



Centro Universitário de Brasília – UniCEUB
Faculdade de Exatas e Tecnologia - FAET
Curso de Engenharia da Computação

Sebastião Fabiano Silva Costa

Aplicação Web de Gerenciamento Eletrônico de Documentos com Criptografia

Brasília

2010



CENTRO UNIVERSITÁRIO DE BRASÍLIA - UniCEUB
CURSO DE ENGENHARIA DE COMPUTAÇÃO

Sebastião Fabiano Silva Costa

Aplicação Web de Gerenciamento Eletrônico de Documentos com Criptografia

Orientador: Prof. Marco Antônio de O. Araújo

Brasília

2010

Sebastião Fabiano Silva Costa

Aplicação Web de Gerenciamento Eletrônico de Documentos com Criptografia

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de Certificado de Conclusão de Curso de Engenharia de Computação.

Orientador: Prof.

(nome completo)

Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro de Computação, e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas -FATECS.

Prof. Abiezer Amarilia Fernandez
Coordenador do Curso

Banca Examinadora:

Prof. Marco Antônio - Mestre em Ciência da Computação.
Orientador

Prof. Antonio Barbosa - Especialista em Engenharia de Software.

Prof. Vera Lúcia Farini Alves Duarte - Mestrado

Prof. Leonardo Pol – Mestre em Ciência da Computação

DEDICATÓRIA

Para Darlene e Felipe, minha vida, minha força.

AGRADECIMENTOS

Agradeço este projeto primeiramente a Deus por ter me concedido a graça de poder estudar e ter a capacidade de aprender, ler, pensar e agir. Dedico a minha esposa que teve muita paciência e apoio para que eu pudesse finalizar esta graduação. Dedico aos meus pais que me ajudam e me apoiarão sempre na vida.

SUMÁRIO

SUMÁRIO	IV
LISTA DE FIGURAS.....	V
LISTA DE TABELAS	VI
LISTA DE ABREVIATURAS	VII
CAPÍTULO 1 – INTRODUÇÃO	1
1.1 MOTIVAÇÃO	1
1.2 OBJETIVOS.....	2
1.2.1 OBJETIVO PRINCIPAL	2
1.2.2 OBJETIVOS SECUNDÁRIOS	3
1.3 METODOLOGIA DE PESQUISA.....	3
1.4 ESTRUTURA DA MONOGRAFIA	3
CAPÍTULO 2 – REFERENCIAL TEÓRICO.....	4
2.1 GED-GERENCIAMENTO ELETRÔNICO DE DOCUMENTOS	4
2.2 CRIPTOGRAFIA	6
2.3 SERVIDOR WEB.....	13
2.4 LINGUAGENS DE PROGRAMAÇÃO PARA WEB	14
2.4.1 ADOBE FLEX	14
2.4.2 JAVA	17
2.4.3 BLAZEDS	18
CAPÍTULO 3 – DESENVOLVIMENTO.....	19
3.1 CONCEITOS INICIAIS	19
3.1.1 INTERAÇÕES COM OS USUÁRIOS	19
3.1.2 MODELAGEM DO SISTEMA.....	21
3.1.3 MODELAGEM DO BANCO DE DADOS.....	25
3.1.4 IMPLEMENTAÇÃO DO SISTEMA.....	26
3.2 MÓDULOS.....	28
3.2.1 MANTER USUÁRIOS	28
3.2.2 MANTER DOCUMENTOS	28
CAPÍTULO 4 – TESTE E RESULTADOS.....	30
4.1.1 TESTE IMPORTAÇÃO	30
4.1.2 RESULTADOS DA IMPORTAÇÃO	30
4.1.3 TESTE LOCALIZAÇÃO	32
4.1.4 RESULTADO LOCALIZAÇÃO.....	33
CAPÍTULO 5 – CONCLUSÃO	34
5.1 FUTUROS PROJETOS	35
REFERÊNCIAS BIBLIOGRÁFICAS.....	36
SITES.....	36
APENDICE I.....	37
APENDICE II.....	58

Lista de Figuras

FIGURA 2.1– PLANO DAS ORGANIZAÇÕES BRASILEIRAS PARA IMPLANTAR SOLUÇÕES GED.	6
FIGURA 2.2 - ESQUEMA GERAL PARA CIFRAGEM E DECIFRAGEM DE UM TEXTO	7
FIGURA 2.3 – MODELOS DE CHAVES DA CRIPTOGRAFIA	10
FIGURA 2.4 - DIAGRAMA DE SERVIDOR WEB.....	13
FIGURA 2.5 - DIAGRAMA DE EVOLUÇÃO DAS APLICAÇÕES.....	14
FIGURA 2.6 – PLATAFORMA DE DESENVOLVIMENTO ADOBE FLEX.....	16
FIGURA 2.7 – FLUXO DE PROCESSO DO ADOBE FLEX.....	16
FIGURA 2.7 – CONVERSOR FLEX+JAVA	18
FIGURA 2.8 – COMUNICAÇÃO FLEX + JAVA	18
FIGURA 3.1 – DIAGRAMA DE CASO DE USO MANTER USUÁRIO E MANTER DOCUMENTOS	20
FIGURA 3.2 – DIAGRAMA DE CASO DE USO REALIZAR LOGIN	20
FIGURA 3.3 – ALGORITMO CRIPTOGRAFIA AES EM BLOCO.	21
FIGURA 3.4 – ALGORITMO CRIPTOGRAFIA DES EM BLOCO.	22
FIGURA 3.5 – ALGORITMO CRIPTOGRAFIA DES EM BLOCO.	22
FIGURA 3.6 – DIAGRAMA DE SEQUÊNCIA MANTER USUÁRIO.....	23
FIGURA 3.7 – DIAGRAMA DE SEQUÊNCIA MANTER DOCUMENTOS.	24
FIGURA 3.8 – DIAGRAMA DE ENTIDADE RELACIONAL.	25
FIGURA 3.9 – ESTRUTURA DO PROJETO SIDOC.	26
FIGURA 3.10 – DESCRITOR DE CONFIGURAÇÃO DE SERVIÇOS JAVA.....	27
FIGURA 3.11 – TELA DE MANTER USUÁRIOS.	28
FIGURA 3.12 – TELA DE IMPORTAÇÃO DE DOCUMENTOS.	29
FIGURA 3.13 – TELA DE LOCALIZAÇÃO DOS DOCUMENTOS.....	29
FIGURA 4.1 – TELA DO ARQUIVO IMPORTADO PELO SISTEMA.....	30
FIGURA 4.2 – TELA DE ENVIO DE E-MAIL.....	31
FIGURA 4.3 – DE ARMAZENAMENTO DE ARQUIVOS.	31
FIGURA 4.4 – TENTANDO ABRIR O ARQUIVO CRIPTOGRAFADO.....	32
FIGURA 4.5 – TELA DE LOCALIZAÇÃO DOS DOCUMENTOS.....	32
FIGURA 4.6 – TELA DE DOWNLOAD DO DOCUMENTO.....	33

Lista de Tabelas

TABELA 2.1 - HISTÓRICO DE EVOLUÇÃO DA CRIPTOGRAFIA	8
TABELA 2.2 – ALGORITMOS POR APLICAÇÃO	9
TABELA 2.3 – ALGORITMOS POR APLICAÇÃO	11
TABELA 2.4 – ALGORITMOS SIMÉTRICOS MAIS CONHECIDOS	11
TABELA 2.5 – ALGORITMOS HASHING, E SUAS CARACTERÍSTICAS	12

Lista de Abreviaturas

- GED** – Gerenciamento Eletrônico de Documentos
- ERP** – Enterprise Resource Planning
- WORKFLOW** - Fluxo de trabalho
- SIDOC** – Sistema de Documentos Eletrônicos
- HTTP** – HyperText Transfer Protocol
- DNS** – Domain Name System
- IP** – Internet Protocol
- XML** – eXtensible Markup Language
- SGBD** – Sistema Gerenciador de Banco de Dados
- CSS** – Cascading Style Sheets
- SQL** – Structured Query Language
- HD** – Hard Disk
- JSE** - Java Standard Edition
- JME** - Java Micro Edition
- JEE** – Java Enterprise Edition
- RIA** – Rich Internet Application
- IDE** – Integrated Development Environment
- RPC** – Remote Procedure Call

RESUMO

O propósito é desenvolver um sistema web capaz de armazenar documentos e gerenciá-los de forma que simplifique o processo de gestão das informações, utilizando mecanismos de criptografia, adicionando assim uma segurança maior para as informações gerenciadas. Será enviado por e-mail, a senha para descriptografar o documento, e no processo de pesquisa dos documentos utilizará filtros, como título do documento, assunto, data em que foi armazenado, características do documento.

Palavras-Chave: GED;Upload;Download,Java, Flex, Criptografia.

ABSTRACT

The purpose and develop a web-based system capable of storing documents and manage them in a way that simplifies the process of managing information, using encryption mechanisms, thus adding greater security to the information managed by sending e-mail password decrypt the document. And the search for documents using filters, such as document title, subject, date it was stored, characteristics of the document

Keywords: GED, Upload, Download.

CAPÍTULO 1 – INTRODUÇÃO

1.1 MOTIVAÇÃO

Atualmente, a maioria das empresas tem uma imensa quantidade de documentos eletrônicos: são documentos em word, excell, pdf e etc. Esse novo mundo de documentos digitais, sem gerenciamento, implica em muitas cópias de diversas versões, duplicação de arquivos, e custos desnecessários com armazenamento. A pesquisa das informações geradas pelas empresas se torna mais onerosas à medida que estas informações são geradas, e com o passar do tempo a quantidade de dados eletrônicos cresce, para reduzir o tempo de pesquisa das informações internas a informática disponibiliza recursos de armazenamento e busca.

Analisando o problema de armazenamento de dados eletrônicos, a informática disponibiliza recursos para que as empresas possam guardar os seus dados. O gerenciamento de documentos é hoje utilizado em vários seguimentos, como por exemplo, em uma empresa de Recursos Humanos que contém milhares de currículos de distintos profissionais para distintas vagas. E para obter currículos que se relacionam a uma determina vaga necessita-se que se faça uma pesquisa no banco de currículos, buscando aqueles profissionais que estão dentro do contexto da vaga, contextos relacionados a atividades exercidas, formação acadêmica, cursos e outros dados que possa rastrear os currículos dos candidatos a vaga selecionada.

E necessário no mundo atual guardar e gerenciar informações eletrônicas, pois as informações são cada vez mais valiosas. As buscas das informações em tempo hábil são indispensáveis, com isso e as soluções tecnológicas são cada vez mais adquiridas pelas pessoas, e um dos mecanismos mais utilizados para buscar as informações hoje, e a internet, através dos sites de busca, assim analisando este pensamento as pessoas e as empresa sentem a necessidade de utilizar um mecanismo próprio de busca e gerenciamento dos seus dados eletrônicos. Os sistemas GED (Gerenciamento Eletrônico de Documentos), automatizam o processo de armazenamento e gerenciamento das informações, esta é a motivação para criar um sistema robusto que possa guardar e localizar dados de forma segura.

1.2 OBJETIVOS

O objetivo deste trabalho é apresentar soluções tecnológicas aplicadas na área de gerenciamento eletrônico de documento e criptografia, tomando como foco principal a criptografia de arquivos.

1.2.1 OBJETIVO PRINCIPAL

O propósito é desenvolver uma aplicação web com a função principal de armazenamento de documentos e localização, criptografando os dados armazenados. A aplicação será denominada de SIDOC (Sistema de Documentos Eletrônicos), e utilizará como tecnologia a linguagem de programação Java na sua plataforma JEE (Java Enterprise Edition) integrada com a plataforma de desenvolvimento de aplicações ricas para internet Adobe Flex. O sistema será distribuído e implantado em um servidor de aplicação, tendo como acesso principal um link via browser.

O contexto do sistema será de arquivar eletronicamente os documentos, e pesquisar os documentos armazenados utilizando filtros de pesquisa, como a data de armazenamento, nome do documento, tipo do documento, se o documento é de um determinado departamento da empresa ou se é de propriedade pessoal.

O mecanismo principal de armazenamento utilizado no sistema é fazer um upload do arquivo, similar ao que utilizamos, por exemplo, quando anexamos um arquivo a um e-mail. Ao fazer o upload do arquivo o sistema compactará o arquivo e criptografará o arquivo compactado. Desta forma o sistema adiciona uma confiabilidade das informações contidas no documento e um segurança na preservação do arquivo armazenado, as informações contidas no arquivo armazenado não serão lidas e nem manipuladas. A criptografia será utilizada no arquivo compactado e quando o sistema criptografar o arquivo será enviado um e-mail para o usuário com a senha da criptografia. Esta senha será utilizada para ele que possa obter o arquivo original descompactado e descriptografado.

1.2.2 OBJETIVOS SECUNDÁRIOS

Como objetivos secundários são dados o aprendizado dos mecanismos de gerenciamento eletrônico de documentos e a criptografia. Este trabalho exigiu um esforço na aprendizagem e pesquisa sobre a utilização de sistemas GED e da criptografia. Além destas duas vertentes principais podemos destacar a utilização das tecnologias Java e Adobe Flex, utilizando conceitos de RIA (Aplicações Ricas para Internet) e integração das duas plataformas.

1.3 METODOLOGIA DE PESQUISA

Este projeto segue uma linha de pesquisa bibliográfica baseada em consultas a materiais publicados em livros, periódicos e sites da Internet. Com base nesta pesquisa o projeto tem finalidade aplicada, tendo com isso aplicação prática e experimental de seu objeto de estudo [SANTOS,2000].

1.4 ESTRUTURA DA MONOGRAFIA

Esta monografia foi dividida em quatro capítulos:

No Capítulo 2 é apresentado o referencial teórico com as definições de cada tecnologia utilizada no trabalho e suas respectivas aplicações ao longo do projeto.

No Capítulo 3 é apresentado o desenvolvimento do projeto, descrevendo a implementação do software em seus módulos. É o capítulo que mostra o funcionamento do sistema.

No Capítulo 4 é apresentado os testes realizados na importação de documentos.

No Capítulo 5 é apresentada a conclusão do projeto e sugestões para futuras evoluções do mesmo.

CAPÍTULO 2 – REFERENCIAL TEÓRICO

2.1 GED-GERENCIAMENTO ELETRÔNICO DE DOCUMENTOS

O Gerenciamento Eletrônico de Documentos se resume basicamente em converter as diversas informações que geramos, sendo uma imagem, um texto, em forma digital. Sistemas GED, utilizam alguns recursos para guardar e digitalizar os dados, sendo que uma das ferramentas mais utilizada para a digitalização das informações é o escâner. Portanto após o dado ser transformado em arquivo digitalizado os sistemas GED armazenam e catalogam as informações em disco rígido (HD).

Muitos dos pioneiros do gerenciamento eletrônico de documentos aprenderam que solucionar as aplicações adequadamente e aplicar métodos corretivos no sistema existente são atividades importantes e devem ser realizadas antes de começar a pesquisa de novos equipamentos. Isso pode significar a diferença entre o fracasso e o sucesso [STARBIRD,1998].

Desta forma não basta termos o melhor sistema de gerenciamento de documentos disponível, temos que fazer que os dados sejam gerenciados antes de utilizarmos recursos computacionais, por exemplo, catalogar e classificar os dados digitais como por contexto e categoria, para assim podermos armazená-los e podermos localizá-los posteriormente de forma ágil. Armazenar somente não basta para que todo processo seja bem aproveitado. Ter um sistema de GED significa ter nas mãos a capacidade de gerenciar todo o capital intelectual da empresa. É a tecnologia do GED que torna o eBusiness uma realidade, pois alicerça todas as informações referentes a qualquer etapa de qualquer processo de negócio.

Dentro do conceito de GED, existem diversas tecnologias agregadas. A tecnologia que antecedeu o GED foi à microfilmagem, e ainda hoje é utilizada por muitas empresas. Algumas tecnologias utilizadas são, Document Imaging (DI), Document Management (DM), Workflow, COLD/ERM, Forms Processing e RIM [STARBIRD,1998]. A Document Imaging (DI), é a tecnologia que possibilita a digitalização de documentos de microfilme, assim diferenciaram os documentos de microfilme e os documentos textuais, pois os documentos textuais são em geral bem menores que os de imagem, e pela a maioria dos sistemas não fazem o processamento de imagens, assim para que seja processadas imagens devem transforma imagens em texto, para que possa ser editados, deste modo utiliza-se a tecnologia de Optical

Character Recognition(OCR), ou a Intelligent Character REcognition (ICR), para reconhecer textos manuscritos. Com o amadurecimento das tecnologias, surge à tecnologia de processamento de formulários a (Form Processing), que utiliza digitalizadores para retirar as informações, assim surge a necessidade de trâmite de documentos em papel e então utiliza-se a tecnologia de Workflow, para imagens de documentos. O Workflow é uma tecnologia que permite gerenciar os dados de uma forma mais pró-ativa, garantindo o acompanhamento constante de todas as atividades no ciclo de gerenciamento de informações digitais. O Workflow atua também com integração de diversos sistemas como Enterprise Resource Planning (ERP) e o Customer Relationship Management (CRM). No caso da tecnologia Records and Information Management (RIM), gerencia o ciclo de vida dos documentos independentemente da mídia, e encontra sendo gerenciado desde a criação do documento até a destruição. O Computer Output to Laser Disk / Enterprise Report Management é o processo de gerenciamento de relatórios, este processo é distribuído para consultas, e muitas vezes revisado e conferido, possibilitando fazer anotações sobre o conteúdo apresentado, sem que se altere o dado original. O processo de gerenciamento eletrônico dos documentos juntamente com o controle de versão é utilizado pela tecnologia de Documento Management (DM). Nela o foco é controlar as versões e históricos de alterações, bem como toda a evolução do documento. Essa tecnologia é utilizada nos sistema de controle de versão, com o Visual Source Safe da Microsoft e o Concurrent Versions System.

A composição de um sistema de GED se dá com o uso de equipamentos de captura/edição de imagens, uma ou várias tecnologias de armazenamento, que variam de acordo com a quantidade de documentos a ser convertidos, e de dispositivos de consulta/impressão. Para gerenciar e manter os documentos, se faz necessário o uso de sistemas operacionais e de banco de dados cliente/servidor, softwares de workflow, softwares de tratamento e recuperação de imagens, além de implementações de segurança no acesso e backups (cópias de segurança). A Figura 2.1 traz o planejamento das organizações brasileiras com relação à implantação de um sistema de GED até o ano de 2001.

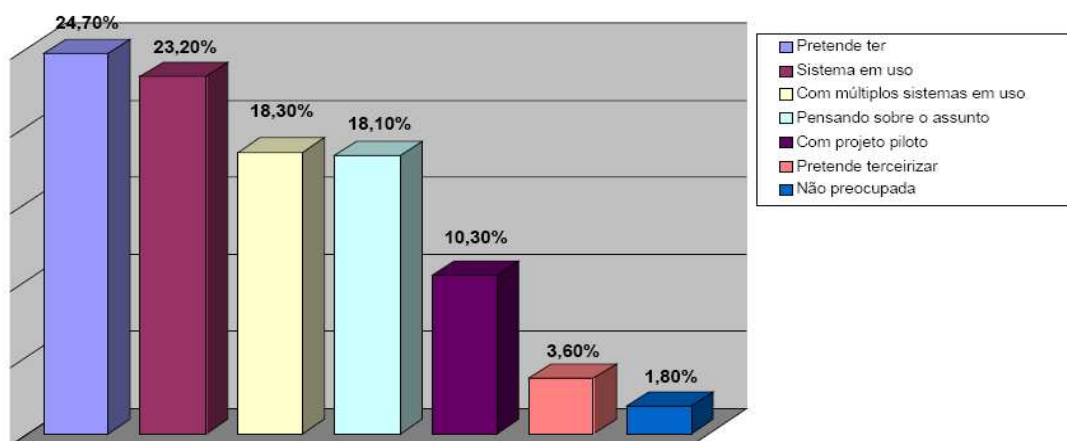


Figura 2.1– Plano das organizações brasileiras para implantar soluções GED.

Fonte: Pesquisa de Mercado CENADEM, 1999. (<http://www.cenadem.com.br>)

Assim o uso de uma ou de várias tecnologias de Gerenciamento Eletrônico de Documento, dependerá exclusivamente da necessidade de cada empresa. Hoje existem várias soluções pagas e livres para o controle e gerenciamento das informações digitais. O proposto do sistema Sidoc é fornecer principalmente segurança através da criptografia, para tal utiliza-se uma pequena amostra do conceito de Gerenciamento Eletrônico de Documentos.

2.2 CRIPTOGRAFIA

A criptografia é um mecanismo de esconder ou camuflar dados, deixando um dado ilegível, e somente quem detém a chave ou o algoritmo que transforma o dado ilegível para legível poderá obter a informação original. A criptografia age no ramo da Matemática, e a informática se utiliza dos recursos matemáticos para criptografar informações digitais. Na tecnologia da informação a criptografia é o processo de embaralhar os bits dos dados digitais. O significado da criptografia é formado a partir dos termos gregos *kryptos* (escondido, oculto) e *graphé* (grafia, escrita) e é a ciência que torna possível a comunicação mais segura entre dois agentes, sobre um canal aberto, convertendo informação legível em algo sem sentido, com a capacidade de ser recuperada ao estado original com o uso de processos inversos ou não.

O estudo da criptografia cobre bem mais do que apenas cifragem e decifragem, é um ramo especializado da teoria da informação com muitas contribuições de outros campos da matemática e do conhecimento, incluindo autores como Maquiavel, Sun Tzu e Karl von Clausewitz. A criptografia moderna é basicamente formada pelo estudo dos algoritmos criptográficos que podem ser implementados em computadores. O primeiro uso documentado da criptografia foi em torno de 1900 a.c. No Egito, quando um escriba usou hieróglifos fora do padrão numa inscrição [WIKIPEDIA, 2010]. A cifragem era e é utilizada em trocas de mensagens, como na guerra, para que o inimigo não conseguisse ler os textos gerados e também em outros seguimentos que a necessidade de esconder a informação. Atualmente, a criptografia é amplamente utilizada em aplicativos, no processo de autenticar usuários, e nas transações financeiras.

O algoritmo para cifrar um texto, é a cifra, e as operações dos algoritmos de cifra costumam utilizar uma chave criptografada e tal chave será um segredo em que somente quem terá conhecido serão os comunicantes. A cifra pode até ser conhecida, mas não a chave; assim como funciona um mecanismo de fechadura comum, não podendo abrir a porta sem uma chave real. A chave é um valor secreto que ao ser passado para o algoritmo de criptografia analisado e validado dentro do algoritmo, que se o valor da chave corresponder ao valor esperado, o dado será descriptografado e caso contrário o dado permanecerá criptografado.

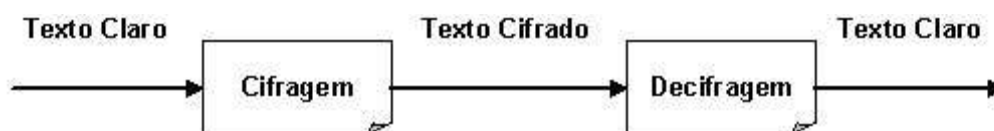


Figura 2.2 - Esquema geral para cifragem e decifragem de um texto

A figura 2.2, mostra a cifragem e decifragem de um texto. Basicamente existem dois meios de se criptografar informações, um usando métodos de cifras e outro com o uso de códigos, as cifras protegem as informações com o uso de substituição ou transposição de caracteres na mensagem original, sendo que apenas as pessoas que conhecem os processos de cifragem podem ter acesso às informações ali contidas. Porém caso se consiga descobrir a cifra, o algoritmo todo é exposto. Já os códigos, fazem uso de uma chave de criptografia que é introduzida no algoritmo, sendo assim mais seguros no aspecto onde, mesmo que se conheça o algoritmo de cifragem, a

informação ainda se manterá segura, sendo que ainda é necessário que se conheça a chave de criptografia para se decifrar a informação.

Com o passar dos anos, surgem novos algoritmos de criptografia e também esforços para quebrar a criptografia dos algoritmos existentes. A tabela abaixo mostra uma breve evolução histórica da criptografia.

