CENTRO UNIVERSITÁRIO DE BRASÍLIA – UniCEUB
FACULDADE DE TECNOLOGIA E CIÊNCIA SOCIAIS APLICADAS - FATECS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

PROJETO FINAL

PAULO GABRIEL RIBACIONKA GÓES DE ARAÚJO

SISTEMA DE CONTROLE DE ACESSO VIA *SMART CARD* COM AUTENTICAÇÃO BIOMÉTRICA DA IMPRESSÃO DIGITAL

Orientador: M. Sc. MARCO ANTÔNIO DE OLIVEIRA ARAÚJO

BRASÍLIA – DF,

DEZEMBRO DE 2010

PAULO GABRIEL RIBACIONKA GÓES DE ARAÚJO

RA: 2051650/7

SISTEMA DE CONTROLE DE ACESSO VIA *SMART CARD* COM AUTENTICAÇÃO BIOMÉTRICA DA IMPRESSÃO DIGITAL

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de Certificado de Conclusão do Curso de Engenharia de Computação.
Orientador: M. Sc. Marco Antônio de Oliveira Araújo.

BRASÍLIA – DF,

DEZEMBRO DE 2010

PAULO GABRIEL RIBACIONKA GÓES DE ARAÚJO

RA: 2051650/7

SISTEMA DE CONTROLE DE ACESSO VIA *SMART CARD* COM AUTENTICAÇÃO BIOMÉTRICA DA IMPRESSÃO DIGITAL

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de Certificado de Conclusão do Curso de Engenharia de Computação.
Orientador: M. Sc. Marco Antônio de Oliveira Araújo.

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.

_____
Abiezer Amarília Fernandez
Coordenador do Curso

Banca Examinadora:

_____
Prof. Marco Antônio de Oliveira Araújo, Mestre em Ciência da Computação.
Orientador – UniCEUB

_____
Prof. Miguel Arcanjo Bacellar Goes Telles Junior, Doutor em Geologia.
UniCEUB

_____
Prof. Vera Lucia Farini Alves Duarte, Mestre em Matemática.
UniCEUB

DEDICATÓRIA

Dedico este trabalho e todos os meus esforços para conclusão do curso de Engenharia de Computação às pessoas mais importantes da minha vida: meu pai Ilton José Góes de Araújo, minha mãe Neusa Ribacionka, minhas quatro irmãs Paula, Juliana, Maysa e Carol e à minha amada namorada Thaís Maia.

“Determinando tu algum negócio, ser-te-á firme, e a luz brilhará em teus caminhos.”

Bíblia – Jó, 22:28

AGRADECIMENTOS

Em primeiro lugar agradeço minha mãe Neusa Ribacionka pelo incondicional apoio. Agradeço ao meu pai Ilton José por toda inspiração e por todo exemplo de vitória e perseverança de sua história de vida.

Agradeço à minha amada Thaís Maia pela companhia durante grande parte da minha jornada acadêmica e pela paciência durante os momentos que exigiram dedicação total de minha parte ao projeto final.

Agradeço ao CEUB, do qual sou aluno desde 1994, por toda a estrutura e oportunidades oferecidas.

Agradeço aos professores do cursos de Engenharia de Computação por todos os ensinamentos, conselhos e dedicação.

Finalmente agradeço aos meus amigos Paulo Ueiner e Said Abd pela companhia durante as horas intermináveis de estudo no decorrer do curso de Engenharia de Computação.

LISTA DE FIGURAS

LISTA DE QUADROS E TABELAS

LISTA DE ABREVIATURAS

RFID – *Radio Frequency Identification*, em português, Identificação por Rádio Frequência.

AFIS - *Automated Fingerprint Identification System*, em português, Sistema de Identificação Automatizada da Impressão Digital.

GSM - *Global System for Mobile Communications*, em português, Sistema Global para Comunicações Móveis. (RFC 4186)

ISO - *International Organization for Standardization*, em português, Organização Internacional para Padronização.

ROM – *Read Only Memory*, em português, Memória Somente de Leitura.

EEPROM - *Electrically-Erasable Programmable Read-Only Memory*.

RAM – *Random Access Memory*, em português, Memória de Acesso Randômico.

APDU - *Application Protocol Data Units.*

TPDU – *Transmission Protocol Data Units.*

ATR – *Answer to Reset.*

JCVM – *Java Card Virtual Machine*.

JCRE – *Java Card Runtime Environment.*

API – *Application Programming Interface.*

JVM – *Java Virtual Machine.*

JCDK – *Java Card Development Kit.*

PIN – *Personal Identification Number*

RESUMO

Este trabalho visa a integração de duas tecnologias de controle de acesso já existentes. A intenção é proporcionar uma forma mais segura e eficaz de proteger um ambiente de acesso restrito. As tecnologias a serem integradas são *smart cards* e identificação biométrica da impressão digital.

A solução foi desenvolvida com a utilização de um computador, um leitor biométrico, uma leitora e gravadora de *smart cards* e dois *smart cards* com suporte a tecnologia *Java Card*. O sistema permite cadastrar usuários, capturando sua impressão digital e os dados cadastrais, posteriormente o usuário é vinculado a um cartão inteligente. No momento da autenticação do acesso o usuário precisa fazer a verificação da impressão digital e apresentar o *smart card*. O acesso só é liberado se a digital estiver cadastrada no sistema e se o indivíduo for o portador legítimo do cartão.

O diferencial do projeto está na unificação de tecnologias de segurança já contempladas. A posse de um cartão inteligente e a impressão digital constituem um controle de acesso baseado em dois fatores: autenticação baseada no que se possui e autenticação baseada nas características individuais.

Palavras-chave: Biometria, impressão digital, *smart cards, Java Card*, controle de acesso, segurança da informação.

ABSTRACT

This project attempts to unify two control access technologies. The objective is create a most secure way to protect a restrict environment. The technologies to be used are smart cards and biometric identification of fingerprints.

The project was developed using a computer, a biometry reader, a smart card reader and writer and two smart cards that supports the Java Card technology. The system allows users to be enrolled by catching their fingerprints and another information. The user must be enrolled to a single smart card too. When the user try to access the restrict environment, he needs to match his fingerprint with the template inserted into the base and shows his smart card. The access is allowed if his fingerprint is enrolled and if him is the legitimate owner of the smart card.

The projects main characteristic is the technologies union. Use a smart card and the fingerprint to access an environment produces a two factor control system based on what the person haves and who the person are.

Key words: Biometry, fingerprint, *smart cards, Java Card*, access control, information security.

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO

Qualquer informação de caráter essencial para os negócios de interesse de uma Empresa é digna de proteção e preservação. É um ativo, ou seja, "é um coisa que tenha valor para a organização"(ABNT 27002, 2007). Tudo aquilo que é importante precisa de proteção.

> A informação pode existir em diversas formas. Pode ser impressa ou escrita em papel, armazenada eletronicamente, transmitida pelo correio ou meios eletrônicos, apresentada em filmes ou falada em conversas. Seja qual for a forma apresentada, ou o meio através do qual a informação é compartilhada ou armazenada, é recomendado que ela seja sempre protegida adequadamente. (ABNT 27002, 2007)

Tomando como exemplo o caso de um banco, pode-se imaginar a perda de informações tais como os valores presentes em cada conta corrente. Pode-se considerar ainda o vazamento das informações de um novo projeto de veículo de uma grande montadora de automóveis. Enfim, são inúmeras as situações em que um desvio ou perda desse ativo pode ocasionar em um desastre.

A sociedade, da maneira que é organizada hoje, permite que a troca de informações ocorra de maneira simples, rápida e eficaz. Esse avanço tecnológico traz tantos benefícios quanto traz problemas. As ameaças e vulnerabilidades preocupam aqueles que desejam manter seguros seus bens preciosos.

A necessidade de segurança para as informações passa a ser vital para a sobrevivência de qualquer organização do mundo moderno. É impossível atuar como mero espectador nessa guerra de ameaças e tentativas de proteção.

A adoção de métodos de segurança da informação torna-se obrigatória para que ocorra a "preservação da confidencialidade, da integridade e da disponibilidade da informação; adicionalmente, outras propriedades, tais como autenticidade, responsabilidade, não repúdio e confiabilidade, podem também estar envolvidas."(ABNT 27002, 2007)

Sendo assim fica evidenciada a importância da proteção dos ativos e de um bom sistema de controle de acesso à ambientes onde são guardadas essas informações, seja um ambiente físico ou virtual.

Este trabalho visa o estudo da integração de duas soluções de controle de acesso já conhecidas e consolidadas no mercado: *Smart Cards* e autenticação biométrica. Os *smart cards* já fazem parte do dia a dia das pessoas. Cartões de crédito, cartões de bancos e alguns documentos de identificação possuem essa tecnologia. Identificadores biométricos são encontrados em catracas, pontos eletrônicos de empresas, cofres, entre outros. A união da complexidade dos registros biométricos com a portabilidade e confiabilidade dos cartões inteligentes é o diferencial para a criação dessa solução de controle de acesso à ambientes restritos.

1.1 Motivação

São muitas as notícias na mídia sobre roubo de informações, vazamentos de dados sigilosos e, principalmente, casos de clonagem de cartões de bancos ou utilização de cartões por terceiros não autorizados pelo titular. Todas essas situações são exemplos claros de pessoas que conseguiram acessar um ambiente restrito e certamente protegido.

Isso mostra a evolução das ameaças e a necessidade de um controle de acesso mais seguro. Juntar dois sistemas consolidados na área de segurança de informação de modo a produzir uma ferramenta confiável e adaptável a diversas situações, essa é a principal ideia motivadora desse projeto.

1.2 Objetivos

O objetivo principal é elaborar uma solução de controle de acesso, seja ela a um sistema computadorizado ou a um ambiente físico, baseada na utilização de *Smart Cards*. A autenticação utilizada pelo *Smart Card* para liberação do acesso será feita através de leitura biométrica da impressão digital do usuário.

Como objetivos específicos:

- Desenvolvimento do *software* para gravar as informações no *chip* do *Smart Card*;
- Desenvolver um *software* que permita a integração entre o leitor biométrico e o cartão inteligente. Serão compreendidas as operações de leitura, escrita e validação das informações;
- Simulação de um ambiente de acesso restrito;
- Integração do *software*, módulo de gravação e leitor biométrico com o ambiente restrito.

1.3 Estrutura da Monografia

A monografia está distribuída da seguinte forma:

Capítulo 1: Introdução. Parte que trata da motivação do trabalho, objetivos gerais e específicos e como a monografia foi estruturada.

Capítulo 2: Referencial Teórico. Oferece o embasamento teórico da proposta de solução do problema. São abordados os temas de biometria e *smart cards.*

Capítulo 3: Problemas de Segurança da Informação. Explanação sobre o problema que o trabalho se propõe a tratar. Aborda assuntos como fraudes bancárias, roubo de informações e problemas no controle de acesso. Também são mostradas algumas ferramentas e técnicas utilizadas por entidades maliciosas. Os principais dispositivos de segurança utilizados atualmente também são tema desse capítulo.

Capítulo 4: Desenvolvimento do Projeto. A parte de projeto de software e hardware é abordada nesse item. Todo o procedimento, metodologia, tecnologias e equipamentos são

mostrados e detalhados. A estimativa de custos do projeto também esta presente neste capítulo.

Capítulo 5: Testes e Resultados. Apresentação dos casos de testes efetuados para comprovar o funcionamento do dispositivo. É apresentado também a análise dos resultados.

Capítulo 6: Conclusão. Síntese conclusiva do projeto final e sugestões para trabalhos futuros.

CAPÍTULO 2 – REFERENCIAL TEÓRICO

2.1 – Biometria

A biometria visa reconhecer uma pessoa com base em características físicas e biológicas. De acordo com PINHEIRO podemos definir biometria como:

> A ciência da aplicação de métodos de estatística quantitativa a fatos biológicos, ou seja, é o ramo da ciência que se ocupa da medida dos seres vivos (do grego bio = vida e métron = medida). Resumindo, a biometria reconhece um indivíduo pelas suas características humanas mensuráveis e assim autentica a identidade de um indivíduo. (PINHEIRO, 2008)

Todas as pessoas apresentam características únicas e mensuráveis. Impressões digitais, íris, voz e topologia da face são exemplos do que pode ser utilizado para diferenciar indivíduos. Obviamente que para efetuar comparações e autenticações é necessário o auxílio de processos automatizados ou de profissionais capacitados.

Historicamente há relatos da utilização de técnicas que remetem à biometria desde os tempos do Egito Antigo. Ainda segundo PINHEIRO, "no século II a.C., os governantes chineses já usavam as impressões digitais para lacrar documentos" (PINHEIRO, 2008). O primeiro método de identificação biométrica, reconhecido oficialmente, foi elaborado pelo francês Alphonse Bertillon no final do século XIX. A figura a seguir retrata como eram feitas as medidas e comparações:

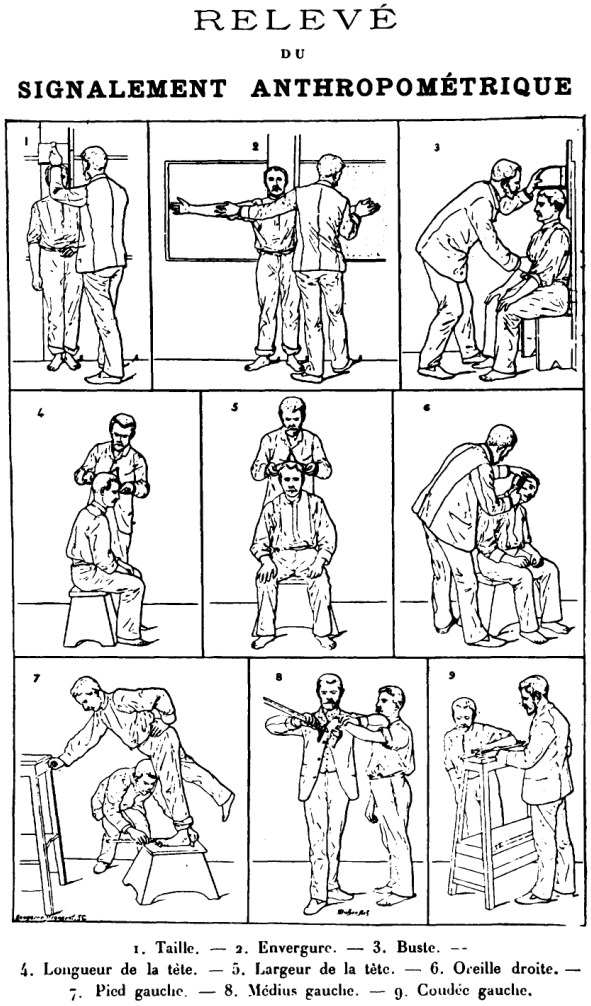

Figura 2.1 – Exemplos do Método de Bertillon / Fonte: PINHEIRO, 2008.

Atualmente a utilização da biometria se dá através de técnicas muito mais modernas e robustas. Identificadores biométricos são encontrados em diversos equipamentos, principalmente os leitores de impressão digital. Inclusive, nos próximos anos, a expectativa da indústria tecnológica é de que empresas dos mais diversos setores adotem dispositivos que identifiquem as pessoas a partir de seus traços físicos ou comportamentais, com o objetivo de aumentar a proteção às seus ativos.

2.1.1 – Métodos de Autenticação

Para realizar um controle de acesso através da identificação e autenticação de um usuário podem ser utilizados um dos três métodos abaixo descritos. Também é possível combinar dois ou até mesmo os três métodos. Eles possuem características e níveis de segurança diferentes e quem define qual deverá ser utilizado é o responsável pela política de segurança da empresa. PINHEIRO define os três métodos da seguinte forma:

- Autenticação Baseada no que se Conhece (O que você sabe?): Trata-se da autenticação baseada em algo que o usuário do sistema conheça. Nessa categoria enquadram-se os nomes de acesso, as senhas e as chaves criptográficas. É o que oferece o mais baixo nível de segurança.

- Autenticação Baseada no que se Possui (O que você tem?): Trata-se da autenticação baseada em algo que o usuário do sistema possua. O dispositivo de posse do usuário pode ser dotado de memória ou dotado de algum tipo de processamento. *Tokens* e *Smart Cards* são exemplos da utilização desse método. É o segundo em termos de nível de segurança.

- Autenticação Baseada nas Características Individuais (O que você é?): Trata-se da autenticação baseada em algo que o indivíduo é. Pode ser uma medida fisiológica, uma característica comportamental ou um padrão ou atividade específica que distingue o indivíduo, de forma confiável, de outros seres humanos, e que pode ser utilizado para autenticar sua identidade. Exemplos: Impressão digital, geometria da mão, íris, assinatura, fala, entre outros. É o que oferece o nível mais alto de segurança entre os métodos.

No projeto proposto serão combinados dois métodos: o baseado nas características do usuário e o baseado no que o usuário possui. Essa combinação deve-se ao fato da utilização do *Smart Card* com o leitor biométrico e o objetivo é agregar mais segurança ao ambiente de acesso restrito.

2.1.2 – Requisitos de um Sistema Biométrico

Como os sistemas biométricos são baseadas em características inerentes ao ser humano, é necessários que o sistema atenda três requisitos básicos, conforme descreve PINHEIRO:

- A característica biométrica deve conter diferenças significativas entre indivíduos diferentes;
- As características devem ser estáveis durante o período de vida do indivíduo;
- O sistema deve ser robusto e oferecer segurança contra tentativas de fraudes.

Já as características humanas a serem utilizadas no sistema biométrico precisa satisfazer os seguintes requisitos, ainda segundo PINHEIRO:

- Universalidade: Todos os indivíduos devem possuir a característica a ser utilizada como medida;
- Singularidade: Indica que a medida da característica utilizada não deve ser igual em pessoas diferentes, ou, no mínimo, que a probabilidade de haver duas pessoas com a mesma medida dessa característica seja muito pequena;
- Permanência: Cada característica não deve variar com o tempo, por exemplo, alterar-se devido ao envelhecimento;
- Mensurabilidade: Pode ser medida quantitativamente, tendo por base um modelo da característica selecionada.

Por fim, mais três características são essenciais para garantir o funcionamento de um sistema biométrico. PINHEIRO as define da seguinte forma:

- Precisão e Desempenho: Refere-se à precisão com que se realiza a identificação, aos recursos necessários para a medição e aos fatores ambientais que afetam a precisão. O desempenho do sistema dependerá da precisão dos resultados obtidos tanto na coleta quanto na autenticação;

- Aceitabilidade: Esta característica indica o nível de aceitação do sistema de reconhecimento biométrico por parte de seus usuários;

- Proteção: Consiste na facilidade ou dificuldade de usuários burlarem o sistema.

2.1.3 – Componentes de um Sistema Biométrico

De acordo com PINHEIRO são componentes de um Sistema Biométrico:

- Subsistema Interface de usuário (Sensor): É o conjunto de elementos que contém o dispositivo ou sensor que capta a amostra biométrica do indivíduo e a converte em um formato adequado para ser utilizado. A qualidade da amostra coletada afeta diretamente o desempenho do resto do sistema.

- Subsistema Estação de Controle (Cérebro): É responsável pelas funções de controle dos dispositivos e também por receber a amostra biométrica fornecida pelo subsistema de interface de usuário e convertê-la em uma forma adequada para o processamento pelo módulo de comparação.

- Subsistema Comparador (Comunicações e Processamento): Faz a comparação da amostra biométrica apresentada com o *template* da base de dados. Aqui é feita a comparação para verificar se as amostras são similares. Também é decidido se a amostra coletada é similar à amostra do usuário constante na base de dados. A decisão de autenticidade ou falsidade é repassada aos outros elementos do sistema pelo subsistema comparador.

- Subsistema de Armazenamento (Banco de Dados): Este módulo mantém os *templates* dos usuários cadastrados no sistema biométrico. Os *templates* podem ser armazenados em cartões de memória, em Banco de Dados centralizado ou em cartões magnéticos, *Smart Cards, tokens*, entre outros. O tipo de armazenamento se dará pela aplicabilidade a que se destina o sistema biométrico.

2.1.4 – Funcionamento de um Sistema Biométrico

Cumpridos os requisitos e premissas de um sistema biométrico é possível passar para o seu funcionamento. Em um primeiro momento é necessário o cadastramento dos usuários em uma base de dados. O usuário deve fornecer a característica física escolhida para a captura. A captura é feita através de um leitor biométrico preparado para tal. O dado capturado é então enviado para uma central de processamento onde é tratado e enfim inserido na base de dados. O registro do usuário é mantido para a verificação futura.

O processo de verificação também se inicia com o fornecimento da característica física do usuário. O leitor biométrico também é utilizado para captura. O dado capturado é processado e, ao invés de ser inserido na base de dados, é comparado com o registro anteriormente inserido no banco de dados. Caso a comparação confirme a similaridade entre os registros o acesso é concedido. Caso contrário é negado.

