

Centro Universitário Positivo - UnicenP
Núcleo de Ciências Exatas e Tecnológicas - NCET
Engenharia da Computação
Tiago Maccagnan Moreira

**Verificação Biométrica da Palma da Mão com
Identificação via *Smart Card***

Curitiba
2004

Centro Universitário Positivo - UnicenP
Núcleo de Ciências Exatas e Tecnológicas - NCET
Engenharia da Computação
Tiago Maccagnan Moreira

Verificação Biométrica da Palma da Mão com Identificação via *Smart Card*

Monografia apresentada à disciplina de
Projeto Final, como requisito parcial à
conclusão do Curso de Engenharia da
Computação. Orientador: Prof.
Alessandro Zimmer

Curitiba
2004

Termo de Aprovação

Tiago Maccagnan Moreira

Verificação biométrica da palma da mão com identificação via *smart card*

Monografia aprovada como requisito parcial à conclusão do curso de Engenharia da Computação do Centro Universitário Positivo, pela seguinte banca examinadora:

Prof. Alessandro Zimmer. (Orientador)

Prof. Adriana Cursino Thomé.

Prof. Álvaro Cantieri.

Curitiba, 04 de Dezembro de 2004.

Agradecimentos

Agradeço em especial meus familiares que sempre colaboraram de uma forma ou de outra para que este projeto tenha sido feito com êxito. Agradeço também os meus amigos que sempre me deram incentivo para ultrapassar todos os obstáculos em busca do objetivo.

Sumário

LISTA DE TABELAS	10
LISTA DE ABREVIATURAS.....	11
RESUMO.....	13
ABSTRACT.....	14
1 INTRODUÇÃO	15
2 ESTUDO TEÓRICO	17
2.1 BIOMETRIA.....	17
2.1.1 Componentes de um Sistema Biométrico.....	19
2.1.2 Como Funcionam os Sistemas Biométricos	20
2.1.3 Falsa Aceitação e Falsa Rejeição.....	22
2.1.4 A Palma da Mão	24
2.2 ESPECIFICAÇÃO	26
2.3 DESCRIÇÃO	26
2.4 ESPECIFICAÇÃO DO HARDWARE	28
2.4.1 SMART CARDS.....	28
2.4.2 Micro-controlador 8031	29
2.5 ESPECIFICAÇÃO DO SOFTWARE	33
2.5.1 Aquisição da Imagem	33
2.5.2 Cadastros das imagens adquiridas.....	34
2.5.3 Cadastro das informações do usuário	35
2.5.4 Processamento da imagem	35
2.6 ESTIMATIVA DE CUSTOS	37
3 PROJETO	39
3.1 PROJETO DE SOFTWARE	39
3.1.1 Aquisição da imagem.....	39
3.1.2 Processamento da Imagem	39
3.1.2.1 Conversão para Escala de Cinza	41
3.1.2.2 Filtro Gaussiano.....	42
3.1.2.3 Binarização	43
3.1.2.4 Limiar Automático de OTSU	44
3.1.2.5 Seleção da Região da Palma da Mão	45

3.1.2.6	Algoritmo de <i>Sobel</i>	45
3.1.2.7	Geração do <i>Biocode</i>	46
3.1.3	<i>Sistema de cadastro</i>	47
3.1.3.1	Cadastro de Usuários	47
3.1.3.2	Verificação do Acesso	49
3.1.3.3	Banco de Dados	51
3.2	PROJETO DE HARDWARE.....	52
3.2.1	<i>Leitura e Escrita no smart card</i>	52
3.2.2	<i>Programa do 8031</i>	56
3.2.3	<i>Elaboração do circuito micro-controlador</i>	58
3.3	CRONOGRAMA	59
3.4	PROTÓTIPO DE TELAS	59
3.4.1	<i>Tela de Login</i>	60
3.4.2	<i>Tela Principal</i>	60
3.4.3	<i>Tela de Cadastro de Usuários</i>	61
4	IMPLEMENTAÇÃO.....	63
4.1	IMPLEMENTAÇÃO DO MÓDULO DE HARDWARE	63
4.1.1	<i>Módulo de comunicação serial com o PC</i>	63
4.1.2	<i>Módulo do circuito micro controlador (integrado com o smart card)</i> 64	
4.1.3	<i>Módulo do protocolo de comunicação com o smart card</i>	66
4.2	IMPLEMENTAÇÃO DO MÓDULO DE SOFTWARE	66
4.2.1	<i>Aquisição da Imagem</i>	67
4.2.2	<i>Processamento da Imagem</i>	67
4.2.3	<i>Geração do Biocode</i>	70
4.3	TESTES E VALIDAÇÃO DO PROJETO.....	72
4.3.1	<i>Testes do Módulo de cadastro de usuário</i>	72
4.3.2	<i>Testes do Módulo de Leitura do Smart Card</i>	72
4.3.3	<i>Testes do Módulo de Reconhecimento Biométrico</i>	72
4.3.4	<i>Testes do Sistema Completo</i>	73
5	RESULTADOS E CONCLUSÕES	74
	REFERÊNCIAS BIBLIOGRÁFICAS	77
	ANEXO 1 – CRONOGRAMA.....	79
	ANEXO 2 – DIAGRAMA DE CASOS DE USO DO SOFTWARE....	80
	ANEXO 3 – MANUAL DO USUÁRIO E MANUAL TÉCNICO.....	81

ANEXO 4 – ARTIGO.....	108
------------------------------	------------

Lista de Figuras

Figura 1 – Percentual de participação das tecnologias de biometria no total.....	19
Figura 2 – Ilustração de falsa aceitação versus falsa rejeição.	23
Figura 3 - Imagem da palma da mão.	24
Figura 4 – Ilustração da palma da mão com as linhas de estudo.....	25
Figura 5 - Diagrama em blocos do sistema completo.	26
Figura 6 – Divisão interna de memória de um smart card de 1Kbyte.....	29
Figura 7 – Esquemático de implementação do circuito completo.....	31
Figura 8 – Diagrama em blocos do circuito completo.....	32
Figura 9 – Diagrama de caso de uso de cadastro das mãos.	35
Figura 10 – Cadastro de novo usuário.	35
Figura 11 – Resultados das etapas de processamento da imagem.....	37
Figura 12 - Seqüência do processamento da Imagem.....	40
Figura 13 - Fluxograma detalhado do cadastro de usuários.	49
Figura 14 - Fluxograma de Verificação do Acesso.....	50
Figura 15 - Estrutura do Banco de Dados	52
Figura 16 - Sinais de controle para a escrita no smart card.	53
Figura 17 - Estabilidade dos dados no barramento e condições de Start e Stop bits.	53
Figura 18 - Protocolo de início de comunicação entre <i>smart card</i> e micro-controlador.	54
Figura 19 - Protocolo para a gravação de dados no <i>smart card</i>	55

Figura 20 - Protocolo para a leitura de dados no <i>smart card</i>	55
Figura 21 - Fluxograma da comunicação entre 8031 e smart card.	57
Figura 22 - Reset no micro-controlador.	58
Figura 23 - Tela de <i>Login</i>	60
Figura 24 - Protótipo da tela principal.	61
Figura 25 - Protótipo da tela de cadastro de usuários.	62
Figura 26 - Palma da mão com as linhas evidenciadas.	68
Figura 27 - Resultados do processamento da imagem.	70
Figura 28 - Palma da mão dividida em 16 regiões.	71
Figura 29 - Palma da mão separada em 16 regiões.	71
Figura 30 - Gráfico de Dispersão dos usuários cadastrados.	75

Lista de Tabelas

Tabela 1 – Comparação entre tecnologias biométricas.	18
Tabela 2 – Estimativa de custos do projeto.....	37

Lista de Abreviaturas

RG – Registro Geral.

DTC – *Distal Transverse Crease*.

PTC – *Proximal Transverse Crease*.

TC – *Thenar Crease* – Linha Thenar.

I2C – *Inter Integrated Circuit*.

Bits – *Binary Digits*.

Kbyte – *Kilobyte*.

EEPROM – *Electrically Erasable/Programmable Read Only Memory*.

ROM – *Read Only Memory*.

RAM – *Random Access Memory*.

EPROM – *Erasable/Programmable Read Only Memory*.

PSEN – *Programmable Strobe Enable*.

ALE – *Address Latch Enable*.

SCL – *Serial Clock*.

SDA – *Serial Data*.

PC – *Personal Computer*.

DOS – *Difference of Slopes*.

RGB – *Red Green Blue*.

SQL – *Structured Query Language*.

t_{wr} – *Time Write*.

MSB – *Most Significant Bit.*

LSB – *Least Significant Bit.*

R ou RD – *Read.*

W ou WR – *Write.*

ACK – *Acknowledgment.*

LED – *Light Emissor Diode.*

RST – *Reset.*

RISC – *Reduced Instruction Set Computing.*

UART - *Universal Asynchronous Receiver/Transmitter.*

REN – *Enable Receiver.*

IE – *Interrupt Enable.*

RESUMO

O estudo biométrico realizado neste projeto tem por objetivo analisar e identificar padrões de reconhecimento que formam as linhas da palma da mão das pessoas. As mãos, previamente coletadas através de um *scanner* de mesa comum, serão analisadas e processadas por métodos pertinentes para a geração de um *biocode*. Cada pessoa possui seu próprio *biocode*, que será gravado dentro do seu *smart card* pessoal.

A utilização de *smart cards* no projeto é tida como a garantia da identificação única do indivíduo, devido à exclusividade que o *biocode* proporciona. Neste projeto, a identificação do usuário será feita pelo *smart card* e a 'senha' associada a este cartão é a verificação biométrica das linhas da palma da mão - *biocode*.

Palavras-chave: biometria, *smart cards*, linhas da palma da mão, segurança de dados.

ABSTRACT

The objective of this research is to analyse and identify rules and aspects of hand recognition. The hands, which data is obtained from one scanner, will be analysed and processed in order to achieve the biocode. Each user shall have his own biocode, which will be consequently recorded in a personal smart card.

The hand skin has particular details that vary from person to person. Each small detail contribute to the achievement of the biocode. The hand surface varies in such a way that no human shall have the same combination of the other, which adds up to unprecedented security in identification purposes.

The use of smart cards guarantees exclusive identification, because of the differences of the biocode. The user identification is made by the smart card, along with the built-in password and hand surface identification.

Index-terms: *smart cards, biometrics, hand print, data security.*

1 INTRODUÇÃO

A cada ano que passa, devido à evolução dos componentes eletrônicos e tecnologias computacionais, novas metodologias são elaboradas para fazer o controle de pessoas autorizadas a locais de acesso restrito com maior facilidade, garantindo sempre a restrição de usuários que não possuem acesso ao local.

Algumas tecnologias atuais para o controle de acesso pessoal são cartões magnéticos (*Magnetic Tags*), *Smart Cards*, e diversos outros, que permitem ao portador do cartão ser portador de um código único e exclusivo de identificação (código pessoal) que o associa aos locais cuja permissão é válida para o acesso.

Um dos maiores problemas encontrados atualmente na segurança de locais de acesso restrito é a vulnerabilidade que os cartões magnéticos proporcionam, devido a facilidade de duplicação destes cartões, estes podem ser facilmente violados e, uma vez roubados, o acesso pode ser feito por pessoas não autorizadas e sem a presença do usuário cadastrado.

Um outro problema que existe hoje em dia é o uso das senhas pessoais. Devido ao grande número de locais restritos que usamos atualmente, faz-se necessário a utilização de uma senha em cada um destes locais. Isto acaba nos gerando dois problemas: ou o usuário utiliza a mesma senha para todos os lugares restritos, e caso a senha de um determinado local de acesso seja descoberta, todos os outros lugares restritos ao qual este usuário faz parte podem ser violados, ou o usuário tem uma senha diferente para cada local de uso restrito, causando um possível esquecimento ou confusão entre as senhas, o que é muito comum atualmente também.

A senha pessoal pode ser facilmente substituída por alguma característica física do usuário que seja extremamente individual, como por exemplo, uma verificação de impressão digital, uma verificação de íris, ou mesmo uma verificação de algumas características da palma da mão do usuário.

Estas características podem ser extraídas através de um meio computacional específico e apropriado. O estudo destas características físicas é

chamado de Biometria. Como estas características são individuais e já estão presentes nas pessoas desde o seu nascimento, estas podem muito bem substituir o uso das senhas anteriormente citadas, garantindo assim que somente na presença do usuário cadastrado, o local seja acessado.

Por biometria entende-se identificação automática de uma pessoa baseada nas suas características fisiológicas (como as linhas da palma da mão) ou comportamentais, que são únicas de indivíduo para indivíduo. Algo que antes parecia uma tecnologia inacessível, hoje já é um caminho para o futuro.

Sabendo-se que este projeto está inserido em uma área de grande crescimento e de grandes incentivos ao desenvolvimento, este projeto propõe, com o uso de técnicas de processamento de imagens e de soluções de engenharia, construir um *software* e um *hardware* capazes de identificar o usuário através do seu *smart card* pessoal e fazer a confirmação do usuário com base na leitura da sua palma da mão.

2 ESTUDO TEÓRICO

Neste ponto serão descritas algumas das técnicas e aspectos teóricos que envolvem as tecnologias das quais este projeto irá utilizar para conseguir atingir os resultados desejados.

2.1 Biometria

A biometria é o ramo da ciência que estuda a mensuração dos seres vivos. Tecnologias biométricas são definidas como métodos automáticos de verificação ou identificação de identidade de uma pessoa viva baseados em características fisiológicas ou de comportamento [2].

Estudos de identificação biométrica têm sido muito abordados ultimamente, principalmente com a finalidade de dificultar as chances de ocorrer fraudes com relação à segurança de dados. Após algum tempo de pesquisa, constatou-se que as linhas da palma da mão são tão peculiares quanto às impressões digitais. Estas linhas são formadas enquanto as pessoas ainda estão na barriga da mãe, ainda nas primeiras semanas de gestação da mulher e permanecem no corpo durante toda a vida (caso nenhum acidente grave interfira nas características físicas da palma da mão da pessoa).

Ainda que os sistemas biométricos não possam ser usados para estabelecer uma certeza absoluta na identificação pessoal, como algumas tecnologias atuais como *smart cards*, cartões magnéticos, e até mesmo o código de barras, estes sistemas podem ser utilizados para fazer uma identificação positiva, com um alto grau de confiança e velocidade em grande escala sem mesmo que o usuário tenha conhecimento, como os sistemas biométricos fisiológicos como o do reconhecimento da face e os sistemas comportamentais como os de reconhecimento da voz ou da digitação.