Tabela 2.1 - Histórico de evolução da criptografia

Ano	Descrição
1943	Máquina Colossus, projetada para quebrar códigos.
1969	James Ellis desenvolve um sistema de chaves públicas e privadas separadas
1976	Diffie-Hellman, algoritmo baseado no problema do logaritmo discreto é criado e a IBM apresenta a cifra Lúçifer ao NBS (<i>National Bureau of Standards</i>) que após algumas modificações a NSA (<i>National Security Agency</i>) adota a cifra como padrão (FIPS 46-3, 1999) conhecido hoje como DES (<i>Data Encryption Standard</i>).
1977	Ronald L. Rivest, Adi Shamir e Leonard M. Adleman discutem como criar um algoritmo de chave pública prático e a partir disto desenvolvem um sistema baseado na dificuldade de fatoração de números primos grandes como técnica de criptografia, que futuramente seria denominado RSA, as iniciais de seus autores.
1978	O algoritmo RSA é publicado na <i>Association for Computing Machinery</i>
1991	Phil Zimmermann torna pública sua primeira versão do PGP (<i>Pretty Good Privacy</i>) em resposta as exigências do FBI (<i>Federal Bureau of Investigation</i>) de acessar qualquer texto claro da comunicação entre usuários de uma rede de comunicação digital. Por ser freeware tornou-se rapidamente padrão mundial.
1994	Ronald L. Rivest, autor dos algoritmos RC3 e RC4, publica a proposta do algoritmo RC5 na internet, o qual usa rotação dependente de dados como sua operação não linear e é parametrizado de forma que o usuário possa variar o tamanho da chave, do bloco de cifragem e o número de estágios usados no processo de cifragem das informações.
1997	O PGP 5.0 é amplamente distribuído para uso não comercial. O código DES de 56 bits é quebrado por uma rede de 14.000 computadores
1998	O código DES é quebrado em 56 horas por pesquisadores do vale do silício
1999	O código DES é quebrado em 22 horas e 15 minutos, mediante a união da Eletronic Frontier Foundation e a Distributed.Net, que reuniam 100.000

	computadores pessoais ao DES Cracker pela internet
2000	O NIST (<i>National Institute of Standards and Technology</i>), antigo NBS, anunciou um novo padrão de uma chave secreta de cifração escolhidos entre 15 candidatos através de concurso. Este novo padrão foi criado para substituir o algoritmo DES cujo tamanho das chaves tornou-se insuficiente para conter ataques de força bruta. O algoritmo Rijndael, cujo nome é abreviação do nome dos autores Rijmen e Daemen, foi escolhido para ser o futuro AES (<i>Advanced Encryption Standard</i>) (FIPS 197, 2001).

Os principais métodos usados de criptografia são a codificação (chaves) e a cifração (substituição e transposição), sendo que os mais utilizados atualmente são os de codificação, pois apresentam uma maior segurança com velocidade e capacidade de gerenciamento de chaves. Os tratamentos das informações podem ser criptografados em blocos com tamanho determinados ou de uma forma mais contínua denominado de fluxo de criptografia. A tabela 2.2, ilustra os algoritmos por sua aplicação.

Tabela 2.2 – Algoritmos por aplicação

Aplicação	Cifração recomendada	Comentários
Banco de Dados	Bloco	A interoperabilidade de um outro software não é relevante, mas é necessário reutilizar as chaves.
E-mail	AES	Se ganha interoperabilidade em todos os pacotes de e-mail utilizando o AES padrão
SSL	RC4	A velocidade é extremamente importante, cada conexão pode ter uma nova chave. Assim a maioria dos navegadores e servidores possuem RC4
Criptografia de Arquivos	Bloco	A interoperabilidade não é relevante, porém cada arquivo pode ser cifrado com a mesma chave.

Ao cifrarmos ou decifrarmos uma mensagem, precisamos de informações confidenciais, denominadas chaves. Os algoritmos de criptografia podem ser classificados em dois tipos, de acordo com o tipo de chave que usam: de chave simétrica e de chave assimétrica.

- Chave simétrica - A chave simétrica conhecida como chave única, utiliza a mesma chave para cifrar e decifrar a informação, ficando assim bastante limitado, pois o emissor e receptor deverão conhecer antecipadamente a chave, tornando difícil de se conseguir um meio seguro para enviar a chave.
- Chave assimétrica – Conhecida com chave pública e privada, utilizando chaves diferentes para cifrar e decifrar a informação. Em um sistema de chave assimétrica cada pessoa tem duas chaves: uma chave pública que pode ser divulgada e outra privada que deve ser mantida em segredo.

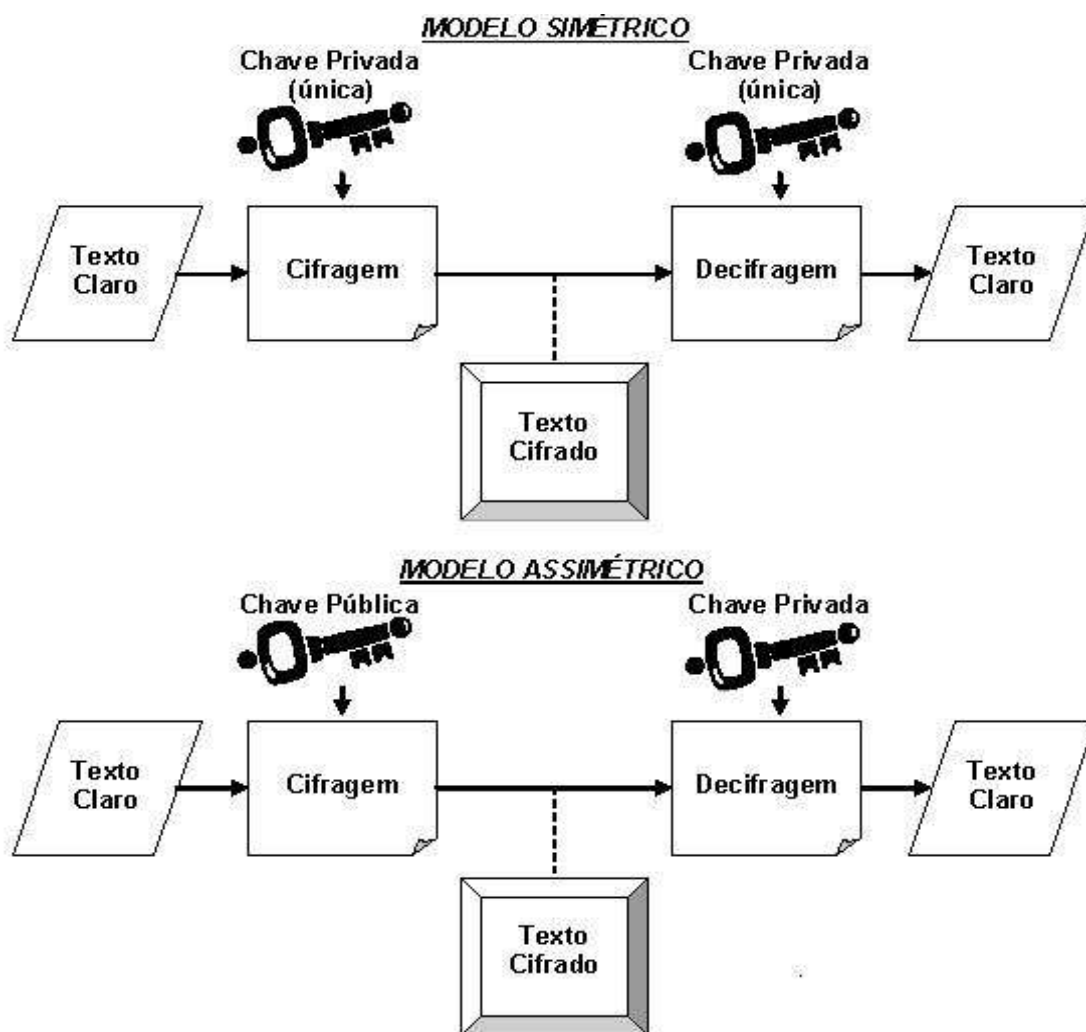


Figura 2.3 – Modelos de Chaves da Criptografia

O algoritmo de chave simétrica é classificado como rápido devido a sua particularidade de usar apenas uma única chave de tamanho pré-determinado, enquanto comparado a um de chave assimétrica que utilizará duas chaves de mesmo tamanho tornando assim o processo mais lento, quando existir a necessidade de troca contínua de chaves. Já com relação ao gerenciamento e distribuição de chaves, como o algoritmo assimétrico possui o recurso de chave pública para cifrar informação, este recurso é muito utilizado para o envio, troca e gerenciamento seguro de suas chaves, enquanto que no simétrico isto não é possível, sendo necessárias outras formas para troca e gerenciamento de chaves [MORENO,2005].

Os recursos extras como a assinatura digital desenvolvida e utilizada hoje, nada mais é do que o uso invertido do algoritmo de chave assimétrica. Seu funcionamento utiliza a chave privativa de quem assina para cifrar a informação e a pública para decifrar, comprovando assim que a informação tem origem confiável uma vez que, pelo funcionamento do algoritmo, apenas quem poderia ter codificado aquela informação seria o portador da chave privativa, a tabela 2.3 abaixo mostra uma comparação entre criptografia simétrica e assimetria.

Tabela 2.3 – Algoritmos por aplicação

Criptografia simétrica	Criptografia assimétrica
Rápida	Lenta
Gerência e distribuição de chaves complexa	Gerência e distribuição de chaves simples
Sem recursos como assinatura digital	Com recursos como assinatura digital

Alguns dos algoritmos simétricos mais utilizados, identificado o tipo e o tamanho da chave. A tabela 2.4 mostra as características de alguns algoritmos simétricos.

Tabela 2.4 – Algoritmos simétricos mais conhecidos [MORENO,2005]

Algoritmo	Tipo	Tamanho da chave	Tamanho do bloco
DES	Bloco	56 bits	64 bits
Triplo DES	Bloco	56 bits	64 bits

IDEA	Bloco	128 bits	64 bits
Blowfish	Bloco	32 - 448 bits	64 bits
RC4	Fluxo	0 - 256 bits	-----
RC5	Bloco	0 - 2040 bits	32-64-128 bits
CAST-128	Bloco	40 - 128 bits	64 bits
AES	Bloco	128-192-256 bits	128-192-256 bits

O algoritmo RSA é um dos algoritmos assimétricos mais utilizados por conter uma chave pública ou privada que realiza o processo de cifragem e decifragem das informações. Nos algoritmos possuem uma evidência de serem mais eficientes, com os métodos de assinatura digital, que na prática não cifram os documentos propriamente ditos, utilizam processamento através de funções hashing. As funções hashing, geram as saídas independentes da informação original, com um tamanho fixo que varia entre 128bit a 256bit dependendo do algoritmo adotado, assim não importando a saída, assim pode-se ter outro dado separado da informação com o qual se pode utilizar para certificar a integridade e originalidade da informação, funções de hash têm as características de ser simples, a tabela 2.5 mostra alguns dos algoritmos de hashing.

Tabela 2.5 – Algoritmos Hashing, e suas características [MORENO,2005]

Algoritmo de hash	Tamanho do hash	Kbytes/s
Abreast Bavies-Meyer (c/IDEA)	128 bits	22
Davies-Meyer (c/DES)	64 bits	9
GOST-Hash	256 bits	11
NAVAL (5 passos)	Variável	118
MD4 – Message Digest 4	128 bits	236
MD5 – Message Digest 5	128 bits	174
N-NASH (15 rounds)	128 bits	24
RIPE-MD	128 bits	182
SHA – Secure Hash Algorithm	160 bits	75
SENEFRU (8 passos)	128 bits	23

2.3 SERVIDOR WEB

O servidor web é um programa que recebe um pedido HTTP feito por um *browser* e esse pedido é composto por um domínio no qual é traduzido pelo DNS para seu respectivo IP. Ao receber o pedido, o servidor procura os documentos relativos ao mesmo e retorna o HTML via protocolo HTTP ao *browser* (Figura 2.4).

Para o sistema SIDOC, utilizamos o Servidor de Aplicações Java da Apache Software Foundation, Tomcat na sua versão 6.28. O Tomcat é um Servidor Web que contempla parte da especificação da plataforma de desenvolvimento Java EE (Java Enterprise Edition), e abrange as tecnologias Servlets e Jsp, que estão especificadas nesta plataforma. O Tomcat tem a capacidade de atuar também como servidor web/HTTP autônomo, ou pode funcionar integrado a um servidor web dedicado, como Apache httpd ou Microsoft IIS, ou ainda como parte integrante de um servidor de aplicações mais amplo, como JBoss AS.

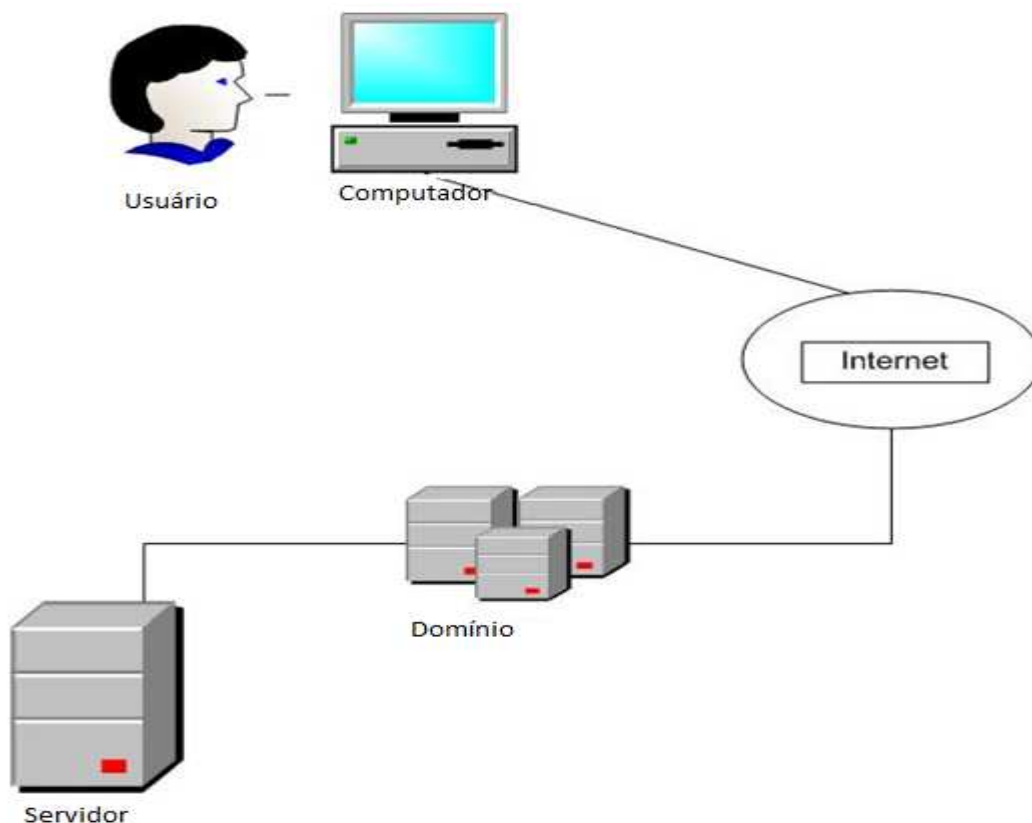


Figura 2.4 - Diagrama de Servidor Web

2.4 LINGUAGENS DE PROGRAMAÇÃO PARA WEB

2.4.1 ADOBE FLEX

O Adobe Flex é uma tecnologia adquirida pela Adobe Systems Incorporated, e que a Macromedia lançou em 2004. Hoje a tecnologia Flex é mantida e comercializada pela Adobe, esta tecnologia possibilita o desenvolvimento de aplicações ricas para Internet (RIA). Ela é executada através do Adobe Flash Player. A figura abaixo ilustra o processo de evolução das aplicações até chegarmos ao conceito de RIA.

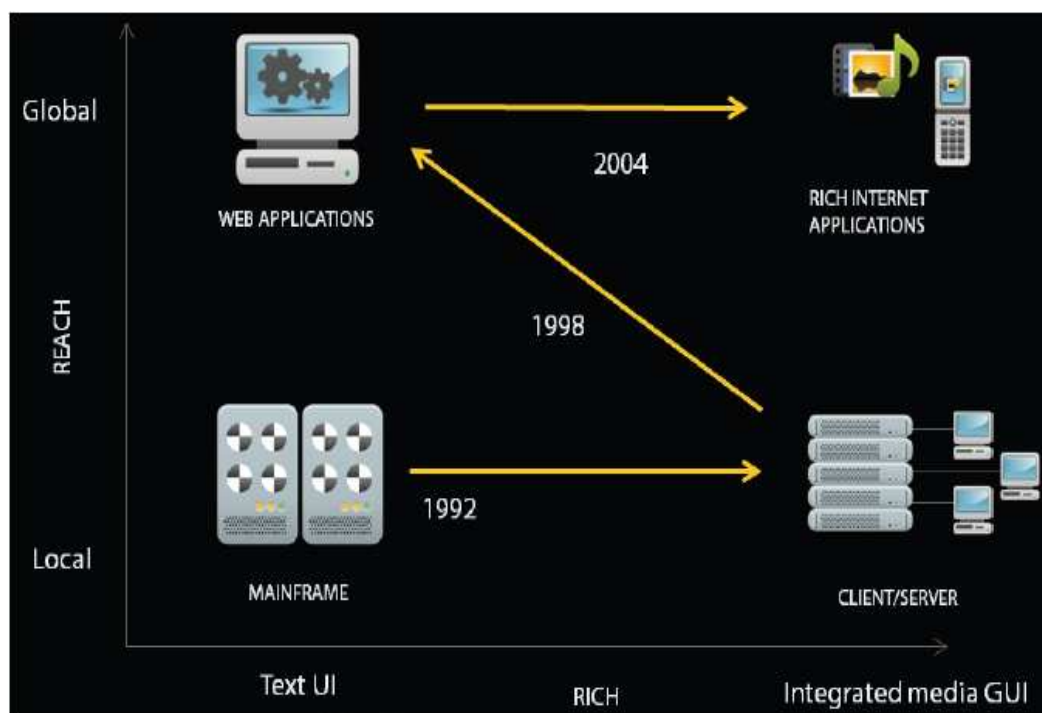


Figura 2.5 - Diagrama de Evolução das Aplicações

O Adobe Flex contém um kit de desenvolvimento, similar ao Kit de desenvolvimento do Java, onde estão as classes de implementação padrão e a máquina virtual para compilação dos artefatos. No Flex é gerado um arquivo .swf após a compilação da classe, e esse arquivo .swf é executado no Adobe Flash Player, e quando o usuário acessa a aplicação via um navegador, e neste momento este arquivo .swf fica instalado no temporário do sistema operacional do usuário.

O Flex tem a característica de executar em navegadores desde que esteja instalado o Adobe Flash Player, e as aplicações RIA são caracterizadas por serem aplicações tradicionais Desktop, porém são executadas em navegadores e não precisam

de instalação, rodando em um ambiente seguro chamado de “Sandbox”. Sandbox pode ser entendido como a zona de segurança na qual um arquivo SWF está operando. O tipo de sandbox indica a zona de segurança, ou seja, no Flash Player, todos os arquivos SWF são colocados em um dos quatro tipos de sandbox:

- **Remote** - Todos os arquivos provenientes de URLs não locais são colocados em um sandbox remoto. Há muitos sandboxes dessa natureza, um para cada domínio de Internet (ou intranet) a partir do qual os arquivos são carregados.
- **Local-with-filesystem** - O sandbox padrão para arquivos locais. Arquivos SWF neste sandbox não podem acessar a Internet (ou qualquer servidor) de forma alguma. Não é permitido o acesso local de rede com endereços tais como URL's e HTTP.
- **Local-with-networking** - Um arquivo SWF neste sandbox pode comunicar com a rede (Internet), mas não pode acessar arquivos locais.
- **Local-trusted** - Este sandbox não possui restrições. Qualquer arquivo local pode ser colocado neste sandbox se a autorização for dada pelo usuário final. Esta autorização pode vir em duas formas: interativa por meio do Settings Manager ou não interativa, por meio de um instalador executável que cria arquivos de configuração do Flash Player no computador do usuário.

O Adobe Flex possui duas linguagens o ActionScript que é orientado a objeto, sendo a classe do Flex, e o MXML que contém todas as diretivas para desenvolver as páginas RIA. A Adobe também disponibiliza uma IDE para desenvolvimento, que é o Flex Builder que está na versão 4, e hoje é gratuita para estudantes e desempregados, a figura 2.6 ilustra os elementos da plataforma de desenvolvimento Adobe Flex.



Figura 2.6 – Plataforma de Desenvolvimento Adobe Flex

O Adobe Flex Builder é uma ferramenta baseada no Eclipse. O Eclipse é um IDE gratuita para desenvolvimento de várias linguagens, com Java, PHP, C++ e outras. O Flex Builder tem algumas características que auxilia no desenvolvimento das aplicações Flex, com a codificação inteligente, o debug interativo, e o seu design de interface visual, a figura abaixo ilustra o fluxo de processo das aplicações Flex.

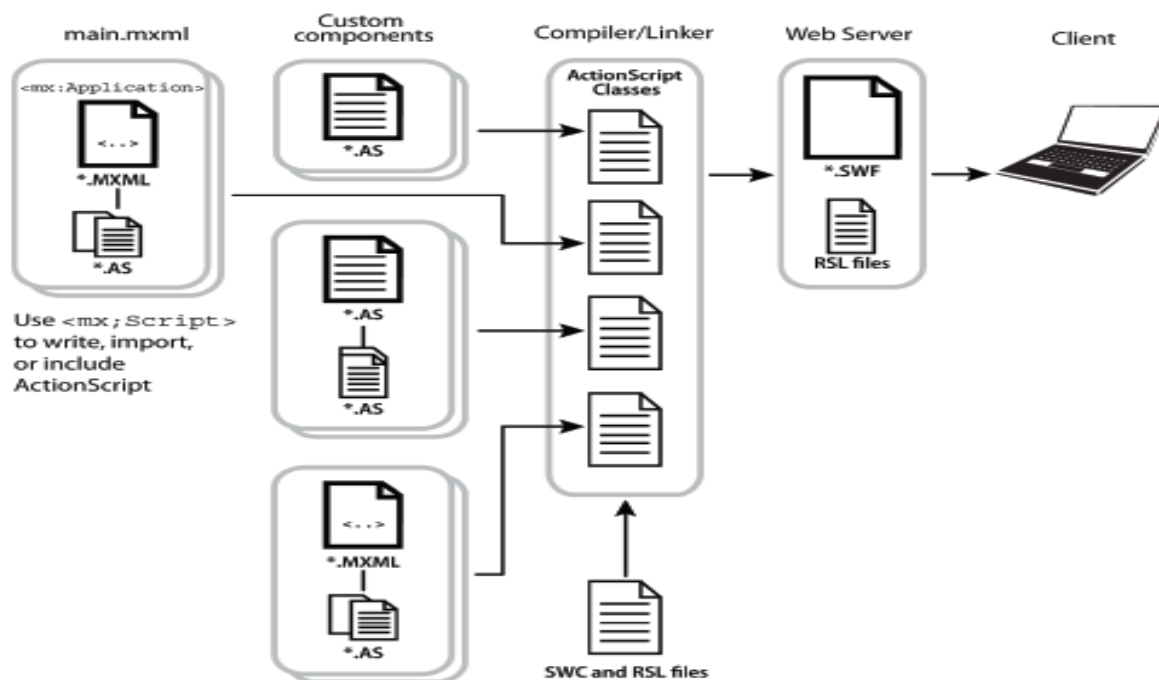


Figura 2.7 – Fluxo de Processo do Adobe Flex

2.4.2 JAVA

A linguagem de programação Java foi criada pela Sun Microsystems, é um linguagem orientada a objeto, compilada para um bytecode e executada numa máquina virtual, possuindo as seguintes plataformas:

- JSE - (Java Standard Edition) é o ambiente de desenvolvimento mais utilizado, por conter classe utilitária que se estende para ambientes Desktops e Servidores, sendo assim a plataforma principal.
- JEE - (Java Enterprise Edition) é a plataforma Java voltada para redes, Internet, Intranets e afins. Ela contém bibliotecas especialmente desenvolvidas para o acesso a servidores, a sistemas de e-mail, a banco de dados, etc.
- JME - (Java Micro Edition) é o ambiente de desenvolvimento para dispositivos móveis ou portáteis, como telefones celulares e palmtops.

No desenvolvimento da aplicação sidoc iremos utilizar os recursos das Plataformas JSE e JEE. As principais características da linguagem Java são: [WIKIPEDIA, 2010]

Orientação a objetos

Portabilidade - Independência de plataforma - "escreva uma vez, execute em qualquer lugar" ("*write once, run anywhere*");

Recursos de Rede - Possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP;

Segurança - Pode executar programas via rede com restrições de execução

Sintaxe similar a Linguagem C/C++.