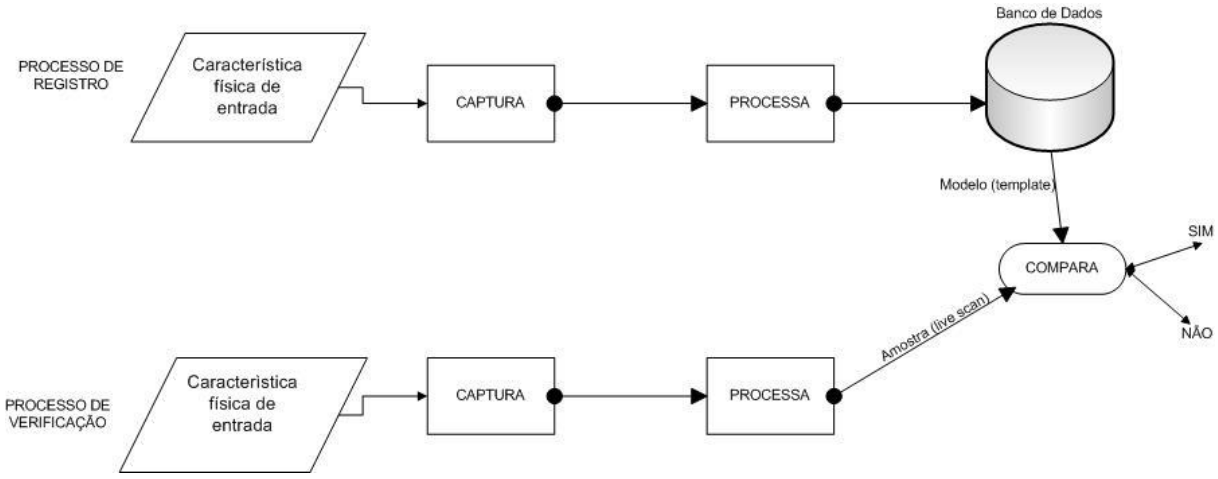

Figura 2.2 – Funcionamento de um Sistema Biométrico / Fonte: O autor.

2.1.5 – Tipos de Biometria

São várias as características que podem ser utilizadas em sistemas biométricos. Elas podem ser divididas em dois grupos: Elementos Biométricos Fisiológicos e Elementos Biométricos Comportamentais. No grupo dos fisiológicos encontramos a face, as mãos, os

dedos, a íris e a orelha. Já no grupo dos comportamentais podemos citar a voz, a escrita e a assinatura.

Os tipos de biometria se diferem em outros aspectos além da questão física ou comportamental. Alguns tipos são mais utilizados, outros são mais fáceis de se utilizar e outros são mais invasivos para os usuários. Além disso podem se diferir no aspecto da confiabilidade e precisão e principalmente em relação aos custos para implementação. O mais importante é conhecer cada tipo para se decidir qual o ideal para a situação ou informação a ser protegida.

São exemplos de tipos de biometria:

- Reconhecimento Facial: Considera as medidas do rosto que nunca se alteram, mesmo que o indivíduo seja submetido a cirurgias plásticas. Essas medidas básicas são: distância entre os olhos, distância entre boca, nariz e olhos e distância entre olhos, queixo, boca e linhas dos cabelos. É um dos sistemas menos intrusivos dentre os existentes. Suas desvantagens são a menor confiabilidade, maior tempo de leitura e pesquisa das informações e o alto custo de implementação. (PINHEIRO, 2008)

- Geometria da Mão: Nesse método é utilizado um scanner especialmente desenhado para capturar as medidas das palmas das mãos, das mãos e dos dedos a partir de uma perspectiva tridimensional. É um sistema com baixa confiabilidade porque as características capturadas não são suficientemente descritivas para a identificação. Há ainda problemas com a presença de anéis e o indivíduo precisa encaixar corretamente a mão no equipamento de leitura. O custo de implementação é baixo. (PINHEIRO, 2008)

- Identificação pela Íris: A íris é a parte colorida do olho, em torno da pupila, e permite 249 pontos de diferenciação que podem ser usados no processo de reconhecimento de um indivíduo. Trata-se de um identificador biométrico estável, pois suas características não são alteradas pelo envelhecimento. O reconhecimento apresenta alto grau de precisão e um alto desempenho no processo de verificação. A desvantagem se dá na difícil captura e no alto custo de implementação. (PINHEIRO, 2008)

- Reconhecimento de Voz: O reconhecimento pela voz é uma tecnologia que analisa os padrões harmônicos e não apenas reproduções de sequências predefinidas de voz. Para capturar a voz pode ser utilizado um microfone. Entre as vantagens desse tipo estão o fato de a voz ser natural e simples para o ser humano. Como desvantagem alguns sistemas podem obrigar o usuário a falar mais alto ou até mesmo repetir várias vezes uma mesma frase, o que pode representar na demora do cadastramento ou validação dos dados. Além disso o sistema é sensível a ruídos do ambiente e estados físicos do indivíduo como gripe ou estresse. (PINHEIRO, 2008)

- Assinatura: Muito utilizada em Bancos e cartórios. A assinatura manuscrita é altamente aceita como forma de identificação e autorização de um indivíduo. O estudo que compreende a análise de padrões e características de uma assinatura chama-se grafoscopia. Na análise de uma assinatura são levados em conta a pressão sobre a caneta, direção e sentido da escrita e velocidade da escrita. A chave do sucesso de um sistema de reconhecimento de assinaturas é encontrar características constantes. É um sistema de baixo custo. (PINHEIRO, 2008)

- Impressão Digital: De todas as características biométricas, a impressão digital é a mais estudada. As impressões digitais são únicas para cada indivíduo e consideradas o tipo biométrico mais seguro para determinar a identidade, depois do teste do DNA. Esse método é baseado na identificação através das irregularidades das impressões digitais, retiradas de um ou mais dedos, as chamadas minúcias (vide figura 2.3). O reconhecimento por impressões digitais requer o uso de um dispositivo capaz de capturar (vide figura 2.4), com um bom grau de precisão, as minúcias, além de um *software* que trate a imagem capturada e faça o reconhecimento da digital. É o tipo biométrico mais utilizado em todo mundo, pouco invasivo ao usuário e de baixo custo. (PINHEIRO, 2008)

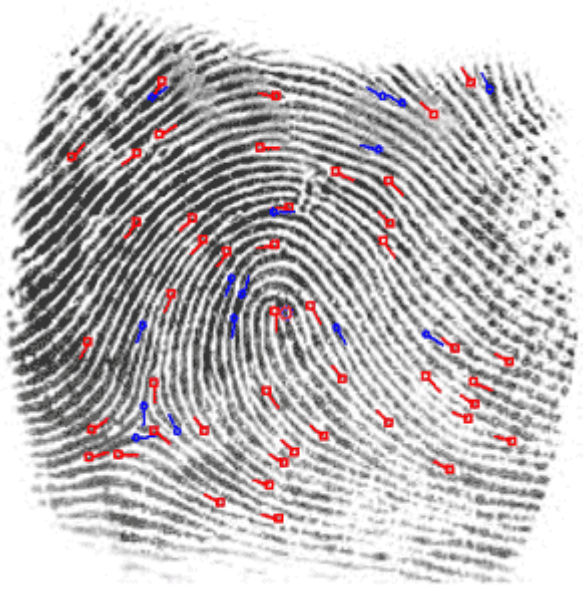

Figura 2.3 – Minúcias da Impressão Digital / Fonte: Site Biometric Group

Figura 2.4 – Leitor Biométrico Nitgen Hamster I / Fonte: Site Nitgen

Por fim segue um comparativo entre os tipos biométricos mais reconhecidos e utilizados atualmente. São evidentes as vantagens e grande aplicabilidade da impressão digital. Tal motivo serve como justificativa para a escolha desse tipo biométrico para o desenvolvimento do projeto em questão.

CARACTERÍSTICAS	IMPRESSÃO DIGITAL	RETINA	GEOMETRIA DA MÃO	ÍRIS	FACE	ASSINATURA
FACILIDADE DE USO	ALTA	BAIXA	ALTA	MÉDIA	MÉDIA	ALTA
INCIDÊNCIA DE ERROS	SECURA, SUJEIRA E IDADE	LENTES E ÓCULOS	DANOS NA MÃO E IDADE	POUCA LUZ	LUZ, IDADE, ÓCULOS E CABELO	MUDANÇA DE ASSINATURA
PRECISÃO	ALTA	MUITO ALTA	ALTA	MUITO ALTA	ALTA	ALTA
ACEITAÇÃO DO USUÁRIO	ALTA	MÉDIA	ALTA	MÉDIA	MÉDIA	ALTA
NÍVEL DE SEGURANÇA GARANTIDA	ALTA	ALTA	MÉDIA	MUITO ALTA	MÉDIA	MÉDIA

Fonte: A Pratical Guide To Biometric Security Technology - Simon Liu Anda Mark Silverman

Quadro 2.5 – Comparativo entre Tipos Biométricos

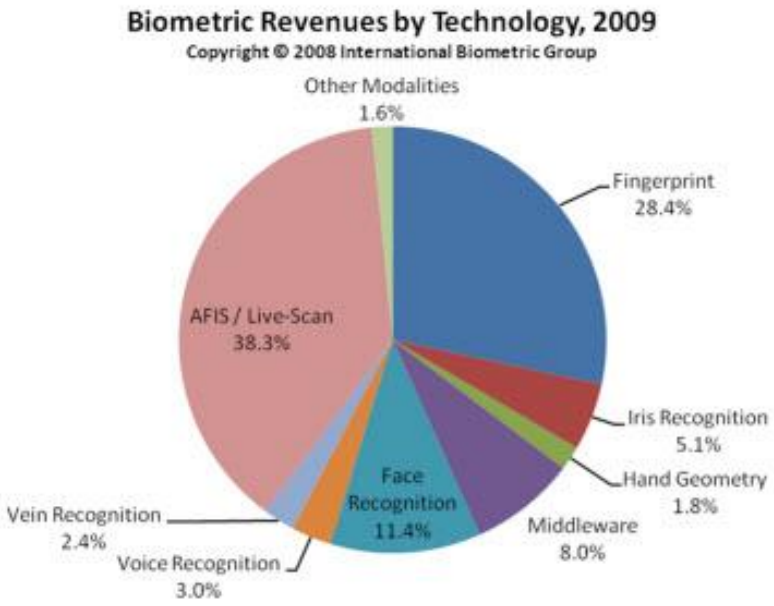

Figura 2.6 – Situação dos Principais Sistemas Biométricos no Mercado / Site Biometric

A situação dos principais sistemas biométricos no mercado também reforça a preferência pelos sistemas de impressão digital. O sistema AFIS também é baseado em impressões digitais. Ele possui uma base bastante robusta e completa e forças policiais de vários países a utilizam. Assim, de acordo com a pesquisa do Grupo Internacional de Biometria, os sistemas biométricos baseados em digitais compreendem 66,7% do total.

2.2 – *Smart Cards*

Os cartões inteligentes se assemelham bastante com um cartão de crédito comum, a começar pelo tamanho e formato. A principal diferença é que o *Smart Card* é capaz de processar e armazenar informações através de um microprocessador inserido no próprio plástico do cartão. Além disso oferecem portabilidade, segurança e independência de bases de dados no momento da transação. Apesar de não ser uma tecnologia nova é bastante utilizada atualmente, principalmente por instituições financeiras.

2.2.1 – Histórico do *Smart Card*

A ideia de inserir um circuito integrado em um cartão vem do final da década de 70. Dois inventores alemães, Jurgen Dethloff e Helmut Grotupp, foram responsáveis pelo feito. No Japão, Kunitaka Arimura patenteou uma invenção semelhante. No entanto, o progresso dessa tecnologia se deu a partir das patentes registradas por Roland Moreno em 11 países, entre eles a França, durante os anos de 1974 e 1979. Foram essas patentes que deram início à fabricação de *Smart Cards* em escala industrial.

A primeira aplicação que utilizava um cartão inteligente foi registrada na França. Foi no ano de 1984 e a ideia era substituir as fichas pelos cartões nos telefones públicos. Três anos mais tarde a Alemanha também apostou na mesma aplicação.

E mesmo com todas as limitações tecnológicas da época, principalmente no que se refere a técnicas de criptografia e capacidade de processamento, os *Smart Cards* não caíram no esquecimento e continuaram sendo objetos de estudo e pesquisa.

A aposta deu resultado e hoje é claramente perceptível a utilização desses cartões nas mais diversas áreas. Cartões de crédito, crachás para controle de acesso, autenticadores de televisão à cabo ou à satélite são exemplos da utilização dessa tecnologia. Entretanto a maior comprovação do sucesso dessa aplicação está na área de comunicação. Os celulares que utilizam a rede GSM precisam de um *chip*, conhecido por *SIM Card*, que nada mais é do que um *Smart Card* de tamanho reduzido.

2.2.2 – Componentes de um *Smart Card*

Toda a aparência física e propriedades de um cartão inteligente são definidas pela ISO 7816. Abaixo, a aparência física de um *Smart Card:*

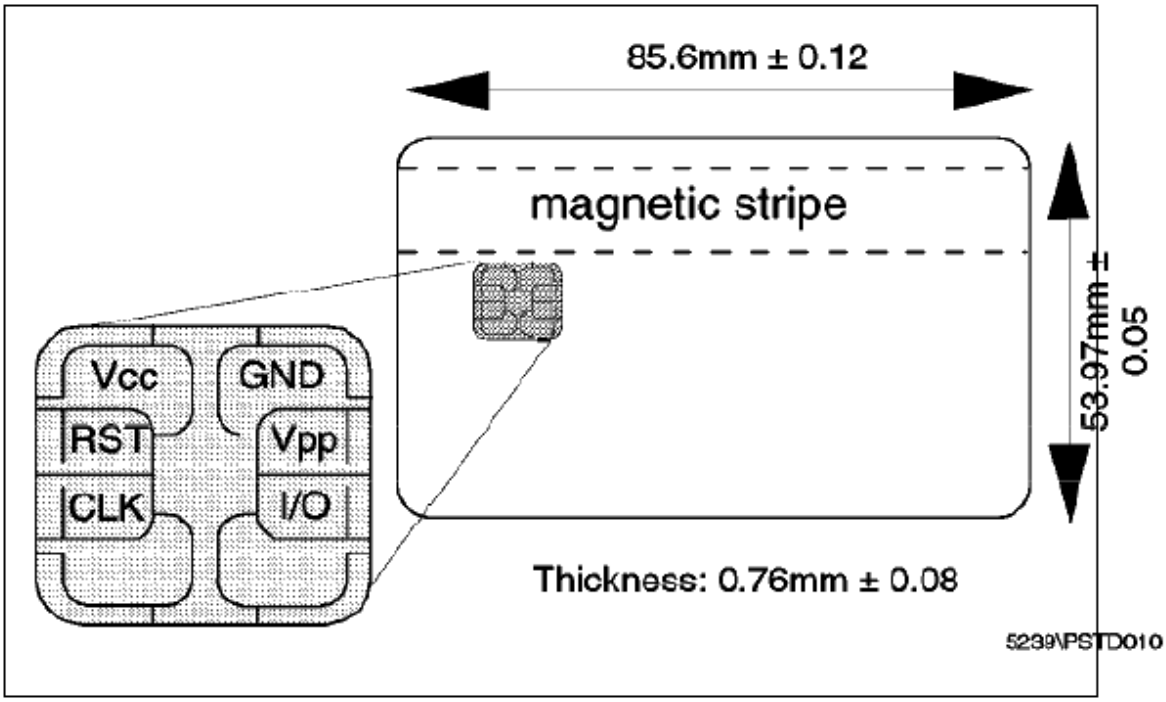

Figura 2.7 – Aparência Física de um *Smart Card* / Fonte: CHEN, 2000.

É forte a semelhança com um cartão de crédito comum. O tamanho e a espessura são praticamente os mesmos. A diferença é apenas a presença do circuito integrado embutido no cartão.

O *Smart Card* possui oito pontos de contato em seu *chip*. CHEN os define da seguinte forma:

- VCC: É o que fornece energia ao *chip*. Varia de 3 a 5 Volts;
- RST: É utilizado para enviar um sinal de reinicialização para o microprocessador;
- CLK: Fornece o sinal de *clock* externo, o qual o sinal de *clock* interno é derivado;
- GND: Utilizado como tensão de referência. Seu valor é considerado 0 Volts;

- VPP: É opcional e apenas cartões antigos o possuem. Utilizado para alterar a voltagem durante a programação;

- I/O: É utilizado para transferir dados e comandos entre o cartão inteligente e o meio externo. O modo de transmissão é *half duplex*, o que significa que pode receber e enviar dados mas não de forma simultânea;

- RFU: São os dois pontos inferiores. Eles estão reservados para uso futuro.

Ainda de acordo com CHEN, a unidade central de processamento de um *Smart Card* normalmente compreende um microcontrolador de 8 *bits* com uma velocidade de *clock* de aproximadamente 5 MHz. Cartões mais modernos podem atingir uma velocidade de até 40 MHz. Os cartões inteligentes atuais podem ainda ter um microcontrolador de 16 ou 32 *bits*.

2.2.3 – Sistema de Memória de um *Smart Card*

Podem ser três os tipos de memória de um cartão inteligente. Ainda de acordo com CHEN, os tipos de memória mais utilizados são:

- ROM: Memória não volátil. Utilizada para armazenar os programas fixos do cartão. Não é necessário energia para acessar esse tipo de memória. Entretanto não é possível alterar o conteúdo dessa memória;

- EEPROM: Parecida com a memória ROM. Também é capaz de preservar o conteúdo gravado quando a energia é interrompida. A diferença é que esse tipo de memória pode ter seu conteúdo modificado durante a utilização do cartão. Aliás é nesse tipo de memória que os dados do usuário são gravados. Sua funcionalidade é parecida com a de um disco rígido de um computador comum;

- RAM: É a memória temporária. Como depende da energia é considerada volátil. É utilizada em operações de modificação e alocação de dados.

Há outros tipos de memórias que estão ganhando força nos cartões inteligentes. Um exemplo é a memória Flash que é muito mais rápida, eficiente e possui mais espaço que a EEPROM. Entretanto essa deve ser uma mudança perceptível somente em uma próxima geração de Smart Cards.

2.2.4 – Protocolos de Transmissão e Comunicação

A comunicação dos *Smart Cards* com o meio externo se dá através do método *half-duplex*, ou seja, há um receptor e um emissor e ambos podem emitir mensagens mas não ao mesmo tempo. No mundo dos cartões inteligentes o modelo adotado é o de mestre-escravo. O cartão faz a parte de escravo e aguarda comandos do ambiente externo. Os comandos recebidos são executados e as respostas são repassadas ao cliente. Esse comandos obedecem um protocolo chamado APDU. (CHEN, 2000)

O protocolo APDU pertence à camada de aplicação portanto é responsável por prover os serviços entre o *Smart Card* e o computador. As mensagens APDU podem ter duas estruturas diferentes, uma é utilizada pelo computador para enviar comandos para o cartão e a outra é utilizada pelo cartão para enviar respostas ao computador.

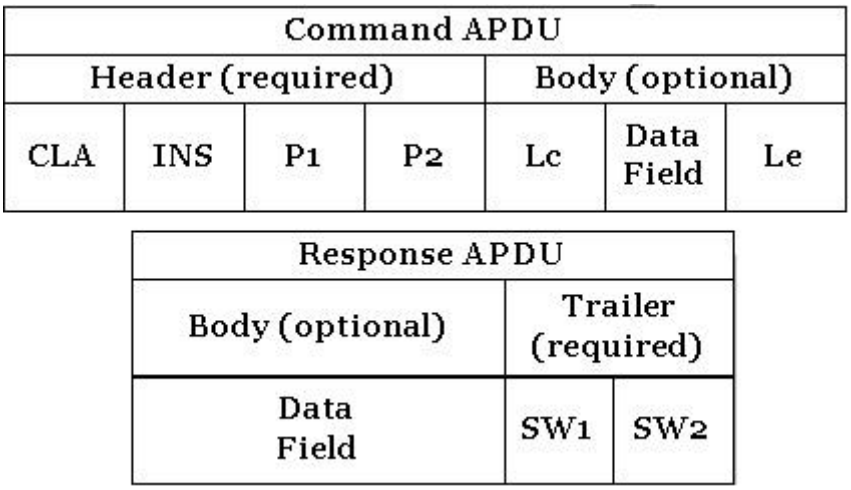

Figura 2.8 – Mensagens APDU de Comando e Resposta / Fonte: CHEN, 2000.

Cada campo de uma mensagem APDU é definido por CHEN da seguinte maneira:

- CLA: Classe de instrução. Identifica a categoria de comando e resposta de um APDU;
- INS: Código de instrução. Especifica a instrução do comando;
- P1 e P2: Parâmetros utilizados para prover qualificações para a instrução;
- LC: Campo opcional. Especifica o tamanho do campo de dados;
- DATA FIELD: Contém os dados que serão enviados ao cartão para serem executados conforme as instruções;
- LE: Número de bytes da resposta enviada pelo cartão ao computador.

Já nas mensagens de resposta, e ainda segundo CHEN, temos:

- DATA FIELD: Contém os dados da resposta. O tamanho da resposta deve estar de acordo com o especificado no campo LE da mensagem de comando;
- SW1 e SW2: Juntos formam a palavra de status. O status que vai informar se o comando foi executado corretamente ou se ocorreu algum erro.

A tabela abaixo contém alguns exemplos de status de mensagens APDU:

SW1 SW2	**Status**
90 00	OK – Comando executado com sucesso
90 01	OK – Utilizando o protocolo T=1
90 10	OK – Protocolo síncrono em utilização
60 01	Tipo de cartão não selecionado
60 02	Sem cartão na leitora
60 03	Tipo de cartão inválido

60 04	Cartão não energizado
60 05	Código de instrução inválido
60 20	Falha no cartão
60 22	Curto circuito no conector do cartão
62 01	Código secreto inválido
67 01	Comando incompatível com o tipo de cartão
67 02	Endereço do cartão inválido
67 03	Tamanho da informação inválido
67 04	Tamanho da resposta inválido
67 05	Código secreto bloqueado
67 12	Comando APDU cancelado

Tabela 2.9 – Exemplos de Mensagens de Status / Fonte: CHEN, 2000.