As diferenças entre métodos de comportamento e fisiológicos são importantes por vários motivos. Primeiro, o grau de variação intrapessoal numa característica física é menor do que em uma característica de comportamento. Exemplificando, isto significa que, com exceção de algum ferimento, sua palma da

mão é a mesma ao longo da sua vida. Uma assinatura, por outro lado, é influenciada tanto por fatores fisicamente controláveis como por fatores emocionais. Assim, sistemas baseados em comportamento têm um grande trabalho em ajustar as variações intrapessoais. Por esse motivo, é mais fácil construir um sistema que, por exemplo, guie o usuário a colocar a palma de sua mão sempre em determinada posição, do que implementar um algoritmo que traduza o estado emocional de uma pessoa.[2]

A Tabela 1 apresenta uma comparação entre as tecnologias biométricas apresentadas anteriormente, levando em consideração três fatores [3]:

- Desempenho: refere-se à capacidade de um sistema em autenticar corretamente um indivíduo devido a um tipo de característica biométrica.
- Aceitabilidade: indica o quanto as pessoas aceitam esse tipo de identificação biométrica na sua vida cotidiana.
- Fraudável: reflete a facilidade com que um sistema pode ser enganado por métodos fraudulentos.

Tabela 1 – Comparação entre tecnologias biométricas.

Característica Biométrica	Desempenho	Aceitabilidade	Fraudável
Impressões digitais	Alta	Média	Baixa
Olhos	Alta	Baixa	Baixa
Mãos	Média	Média	Média
Face	Baixa	Alta	Alta
Digitação	Baixa	Média	Média
Voz	Baixa	Alta	Baixa
Assinaturas	Baixa	Alta	Alta

A Figura 1 mostra um gráfico comparativo das pesquisas das tecnologias de biometria estimadas para 2004. [4]

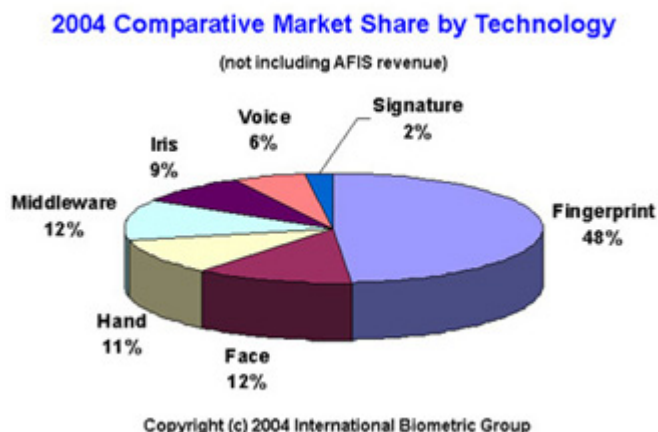


Figura 1 – Percentual de participação das tecnologias de biometria no total.

2.1.1 Componentes de um Sistema Biométrico

Um sistema biométrico padrão possui os seguintes componentes:

- Um dispositivo de medida, o qual forma a interface de entrada do usuário. A facilidade de uso é um fator importante para os biométricos: o dispositivo deve deixar pouca possibilidade para erros. Ele deve ser satisfatório para o uso de uma grande quantidade de pessoas, incluindo aquelas que são destreinadas;
- Um *software* de operação, incluindo o algoritmo matemático que irá checar a medida contra um modelo (*template*). Os algoritmos mais recentes dependem menos da modelagem estatística e mais da programação dinâmica, das redes neurais, e da lógica *fuzzy* (*fuzzy logic*). Isto aumenta sua flexibilidade. Eles são menos suscetíveis a rejeitar alguém por causa de uma sujeira, por exemplo, se o resto do modelo estiver de acordo;
- Um *hardware* e sistemas externos em que a usabilidade, confiança e o custo do sistema irá frequentemente depender tanto destes sistemas externos como dos dispositivos de medida. Sistemas de análise da palma da mão, por exemplo, são bem utilizados na implementação voltada para sistemas centralizados.

2.1.2 Como Funcionam os Sistemas Biométricos

O processo de verificação da biometria do usuário inclui passos distintos de identificação e autenticação. A segurança e desempenho dos procedimentos de identificação e autenticação podem afetar diretamente outras funções do sistema. A identificação é a forma com que o usuário fornece sua identidade única, que pode representar um indivíduo ou um grupo de indivíduos, ao sistema. A autenticação é o processo que associa o indivíduo àquela identificação única, ou seja, é a forma pela qual o usuário estabelece a validação de sua identidade. A autenticação é a prova da identificação [6].

Os procedimentos de autenticação são baseados em três formas básicas, que podem ser combinadas para garantir maior confiabilidade no processo de autenticação:

- Autenticação por propriedade: algo que o usuário possua, como uma chave ou cartão;
- Autenticação por conhecimento: algo que o usuário saiba, como uma senha;
- Autenticação por característica: alguma característica comportamental ou fisiológica do usuário [7].

Todos os sistemas biométricos funcionam essencialmente da mesma forma. Inicialmente, o sistema captura uma amostra da característica biométrica durante o processo de cadastramento do usuário (para alguns sistemas pode ser necessário que mais de uma amostra seja fornecida para a elaboração do perfil da característica biométrica). Deste conjunto de amostras, são extraídas características únicas que são convertidas pelo sistema num código matemático, que é armazenado como o *template* para aquele usuário [7].

O passo seguinte é a comparação de uma nova amostra ao *template* armazenado. O usuário interage mais uma vez com o sistema biométrico, produzindo uma nova amostra, que será comparada com um *template* específico

(verificação) ou com todos os *templates* armazenados na base de dados (identificação). A partir desta comparação, o sistema decidirá se as características retiradas da amostra são suficientemente similares às características contidas no *template* armazenado, confirmando ou não a coincidência (*match*).

O processo de identificação é um processo um-para-muitos, onde uma amostra é submetida ao sistema, que a compara com todos os *templates* de referência contidos em uma base de dados, a fim de verificar se esta amostra coincide com qualquer um destes *templates* e, em caso positivo, determina a identidade do usuário a quem aquele *template* pertence.

O processo de verificação é um processo um-para-um, onde o sistema verifica a identidade de um usuário comparando a amostra com um *template* específico. Através de uma identificação fornecida, o sistema localiza o *template* desejado e o compara com a amostra apresentada. Se houver a coincidência da amostra com o *template* armazenado, o sistema confirma que o usuário realmente possui a identidade afirmada. Este é o próprio processo de autenticação do usuário [7].

Os sistemas biométricos apresentam variações na medida das características ou comportamento humanos [7], justamente porque os seres humanos são inconsistentes e suas características físicas e comportamentais podem variar sutilmente com o passar do tempo. Além disso, a própria forma como o usuário interage com a máquina nunca é constante. *Para que os sistemas biométricos possam lidar com estas variações, um limite de tolerância deve ser definido* [7]. Cada sistema biométrico tem seu método próprio de associar um escore para a coincidência biométrica – é definido um valor limite que determina quando a coincidência ocorre. Um escore acima do valor limite pode ser considerado "sucesso" e, um escore abaixo deste valor é considerado "não-sucesso" [7].

Assim, a comparação entre a nova amostra e o *template* deve ultrapassar o valor limite para que a amostra seja considerada suficientemente similar ao *template* e o sistema considere que as duas realmente coincidem. Caso contrário, o sistema não identificará a coincidência e o usuário não será reconhecido. Os

sistemas devem permitir que o valor limite seja alterado a fim de aumentar ou diminuir o nível de segurança, conforme a necessidade da aplicação e do nível de aceitação por parte dos usuários [7].

A forma de armazenamento dos *templates* depende do tipo de aplicação para qual o dispositivo biométrico será utilizado e do tamanho dos *templates*. Geralmente os *templates* são armazenados na memória do próprio dispositivo, numa base de dados centralizada ou em cartões magnéticos [7].

A utilização de uma base de dados central para o armazenamento de *templates* é a solução adequada para aplicações onde o número de usuários é grande ou quando é necessária verificação remota. Este processo pode ser comprometido quando a segurança dos dados é ameaçada por sistemas de comunicação ou redes vulneráveis ou por abuso de privilégios na manipulação da base de dados. O método de armazenamento de *templates* utilizando cartões magnéticos permite que o usuário carregue seu próprio *template* para a utilização nos dispositivos de verificação, sendo indicado para aplicações onde o grupo de usuários seja numeroso demais para ser armazenado numa base de dados central, quando é necessário que os usuários sejam verificados remotamente ou quando há necessidade de uma transmissão rápida dos *templates*.

2.1.3 Falsa Aceitação e Falsa Rejeição

Para avaliar a habilidade do sistema em permitir o acesso de usuários autorizados e proibir o acesso de usuários sem permissão, são utilizadas duas medidas de performance que consideram o nível de acuracidade da coincidência, que são a Taxa de Falsa Rejeição e a Taxa de Falsa Aceitação. Erros do tipo falsa rejeição ocorrem quando uma coincidência legítima é negada, ou seja, quando uma coincidência legítima não gera um escore acima do valor limite. Erros do tipo falsa aceitação ocorrem quando um impostor é aceito como válido, ou seja, um impostor gera um escore acima do valor limite. A configuração do valor limite para tolerância a estes erros é crítica na performance do sistema. A falsa rejeição causa frustração e a falsa aceitação ocasiona fraude [7].

A taxa de falsa aceitação pode ser representada pela aceitação pelo sistema do indivíduo que não efetuou o cadastro, aonde a amostra não corresponde a nenhum indivíduo habilitado a utilizar o sistema. Este é um ponto muito perigoso nos sistemas, justamente por isso, este valor deve ser o mais próximo possível de zero.

A taxa de falsa rejeição indica qual é o percentual de indivíduos que não conseguirão passar pelo sistema, mesmo apresentando a biometria correta, ou seja, o usuário cadastrado no sistema não possui acesso devido a alguma característica momentânea e passageira.

Os dispositivos biométricos físicos tendem a ter uma melhor taxa de falsa aceitação por causa da estabilidade da característica medida e porque as características comportamentais são mais fáceis de serem duplicadas por outros usuários. A configuração do valor limite para tolerância a estes erros é crítica no desempenho do sistema. Vale mais a pena submeter o usuário a fornecer uma nova amostra da sua palma da mão do que liberar o acesso em caso de dúvida. A Figura 2 demonstra um gráfico de taxas de falsa aceitação e falsa rejeição e o valor limite de cada tipo [8].

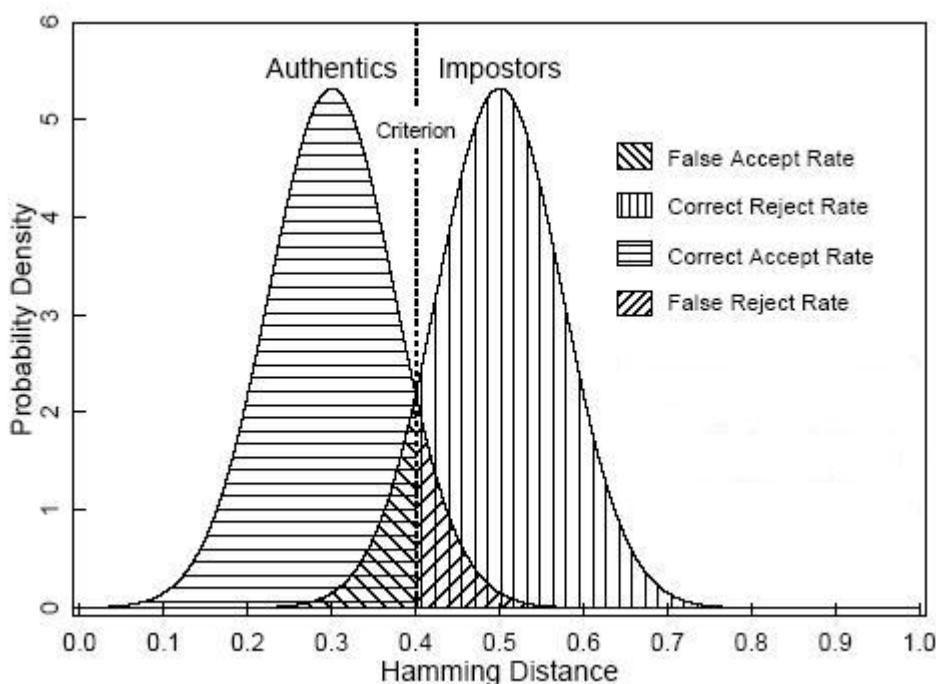


Figura 2 – Ilustração de falsa aceitação versus falsa rejeição.

2.1.4 A Palma da Mão

A palma da mão numa pessoa é formada ainda na barriga da mãe, enquanto ainda é o 7º mês de gestação e é uma característica que permanece na pessoa até o fim da vida, não sofrendo alterações e nem mudanças naturalmente.

Algumas das linhas da palma da mão serão extremamente importantes durante o estudo e implementação deste projeto. A Figura 3 mostra a imagem da palma da mão retirada pintando-se a mão com tinta preta e colocando no papel [10].



Figura 3 - Imagem da palma da mão.

As principais características que serão analisadas neste projeto são as linhas da palma da mão. A Figura 4 mostra um desenho ilustrativo da palma da mão por completo e caracteriza as linhas de estudo deste projeto [11].

