Property valor() As Byte()
660     Get
            Return Me.valorField
        End Get
        Set
            Me.valorField = value
```



```

665         End Set
        End Property
End Class

'''<remarks/>
670 <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Xml", "
        2.0.50727.3053"), _
        System.SerializableAttribute(), _
        System.Diagnostics.DebuggerStepThroughAttribute(), _
        System.ComponentModel.DesignerCategoryAttribute("code"), _
        System.Xml.Serialization.XmlTypeAttribute([Namespace]="http://
        requisicao.br.ufsc.inf/")> _
675 Partial Public Class atributo

        Private descricaoField As String

        Private oIDField As String
680
        Private tipoField As tipoAtributo

        Private tipoFieldSpecified As Boolean

685 Private valorPadraoField() As Byte

        '''<remarks/>
        <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
            Schema.XmlSchemaForm.Unqualified)> _
        Public Property descricao() As String
690         Get
            Return Me.descricaoField
        End Get
        Set
            Me.descricaoField = value
695 End Set
End Property

        '''<remarks/>
        <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
            Schema.XmlSchemaForm.Unqualified)> _
700 Public Property OID() As String
        Get
            Return Me.oIDField
        End Get

```

```

        Set
705         Me.oIDField = value
        End Set
    End Property

    '''<remarks/>
710    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified)> _
    Public Property tipo() As tipoAtributo
        Get
            Return Me.tipoField
        End Get
715        Set
            Me.tipoField = value
        End Set
    End Property

    '''<remarks/>
720    <System.Xml.Serialization.XmlIgnoreAttribute()> _
    Public Property tipoSpecified() As Boolean
        Get
            Return Me.tipoFieldSpecified
725        End Get
        Set
            Me.tipoFieldSpecified = value
        End Set
    End Property

730    '''<remarks/>
    <System.Xml.Serialization.XmlElementAttribute(Form:=System.Xml.
        Schema.XmlSchemaForm.Unqualified, DataType:="base64Binary")> _
    Public Property valorPadrao() As Byte()
        Get
735            Return Me.valorPadraoField
        End Get
        Set
            Me.valorPadraoField = value
        End Set
740    End Property
End Class

    '''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Xml", "
```

```

        2.0.50727.3053"), _
745   System.SerializableAttribute(), _
      System.Xml.Serialization.XmlTypeAttribute([Namespace]:="http://
        requisicao.br.ufsc.inf/")> _
Public Enum tipoAtributo

    '''<remarks/>
750   NUMERO

    '''<remarks/>
      [STRING]

755   '''<remarks/>
      BYTEARRAY
End Enum

    '''<remarks/>
760 <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
      "2.0.50727.3053")> _
Public Delegate Sub gerarRequisicaoCertificadoCompletedEventHandler(
    ByVal sender As Object, ByVal e As
      gerarRequisicaoCertificadoCompletedEventArgs)

    '''<remarks/>
765 <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
      "2.0.50727.3053"), _
      System.Diagnostics.DebuggerStepThroughAttribute(), _
      System.ComponentModel.DesignerCategoryAttribute("code")> _
Partial Public Class gerarRequisicaoCertificadoCompletedEventArgs
    Inherits System.ComponentModel.AsyncCompletedEventArgs

770   Private results() As Object

    Friend Sub New(ByVal results() As Object, ByVal exception As System
      .Exception, ByVal cancelled As Boolean, ByVal userState As
      Object)
        MyBase.New(exception, cancelled, userState)
        Me.results = results
775   End Sub

    '''<remarks/>
Public ReadOnly Property Result() As Long
    Get

```

```

780         Me.RaiseExceptionIfNecessary
           Return CType(Me.results(0), Long)
       End Get
     End Property
End Class

785   '''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053")> _
Public Delegate Sub adicionarAtributoRequisicaoCompletedEventHandler(
    ByVal sender As Object, ByVal e As System.ComponentModel.
        AsyncCompletedEventArgs)

790   '''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053")> _
Public Delegate Sub
    getCertificadoAtributoPorRequisicaoCompletedEventHandler(ByVal
        sender As Object, ByVal e As
            getCertificadoAtributoPorRequisicaoCompletedEventArgs)

800   '''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053"), _
    System.Diagnostics.DebuggerStepThroughAttribute(), _
    System.ComponentModel.DesignerCategoryAttribute("code")> _
Partial Public Class
    getCertificadoAtributoPorRequisicaoCompletedEventArgs
        Inherits System.ComponentModel.AsyncCompletedEventArgs

805   Private results() As Object

    Friend Sub New(ByVal results() As Object, ByVal exception As System
        .Exception, ByVal cancelled As Boolean, ByVal userState As
        Object)
        MyBase.New(exception, cancelled, userState)
810        Me.results = results
    End Sub

    '''<remarks/>
Public ReadOnly Property Result() As certificadoAtributo
    Get
        Me.RaiseExceptionIfNecessary

```

```

        Return CType(Me.results(0), certificadoAtributo)
    End Get
End Property
815 End Class

'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053")> _
Public Delegate Sub getAtributosCadastradosCompletedEventHandler (ByVal
    sender As Object, ByVal e As
        getAtributosCadastradosCompletedEventArgs)
820

'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053"), _
    System.Diagnostics.DebuggerStepThroughAttribute(), _
    System.ComponentModel.DesignerCategoryAttribute("code")> _
825 Partial Public Class getAtributosCadastradosCompletedEventArgs
    Inherits System.ComponentModel.AsyncCompletedEventArgs