Para o transporte das mensagens o protocolo utilizado é o TPDU. Ele pertence à camada de transporte. CHEN menciona que os dois tipos de TPDU que estão em uso hoje são o T=0 e T=1. O T=0 faz a transmissão das mensagens entre o cartão e o computador byte por byte. Já o T=1 faz a transmissão através de blocos de bytes.

Outra informação importante sobre a comunicação dos *Smart Cards* é o ATR. CHEN define o ATR como sendo uma mensagem inicial que o cartão envia para o computador especificando todos os parâmetros necessários para que a conexão entre eles seja estabelecida. O ATR é enviado imediatamente após a energização do cartão. Ele contém 33 bytes e possui informações como a taxa de transmissão, os parâmetros de hardware do cartão, o número serial do *chip* e o número de versão.

2.2.5 – Segurança no *Smart Card*

Os cartões inteligentes possuem uma maneira muito segura de lidar com as informações restritas. Essa tecnologia de processamento no próprio cartão gera muitos benefícios do ponto de vista da segurança. Outra grande vantagem destes cartões reside na facilidade de desenvolver aplicações já que os *smart cards* possuem uma infraestrutura preparada para tal com memória e unidade de processamento. (SOUZA, 2010)

O *smart card* permite um processamento seguro fora da rede e através da função de leitura e escrita permite duas funções importantes dos sistemas de controle de acesso. De acordo com SOUZA as funções são:

- Anti dupla passagem (Anti "*Passback*"): Função utilizada para monitorar a entrada e saída de pessoas. O sistema verifica o fluxo da atividade do cartão checando suas entradas e saídas. Isso pode gerar uma regra que obrigue o usuário a registrar a entrada antes de registrar sua saída. Caso não seja obedecida, sua presença no local não será registrada e um evento de alerta pode ser gerado.

- Anti dupla passagem global: Também visa monitorar a entrada e saída de pessoas, mas em um ambiente com várias entradas e saídas.

A maioria dos sistemas que utilizam cartões para fazer o controle de acesso necessitam de uma comunicação *online* para registrar e verificar o acesso do usuário. A utilização de cartões inteligentes em sistemas que permitem a leitura e escrita no cartão tornou viável o funcionamento do *antipassback* de forma *off-line*, ou seja, sem a necessidade de uma rede de comunicação. Tudo graças ao processamento interno. (SOUZA, 2010)

Outro benefício de importante destaque destes cartões é a proteção aos dados inseridos na memória do cartão. Há algoritmos criptográficos que blindam as informações, dificultando a interceptação ou extração. Cada fabricante define qual algoritmo será implantado em seu cartão, bem como a chave criptográfica. Algoritmos de criptografia simétrica como DES e

DES Triplo e de chave assimétrica como o RSA, são exemplos de funcionalidades criptográficas presentes em muitos *smart cards*.

Todas essas vantagens dos cartões inteligentes dificultam muito o trabalho de entes maliciosos. A clonagem de cartões, por exemplo, torna-se muito mais trabalhosa e onerosa, chegando ao ponto de inibir a ação de bandidos devidos aos gastos e empenhos dedicados para clonar ou capturar a informação de um chip de *smart card*.

2.2.6 - Tecnologia *Java Card*

Os *Smart Cards* são uma das menores plataformas de computação existentes no mercado. E mais uma vez a tecnologia Java está presente. O conceito principal da linguagem no que remete a portabilidade se adequa perfeitamente ao mundo dos cartões inteligentes. "A tecnologia *Java Card* permite que programas escritos em Java rodem em *Smart Cards*." (CHEN, 2000)

A tecnologia *Java Card* possui algumas particularidades. Embora a forma de codificar, os conceitos da linguagem, a sintaxe e a semântica sejam iguais a qualquer outro programa Java, há alguns detalhes que são muito importantes para o programador. As principais exceções se concentram na máquina virtual, no JCRE, nas APIs e na forma de desenvolvimento dos *Applets*.

São particularidades da tecnologia *Java Card*:

- JCVM: A principal diferença entre a máquina virtual *Java Card* e máquina virtual Java é que a JCVM é dividida em duas partes. Uma parte está presente dentro do cartão e possui o interpretador do código em bytes. A outra parte roda no computador e possui o conversor do código. Juntas as partes implementam a função da máquina virtual que é carregar e executar as classes Java. (CHEN, 2000)

- Arquivo CAP: O arquivo CAP contém a representação binária de uma classe Java. As informações da classe, os códigos executáveis, informações de conexão e informações de verificação são compactadas nesse arquivo. Uma classe Java é a parte mais

importante da arquitetura da linguagem. Devido às limitações do Smart Card é necessária a conversão das classes Java em arquivos CAP. Dessa maneira é possível que o programa seja carregado e executado dentro do cartão inteligente. (CHEN, 2000)

- *Java Card Converter*: Para converter as classes Java em arquivos CAP é necessário o auxílio de um conversor. Durante a conversão o Converter realiza todas as tarefas que uma máquina virtual comum executaria durante o carregamento de uma classe. Portanto atividades como verificar violações da linguagem no código, inicialização das variáveis, otimização do código e alocação de espaço em memória são realizadas nesse momento. Após a conversão do arquivo .CLASS de uma classe Java é gerado o arquivo CAP. (CHEN, 2000)

- JCRE: É responsável pelo sistema que roda dentro do cartão inteligente. Toda a administração de recursos, comunicação de rede e execução do *applet*, é de responsabilidade do JCRE. (CHEN, 2000)

- API: É a biblioteca de informações. Contém várias classes customizadas para tornar possível a programação e desenvolvimento de aplicativos para *Smart Cards*. São essenciais para os programadores. (CHEN, 2000)

A figura abaixo ilustra como é a divisão entre a parte *on-card* e a parte *off-card*:

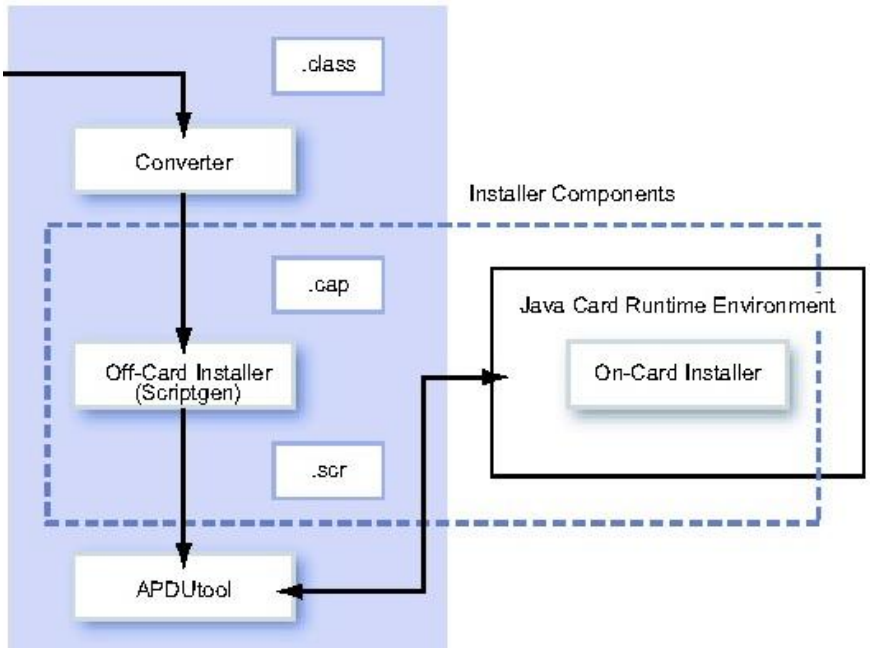

Figura 2.10 – Particularidades do *Java Card* / Fonte: CHEN, 2000.

O desenvolvimento de um aplicativo *Java Card* começa como qualquer outro programa Java. O código desenvolvido é então compilado e gerado o arquivo .CLASS. Para ser compatível com as limitações do cartão inteligente é feita uma conversão e assim obtêm-se o arquivo CAP. O arquivo CAP é inserido dentro do *Smart Card* e o JCRE cuida do restante.

Quando da instalação do aplicativo no cartão inteligente pelo JCRE não é possível fazer alterações dinâmicas. Por consequência os *applets* carregados no cartão devem referenciar classes Java existentes no cartão. Não é possível instalar novas classes Java durante a execução do aplicativo. Eventuais mudanças ou necessidades de novas classes tornam necessário a geração de um novo arquivo CAP a ser instalado no *Smart Card*.

Neste projeto será utilizado um *kit* de desenvolvimento *Java Card* disponibilizado pela Sun em seu *site* na internet. O JCDK 2.2.1 possui toda a API necessária para o desenvolvimento dos aplicativos, bem como o conversor para gerar o arquivo CAP.

CAPÍTULO 3 – PROBLEMAS DE SEGURANÇA DA INFORMAÇÃO

3.1 - Identificação do Problema e Cenário Atual

No Brasil há várias instituições financeiras de grande porte e, consequentemente, uma base de clientes bancários considerável, que por sua vez, utiliza os serviços e canais de atendimento que os bancos oferecem. De acordo com dados divulgados pela Federação Brasileira de Bancos – FEBRABAN, o número de clientes, cartões de crédito e de usuários do internet *banking* é:

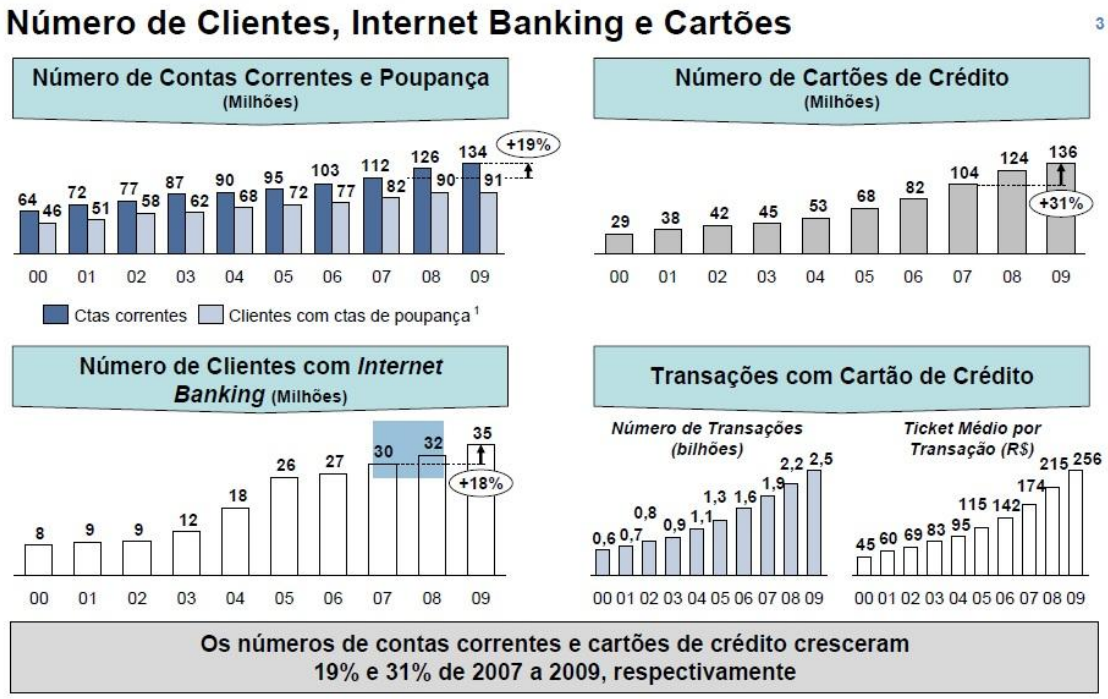

Figura 3.1 – Número de Clientes, Internet *Banking* e Cartões / Fonte: FEBRABAN

O Banco do Brasil, por exemplo, segundo dados do próprio sítio da empresa, tem uma base de 52,7 milhões de clientes e obteve um lucro líquido de R$5,1 bilhões no primeiro semestre de 2010. Essas informações mostram o quanto é alta a movimentação financeira no país.

Todo esse montante de ativos desperta a cobiça de quadrilhas especializadas. Ainda segundo a pesquisa divulgada pela FEBRABAN em 31 de Agosto de 2010, os bancos

brasileiros devem somar R$900 milhões em prejuízo até o final de 2010 somente levando-se em conta crimes eletrônicos. Em 2009 o prejuízo foi semelhante. A maior parte das fraudes bancárias envolvem cartões de crédito (45%), seguido de golpes via internet *banking* (30%) e cartões de débito (20%). O restante se divide por outros canais.

O cenário não é pior porque as instituições investiram aproximadamente R$19,4 bilhões em Tecnologia da Informação e R$1,4 bilhões em Segurança, o que prova a preocupação com os acontecimentos. A figura abaixo ilustra os gastos:

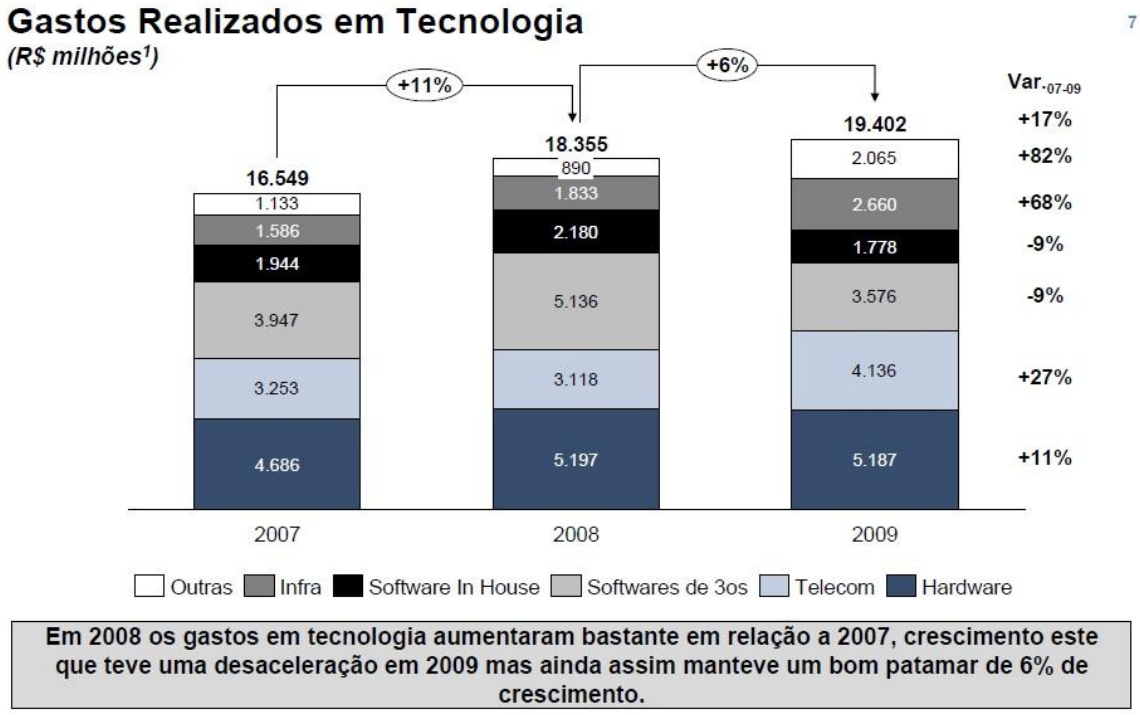

Figura 3.2 – Gastos realizados em Tecnologia / Fonte: FEBRABAN

Outra situações também podem ser dadas como exemplo. Em Agosto de 2008 um contêiner da Petrobrás foi arrombado e foram furtados dados sobre pesquisas sísmicas, que incluíam a descoberta de petróleo e gás. Outra situação aconteceu em Agosto do presente ano, quando dados de 12 milhões de inscritos no ENEM desde 2007 vazaram na internet. Mais recente ainda é o caso do acesso indevido aos dados do imposto de renda de Verônica Serra, filha de José Serra, candidato à presidência da República em 2010. Aliado aos fatos expostos ainda há os casos de venda de mídias com declarações de imposto de renda de vários contribuintes, venda de cadastros, desvio de informações dentro de corporações, roubo de provas de concursos públicos e vestibulares e muitos outros casos.

Todas as situações apontam para uma única diretriz: a fragilidade do elo ser humano e informação. No caso de roubos de senhas a maioria das ocorrências deve-se ao fato de senhas de fácil combinação, guarda inapropriada do código secreto ou divulgação do mesmo a parente, amigos e companheiros. As situações de furto mostram fragilidade no controle de acesso ao ambiente restrito. Já o desvio e venda de informações caracteriza claramente o acesso por pessoas não autorizadas e a falta de registro dos acessos.

A alternativa para aqueles que desejam manter seus dados seguros é investir em equipamentos de Segurança da Informação. A proposta deste projeto é integrar dois equipamentos de segurança.

3.2 – Ferramentas e Técnicas Maliciosas

Os motivos para atacar, invadir e roubar informações são os mais diversos. Há quem faz por diversão, quem faz por visibilidade social, quem faz por vingança, que faz por questões financeiras e quem faz por motivos políticos. Para atingirem seus objetivos, as entidades maliciosas se aproveitam de brechas e muitas vezes utilizam ferramentas para capturar o que desejam. São exemplos dessas ferramentas:

- *Keyloggers*: *Software* que captura as teclas digitas no computador. Atualmente, o termo foi expandido para incluir vários softwares de monitoração completa do sistema, tais como o *Perfect Keylogger*. (LINHA DEFENSIVA, 2010)

- *Phishing*: Mensagens fraudulentas que tentam se passar por avisos reais de grandes empresas, como bancos, antivírus e cartões de crédito. Mensagens desse tipo possuem um *link* que aponta para algum *site* malicioso ou contém algum programa como *keyloggers* ou *trojans*. As mensagens podem ser enviadas via e-mail ou bate-papos. (LINHA DEFENSIVA, 2010)

- Vírus: É todo programa de computador que funciona como parasita, infectando os arquivos que existem em um computador. Utilizam a rede para se espalhar. (LINHA DEFENSIVA, 2010)

- *Denial Of Service*: Em português é conhecido como Negação de Serviço. Trata-se de uma atividade maliciosa através da Internet que busca derrubar um serviço de um servidor. O "Serviço" se refere à algum serviço da internet, tal como páginas da internet. *Sites* atingidos por ataques de negação de serviço ficam indisponíveis. (LINHA DEFENSIVA, 2010)

- *Trojans*: Um *Trojan Horse* é um programa que faz algum tipo de atividade maléfica como capturar informações do usuários e enviá-las para um computador receptor. Dados bancários são frequentemente roubados com a ajuda dos cavalos de tróia. (LINHA DEFENSIVA, 2010)

- Chupa-cabra: Acessórios adicionados aos leitores de cartão do banco. São como um acabamento adicional que se sobrepõem ao leitor de cartão, e que possuem um dispositivo de leitura adicional que grava os códigos da tarja magnética do plástico e as envia para um computador receptor. Os bandidos utilizam ainda uma micro câmera para capturar as senhas digitadas pelos clientes. Com as senhas e os dados do cartão é possível criar clones e efetuar compras e saques. (LINHA DEFENSIVA, 2010)

Figura 3.3 – Chupa-cabra parte externa / Fonte: Site Linha Defensiva

Figura 3.4 – Chupa-cabra parte interna / Fonte: Site Linha Defensiva

Embora eficazes, as ferramentas só são eficientes devido a técnica mais utilizada e difundida desde os primórdios da humanidade, a Engenharia Social.

> A habilidade de se manipular pessoas para obter informações necessárias para conseguir acessar um sistema, roubar dados de bancos ou qualquer outra coisa. Visa manipular sentimentos, emoções e aspirações das pessoas para se obter informações privilegiadas, as quais normalmente não seriam entregues. Os meios para a técnica acontecer podem ser vários: Internet, telefone, carta ou mesmo pessoalmente. O que é mais manipulado pelos engenheiros sociais: Curiosidade, Confiança, Simpatia, Culpa, Medo. Toda a base dos ataques se encaixa em uma dessas categorias. (MITINICK, 2010)

Os estelionatários estão sempre aprimorando seus golpes e os *crackers*, ou piratas virtuais, não param de lançar novas versões das ferramentas e até mesmo novos programas maliciosos. Empresas e políticos também usufruem de técnicas ilegais para confrontar seu concorrentes. Funcionários insatisfeitos, maridos traídos e até jovens viciados em drogas podem ser protagonistas nesse cenário criminoso. Ou seja, não faltam motivações e interessados em invadir ou roubar. Proteção é algo extremamente necessário.

3.3 – Dispositivos de Segurança Utilizados Atualmente

As técnicas e ferramentas maliciosas estão em constante desenvolvimento e, se os bandidos evoluem, os prejudicados não podem ficar inertes. Há muitos dispositivos que se consolidaram no mercado por atender as necessidades dos usuários, diminuindo fraudes, protegendo informações e evitando acessos não autorizados. Abaixo são detalhados alguns desses dispositivos:

- Cartões bancários com *chip*: O desenvolvimento dos cartões de crédito baseados em *smart cards*, pode ser considerado uma mudança fundamental na indústria mundial dos sistemas de pagamento. Os cartões que utilizam *chip* tem capacidade de armazenar dados de forma segura (criptografados) e tem uma maior capacidade de memória. Além disso, os cartões com *chip* não podem ser clonados, pelo menos não com meios simples. (PARODI, 2010)

- Leitores biométricos: Catracas, salas de servidores e computadores pessoais são exemplos do que pode ser protegido por esse sistema complexo e que visa identificar o usuário com base em perfis e características biométricas.