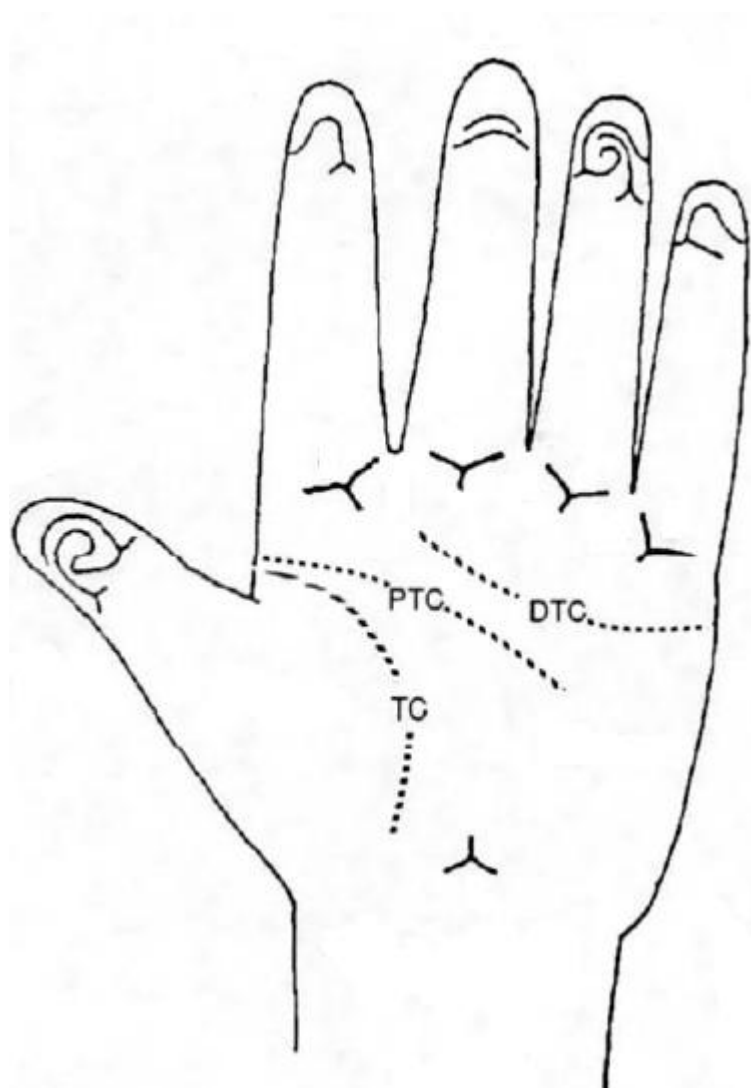


Figura 4 – Ilustração da palma da mão com as linhas de estudo.

Como mostrado acima, são três (03) as características mais importantes da palma da mão são as linhas *DTC* (*Distal Transverse Crease*), *PTC* (*Proximal Transverse Crease*) e *TC* (*Thenar Crease*), que significam respectivamente linha transversa distal, linha transversa proximal e linha *thenar*.

2.2 ESPECIFICAÇÃO

A seguir será explanada a especificação do projeto, onde alguns aspectos das funcionalidades do sistema serão abordados com mais clareza, descrevendo alguns dos requisitos iniciais para se desenvolver o projeto.

2.3 DESCRIÇÃO

O projeto consiste em um sistema de controle de acesso, baseado na identificação do usuário através da inserção do seu *smart card* no local de leitura e depois, colocando a sua mão no *scanner* para que a verificação seja feita. Todo o processamento do *software* consiste em analisar detalhadamente as linhas da palma da mão do usuário e comparar com características peculiares previamente analisadas e cadastradas no banco de dados do sistema. O processamento do *hardware* consiste em fazer a leitura do *smart card*, retirando os dados que possam identificar a qual usuário pertence o mesmo. A Figura 5 demonstra graficamente o diagrama em blocos do sistema completo.

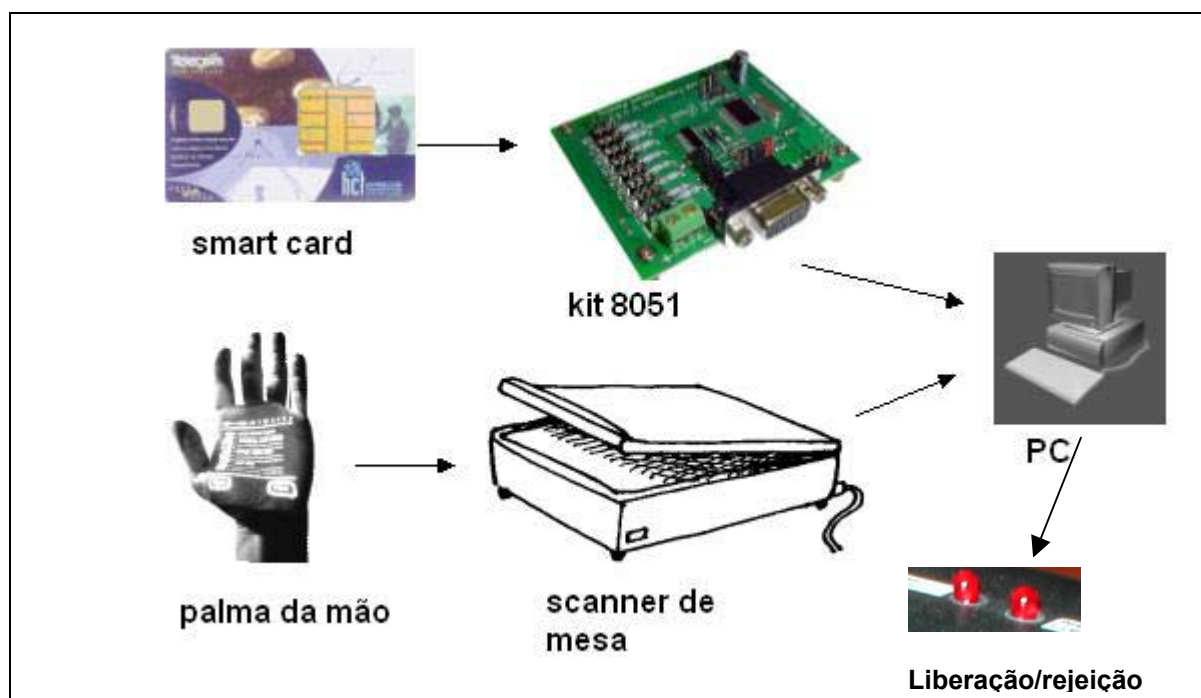


Figura 5 - Diagrama em blocos do sistema completo.

Inicialmente será feita a inserção do *smart card* no soquete de comunicação com a placa controladora do *kit 8051* e será lido o identificador do usuário, gravado neste cartão. O identificador passará através da porta de

comunicação serial para o *software* principal e este já fará a identificação do usuário através do dado coletado. A segunda etapa é adquirir através de um *scanner* de mesa padrão a imagem da palma da mão do usuário e analisá-la, iniciando assim a etapa de reconhecimento biométrico, separando as primitivas de identificação, que no caso deste projeto são as linhas da palma da mão, e verificando as informações obtidas do processamento desta imagem com os dados coletados do *smart card* será possível saber se o acesso será permitido ou não.

Todo o cadastro de usuários e cadastro de informações relevantes ao funcionamento do sistema como um todo será feito por um administrador do sistema, aonde, mediante confirmação feita através de uma senha alfa-numérica, poderá alterar os privilégios dos diversos usuários.

Para efetuar o cadastro da mão, o usuário deverá colocar a mão no *scanner* três (03) vezes, para que o sistema seja capaz de reconhecer as variações de uma mesma palma de mão da mesma pessoa e assim, extrair as primitivas de verificação desta pessoa. Estes valores padrões ficarão armazenados no banco de dados, juntamente com informações pessoais, tais como endereço, telefone e RG.

Portanto, qualquer pessoa que deseje ter acesso ao local protegido ou à informação secreta, deve atender os seguintes requisitos de identificação descritos abaixo:

- O seu *smart card* deverá ser cadastrado na base de dados do sistema;
- A sua palma da mão deverá estar cadastrada no banco de dados do sistema e deve estar dentro dos limites de aceitação para comparação com os dados das outras palmas de mãos armazenadas na base de dados.

2.4 ESPECIFICAÇÃO DO HARDWARE

Este projeto possui, além da verificação da biometria da palma da mão, a identificação do usuário feita via *smart card* e toda a sua manipulação e aquisição dos dados será feita com a utilização de um protocolo definido e de um circuito micro-controlado.

2.4.1 SMART CARDS

Smart card é um cartão em formato convencional de cartão de crédito, envolvido em um plástico que possui no seu interior um *chip* ou de memória ou micro-processado.

A diferença técnica entre *smart cards* do tipo memória (*memory cards*) e *smart cards* propriamente dito é que os cartões de memória funcionam apenas como armazenamento de dados e qualquer equipamento que utilize o protocolo de acesso a esta memória (no caso deste projeto o protocolo de acesso à memória do cartão será o I2C) pode fazer uso deste cartão, enquanto que um cartão genuinamente *smart* tem dentro de si um micro-controlador (variando de 8 bits a 16 bits) que apenas as máquinas de leitura e gravação apropriadas possam interagir com o conteúdo da memória do cartão, pois para fazer acesso a regiões da memória do cartão, faz-se necessário uma chave, diferente para leitura e gravação.

O papel do micro-controlador que o cartão *smart* possui dentro dele é exatamente fazer a verificação destas chaves, tornando assim a manipulação dos dados mais segura. A Figura 6 abaixo exemplifica a divisão de memórias que um cartão *smart* genérico possui.

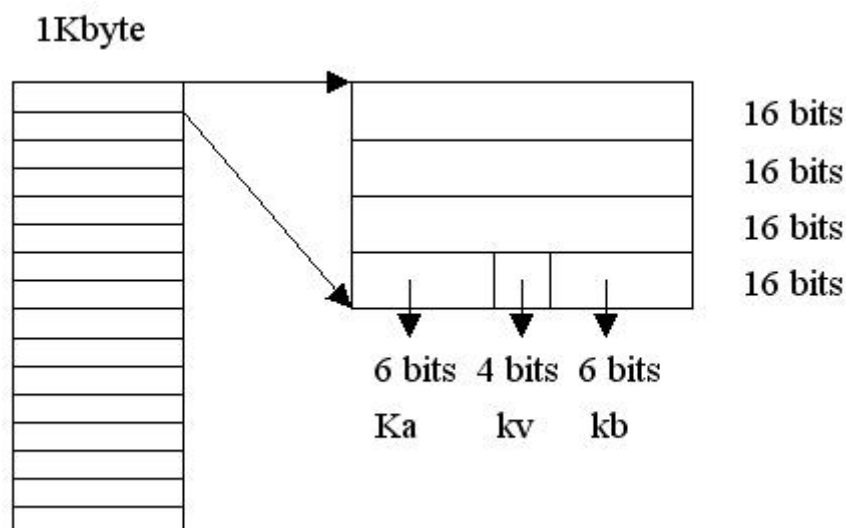


Figura 6 – Divisão interna de memória de um smart card de 1Kbyte.

O *smart card* comprado para o desenvolvimento do projeto tem como características a capacidade de 16Kbytes de armazenamento e é uma memória do tipo EEPROM. Para se fazer o acesso à memória deste cartão é necessário que se siga o protocolo de comunicação I2C.

2.4.2 Micro-controlador 8031

Micro-controlador corresponde a um microprocessador (elemento eletrônico, desenvolvido para executar tarefas específicas, com linguagem de comando específica) e seus periféricos típicos, todos juntos num só componente.

O 8031 é um Micro-controlador rápido com *clock* típico de 12 MHz, sendo que existem versões de até 30 MHz, tem como características um poderoso controlador de *hardware* e *software*, sobretudo em sistema para lógica sequencial e combinatória. Pode trabalhar com até 64 Kbytes de memória de programa e mais 64 Kbytes de memória de dados isoladamente, além de sua memória interna.

Suas características de *hardware* incluem a capacidade de expansão de memória de programa (inicialmente de 4 Kbytes internamente) para até 64 Kbytes totais, colocando mais de 60 Kbytes externos, ou então utilizar 64 Kbytes de memória externa de programa, com a versão 8031, que é mais utilizada

comercialmente, devido ao baixo custo. Com relação ao *software*, permitem a execução de complexas operações aritméticas e lógicas, trabalhar com bancos de registradores nominais e inclusive trabalhar com 128 bits individualmente endereçáveis na RAM.

O barramento de endereços é de 16 bits, em que os 8 bits menos significativos são derivados junto com os 8 bits de dados. O que separa o dado do endereço é um *latch* que copia a informação de endereço, quando este é apresentado em P0. Isto se faz por meio do pino de comando chamado ALE (*Address Latch Enable*). Os dados e endereços são multiplexados pela via P0 (misturados fisicamente na mesma porta P0). Com esse processo são economizados 8 pinos físicos do microprocessador, tornando-o de menor tamanho.

Os outros 8 bits mais significativos do barramento de endereços são dados por P2, e este é direto, não é multiplexado como o P0. O conjunto completo possui 65536 posições diferentes de endereçamento de memória, já que possui 16 bits de endereço ($2^{16}=65536$) e 256 combinações possíveis de dados a serem transmitidas e/ou recebidas, que por sua vez existem 8 bits de dados ($2^8=256$).

Existe um pino de acionamento da ROM (no caso, uma EPROM) chamado PSEN\ e de RAM, que atribuem a função de leitura e escrita de dados.

A Figura 7 demonstra o projeto de todo o circuito referente ao conjunto do micro-controlador 8031 e a interface com o *smart card*.

Todas as adaptações necessárias para o encaixe do cartão serão feitas em uma placa padrão e apenas será uma extensão desta placa de circuito impressa referenciada abaixo. Apenas o soquete de encaixe será montado nesta placa padrão no desenvolvimento do projeto.