    Private results() As Object

830 Friend Sub New(ByVal results() As Object, ByVal exception As System
        .Exception, ByVal cancelled As Boolean, ByVal userState As
        Object)
        MyBase.New(exception, cancelled, userState)
        Me.results = results
    End Sub

835 '''<remarks/>
    Public ReadOnly Property Result() As atributo()
        Get
            Me.RaiseExceptionIfNecessary
            Return CType(Me.results(0), atributo())
840        End Get
    End Property
End Class

'''<remarks/>
845 <System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053")> _
Public Delegate Sub getTodasRequisicoesCompletedEventHandler (ByVal
    sender As Object, ByVal e As getTodasRequisicoesCompletedEventArgs)

```

```

'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053"), _
850   System.Diagnostics.DebuggerStepThroughAttribute(), _
    System.ComponentModel.DesignerCategoryAttribute("code")> _
Partial Public Class getTodasRequisicoesCompletedEventArgs
    Inherits System.ComponentModel.AsyncCompletedEventArgs

855   Private results() As Object

    Friend Sub New(ByVal results() As Object, ByVal exception As System
        .Exception, ByVal cancelled As Boolean, ByVal userState As
        Object)
        MyBase.New(exception, cancelled, userState)
        Me.results = results
860   End Sub

    '''<remarks/>
    Public ReadOnly Property Result() As requisicaoBL()
        Get
865            Me.RaiseExceptionIfNecessary
            Return CType(Me.results(0), requisicaoBL())
        End Get
    End Property
End Class

870
'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053")> _
Public Delegate Sub
    getRequisicaoPorNumeroIdentificadorCompletedEventHandler(ByVal
        sender As Object, ByVal e As
        getRequisicaoPorNumeroIdentificadorCompletedEventArgs)

875
'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services",
    "2.0.50727.3053"), _
    System.Diagnostics.DebuggerStepThroughAttribute(), _
    System.ComponentModel.DesignerCategoryAttribute("code")> _
Partial Public Class
    getRequisicaoPorNumeroIdentificadorCompletedEventArgs
880    Inherits System.ComponentModel.AsyncCompletedEventArgs

```

```

Private results () As Object

Friend Sub New(ByVal results () As Object, ByVal exception As System
    .Exception, ByVal cancelled As Boolean, ByVal userState As
    Object)
885     MyBase.New(exception, cancelled, userState)
        Me.results = results
End Sub

'''<remarks/>
890 Public ReadOnly Property Result () As requisicaoBL
    Get
        Me.RaiseExceptionIfNecessary
        Return CType(Me.results (0), requisicaoBL)
    End Get
895 End Property
End Class
End Namespace

```

Listagem A.83: app.config

```

ï»¿<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <configSections>
        <sectionGroup name="applicationSettings" type="System.Configuration
            .ApplicationSettingsGroup, System, Version=2.0.0.0, Culture=
            neutral, PublicKeyToken=b77a5c561934e089" >
5            <section name="WindowsApplication1.My.MySettings" type="System.
                Configuration.ClientSettingsSection, System, Version
                =2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
                requirePermission="false" />
        </sectionGroup>
    </configSections>
    <system.diagnostics>
        <sources>
10        <!-- This section defines the logging configuration for My.
            Application.Log -->
            <source name="DefaultSource" switchName="DefaultSwitch">
                <listeners>
                    <add name="FileLog"/>
                    <!-- Uncomment the below section to write to the
                        Application Event Log -->
15        <!--<add name="EventLog"/>-->

```

```

        </listeners>
    </source>
</sources>
<switches>
20     <add name="DefaultSwitch" value="Information" />
</switches>
<sharedListeners>
    <add name="FileLog"
        type="Microsoft.VisualBasic.Logging.FileLogTraceListener,
            Microsoft.VisualBasic, Version=8.0.0.0, Culture=neutral
            , PublicKeyToken=b03f5f7f11d50a3a,
25         processorArchitecture=MSIL"
        initializeData="FileLogWriter"/>
    <!-- Uncomment the below section and replace APPLICATION_NAME
        with the name of your application to write to the
        Application Event Log -->
    <!--<add name="EventLog" type="System.Diagnostics.
        EventLogTraceListener" initializeData="APPLICATION_NAME"/>
        -->
</sharedListeners>
</system.diagnostics>
30 <applicationSettings>
    <WindowsApplication1.My.MySettings>
        <setting name="
            WindowsApplication1_Requisicao_RequisicaoCertificadoWSService
            "
            serializeAs="String">
            <value>http://localhost:8080/ACA_WS/aca/webservicereq.ws</
                value>
35     </setting>
        <setting name="
            WindowsApplication1_emissao_EmissaoCertificadoWSService"
            serializeAs="String">
            <value>http://localhost:8080/ACA_WS/aca/webserviceemis.ws</
                value>
40     </setting>
        <setting name="
            WindowsApplication1_Configuracao_ACAConfiguracaoWSService"
            serializeAs="String">
            <value>http://localhost:8080/ACA_WS/aca/webserviceconfig.ws
                </value>
            </setting>
        <setting name="

```



```
WindowsApplication1_LCR_RevogarCertificadoWSService"  
45   serializeAs="String">  
      <value>http://localhost:8080/ACA_WS/aca/websvicerevogacao  
        .ws</value>  
    </setting>  
  </WindowsApplication1.My.MySettings>  
  </applicationSettings>  
50 </configuration>
```