- Cartões de proximidade RFID: Também bastante utilizado em catracas, portarias de prédios residenciais ou comerciais, garagens e vale transporte. Funciona com uma fonte, normalmente chamada de antena, que emite uma onda de rádio frequência. Dentro do cartão há outro componente, chamado *transponder*. Ao receber o sinal da antena, o *transponder* responde emitindo um sinal que é identificado pela fonte emissora. As informações transmitidas e as respostas dependem do tipo de serviço executado. (MARX TECNOLOGIA, 2010)

- Certificado Digital: O Certificado Digital é uma credencial que identifica uma entidade, seja ela empresa, pessoa física, máquina, aplicação ou *site* na *web*. O arquivo de computador gerado pelo Certificado Digital contém um conjunto de informações que garante a autenticidade de autoria na relação existente entre uma chave de criptografia e uma pessoa física, jurídica, máquina ou aplicação. Os Certificados Digitais são compostos por um par de chaves (chave pública e

> privada) e a assinatura de uma terceira parte confiável - a Autoridade Certificadora – AC. As Autoridades Certificadoras emitem, suspendem, renovam ou revogam certificados, vinculando pares de chaves criptográficas ao respectivo titular. (CERTISIGN, 2010)

Embora sejam apenas alguns exemplos e existam muito mais aplicações e equipamentos de segurança, o cenário atual demanda o estudo contínuo e o desenvolvimento de novos aplicativos. Aproveitando o reconhecimento do mercado e o fato de permitir a criação de um sistema de autenticação de dois fatores (o que você tem e o que você é) optou-se pelo estudo da integração de dois dos dispositivos citados a cima: *Smart card* e leitor biométrico.

CAPÍTULO 4 – DESENVOLVIMENTO DO PROJETO

4.1 - Proposta para Solução do Problema

Para desenvolver a solução integrada serão necessários um leitor biométrico, para capturar a impressão digital do usuário, uma leitora e gravadora de *smart card* para proporcionar a gravação das informações no cartão inteligente, e um *smart card*.

A integração das tecnologias se dará através de um software. Uma interface será responsável pelo e captura da impressão digital. Após capturada a digital será convertida em um código único de caracteres e é esse código que será utilizado posteriormente para a validação do acesso. Os dados básicos do cliente e o código da impressão digital serão inseridos em um banco de dados.

No momento da inserção no banco de dados, as informações básicas do usuário serão concatenadas com o código do usuário e com o ATR do cartão inteligente. A *string* gerada pela concatenação será convertida em um *hash code* através do algoritmo MD5 e posteriormente gravada na base de dados. Esse código *hash* será inserido na memória do *smart card* para posteriormente identificar o dono do cartão.

Quando o usuário inserir o cartão na leitora será solicitado a validação com a leitura da impressão digital. O usuário então colocará na leitora o mesmo dedo que cadastrou anteriormente. O software irá converter a imagem da impressão digital, em código de caracteres, e providenciará a comparação com os códigos de digitais gravados no banco de dados. Se a digital coincidir com alguma inserida no banco de dados, será feita a verificação do portador do cartão. A impressão digital irá pertencer há algum usuário e esse usuário terá um *hash code* associado. Para confirmar se o indivíduo é o dono do *smart card*, o código *hash* será comparado com o código inserido no cartão. Se forem iguais significa que o usuário está devidamente cadastrado no sistema e é o portador do cartão, portanto seu acesso é válido. Caso seja encontrada a digital no banco de dados mas o código não coincida com a informação gravada no cartão, significa que o usuário está cadastrado mas não é o dono do *smart card*, assim, o acesso será negado. Se a digital não for encontrada o acesso também será

negado pois indica o não cadastramento do indivíduo. Tem-se como premissa que os *smart cards* devem possuir um código gravado em sua memória, portanto devem ter um portador associado.

Dessa maneira será possível utilizar as duas soluções em conjunto para fazer o controle de acesso. A portabilidade do cartão com a complexidade da solução biométrica aumentará o nível de segurança além de permitir o registro de entrada dos usuários e de inviabilizar a transferência das senhas de acesso.

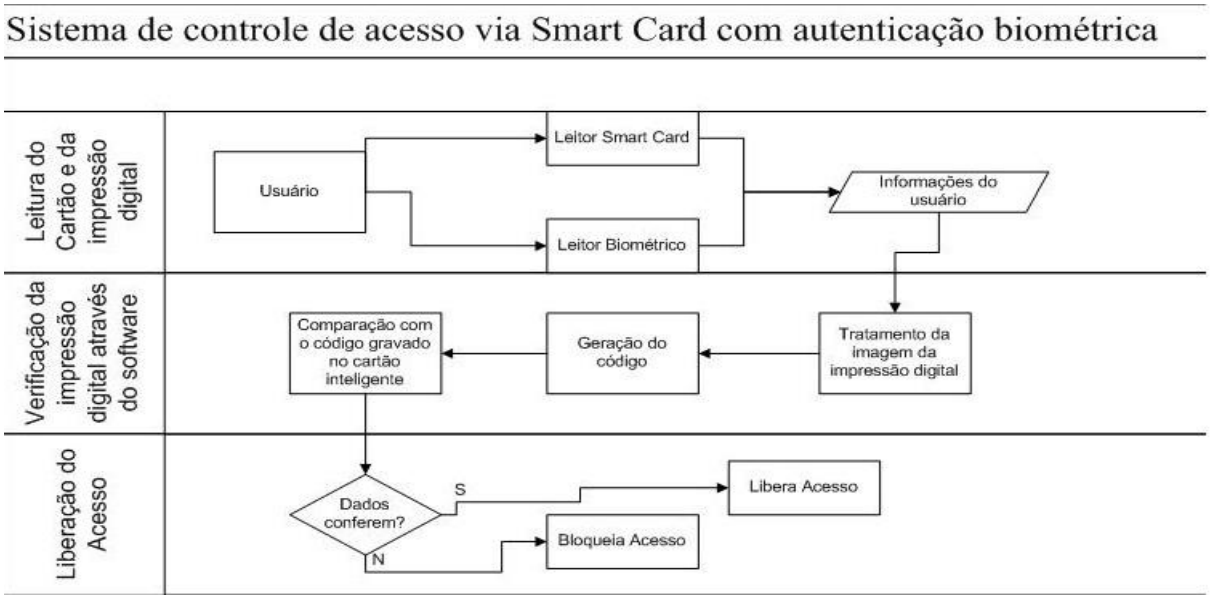

Figura 4.1 – Topologia da Proposta de Solução do Problema / Fonte: O autor.

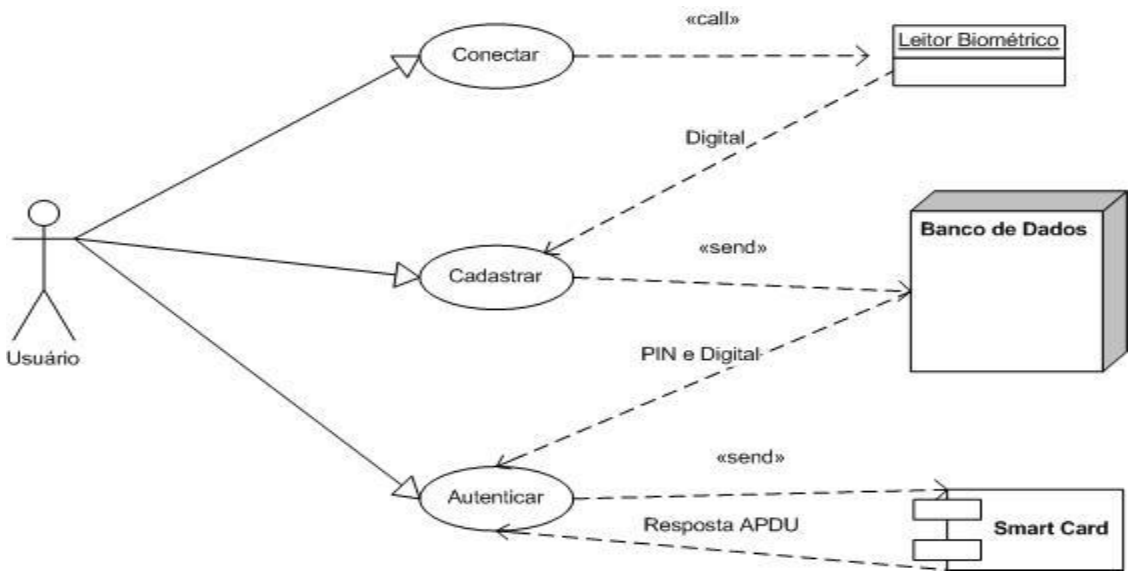

Figura 4.2 – Diagrama de Caso de Uso / Fonte: O autor.

4.2 – Pré-Requisitos

São pré-requisitos para o funcionamento da aplicação:

- Usuário com impressão digital possível de ser capturada pela leitora;
- Todo *smart card* deve ter apenas um usuário vinculado e para acessar o sistema o usuário deve ser cadastrado previamente;
- Ambiente de acesso restrito deve possuir computador para que a aplicação seja instalada;
- Configuração mínima do computador: Processador de 1,5 GHz, 1 GB de memória RAM, 20 GB de disco rígido, placa de rede habilitada e sistema operacional *Windows XP* ou superior
- São necessárias duas portas USB para conexão da leitora de *smart card* e do leitor biométrico;
- Os cartões inteligentes devem suportar a versão 2.2.1 do *Java Card*;
- Instalação e configuração do Java no computador (configuração das variáveis de ambiente e aplicativos JAVAC e JAVA funcionando corretamente);
- Instalação e configuração do banco de dados.

4.3 – Tecnologias Utilizadas

4.3.1 - *Softwares*

Os programas de computador necessários para o desenvolvimento e execução da aplicação são:

- Java SE *Runtime Environment* 6: Embora os recursos disponíveis na versão 6 do Java não sejam utilizados, optou-se por essa versão por ser a mais recente disponível no mercado. A versão 3 do Java atende perfeitamente a solução;

- *Java Card Development Kit* 2.2.1: Possui toda a API necessária para o desenvolvimento da *Applet* a ser inserida no cartão inteligente. Possui também o aplicativo de conversão da Classe para o arquivo CAP;

- IDE Eclipse: Ferramenta para desenvolvimento e compilação de aplicativos Java. Possui ferramentas e muitos recursos essenciais;

- GPShell: Ferramenta responsável por fazer a inserção do arquivo CAP no *smart card;*

- *Microsoft Visual Basic* 6: Ferramenta para desenvolvimento e compilação de aplicativos *Visual Basic*;

- SDK Nitgen: Possui as bibliotecas necessárias para o desenvolvimentos de aplicativos utilizando o leitor biométrico fabricado pela Nitgen. Além disso, o programa da Nitgen providencia a conversão da impressão digital para o código em caracteres;

- MySQL: Banco de Dados utilizado para persistência das informações.

4.3.2 - *Hardware*

As ferramentas de hardware necessárias para o desenvolvimento e execução da aplicação são:

- Leitor biométrico Nitgen Hamster I: Responsável pela captura e verificação da impressão digital;

- Leitora e gravadora de *Smart Card* ACR 38: Responsável pela leitura e gravação de dados na memória do cartão inteligente;

- *Smart Cards* JCOP 2.2.1: Cartões inteligentes que suportam a tecnologia *Java Card* 2.2.1.

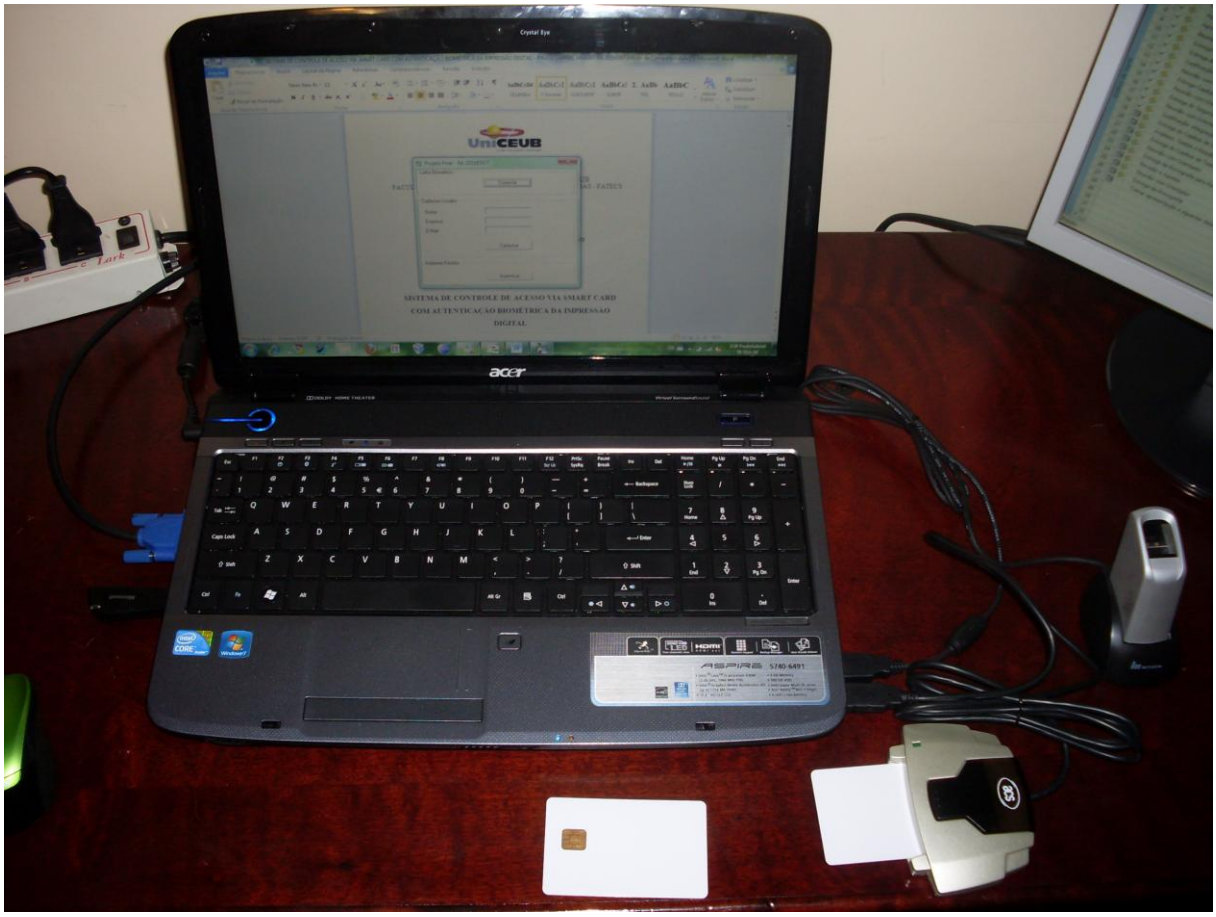

Figura 4.3 – Sistema Montado / Fonte: O autor.

4.4 – Modelo de Dados

Em razão da aplicação utilizar minimamente o Banco de Dados, adotou-se um modelo bem simples. Há apenas uma tabela e atende a solução perfeitamente. A seguir segue uma figura do modelo de dados.

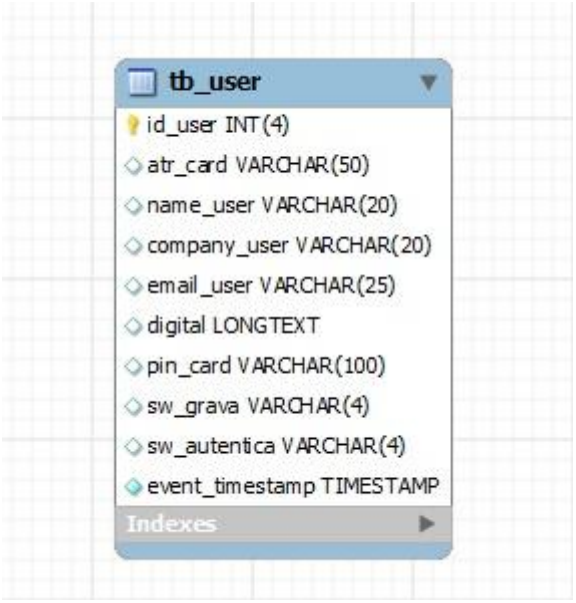

Figura 4.4 – Modelo de Dados / Fonte: O autor.

4.5 – Desenvolvimento da Aplicação

4.5.1 – Cadastramento do Usuário

O processo tem início com a conexão do leitor biométrico. Na tela principal do programa, é necessário clicar no botão Conectar (vide figura 4.5) para que as funções responsáveis pela conexão sejam executadas. Esse botão também irá executar dois processos *batch*, um que compila a classe Java da *Apple*t que será inserida no *smart card* e outro que providencia a criação do arquivo CAP da *Applet*.

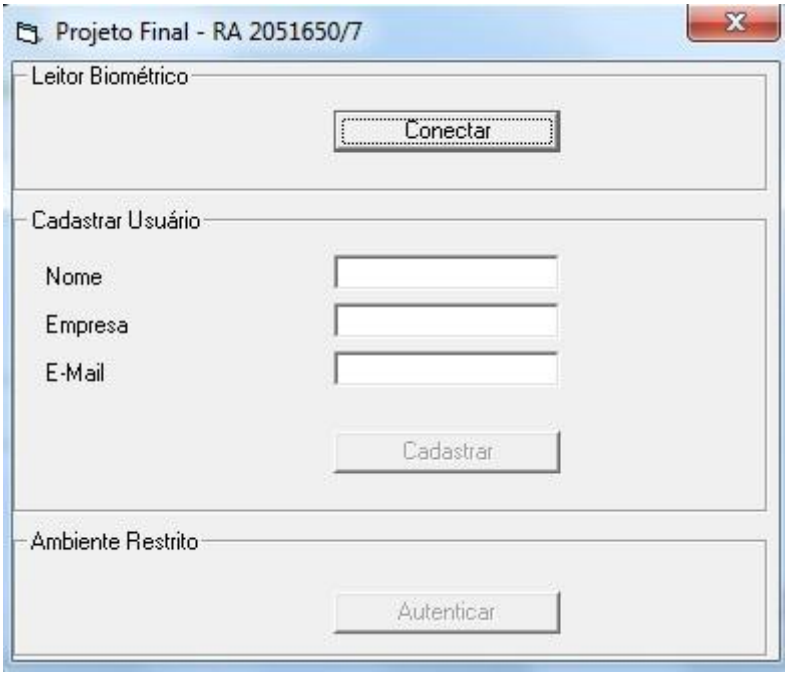

Figura 4.5 – Tela Principal / Fonte: O autor.

Feita a conexão e com o arquivo CAP devidamente gerado é possível cadastrar um usuário no sistema. O cadastro é bem simples e compreende apenas três campos obrigatórios: Nome, Empresa e *E-Mail*. Preenchidas as informações deve-se clicar em Cadastrar (vide Figura 4.6). Esse botão irá primeiramente providenciar a instalação da *Applet* no *Smart Card*, portanto mais um arquivo *batch* será executado. O programa GPShell é o responsável por inserir o arquivo CAP na memória do cartão inteligente. Posteriormente a impressão digital será capturada (vide Figura 4.7). O *software* responsável pela captura e conversão da impressão digital não faz parte do escopo deste projeto, portanto é utilizado um *software* de propriedade da Nitgen. Com a digital capturada e transformada em código de caracteres é possível inserir o usuário no banco de dados.

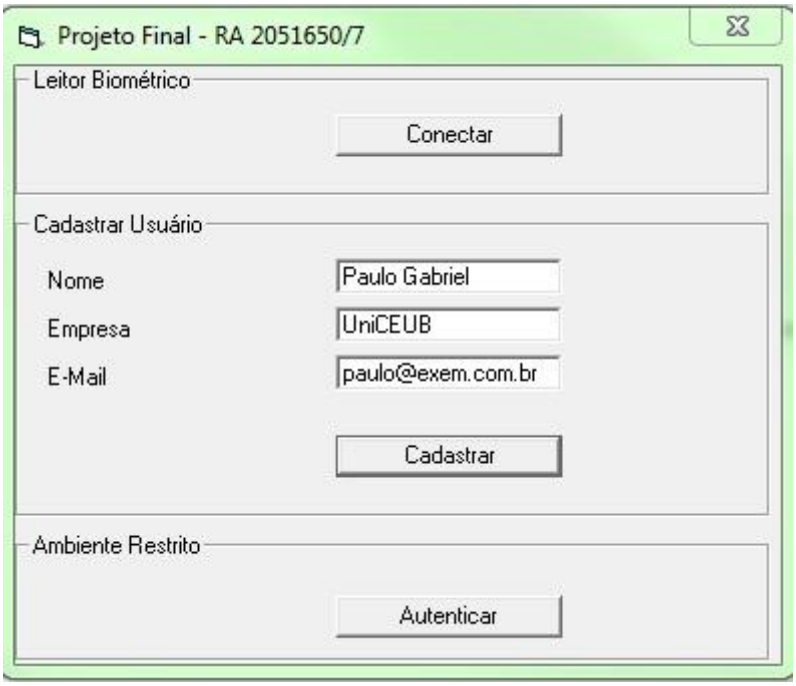

Figura 4.6 – Cadastro do Usuário / Fonte: O autor.