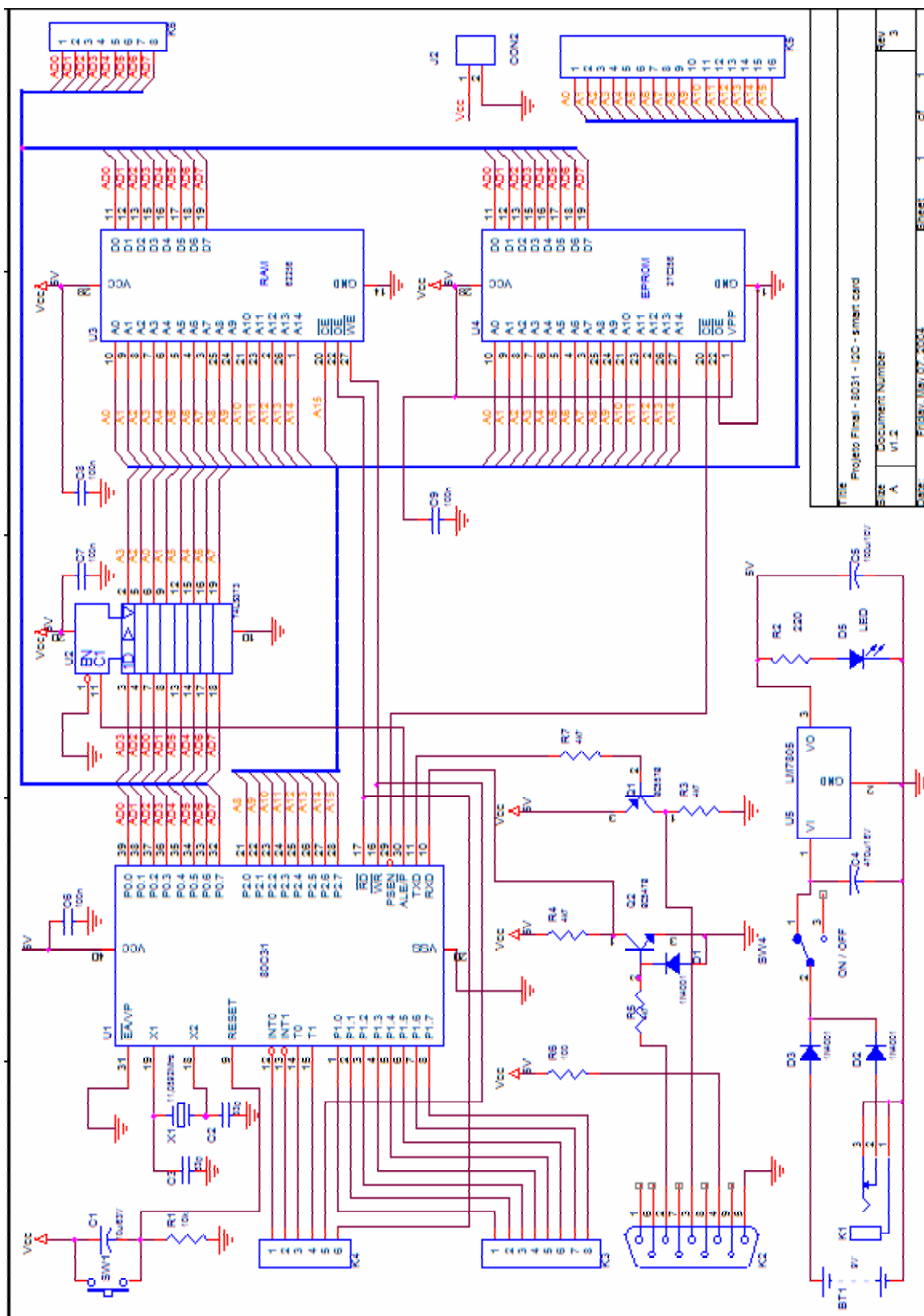


Figura 7 – Esquemático de implementação do circuito completo.

A porta P1 foi destinada para a interface com o *smart card* que neste caso ainda não usará todos os bits destinados pela porta de comunicação P1, apenas os bits que são representados pelos pinos P1.0 e P1.1, que respectivamente serão responsáveis pelo SCL e SDA do *smart card*.

A Figura 8 demonstra o diagrama em blocos de todo o circuito que será implementado.

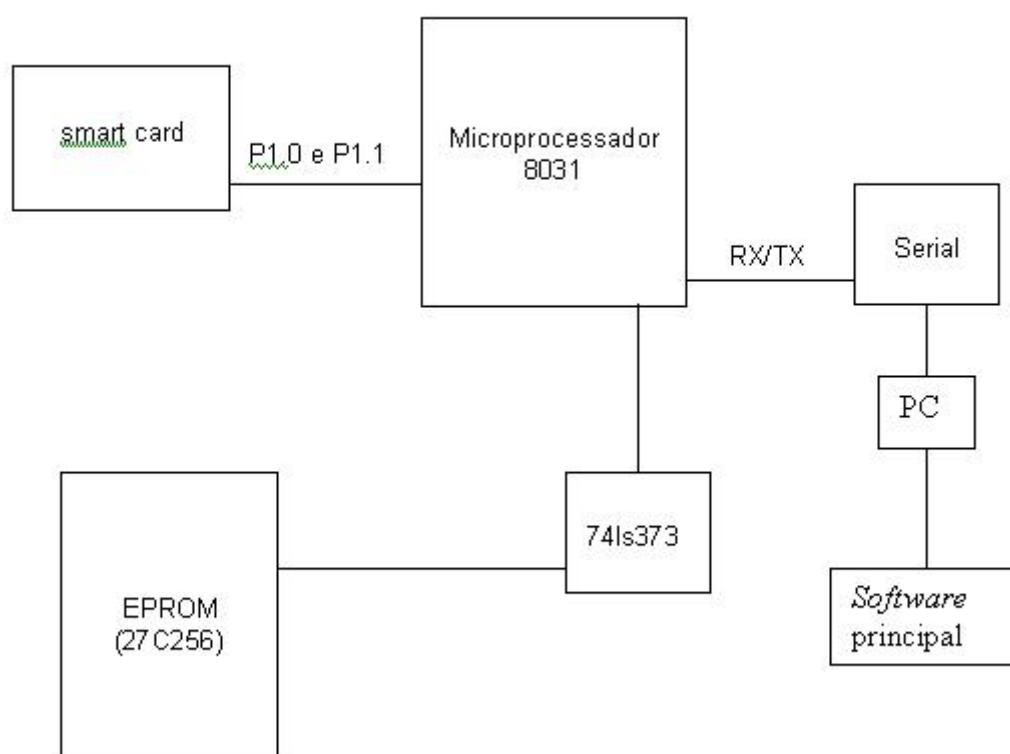


Figura 8 – Diagrama em blocos do circuito completo.

Conforme mostrado na Figura 8, o funcionamento do circuito se dará da seguinte maneira: primeiramente o *software* principal irá enviar serialmente um comando que indicará que o usuário já inseriu seu respectivo *smart card*, posteriormente, o 80C31 irá gerenciar a leitura do identificador do usuário e, em seguida, este será transmitido para o programa principal, para que outras partes do projeto sejam feitas. No circuito, o papel da EPROM (27C256) é de fornecer

todos os comandos que o micro controlador 80C31 irá realizar e o *buffer* 74LS373 serve apenas para manter na linha de dados o comando que está sendo executado, antes de atualizar o seu valor.

2.5 ESPECIFICAÇÃO DO SOFTWARE

Nesta seção serão abordados os módulos que o software contém sendo descritos todas as funcionalidades individualmente e suas metodologias.

2.5.1 Aquisição da Imagem

Em um modo mais natural de se ver, imagens não são diretamente próprias para se fazer uma análise no computador, pois computadores trabalham com informações numéricas e não de figuras, então uma imagem tem que ser convertida para números antes de qualquer tentativa de implementação de métodos de tratamento de imagens. [5]

As imagens digitais são tipicamente informações de uma representação digital copiada do mundo real, por meio de sensor óptico ou artificial, que foi gerada por *software*. As imagens digitais estão relacionadas à área de computação gráfica que trata do estudo da geração, processamento, interpretação e análise destas imagens.

A forma de capturar uma imagem e representá-la em uma matriz de números consiste em “dividir” a imagem real em pequenas regiões denominada elementos da figura, ou mais conhecida como pixels. A imagem é dividida em linhas horizontais de pixels adjacentes e cada parte destas divisões representa a cor e o brilho no determinado ponto real desta imagem. Todo este processo é chamado de digitalização da imagem. No final da amostragem de toda a imagem original, temos como resultado uma matriz numérica que representará a nossa imagem no computador. [5]

Imagens digitais podem ser adquiridas através de câmeras de vídeo, *scanners* ou outros sensores especializados que sejam sensíveis a uma banda do espectro de energia eletromagnética como raios-X, ultravioleta, banda visível ou

banda infravermelha, que produza um sinal elétrico de saída proporcional ao nível de energia percebida e possa ser digitalizado por um dispositivo que converta a saída elétrica para a forma digital. Conseqüentemente um sistema computacional que seja capaz de adquirir uma imagem digital deve permitir o armazenamento em discos e fitas magnéticas, processamento, e a exibição da imagem pela placa de vídeo e monitor.

2.5.2 Cadastros das imagens adquiridas

Todo o procedimento de cadastro das imagens no banco de dados do *software* deve ser feito por um administrador do sistema e não por um usuário comum.

Antes de se iniciar o cadastro e controle das informações contidas no banco de dados, será solicitada a senha de administrador do sistema, e após a confirmação, os registros poderão ser modificados e/ou inseridos.

Serão amostradas cinco (05) vezes a imagem da palma da mão do usuário, para que os padrões da palma da mão sejam extraídos levando-se em consideração o movimento de uma imagem para outra.

Quando as imagens forem cadastradas, serão submetidas a todo o processamento de reconhecimento de padrões, para o administrador poder definir que este padrão encontrado nestas imagens é pertencente ao usuário que acabou de submeter sua mão para o cadastro.

A Figura 9 representa o diagrama de caso de uso do cadastro das imagens das mãos dos usuários.

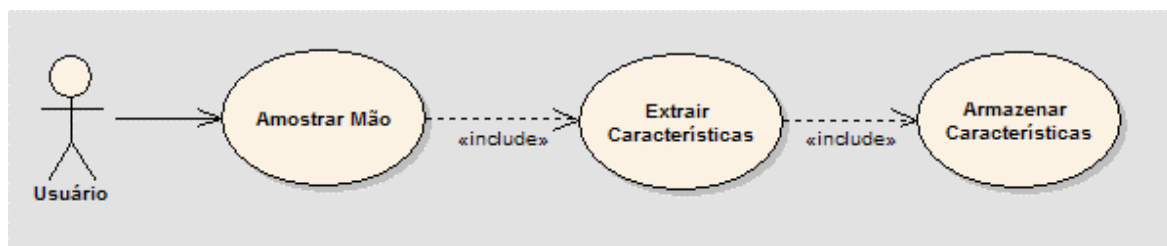


Figura 9 – Diagrama de caso de uso de cadastro das mãos.

2.5.3 Cadastro das informações do usuário

Como todas as formas de cadastro, esta também só poderá ser executada mediante senha de administrador, nunca por um usuário qualquer.

O código das informações pessoais, tais como RG, endereço residencial, telefones e principalmente o cadastro do seu respectivo *smart card*. Para entrar no ambiente, as imagens coletadas serão analisadas com estas imagens cadastradas (*biocodes*).

A Figura 10 a seguir exemplifica o cadastro de usuários feito pelo administrador do sistema.

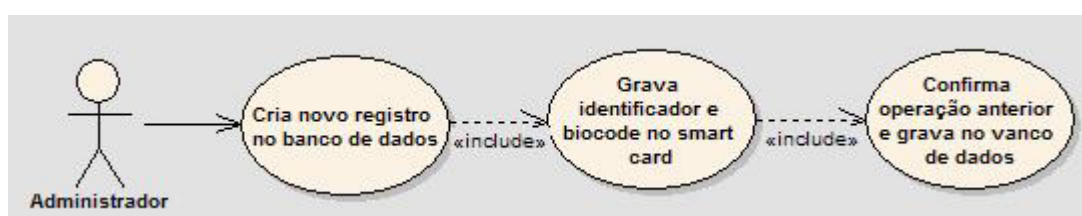


Figura 10 – Cadastro de novo usuário.

2.5.4 Processamento da imagem

A imagem a ser analisada é extraída de um *scanner* de mesa padrão e fica armazenada em memória no programa. Esta imagem recém digitalizada, não estará ainda em condições de análise final, pois conterá informações que não são úteis e na verdade são prejudiciais à eficácia do sistema [1], devendo antes passar pelo pré-processamento, para depois ir para o algoritmo de reconhecimento de padrões. Sendo assim, o tratamento da imagem começa

transformando-se a imagem para níveis de cinza, pois nenhuma informação relacionada à cor da imagem será útil em algum ponto do processamento da mesma e nem para obter o seu padrão.

Com a imagem da região de interesse, o próximo passo é identificar as linhas da palma da mão nesta imagem, mas para isso, são necessários ainda alguns tratamentos, como por exemplo, o processamento feito por um filtro gaussiano. O filtro gaussiano atua para diminuir as regiões de alto contraste da imagem, proporcionando um equilíbrio um pouco maior no histograma gerado pela imagem, a fim de eliminar pequenos ruídos contidos na imagem, para que as linhas sejam salientadas e evidenciadas na imagem.

A partir de então, o método de verificação de padrão é invocado. Tendo a imagem da palma da mão selecionada, filtrada e as suas linhas bem evidentes, é aplicado o algoritmo de *sobel*, para saber a direção de cada pixel em relação ao resto da imagem. Com isto, é possível de se fazer um histograma direcional dos pixels da imagem, a fim de detectar um padrão de inclinação para as linhas da mão.

A Figura 11 representa a seqüência de tratamento de imagens, desde a aquisição da região de interesse até a separação das características de interesse, porém, de uma outra abordagem daquela implementada neste projeto [10].

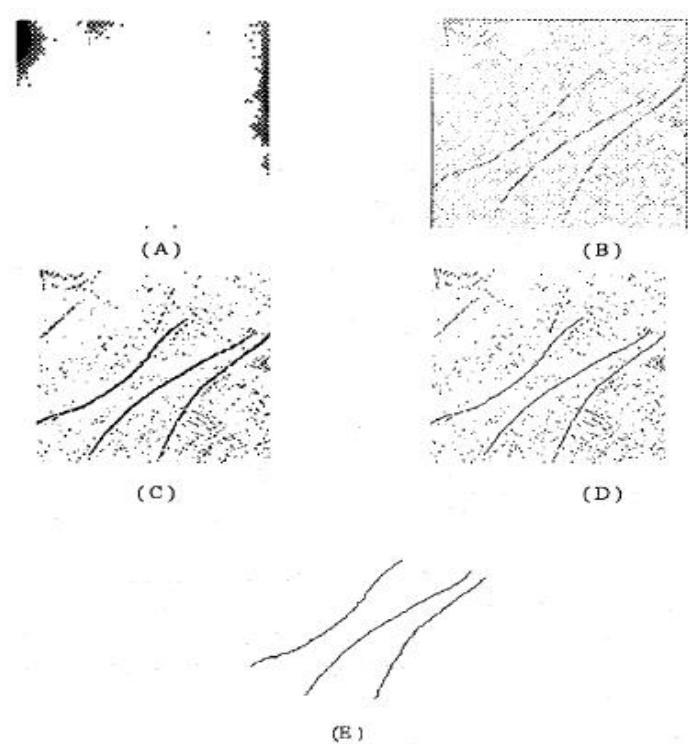


Figura 11 – Resultados das etapas de processamento da imagem.

2.6 ESTIMATIVA DE CUSTOS

Os custos do projeto foram estimados com base no valor de mercado dos itens e componentes que estão sendo utilizados durante a implementação, porém não refletem uma pesquisa detalhada em busca do menor preço praticado.