Facilidades de Internacionalização - Suporta nativamente caracteres Unicode;

Simplicidade na especificação, tanto da linguagem como do "ambiente" de execução (JVM);

É distribuída com um vasto conjunto de bibliotecas (ou API);

Possui facilidades para criação de programas distribuídos e multitarefa (múltiplas linhas de execução num mesmo programa);

Alocação de memória automática por processo de coletor de lixo (*garbage collector*);

2.4.3 BLAZEDS

Para utilizar Flex e Java juntos precisamos de um framework para integrar as duas linguagens. Desta forma decidimos utilizar o framework BlazeDS que é mantido pela Adobe. Esta integração Java e Flex utiliza chamadas RPC, e possui um protocolo específico chamado de AMF.

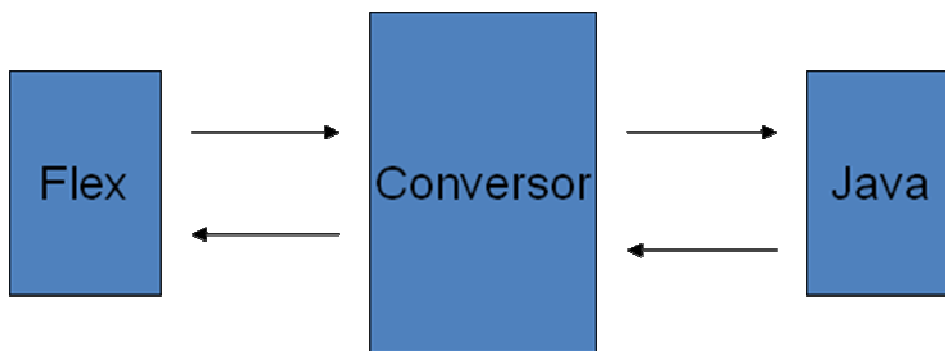


Figura 2.7 – Conversor Flex+Java

O BlazeDS [ADOBE, 2010] é a tecnologia remota Java com base em servidor de mensagens na Web que permite que os desenvolvedores se conectem facilmente aos dados distribuídos e envia os dados em tempo real para os aplicativos Flex. Ele consiste em duas partes cliente e servidor de aplicação JEE.

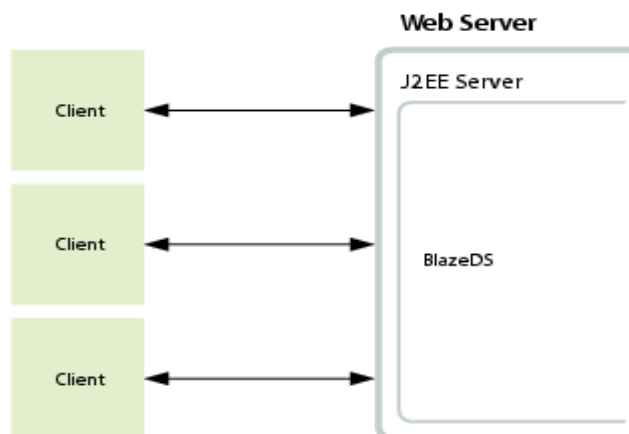


Figura 2.8 – Comunicação Flex + Java

CAPÍTULO 3 – DESENVOLVIMENTO

Este capítulo tem como objetivo apresentar a análise desde a proposta em desenvolver um sistema de Gerenciamento Eletrônico de Documentos com Criptografia, denominado de Sidoc (Sistema de Documentos Eletrônicos).

3.1 CONCEITOS INICIAIS

O Sidoc é um sistema projetado para gerenciar documentos eletrônicos, e foi adicionada na arquitetura do sistema mecanismos de criptografia. A arquitetura foi dividida em duas tecnologias, sendo na camada de apresentação a plataforma da Adobe Flex, e na camada de negócio com os mecanismos de criptografia a linguagem Java, as duas tecnologias se comunicam via protocolo AMF.

O sistema WEB é executando no servidor de aplicação Tomcat e banco de dados MySQL, o sistema possui dois módulos, sendo um de cadastro de usuários e um de gerenciamento de documentos. A utilização do sistema é simples e intuitivo, tendo uma interface visual amigável com janelas deslizantes e efeitos gráficos parecido com uma aplicação desktop.

3.1.1 Interações com os usuários

O sistema possui dois módulos principais:

- Manter Usuários
- Manter Documentos

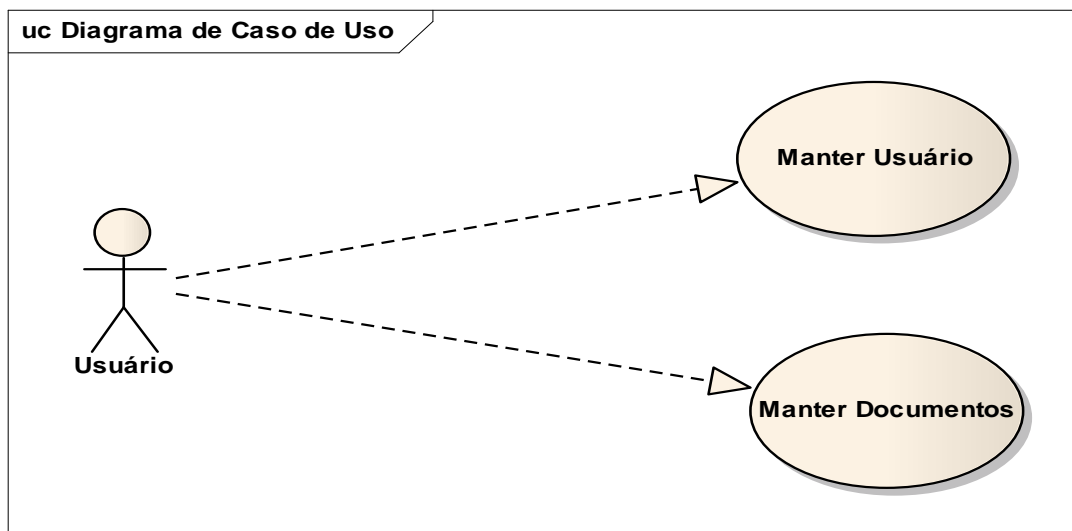


Figura 3.1 – Diagrama de Caso de Uso Manter Usuário e Manter Documentos

Na figura 3.1 observamos os dois casos de uso principais do sistema, e notamos que toda interação é feita pelo usuário do sistema. O caso de uso manter usuários tem a funcionalidade de cadastro dos usuários do sistema, nele será cadastrado o perfil do usuário e seu departamento, estes dados de perfil e departamento são fictícios, pois depende de cada empresa informar os seus respectivos perfis e departamentos. E no caso de uso manter documentos é realizado em duas etapas a importação e a localização do documento.

Além dos dois casos de usos principais mencionados, temos o início de toda interação que é a autenticação do usuário, neste ponto descrevemos o caso de uso realizar login, a figura 3.2 mostra esta interação.

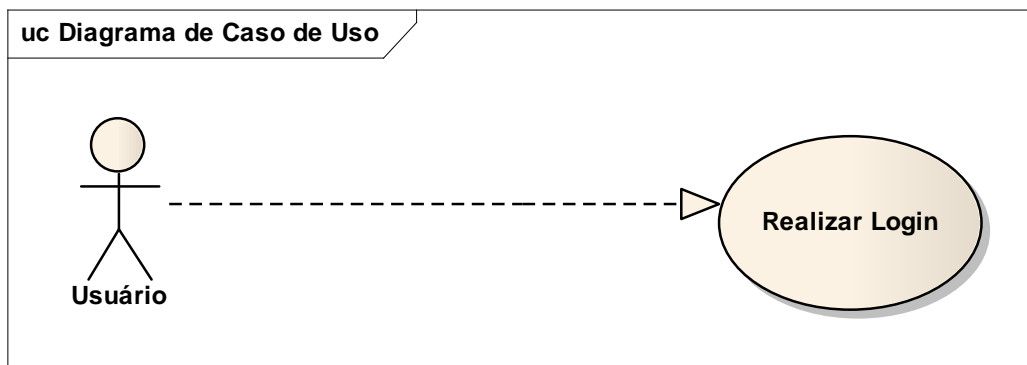


Figura 3.2 – Diagrama de Caso de Uso Realizar Login

3.1.2 Modelagem do Sistema

O sistema foi desenvolvido utilizando o conceito de módulos, onde cada um possui sua atribuição de forma bem definida. Apesar de utilizar este conceito, um módulo depende da ação de outro. Sua arquitetura é totalmente baseada em sistemas web utilizando a tecnologia Java e Adobe Flex.

Nas interfaces com o usuário é utilizada a tecnologia Adobe Flex, esta tecnologia é composta por duas linguagens a Action Script e a MXML, ambas composta na arquitetura do sistema. A linguagem Action Script é utilizada nas ações dos componentes com chamadas a serviços Java e manipulação de informações, a linguagem MXML é utilizada na composição de componentes como caixas de texto e grid's. A camada de negócio é composta por serviços Java, que são responsáveis por todo fluxo de dados do sistema, e nela foi utilizado a API Java Cryptography Extension (JCE), que contém todos os mecanismos necessários para implementar a criptografia de arquivos.

O sistema possui três implementações de criptografia para demonstração acadêmica, sendo os algoritmos AES, DES e Triple DES, visto que, não tem a necessidade de mais de um algoritmo seguro. A figura 3.3 abaixo mostra o trecho de implementação do algoritmo de criptografia.

```

Cipher ecipher;
Cipher dcipher;

public CriptografaArquivosAES(SecretKey key) {
    // Cria um vetor de 16-byte;
    byte[] iv = new byte[] { (byte) 0x8E, 0x12, 0x39, (byte) 0x9C, 0x07,
        0x72, 0x6F, 0x5A, (byte) 0x8E, 0x12, 0x39, (byte) 0x9C, 0x07,
        0x72, 0x6F, 0x5A};

    AlgorithmParameterSpec paramSpec = new IvParameterSpec(iv);
    try {
        ecipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        dcipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

        // CBC requer inicialização do vetor dos blocos.
        ecipher.init(Cipher.ENCRYPT_MODE, key, paramSpec);
        dcipher.init(Cipher.DECRYPT_MODE, key, paramSpec);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Figura 3.3 – Algoritmo Criptografia AES em bloco.

A classe Cipher é a classe principal da API de criptografia, ela é criada pelo método getInstance, que recebe como parâmetro uma string, que pode ser:

- "algoritmo/modo/regime"
- "algorithm"

Nesta implementação é utilizado o Algoritmo AES, o modo CBC que é criptografia e blocos, e o regime PKCS5Padding que é criptografia com senha implementada pelos laboratórios RSA na versão 1.5 de novembro de 1993.

Para implementar os outros algoritmos, basta modificar o algoritmo para o algoritmo desejado, como demonstrado nas figuras 3.4 e 3.5 abaixo:

```
public CriptografaArquivosDES(SecretKey key) {
    // // Cria um vetor de 8-byte;
    byte[] iv = new byte[] { (byte) 0x8E, 0x12, 0x39, (byte) 0x9C, 0x07,
        0x72, 0x6F, 0x5A };

    AlgorithmParameterSpec paramSpec = new IvParameterSpec(iv);
    try {
        ecipher = Cipher.getInstance("DES/CBC/PKCS5Padding");
        dcipher = Cipher.getInstance("DES/CBC/PKCS5Padding");

        // CBC requer inicialização do vetor dos blocos.
        ecipher.init(Cipher.ENCRYPT_MODE, key, paramSpec);
        dcipher.init(Cipher.DECRYPT_MODE, key, paramSpec);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figura 3.4 – Algoritmo Criptografia DES em bloco.

```
Cipher ecipher;
Cipher dcipher;

public CriptografaArquivosTripleDes(SecretKey key) {
    // Cria um vetor de 8-byte;
    byte[] iv = new byte[] { (byte) 0x8E, 0x12, 0x39, (byte) 0x9C, 0x07,
        0x72, 0x6F, 0x5A };

    AlgorithmParameterSpec paramSpec = new IvParameterSpec(iv);
    try {
        ecipher = Cipher.getInstance("TripleDES/CBC/PKCS5Padding");
        dcipher = Cipher.getInstance("TripleDES/CBC/PKCS5Padding");

        // CBC requer inicialização do vetor dos blocos.
        ecipher.init(Cipher.ENCRYPT_MODE, key, paramSpec);
        dcipher.init(Cipher.DECRYPT_MODE, key, paramSpec);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figura 3.5 – Algoritmo Criptografia DES em bloco.

O fluxo de sequência do Módulo de Manter Usuário são descritas no diagrama na figura 3.6.

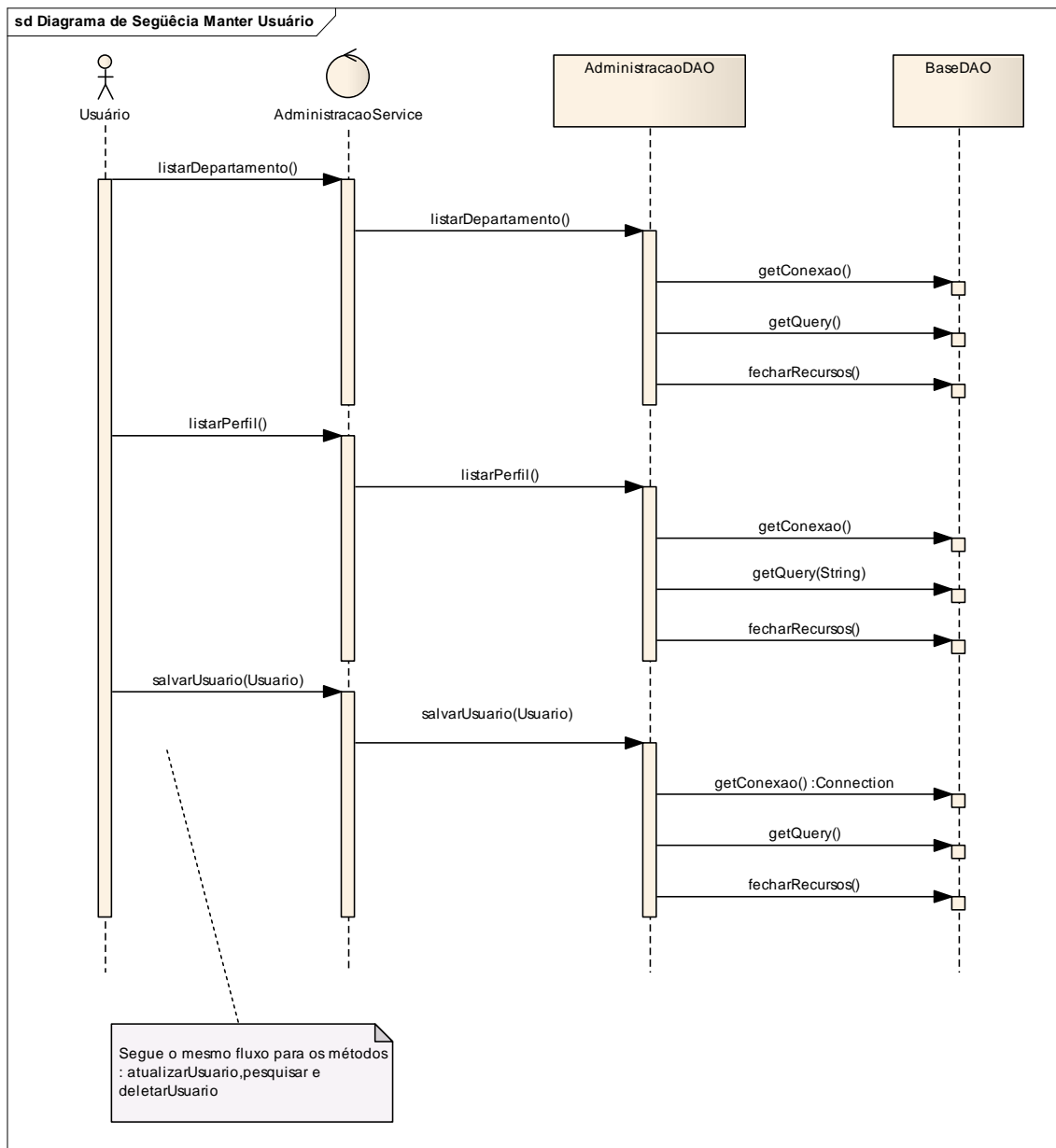


Figura 3.6 – Diagrama de Sequência Manter Usuário.

Neste diagrama vimos o fluxo de interação do usuário com as funcionalidades do módulo de manter usuário. O diagrama de seqüência do módulo manter documentos é demonstrado na figura 3.7.

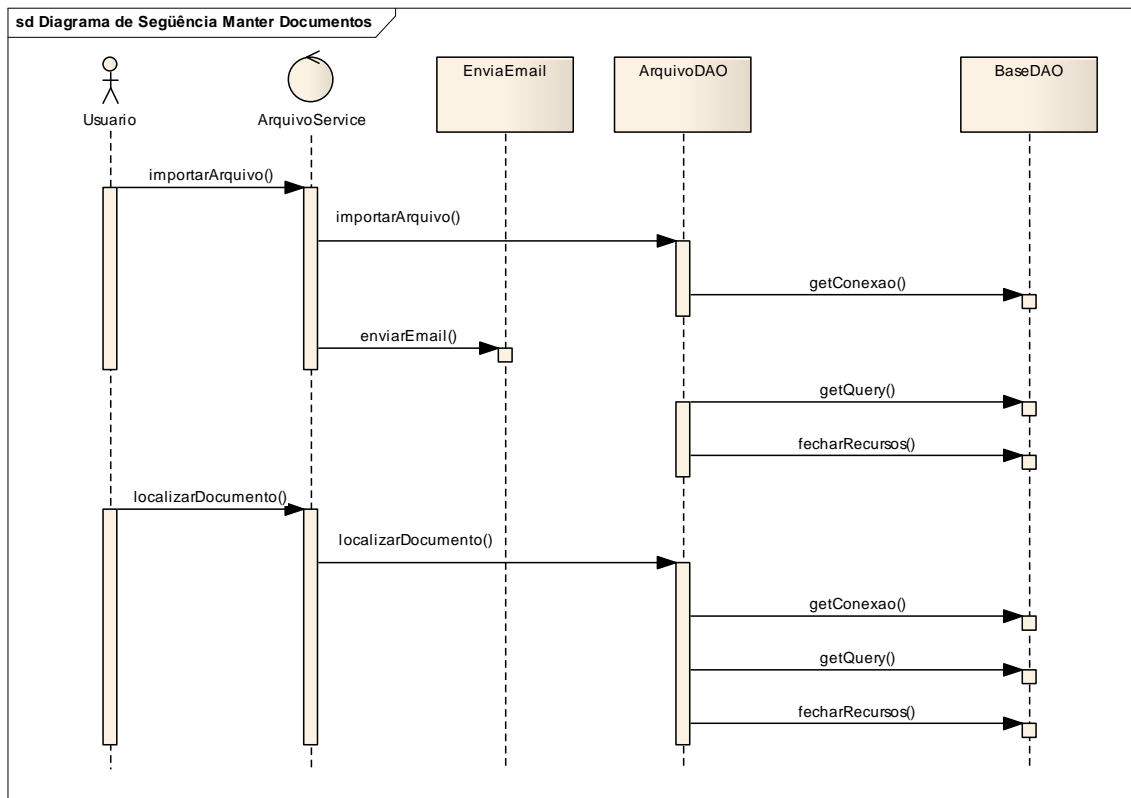


Figura 3.7 – Diagrama de Sequência Manter Documentos.

3.1.3 Modelagem do Banco de dados

O banco de dados foi desenvolvido de acordo com a regra de negócio do sistema definido nos conceitos iniciais e pelos objetos definidos na modelagem do sistema. O banco de dados utilizado é o MySQL na sua versão 5.1, o modelo de entidade relacional é mostrado na figura 3.8 abaixo:

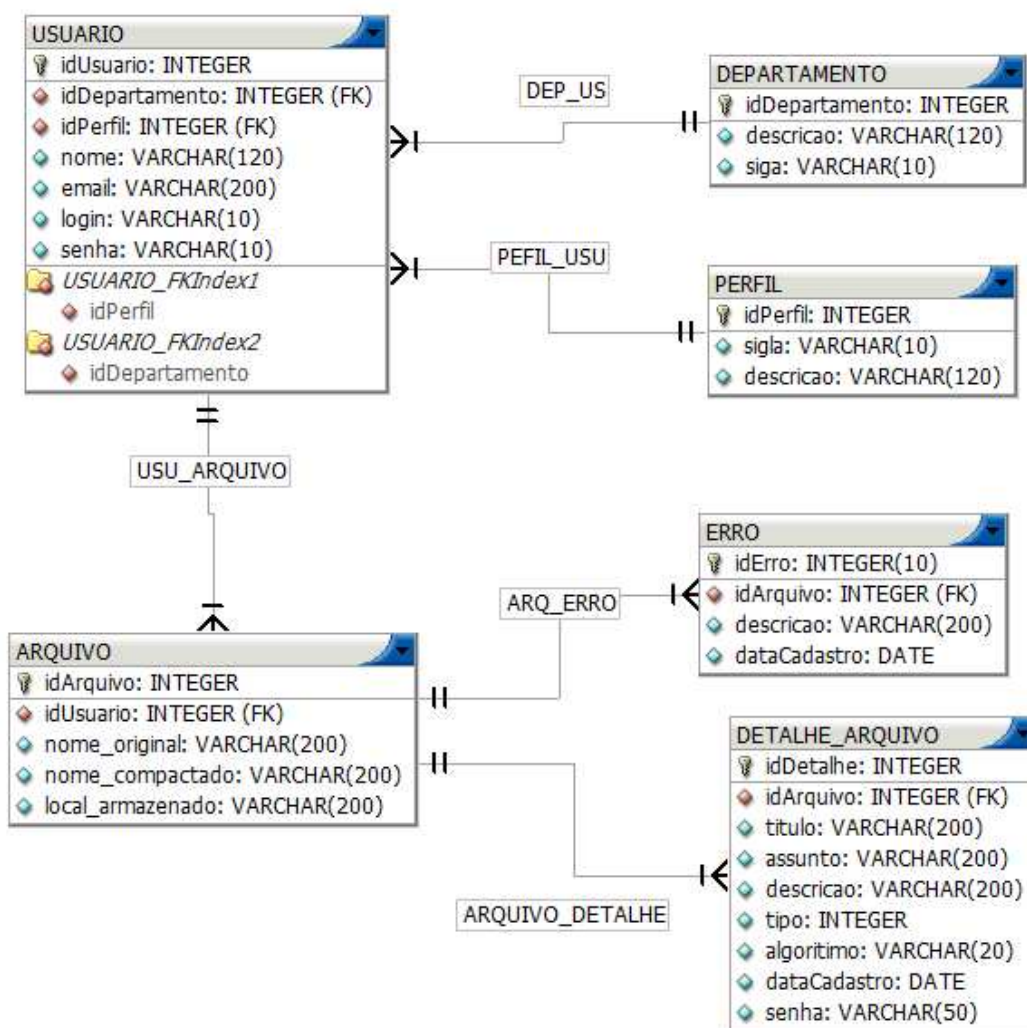


Figura 3.8 – Diagrama de Entidade Relacional.

As tabelas Usuário, Perfil e departamento contêm os dados principais de acesso ao sistema e de manipulação das funcionalidades. Nas tabelas Arquivo e Detalhe Arquivo, contém os dados dos arquivos gerenciados pelo sistema, e na tabela Erro, é gerado logs dos erros decorrentes no gerenciamento dos dados.

3.1.4 Implementação do Sistema

O projeto do sistema foi criado utilizando a IDE de desenvolvimento eclipse na sua versão 3.4.2 com o plugin do FlexBuilder versão 3, e tem a sua estrutura demonstrada na figura 3.9. Este projeto é composto por artefatos Java e Flex, separadamente por pacotes, os projetos construídos nestas duas plataformas podem ser unidos ou separados, inclusive separação física, pois as duas tecnologias são desacopladas por inteiro. A vantagem de se construir sistemas com baixo acoplamento entre camadas é a evolução que cada camada pode ocorrer sem que se tenha interfere com o todo, e além disso poder substituir uma ou tecnologia por outra que se adequada a necessidade de cada empresa.

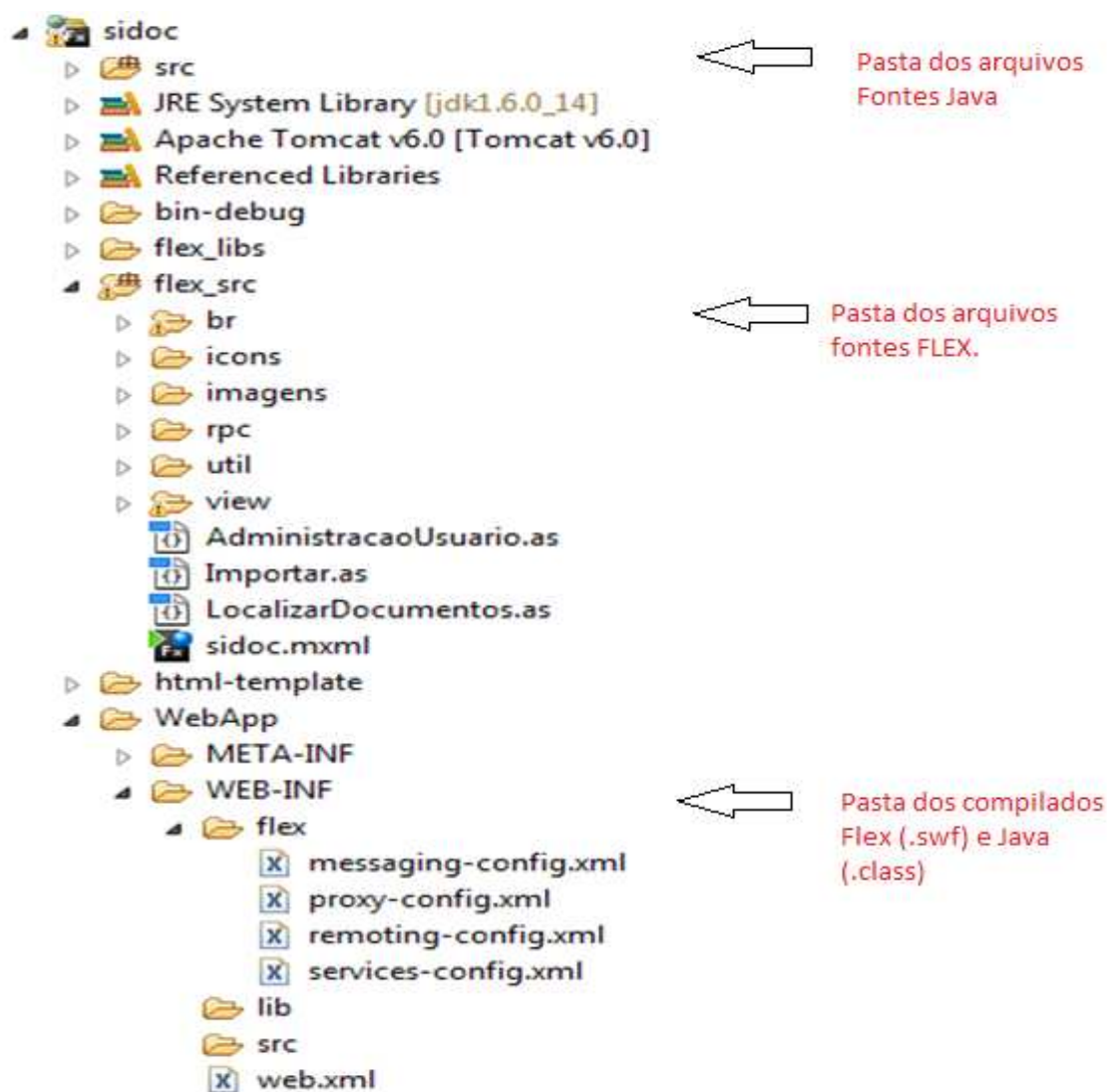


Figura 3.9 – Estrutura do Projeto Sidoc.

Os projetos Java Web são compactados em um arquivo .war, nele tem todas as classes e bibliotecas utilizadas, no caso da integração Java Flex utiliza-se de mecanismo de comunicação, que é o protocolo AMF(Action Message Format) da Adobe. A figura 3.10 abaixo demonstra a configuração do arquivo de serviços para comunicação Java e Flex, esta comunicação deve-se ao conjunto de bibliotecas do BlazeDS que abre um canal de comunicação entre as duas linguagens.