Figura 4.7 – Captura da Digital / Fonte: O autor.

Ainda como resultado do clique no botão Cadastrar, mais um processo *batch* é executado. Tal processo é executado pela JCVM e o resultado é o número ATR do cartão. Esse número é gravado no banco de dados. O passo seguinte é gerar o PIN que será inserido no cartão. O PIN é definido pela concatenação de cinco *strings*: ID do usuário, ATR do cartão, Nome do usuário, Empresa do usuário e Email do usuário. Após a concatenação o PIN passa por um algoritmo chamado MD5 que gera um *hash code* com o objetivo de manter a integridade do PIN no momento da escrita e da leitura no *smart card*. O PIN é então inserido na base de dados na linha correspondente ao usuário.

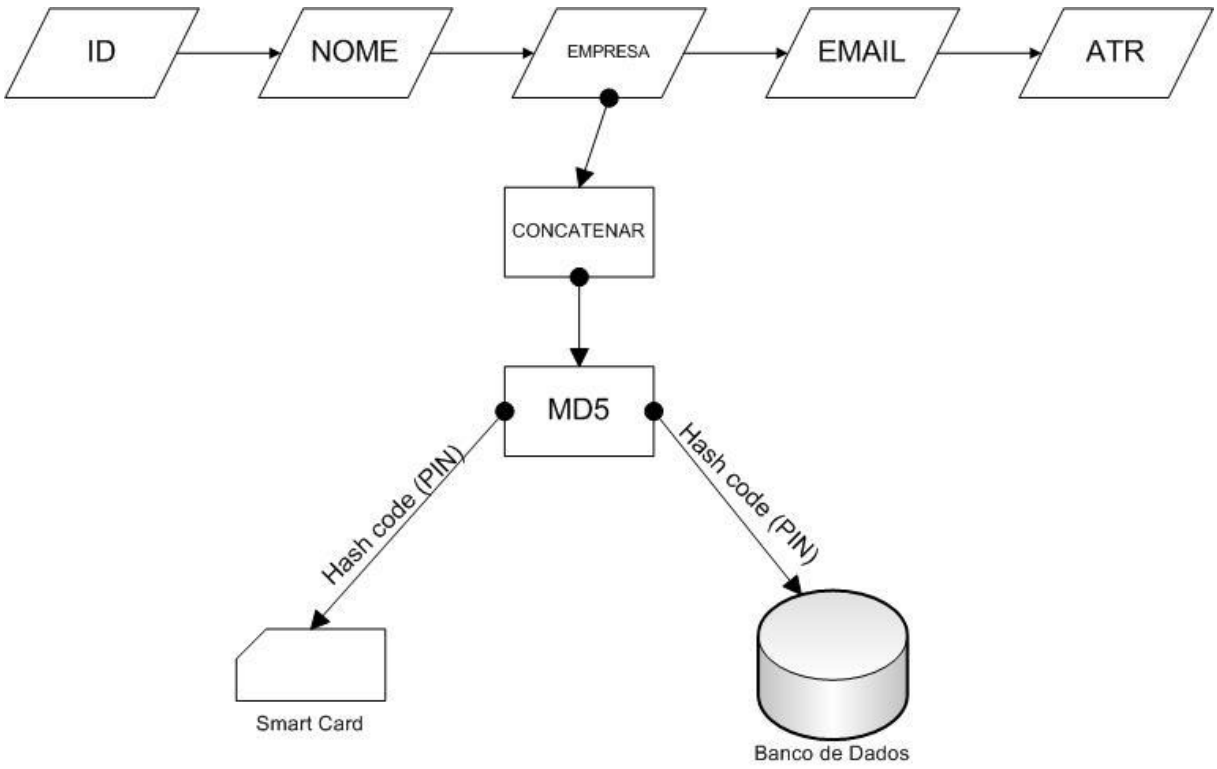

Figura 4.8 – Geração e Inserção do PIN / Fonte: O autor.

Por fim outro processo *batch* é executado, dessa vez o objetivo é compilar e executar a classe Java responsável por enviar a mensagem APDU que insere o PIN na memória do *smart card*. Com a finalização desse processo, o usuário estará cadastrado e apto para acessar o sistema.

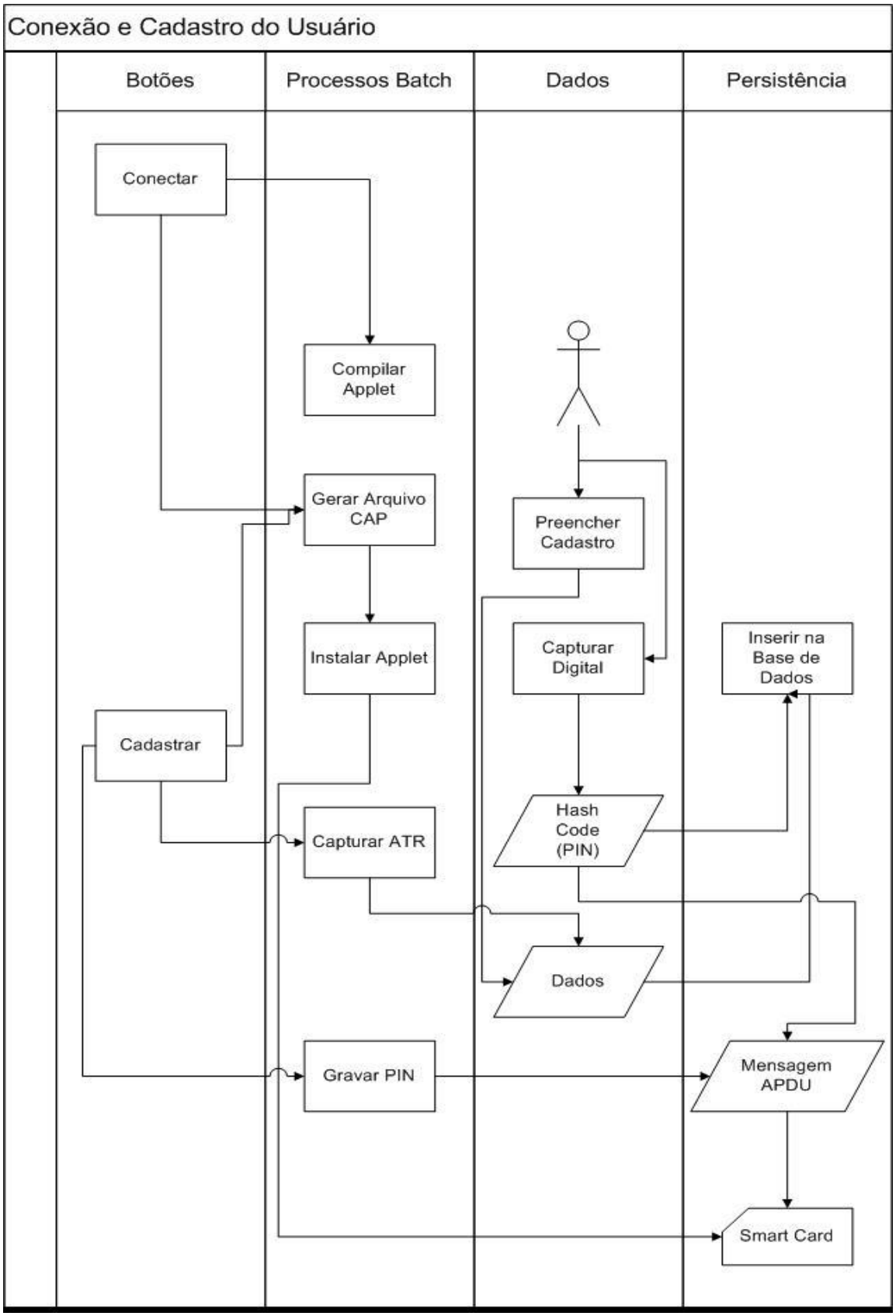

Figura 4.9 – Diagrama de Atividades Conexão e Cadastro do Usuário / Fonte: O autor.

4.5.2 – Autenticação do Usuário

Para realizar a autenticação do usuário e verificar se o mesmo esta apto para ingressar no ambiente de acesso restrito basta um clique no botão Autenticar. Optou-se pela simulação de um ambiente restrito de maneira simples para não destoar o foco do projeto, que é a integração das ferramentas de cartão e biometria.

A primeira ação do botão é fazer a captura da impressão digital do usuário que está tentando o acesso. Novamente o *software* proprietário da Nitgen é utilizado para fazer a captura e conversão da digital em *string* de caracteres. A impressão digital capturada será então comparada com as impressões digitais cadastradas no banco de dados. Quando houver a coincidência de digitais o sistema retornará o ID do usuário dono da impressão digital.

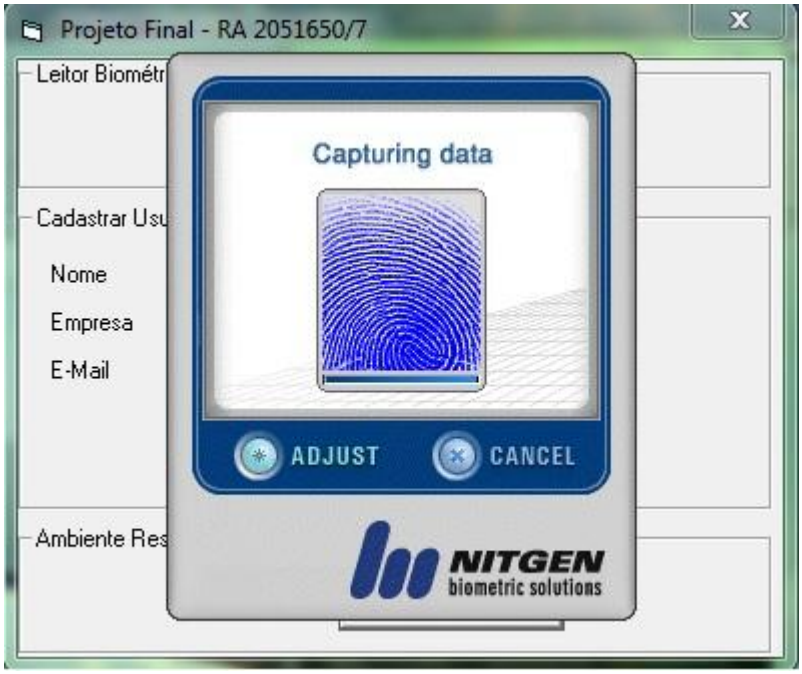

Figura 4.10 – Verificação da Impressão Digital / Fonte: O autor.

Sabendo-se o ID do usuário, outra rotina *batch* é executada, com o objetivo de fazer a comparação do PIN do usuário com o PIN gravado no *smart card*. O processo de comparação é iniciado através de uma mensagem APDU enviada para o cartão. Por sua vez, o cartão

responde e grava essa resposta no banco de dados. Com base na resposta é possível saber se o usuário que tentou o acesso é o portador ou não do *smart card*. Segue diagrama de atividades:

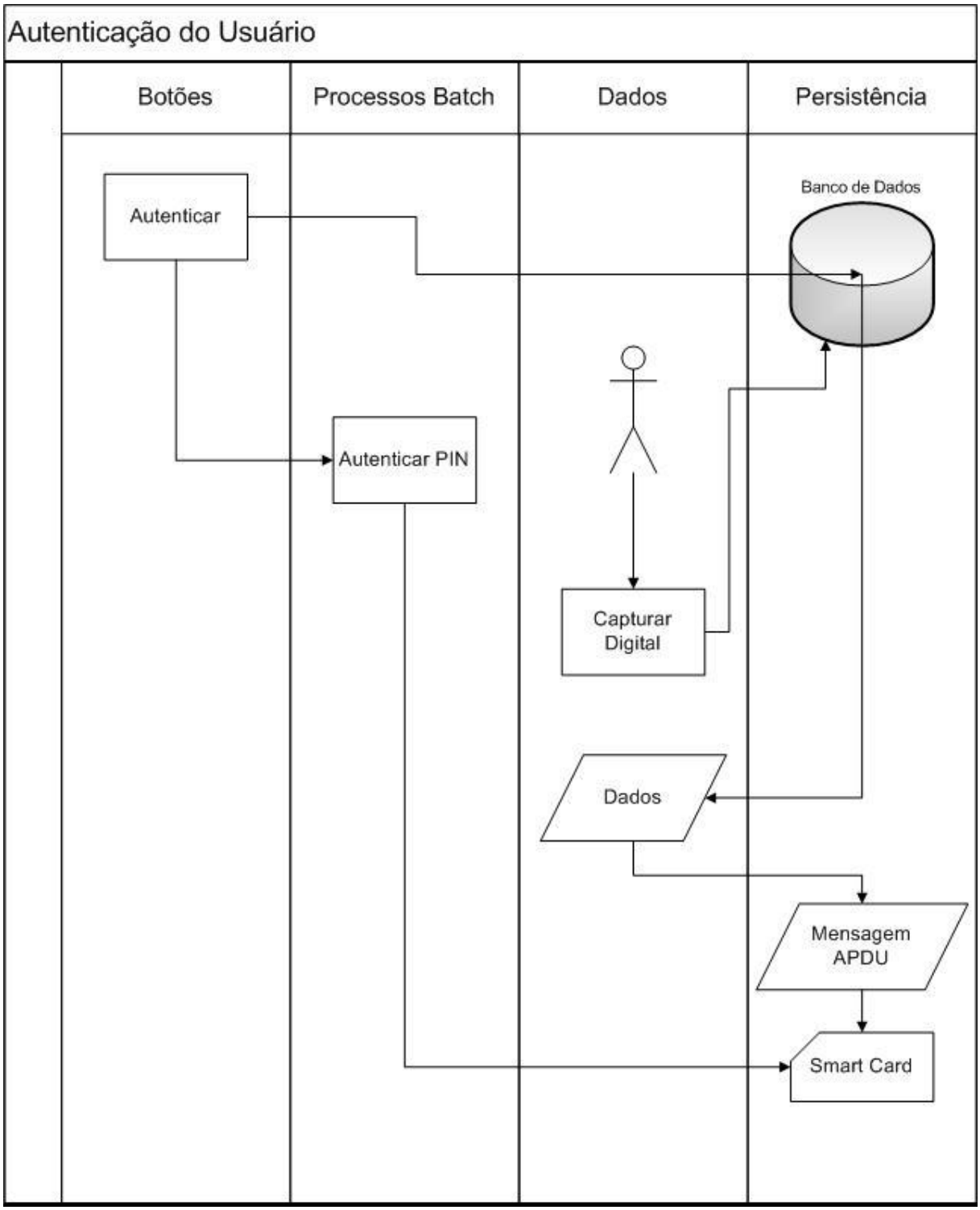

Figura 4.11 – Diagrama de Atividades Autenticação do Usuário / Fonte: O autor.

4.5.3 – Controle do Acesso

O controle de acesso "é o sistema que permite ou não a entrada de um indivíduo ou objeto em determinados locais, em determinados horários, mediante sua identificação". (SOUZA, 2010). Diante do exposto, define-se que neste projeto o controle será feito com base em duas informações: se a impressão digital existe e se o usuário é o portador do *smart card*.

Caso a digital do indivíduo for encontrada na base de dados e posteriormente o PIN gravado no cartão inteligente for igual ao PIN relacionado ao indivíduo, o acesso ao sistema será liberado. Neste caso o cartão terá enviado a resposta 9000, que corresponde ao sucesso na validação do PIN.

Se a digital seja encontrada mas o PIN inserido no cartão não coincida com o do usuário, o acesso será negado. Neste caso o cartão terá enviado a resposta 6300, que corresponde ao caso de PIN inválido.

Se a digital não for encontrada o acesso será imediatamente negado pois supõe-se que o indivíduo não está cadastrado no sistema e portanto não tem direto ao acesso.

Por fim há o caso de o cartão responder 6e89, o que significa que o número máximo de tentativas de acesso com aquele *smart card* terá sido atingido. Essa situação implica o bloqueio do cartão e a negação do acesso ao ambiente. Abaixo o caso de uso de autenticação:

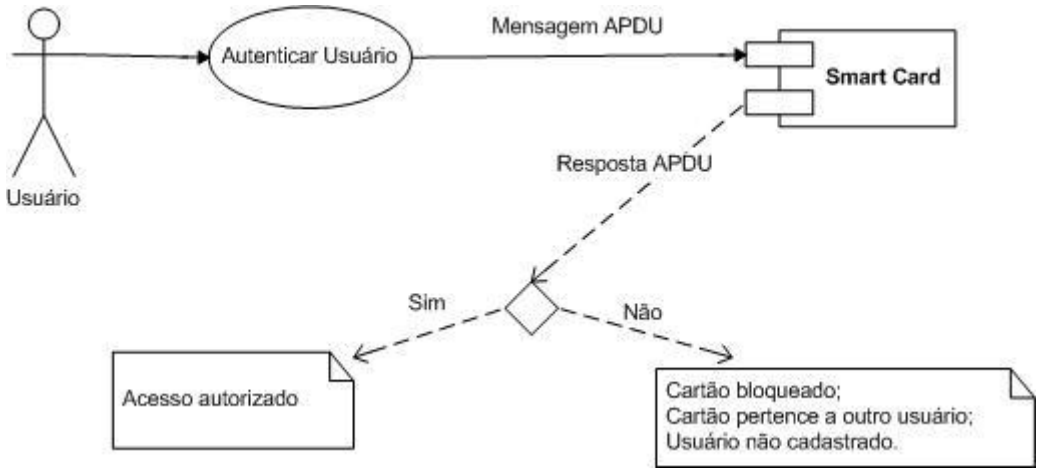

Figura 4.12 – Caso de Uso Autenticar Usuário / Fonte: O autor.

4.5.4 – Relatório de Acessos

O objetivo do relatório de acessos é possibilitar uma auditoria das entradas realizadas no sistema. Espera-se registrar todas os eventos de tentativas de acesso, que tiveram ou não sucesso. O relatório é de extrema importância pois permite a geração de planos de ações corretivas e preditivas. Além disso atua como fator inibidor e até mesmo como controle do horário de entrada e saída dos usuários.

O relatório possui características simples e é criado a partir de uma lista que contém todos os registros do banco de dados e mais aqueles que tentaram o acesso mas não estão cadastrados. A lista é transcrita para um arquivo no formato TXT e finalmente salva no disco rígido do computador.

É sempre criado um arquivo de nome igual, portanto para manter um histórico é necessário renomear o arquivo ou movê-lo para um outro ambiente. A ação que chama a rotina de geração do relatório é a saída do programa, acionada através do botão Sair. Na figura abaixo há um exemplo do Relatório de Acessos.

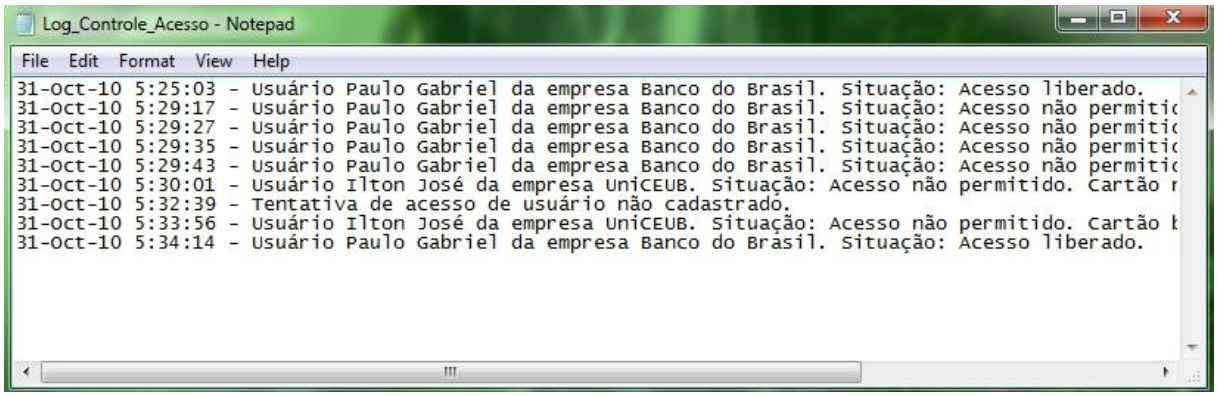

Figura 4.13 – Relatório de Acessos / Fonte: O autor.

4.6 – Estimativa de Custos

Equipamentos	Valor
Leitor Biométrico Nitgen Hamster I	R$287,90
Kit de Desenvolvimento *Java Card* (Leitora ACR38 + 2 *Smart Cards*)	R$220,00
Softwares (Livres ou com licença para estudantes)	R$0,00

Tabela 4.14 – Custos Envolvidos no Projeto / Fonte: O autor.

Valor total: R$507,90.

Em caso de implementação em maior escala seria necessário a aquisição de mais *Smart Cards*, cujo valor varia de R$15,00 a R$35,00. Sendo assim, estima-se que uma aplicação semelhante com suporte a 50 usuários e considerando apenas um ambiente para registro e autenticação dos mesmos, custaria em torno de R$1757,90.

CAPÍTULO 5 – TESTES E RESULTADOS

5.1 – Casos de Teste

5.1.1 – Caso Número 1

Cadastro do usuário A no cartão A. Tentativa de acesso do usuário A utilizando o cartão A.

O resultado esperado é a liberação de acesso do usuário A.