A Tabela 2, mostrada logo a seguir mostra todos os equipamentos, componentes de *hardware* e *software* que estão sendo utilizados, bem como os seus respectivos preços praticados.

Tabela 2 – Estimativa de custos do projeto.

Recurso	Quantidade	Investimento
Sistema Operacional Microsoft Windows XP Professional	1	R\$800,00
Borland C++ Builder 5.0	1	R\$350,00
Kit UnicenP desenvolvimento 8051	1	R\$75,00
Scanner Hewlet Packard ScanJet 6300J	1	R\$1.500,00
<i>Smart cards</i>	3	R\$30,00
Conector <i>smart card</i>	1	R\$30,00
Computador pessoal AthlonXP 2000+	1	R\$2.000,00

Cristal de 11,0592 MHz	1	R\$1,40
Capacitor 33pF	2	R\$0,30
Capacitor 100nF	4	R\$0,60
Capacitor 470uF	1	R\$0,15
Capacitor 10uF	1	R\$0,15
Capacitor 100uF	1	R\$0,15
Resistores	9	R\$0,90
Micro controlador 80C31	1	R\$4,00
Buffer 74LS373	1	R\$0,75
EPROM 27C256	1	R\$2,00
Conector DB-9	3	R\$2,50
Diodo 1N4004	3	R\$0,60
Regulador de tensão LM7805	1	R\$1,00
Soquetes	4	R\$6,00
Microsoft Office 2000	1	R\$ 200,00
OrCAD Release 9.2	1	R\$ 25.000,00
Horas de desenvolvimento	750	R\$18.750,00
TOTAL		R\$48.755,50

3 PROJETO

Esta parte visa esquematizar todas as funcionalidades do projeto, todas as suas estruturas e todas as rotinas de implementação necessárias para atingir os objetivos descritos anteriormente.

3.1 Projeto de Software

Aqui serão descritas algumas rotinas de implementação relacionadas ao tratamento digital da imagem, processamento de imagens digitais, e métodos do *software* de controle, como cada módulo deverá ser implementado e o motivo de escolha de cada rotina. Para exemplificar como será implementado o sistema, o ANEXO 2 demonstra um diagrama de casos de uso deste *software*.

3.1.1 Aquisição da imagem

Para fazer a aquisição da imagem, está sendo utilizado um *scanner* de mesa padrão e a imagem é salva em memória para ser analisada pelo *software* principal. A resolução do *scanner* que está sendo adotada para este projeto é de 100dpi. A imagem digitalizada tem característica de ser uma imagem de *bitmaps* de 24bits e colorido.

Após a etapa de aquisição da imagem, todo o processamento de imagem se inicia.

O *driver twain* [17] está sendo usado para fazer a captura da palma da mão no *scanner*. Ele é responsável por fazer toda a comunicação entre o *scanner* e o *software* implementado neste projeto.

3.1.2 Processamento da Imagem

O processamento de imagens digitais envolve procedimentos que são geralmente expressos em forma algorítmica. Assim, com exceção da aquisição e exibição de imagens, a maioria das funções de processamento de imagens pode ser implementada via *software* [9].

O processamento de imagens é caracterizado por soluções específicas. Desse modo, técnicas que funcionam bem em uma área podem se mostrar totalmente inadequadas em uma outra. Logo após a digitalização da imagem, faz-se necessário que a mesma seja submetida a alguns procedimentos de filtragem e tratamento, para que o algoritmo de verificação de padrões possa ser implementado e possa retornar um resultado mais preciso.

A Figura 12 demonstra em uma forma genérica como é o fluxograma de processamento da imagem como um todo.

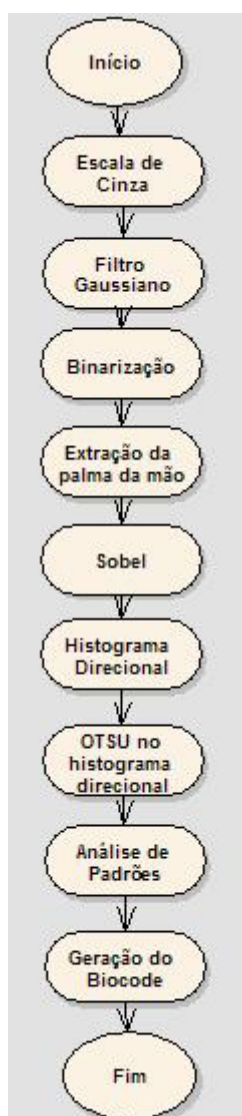


Figura 12 - Sequência do processamento da Imagem.

3.1.2.1 Conversão para Escala de Cinza

Todo o início do processamento se dá retirando as cores que a imagem possui e trabalhando apenas com tons de cinza, pois a imagem coletada possui muitas características que não serão abordadas e nem tratadas pelo programa e uma destas características é a cor e por isso a imagem é convertida para a escala de cinza.

Para fazer a conversão de uma imagem em cores para escala de cinza, é interessante sabermos como as informações que representam as cores são armazenadas. A forma mais comum que encontramos a estrutura de cores dentro do computador é em formato RGB (*Red, Green, Blue*), aonde cada pixel da imagem possui uma componente RGB para representar a sua cor e é armazenada em 3 *bytes*, aonde cada uma das componentes do formato RGB possui 1 *byte* de representação. Com esta maneira de representar as cores das imagens, temos num total uma imagem com 16 milhões de cores.

Para que um pixel seja representado na escala de cinza, ele deve ter as 3 componentes (R, G e B) com o mesmo valor, assim a resposta na tela representará uma imagem acinzentada, sendo que os valores limites para a representação são RGB=0,0,0 para um pixel com uma cor preta e RGB=255,255,255 para a cor branca.

A forma de se converter uma imagem colorida em tons de cinza é fazer uma média aritmética simples entre os valores RGB da imagem, sendo que então, os valores RGB são setados com o valor desta média.

Um pseudocódigo está demonstrando como se faz este procedimento de conversão para a escala de cinza, logo abaixo.

```

Para i de 1 até [Largura da Imagem]
  Para j de 1 até [Comprimento da Imagem]
    ValorCinza=(ValorRPixel(i, j)+ValorGPixel(i, j)+ValorBPixel(i, j))/3;
    Pixel(i, j)=RGB(ValorCinza, ValorCinza, ValorCinza);
  Fim-para;
Fim-para;

```

3.1.2.2 Filtro Gaussiano

A segunda etapa após ter a imagem convertida em tons de cinza é aplicar um algoritmo de filtragem de ruídos. O filtro Gaussiano tem por característica ser do tipo passa-baixa, ou seja, as grandes diferenças da frequência na imagem são atenuadas, deixando a imagem um pouco borrada, porém com a eliminação de algumas componentes irrelevantes no processamento da imagem, comumente chamadas de ruído.

Para aplicar o conceito de filtro gaussiano, é necessário que a imagem esteja pré-processada em escala de cinza, para que os valores não sofram nenhum desvio. A curva de Gauss a ser implementada abrange 15 pontos da imagem, sendo o ponto central o respectivo ponto analisado no momento e os seus 7 vizinhos à esquerda e 7 vizinhos à direita.

Esta implementação é feita com base em dois laços que percorrem a matriz inteira primeiro no sentido horizontal, varrendo a imagem inteira aplicando este conceito de filtro passa-baixa. Quando a imagem for completamente percorrida no sentido horizontal, o laço que faz no sentido vertical se inicia, aplicando os mesmos conceitos e os mesmos valores para este sentido.

Um pseudocódigo está demonstrando como é o funcionamento deste procedimento de filtro da imagem, a seguir.

```

float tfvPixels[15];

tfvPixels={0.05;0.1; 0.2; 0.3; 0.4; 0.6; 0.9; 1; 0.9; 0.6; 0.4; 0.3; 0.2; 0.1; 0.05};

Para i de 1 até [Largura da imagem]
    tiPixel=0;

    Para j de 8 até [Comprimento da Imagem]
        l=0; k=-7; tiPixel=0;

        Enquanto (k<8) faça
            tiPixel+=(MatrizCinza[i][j+k]*tfvPixels[l]);
            k++; l++;

        Fim-Enquanto;

        tiPixel/=6.1;

        MatrizCinza[i][j]=RGB(tiPixel, tiPixel, tiPixel);

    Fim-para;

Fim-para;

Para j de 1 até [Comprimento da Imagem]
    tiPixel=0;

    Para i de 8 até [Largura da Imagem]
        l=0; k=-7; tiPixel=0;

        Enquanto (k<8) faça
            tiPixel+=(MatrizCinza[i][j+k]*tfvPixels[l]);
            k++; l++;

        Fim-Enquanto;

        tiPixel/=6.1;

        MatrizCinza[i][j]=RGB(tiPixel, tiPixel, tiPixel);

    Fim-para;

Fim-para;

```

3.1.2.3 Binarização

Depois do processamento realizado pelo algoritmo do filtro gaussiano, a imagem precisará ser binarizada também, pois existe a necessidade de se trabalhar apenas com 2 valores de cor, a branca e a preta.

Por binarização entende-se a operação de converter uma imagem que se encontra em escala de cinza para apenas dois níveis de cor (branco e preto). A operação é relativamente simples, pois basta analisar cada pixel da imagem e verificar se este se encontra acima ou abaixo de um determinado limite de separação (threshold), sendo que um pixel abaixo deste limite será convertido para a cor preta e um pixel acima deste limite será convertido para branco [1].

Um problema associado pela binarização está na seleção deste limite, pois pode ser diferente para cada imagem, uma vez que a configuração do *scanner* utilizado pode influenciar na distribuição de tons de cinza da imagem.

Abaixo está sendo representado um pseudocódigo de um método simples de binarização, baseado em um limite qualquer.

```

Para i de 1 até [Largura da Imagem]
    Para j de 1 até [Comprimento da Imagem]
        Se ValorRPixel(i, j) > [Threshold] então
            Pixel(i, j)=[Cor Branca];
        Senão
            Pixel(i, j)=[Cor Preta];
        Fim-Se;
    Fim-para;
Fim-para;
  
```

Na implementação do método de binarização, o “método de limiarização de OTSU” [12] é um dos métodos usados.

3.1.2.4 Limiar Automático de OTSU

Para utilizar o método de binarização, na seleção das informações de interesse (linhas da palma da mão), o algoritmo de OTSU será chamado logo após o algoritmo de Sobel.

O método de Otsu se baseia na informação de que em uma imagem ideal, o histograma dos níveis de cinza da imagem possui dois grandes grupos de cores distintos, sendo um dos dados da imagem e o outro do fundo, através de cálculos baseados principalmente em estatísticas e probabilidades, o método de Otsu encontra o vale entre os “picos” do histograma, que será um valor aproximado para separar a imagem do fundo [12].

Contando com este limiar automático calculado, a imagem passa a ter mais evidenciado os dados em questão.

3.1.2.5 Seleção da Região da Palma da Mão

A palma da mão é digitalizada de maneira com que a palma da mão seja separada de todas as outras partes da mão e do fundo da imagem, através de uma máscara preta que é aplicada por cima do *scanner*, aonde existe apenas um retângulo de tamanho fixo, aonde a palma da mão é “encaixada” durante a digitalização.

Quando se inicia o tratamento desta imagem, basta utilizar um algoritmo que possa identificar uma grande variação entre a máscara (preta) e a palma da mão (cor clara).

3.1.2.6 Algoritmo de Sobel

O algoritmo de Sobel é implementado para obter um resultado da função de derivação da imagem, de forma que possa detectar as variações de contraste da imagem, ou seja, no caso deste projeto, as linhas da palma da mão. Aplicando-se uma máscara em forma de uma matriz bi-dimensional 3x3, com fatores de derivação sendo respectivamente o eixo de coordenadas x e eixo de coordenadas y, podemos detectar e salientar na imagem as linhas. Com base em um histograma direcional, gerado pelo cálculo da probabilidade direcional de cada pixel, podemos fazer uma limiarização nesta imagem, separando as linhas da palma da mão, com o resto da palma da mão [16].

A seguir está sendo representado um pseudocódigo do funcionamento do algoritmo de *sobel* [16].

```

//máscara do algoritmo de sobel.
GX[0][0] = -1; GX[0][1] = 0; GX[0][2] = 1;
GX[1][0] = -2; GX[1][1] = 0; GX[1][2] = 2;
GX[2][0] = -1; GX[2][1] = 0; GX[2][2] = 1;
GY[0][0] = 1; GY[0][1] = 2; GY[0][2] = 1;
GY[1][0] = 0; GY[1][1] = 0; GY[1][2] = 0;
GY[2][0] = -1; GY[2][1] = -2; GY[2][2] = -1;
Para Y de 1 até [Largura da Imagem-1]
    Para X de 1 até [Comprimento da Imagem-1]
        SomaX = 0;
        SomaY = 0;
        Se (Y=0 ou Y=[Largura da Imagem-1]) então
            Soma = 0;
            senão se (X=0 ou X=[Comprimento da Imagem-1]) então
                Soma = 0;
            Fim-se;
            senão
                Para I de -1 até 1
                    Para J de -1 até 1
                        SomaX = SomaX + ValorRPixel(x+i, y+j)* GX[I+1][J+1]);
                    Fim-para;
                Fim-para;
                Se (SomaX<0) então
                    SomaX=0;
                Senão
                    SomaX=mod(SomaX);
                Fim-se;
                Para I de -1 até 1
                    Para J de -1 até 1
                        SomaY = somaY+ ValorRPixel(x+i, y+j)* GY[I+1][J+1]);
                    Fim-para;
                Fim-para;
                Se (SomaY<0) então
                    SomaY=0;
                Senão
                    SomaY=mod(SomaY);
                Fim-se;
                Soma = abs(SomaX) + abs(SomaY);
            Fim-se;
            Pixel(x, y) = 255 - (Soma);
        Fim-para;
    Fim-para;

```

Depois de feito o cálculo de densidade direcional, é montado um histograma direcional, para podermos separar a imagem em regiões que possuam linhas com as maiores direções (as linhas de interesse neste estudo). Este método de separação em regiões que será utilizado irá filtrar a imagem de uma maneira que somente sobre na imagem as regiões que possuam determinadas linhas.

3.1.2.7 Geração do *Biocode*

Entende-se por biocode o código numérico utilizado para persistir as informações a respeito das características de uma determinada chave biométrica. A quantidade de dados que será persistida no biocode representará a quantidade de informações que realmente serão utilizadas na verificação dos dados [1].