```
<?xml version="1.0" encoding="UTF-8"?>
<service id="remoting-service"
  class="flex.messaging.services.RemotingService">

  <adapters>
    <adapter-definition id="java-object" class="flex.messaging.services.remoting.adapters.JavaAdapter" default="true"/>
  </adapters>

  <default-channels>
    <channel ref="my-amf"/>
  </default-channels>

  <destination id="administracaoService">
    <properties>
      <source>br.com.sidoc.servico.AdministracaoService</source>
    </properties>
  </destination>

  <destination id="gerenciarArquivosService">
    <properties>
      <source>br.com.sidoc.servico.ArquivoService</source>
    </properties>
  </destination>

</service>
```

Figura 3.10 – Descritor de configuração de Serviços Java.

Nos trechos de tag destination possui o campo id que é um código identificador de qual classe Java a ser invocada, o código Flex faz uma chamada remota a serviço de middleware BlazeDS, que tem o canal de comunicação.

3.2 MÓDULOS

3.2.1 Manter Usuários

O módulo Manter Usuário começa quando o usuário clica no menu superior Administração, Usuários, o sistema retorna a tela de pesquisa com o filtro por nome do usuário, departamento ou perfil. Nesta tela o usuário tem as opções de adicionar um novo usuário, alterar ou remover usuários existentes, a figura 3.11 mostra a tela de manter o usuário.

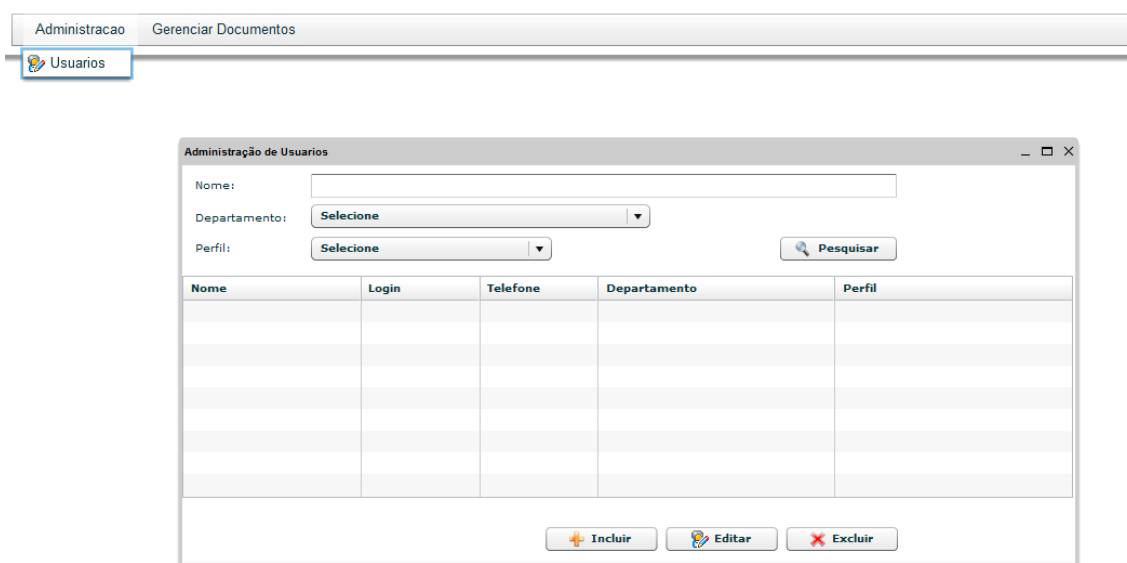


Figura 3.11 – Tela de Manter Usuários.

3.2.2 Manter Documentos

Este módulo é composto por duas telas, uma de cadastro do documento, com opções de filtro como:

- Título
- Assunto
- Descrição
- Tipo do Documento
- Algoritmo de Criptografia
- Url de importação do arquivo

Estes filtros são utilizados para facilitar a busca do documento importado pelo sistema, a figura 3.12 abaixo mostra a tela de importação dos arquivos.

Figura 3.12 – Tela de Importação de Documentos.

Ao importar o arquivo, o sistema mandará a senha de criptografia para o e-mail do usuário ou do departamento.

A localização do arquivo é feita pelos filtros previamente cadastrados na importação, a figura 3.13 mostra este processo.

Nome do Arquivo	Tipo do Documento	Algoritmo de Criptografia

Figura 3.13 – Tela de Localização dos Documentos.

CAPÍTULO 4 – TESTE E RESULTADOS

Este capítulo mostra alguns testes e resultados realizados baseados na solução proposta de desenvolvimento do sistema Sidoc.

4.1.1 TESTE IMPORTAÇÃO

Todo o sistema foi desenvolvido sobre uma arquitetura Java e integrado com a plataforma Flex da Adobe, deste modo demonstraremos como foco principal a criptografia do arquivo armazenado em disco.

A figura 4.1 ilustra a tela de importação dos documentos preenchida e a resposta do sistema para o usuário

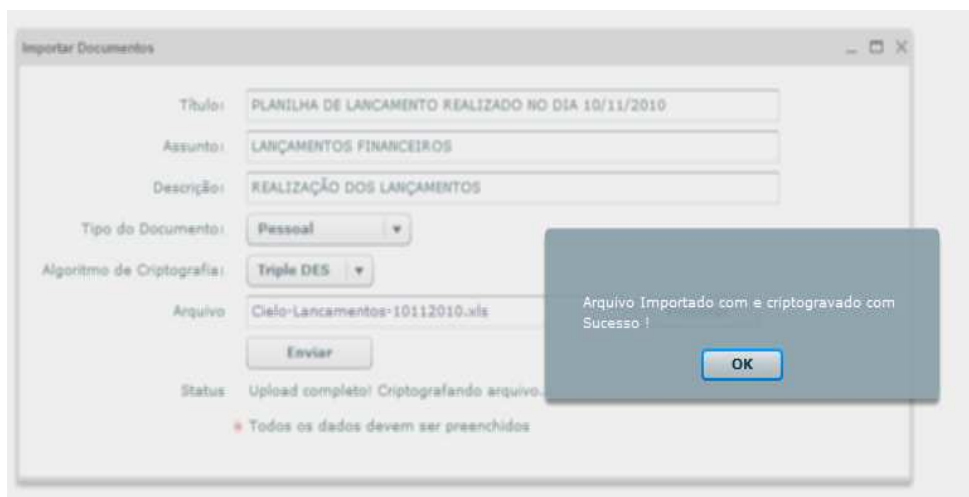


Figura 4.1 – Tela do Arquivo Importado pelo sistema.

4.1.2 RESULTADOS DA IMPORTAÇÃO

Após ser importado o arquivo, o sistema envia um e-mail para o usuário com a senha para descriptografar o documento. Este processo foi utilizado o emulador de e-mail PostCastServer na versão 2.6.0, a figura 4.2 mostra o layout do e-mail.

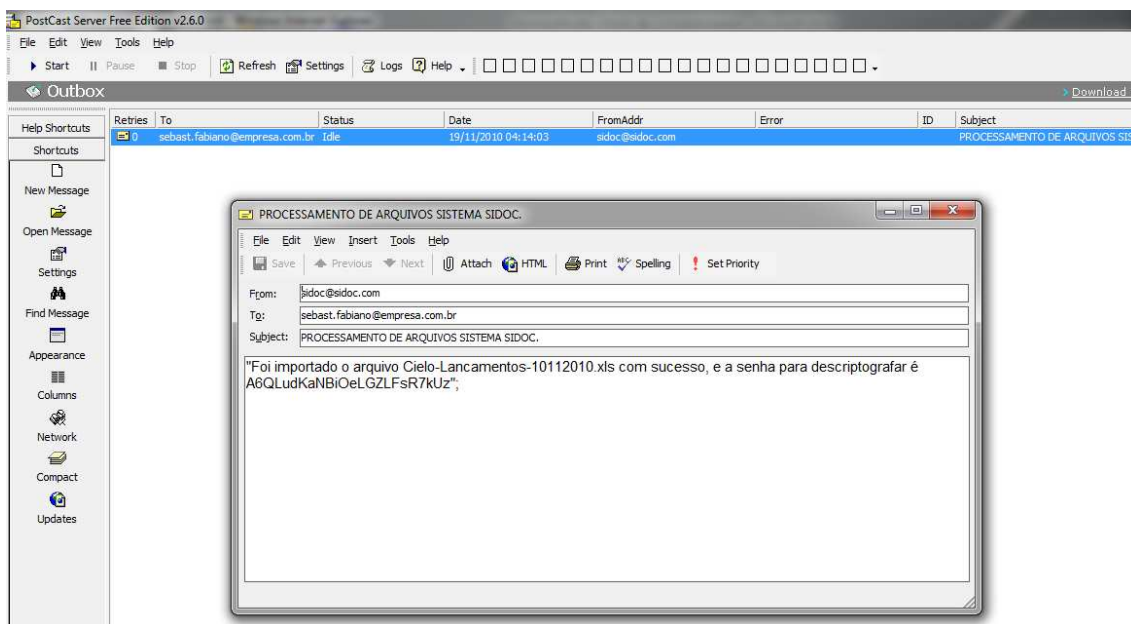


Figura 4.2 – Tela de envio de e-mail.

Analisando o diretório que o sistema armazenou o arquivo, verificamos que o sistema cria uma pasta com o nome do usuário ou do departamento, e na sua raiz ele cria duas pasta. Uma para arquivos criptografados e outra para os arquivos descriptografados.

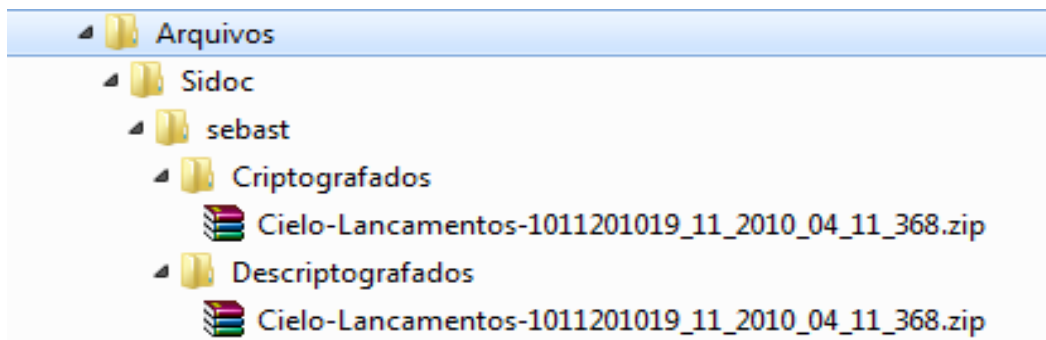


Figura 4.3 – de Armazenamento de Arquivos.

Ao tentar abrir o arquivo criptografado é exibido um erro do sistema operacional, informando que o arquivo está em formato desconhecido ou danificado. Esse erro é proveniente da criptografia do arquivo, pois foram alterados os binários, evitando assim que o arquivo seja aberto.

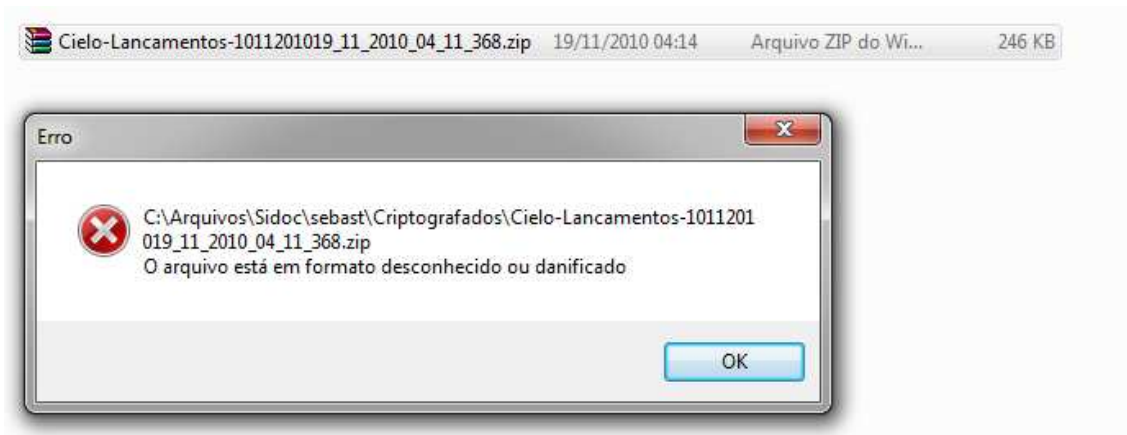


Figura 4.4 – Tentando abrir o Arquivo Criptografado.

4.1.3 TESTE LOCALIZAÇÃO

Na localização dos arquivos importados no sistema, entramos na tela de localização dos documentos. Ela contém os campos de filtros, que possibilita a busca por título, assunto, descrição e tipo do documento.

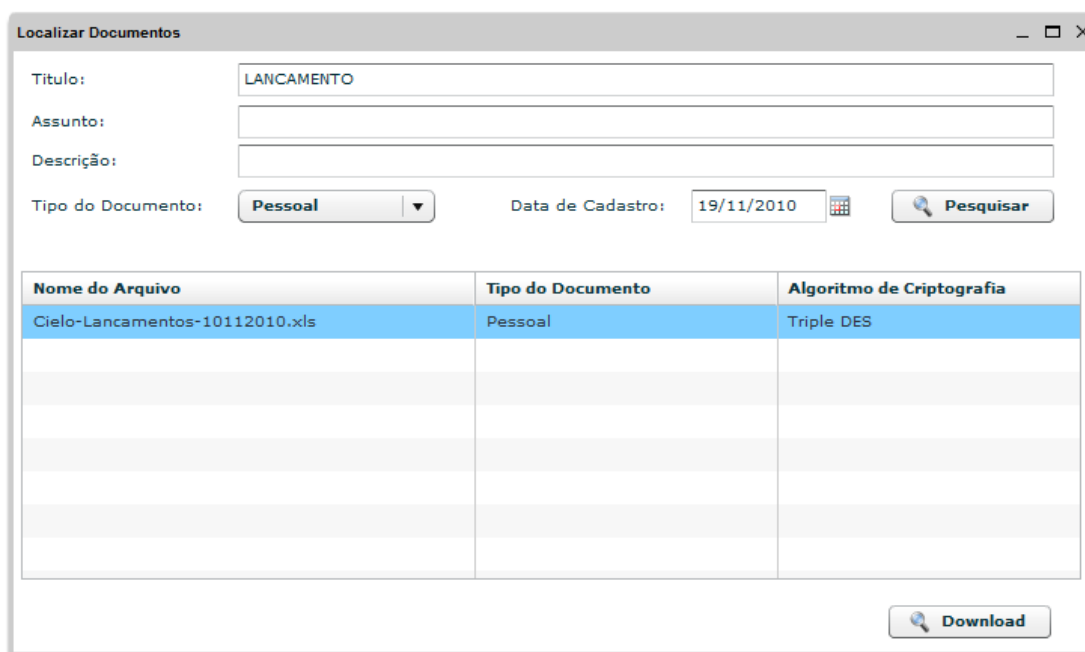


Figura 4.5 – Tela de Localização dos Documentos.

4.1.4 RESULTADO LOCALIZAÇÃO

Após a localização do documento o sistema emite um tela de PopUp para que seja adicionado a senha para descriptografar o arquivo. A figura 4.6 mostra este processo.

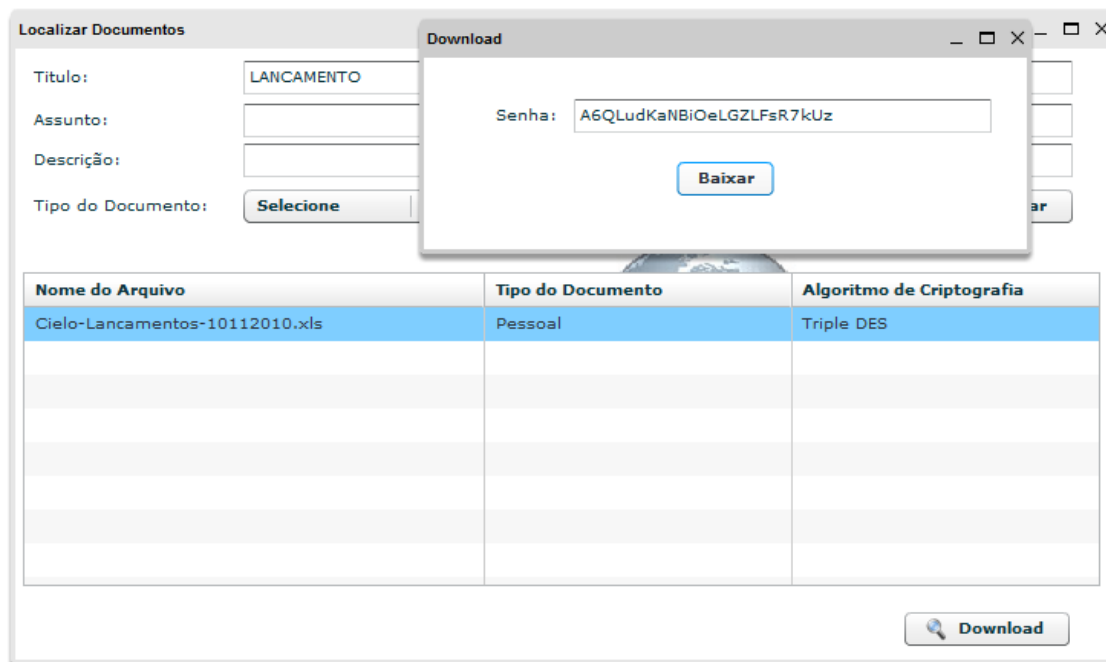


Figura 4.6 – Tela de Download do Documento.

Ao adicionar a senha e clicar no botão baixar, o sistema utiliza a senha para descriptografar o arquivo compactado e descompacta, emitindo para o usuário o arquivo original armazenado.

CAPÍTULO 5 – CONCLUSÃO

O Sidoc é um sistema que surgiu de uma idéia inicial, de se construir uma solução de Gerenciamento Eletrônico de Documentos, mas tomou uma direção a outro contexto de automação, assim passando a ter um foco maior na parte de segurança da informação e tornando o assunto deste trabalho. As tecnologias escolhidas não foram por acaso, pois são tecnologias que tenho conhecimento já faz alguns anos, trabalhando e ministrando cursos, e neste sistema adicionei uma arquitetura que está sendo o diferencial dos projetos web.

Adobe Flex iniciou com as tecnologias RIA (Aplicações Ricas para Internet) e consolidou mais ainda com a integração com a plataforma Java. Evoluindo ao passar do tempo, e hoje tendo integração não somente com Java, mas com PHP. Tentei demonstrar a simplicidade da integração das duas tecnologias e também o resultado final desta integração, e na parte da API de criptografia Java, fica bem transparente para o desenvolvedor implementar soluções robustas, pois esta API faz parte desde a versão do JSDK 1.1, e foi evoluído com o passar dos anos, implementado novos algoritmos mais seguros como é o caso o algoritmo AES, que surgiu em 2001. No caso de sistema para demonstração acadêmica implantei três tipos de algoritmos, AES, Triple DES e DES, caso este sistema torne-se um produto comercial é desnecessário ter este três, sendo satisfatório só o AES, até que seja quebrado substituído por outro logicamente.

Assim o objetivo deste projeto foi de automatizar o processo de manipulação de arquivos eletrônicos e conseguir adicionar uma segurança nas informações armazenadas. No contexto das tecnologias de Gerenciamento Eletrônico de Documentos, este projeto se enquadra, com ressalvas pois o sistema de Gerenciamento Eletrônico de Documentos por si só não soluciona a automação das informações geridas, para ter sucesso é preciso ter um esforço em catalogação e cada empresa tem necessidades distintas, e o GED é mais do que armazenar dados. A criptografia dos documentos geridos pelo sistema foi efetivada com sucesso, tornando assim um sistema arquitetonicamente poderoso, porém podendo se expandir para outros contextos com Workflow e ERP.

5.1 FUTUROS PROJETOS

Para projetos futuros sugiro implementar as seguintes melhorias:

- Implementar o framework lucene para indexação dos arquivos armazenados, este framework é free do projeto Apache.
- Estudar os mecanismos de localização de dados no MySQL, com fulltext, se torna viável.
- Implementação do Módulo de relatório e acompanhamentos.
- Implementar mecanismos de workflow.
- Estudar viabilidade de integração com o framework Spring.
- Estudar performance com Hibernate e JPA.

REFERÊNCIAS BIBLIOGRÁFICAS

MORENO E. D.; PEREIRA F. D.; CHIARAMONTE R. B.; *Criptografia em Software e hardware*. Novatec. 2005.

SANTOS, Antonio Raimundo dos, *Metodologia cinetifica: A **construção do conhecimento***. 3. ed. Rio de Janeiro: DP&A editora, 2000.

STARBIRD, Robert W. e VILHAUER, Gerald C. **Como tomar a decisão de implantar a tecnologia do gerenciamento eletrônico de documentos** . São Paulo: Cenadem, 1998.

RONDINELLI, Rosely Curi. **Gerenciamento Arquivístico de Documentos Eletrônicos: Uma Abordagem Teórica e Diplomática Arquivística Contemporânea**. Rio de Janeiro FGV: 2004, 2.ed

TAPPER, Jaff; MICHEL, Labriola; MATTHEW, Boles; JAMES; Talbot. **Adobe Flex Treinamento - Direto da Fonte**. Rio de Janeiro: Altabooks, 2009.

Sites

ADOBE SYSTEMS INCORPORATED. Adobe Flex. Disponível em <http://www.adobe.com/prodindex/framemaker/prodinfosgml.html>. Acesso em 23/08/2010.

APACHE SOFTWARE FOUNDATION. Apache. Disponível em <http://www.apache.org> .Acesso em 23/08/2010.

ORACLE. Java Techology. Disponível em <http://www.sun.com/java/>. Acesso em 23/08/2010.

TKOTZ, V.: *Criptografia Numaboa*. Acesso em outubro 2010, disponível em: <http://www.numaboa.com/criptografia>.

WIKIPEDIA, Criptografia. Apache. Disponível em <http://pt.wikipedia.org/wiki/Criptografia>. Acesso em 23/08/2010.

WIKIPEDIA, Criptografia. Apache. Disponível em [http://pt.wikipedia.org/wiki/Java_\(linguagem_de_programação\)](http://pt.wikipedia.org/wiki/Java_(linguagem_de_programação)). Acesso em 23/08/2010.

APENDICE I

Código Flex:

Sidoc.mxml

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
                xmlns:flexmdi="http://flexmdi.googlecode.com/"
                layout="absolute" creationComplete="initCollections()"
backgroundImage="imagens/globaldoc.jpg" xmlns:vo="br.com.sidoc.vo.*"
xmlns:ns1="br.com.sidoc.componentes.*">

<mx:Script>
    <![CDATA[
        import view.LoginFormView;
        import br.com.sidoc.vo.Departamento;
        import flexmdi.effects.effectsLib.MDIRelationalEffects;
        import flexmdi.effects.effectsLib.MDILinearEffects;
        import util.RegistraComponentes;
        import flexmdi.effects.effectsLib.MDIVistaEffects;
        import br.com.sidoc.vo.Usuario;
        import flexmdi.containers.MDIWindow;
        import flexmdi.events.MDIWindowEvent;

        import mx.collections.*;
        import mx.controls.Alert;
        import mx.core.Container;
        import mx.events.MenuEvent;
        import mx.events.ModuleEvent;
        import mx.managers.PopUpManager;
        import mx.modules.IModuleInfo;
        import mx.modules.Module;
        import mx.modules.ModuleLoader;
        import mx.modules.ModuleManager;
        import mx.rpc.events.ResultEvent;

        import rpc.RemoteDestination;
        import rpc.RemoteService;

        import util.ImageAssets;

        import view.SplashModule;

        [Bindable]
        public var menuBarCollection:XMLListCollection;

        [Embed(source="imagens/icon/user5-edit.png")]
        public var clienteEdit:Class;

        [Embed(source="imagens/icon/icoPastaAberta.png")]
        public var icoPastaAberta:Class;

        [Embed(source="imagens/icon/icoPastaFechada.png")]
```



```

public var icoPastaFechada:Class;

private function initCollections():void {
    RegistraComponentes.registrar();
    carregarTelaLogin();
}

public function retorno(event:ResultEvent):void{
    //Alert.show(String(event.result),"Alerta de
Exemplo!");
}

private function menuHandler(event:MenuEvent):void {
    if (event.item.@tipo == "finalizar") {
        onFinalizar();
    }
    else{

if(ModuleManager.getModule(event.item.@swf).loaded){
    return;
}

//Mostra a tela de "aguarde"
var splash:SplashModule = new SplashModule;
PopUpManager.addPopUp(splash, this, true);
PopUpManager.centerPopUp(splash);
//Cria a instancia da janela MDI
var janela:MDIWindow = new MDIWindow;

janela.title = event.item.@tituloJanela;
var mdl:ModuleLoader = new ModuleLoader;
//Fecha a janela quando o mão estiver carregado

var loadedModule:DisplayObject;

mdl.addEventListener(ModuleEvent.READY,
function(event:ModuleEvent):void{
    PopUpManager.removePopUp(splash);
    trace(event.currentTarget.child.height);
    janela.height =
event.currentTarget.child.height+30;
    janela.width =
event.currentTarget.child.width+8;

});

mdl.addEventListener(ModuleEvent.PROGRESS,
function(event:ModuleEvent):void{
    var percentual:int =
Math.round((event.bytesLoaded / event.bytesTotal) * 100);
    splash.lbAguarde.text = percentual + "%";
});
mdl.url = event.item.@swf;
mdl.loadModule();

janela.addChild(mdl);