Realização do teste:

O primeiro passo é clicar no botão conectar para que seja realizada a conexão entre o leitor biométrico e o computador.

O segundo passo é o cadastro das informações do Usuário A e a captura da impressão digital. A figura abaixo ilustra o procedimento.

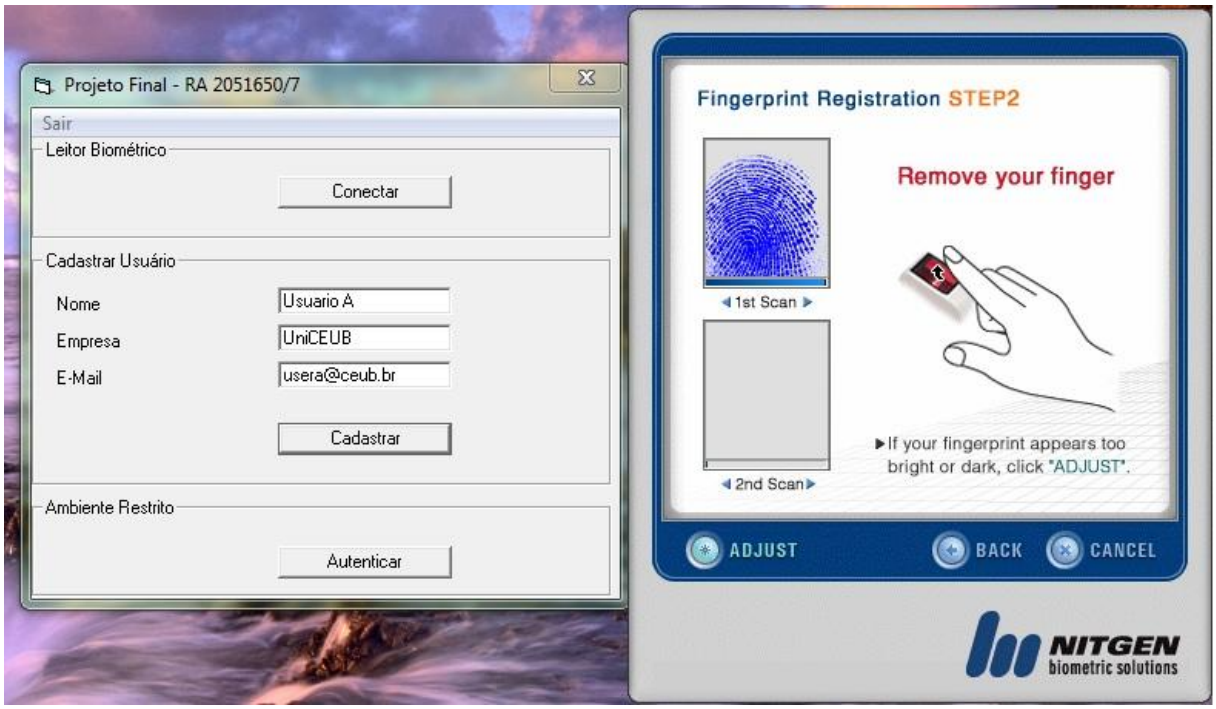

Figura 5.1 – Cadastro do Usuário A / Fonte: O autor.

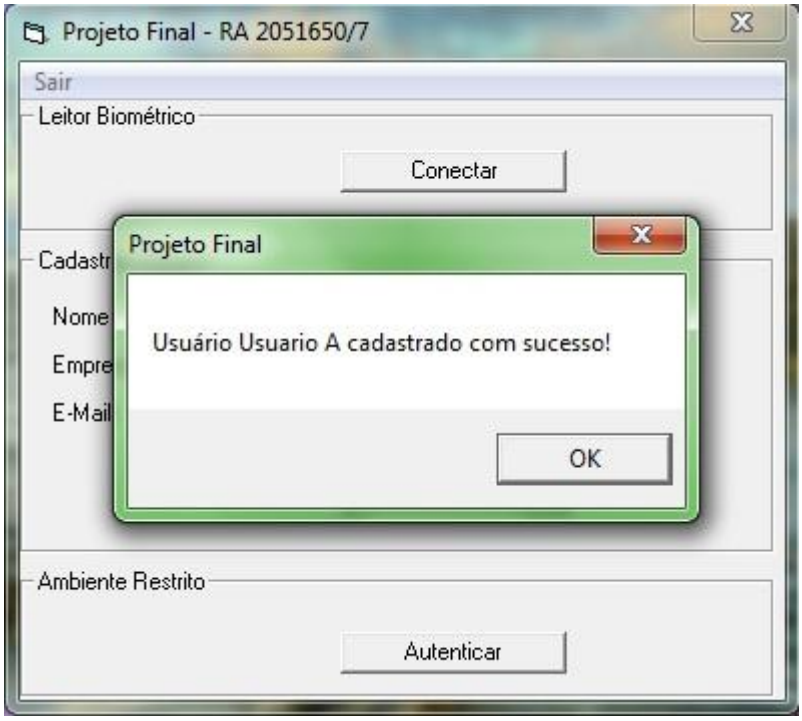

Figura 5.2 – Confirmação de Cadastro do Usuário A / Fonte: O autor.

O terceiro passo é a tentativa de autenticação do usuário. Vide figura abaixo.

Figura 5.3 – Autenticação de Acesso do Usuário A / Fonte: O autor.

O último passo é a liberação do acesso.

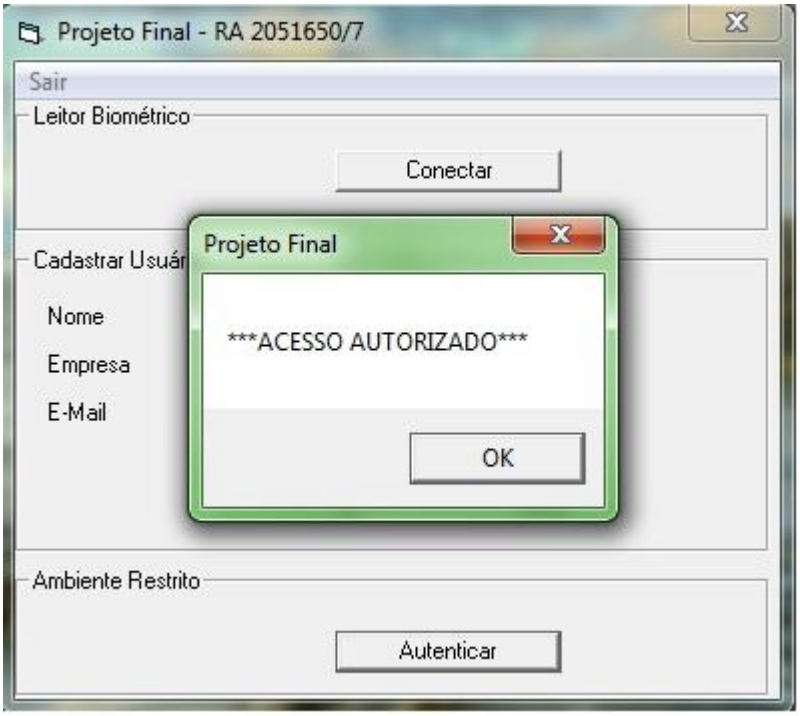

Figura 5.4 – Liberação de Acesso do Usuário A / Fonte: O autor.

5.1.2 – Caso Número 2

Cadastro do usuário A no cartão A e cadastro do usuário B no cartão B. Quatro tentativas de acesso do usuário A utilizando o cartão B e uma tentativa de acesso do usuário B utilizando o cartão A.

O resultado esperado é a negação de acesso a ambos os usuários e o bloqueio do cartão B.

Realização do teste:

Como a conexão do leitor biométrico já foi realizada no teste anterior, ela é dispensada.

Assim o primeiro passo é o cadastro das informações do Usuário B e a captura da impressão digital. O Usuário A já foi cadastrado. A figura abaixo ilustra o procedimento.

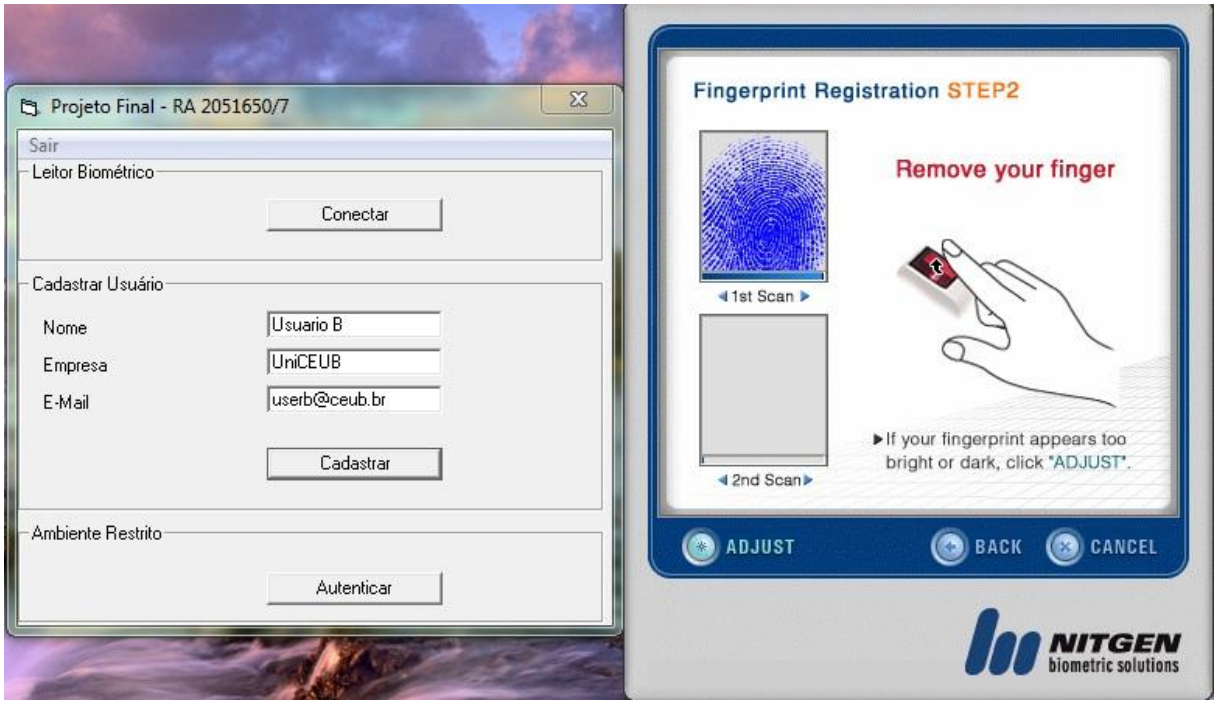

Figura 5.5 – Cadastro do Usuário B / Fonte: O autor.

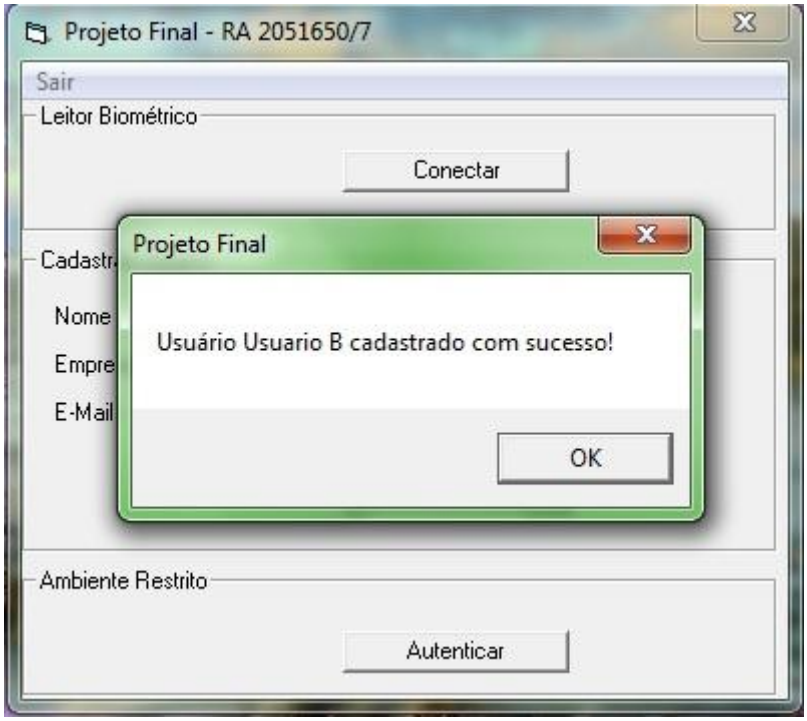

Figura 5.6 – Confirmação de Cadastro do Usuário B / Fonte: O autor.

O terceiro passo são as quatro tentativas de autenticação do Usuário A utilizando o cartão B. Vide figura abaixo.

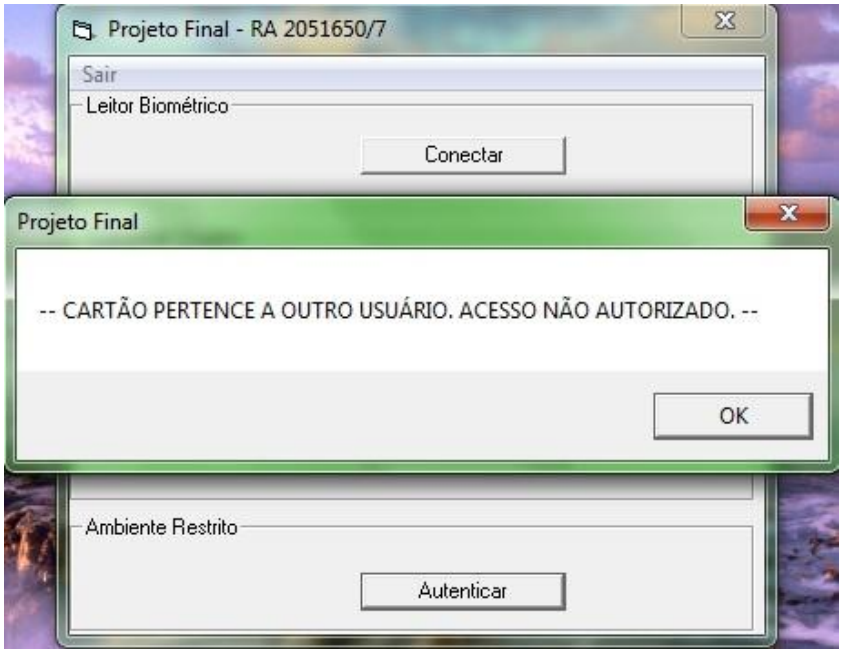

Figura 5.7 – Acesso Não Autorizado do Usuário A / Fonte: O autor.

Conforme a figura abaixo percebe-se a negação do acesso e, por consequência das várias tentativas de acesso, o bloqueio do cartão B.

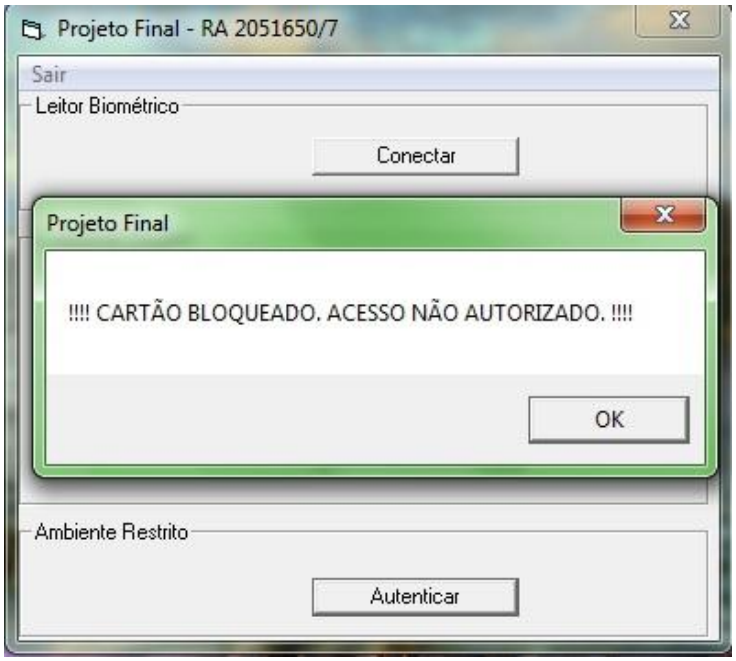

Figura 5.8 – Cartão B Bloqueado / Fonte: O autor.

O quarto passo é a tentativa de acesso do Usuário B com o cartão A. A figura abaixo mostra a negação do acesso.

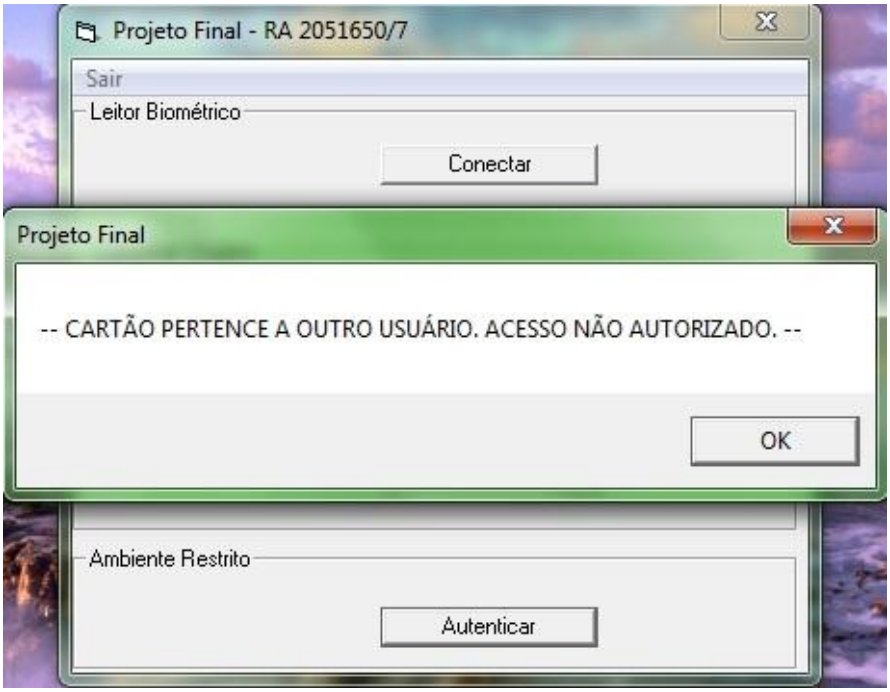

Figura 5.9 – Acesso Não Autorizado do Usuário B / Fonte: O autor.

5.1.3 – Caso Número 3

Tentativa de acesso do usuário não cadastrado C utilizando o cartão A.

O resultado esperado é a negação de acesso porque o usuário C não está cadastrado no sistema.

Realização do teste:

Realização de tentativa de acesso de um usuário não cadastrado utilizando o cartão A.

Conforme é mostrado na figura abaixo, o acesso não é concedido.

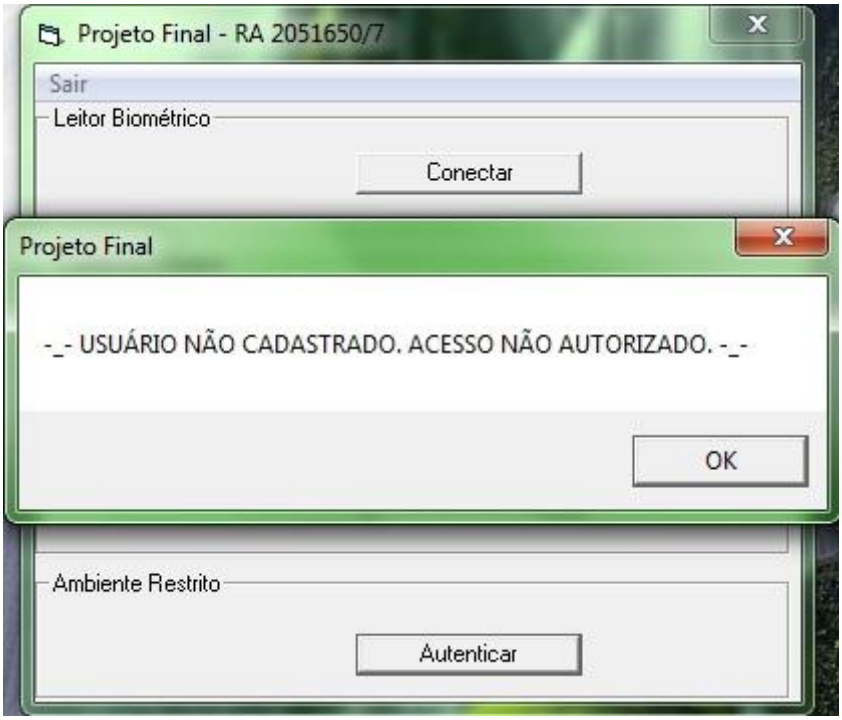

Figura 5.10 – Acesso Negado de Usuário Não Cadastrado / Fonte: O autor.

5.1.4 – Caso Número 4

Tentativa de acesso do usuário B com o cartão bloqueado B. Tentativa de acesso do usuário A com o cartão A. Geração do relatório de acessos.

O resultado esperado é a negação de acesso ao usuário B porque o cartão B deve estar bloqueado. Já o acesso ao usuário A deve ser concedido. Além disso, ao pressionar o botão Sair o relatório de acessos deve ser gerado.