O *biocode* que é calculado neste projeto é baseado em informações pertencentes a janelas da imagem, ou seja, logo após o processamento completo da imagem, esta será dividida em 16 partes iguais, de aonde serão extraídas a quantidade de pontos em cada uma destas partes, a soma das magnitudes da mesma e o ângulo resultante das linhas desta parte.

A medição de cada amostra da palma da mão pode apresentar uma variação com relação a uma outra medição anterior realizada, por isso, durante o cadastro da mão do usuário, será analisado o fator de variância do *biocode* para que este número também seja levado em consideração durante a verificação biométrica. O valor de número de pontos extraído da palma da mão varia de uma amostra para outra (considerando o mesmo usuário), portanto um fator de variação será calculado, com base nestas amostras iniciais.

O valor de variação nestes aspectos depende de usuário para usuário e por isso não pode ser definido um valor máximo e somente durante a aquisição das imagens estes valores estão sendo calculados e armazenados no banco de dados.

3.1.3 Sistema de cadastro

O sistema de cadastro de usuários é responsável por armazenar novos usuários para acessar determinado local ou informação. Basicamente, este módulo é executado por um administrador do sistema, reconhecido através de senha alfa-numérica e *login* próprio do programa. O banco de dados que está sendo utilizado no sistema é o Paradox do próprio compilador utilizado para implementar este projeto, o Borland C++ Builder, devido a sua facilidade e rapidez de montar as tabelas e os registros de dados e a fácil interação com o sistema e com o compilador.

3.1.3.1 Cadastro de Usuários

O cadastro de usuários é uma parte relativamente simples que consiste no cadastramento das informações pessoais do usuário em si, assim como os dados necessários para que seja possível fazer a identificação deste usuário com base nas imagens.

Toda a parte de pré-processamento e geração de *biocode* anteriormente descrita é utilizada nesta fase de cadastro de usuário, pois além dos dados de endereço, telefone, etc., está sendo armazenado junto o seu *biocode*. Por se tratar de um sistema que demonstre muita variação na aquisição das medidas, serão amostradas cinco (05) vezes a mão do usuário, para que se chegue em um valor médio entre as amostras selecionadas dentre as cinco adquiridas (média de apenas três). Ao contrário de um modelo de verificação de um sistema biométrico comportamental (voz, digitação) aonde em um mesmo indivíduo pode haver uma grande variação do padrão analisado, neste projeto aonde a chave biométrica se encontra nas linhas da palma da mão, ou seja, em uma característica física aonde há pouca variação durante toda a vida, pode ser extraída fazendo-se esta quantidade de amostragens.

As três amostras servem para eliminar qualquer tentativa de uso incorreto do sistema, de alguma interferência de algum agente externo ao sistema, ou mesmo ainda por alguma diferença causada pelos métodos de pré-tratamento da imagem.

A Figura 13 demonstra como é o sistema de cadastro de novo usuário.

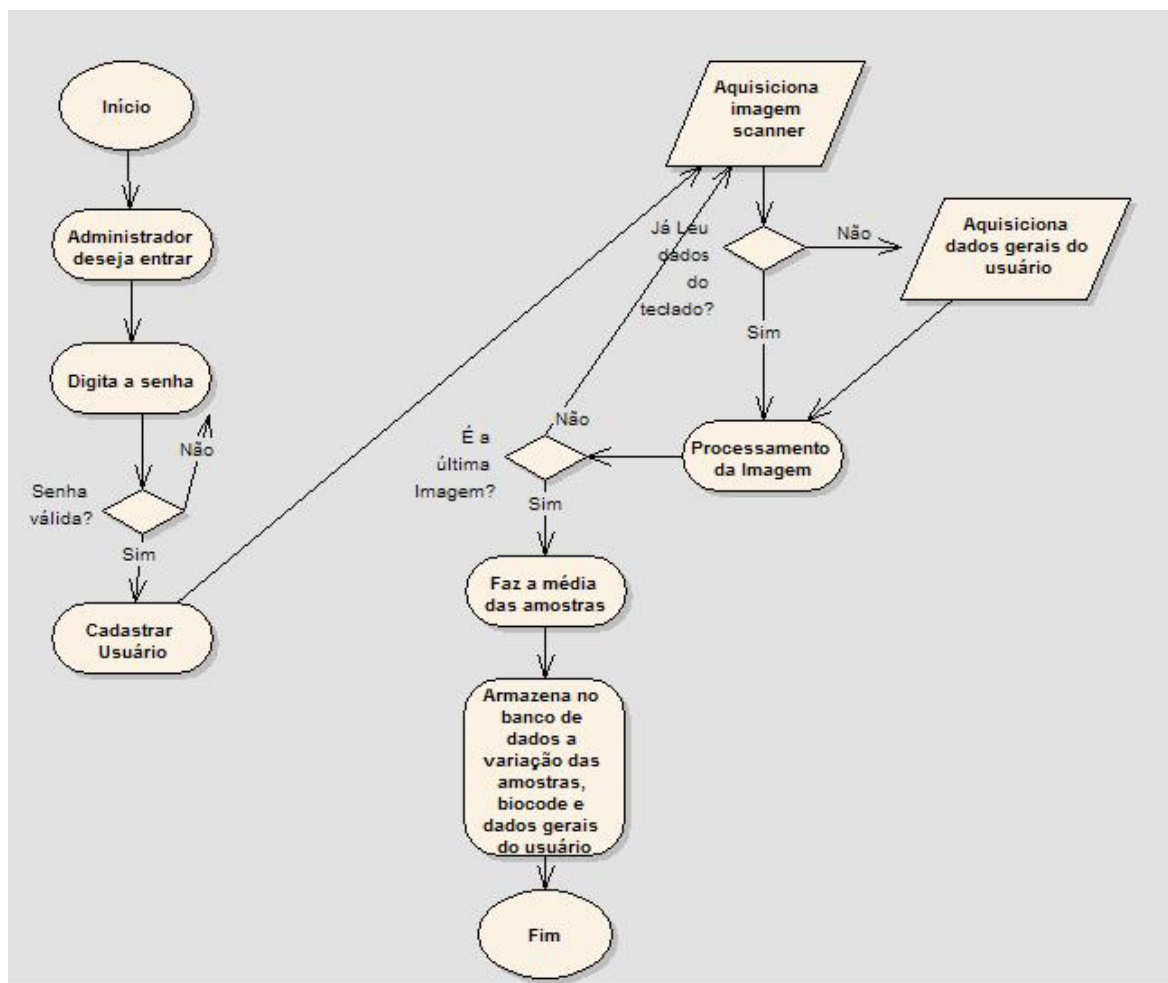


Figura 13 - Fluxograma detalhado do cadastro de usuários.

3.1.3.2 Verificação do Acesso

O sistema faz a verificação do acesso do usuário primeiramente lendo os dados identificadores que estão dentro do *smart card* do usuário e enviando para o módulo principal (módulo de verificação biométrica). Então o sistema faz uma busca no banco de dados através deste código identificador, que no banco de dados será a chave primária da tabela de usuários.

Em seguida, o sistema autoriza o usuário a inserir a sua mão no *scanner* para que seja amostrada uma imagem de sua palma da mão. Com a imagem em memória, todo o processamento de imagens acima descrito será realizado e então, deve-se comparar o código biométrico previamente armazenado no cadastro com o mesmo código amostrado neste momento. É uma comparação simples, aonde o programa apenas verifica se os campos são iguais (dentro do

desvio padrão previamente calculado no cadastro) e caso não forem, verificar se o módulo da diferença entre os valores amostrado e armazenado for menor ou igual ao valor máximo da variação permitida no sistema. Caso um dos testes anteriores citados resultar em verdadeiro, pronto, o sistema libera o acesso a este determinado usuário, mas caso a senha biométrica não combine com a senha armazenada, mesmo com esta tolerância (variação) então o sistema irá acusar que houve algum tipo de problema na identificação do usuário, neste caso, não liberando o acesso deste.

A Figura 14 apresenta um fluxograma de verificação de acesso, desde a inserção do *smart card* até a liberação/rejeição do usuário.

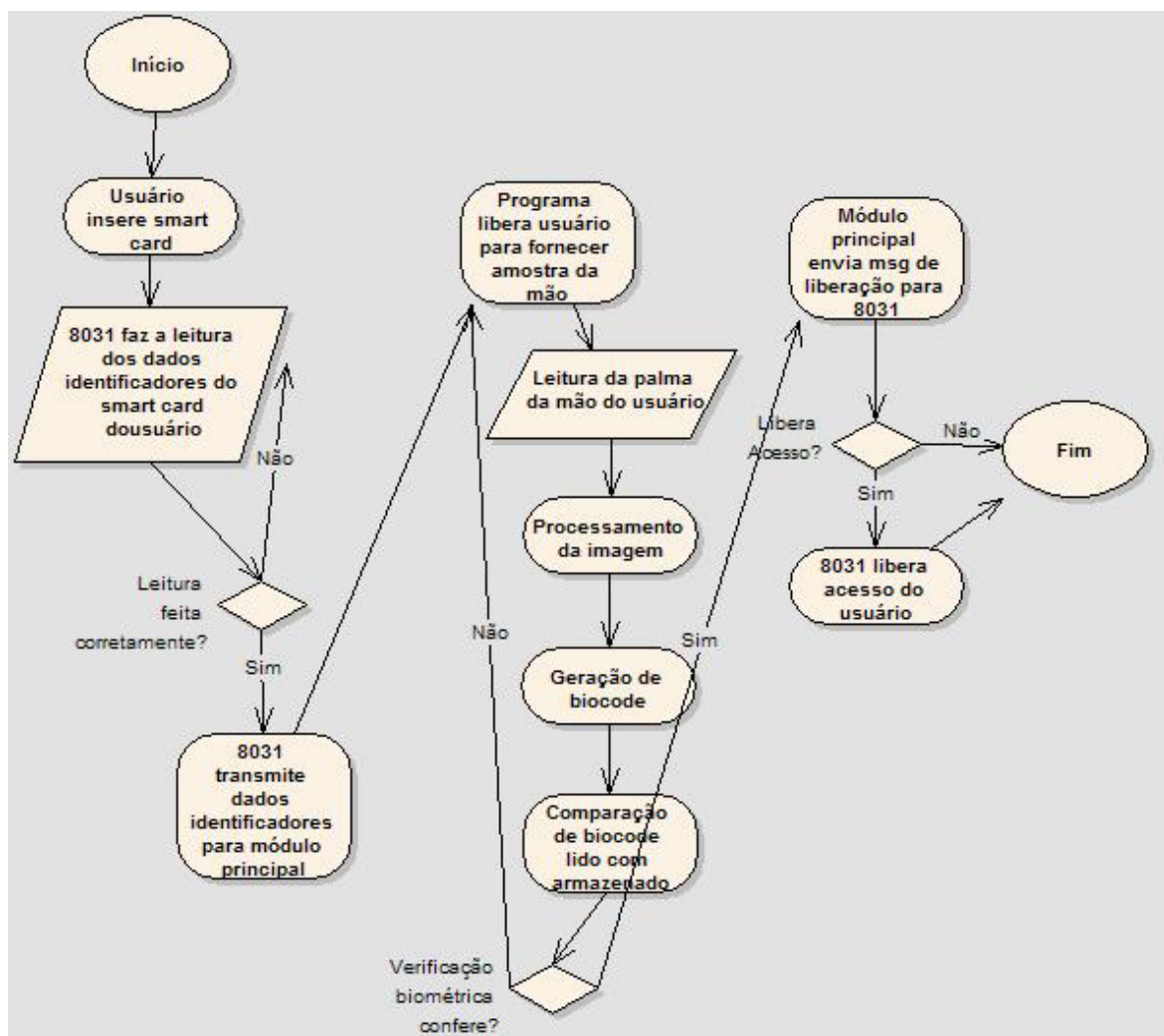


Figura 14 - Fluxograma de Verificação do Acesso.

Este módulo do sistema é o mais utilizado pelos usuários do sistema, pois é exatamente este o objetivo do projeto, limitar o acesso apenas a pessoas cadastradas, e para fazer a conferência dos usuários, o método de verificação de acesso é chamado. É importante que o usuário saiba que para que seu acesso seja bem sucedido, primeiramente deve ser feita a identificação via *smart card* e posteriormente, após a liberação do sistema, a verificação via leitura da palma da mão.

3.1.3.3 Banco de Dados

O banco de dados que será utilizado neste projeto é o *Paradox*, que vem junto com o pacote de instalação do Borland C++ Builder.

A escolha de utilização deste banco é baseada na forma de utilização e a facilidade de implementação neste banco de dados. Para se criar uma nova tabela com este banco de dados, basta executar uma instrução SQL (*Structured Query Language*) de criar a tabela, juntamente com os campos pré-definidos e a chave primária estabelecida.

As definições da tabela seguem o seguinte padrão: o nome da tabela que armazenará todas as características do usuário se chamará '*tb/IPF*'. Todos os campos são criados com a característica de serem alfanuméricos e apenas o campo com o nome *smart card* deverá possuir número exclusivo e não equivalente a nenhum dos outros valores deste campo (chave primária).

Os demais campos podem ter valores repetidos, pois fazem parte dos cálculos resultantes do processamento da imagem como um todo.

A linguagem de manipulação de dados que é utilizada no projeto é a linguagem SQL a partir do próprio compilador que está sendo utilizado no projeto.

Para armazenar apenas um tipo de dados, que são as informações pessoais de cada usuário, é necessária apenas a criação de uma tabela para a solução do problema como um todo. Os principais campos que estão sendo criados são: *smartcard*, endereço, cidade, telefone, celular, *biocode*, desvio padrão do *biocode*, ângulo resultante de cada um dos 1/16 da imagem e desvio

padrão destes ângulos também. Os dados cadastrais comuns são criados com tamanho de 50 *bytes* cada e os campos de valores calculados no programa são criados com tamanho de 10 *bytes* cada. A Figura 15 exemplifica a estrutura do banco de dados criada.

tbIPF : Tabela

smartcard	Nome	Endereço	Cidade	Telefone	Celular	Email	Biocode	DesvioPadraoB	ang1
1	Tiago	rua1	Curitiba		8412-4253	tmmoreira@ya	235689	2154	32

Registro: 1 de 1

Figura 15 - Estrutura do Banco de Dados

3.2 Projeto de Hardware

Aqui serão descritas todas as implementações que foram realizadas em nível de *hardware* do sistema, suas funcionalidades e sua maneira de usar, para o correto funcionamento do sistema. Isto tudo inclui o interfaceamento entre o *smart card* e o micro-controlador 8031.