```

```

        janela.addEventListener(MDIWindowEvent.CLOSE,
function(event:Event):void{
            mdl.unloadModule();
        });

        mdiCanvas.windowManager.center(janela);
        mdiCanvas.windowManager.add(janela);

    }

}

public function sair(event:MouseEvent):void{
    mdiCanvas.removeAllChildren();
    carregarTelaLogin();
}

private function onFinalizar():void{
    mdiCanvas.removeAllChildren();
}

private function carregarTelaLogin():void{
    var telaLogin:LoginFormView = new
LoginFormView;

    PopUpManager.addPopUp(telaLogin, this, true);
    PopUpManager.centerPopUp(telaLogin);
}

]]>

</mx:Script>

<vo:Usuario id="usuario" />

    <mx:XMLList id="menuXML">
        <menuitem label="Administração" >
            <menuitem label="Usuarios" tipo="filho"
tituloJanela="Administração de Usuarios"
swf="AdministracaoUsuario.swf" icone="clienteEdit"/>
        </menuitem>
        <menuitem label="Gerenciar Documentos" >
            <menuitem label="Importar Documentos"
tipo="filho" tituloJanela="Importar Documentos" swf="Importar.swf"
icone="icoPastaFechada"/>
            <menuitem label="Localizar Documentos"
tipo="filho" tituloJanela="Localizar Documentos"
swf="LocalizarDocumentos.swf" icone="icoPastaAberta"/>
        </menuitem>
    </mx:XMLList>

    <mx:VBox id="vbox" width="100%" height="100%">

        <mx:ApplicationControlBar y="0" width="100%" height="46"
visible="true" id="pnMenu"
horizontalCenter="0"
fillColors="[#B7BABC, #B7BABC]" fontFamily="Arial" fontSize="12">

```

```

        <mx:MenuBar labelField="@label"
itemClick="menuHandler(event);" iconField="@icone"
                dataProvider="{menuXML}"
width="95%" height="30" fontFamily="Arial" fontSize="13">

        </mx:MenuBar>
        <!--
        <mx:MenuBar labelField="@label"
itemClick="menuHandler(event);" iconField="@icone"
                dataProvider="{menuXML}"
width="95%" height="30" visible="true" fontFamily="Arial"
fontSize="13" alpha="3">

        </mx:MenuBar>
        -->

        <mx:Text y="138" id="usuarioLogado"
text="{usuario.login}" width="89"    fontSize="13" fontFamily="Arial" />
        <mx:Image source="imagens/icon/Sair24.png"
height="24" width="24" click="sair(event)" />
        </mx:ApplicationControlBar>

        <flexmdi:MDICanvas id="mdiCanvas" width="100%"
height="100%" minWidth="100"

        effectsLib="{flexmdi.effects.effectsLib.MDIVistaEffects}" >

        <mx:DataGrid x="203" y="84" visible="false">

        </mx:DataGrid>
        <mx:ComboBox id="Combo" visible="false" />
        <mx:CheckBox visible="false" />
        <mx:FormItem visible="false" />
        <ns1:DataField x="435" y="185" visible="false" />
        </flexmdi:MDICanvas>

    </mx:VBox>

</mx:Application>

```

LoginFormView.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<mx:TitleWindow xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" width="334" height="196" title="Acesso ao Sistema"
cornerRadius="0" backgroundAlpha="1.0"
xmlns:ns1="br.com.cursoflex.componentes.*">

    <mx:Script>
        <![CDATA[
            import br.com.sidoc.vo.Usuario;
            import mx.managers.PopUpManager;
            import mx.rpc.events.ResultEvent;
            import mx.controls.Alert;
            import mx.utils.StringUtil;
            import util.ImageAssets;

```

```

import mx.rpc.events.ResultEvent;
import rpc.RemoteDestination;
import rpc.RemoteService;

private function onClickBtnConfirmar():void{

    new
RemoteService(RemoteDestination.ADMINISTRACAO_SERVICE).call("pesquisar
Login",

onClickBtnConfirmarResult, null, txtLogin.text, txtSenha.text);

}

private function
onClickBtnConfirmarResult(event:ResultEvent):void{
    var usuarioLogado:Usuario = event.result as
Usuario;

    if(usuarioLogado!=null &&
usuarioLogado.login!="" && usuarioLogado.senha!=null){
        this.parentApplication.usuario =
usuarioLogado;

        PopUpManager.removePopUp(this);
    }else{
        Alert.show("Login ou Senha inválida!");
    }
}

}]>
</mx:Script>

<mx:Label x="35.5" y="34" text="Login:" />
<mx:Label x="35.5" y="74" text="Senha:" />
<mx:TextInput x="86.5" y="30" width="174" id="txtLogin"
maxChars="10" />
<mx:TextInput x="86.5" y="70" width="174" id="txtSenha"
maxChars="10" displayAsPassword="true" />
<mx:Button click="onClickBtnConfirmar()" label="Confirmar"
icon="{ImageAssets.iconConfirmar}" x="106.5" y="113" />
</mx:TitleWindow>

```

Modulo.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<mx:ModuleLoader xmlns:mx="http://www.adobe.com/2006/mxml"
width="252" height="110"
horizontalAlign="center"
verticalAlign="middle"
ready="readyHandler(event)"
loading="this.loading.visible = false">
<mx:Script>
    <![CDATA[
        import mx.controls.Alert;
        private function readyHandler(event:Event):void{
            event.target.removeChild(loading);
        }
    ]]>

```

```

    }
    ]]>
</mx:Script>

<mx:Label id="loading" text="CARREGANDO" />

</mx:ModuleLoader>

```

SplashModule.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<mx:TitleWindow xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" width="320" height="102" borderStyle="outset"
cornerRadius="0" alpha="1.0">
    <mx:ProgressBar x="0" y="27" indeterminate="false"
labelPlacement="bottom" width="296" height="5" mode="manual"
minimum="0" maximum="100" id="prbBarra" label="Carregando"/>
    <mx:Label x="118" y="0" id="lbAguarde" text="Aguarde"
fontFamily="Verdana" fontSize="12" fontWeight="bold"/>
</mx:TitleWindow>

```

ListarUsuarioView.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Module xmlns:mx="http://www.adobe.com/2006/mxml"
xmlns:flexmdi="http://flexmdi.googlecode.com/"
layout="absolute" width="824" height="366">

<mx:Script>
    <![CDATA[
        import util.ImageAssets;

    ]]>
</mx:Script>

<mx:VBox verticalGap="0" width="100%" height="326.9"
horizontalAlign="center">

    <mx:Canvas x="88" y="62" width="100%" height="102">

        <mx:Label x="10" y="10" text="Nome:" />

        <mx:TextInput x="119" y="8" width="542" id="txtNome" />
        <mx:Button x="552.9" y="66" id="btnPesquisar"
label="Pesquisar" icon="{ImageAssets.iconFiltrar}" width="108.1"
height="22.045454" />
        <mx:ComboBox y="36" width="314" id="cmbDepartamento"
prompt="Selecione" labelField="nome" x="119" editable="false" />
        <mx:ComboBox y="66" width="223" id="cmbPerfil"
prompt="Selecione" labelField="nome" x="119" editable="false" />
        <mx:Label x="10" y="40" text="Departamento:" width="101" />
        <mx:Label x="10" y="68" text="Perfil:" width="58" />

    </mx:Canvas>

```

```

        <mx:DataGrid width="100%" height="205"
doubleClickEnabled="true" id="gridUsuario" >
            <mx:columns>
                <mx:DataGridColumn headerText="Nome" dataField="nome"
width="30" />
                <mx:DataGridColumn headerText="Login"
dataField="login" width="20" />
                <mx:DataGridColumn headerText="Telefone"
dataField="telefone" width="20" />
                <mx:DataGridColumn headerText="Departamento"
dataField="departamento.nome" width="40" />
                <mx:DataGridColumn headerText="Perfil"
dataField="perfil.nome" width="40" />
            </mx:columns>
        </mx:DataGrid>
    </mx:VBox>
    <mx:Button label="Incluir" width="103.25" id="btnNovo"
icon="{ImageAssets.iconNovo}" x="336.25" y="334.05" height="21.95" />
    <mx:Button label="Editar" width="103.25" id="btnEditar"
icon="{ImageAssets.iconEdit}" x="447.5" y="334.05" height="21.95" />
    <mx:Button label="Excluir" width="103.25" id="btnExcluir"
icon="{ImageAssets.iconExcluir}" x="558.75" y="334.05"
height="21.95" />
</mx:Module>

```

ManterUsuarioView.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<MDIWindow xmlns="flexmdi.containers.*"
xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" width="562" height="332" title="Manter
Usuário">
    <mx:Button x="243" y="272" label="Salvar" id="btnSalvar" />
    <mx:Label x="49" y="87" text="Nome:&#xd;" width="90" height="22"
textAlign="right" />
    <mx:Label x="49" y="53" text="Perfil:" width="90" height="22"
textAlign="right" />
    <mx:Label x="49" y="19" text="Departamento:" width="90"
height="22" textAlign="right" />
    <mx:ComboBox y="18" width="341" id="cmbDepartamento"
labelField="nome" x="156" editable="false" />
    <mx:TextInput x="156" y="86" width="341" height="22"
id="txtNome" />
    <mx:Label x="49" y="121" text="Email:&#xd;" width="90"
height="22" textAlign="right" />
    <mx:TextInput x="156" y="120" width="342" height="22"
id="txtEmail" />
    <mx:TextInput x="156" y="155" width="102" height="22"
maxChars="12" restrict="0-9" id="txtTelefone" />
    <mx:TextInput x="155" y="189" width="134" height="22"
id="txtLogin" />
    <mx:Label x="48" y="189" text="Login:&#xd;" width="90"
height="22" textAlign="right" />
    <mx:TextInput x="155" y="227" width="134" height="22"
id="txtSenha" displayAsPassword="true" />
    <mx:Label x="48" y="227" text="Senha:&#xd;" width="90"
height="22" textAlign="right" />
    <mx:Label x="49" y="155" text="Telefone:" width="90" height="22"
textAlign="right" />

```

```

        <mx:ComboBox x="156" y="52" width="211" id="cmbPerfil"
        labelField="nome" editable="false"/>

</MDIWindow>

```

DataField.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<mx.DateField xmlns:mx="http://www.adobe.com/2006/mxml"
    dayNames="{dias}"
    monthNames="{meses}"
    formatString="DD/MM/YYYY"
    creationComplete="configApp();"
    yearNavigationEnabled="true"
    editable="true"
    restrict="0-9"
    width="94"
    valueCommit="validar(event);">

    <mx:Script>
        <![CDATA[
            import mx.events.FlexEvent;

            private function configApp():void {
                // this.selectedDate = new Date();
            }

            private function validar(event:FlexEvent):void {
                var str:String = event.target.text as String;

                if ( str.charAt(2) != "/" || str.charAt(5) != "/" ) {
                    var mask:String;

                    if ( str.length == 6 ) {
                        mask = str.substr(0, 2) + "/" +
str.substr(2,2) + "/" + str.substr(4, 4);
                        event.target.text = mask;
                    } else if ( str.length == 8 ) {
                        mask = str.substr(0, 2) + "/" +
str.substr(2,2) + "/" + str.substr(4, 4);
                        event.target.text = mask;
                    }
                }
            }
        ]]>
    </mx:Script>

    <mx:Array id="dias">
        <mx:String>D</mx:String>
        <mx:String>S</mx:String>
        <mx:String>T</mx:String>
        <mx:String>Q</mx:String>
        <mx:String>Q</mx:String>
        <mx:String>S</mx:String>
        <mx:String>S</mx:String>
    </mx:Array>

    <mx:Array id="meses">

```

```

<mx:String>Janeiro</mx:String>
<mx:String>Fevereiro</mx:String>
<mx:String>Março</mx:String>
<mx:String>Abril</mx:String>
<mx:String>Maio</mx:String>
<mx:String>Junho</mx:String>
<mx:String>Julho</mx:String>
<mx:String>Agosto</mx:String>
<mx:String>Setembro</mx:String>
<mx:String>Outubro</mx:String>
<mx:String>Novembro</mx:String>
<mx:String>Dezembro</mx:String>
</mx:Array>

```

```
</mx:DateField>
```

DownloadArquivoView.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<MDIWindow xmlns="flexmdi.containers.*"
xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute" width="520" height="154" title="Download">

    <mx:Button x="225.5" y="83" label="Baixar" id="btnBaixar"/>
    <mx:Label x="10" y="29" text="Senha:" width="65" height="22"
textAlign="right"/>
    <mx:TextInput x="83" y="27" width="273" height="22"
id="txtSenha" />
    <mx:Label id="uploadStatus" visible="false" text="" x="98"
y="57"/>
    <mx:Button x="364" y="27" label="Descriptografar" width="120"
id="btnDescriptar"/>
</MDIWindow>

```

ImportaView.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Module xmlns:mx="http://www.adobe.com/2006/mxml" layout="vertical"
width="704" height="278">
    <mx:Script>
        <![CDATA[

            ]]>
    </mx:Script>

    <mx:Form width="100%" height="100%">
        <mx:FormItem label="Título:" width="589">
            <mx:TextInput id="txtTitulo"
toolTip="Título é Obrigatório!" width="423" maxChars="200"/>
        </mx:FormItem>
        <mx:FormItem label="Assunto:" width="589">
            <mx:TextInput id="txtAssunto"
toolTip="Assunto é Obrigatório!" width="423" maxChars="200"/>
        </mx:FormItem>
        <mx:FormItem label="Descrição:" width="589">
            <mx:TextInput id="txtDescricao"
toolTip="Descrição é Obrigatório!" width="423" maxChars="200"/>
        </mx:FormItem>
    </mx:Form>

```



```

width="336">
    <mx:FormItem label="Tipo do Documento:"
    <mx:ComboBox id="cmbTipoDocumento"
toolTip="Tipo do Documento é Obrigatório!">
    <mx:ArrayCollection>
        <mx:String>Selecione</mx:String>
        <mx:String>Pessoal</mx:String>
        <mx:String>Departamental</mx:String>
    </mx:ArrayCollection>
    </mx:ComboBox>
</mx:FormItem>
    <mx:FormItem label="Algoritmo de Criptografia:"
width="328">
    <mx:ComboBox id="cmbAlgoritmo"
toolTip="Algoritmo de Criptografia é Obrigatório!">
    <mx:ArrayCollection>
        <mx:String>Selecione</mx:String>
        <mx:String>AES</mx:String>
        <mx:String>Triple DES</mx:String>
        <mx:String>DES</mx:String>
    </mx:ArrayCollection>
    </mx:ComboBox>
</mx:FormItem>

    <mx:FormItem label="Arquivo" direction="horizontal">
        <mx:TextInput id="fileLocation"
toolTip="Arquivo é Obrigatório!" width="300" editable="false"
color="#001234"/>
        <mx:Button id="browseButton" label="Procurar"
width="100"/>
    </mx:FormItem>
    <mx:FormItem label="">
        <mx:Button id="uploadButton" label="Enviar"
width="100"/>
    </mx:FormItem>
    <mx:FormItem label="Status">
        <mx:Label id="uploadStatus" text="Selecione um
arquivo para fazer o upload"/>
    </mx:FormItem>
    <mx:FormItem required="true">
        <mx:Label text="Todos os dados devem ser
preenchidos"/>
    </mx:FormItem>
</mx:Form>
</mx:Module>

```

LocalizarDocumentosView.mxml

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Module xmlns:mx="http://www.adobe.com/2006/mxml"
    xmlns:flexmdi="http://flexmdi.googlecode.com/"
    layout="absolute" width="714" height="398"
xmlns:ns1="br.com.sidoc.componentes.*"
xmlns:control="br.com.thinklab.util.control.*">

<mx:Script>

```

```

<![CDATA[
    import util.ImageAssets;

    private function tratarDados(item:Object,
coluna:DataGridColumn):String{
        if(coluna.dataField=="tipo" && item.tipo != null){
            if(item.tipo == 1){
                return "Pessoal";
            }else{
                return "Departamental";
            }
        }else if(coluna.dataField=="algoritmo" &&
item.algoritmo != null){
            if(item.algoritmo == 1){
                return "AES";
            }else if(item.algoritmo == 2){
                return "Triple DES";
            }else if(item.algoritmo == 3){
                return "DES";
            }
        }
        return "";
    }
}]>
</mx:Script>

<mx:VBox verticalGap="0" width="100%" height="360"
horizontalAlign="center">

    <mx:Canvas x="88" y="62" width="100%" height="146">

        <mx:Label x="10" y="10" text="Titulo:" width="69"/>

        <mx:TextInput x="149" y="8" width="542" id="txtTitulo"/>
        <mx:Button x="582.9" y="92" id="btnPesquisar"
label="Pesquisar" icon="{ImageAssets.iconFiltrar}" width="108.1"
height="22.045454"/>
        <mx:Label x="10" y="38" text="Assunto:" width="69"/>
        <mx:TextInput x="149" y="36" width="542" id="txtAssunto"/>
        <mx:Label x="10" y="64" text="Descrição:" width="69"/>
        <mx:Label x="10" y="94" text="Tipo do Documento:"
width="131"/>
        <mx:TextInput x="149" y="62" width="542"
id="txtDescricao"/>

        <mx:ComboBox id="cmbTipoDocumento" x="149" y="92">
            <mx:ArrayCollection>
                <mx:String>Selecione</mx:String>
                <mx:String>Pessoal</mx:String>
                <mx:String>Departamental</mx:String>
            </mx:ArrayCollection>
        </mx:ComboBox>
        <mx:Label x="328" y="94" text="Data de Cadastro:"
width="114"/>
        <ns1:DataField x="450" y="92" width="106"
id="dataCadastro"/>

    </mx:Canvas>

```

```

        <mx:DataGrid width="704" height="205"
doubleClickEnabled="true" id="gridArquivo" >
            <mx:columns>
                <mx:DataGridColumn headerText="Nome do Arquivo"
dataField="arquivo.nomeOriginal" width="30"/>
                <mx:DataGridColumn headerText="Tipo do Documento"
dataField="tipo" width="20" labelFunction="tratarDados"/>
                <mx:DataGridColumn headerText="Algoritmo de
Criptografia" dataField="algoritmo" width="20"
labelFunction="tratarDados"/>
                <mx:DataGridColumn headerText="Login Usuário"
dataField="arquivo.usuario.login" width="20"/>
            </mx:columns>
        </mx:DataGrid>
    </mx:VBox>
    <mx:Button id="btnDownload" label="Verificar"
icon="{ImageAssets.iconFiltrar}" width="108.1" height="22.045454"
x="580.9" y="368"/>
</mx:Module>

```

LocalizarDocumentos.as

```

package
{
    import br.com.sidoc.documentos.DownloadArquivo;
    import br.com.sidoc.documentos.LocalizarDocumentosView;
    import br.com.sidoc.vo.DetalheArquivo;
    import br.com.sidoc.vo.Usuario;

    import flash.events.Event;
    import flash.events.MouseEvent;

    import mx.collections.ArrayCollection;
    import mx.controls.Alert;
    import mx.events.FlexEvent;
    import mx.rpc.events.ResultEvent;

    import rpc.RemoteDestination;
    import rpc.RemoteService;

    public class LocalizarDocumentos extends LocalizarDocumentosView
    {
        public function LocalizarDocumentos()
        {
            addEventListener(FlexEvent.CREATION_COMPLETE,
this.init);
        }

        private function init(event:FlexEvent):void{
            btnPesquisar.addEventListener(MouseEvent.CLICK,
localizarDocumentos);

            btnDownload.addEventListener(MouseEvent.CLICK,download);

            gridArquivo.addEventListener(MouseEvent.DOUBLE_CLICK,download);