Realização do teste:

Novamente, como a conexão do leitor biométrico já foi realizada no primeiro teste, esse procedimento é dispensado.

Assim o primeiro passo é a tentativa de acesso do Usuário B com o cartão bloqueado B. A figura abaixo mostra a negação do acesso devido ao bloqueio.

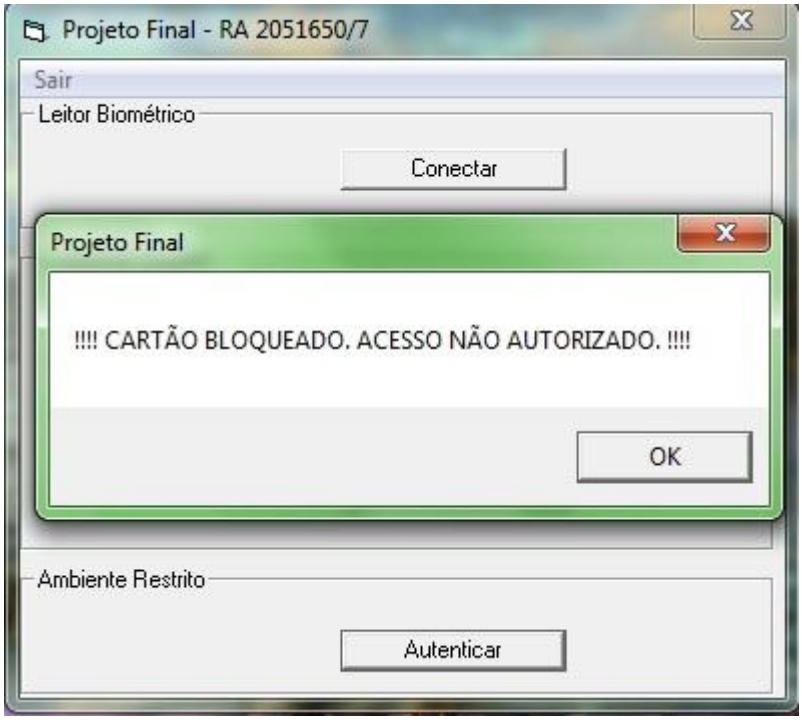

Figura 5.11 – Acesso Negado Devido a Cartão Bloqueado / Fonte: O autor.

O segundo passo é a tentativa de autenticação do Usuário A utilizando o cartão A. A figura abaixo mostra a liberação.

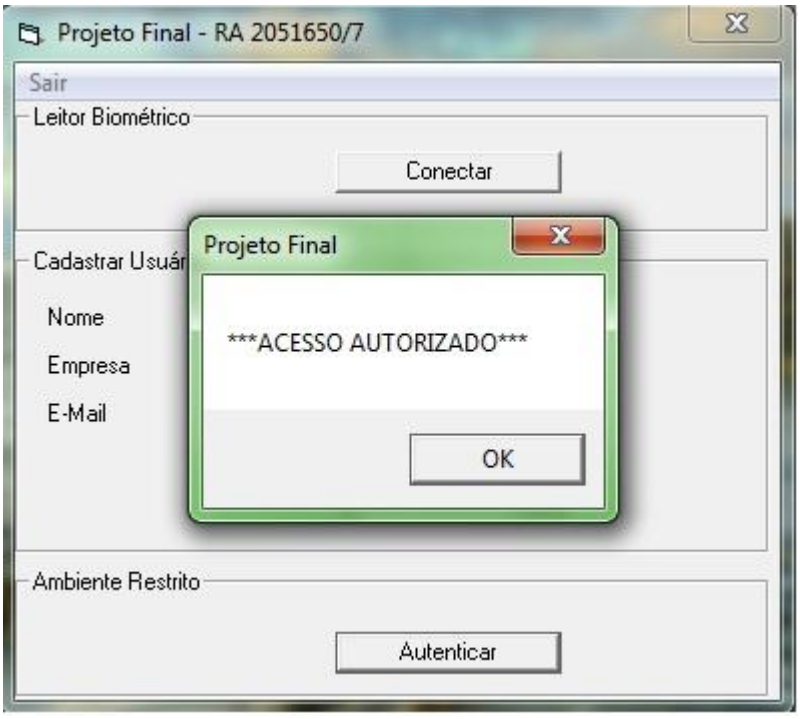

Figura 5.12 – Acesso Liberado para Usuário A / Fonte: O autor.

O terceiro passo é o relatório de acessos gerado. A figura abaixo demostra o resultado.

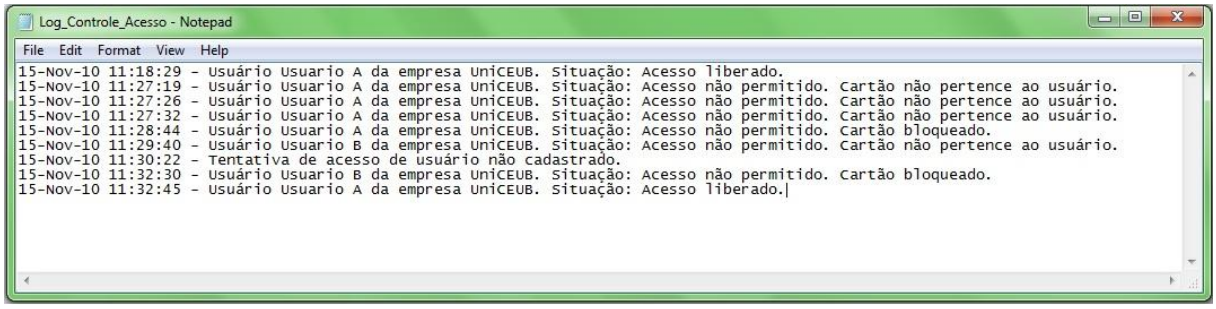

Figura 5.13 – Relatório de Acessos Gerado / Fonte: O autor.

5.1.5 – Tabela de Resultados dos Testes

Caso de Teste	Resultado
Número 1	Aprovado
Número 2	Aprovado
Número 3	Aprovado
Número 4	Aprovado

Tabela 5.14 – Resultado dos Teste / Fonte: O autor.

5.2 – Análise dos Resultados

5.2.1 – Pontos Positivos

O resultado dos testes representam o sucesso da aplicação desenvolvida. A simulação de entrada em um ambiente de acesso restrito foi realizada utilizando os dois sistemas de autenticação, portanto considera-se um sucesso a integração de ambos. Mostrou-se também que o cartão inteligente e o leitor biométrico não funcionam de maneira isolada e sim de

maneira conjunta. Além disso é importante destacar a simplicidade e rapidez da realização do cadastro e a eficiência na autenticação dos usuários.

5.2.2 – Dificuldades e Desvantagens

Algumas dificuldades encontradas durante o desenvolvimento do projeto:

- Apesar de ser uma tecnologia não muito recente, há pouco material de estudo sobre *smart card*;
- A tecnologia *Java Card* também não é muito difundida. Há poucos livros na área e poucos profissionais especializados. A fonte de material se concentra em estudos acadêmicos e comunidades especializadas na internet, como o fórum de discussão da Oracle – Sun;
- A memória limitada do *smart card* não suportou a inserção da *string* de caracteres contendo a digital do usuário, por isso foi necessário uma adaptação utilizando um banco de dados;
- O *smart card* utilizado suporta a versão 2.2.1 do Java Card. Essa versão não abrange a API de Biometria e por isso não foi possível inserir o padrão biométrico da digital diretamente na memória do cartão;
- Dependência do banco de dados. Com cartões que suportem versões superiores do *Java Card* será possível eliminar o banco de dados.

CAPÍTULO 6 – CONCLUSÃO

6.1 – Síntese Conclusiva

Toda situação, que envolve seres humanos e ativos confidenciais, tem um risco. O homem é suscetível à corrupção, pode possuir ideologias divergentes, pode ter um desejo de vingança, pode ter alguma motivação religiosa, política ou empresarial, entre outros motivos. Portanto o cenário expõe a necessidade de estudos frequentes de metodologias, técnicas e ferramentas que foquem na proteção de ambientes sensíveis. Levando-se em conta os fatos foi definido como objetivo deste trabalho a integração de duas tecnologias de segurança da informação visando somar seus benefícios e agregar maior confidencialidade aos sistemas de controle de acesso.

O desenvolvimento do projeto resultou em um protótipo que permite a integração do leitor biométrico com o *smart card* e assim obteve-se um sistema de segurança baseado no que o usuário possui e no que o usuário é. A simulação de registro e autenticação no sistema apresentou resultado satisfatório na fase de testes, onde quatro casos foram realizados e todos eles bem sucedidos.

Em suma, pode-se considerar o cumprimento do objetivo geral e dos objetivos específicos, já que as tecnologias foram integradas e o sistema para cadastro e simulação de entrada em um ambiente de acesso restrito foi desenvolvido. A viabilização da integração traz uma nova diretriz para os sistemas de controle de acesso e de segurança da informação. As características biométricas tem um índice muito baixo de falhas e os cartões inteligentes possuem uma boa proteção contra a clonagem. O estudo e investimento na área já são realidade e as perspectivas para o futuro são muito boas.

6.2 – Sugestões para Trabalhos Futuros

As dificuldades encontradas durante a realização deste projeto abrem oportunidades para o desenvolvimento de novos trabalhos relacionados ao tema. Entre essas oportunidades destacam-se:

- Elaborar solução semelhante utilizando *Smart Cards* que suportem versão superior da tecnologia *Java Card*, para assim abranger a API de Biometria e permitir a eliminação da dependência do banco de dados;
- Propor e apresentar uma solução semelhante utilizando outro tipo de autenticação biométrica;
- Propor e apresentar uma solução semelhante utilizando outro tipo de cartão, como os cartões sem contato (*contactless card*).

REFERÊNCIAS BIBLIOGRÁFICAS

BANCO DO BRASIL. **Grandes números do Banco do Brasil**. Disponível em <http://www.bb.com.br/portalbb/home23,116,116,1,1,1,1.bb> Acesso em: 31 de Agosto de 2010.

BIOMETRIC, Group. **Biometrics Market and Industry Report 2009-2014**. Disponível em <http://www.biometricgroup.com/reports/public/market_report.php> Acesso em: 01 de Outubro de 2010.

CEFET, Rio Grande do Norte. **Projeto JAVACARD Cefet-RN.** Disponível em <http://www.cefetrn.br/javacard/doku.php?do=search&id=> Acesso em: 25 de Julho de 2010.

CHEN, Ziqun. **Java Card Technology for Smart Cards**. Prentice Hall, 2000.

CERTISIGN, Empresa**. O que é Certificação Digital?** Disponível em < http://www.certisign.com.br/certificacao-digital/por-dentro-da-certificacao-digital> Acesso em: 02 de Setembro de 2010.

CUSTODIO, Klecius Vinicius Assis. **Estudo do uso de biometria para autenticação tem terminais de auto-atendimento.** Monografia de Graduação do Curso de Engenharia de Computação. Brasília: UniCEUB, 1º Semestre de 2007.

ESTADÃO, Jornal. **Dados do ENEM vazam na internet.** Disponível em <http://www.estadao.com.br/estadaodehoje/20100804/not_imp590058,0.php> Acesso em: 01 de Setembro de 2010.

FEBRABAN. **O setor bancário em números.** Disponível em <http://www.febraban.org.br/> Acesso em: 31 de Agosto de 2010.

FOLHA DE SÃO PAULO , Jornal. **Furto de Informações da Petrobrás.** Disponível em <http://www1.folha.uol.com.br/folha/dinheiro/ult91u372319.shtml> Acesso em: 01 de Setembro de 2010.

GLOBO, Reportagem. **Bancos brasileiros perderão R$ 900 milhões com fraudes online no ano**. Disponível em <http://g1.globo.com/economia-e-negocios/noticia/2010/08/bancos-brasileiros-perderao-r-900-milhoes-com-fraudes-online-no-ano.html> Acesso em: 31 de Agosto de 2010.

IETF. **Request For Comments**. Disponível em <http://www.ietf.org/rfc.html> Acesso em: 03 de Outubro de 2010.

LINHA DEFENSIVA, Site. **Dicionário de Termos de Segurança da Informação.** Disponível em <http://www.linhadefensiva.org/dicionario/> Acesso em: 01 de Setembro de 2010.

MACORATTI, José Carlos. **Dicas e tutoriais sobre VB6.** Disponível em <http://www.macoratti.net/Default.aspx> Acesso em: 03 de Setembro de 2010.

MARX TECNOLOGIA, Empresa. **O que é RFID?** Disponível em < http://www.marx.com.br/rfid/o-que-e-rfid> Acesso em: 02 de Setembro de 2010.

MICROSOFT, TechNet Brasil. **Entendendo e implementando a Norma ABNT NBR ISO/IEC 17799:2005. Academia de Segurança.** Disponível em <https://www.technetbrasil.com.br/academia2007/seguranca/home.aspx> Acesso em: 25 de Agosto de 2010.

MITNICK, Kevin**. Engenharia Social.** Disponível em <http://www.invasao.com.br/2010/02/01/kevin-mitnick-explica-o-que-e-engenharia-social/> Acesso em: 01 de Setembro de 2010.

MOREIRA, Tiago Maccagnan. **Verificação biométrica da palma da mão com identificação via Smart Card**. Monografia de Graduação do Curso de Engenharia de Computação. Curitiba: UnicenP, 2004.

NBR/ISO/IEC 27002**. Tecnologia da Informação – Código de prática para a gestão da segurança da informação**. Rio de Janeiro. Associação Brasileira de Normas Técnicas, 2007.

PARODI, Lorenzo. **Cartões com Chip.** Disponível em < http://www.fraudes.org> Acesso em: 02 de Setembro de 2010.

PINHEIRO, José Maurício. **Biometria nos Sistemas Computacionais - Você é a Senha**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2008.

RANGEL, Alexandre. **MySQL: Projeto, Modelagem e Desenvolvimento de Banco de Dados**. Rio de Janeiro: Alta Books, 2004.

SIERRA, Kathy; BATES, Bert**. Use a cabeça! Java**. 2ª Edição. Rio de Janeiro: Editora Alta Books, 2005.

SONSUN. **Kit de Desenvolvimento JAVACARD.** Disponível em <http://www.sonsun.com.br/sonsun_JCOP.php> Acesso em: 10 de Julho de 2010.

SOUSA, Giovanni Ferreira. **Controle de ponto utilizando a tecnologia Java Card.** Monografia de Graduação do Curso de Engenharia de Computação. Brasília: UniCEUB, 2º Semestre de 2009.

SOUZA, Marcelo Barbosa. **Controle de Acesso: Conceitos, Tecnologia e Benefícios.** São Paulo: Editora Sicurezza, 2010.

SUN, Java. **Java Card Development Kit 2.2.1**. Disponível em <http://java.sun.com/javacard/devkit/> Acesso em: 11 de Julho de 2010.

APÊNDICES

Transcreve-se abaixo o código de interface do usuário:

```
'
#########################################################################
######################

'

'   UniCEUB - FATECS - Engenharia de Computacao

'   Disciplina: Projeto Final

'   2o. Semestre de 2010

'

'       Titulo: SISTEMA DE CONTROLE DE ACESSO VIA SMART CARD COM
AUTENTICAÇÃO BIOMÉTRICA DA IMPRESSÃO DIGITAL

'

'   Aluno: Paulo Gabriel Ribacionka Goes de Araujo

'   RA:    2051650/7

'

'
#########################################################################
######################

' Variáveis Globais

Dim objNBioBSP As NBioBSPCOMLib.NBioBSP
```

```
Dim objDevice As IDevice
Dim objExtraction As IExtraction
Dim objMatching As IMatching

Dim impressaoDigital As String

Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

' ------------------------------------------------------------------------------------------
' Initialize application / Copyright: NITGEN Co., Ltd.
' ------------------------------------------------------------------------------------------
Private Sub Form_Load()

    ' Create NBioBSP object
    Set objNBioBSP = New NBioBSPCOMLib.NBioBSP
    Set objDevice = objNBioBSP.Device
    Set objExtraction = objNBioBSP.Extraction
    Set objMatching = objNBioBSP.Matching

    btAutenticar.Enabled = False
    btCadastrar.Enabled = False

End Sub

' ------------------------------------------------------------------------------------------
' Terminate application / Copyright: NITGEN Co., Ltd.
```

```
' ------------------------------------------------------------------------------------------

Private Sub Form_Terminate()

    Set objDevice = Nothing
    Set objExtraction = Nothing
    Set objMatching = Nothing
    Set objNBioBSP = Nothing

End Sub



' ------------------------------------------------------------------------------------------
' Open device / Copyright: NITGEN Co., Ltd.
' ------------------------------------------------------------------------------------------

Private Sub btConectar_Click()

    Call FCarregaArquivoJAVA("Compila_ClasseGravaValidaSenha")
    Call FCarregaArquivoJAVA("Gera_ArquivoCAP")

    ' Select device type
    nNameID = 0
    iDeviceID = NBioAPI_DEVICE_ID_AUTO_DETECT
    id = 0
    iDeviceID = objDevice.MakeDeviceID(iDeviceID, id)

    ' Close Device if before opened
```

```
    Call objDevice.Close(objDevice.OpenedDeviceID)

    ' Open device
    Call objDevice.Open(iDeviceID)

    If objDevice.ErrorCode = NBioAPIERROR_NONE Then
        btAutenticar.Enabled = True
        btCadastrar.Enabled = True
        nDeviceID = iDeviceID
    End If


End Sub

' ------------------------------------------------------------------------------------------
' Cadastrar Usuário
' ------------------------------------------------------------------------------------------
Private Sub btCadastrar_Click()


    Dim nomeUsuario As String
    Dim empresaUsuario As String
    Dim emailUsuario As String
    Dim db As ADODB.Connection
    Dim stringConexao As String
    Dim sqlINSERT As String
```

```
Dim sqlSELECT As String
Dim sqlUPDATE As String
Dim rs As ADODB.Recordset
Dim codigoUsuario As String
Dim Hash As New MD5Hash
Dim bytBlock() As Byte
Dim pin_card As String

nomeUsuario = nome.Text
empresaUsuario = empresa.Text
emailUsuario = email.Text

If FCarregaArquivoJAVA("Instalar_AppletSmartCard") Then

    impressaoDigital = ""
    Call objExtraction.Enroll("", Null)

End If

If objExtraction.ErrorCode = NBioAPIERROR_NONE Then

    stringConexao = "PROVIDER=MSDASQL;dsn=projeto;uid=;pwd=;"
    Set db = New ADODB.Connection
    db.CursorLocation = adUseClient
    db.Open stringConexao
```

```
impressaoDigital = objExtraction.TextEncodeFIR

sqlINSERT = "INSERT INTO tb_user (name_user, company_user, email_user, digital) " & _
            "VALUES ('" & nomeUsuario & "', '" & empresaUsuario & "','" & emailUsuario & "', " & _
            "'" & impressaoDigital & "'); "

db.Execute (sqlINSERT)

If FCarregaArquivoJAVA("CompilaExecuta_ClasseATR") Then

    Set rs = New ADODB.Recordset
    sqlSELECT = "SELECT * from tb_user ORDER BY id_user desc LIMIT 1;"
    rs.Open LCase(sqlSELECT), db, adOpenStatic, adLockOptimistic

    codigoUsuario = rs!id_user & rs!atr_card & rs!name_user & rs!company_user & rs!email_user

    bytBlock = StrConv(codigoUsuario, vbFromUnicode)

    pin_card = Hash.HashBytes(bytBlock)

    sqlUPDATE = "UPDATE tb_user SET pin_card = '" & pin_card & "' WHERE id_user = " & rs!id_user & ";"
    db.Execute (sqlUPDATE)
```

```
        End If

        If FCarregaArquivoJAVA("CompilaExecuta_GravaSenha") Then

            MsgBox ("Usuário " & nomeUsuario & " cadastrado com sucesso!")

            nome.Text = ""
            empresa.Text = ""
            email.Text = ""

        End If

    End If

    rs.Close
    Set rs = Nothing
    db.Close
    Set db = Nothing

End Sub

' ----------------------------------------------------------------------------------------
' Autenticar Acesso
' ----------------------------------------------------------------------------------------
Private Sub btAutenticar_Click()
```

```
Dim db As ADODB.Connection
Dim stringConexao As String
Dim sqlSELECT As String
Dim sqlINSERT As String
Dim sqlDELETE As String
Dim sqlUPDATE As String
Dim rs As ADODB.Recordset
Dim rs2 As ADODB.Recordset
Dim rs3 As ADODB.Recordset
Dim id As Integer


stringConexao = "PROVIDER=MSDASQL;dsn=projeto;uid=;pwd=;"
Set db = New ADODB.Connection
db.CursorLocation = adUseClient
db.Open stringConexao

sqlSELECT = "SELECT * from tb_user;"

Set rs = New ADODB.Recordset
rs.Open LCase(sqlSELECT), db, adOpenStatic, adLockOptimistic

id = 0
objExtraction.Capture
Do While Not rs.EOF
    impressaoDigital = rs!digital
```

```
    Call objMatching.VerifyMatch(objExtraction.TextEncodeFIR, impressaoDigital)
    If objMatching.MatchingResult = NBioAPI_TRUE Then
       id = rs!id_user
       Exit Do
    End If
    rs.MoveNext
  Loop

  If id <> 0 Then

    sqlINSERT = "INSERT INTO tb_user (atr_card, name_user, company_user, email_user,
digital, pin_card, sw_grava) " & _
                "VALUES ('" & rs!atr_card & "','" & rs!name_user & "', '" & rs!company_user &
"','" & rs!email_user & "', " & _
                "'" & rs!digital & "', '" & rs!pin_card & "', '" & rs!sw_grava & "');"
    db.Execute (sqlINSERT)

    If FCarregaArquivoJAVA("CompilaExecuta_AutenticaSenha") Then

        sqlSELECT = "select * from tb_user order by id_user desc limit 1;"
        Set rs2 = New ADODB.Recordset
        rs2.Open LCase(sqlSELECT), db, adOpenStatic, adLockOptimistic

        sqlUPDATE = "UPDATE tb_user SET sw_autentica = '" & rs2!sw_autentica & "'
WHERE id_user = " & rs!id_user & ";"
        db.Execute (sqlUPDATE)

        sqlDELETE = "DELETE from tb_user WHERE id_user = " & rs2!id_user & ";"
```

```
        db.Execute (sqlDELETE)


        sqlSELECT = "select * from tb_user WHERE id_user = " & rs!id_user & ";"
        Set rs3 = New ADODB.Recordset
        rs3.Open LCase(sqlSELECT), db, adOpenStatic, adLockOptimistic


        If rs3!sw_autentica = "9000" Then
            MsgBox ("***ACESSO AUTORIZADO***")
            relatorio.AddItem (Now & " - Usuário " & rs3!name_user & " da empresa " & rs3!company_user & ". Situação: Acesso liberado.")
        ElseIf rs3!sw_autentica = "6e89" Then
            MsgBox ("!!!! CARTÃO BLOQUEADO. ACESSO NÃO AUTORIZADO. !!!!")
            relatorio.AddItem (Now & " - Usuário " & rs3!name_user & " da empresa " & rs3!company_user & ". Situação: Acesso não permitido. Cartão bloqueado.")
        Else
            MsgBox ("-- CARTÃO PERTENCE A OUTRO USUÁRIO. ACESSO NÃO AUTORIZADO. --")
            relatorio.AddItem (Now & " - Usuário " & rs3!name_user & " da empresa " & rs3!company_user & ". Situação: Acesso não permitido. Cartão não pertence ao usuário.")
        End If


        rs2.Close
        Set rs2 = Nothing
        rs3.Close
        Set rs3 = Nothing


    End If
```

```
    Else
        relatorio.AddItem (Now & " - Tentativa de acesso de usuário não cadastrado.")
        MsgBox ("-_- USUÁRIO NÃO CADASTRADO. ACESSO NÃO AUTORIZADO. -_-")
    End If

    rs.Close
    Set rs = Nothing
    db.Close
    Set db = Nothing

End Sub

' ------------------------------------------------------------------------------------------
' Carrega o arquivo .bat especificado
' ------------------------------------------------------------------------------------------

Private Function FCarregaArquivoJAVA(arquivo_bat As String) As Boolean


    FCarregaArquivoJAVA = False

    Select Case arquivo_bat

        Case "Compila_ClasseGravaValidaSenha"
            ' Compila a Classe que sera inserida no cartao
            Call ExecCmd("C:\javacard\projetoFinal\Compila_ClasseGravaValidaSenha.bat")
```

```
    FCarregaArquivoJAVA = True

Case "Gera_ArquivoCAP"
   ' Gera o arquivo CAP
   Call ExecCmd("C:\javacard\projetoFinal\Gera_ArquivoCAP.bat")
   FCarregaArquivoJAVA = True

Case "Instalar_AppletSmartCard"
   ' Instala a Applet no cartao
   Call ExecCmd("C:\javacard\projetoFinal\Instalar_AppletSmartCard.bat")
   FCarregaArquivoJAVA = True

Case "CompilaExecuta_ClasseATR"
   ' Captura o ATR do cartao e salva no Banco de Dados
   Call ExecCmd("C:\javacard\projetoFinal\CompilaExecuta_ClasseATR.bat")
   FCarregaArquivoJAVA = True

Case "CompilaExecuta_GravaSenha"
   ' Grava as informações no cartão
   Call ExecCmd("C:\javacard\projetoFinal\CompilaExecuta_GravaSenha.bat")
   FCarregaArquivoJAVA = True

Case "CompilaExecuta_AutenticaSenha"
  ' Confere usuário cadastrado no cartão
  Call ExecCmd("C:\javacard\projetoFinal\CompilaExecuta_AutenticaSenha.bat")
  FCarregaArquivoJAVA = True
```

```
    End Select


End Function


' ----------------------------------------------------------------------------------------------
' Gera o relatorio
' ----------------------------------------------------------------------------------------------


Private Sub SGerarRelatorioTXT()


    Dim diretorio As String
    Dim log As String
    Dim i As Integer


    diretorio = "C:\Users\Paulo Gabriel\Desktop\UniCEUB - Engenharia de Computacao - Projeto Final - Paulo Gabriel Araujo ra20516507\Programas e Codificacao\Projeto Final - Relatorio de Controle de Acesso\Log_Controle_Acesso.txt"


    Open diretorio For Output As #1


        For i = 0 To relatorio.ListCount - 1
            log = relatorio.List(i)
            Print #1, log
        Next
```

```
    Close #1

End Sub

' ------------------------------------------------------------------------------------------------------------
-------

' Sub para executar comandos Shell (via CMD)  /   Autor: MACORATTI, Jose Carlos Fonte:
http://www.macoratti.net/d250901.htm

' ------------------------------------------------------------------------------------------------------------
-------

Private Sub ExecCmd(cmdline$)

    Dim proc As PROCESS_INFORMATION
    Dim start As STARTUPINFO

    'Inicia a strutura STARTUPINFO
    start.cb = Len(start)

    'Inicia a aplicação escolhida para ser executada
    ret& = CreateProcessA(0&, cmdline$, 0&, 0&, 1&, _
    NORMAL_PRIORITY_CLASS, 0&, 0&, start, proc)

    'Aguarda a aplicação iniciada terminar
    ret& = WaitForSingleObject(proc.hProcess, INFINITE)
    ret& = CloseHandle(proc.hProcess)
```

```
End Sub

' ------------------------------------------------------------------------------------------------------------
' Chama a função de gerar o relatório e a função para encerrar o programa
' ------------------------------------------------------------------------------------------------------------

Private Sub menuSair_Click()

    Call SGerarRelatorioTXT 'Gera o relatório gerencial

    Call Form_Terminate

    Call Unload(Me)

End Sub
```