3.2.1 Leitura e Escrita no *smart card*

Esta parte do projeto visa descrever o funcionamento do sistema de leitura e escrita dos dados no *smart card* do usuário. A partir do protocolo de comunicação I2C, aonde a comunicação é feita por 2 fios (SDA e SCL – respectivamente *serial data*, *serial clock*) aonde estes estão sendo projetados para serem utilizados respectivamente nos pinos P1.1 e P1.0 do micro-controlador 8031, conforme explicados anteriormente na fase de especificação do projeto.

A Figura 16 abaixo relacionada exemplifica, segundo o fabricante [16], o controle necessário do micro-controlador em cima dos sinais SCL e SDA no momento da escrita do sinal no *smart card*.

O tempo do ciclo de escrita é representado na Figura 16 pela sigla t_{wr} que representa o tempo para uma seqüência válida de escrita ou leitura no *smart card*, representando um *stop bit* da seqüência de manipulação dos dados da memória do *smart card*.

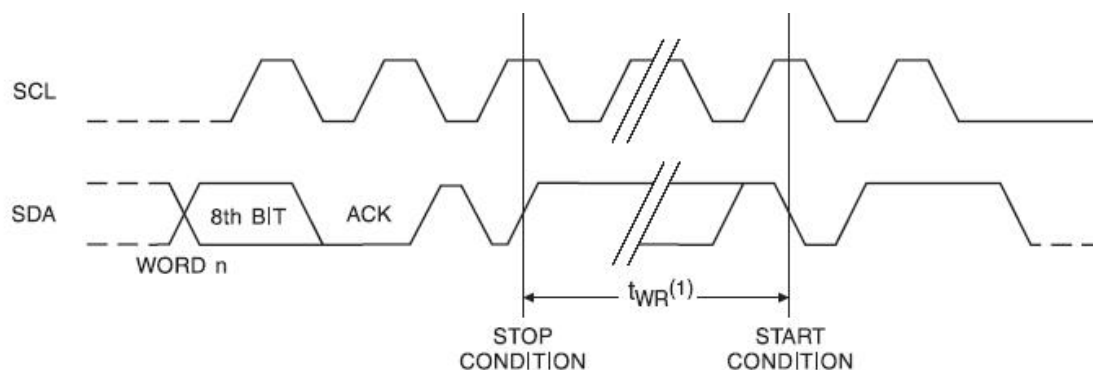


Figura 16 - Sinais de controle para a escrita no smart card.

A forma pela qual a estabilização dos dados no barramento de comunicação do cartão é feita é demonstrada na ilustração da Figura 17 abaixo [16], juntamente com a forma de se indicar o bit de início e o bit de fim do protocolo [16].

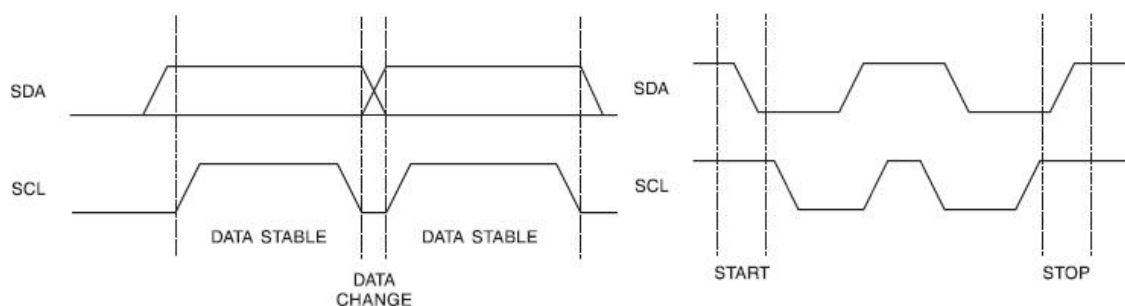


Figura 17 - Estabilidade dos dados no barramento e condições de Start e Stop bits.

Toda a interpretação que é feita a partir destas duas exemplificações acima nos fornece a visão de como é feita a comunicação com este dispositivo que está sendo utilizado (*smart card*). A forma de início de comunicação com o dispositivo se dá da seguinte maneira: são enviados 8 bits para o dispositivo, sendo que os quatro (04) primeiros são uma seqüência alternada entre zero e um (0 e 1), sendo que o mais significativo deles é um (1). Os próximos 3 bits devem ser obrigatoriamente setados com o valor zero (0). O último bit do cabeçalho serve para escolher entre leitura e escrita da memória do *smart card*, sendo que o bit 1 representa um ciclo de leitura da memória e o bit 0 representa uma operação de

escrita na memória. Depois de todo o envio do cabeçalho, a EEPROM irá retornar o valor zero (0) para o controlador, caso isso não ocorra, a memória voltará para o modo de espera de comandos. Com isso já temos o início da comunicação com a EEPROM do *smart card*. A Figura 18 demonstra a seqüência inicial destes 8 bits [16].

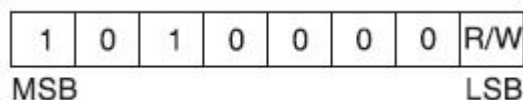


Figura 18 - Protocolo de início de comunicação entre *smart card* e micro-controlador.

Detalhando um pouco mais as operações de gravação de dados no cartão, este pode ser gravado de duas maneiras, sendo que a maneira pela qual ele vai ser implementada vai ser a forma *byte write* (escrita em bytes). Esta forma de gravação funciona escrevendo-se o cabeçalho inicial de comunicação com o cartão (EEPROM) juntamente com a posição de memória que se deseja inserir o *byte*. Logo após o envio do cabeçalho, o cartão irá responder ao chamado com um sinal de *acknowledgment* (reconhecimento). Este sinal indicará que a EEPROM estará pronta para iniciar a comunicação. O próximo passo é colocar a posição da memória que será usada para se fazer a gravação do dado e em seguida, enviar o dado para a memória, terminando a comunicação com uma condição de parada (*stop bit*). Qualquer outra operação que não seja a escrita deste *byte* não será realizada, enquanto estiver neste processo de gravação na memória. Antes ainda de enviar o sinal de fim de operação (*stop bit*) a memória enviará ao micro-controlador um *acknowledgment* (confirmação de operação bem sucedida).

A Figura 19 demonstra como vai ser implementada a escrita de dados na memória do *smart card* [16].

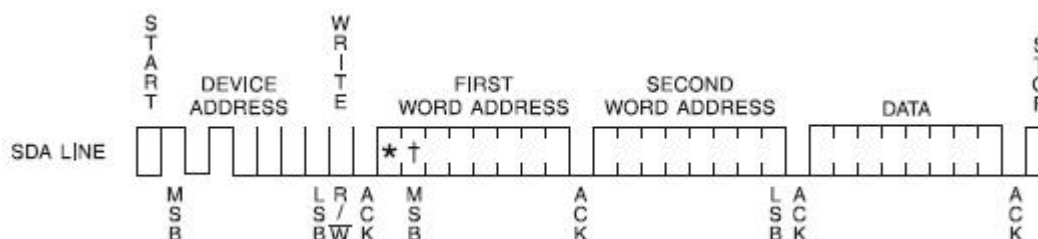


Figura 19 - Protocolo para a gravação de dados no *smart card*.

Dentre as três (03) maneiras possíveis de se fazer a leitura dos dados do cartão, a que é usada no projeto é a leitura a partir de um endereço inicial escolhido (*random read*). Operações de leitura são iniciadas da mesma maneira que as operações de gravação de informações no *smart card*. Para que se inicie a leitura, é necessário que se faça a transmissão do primeiro byte de identificação de dispositivo, com o bit de seleção de leitura/escrita setado em 0 para fazer a seleção do endereço de memória a ser utilizado e depois, mais um sinal de *start bit* será enviado e mais uma vez o cabeçalho de *device address*, mas agora com o bit de escolha entre leitura e gravação é levado a nível lógico alto (1), que representa a leitura de dados. O próximo byte irá representar os dados que estão sendo lidos do cartão a partir do endereço inicial da memória. Estes bytes estão sendo armazenados previamente em um registrador do micro-controlador e em seguida, ao fim de cada byte, é transmitido serialmente.

A Figura 20 abaixo está fazendo a ilustração de como é feito esta montagem do protocolo [16].

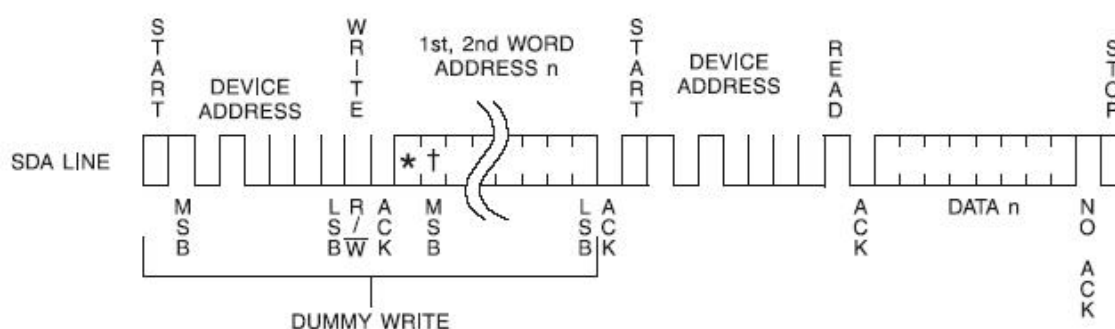


Figura 20 - Protocolo para a leitura de dados no *smart card*.

3.2.2 Programa do 8031

Todos estes procedimentos de leitura e de gravação de informações no *smart card* são partes de rotinas pré-estabelecidas que estão dentro da memória de programa do micro-controlador 8031, mais especificamente, gravadas na EPROM do circuito mencionado na especificação deste projeto.

O programa que será necessário está sub-dividido em 3 módulos:

- Módulo de comunicação com o *smart card*;
- Módulo de comunicação com o PC (serial);
- Módulo de liberação/rejeição do acesso.

Para simplificar um pouco o modo de abrangência destes módulos, eles serão detalhados em etapas.

O módulo de comunicação com o *smart card* terá dentro de si a implementação do protocolo de comunicação I2C, que forma toda aquele protocolo vistos já no item de leitura e escrita de informações no *smart card*. A Figura 21 demonstra um fluxograma da montagem e da comunicação entre os dispositivos.

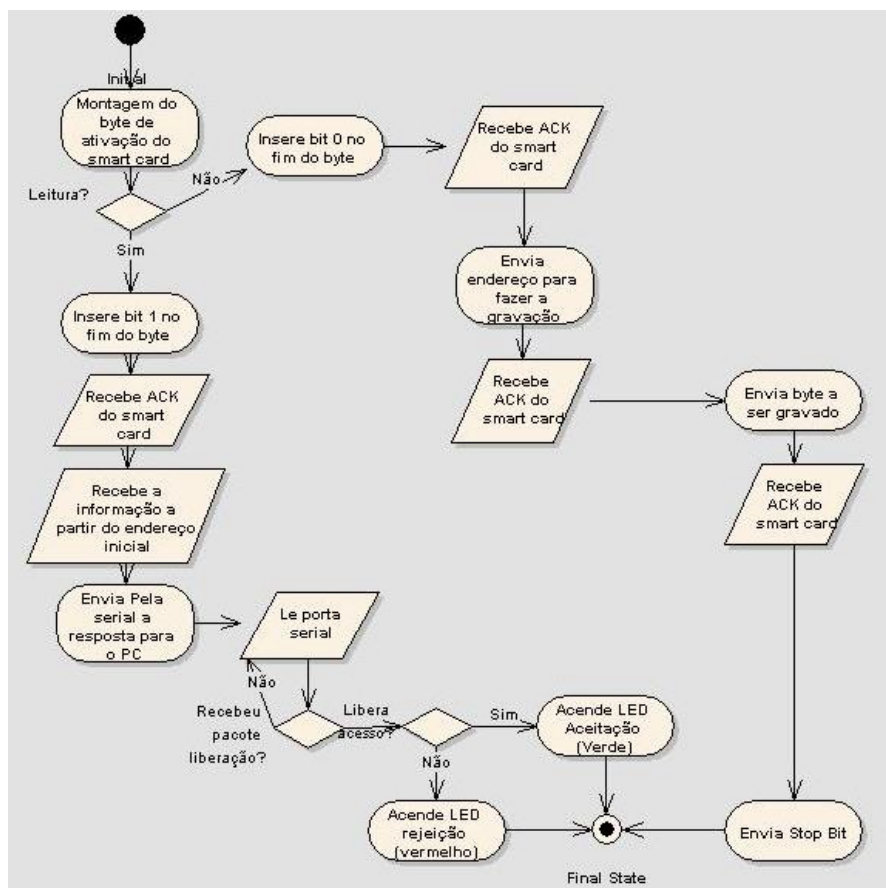


Figura 21 - Fluxograma da comunicação entre 8031 e smart card.

O módulo de comunicação com a serial foi feito logo após a comunicação com o *smart card*. Depois de feita a leitura das informações necessárias para a identificação do usuário, estes mesmos dados são enviados para o programa principal para que seja feita a identificação.

Logo após o envio destes dados para o programa principal, o 8031 ficará aguardando um sinal que indicará que o PC vai enviar a resposta da liberação ou da rejeição do usuário. Quando ele receber este pacote, ele faz um teste interno com um valor previamente escolhido para cada uma das operações. Por exemplo, para rejeitar, ele recebe a letra 'E' do PC e para liberar o acesso ele recebe a letra 'D' do PC. Depois de fazer este teste, ele envia a resposta para o respectivo LED. O LED de rejeição de acesso está no esquemático sendo ligado no pino P1.2 do 8031 e o LED de liberação do acesso está sendo conectado no pino P1.3 do 8031.

Caso o usuário seja rejeitado e ele seja um usuário cadastrado, fica a critério dele fazer uma nova verificação ou não. Para que isto seja possível, ele deverá retirar o seu *smart card* do conector e inseri-lo novamente e repetir todo o procedimento já descrito.

3.2.3 Elaboração do circuito micro-controlador

Algumas características para o funcionamento do circuito são descritas a seguir como uma maneira de certificação e testes de circuito, e também para um melhor entendimento de toda a ligação entre os componentes.