        }
        private function download(event:Event):void{
            if(gridArquivo.selectedItem != null){

```



```

import br.com.sidoc.vo.Arquivo;
import br.com.sidoc.vo.DetalheArquivo;
import br.com.sidoc.vo.Usuario;

import flash.events.Event;
import flash.events.MouseEvent;
import flash.events.ProgressEvent;
import flash.net.FileReference;
import flash.net.URLRequest;
import flash.net.URLRequestMethod;
import flash.net.URLVariables;

import mx.controls.Alert;
import mx.events.FlexEvent;
import mx.rpc.events.ResultEvent;

import rpc.RemoteDestination;
import rpc.RemoteService;

public class Importar extends ImportaView
{
    private var fileRef:FileReference=new FileReference();
    private var percentLoaded:int;

    public function Importar()
    {
        addEventListener(FlexEvent.CREATION_COMPLETE,
this.init);
    }

    private function init(event:FlexEvent):void{

        browseButton.addEventListener(MouseEvent.CLICK,fileBrowse);
        uploadButton.addEventListener(MouseEvent.CLICK,
fileUpload);
    }

    private function fileBrowse(event:MouseEvent):void
    {
        fileRef.addEventListener(Event.SELECT,
onSelect);

        fileRef.addEventListener(ProgressEvent.PROGRESS,
progressHandler);
        fileRef.addEventListener(Event.COMPLETE,
criptografa);

        uploadStatus.text="Procurando o arquivo";
        fileLocation.text="";
        fileRef.browse();
    }

    private function criptografa(event:Event):void
    {
        if(percentLoaded == 100){
            uploadStatus.text = "Upload completo!
Criptografando arquivo..."
        }

        var usuario:Number =
this.parentApplication.usuario.idUsuario;

```

```

DetalheArquivo();
    var detalhe:DetalheArquivo = new
    var arq:Arquivo = new Arquivo();
    arq.nomeOriginal = fileRef.name;
    arq.extensao = fileRef.type;
    arq.usuario = parentApplication.usuario as

Usuario;

    detalhe.algoritmo = cmbAlgoritmo.selectedIndex;
    detalhe.tipo = cmbTipoDocumento.selectedIndex;
    detalhe.assunto = txtAssunto.text;
    detalhe.titulo = txtTitulo.text;
    detalhe.descricao = txtDescricao.text;
    detalhe.dataCadastro = new Date();

    detalhe.arquivo = arq;
    new
RemoteService(RemoteDestination.ARQUIVO_SERVICE).call("importarArquivo
",retornoImportacao,null,detalhe);
    }

    private function
retornoImportacao(event:ResultEvent):void{
    var retorno:String = event.result as String;
    if(retorno == ""){
        Alert.show(retorno);

    }else{
        Alert.show(retorno);
    }
}

    private function onSelect(event:Event):void
{
    uploadStatus.text="Arquivo selecionado";
    fileLocation.text=fileRef.name;
}

    private function
progressHandler(event:ProgressEvent):void
{
    percentLoaded = int((event.bytesLoaded /
event.bytesTotal) * 100);
    uploadStatus.text="Arquivo " + percentLoaded +
"% completado";
}

    private function fileUpload(event:MouseEvent):void
{
    if(txtTitulo.text == ""){
        Alert.show("O Título é de preenchimento
obrigatório !");
        return;
    }else if(txtAssunto.text == ""){
        Alert.show("O Assunto é de preenchimento
obrigatório !");
        return;
    }else if(txtDescricao.text == ""){
        Alert.show("A Descrição do Documento é de
preenchimento obrigatório !");
        return;
}

```

```

        }else if(cmbTipoDocumento.selectedLabel ==
"Selecione"){
            Alert.show("O Tipo do Documento é de
preencimento obrigatório !");
            return;
        }else if(cmbAlgoritmo.selectedLabel ==
"Selecione"){
            Alert.show("O Tipo de Algoritmo é de
preencimento obrigatório !");
            return;
        }

        uploadStatus.text="Enviando...";
        var menuURLVars:URLVariables=new
URLVariables();
        var usuario:Usuario = parentApplication.usuario
as Usuario;
        var servletTarget:URLRequest=null;
        if(cmbTipoDocumento.selectedIndex == 1){
            servletTarget = new
URLRequest("/sidoc/ArquivoUpload?loginUsuario="+usuario.login);
        }else{
            servletTarget = new
URLRequest("/sidoc/ArquivoUpload?loginUsuario="+usuario.departamento.s
igla);
        }
        servletTarget.method= URLRequestMethod.POST;
        if( fileRef.type == ".xls" || fileRef.type ==
".csv"
|| fileRef.type == ".doc" || fileRef.type
== ".docx"
|| fileRef.type == ".rtf" || fileRef.type
== ".pdf"
|| fileRef.type == ".ppt" || fileRef.type
== ".ppt"){
            fileRef.upload(servletTarget);
        }else{
            Alert.show(" O tipo do arquivo não pode ser
importado !");
        }
    }
}

```

AdministracaoUsuario.as

```

package
{
    import br.com.sidoc.administracao.ListarUsuarioView;
    import br.com.sidoc.administracao.ManterUsuario;
    import br.com.sidoc.vo.Departamento;
    import br.com.sidoc.vo.Perfil;
    import br.com.sidoc.vo.Usuario;

    import flash.events.MouseEvent;

    import mx.collections.ArrayCollection;
    import mx.controls.Alert;
    import mx.events.CloseEvent;
}

```

```

import mx.events.FlexEvent;
import mx.managers.PopUpManager;
import mx.rpc.events.ResultEvent;

import rpc.RemoteDestination;
import rpc.RemoteService;

public class AdministracaoUsuario extends ListarUsuarioView
{
    [Bindable]
    private var clientes:ArrayCollection = new
ArrayCollection;

    public function AdministracaoUsuario()
    {
        addEventListener(FlexEvent.CREATION_COMPLETE,
this.init);
    }

    private function init(event:FlexEvent):void{

        btnNovo.addEventListener(MouseEvent.CLICK,criarUsuario);

        btnPesquisar.addEventListener(MouseEvent.CLICK,pesquisarUsuario);
        btnExcluir.addEventListener(MouseEvent.CLICK, excluir);

        gridUsuario.addEventListener(MouseEvent.DOUBLE_CLICK,editarUsuar
io);

        listarCombos();

    }

    private function listarCombos():void{
        new
RemoteService(RemoteDestination.ADMINISTRACAO_SERVICE).call("listarDep
artamento",retornoDepartamentos);
        new
RemoteService(RemoteDestination.ADMINISTRACAO_SERVICE).call("listarPer
fil",retornoPerfiles);
    }

    private function retornoDepartamentos(event:ResultEvent):void{
        cmbDepartamento.dataProvider = event.result as
ArrayCollection;
    }

    private function retornoPerfiles(event:ResultEvent):void{
        cmbPerfil.dataProvider = event.result as ArrayCollection;
    }

    private function pesquisarUsuario(event:MouseEvent):void{
        var service:RemoteService = new
RemoteService(RemoteDestination.ADMINISTRACAO_SERVICE);
        var nome:String = txtNome.text;
        var idDepartamento:Number;
        var idPerfil:Number;
        if(cmbDepartamento.selectedItem!=null){

```



```

        idDepartamento = (cmbDepartamento.selectedItem as
Departamento).idDepartamento;
    }
    if(cmbPerfil.selectedItem !=null ){
        idPerfil = (cmbPerfil.selectedItem as
Perfil).idPerfil;
    }

service.call("pesquisar",retornoPesquisaUsuario,null,nome,idPerfil,idD
epartamento);
    }

    private function
retornoPesquisaUsuario(ev:ResultEvent):void{
        gridUsuario.dataProvider = ev.result as
ArrayCollection;
    }

    private function carregaUsuarios():void{
        var service:RemoteService = new
RemoteService(RemoteDestination.ADMINISTRACAO_SERVICE);
        service.call("listarUsuarios",retornoCarregaUsurios);
    }

    private function
retornoCarregaUsurios(event:ResultEvent):void{
        gridUsuario.dataProvider = event.result as
ArrayCollection;
    }

    private function onClose():void{
        PopUpManager.removePopUp(this);
    }

    private function criarUsuario(event:MouseEvent):void{
        var manterUsuario:ManterUsuario = new ManterUsuario();
        var app:sidoc = (this.parentApplication as sidoc);
        app.mdiCanvas.windowManager.add(manterUsuario);
    }

    private function editarUsuario(event:MouseEvent):void{
        var manterUsuario:ManterUsuario = new
ManterUsuario();
        var usuarioGrid:Usuario = gridUsuario.selectedItem as
Usuario;
        manterUsuario.usuario = usuarioGrid;
        var app:sidoc = (this.parentApplication as sidoc);
        app.mdiCanvas.windowManager.add(manterUsuario);
    }

    private function excluir(event:MouseEvent):void{
        Alert.yesLabel='Sim';
        Alert.noLabel='Não';
        if(gridUsuario.selectedItem != null )
            Alert.show('Deseja realmente
excluir?','ATENÇÃO',Alert.YES|Alert.NO, null, excluirUsuario);
    }

    private function excluirUsuario(e:CloseEvent):void{

```

```

        var id:Number = (gridUsuario.selectedItem as
Usuario).idUsuario;
        if(e.detail==Alert.YES){
            new
RemoteService(RemoteDestination.ADMINISTRACAO_SERVICE).call("deletarUs
uario",retornoExcluir,null,id);
        }
    }
    private function retornoExcluir(event:ResultEvent):void{
        Alert.show("Dados Excluído com sucesso!");
        var service:RemoteService = new
RemoteService(RemoteDestination.ADMINISTRACAO_SERVICE);
        var nome:String = txtNome.text;
        var idDepartamento:Number;
        var idPerfil:Number;
        if(cmbDepartamento.selectedItem!=null){
            idDepartamento =
(cmbDepartamento.selectedItem as Departamento).idDepartamento;
        }
        if(cmbPerfil.selectedItem){
            idPerfil = (cmbPerfil.selectedItem
as Perfil).idPerfil;
        }
        service.call("pesquisar",retornoPesquisaUsuario,null,nome,idPerfil,idD
epartamento);
    }
}
}
}

```

Arquivo.as

```

package br.com.sidoc.vo
{
    [Bindable]
    [RemoteClass(alias="br.com.sidoc.vo.Arquivo")]
    public class Arquivo
    {
        public function Arquivo(){}

        public var idArquivo:Number;
        public var usuario:Usuario;
        public var nomeOriginal:String;
        public var extensao:String;
        public var nomeCompactado:String;
        public var localArmazenado:String;
    }
}

```

Departamento.as

```

package br.com.sidoc.vo
{
    [Bindable]
    [RemoteClass(alias="br.com.sidoc.vo.Departamento")]
    public class Departamento

```

```

    {
        public function Departamento() {}

        public var idDepartamento: Number;
        public var nome: String;
        public var sigla: String;
    }
}

```

DetalheArquivo.as

```

package br.com.sidoc.vo
{
    [Bindable]
    [RemoteClass(alias="br.com.sidoc.vo.DetalheArquivo")]
    public class DetalheArquivo
    {
        public function DetalheArquivo() {}

        public var idDetalhe: Number;
        public var arquivo: Arquivo;
        public var titulo: String;
        public var assunto: String;
        public var descricao: String;
        public var tipo: Number;
        public var algoritmo: Number;
        public var dataCadastro: Date;
        public var senha: String;
    }
}

```

Perfil.as

```

package br.com.sidoc.vo
{
    [Bindable]
    [RemoteClass(alias="br.com.sidoc.vo.DetalheArquivo")]
    public class DetalheArquivo
    {
        public function DetalheArquivo() {}

        public var idDetalhe: Number;
        public var arquivo: Arquivo;
        public var titulo: String;
        public var assunto: String;
        public var descricao: String;
        public var tipo: Number;
        public var algoritmo: Number;
        public var dataCadastro: Date;
        public var senha: String;
    }
}

```

```
}
```

Usuario.as

```
package br.com.sidoc.vo
{
    [Bindable]
    [RemoteClass(alias="br.com.sidoc.vo.Usuario")]
    public class Usuario
    {
        public function Usuario(){}

        public var idUsuario:Number;
        public var departamento:Departamento;
        public var perfil:Perfil;
        public var nome:String;
        public var login:String;
        public var senha:String;
        public var telefone:String;
        public var email:String;

    }
}
```

ManterUsuario.as

```
package br.com.sidoc.vo
{
    [Bindable]
    [RemoteClass(alias="br.com.sidoc.vo.Usuario")]
    public class Usuario
    {
        public function Usuario(){}

        public var idUsuario:Number;
        public var departamento:Departamento;
        public var perfil:Perfil;
        public var nome:String;
        public var login:String;
        public var senha:String;
        public var telefone:String;
        public var email:String;

    }
}
```

DownloadArquivo.as

```
package br.com.sidoc.vo
{
    [Bindable]
    [RemoteClass(alias="br.com.sidoc.vo.Usuario")]
    public class Usuario
    {
        public function Usuario(){}

        public var idUsuario:Number;
        public var departamento:Departamento;
```

```

        public var perfil:Perfil;
        public var nome:String;
        public var login:String;
        public var senha:String;
        public var telefone:String;
        public var email:String;
    }
}

```

APENDICE II

Código JAVA.

CriptografaArquivosAES

```

package br.com.sidoc.criptografia;

import java.io.InputStream;
import java.io.OutputStream;
import java.security.spec.AlgorithmParameterSpec;

import javax.crypto.Cipher;
import javax.crypto.CipherInputStream;
import javax.crypto.CipherOutputStream;
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;

/**
 * Projeto Final - UniCeub Engenharia da Computação
 *
 * Classe de criptografia com algoritmo AES.
 *
 * @author Sebastião Fabiano
 *
 */

public class CriptografaArquivosAES implements ICriptoSidoc {

    Cipher ecipher;
    Cipher dcipher;
    byte[] buf = new byte[1024];

    public CriptografaArquivosAES(SecretKey key) {
        // Cria um vetor de 16-byte;
        byte[] iv = new byte[] { (byte) 0x8E, 0x12, 0x39, (byte) 0x9C, 0x07,
                                0x72, 0x6F, 0x5A, (byte) 0x8E, 0x12, 0x39, (byte) 0x9C,
0x07,
                                0x72, 0x6F, 0x5A };

```

```

AlgorithmParameterSpec paramSpec = new IvParameterSpec(iv);
try {
    ecipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
    dcipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

    // CBC requer inicialização do vetor dos blocos.
    ecipher.init(Cipher.ENCRYPT_MODE, key, paramSpec);
    dcipher.init(Cipher.DECRYPT_MODE, key, paramSpec);
} catch (Exception e) {
    e.printStackTrace();
}
}

public void encrypt(InputStream in, OutputStream out) {
    try {
        out = new CipherOutputStream(out, ecipher);

        int numRead = 0;
        while ((numRead = in.read(buf)) >= 0) {
            out.write(buf, 0, numRead);
        }
        out.close();
    } catch (java.io.IOException e) {
        e.printStackTrace();
    }
}

public void decrypt(InputStream in, OutputStream out) {
    try {
        in = new CipherInputStream(in, dcipher);

        int numRead = 0;
        while ((numRead = in.read(buf)) >= 0) {
            out.write(buf, 0, numRead);
        }
        out.close();
    } catch (java.io.IOException e) {
        e.printStackTrace();
    }
}
}
}

```

CriptografaArquivosDES

```

package br.com.sidoc.criptografia;

import java.io.InputStream;
import java.io.OutputStream;
import java.security.spec.AlgorithmParameterSpec;

```

```

import javax.crypto.Cipher;
import javax.crypto.CipherInputStream;
import javax.crypto.CipherOutputStream;
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;

/**
 * Projeto Final - UniCeub Engenharia da Computação
 *
 * Classe de criptografia com algoritmo DES.
 *
 * @author Sebastião Fabiano
 */
public class CriptografaArquivosDES implements ICriptoSidoc{

    Cipher ecipher;
    Cipher dcipher;
    byte[] buf = new byte[1024];

    public CriptografaArquivosDES(SecretKey key) {
        // // Cria um vetor de 8-byte;
        byte[] iv = new byte[] { (byte) 0x8E, 0x12, 0x39, (byte) 0x9C, 0x07,
            0x72, 0x6F, 0x5A };

        AlgorithmParameterSpec paramSpec = new IvParameterSpec(iv);
        try {
            ecipher = Cipher.getInstance("DES/CBC/PKCS5Padding");
            dcipher = Cipher.getInstance("DES/CBC/PKCS5Padding");

            // CBC requer inicialização do vetor dos blocos.
            ecipher.init(Cipher.ENCRYPT_MODE, key, paramSpec);
            dcipher.init(Cipher.DECRYPT_MODE, key, paramSpec);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void encrypt(InputStream in, OutputStream out) {
        try {
            out = new CipherOutputStream(out, ecipher);
            int numRead = 0;
            while ((numRead = in.read(buf)) >= 0) {
                out.write(buf, 0, numRead);
            }
            out.close();
        } catch (java.io.IOException e) {
        }
    }
}

```

```

        public void decrypt(InputStream in, OutputStream out) {
            try {
                in = new CipherInputStream(in, dcipher);
                int numRead = 0;
                while ((numRead = in.read(buf)) >= 0) {
                    out.write(buf, 0, numRead);
                }
                out.close();
            } catch (java.io.IOException e) {
            }
        }
    }
}

```

CriptografaArquivosTripleDES

```

package br.com.sidoc.criptografia;

import java.io.InputStream;
import java.io.OutputStream;
import java.security.spec.AlgorithmParameterSpec;

import javax.crypto.Cipher;
import javax.crypto.CipherInputStream;
import javax.crypto.CipherOutputStream;
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;

/**
 * Projeto Final - UniCeub Engenharia da Computação
 *
 * Classe de criptografia com algoritmo Triple DES.
 *
 * @author Sebastião Fabiano
 *
 */
public class CriptografaArquivosTripleDES implements ICriptoSidoc{

    Cipher ecipher;
    Cipher dcipher;
    byte[] buf = new byte[1024];

    public CriptografaArquivosTripleDES(SecretKey key) {
        // Cria um vetor de 8-byte;
        byte[] iv = new byte[] { (byte) 0x8E, 0x12, 0x39, (byte) 0x9C, 0x07,
            0x72, 0x6F, 0x5A };

        AlgorithmParameterSpec paramSpec = new IvParameterSpec(iv);
        try {

```



```

    ecipher = Cipher.getInstance("TripleDES/CBC/PKCS5Padding");
    dcipher = Cipher.getInstance("TripleDES/CBC/PKCS5Padding");

    // CBC requer inicialização do vetor dos blocos.
    ecipher.init(Cipher.ENCRYPT_MODE, key, paramSpec);
    dcipher.init(Cipher.DECRYPT_MODE, key, paramSpec);
} catch (Exception e) {
    e.printStackTrace();
}
}

public void encrypt(InputStream in, OutputStream out) {
    try {
        out = new CipherOutputStream(out, ecipher);

        int numRead = 0;
        while ((numRead = in.read(buf)) >= 0) {
            out.write(buf, 0, numRead);
        }
        out.close();
    } catch (java.io.IOException e) {
    }
}

public void decrypt(InputStream in, OutputStream out) {
    try {
        in = new CipherInputStream(in, dcipher);

        int numRead = 0;
        while ((numRead = in.read(buf)) >= 0) {
            out.write(buf, 0, numRead);
        }
        out.close();
    } catch (java.io.IOException e) {
    }
}
}

```

ICriptoSidoc

```
package br.com.sidoc.criptografia;
```

```
import java.io.InputStream;
import java.io.OutputStream;
```

```
/**
```

```
* Projeto Final - UniCeub Engenharia da Computação
```

```
*
```

```
* Classe abstrata dos algoritmos de criptografia.
```

```

*
* @author Sebastião Fabiano
*
*/
public interface ICriptoSidoc {

    /**
     * Encripta arquivo
     * @param in
     * @param out
     */
    public void encrypt(InputStream in, OutputStream out);

    /**
     * Descripta arquivos
     * @param in
     * @param out
     */
    public void decrypt(InputStream in, OutputStream out);

}

```

BaseDAO

```

package br.com.sidoc.dao;

import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.HashMap;
import java.util.Map;
import java.util.Properties;
/**
 * Projeto Final - UniCeub Engenharia da Computação
 *
 * Classe de acesso ao banco de dados.
 *
 * @author Sebastião Fabiano
 *
 */
public class BaseDAO {

    private static String NAME = "com.mysql.jdbc.Driver";
    private static String URL = "jdbc:mysql://localhost:3306/sidoc";
    private static String LOGIN = "root";
    private static String PASS = "root";

```

```

private static final String PATH_SQL = "/sql/";
private static Map<String, Properties> cache;

private String comando;

public BaseDAO() {
    cache = new HashMap<String, Properties>();
}

protected String getQuery(String queryName) {
    return this.getQuery(comando, queryName);
}

public String getQuery(String comando, String queryName) {
    String retorno = null;
    if (cache.containsKey(comando)) {
        retorno = cache.get(comando).getProperty(queryName);
    } else {
        final String path = PATH_SQL + comando + ".xml";
        try {
            Properties props = new Properties();

            props.loadFromXML(getClass().getResourceAsStream(path));
            cache.put(comando, props);
            retorno = props.getProperty(queryName);
        } catch (Exception ex) {
            throw new RuntimeException(
                "Ocorreu um erro ao carregar o arquivo " +
path + ".",
                ex);
        }
    }
    return retorno;
}

/**
 *
 * @return Conexao.
 * @throws SQLException
 * @throws IOException
 */
public Connection getConexao() throws SQLException, IOException {
    Connection Connection = null;
    try {
        Class.forName(NAME);
        Connection = DriverManager.getConnection(URL, LOGIN,
PASS);
    } catch (ClassNotFoundException e) {
        System.out

```

```

        .print("\nNão foi possível estabelecer conexão com
a base de dados.\n");
        e.printStackTrace();
        return null;
    }
    return Connection;
}

protected void fecharRecursos(Connection connection, ResultSet rs,
    PreparedStatement ps) {
    try {
        fecharConexao(connection);
        fecharResultSet(rs);
        fecharStatement(ps);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

/**
 * Fecha o preparestatement.
 * @param ps
 * @throws SQLException
 */
private void fecharStatement(PreparedStatement ps) throws SQLException {
    if (ps != null)
        ps.close();
}

/**
 * Fecha o ResultSet.
 * @param rs
 * @throws SQLException
 */
private void fecharResultSet(ResultSet rs) throws SQLException {
    if (rs != null)
        rs.close();
}

/**
 * Fecha a conexão do banco de dados
 * @param connection
 * @throws SQLException
 */
private void fecharConexao(Connection connection) throws SQLException {
    if (connection != null && !connection.isClosed())
        connection.close();
}

/**

```

```

    * @return the comando
    */
    public String getComando() {
        return comando;
    }

    /**
     * @param comando the comando to set
     */
    public void setComando(String comando) {
        this.comando = comando;
    }

    /**
     * Retorna uma data sql.
     * @param longTime
     * @return
     */
    public java.sql.Date dataBanco(long longTime){
        return new java.sql.Date(longTime);
    }

    public void commit(Connection connection) throws SQLException {
        connection.commit();
        if(!connection.isClosed())
            connection.close();
    }
}

```

ArquivoDAO

```

package br.com.sidoc.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

import br.com.sidoc.exception.SidocException;
import br.com.sidoc.util.DataUtil;
import br.com.sidoc.vo.Arquivo;
import br.com.sidoc.vo.DetalheArquivo;
import br.com.sidoc.vo.Erro;
import br.com.sidoc.vo.Usuario;
/**
 * Projeto Final - UniCeub Engenharia da Computação
 *
 * Classe acesso aos referentea a cadastro de arquivos.

```