Transcreve-se abaixo o código da classe Java que envia a mensagem APDU para autenticação do PIN:

```
import javax.smartcardio.*;

import java.sql.*;
import java.util.List;
import java.util.ListIterator;

import com.sinergbrasil.jc.*;

//Classe responsavel por autenticar o PIN do usuario
public class AutenticaSenha  extends java.lang.Object  {

    // APDU de Select
    private static CommandAPDU SELECT_APDU = new CommandAPDU(

        0x00, 0xa4, 0x04, 0x00,
        new byte[] {(byte)0xa0,(byte)0x00,(byte)0x00,(byte)0x00,(byte)0x62,(byte)0x03,(byte)0x01,(byte)0x0d,(byte)0x01,(byte)0x02}
    );

    private static int id;
    private static String sqlSELECT;
    private static String pin_card;
    private static String sqlINSERT;
```

```
public AutenticaSenha() {
}

public static void main(java.lang.String[] argv) {

        try {
                // Leitores de Smart Card
                TerminalFactory factory = TerminalFactory.getDefault();
                List terminalList = factory.terminals().list();

                // Seleciona leitor de Smart Card
                CardTerminal terminal = (CardTerminal) terminalList.get(0);

                // Aguarda o cartao ser inserido
                System.out.println("Por favor, insira o Smart Card na leitora...");
                terminal.waitForCardPresent(10000);

                // Faz a conexao com o cartao. "*" para automatico
                Card card = terminal.connect("*");

                // Abertura do canal de comunicaco
                CardChannel ch = card.getBasicChannel();

                // Envia o APDU de SELECT
                ResponseAPDU ra = ch.transmit(SELECT_APDU);
```

```
// Conexao com o Banco de Dados

Class.forName("com.mysql.jdbc.Driver").newInstance();


Connection conexao = DriverManager.getConnection("jdbc:mysql://localhost/db_projetofinal?user=root&password=pgrga13");

Statement stm = conexao.createStatement();

ResultSet rs = stm.executeQuery("select id_user from tb_user order by id_user desc limit 1;");

while (rs.next()) {

  id = rs.getInt("id_user");

}


sqlSELECT = "select pin_card from tb_user where id_user = " + id + ";";


ResultSet rs2 = stm.executeQuery(sqlSELECT);

while (rs2.next()) {

  pin_card = rs2.getString("pin_card");

}


//Autenticar senha

CommandAPDU MENSAGEM_APDU = new CommandAPDU(

        0x33, 0x20, 0x00, 0x00,

        pin_card.getBytes()


);

ra = ch.transmit(MENSAGEM_APDU);
```

```
                    sqlINSERT = "UPDATE tb_user SET sw_autentica = '" +
arrayToHex(ra.getBytes()) + "' WHERE id_user =" + id + ";";
                    stm.executeUpdate(sqlINSERT);

                    conexao.close();
                    card.disconnect(false);

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

    // Converte array de bytes em uma string hexadecimal
    public static String arrayToHex(byte[] data) {
            StringBuffer sb = new StringBuffer();
            for(int i = 0; i < data.length; i++) {
                    String bs = Integer.toHexString(data[i] & 0xFF);
                    if(bs.length() == 1) {
                            sb.append(0);
                    }
                    sb.append(bs);
            }
            return sb.toString();
    }
```

```
}
```

Transcreve-se abaixo o código da classe Java que captura o ATR do *Smart Card*:

```
import javax.smartcardio.*;

import java.sql.*;
import java.util.List;
import java.util.ListIterator;

import com.sinergbrasil.jc.*;


//Classe reponsavel por capturar o ATR do Smart Card
public class ClasseATR extends java.lang.Object {

        private static String nroATR;
        private static String sqlINSERT;
        private static String ultimo_registro;

        public ClasseATR() {
        }

        public static void main(java.lang.String[] argv) {
```

```
try {
        // Leitores de Smart Card
        TerminalFactory factory = TerminalFactory.getDefault();
        List terminalList = factory.terminals().list();

        // Seleciona leitor de Smart Card
        CardTerminal terminal = (CardTerminal) terminalList.get(0);

        // Aguarda o cartao ser inserido
        System.out.println("Por favor, insira o Smart Card na leitora...");
        terminal.waitForCardPresent(10000);

        // Faz a conexao com o cartao. "*" para automatico
        Card card = terminal.connect("*");

        // Abertura do canal de comunicaco
        CardChannel ch = card.getBasicChannel();

        // Captura o ATR do cartao
        ATR atr = card.getATR();
        nroATR = arrayToHex(atr.getBytes());

        // Conexao com o Banco de Dados
        Class.forName("com.mysql.jdbc.Driver").newInstance();
```

```
                    Connection conexao = DriverManager.getConnection("jdbc:mysql://localhost/db_projetofinal?user=root&password=pgrga13");

                    Statement stm = conexao.createStatement();


                    ResultSet rs = stm.executeQuery("select id_user from tb_user order by id_user desc limit 1;");

                    while (rs.next()) {

                        ultimo_registro = rs.getString("id_user");

                    }


                    sqlINSERT = "UPDATE tb_user SET atr_card = '" +  nroATR + "' WHERE id_user =" + ultimo_registro + ";";

                    stm.executeUpdate(sqlINSERT);


                    System.out.println("ATR Capturado!");


                    conexao.close();

                    card.disconnect(false);


                } catch (Exception e) {

                    e.printStackTrace();

                }

            }



            // Converte array de bytes em uma string hexadecimal

            public static String arrayToHex(byte[] data) {
```

```
            StringBuffer sb = new StringBuffer();
            for(int i = 0; i < data.length; i++) {
                String bs = Integer.toHexString(data[i] & 0xFF);
                if(bs.length() == 1) {
                    sb.append(0);
                }
                sb.append(bs);
            }
            return sb.toString();
        }

}
```

Transcreve-se abaixo o código da classe Java que envia a mensagem APDU para gravar o PIN no *Smart Card*:

```
import javax.smartcardio.*;

import java.sql.*;
import java.util.List;
import java.util.ListIterator;

import com.sinergbrasil.jc.*;
```

```
//Classe responsavel por gravar o PIN do usuario no Smart Card

public class GravarSenha extends java.lang.Object {


        // APDU de Select

        private static CommandAPDU SELECT_APDU = new CommandAPDU(


                0x00, 0xa4, 0x04, 0x00,

                new byte[] {(byte)0xa0,(byte)0x00,(byte)0x00,(byte)0x00,(byte)0x62,(byte)0x03,(byte)0x01,(byte)0x0d,(byte)0x01,(byte)0x02}

        );


        private static int id;

        private static String sqlSELECT;

        private static String pin_card;

        private static String sqlINSERT;


        public GravarSenha() {

        }


        public static void main(java.lang.String[] argv) {


                try {

                        // Leitores de Smart Card

                        TerminalFactory factory = TerminalFactory.getDefault();

                        List terminalList = factory.terminals().list();


                        // Seleciona leitor de Smart Card
```

```
                    CardTerminal terminal = (CardTerminal) terminalList.get(0);

                    // Faz a conexao com o cartao. "*" para automatico
                    Card card = terminal.connect("*");

                    // Abertura do canal de comunicaco
                    CardChannel ch = card.getBasicChannel();

                    // Envia o APDU de SELECT
                    ResponseAPDU ra = ch.transmit(SELECT_APDU);

                    // Conexao com o Banco de Dados
                    Class.forName("com.mysql.jdbc.Driver").newInstance();

                    Connection conexao = DriverManager.getConnection("jdbc:mysql://localhost/db_projetofinal?user=root&password=pgrga13");
                    Statement stm = conexao.createStatement();
                    ResultSet rs = stm.executeQuery("select id_user from tb_user order by id_user desc limit 1;");
                    while (rs.next()) {
                      id = rs.getInt("id_user");
                    }

                    sqlSELECT = "select pin_card from tb_user where id_user = " + id + ";";

                    ResultSet rs2 = stm.executeQuery(sqlSELECT);
```

```
                while (rs2.next()) {
                  pin_card = rs2.getString("pin_card");
                }



                //Gravar Senha
                CommandAPDU MENSAGEM_APDU = new CommandAPDU(
                        0x33, 0x60, 0x00, 0x00,
                        pin_card.getBytes()
                );

                ra = ch.transmit(MENSAGEM_APDU);

                sqlINSERT = "UPDATE tb_user SET sw_grava = '" + arrayToHex(ra.getBytes()) + "' WHERE id_user =" + id + ";";
                stm.executeUpdate(sqlINSERT);

                conexao.close();
                card.disconnect(false);

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


        // Converte array de bytes em uma string hexadecimal
```

```
        public static String arrayToHex(byte[] data) {
                StringBuffer sb = new StringBuffer();
                for(int i = 0; i < data.length; i++) {
                        String bs = Integer.toHexString(data[i] & 0xFF);
                        if(bs.length() == 1) {
                                sb.append(0);
                        }
                        sb.append(bs);
                }
                return sb.toString();
        }

}
```

Transcreve-se abaixo o código da Applet Java inserida no *Smart Card*:

```
package cartao;

import javacard.framework.APDU;
import javacard.framework.Applet;
import javacard.framework.ISO7816;
import javacard.framework.ISOException;
import javacard.framework.JCSystem;
import javacard.framework.OwnerPIN;
import javacard.framework.Util;
```

```
public class GravaValidaSenha extends Applet {

        // Constantes APDU
        private final static byte CLA = (byte)0x33;
        private final static byte AUTENTICAR_PIN = (byte)0x20;

        private final static byte ALTERAR_PIN = (byte) 0x60;
        private final static byte TENTATIVAS_PIN = (byte) 0x03;
        private final static byte TAMANHO_MAX_PIN = (byte) 99;
        private final static byte TAMANHO_MIN_PIN = (byte) 1;
        private final static short SW_PIN_INCORRETO = (short) 0x6300;
        private final static short SW_PIN_NECESSARIO = (short) 0x6301;
        private final static short SW_PIN_MUITO_GRANDE = (short) 0x6E86;
        private final static short SW_PIN_MUITO_PEQUENO = (short) 0x6E87;
        private final static short SW_PIN_BLOQUEADO = (short) 0x6E89;
        private final static short SW_PIN_EM_BRANCO = (short) 0x6E88;

        // Variaveis de escopo
        private byte[] dados = null;
        private OwnerPIN ownerPin;
        private boolean inicioUPIN = false;


        // Metodo construtor da Classe
        public GravaValidaSenha() {
```

```
                ownerPin = new OwnerPIN(TENTATIVAS_PIN,
TAMANHO_MAX_PIN);
                register(); // Registra a instancia da applet
        }


        // Metodo que instala a Applet no Smart Card
        public static void install(byte[] bArray, short bOffset, byte bLength) {

                new GravaValidaSenha();
        }


        // Metodo Autenticar PIN
        private void autenticarPin(APDU apdu) {
                if(!iniciouPIN) {
                        ISOException.throwIt(SW_PIN_EM_BRANCO);
                        return;
                }
                if(ownerPin.getTriesRemaining()==0){
                        ISOException.throwIt(SW_PIN_BLOQUEADO);
                        return;
                }
                byte[] buffer = apdu.getBuffer(); //valor do PIN
                byte byteRead = (byte)(apdu.setIncomingAndReceive()); //tamanho do
PIN
```

```
            if (ownerPin.check(buffer, ISO7816.OFFSET_CDATA,byteRead) ==
false){
                ISOException.throwIt(SW_PIN_INCORRETO);
            }
        }

        // Metodo Alterar PIN
        private void alterarPin(APDU apdu) {
            if (iniciouPIN && !ownerPin.isValidated()){
                ISOException.throwIt(SW_PIN_NECESSARIO);
            }
            byte[] buffer = apdu.getBuffer(); //valor do PIN
            byte numBytes = buffer[ISO7816.OFFSET_LC]; //tamanho do PIN


            if ( numBytes > TAMANHO_MAX_PIN ){
                ISOException.throwIt(SW_PIN_MUITO_GRANDE);
            }
            if ( numBytes < TAMANHO_MIN_PIN ){
                ISOException.throwIt(SW_PIN_MUITO_PEQUENO);
            }
            short offset_cdata = 05;
            ownerPin.update(buffer, offset_cdata, numBytes);
            ownerPin.resetAndUnblock();
            iniciouPIN = true;
        }
```

```
        // Processa a mensagem APDU
        public void process(APDU apdu) throws ISOException {
            // Captura o conteudo da mensagem APDU
            byte[] buffer = apdu.getBuffer();

            // Processa a mensagem APDU de acordo com o INS
            switch (buffer[ISO7816.OFFSET_INS]) {
                case AUTENTICAR_PIN: autenticarPin(apdu);break;

                case ALTERAR_PIN: alterarPin(apdu); break;
            }
        }
}
```

Transcreve-se abaixo o *script* do banco de dados:

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci ;
CREATE SCHEMA IF NOT EXISTS `db_projetofinal` DEFAULT CHARACTER SET latin1 ;
USE `mydb` ;
```

```
USE `db_projetofinal` ;


-- -----------------------------------------------------
-- Table `db_projetofinal`.`tb_user`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `db_projetofinal`.`tb_user` (
  `id_user` INT(4) UNSIGNED NOT NULL AUTO_INCREMENT ,
  `atr_card` VARCHAR(50) NULL DEFAULT NULL ,
  `name_user` VARCHAR(20) NULL DEFAULT NULL ,
  `company_user` VARCHAR(20) NULL DEFAULT NULL ,
  `email_user` VARCHAR(25) NULL DEFAULT NULL ,
  `digital` LONGTEXT NULL DEFAULT NULL ,
  `pin_card` VARCHAR(100) NULL DEFAULT NULL ,
  `sw_grava` VARCHAR(4) NULL DEFAULT NULL ,
  `sw_autentica` VARCHAR(4) NULL DEFAULT '0000' ,
  `event_timestamp` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP ,
  PRIMARY KEY (`id_user`) )
ENGINE = InnoDB
AUTO_INCREMENT = 510
DEFAULT CHARACTER SET = latin1
COMMENT = 'Tabela de usuarios cadastrados';


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

ANEXOS

Transcreve-se abaixo o código de apoio para geração do arquivo CAP:

```
-out EXP JCA CAP
-verbose
-exportpath .
-applet 0xa0:0x00:0x00:0x00:0x62:0x03:0x01:0x0d:0x01:0x02 cartao.GravaValidaSenha
cartao
0xa0:0x00:0x00:0x00:0x62:0x03:0x01:0x0d:0x01 1.0
```

Transcreve-se abaixo o código para instalação da *Applet* no *smart card*:

```
mode_211

enable_trace

establish_context

card_connect

select -AID a000000003000000

open_sc -security 1 -keyind 0 -keyver 0 -mac_key 404142434445464748494a4b4c4d4e4f -
enc_key 404142434445464748494a4b4c4d4e4f // Open secure channel

delete -AID a00000006203010d01

install -file c:\javacard\projetoFinal\cartao\javacard\cartao.cap -nvDataLimit 2000 -instParam
00 -priv 2

getdata

close_sc // Close secure channel

# putkey // Put key

  // options:

  //          -keyind Key index

  //          -keyver Key version

  //          -key Key value in hex

card_disconnect

release_context
```