```

*
* @author Sebastião Fabiano
*
*/
public class ArquivoDAO extends BaseDAO {

    private static final String INSERIR_ARQUIVO = "Arquivo.inserirArquivo";
    private static final String INSERIR_ERROR = "Arquivo.inserirErro";
    private static final String INSERIR_DETALHE_ARQUIVO =
"Arquivo.inserirDetalhe";
    private static final String LOCALIZAR_DOCUMENTOS =
"Arquivo.localizarDocumentos";

    private final String COMANDO = "Arquivo";

    public ArquivoDAO() {
        this.setComando(COMANDO);
    }

    public void inserirDetalhe(DetalheArquivo detalheArquivo,
        Connection connection) throws SidocException {
        PreparedStatement ps = null;
        try {
            connection.setAutoCommit(false);

            ps =
connection.prepareStatement(getQuery(INSERIR_DETALHE_ARQUIVO));
            ps.setLong(1, detalheArquivo.getArquivo().getIdArquivo());
            ps.setString(2, detalheArquivo.getTitulo());
            ps.setString(3, detalheArquivo.getAssunto());
            ps.setString(4, detalheArquivo.getDescricao());
            ps.setInt(5, detalheArquivo.getTipo().intValue());
            ps.setInt(6, detalheArquivo.getAlgoritmo().intValue());
            ps
                .setDate(7,
dataBanco(detalheArquivo.getDataCadastro)
                    .getTime());
            ps.setString(8, detalheArquivo.getSenha());
            ps.execute();
        } catch (Exception ex) {
            throw new SidocException(
                "Ocorreu um erro ao tentar salvar o arquivo #"
                    +
detalheArquivo.getArquivo().getNomeOriginal(), ex);
        } finally {
            fecharRecursos(null, null, ps);
        }
    }

    public Long inserirArquivo(Connection connection, Arquivo arquivo)

```

```

        throws Exception {
    Long idArquivo = 0L;
    PreparedStatement ps = null;
    ps = connection.prepareStatement(getQuery(INSERIR_ARQUIVO));
    ps.setInt(1, arquivo.getUsuario().getIdUsuario());
    ps.setString(2, arquivo.getNomeOriginal());
    ps.setString(3, arquivo.getNomeCompactado());
    ps.setString(4, arquivo.getLocalArmazenado());
    ps.execute();
    ResultSet rs = ps.getGeneratedKeys();
    rs.next();
    idArquivo = rs.getLong(1);
    arquivo.setIdArquivo(idArquivo);
    return idArquivo;
}

public void inserirErro(Connection connection, Erro erro) throws Exception {
    PreparedStatement ps = null;
    ps = connection.prepareStatement(getQuery(INSERIR_ERROR));
    ps.setLong(1, erro.getIdArquivo());
    ps.setString(2, erro.getDescricao());
    ps.setDate(3, dataBanco(erro.getDataCadastro().getTime()));
    ps.execute();
}

public List<DetalheArquivo> localizarDocumentos(DetalheArquivo detalhe)
    throws SidocException {
    Connection connection = null;
    ResultSet rs = null;
    PreparedStatement ps = null;
    List<DetalheArquivo> lista = new ArrayList<DetalheArquivo>();
    try {
        connection = getConexao();
        String query = getQuery(LOCALIZAR_DOCUMENTOS);
        if (detalhe.getTitulo() != null && !detalhe.getTitulo().equals(""))
            query = query.replace("{1}", " and d.titulo like '%"
                + detalhe.getTitulo() + "%'");
        } else {
            query = query.replace("{1}", "");
        }
        if (detalhe.getAssunto() != null
            && !detalhe.getAssunto().equals("")) {
            query = query.replace("{2}", " and d.assunto like '%"
                + detalhe.getAssunto() + "%'");
        } else {
            query = query.replace("{2}", "");
        }
        if (detalhe.getDescricao() != null

```

```

        && !detalhe.getDescricao().equals("")) {
            query = query.replace("{3}", " and d.descricao like '%"
                + detalhe.getAssunto() + "%'");
        } else {
            query = query.replace("{3}", "");
        }
        if (detalhe.getTipo() != null && detalhe.getTipo().intValue() !=
0) {
            query = query.replace("{4}", " and d.tipo = "
                + detalhe.getTipo().intValue());
        } else {
            query = query.replace("{4}", "");
        }
        if (detalhe.getDataCadastro() != null) {
            query = query.replace("{5}", " and d.dataCadastro = "
                +
DataUtil.formataData(detalhe.getDataCadastro(),
                    DataUtil.YYYY_MM_DD)
                + "");
        } else {
            query = query.replace("{5}", "");
        }

        ps = connection.prepareStatement(query);
        rs = ps.executeQuery();
        while (rs.next()) {
            DetalheArquivo detalheArquivo = new DetalheArquivo();
            detalheArquivo.setAlgoritmo(rs.getInt("algoritmo"));
            detalheArquivo.setAssunto(rs.getString("assunto"));

            detalheArquivo.setDataCadastro(rs.getDate("dataCadastro"));
            detalheArquivo.setDescricao(rs.getString("descricao"));
            detalheArquivo.setTitulo(rs.getString("titulo"));
            detalheArquivo.setSenha(rs.getString("senha"));
            detalheArquivo.setTipo(rs.getInt("tipo"));

            Arquivo arquivo = new Arquivo();

            arquivo.setNomeCompactado(rs.getString("nome_compactado"));
            arquivo.setNomeOriginal(rs.getString("nome_original"));

            arquivo.setLocalArmazenado(rs.getString("local_armazenado"));

            Usuario usuario = new Usuario();
            usuario.setIdUsuario(rs.getInt("idUsuario"));
            usuario.setLogin(rs.getString("login"));

            arquivo.setUsuario(usuario);

```



```

        detalheArquivo.setArquivo(arquivo);
        lista.add(detalheArquivo);
    }
    return lista;
} catch (Exception e) {
    e.printStackTrace();
    throw new SidocException("Erro ao localizar o documento!");
}
}
}
}

```

AdministracaoDAO

```

package br.com.sidoc.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

import br.com.sidoc.exception.SidocException;
import br.com.sidoc.vo.Departamento;
import br.com.sidoc.vo.Perfil;
import br.com.sidoc.vo.Usuario;
/**
 * Projeto Final - UniCeub Engenharia da Computação
 *
 * Classe acesso aos referentea a cadastro de usuários.
 *
 * @author Sebastião Fabiano
 *
 */
public class AdministracaoDAO extends BaseDAO {

    private static final String LISTAR_PERFIL = "Administracao.listarPerfil";
    private static final String LISTAR_DEPARTAMETO =
"Administracao.listarDepartamento";
    private static final String PESQUISAR_USUARIO =
"Administracao.pesquisarUsuario";
    private static final String EXCLUIR_USUARIO =
"Administracao.deletarUsuario";
    private static final String INCLUIR_USUARIO =
"Administracao.salvarUsuario";
    private static final String ATUALIZAR_USUARIO =
"Administracao.atualizarUsuario";
    private static final String BUSCAR_USUARIO_POR_ID =
"Administracao.buscaUsuarioPorID";

```

```

        private static final String LISTA_USUARIOS_DEPARTAMENTO =
"Administracao.listarUsuariosDepartamento";
        private static final String BUSCAR_USUARIO_LOGIN =
"Administracao.pesquisarLogin";

        private final String COMANDO = "Administracao";

        public AdministracaoDAO() {
            this.setComando(COMANDO);
        }

        public List<Usuario> listarUsuarios() {
return null;
        }

        public void salvarUsuario(Usuario usuario) throws SidocException {
            PreparedStatement ps = null;
            Connection connection = null;
            try{
                connection = getConexao();

                ps =
connection.prepareStatement(getQuery(INCLUIR_USUARIO));
                ps.setInt(1, usuario.getDepartamento().getIdDepartamento());
                ps.setInt(2, usuario.getPerfil().getIdPerfil());
                ps.setString(3, usuario.getEmail());
                ps.setString(4, usuario.getLogin());
                ps.setString(5, usuario.getNome());
                ps.setString(6, usuario.getSenha());
                ps.setString(7, usuario.getTelefone());
                ps.execute();

            }catch(Exception ex){
                ex.printStackTrace();
                throw new SidocException("Ocorreu um erro ao tentar salvar o
usuário #" + usuario.getNome(), ex);
            }finally{
                fecharRecursos(connection, null, ps);
            }
        }

        public void atualizarUsuario(Usuario usuario) throws SidocException {
            PreparedStatement ps = null;
            Connection connection = null;
            try{
                connection = getConexao();
                ps =
connection.prepareStatement(getQuery(ATUALIZAR_USUARIO));

```

```

        ps.setInt(1, usuario.getDepartamento().getIdDepartamento());
        ps.setInt(2, usuario.getPerfil().getIdPerfil());
        ps.setString(3, usuario.getEmail());
        ps.setString(4, usuario.getLogin());
        ps.setString(5, usuario.getNome());
        ps.setString(6, usuario.getSenha());
        ps.setString(7, usuario.getTelefone());
        ps.setInt(8, usuario.getIdUsuario());
        ps.execute();
    }catch(Exception ex){
        throw new SidocException("Ocorreu um erro ao tentar salvar o
usuário #" + usuario.getNome(), ex);
    }finally{
        fecharRecursos(connection, null, ps);
    }
}

    public List<Usuario> pesquisar(String nome,Integer idPerfil, Integer
idDepartamento) throws SidocException {
        Connection connection = null;
        ResultSet rs = null;
        PreparedStatement ps = null;
        List<Usuario> lista = new ArrayList<Usuario>();
        try {
            connection = getConexao();
            String query = getQuery(PESQUISAR_USUARIO);
            if(nome!=null && !nome.equals("")){
                query = query.replace("{1}", "U.nome like
"+nome+"%");
            }
            if(idPerfil!=null && idPerfil > 0 ){
                if(nome!=null && !nome.equals("")){
                    query = query.concat(" AND P.idPerfil =
"+idPerfil);
                }else{
                    query = query.replace("{1}", " P.idPerfil =
"+idPerfil);
                }
            }
            if(idDepartamento!=null && idDepartamento > 0 ){
                if( (nome!=null && !nome.equals("")) || (idPerfil!=null
&& idPerfil > 0) ){
                    query = query.concat(" AND D.idDepartamento =
"+idDepartamento);
                }else{
                    query = query.replace("{1}", " D.idDepartamento
="+idDepartamento);
                }
            }
        }
    }
}

```

```

        ps = connection.prepareStatement(query);
        rs = ps.executeQuery();
        while(rs.next()){
            Usuario usuario = new Usuario();
            usuario.setDepartamento(new Departamento());

usuario.getDepartamento().setIdDepartamento(rs.getInt("idDepartamento"));

usuario.getDepartamento().setNome(rs.getString("nomeDepartamento"));
            usuario.setPerfil(new Perfil());
            usuario.getPerfil().setIdPerfil(rs.getInt("idPerfil"));
            usuario.getPerfil().setNome(rs.getString("nomePerfil"));
            usuario.setIdUsuario(rs.getInt("idUsuario"));
            usuario.setEmail(rs.getString("email"));
            usuario.setLogin(rs.getString("login"));
            usuario.setNome(rs.getString("nome"));
            usuario.setSenha(rs.getString("senha"));
            usuario.setTelefone(rs.getString("telefone"));
            lista.add(usuario);
        }
        return lista;
    } catch (Exception e) {
        throw new SidocException("ERRO AO BUSCAR OS Perfil",e);
    }
}

public List<Departamento> listarDepartamento() throws SidocException {
    Connection connection = null;
    ResultSet rs = null;
    PreparedStatement ps = null;
    List<Departamento> lista = new ArrayList<Departamento>();
    try {
        connection = getConexao();
        String query = getQuery(LISTAR_DEPARTAMETO);
        ps = connection.prepareStatement(query);
        rs = ps.executeQuery();
        while(rs.next()){
            Departamento departamento = new Departamento();

departamento.setIdDepartamento(rs.getInt("idDepartamento"));
            departamento.setNome(rs.getString("descricao"));
            lista.add(departamento);
        }
        return lista;
    } catch (Exception e) {
        throw new SidocException("ERRO AO BUSCAR OS
Departamento",e);
    } finally{
        fecharRecursos(connection,rs,ps);
    }
}

```

```

}

public List<Perfil> listarPerfil() throws SidocException {
    Connection connection = null;
    ResultSet rs = null;
    PreparedStatement ps = null;
    List<Perfil> lista = new ArrayList<Perfil>();
    try {
        connection = getConexao();
        String query = getQuery(LISTAR_PERFIL);
        ps = connection.prepareStatement(query);
        rs = ps.executeQuery();
        while(rs.next()){
            Perfil perfil = new Perfil();
            perfil.setIdPerfil(rs.getInt("idPerfil"));
            perfil.setNome(rs.getString("descricao"));
            lista.add(perfil);
        }
        return lista;
    } catch (Exception e) {
        throw new SidocException("ERRO AO BUSCAR OS Perfil",e);
    }finally{
        fecharRecursos(connection,rs,ps);
    }
}

public void deletarUsuario(Integer idUsuario) throws SidocException {
    PreparedStatement ps = null;
    Connection connection = null;
    try{
        connection = getConexao();
        ps =
connection.prepareStatement(getQuery(EXCLUIR_USUARIO));
        ps.setInt(1, idUsuario);
        ps.execute();
    }catch(Exception ex){
        throw new SidocException("Ocorreu um erro ao tentar excluir o
usuário #" + idUsuario, ex);
    }finally{
        fecharRecursos(connection,null,ps);
    }
}

public Usuario buscaUsuarioPorID(Integer id) throws SidocException {
    Connection connection = null;
    ResultSet rs = null;
    PreparedStatement ps = null;
    Usuario usuario = new Usuario();
    try {

```

```

        connection = getConexao();
        String query = getQuery(BUSCAR_USUARIO_POR_ID);
        ps = connection.prepareStatement(query);
        ps.setInt(1, id);
        rs = ps.executeQuery();
        if(rs.next()){
            usuario.setEmail(rs.getString("email"));
            usuario.setDepartamento(new Departamento());

        usuario.getDepartamento().setIdDepartamento(rs.getInt("idDepartamento"));
        }
        return usuario;
    } catch (Exception e) {
        throw new SidocException("ERRO AO BUSCAR
buscaUsuarioPorID ",e);
    }finally{
        fecharRecursos(connection,rs,ps);
    }
}

    public List<Usuario> listarUsuariosDepartamento(Integer idDepartamento)
throws SidocException {
        Connection connection = null;
        ResultSet rs = null;
        PreparedStatement ps = null;
        List<Usuario> usuarios = new ArrayList<Usuario>();
        try {
            connection = getConexao();
            String query =
getQuery(LISTA_USUARIOS_DEPARTAMENTO);
            ps = connection.prepareStatement(query);
            ps.setInt(1, idDepartamento);
            rs = ps.executeQuery();
            while(rs.next()){
                Usuario usuario = new Usuario();
                usuario.setEmail(rs.getString("email"));
                usuarios.add(usuario);
            }
            return usuarios;
        } catch (Exception e) {
            throw new SidocException("ERRO AO BUSCAR
UsuariosDepartamento",e);
        }finally{
            fecharRecursos(connection,rs,ps);
        }
    }

    public Usuario pesquisarLogin(String login, String senha) throws
SidocException {
        Connection connection = null;

```

```

ResultSet rs = null;
PreparedStatement ps = null;
Usuario usuario = new Usuario();
try {
    connection = getConexao();
    String query = getQuery(BUSCAR_USUARIO_LOGIN);
    ps = connection.prepareStatement(query);
    ps.setString(1, login);
    ps.setString(2, senha);
    rs = ps.executeQuery();
    if(rs.next()){
        usuario.setIdUsuario(rs.getInt("idUsuario"));
        usuario.setEmail(rs.getString("email"));
        usuario.setLogin(rs.getString("login"));
        usuario.setNome(rs.getString("nome"));
        usuario.setSenha(rs.getString("senha"));
        usuario.setTelefone(rs.getString("telefone"));

        usuario.setDepartamento(new Departamento());

        usuario.getDepartamento().setIdDepartamento(rs.getInt("idDepartamento"));

        usuario.getDepartamento().setNome(rs.getString("nomeDepartamento"));

        usuario.getDepartamento().setSigla(rs.getString("siglaDepartamento"));

        usuario.setPerfil(new Perfil());
        usuario.getPerfil().setIdPerfil(rs.getInt("idPerfil"));
        usuario.getPerfil().setNome(rs.getString("nomePerfil"));
        usuario.getPerfil().setSigla(rs.getString("siglaPerfil"));
    }
    return usuario;
} catch (Exception e) {
    throw new SidocException("ERRO AO pesquisar o usuário
pesquisarLogin ",e);
}finally{
    fecharRecursos(connection,rs,ps);
}
}
}

```

EnviaEmail

```
package br.com.sidoc.email;
```

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
```

```

import java.util.Properties;

import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.NoSuchProviderException;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
/**
 * Projeto Final - UniCeub Engenharia da Computação
 *
 * Classe responsável por enviar email.
 *
 * @author Sebastião Fabiano
 *
 */
public class EnviaEmail {

    private Session session;
    private List<String> emailCC = new ArrayList<String>();

    /**
     * @return the emailCC
     */
    public List<String> getEmailCC() {
        return emailCC;
    }

    /**
     * @param emailCC the emailCC to set
     */
    public void setEmailCC(List<String> emailCC) {
        this.emailCC = emailCC;
    }

    public static void main(String[] args) {
        try {
            String emailDestino = "sebast.fabiano@gmail.com";

            EnviaEmail email = new EnviaEmail();
            //email.getEmailCC().add("waleons.negreiros@sicoob.com.br");
            for (int i = 0; i < 5; i++) {
                email.enviar(emailDestino, "EMAIL ENVIADO "+i);
            }

        } catch (Exception e) {

```



```

        e.printStackTrace();
    }
}

public void enviar(String emailDestino, String mensagem)
    throws MessagingException, AddressException,
    NoSuchProviderException {
    Properties p = getProperties();

    session = Session.getDefaultInstance(p,null);

    MimeMessage msg = new MimeMessage(session);
    msg.setFrom(new InternetAddress("sidoc@sidoc.com"));
    msg.setRecipient(Message.RecipientType.TO, new InternetAddress(
emailDestino ));
    msg.setRecipients(Message.RecipientType.CC, buildInternetAddress(
this.emailCC ));
    msg.setSubject("PROCESSAMENTO DE ARQUIVOS SISTEMA
SIDOC.");
    msg.setText(mensagem);

    Transport.send(msg);
    session = null;

}

private static Properties getProperties() {
    Properties p = new Properties();

    p.put("mail.transport.protocol", "smtp");
    p.put("mail.smtp.starttls.enable", "false");
    p.put("mail.smtp.host", "ENGENHARIA");

    p.put("mail.debug", "true");
    p.put("mail.smtp.starttls.enable", "false");
    p.put("mail.smtp.port", "25");
    return p;
}
/**
 * Método para disponibilizar a lista de enderecos.
 *
 * @param list É uma variável do tipo List que representa os enderecos em
cópia.
 * @return Seta uma lista (InternetAddress[]) com o endereço de destino
(Cópia) do E-mail.
 */
private InternetAddress[] buildInternetAddress (List<?> list) throws
AddressException
{

```

```

        InetAddress [] arr = new InetAddress [list.size()];

        Iterator<?> it = list.iterator();

        for (int i = 0; it.hasNext(); i++)
        {
            arr[i] = new InetAddress (((String)it.next()).toLowerCase());
        }

        return arr;
    }
}

```

TipoAlgoritmo

```

package br.com.sidoc.enums;

public enum TipoAlgoritmo {

    AES(1, "AES"),
    TRIPLE_DES(2, "Triple DES"),
    DES(3, "DES");

    private Integer id;
    private String descricao;

    private TipoAlgoritmo(Integer id, String descricao) {
        this.id = id;
        this.descricao = descricao;
    }

    /**
     * @return the id
     */
    public Integer getId() {
        return id;
    }

    /**
     * @param id the id to set
     */
    public void setId(Integer id) {
        this.id = id;
    }

    /**
     * @return the descricao
     */
    public String getDescricao() {
        return descricao;
    }

    /**
     * @param descricao the descricao to set
     */
    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
}

```

TipoArquivo

```

package br.com.sidoc.enums;

public enum TipoArquivo {

    PESSOAL(1, "Pessoal"),
    DEPARTAMENTAL(2, "Departamental");

    private Integer id;
    private String descricao;

    private TipoArquivo(Integer id, String descricao) {
        this.id = id;
        this.descricao = descricao;
    }

    /**
     * @return the id
     */
    public Integer getId() {
        return id;
    }

    /**
     * @param id the id to set
     */
    public void setId(Integer id) {
        this.id = id;
    }

    /**
     * @return the descricao
     */
    public String getDescricao() {
        return descricao;
    }

    /**
     * @param descricao the descricao to set
     */
    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }

}

```

SidocException

```

package br.com.sidoc.exception;

import br.com.sidoc.util.PropertiesUtil;
/**
 * Projeto Final - UniCeub Engenharia da Computação
 */

```

```

* Classe representativa das exceções do sistema.
*
* @author Sebastião Fabiano
*
*/
public class SidocException extends Exception{

    private static final long serialVersionUID = 1L;

    public SidocException() {
        super();
    }

    public SidocException(String message, Throwable cause) {
        super(PropertiesUtil.mensagem(message), cause);
    }

    public SidocException(String message) {
        super(PropertiesUtil.mensagem(message));
    }

    public SidocException(Throwable cause) {
        super(cause);
    }

}

```

AdministracaoService

```
package br.com.sidoc.servico;
```

```
import java.util.List;
```

```
import org.apache.log4j.Logger;
```

```
import br.com.sidoc.dao.AdministracaoDAO;
import br.com.sidoc.exception.SidocException;
import br.com.sidoc.vo.Departamento;
import br.com.sidoc.vo.Perfil;
import br.com.sidoc.vo.Usuario;
/**
```

```

* Projeto Final - UniCeub
* Engenharia da Computação
*
* Classe responsável pela lógica de negocio do
* Módulo Manter Usuários.
* @author Sebastião Fabiano
*
*/

```

```
public class AdministracaoService {
```

```
    private static Logger logger = Logger.getLogger(AdministracaoService.class);
```

```

public String getMensagem() {
    return "Oi";
}
/**
 * Lista de usuários
 * @return java.util.List
 */
public List<Usuario> listarUsuarios() {
    return getDAO().listarUsuarios();
}
/**
 * Lista os departamentos
 * @return java.util.List
 * @throws SidocException
 */
public List<Departamento> listarDepartamento() throws SidocException {
    return getDAO().listarDepartamento();
}
/**
 * Lista os Perfils
 * @return java.util.List
 * @throws SidocException
 */
public List<Perfil> listarPerfil() throws SidocException {
    return getDAO().listarPerfil();
}
/**
 * Lobtem uma instancia da DAO de Administração.
 * @return AdministracaoDAO
 */
public AdministracaoDAO getDAO() {
    return new AdministracaoDAO();
}
/**
 * Salva o usuário.
 * @param usuario
 * @return String
 */
public String salvarUsuario(Usuario usuario) {
    logger.debug("Salvando Usuário...");
    try {
        if (usuario.getIdUsuario().intValue() == 0) {
            getDAO().salvarUsuario(usuario);
        } else {
            getDAO().atualizarUsuario(usuario);
        }
        logger.debug("Usuário salvo...");
    } catch (Exception e) {
        return "Erro ao Salvar :" + e.getMessage();
    }
}

```

```

        return "Usuário Salvo com Sucesso!";
    }
    /**
     * Pesquisa os usuários
     * @param nome
     * @param idPerfil
     * @param idDepartamento
     * @return java.util.List
     * @throws SidocException
     */
    public List<Usuario> pesquisar(String nome, Integer idPerfil,
        Integer idDepartamento) throws SidocException {
        logger.debug("Pesquisando Usuários...");
        return getDAO().pesquisar(nome, idPerfil, idDepartamento);
    }
    /**
     * Pesquisa os usuários
     * @param nome
     * @param idPerfil
     * @param idDepartamento
     * @return java.util.List
     * @throws SidocException
     */
    public Usuario pesquisarLogin(String login, String senha) throws
SidocException {
        logger.debug("Pesquisando Usuários...");
        return getDAO().pesquisarLogin(login,senha);
    }
    /**
     * Deleta os usuários
     * @param idUsuario
     * @return booleana
     * @throws SidocException
     */
    public boolean deletarUsuario(Integer idUsuario) throws SidocException{
        logger.debug("Excluido Usuário..." +idUsuario);
        getDAO().deletarUsuario(idUsuario);
        return true;
    }
}

```

ArquivoService

```

package br.com.sidoc.servico;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.sql.Connection;

```

```

import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import javax.crypto.spec.SecretKeySpec;

import org.apache.log4j.Logger;

import br.com.sidoc.criptografia.ICriptoSidoc;
import br.com.sidoc.criptografia.CriptografaArquivosAES;
import br.com.sidoc.criptografia.CriptografaArquivosDES;
import br.com.sidoc.criptografia.CriptografaArquivosTripleDES;
import br.com.sidoc.dao.AdministracaoDAO;
import br.com.sidoc.dao.ArquivoDAO;
import br.com.sidoc.email.EnviaEmail;
import br.com.sidoc.enums.TipoAlgoritmo;
import br.com.sidoc.enums.TipoArquivo;
import br.com.sidoc.exception.SidocException;
import br.com.sidoc.util.DataUtil;
import br.com.sidoc.util.FileUtil;
import br.com.sidoc.util.GerarSenha;
import br.com.sidoc.util.PropertiesUtil;
import br.com.sidoc.util.StringUtil;
import br.com.sidoc.vo.DetalheArquivo;
import br.com.sidoc.vo.Erro;
import br.com.sidoc.vo.Usuario;

/**
 * Projeto Final - UniCeub
 * Engenharia da Computação
 *
 * Classe responsável pela lógica de negocio do
 * Módulo Manter Documentos.
 * @author Sebastião Fabiano
 *
 */

public class ArquivoService {

    private static final String CRIPTOGRAFADOS = "Criptografados";
    private static final String DESCRIPTOGRAFADOS = "Descriptografados";

    private ArquivoDAO dao = new ArquivoDAO();
    private AdministracaoDAO administracaoDAO = new AdministracaoDAO();
    private static Logger logger = Logger.getLogger(ArquivoService.class);

    /**
     * Método responsável pela importação de
     * arquivos
     * @param detalhe

```

```

* @return String
* @throws Exception
*/
public String importarArquivo(DetalheArquivo detalhe) throws Exception {
    logger.debug("Iniciando processo de importação de documentos...");
    Connection connection = null;
    try {
        connection = administracaoDAO.getConexao();

        detalhe.getArquivo().setNomeOriginal(StringUtil.removeAcento(detalhe.getArq
uivo()
        .getNomeOriginal()));

        String arquivoZipado = geraNomeArquivoZip(detalhe);
        String diretorioArquivo =
criaDiretorioArquivoCompactado(detalhe);

        FileUtil.zipArquivo(pastaRaiz(detalhe), detalhe.getArquivo()
        .getNomeOriginal(),pastaRaiz(detalhe),
arquivoZipado);

        logger.debug("Nome arquivo zip "+arquivoZipado);

        String senha = geraSenhaAlgoritmo(detalhe);

        logger.debug("Senha de criptografia "+senha);

        detalhe.getArquivo().setNomeCompactado(arquivoZipado);
        detalhe.getArquivo().setLocalArmazenado(diretorioArquivo);
        detalhe.setDataCadastro(new Date());
        detalhe.setSenha(senha);

        ICriptoSidoc cripto = obtemAlgoritmo(detalhe.getSenha(),
detalhe.getAlgoritmo());

        cripto.encrypt(new FileInputStream(new
File(pastaRaiz(detalhe)+"\\"+arquivoZipado)),
        new
FileOutputStream(diretorioArquivo+"\\"+arquivoZipado));

        deletaArquivo(pastaRaiz(detalhe),detalhe.getArquivo().getNomeOriginal());
        deletaArquivo(pastaRaiz(detalhe),arquivoZipado);

        enviaEmailSucesso(detalhe);

```



```

        dao.inserirArquivo(connection, detalhe.getArquivo());
        dao.inserirDetalhe(detalhe,connection);
        dao.commit(connection);

    } catch (Exception e) {
        e.printStackTrace();
        dao.commit(connection);
        logger.error("ERROR processo de importação de
documentos..." +e.getMessage());
        gravarErro(detalhe, e);
        return
PropertiesUtil.getInstancia().getValor(PropertiesUtil.SDM0001).
        replace("{1}",detalhe.getArquivo().getNomeOriginal());
    }
    logger.debug("Fim processo de importação de documentos...");
    return PropertiesUtil.mensagem("SDM0006");
}

/**
 * Método responsável por gravar os eventuais
 * error da aplicação
 * @param detalhe
 * @param e
 * @throws SidocException
 */
private void gravarErro(DetalheArquivo detalhe, Exception e) throws
SidocException {
    try {
        Connection connection = administracaoDAO.getConexao();
        Erro erro = new Erro();
        erro.setDescricao(e.toString()+" : "+e.getMessage());
        erro.setDataCadastro(new Date());
        erro.setIdArquivo(dao.inserirArquivo(connection,
detalhe.getArquivo()));
        dao.inserirErro(connection, erro);
        dao.commit(connection);
    } catch (Exception e1) {
        e.printStackTrace();
        throw new SidocException("SDM0004");
    }
}

/**
 * Método de envio de email.
 * @param detalhe
 * @throws SidocException
 */
private void enviaEmailSucesso(DetalheArquivo detalhe) throws
SidocException {
    try {

```

```

        logger.debug("Enviando email...");
        EnviaEmail email = new EnviaEmail();
        Usuario us =
administracaoDAO.buscaUsuarioPorID(detalle.getArquivo().getUsuario().getIdUsuario
());
        String mensagem =
PropertiesUtil.getInstancia().getValor(PropertiesUtil.SDM0003).replace("{1}",
        detalle.getArquivo().getNomeOriginal()).replace("{2}", detalle.getSenha());
        mensagem = mensagem.replace("{3}", detalle.getTitulo());
        mensagem = mensagem.replace("{4}", detalle.getAssunto());
        mensagem = mensagem.replace("{5}", detalle.getDescricao());
        List<String> listEmailCC = new ArrayList<String>();
        if(!arquivoEhPessoal(detalle)){
            List<Usuario> lista =
administracaoDAO.listarUsuariosDepartamento(
us.getDepartamento().getIdDepartamento() );
            for (Usuario usuario : lista) {
                if(!us.getEmail().equals(usuario.getEmail()))
                    listEmailCC.add(usuario.getEmail());
            }
            email.setEmailCC(listEmailCC);
        }
        email.enviar(us.getEmail(),mensagem);
        logger.debug("Email enviado...");

deletaArquivo(pastaRaiz(detalle),detalle.getArquivo().getNomeCompactado());
    } catch (Exception e) {
        logger.error("Error ao enviar email..." + e.getMessage());
        throw new SidocException("Erro ao enviar email");
    }
}
/**
 * Método de gerar senhas randomicamente
 * @param detalle
 * @return String
 */
private String geraSenhaAlgoritmo(DetalleArquivo detalle) {
    String senha;
    if(TipoAlgoritmo.AES.getId() == detalle.getAlgoritmo().intValue()){
        senha = GerarSenha.gerarSenha(16);
    }else if(TipoAlgoritmo.TRIPLE_DES.getId() ==
detalle.getAlgoritmo().intValue()){
        senha = GerarSenha.gerarSenha(24);
    }else{
        senha = GerarSenha.gerarSenha(8);
    }
    return senha;
}
/**

```

```

    * Cria diretorio de arquivos compactados.
    * @param detalhe
    * @return String
    */
    private String criaDiretorioArquivoCompactado(DetalheArquivo detalhe) {
        String diretorioArquivo = "TEMP";
        if(arquivoEhPessoal(detalhe)){
            diretorioArquivo =
PropertiesUtil.getInstancia().getValor(PropertiesUtil.LOCAL_FILES)

+"\\ "+detalhe.getArquivo().getUsuario().getLogin()+"\\ "+CRIPTOGRAFADOS;
        }else{
            diretorioArquivo =
PropertiesUtil.getInstancia().getValor(PropertiesUtil.LOCAL_FILES)

+"\\ "+detalhe.getArquivo().getUsuario().getDepartamento().getSigla()+"\\ "+CRIPTOG
RAFADOS;
        }
        logger.debug("Diretorio arquivo compactado :"+diretorioArquivo);
        new File(diretorioArquivo).mkdirs();
        return diretorioArquivo;
    }
/**
 * Verifica se o arquivo e pessoal ou departamental
 * @param detalheArquivo
 * @return booleana
 */
    private boolean arquivoEhPessoal(DetalheArquivo detalheArquivo){
        if(detalheArquivo.getTipo().intValue() ==
TipoArquivo.PESSOAL.getId().intValue()){
            return true;
        }else{
            return false;
        }
    }
/**
 * Localiza o documento armazenado
 * @param detalhe
 * @return java.util.List
 * @throws SidocException
 */
    public List<DetalheArquivo> localizarDocumentos(DetalheArquivo detalhe)
throws SidocException{
        try {
            return dao.localizarDocumentos(detalhe);
        } catch (SidocException e) {
            throw e;
        }
    }
}

```

```

/**
 * Descriptografa o arquivo
 * @param detalheArquivo
 * @return boolean
 * @throws SidocException
 */
public String descriptografaArquivo(DetalheArquivo detalheArquivo) throws
SidocException{
    try {
        logger.debug("Inicio da descriptografia do arquivo
:"+detalheArquivo.getArquivo().getNomeCompactado());
        ICriptoSidoc cripto =
obtemAlgoritmo(detalheArquivo.getSenha(), detalheArquivo.getAlgoritmo());

        String localDescripto =
detalheArquivo.getArquivo().getLocalArmazenado().replace(CRIPTOGRAFADOS,
DESCRITOGRAMADOS);

        new File(localDescripto).mkdirs();

        cripto.decrypt(new FileInputStream(new
File(detalheArquivo.getArquivo().getLocalArmazenado()

+"\""+detalheArquivo.getArquivo().getNomeCompactado()))),
        new FileOutputStream(new
File(localDescripto+"\""+detalheArquivo.getArquivo().getNomeCompactado())));

        File fileZip = new
File(localDescripto+"\""+detalheArquivo.getArquivo().getNomeCompactado());

        FileUtil.unzip(fileZip ,new File(localDescripto) );

        fileZip.delete();

        String localAbsoluto =
localDescripto+"\""+detalheArquivo.getArquivo().getNomeOriginal();

        logger.debug("Fim da descriptografia do arquivo
:"+localAbsoluto);

        return localAbsoluto;

    } catch (Exception e) {
        e.printStackTrace();
        gravarErro(detalheArquivo, e);
        throw new SidocException("SDM0005");
    }
}

```

```

/**
 * Deleta arquivos temporários
 * @param local
 * @param nome
 *
 */
private void deletaArquivo(String local, String nome) {
    File file = new File(local+"\\")+nome);
    if (file.exists()) {
        file.delete();
    }
}

/**
 * Gera nome arquivo compactado
 * @param detalhe
 * @return String
 */
private String geraNomeArquivoZip(DetalheArquivo detalhe) {
    String extensao = detalhe.getArquivo().getExtensao();
    String arquivoZipado =
detalhe.getArquivo().getNomeOriginal().replace(extensao, "")
        + DataUtil.formataData(new Date(),
            DataUtil.DD_MM_YYYY_HH_MM_SS)
+ ".zip";
    return arquivoZipado;
}

/**
 * Retorna a pasta raiz
 * @return String
 */
private String pastaRaiz(DetalheArquivo detalhe) {
    String local = "";
    if(arquivoEhPessoal(detalhe)){
        local = PropertiesUtil.getInstancia().getValor(
            PropertiesUtil.LOCAL_FILES)
            + "\\")+detalhe.getArquivo().getUsuario().getLogin();
    }else{
        local = PropertiesUtil.getInstancia().getValor(
            PropertiesUtil.LOCAL_FILES)
            +
            "\\")+detalhe.getArquivo().getUsuario().getDepartamento().getSigla();
    }
    logger.debug("pastaRaiz "+local);
    new File(local).mkdirs();
    return local;
}

/**
 * Obtem o algoritmo de criptografia.
 * @param chave

```

```

    * @param tipo
    * @return AbstractCripto
    * @throws SidocException
    */
    private ICriptoSidoc obtemAlgoritmo(String chave, Integer tipo) throws
SidocException{
        if(TipoAlgoritmo.AES.getId().intValue() == tipo.intValue()){
            logger.debug("Algoritmo escolhido foi
:"+TipoAlgoritmo.AES.getDescricao());
            return new CriptografaArquivosAES(new
SecretKeySpec(chave.getBytes(), "AES"));
        }else if(TipoAlgoritmo.TRIPLE_DES.getId().intValue() ==
tipo.intValue()){
            logger.debug("Algoritmo escolhido foi
:"+TipoAlgoritmo.TRIPLE_DES.getDescricao());
            return new CriptografaArquivosTripleDES(new
SecretKeySpec(chave.getBytes(), "TripleDES"));
        }else if(TipoAlgoritmo.DES.getId().intValue() == tipo.intValue()){
            logger.debug("Algoritmo escolhido foi
:"+TipoAlgoritmo.DES.getDescricao());
            return new CriptografaArquivosDES(new
SecretKeySpec(chave.getBytes(), "DES"));
        }else{
            throw new SidocException("SDM0007");
        }
    }
}

```

ArquivoDownload

```

package br.com.sidoc.servlet;

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

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ArquivoDownload extends HttpServlet {

    private static final long serialVersionUID = -8903751393907460009L;

    protected void doGet(HttpServletRequest request,

```

```

        HttpServletResponse response) throws ServletException,
IOException {
    try {
        String arquivoStr = request.getParameter("nomeArquivo");

        File arquivo = new File(arquivoStr);
        FileInputStream input = new FileInputStream(arquivo);

        response.setHeader("Content-Disposition",
"attachment;filename=\"" + arquivo.getName() + "\"");
        response.setContentLength((int) arquivo.length());
        ServletContext context = super.getServletContext();

        response.setContentType(context.getMimeType(arquivo.getName()));
        System.out.println("Mimetype "
            + context.getMimeType(arquivo.getName()));

        ServletOutputStream out = response.getOutputStream();
        byte[] buffer = new byte[1024];
        int i;
        while ((i = input.read(buffer)) != -1) {
            out.write(buffer, 0, i);
        }
        out.flush();
        input.close();
        out.close();
        System.out.println("FIM DO DOWNLOAD..
ARQUIVODOWNLOAD");
        new File(arquivoStr).delete();
    } catch (IOException err) {
        err.printStackTrace();
    }
}
}

```

ArquivoUpload

```

package br.com.sidoc.servlet;

import java.io.File;
import java.io.IOException;
import java.util.Iterator;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```

```

import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileItemFactory;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;

import br.com.sidoc.exception.SidocException;
import br.com.sidoc.util.PropertiesUtil;

public class ArquivoUpload extends HttpServlet {

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    public ArquivoUpload() {

    }

    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
IOException {

        doPost(request, response);

    }

    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
IOException {

        try {
            boolean isMultipart =
ServletFileUpload.isMultipartContent(request);
            if (isMultipart) {
                FileItemFactory factory = new DiskFileItemFactory();
                ServletFileUpload upload = new
ServletFileUpload(factory);
                List<?> /* FileItem */items =
upload.parseRequest(request);
                Iterator<?> iter = items.iterator();
                while (iter.hasNext()) {
                    FileItem item = (FileItem) iter.next();
                    if (item.isFormField()) {
                        processFormField(item);
                    } else {
                        String login =
request.getParameter("loginUsuario");
                        processUploadedFile(item,login);
                    }
                }
            }
        }
    }
}

```



```

        }
    }
}

catch (Exception e) {
    e.printStackTrace();
}

}

private void processFormField(FileItem item) {
    String name = item.getFieldName();
    String value = item.getString();
    System.out.println("Item name: " + name + " ; value: " + value);
}

private void processUploadedFile(FileItem item,String login) throws
SidocException {
    String fileName = item.getName();
    boolean writeToFile = true;
    try {
        new File(local()+"\\"+login).mkdir();
        String path = local()+"\\"+login+"\\"+fileName;
        if (writeToFile) {
            File uploadedFile = new File(path);
            if (!uploadedFile.exists()) {
                uploadedFile.createNewFile();
            }
            item.write(uploadedFile);
        } else {
            System.out.println("Trying to write a large file.");
        }
    } catch (Exception e) {
        throw new SidocException(e);
    }
}

private String local() {
    return
PropertiesUtil.getInstancia().getValor(PropertiesUtil.LOCAL_FILES);
}

}

```

Administracao.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>Queries do modulo de administracao </comment>

<entry key="Administracao.listarPerfil">

```

```

    <![CDATA[
        SELECT
            idPerfil,
            descricao
        FROM
            perfil
    ]]>
</entry>

<entry key="Administracao.listarDepartamento">
    <![CDATA[
        SELECT
            idDepartamento,
            descricao
        FROM
            departamento
    ]]>
</entry>

<entry key="Administracao.deletarUsuario">
    <![CDATA[
        DELETE
        FROM
            usuario
        WHERE
            idUsuario = ?
    ]]>
</entry>

<entry key="Administracao.pesquisarUsuario">
    <![CDATA[
        SELECT
            U.idUsuario,
            U.email,
            U.login,
            U.nome,
            U.senha,
            U.telefone,
            D.idDepartamento,
            D.descricao AS nomeDepartamento,
            P.idPerfil,
            P.descricao AS nomePerfil
        FROM
            sidoc.usuario U
        LEFT JOIN sidoc.departamento D
            ON
                D.idDepartamento = U.idDepartamento
        LEFT JOIN sidoc.perfil P
            ON
                P.idPerfil = U.idPerfil
        WHERE
            {1}
    ]]>
</entry>

<entry key="Administracao.pesquisarLogin">
    <![CDATA[
        SELECT
            U.idUsuario,
            U.email,
            U.login,
    ]]>

```

```

        U.nome,
        U.senha,
        U.telefone,
        D.idDepartamento,
        D.descricao AS nomeDepartamento,
          D.siga as siglaDepartamento,
        P.idPerfil,
        P.descricao AS nomePerfil,
          P.sigla AS siglaPerfil
FROM
    sidoc.usuario U
LEFT JOIN sidoc.departamento D
ON
    D.idDepartamento = U.idDepartamento
LEFT JOIN sidoc.perfil P
ON
    P.idPerfil = U.idPerfil
WHERE
    U.login = ?
    and U.senha = ?

]]>
</entry>

<entry key="Administracao.salvarUsuario">
  <![CDATA[
    INSERT
    INTO
      usuario
      (
        idDepartamento,
        idPerfil,
        email,
        login,
        nome,
        senha,
        telefone
      )
    VALUES
      (
        ?,
        ?,
        ?,
        ?,
        ?,
        ?,
        ?
      )
  ]]>
</entry>
<entry key="Administracao.atualizarUsuario">
  <![CDATA[
    UPDATE
      usuario
    SET
      idDepartamento = ?,
      idPerfil = ?,
      email = ?,
      login = ?,
  ]]>

```

```

        nome = ?,
        senha = ?,
        telefone = ?
WHERE
    idUsuario = ?
    ]]>
</entry>

<entry key="Administracao.listarUsuariosDepartamento">
    <![CDATA[
        SELECT
            u.email
        FROM
            sidoc.departamento d
        INNER JOIN sidoc.usuario u
        ON
            d.idDepartamento = u.idDepartamento
        where
            u.idDepartamento = ?
    ]]>
</entry>
<entry key="Administracao.buscaUsuarioPorID">
    <![CDATA[
        SELECT
            u.email,
            u.idDepartamento
        FROM
            sidoc.usuario u
        where
            u.idUsuario = ?
    ]]>
</entry>

</properties>

```

Arquivo.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>Queries do modulo de gerenciar arquivos </comment>

<entry key="Arquivo.inserirDetalhe">
    <![CDATA[
        INSERT
        INTO
            detalhe_arquivo
        (
            idArquivo,
            titulo,
            assunto,
            descricao,
            tipo,
            algoritimo,
            dataCadastro,
            senha
        )
        VALUES
        (

```

```

        ?,
        ?,
        ?,
        ?,
        ?,
        ?,
        ?,
        ?
    )
]]>
</entry>

<entry key="Arquivo.inserirArquivo">
    <![CDATA[
        INSERT
        INTO
        arquivo
        (
            idUsuario,
            nome_original,
            nome_compactado,
            local_armazenado
        )
        VALUES
        (
            ?,
            ?,
            ?,
            ?
        )
    ]]>
</entry>

<entry key="Arquivo.inserirErro">
    <![CDATA[
        INSERT
        INTO
        erro
        (
            idArquivo,
            descricao,
            dataCadastro
        )
        VALUES
        (
            ?,
            ?,
            ?
        )
    ]]>
</entry>

<entry key="Arquivo.localizarDocumentos">
    <![CDATA[
        SELECT
            d.idDetalhe,
            d.idArquivo,
            d.titulo,
            d.assunto,
            d.descricao,

```

```

        d.tipo,
        d.algoritmo,
        d.dataCadastro,
        d.senha,
        a.idArquivo,
        a.idUsuario,
        a.nome_original,
        a.nome_compactado,
        a.local_armazenado,
        u.login
    FROM
        sidoc.detalhe_arquivo d
    INNER JOIN sidoc.arquivo a
    ON
        a.idArquivo = d.idArquivo
    INNER JOIN sidoc.usuario u
    ON u.idUsuario = a.idUsuario
    WHERE
        1 = 1
        {1} {2} {3} {4} {5}
    ]]>
</entry>
</properties>

```

remoting-config.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<service id="remoting-service"
    class="flex.messaging.services.RemotingService">

    <adapters>
        <adapter-definition id="java-object"
class="flex.messaging.services.remoting.adapters.JavaAdapter"
default="true"/>
    </adapters>

    <default-channels>
        <channel ref="my-amf" />
    </default-channels>

    <destination id="administracaoService">
        <properties>
            <source>br.com.sidoc.servico.AdministracaoService</source>
        </properties>
    </destination>

    <destination id="gerenciarArquivosService">
        <properties>
            <source>br.com.sidoc.servico.ArquivoService</source>
        </properties>
    </destination>

</service>

```

Script de Banco MYQL

```
DROP TABLE arquivo;
CREATE TABLE arquivo ( idArquivo int(10) unsigned NOT NULL
AUTO_INCREMENT, idUsuario int(10) unsigned NOT NULL, nome_original
varchar(200), nome_compactado varchar(200), local_armazenado varchar(200),
PRIMARY KEY USING BTREE (idArquivo) ) ENGINE=InnoDB DEFAULT
CHARSET=latin1;
DROP TABLE departamento;
CREATE TABLE departamento ( idDepartamento int(10) unsigned NOT NULL
AUTO_INCREMENT, descricao varchar(120), siga varchar(10), PRIMARY KEY
USING BTREE (idDepartamento) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
DROP TABLE detalhe_arquivo;
CREATE TABLE detalhe_arquivo ( idDetalhe int(10) unsigned NOT NULL
AUTO_INCREMENT, idArquivo int(10) unsigned NOT NULL, titulo varchar(200),
assunto varchar(200), descricao varchar(200), tipo int(10) unsigned, algoritmo
varchar(20), dataCadastro date, senha varchar(50), PRIMARY KEY USING BTREE
(idDetalhe) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
DROP TABLE erro;
CREATE TABLE erro ( idErro int(10) unsigned NOT NULL AUTO_INCREMENT,
idArquivo int(10) unsigned NOT NULL, descricao varchar(200), dataCadastro date,
PRIMARY KEY USING BTREE (idErro) ) ENGINE=InnoDB DEFAULT
CHARSET=latin1;
DROP TABLE perfil;
CREATE TABLE perfil ( idPerfil int(10) unsigned NOT NULL AUTO_INCREMENT,
sigla varchar(10), descricao varchar(120), PRIMARY KEY USING BTREE (idPerfil) )
ENGINE=InnoDB DEFAULT CHARSET=latin1;
DROP TABLE usuario;
CREATE TABLE usuario ( idUsuario int(10) unsigned NOT NULL
AUTO_INCREMENT, idDepartamento int(10) unsigned NOT NULL, idPerfil int(10)
unsigned NOT NULL, nome varchar(120), email varchar(200), login varchar(10), senha
varchar(10), telefone varchar(20), PRIMARY KEY USING BTREE (idUsuario) )
ENGINE=InnoDB DEFAULT CHARSET=latin1;
ALTER TABLE arquivo ADD FOREIGN KEY (idUsuario) REFERENCES usuario
(idUsuario) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE detalhe_arquivo ADD FOREIGN KEY (idArquivo) REFERENCES
arquivo (idArquivo) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE erro ADD FOREIGN KEY (idArquivo) REFERENCES arquivo
(idArquivo) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE usuario ADD FOREIGN KEY (idPerfil) REFERENCES perfil
(idPerfil) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE usuario ADD FOREIGN KEY (idDepartamento) REFERENCES
departamento (idDepartamento) ON DELETE NO ACTION ON UPDATE NO
ACTION;
```