O circuito integrado 8031 é a versão sem ROM interna do 8051. O mesmo é muito utilizado em fase de desenvolvimento ou quando se quer produzir em pequenas quantidades. No esquemático descrito na fase de especificação, podemos ver que estamos utilizando 8kB de RAM externa, além dos 256 bytes de RAM interna. Além disso, temos um total de 8Kbytes de memória ROM.

Para fazermos a leitura na ROM externa, o pino PSEN é levado a nível baixo, para fazer uma leitura na RAM externa, o pino de leitura RD (*read*) é levado a nível baixo e, para fazer uma escrita na RAM externa, o pino de escrita WR (*write*) é levado a nível baixo.

A Figura 22 abaixo mostra como funciona o *reset* no circuito.

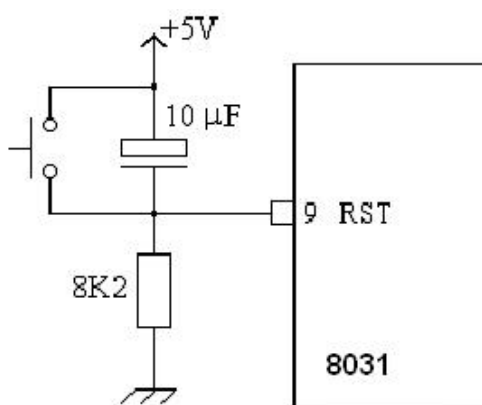


Figura 22 - Reset no micro-controlador.

O *reset* é conseguido mantendo-se nível alto no pino 9 (RST) por pelo menos dois ciclos de máquina (24 pulsos de clock). Por se tratar de um micro-controlador RISC (*reduced instruction set computing*), ele possui um ciclo de máquina equivalente a 12 ciclos de clock, isso significa que ele possui poucas instruções internas, mas que elas fazem todo o processamento desejado pelo controlador.

O efeito do *reset* nos registros do micro-controlador afeta de forma diferenciada cada parte deles, fazendo com que a maioria dos valores dos registradores e portas para comunicação tenham valores equivalentes a 0x00h ou 0xFFh.

Como pode ser visto no esquemático da especificação do projeto, além das memórias faz-se necessário a utilização de um outro CI (no caso o 74LS373) para a estabilidade de dados e endereços. Ocorre da seguinte forma: no primeiro ciclo de máquina, o micro-controlador 8031 coloca nos pinos AD0 a AD7 o byte menos significativo do endereço externo e leva o pino ALE (*Address Latch Enable* - Habilitador da Trava de Endereços) a nível alto, de modo que o 74LS373 (oito Latches tipo D) coloque em suas saídas essa informação, e logo em seguida passa este pino para nível baixo, para que esse byte fique retido no 74LS373. Após isso, os pinos AD0 a AD7 estarão livres para o transporte dos dados, já que o barramento é comum para alguns componentes.

3.3 Cronograma

O cronograma que está sendo seguido para o desenvolvimento do projeto está sendo mostrado no ANEXO 1, aonde mostra todas as suas fases, desde implementação, apresentações, datas de defesas e de entregas de documentos.

3.4 Protótipo de telas

Nesta parte serão mostradas todas as formas com que o programa foi implementado, especificamente nesta seção, será mais abordado o desenvolvimento envolvido no protótipo das telas de interface com o usuário. Vários aspectos foram considerados na elaboração destes protótipos, entre eles,

a facilidade com que o usuário tem de interagir com o programa e suas funcionalidades.

3.4.1 Tela de *Login*

Para fazer o cadastro de usuários no sistema é necessário que isso seja feito por um administrador do sistema. Para identificar quem é administrador, este deve entrar com a senha e nome de usuário contidos no programa para fazer acesso a seção de cadastro de novos registros (usuários). O protótipo desta tela está mostrado abaixo na Figura 23.



Figura 23 - Tela de *Login*

3.4.2 Tela Principal

A tela principal do programa tem por característica ter todos os caminhos de todos os métodos de tratamento da imagem do *software*. É nela que temos as funções de filtro, funções de leitura de imagem, funções de trocar de usuário. A Figura 24 exemplifica a forma desta tela principal. Opções de menu mostram algumas seções como, por exemplo, salvar imagem e entrar na seção de cadastro (passando pelo login antes).

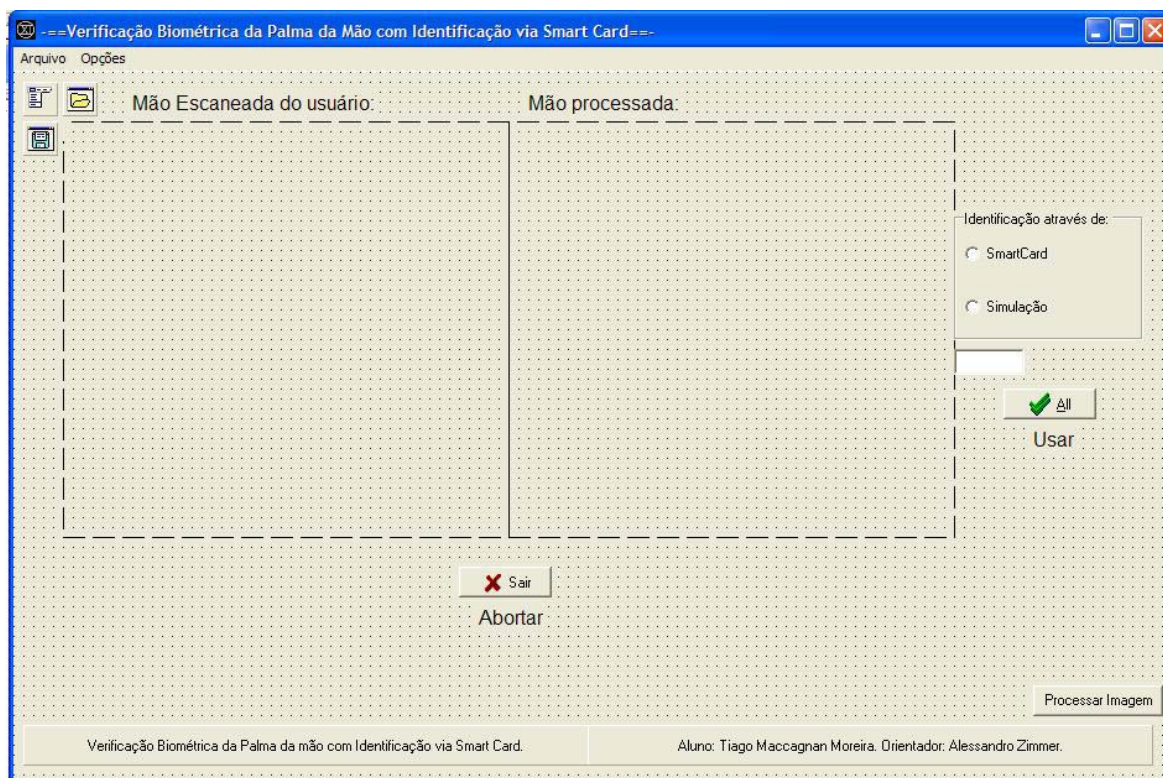
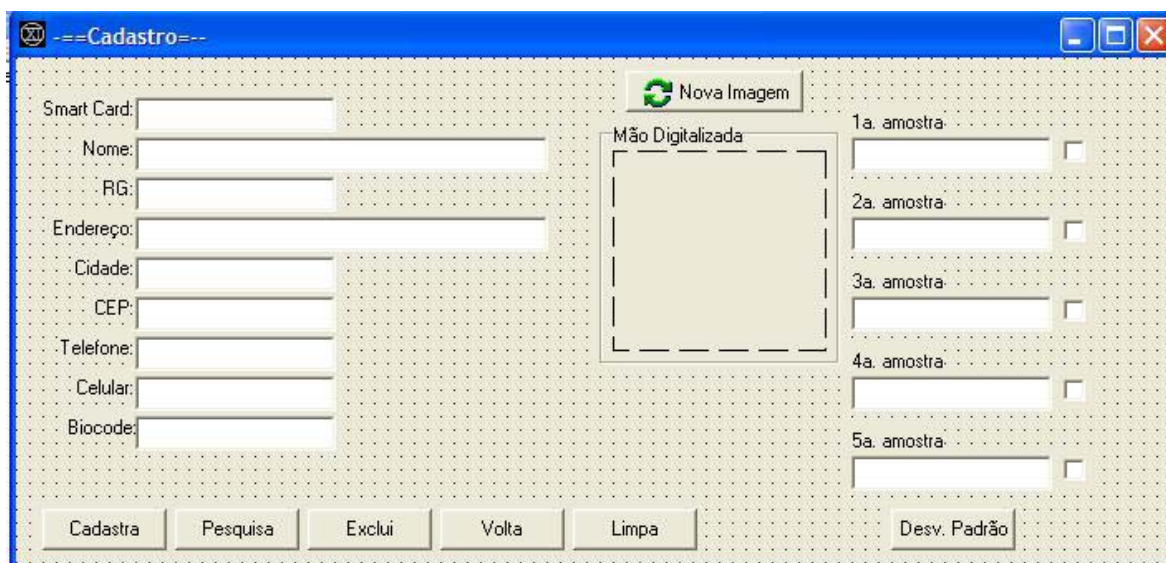


Figura 24 - Protótipo da tela principal.

3.4.3 Tela de Cadastro de Usuários

A tela de cadastro de usuários é operada pelo administrador do sistema, onde os campos deverão ser preenchidos através do teclado e a imagem da palma da mão irá aparecer para demonstrar eventuais falhas graves em algum ponto crítico da imagem. Os botões que ficam ao lado da caixa da imagem da mão tomam toda a ação desta tela, uma vez que eles são os responsáveis pelo início do cadastro, pesquisa e remoção de dados do banco. Logo após cinco (05) amostras da mão do indivíduo, o administrador poderá apertar a tecla “Volta” para voltar para o modo normal de operação do programa. Quando uma nova imagem tiver que ser digitalizada, o botão “Nova Imagem” deverá se apertado. Neste caso, o programa recalculará o *biocode* da palma da mão em questão. O botão “Desv. Padrão” serve para calcular o desvio padrão de apenas três (03) amostras selecionadas pelo administrador do sistema. Estes valores serão gravados no banco de dados, para futura verificação. Lembrando que enquanto está sendo realizado o cadastro de usuários, nenhum outro membro cadastrado poderá fazer uso do programa, pois o programa não está sendo preparado para atender várias

atividades concorrentes. Abaixo, na Figura 25 está sendo exemplificado esta tela de Cadastro de Usuários.



Protótipo da tela de cadastro de usuários. A interface possui uma barra de título azul com o texto "===Cadastro===" e ícones de minimizar, maximizar e fechar. O formulário é dividido em seções:

- Smart Card:** Campo de texto.
- Nome:** Campo de texto.
- RG:** Campo de texto.
- Endereço:** Campo de texto.
- Cidade:** Campo de texto.
- CEP:** Campo de texto.
- Telefone:** Campo de texto.
- Celular:** Campo de texto.
- Biocode:** Campo de texto.
- Nova Imagem:** Botão com ícone de seta circular verde.
- Mão Digitalizada:** Área reservada para a imagem da mão digitalizada, delimitada por uma caixa tracejada.
- amostras:** Seção com cinco campos rotulados "1a. amostra:", "2a. amostra:", "3a. amostra:", "4a. amostra:" e "5a. amostra:", cada um com um campo de texto e uma caixa de seleção.

Na base da tela, há uma barra com os seguintes botões: "Cadastra", "Pesquisa", "Exclui", "Volta", "Limpa" e "Desv. Padrão".

Figura 25 - Protótipo da tela de cadastro de usuários.

4 IMPLEMENTAÇÃO

Neste ponto, serão discutidos todos os aspectos referentes à implementação do que foi especificado na fase de projeto. A implementação do projeto ocorreu de maneira modular, onde, em uma primeira instância, o módulo de *hardware* foi concluído e, posteriormente, o módulo de *software*.

4.1 Implementação do módulo de Hardware

A implementação do módulo de *hardware* pode ser sub-dividida em 3 grandes partes, sendo elas:

- Módulo de comunicação serial com o PC;
- Módulo do circuito micro controlador (integrado com *smart card*);
- Módulo do protocolo de comunicação com o *smart card*.

O ANEXO 3 mostra o código comentado das rotinas em assembly para o micro controlador 80C31.

4.1.1 Módulo de comunicação serial com o PC

Este módulo é composto por rotinas de configuração do micro controlador, envolvendo seus *timers* e seus registradores de configuração para o funcionamento adequado.

O registrador de configuração da porta serial está sendo configurado com o valor 0x50h, que representa no micro controlador o uso da serial em modo 1 e a taxa de transmissão variável, utilizando uma UART de 8 bits. O bit que habilita a recepção serial também está sendo setado, ele é o REN (scon.4). Como a taxa de transmissão, neste modo de configuração, é variável e determinada pela taxa de *overflow* do *timer 1*, foi implementado uma taxa de comunicação de 4800bps, com os seguintes valores para o *timer 1*: 0xF4h tanto para a parte alta quanto para a parte baixa do registrador (é um registrador de 16 bits, por isso ele é dividido em 2 partes, alta e baixa). Lembrando que os valores setados no registrador *timer 1*

são valores configurados com *auto-reload*, ou seja, quando o *timer* chega ao máximo de sua representação, ele é recarregado para o valor estipulado da taxa de transmissão da serial, ou seja, neste projeto com o valor 0xF4h para a parte alta e para a parte baixa.

As interrupções geradas pela porta serial no micro controlador são geradas por um OR lógico entre os *flags* RI (recepção serial) e TI (transmissão serial). Os *flags* RI e TI não são resetados pela CPU no atendimento das interrupções e devem ser resetados pela rotina de atendimento a esta interrupção. É justamente isto que está sendo feito depois da recepção e do envio de dados pela serial, aonde estes dois *flags* recebem 0 novamente.

O registrador tmod (modo do *timer*) está sendo configurado com o valor 0x21h, que representa o *timer 0* e o *timer 1* em modo 2, aonde os dois *timers* operam com todos os seus 16 bits e com recarga automática (*auto-reload*).

O registrador IE (*Interrupt Enable*) está sendo configurado com o valor 0x90h, aonde todas as interrupções estão sendo ativadas, inclusive a serial. As interrupções externas não serão utilizadas neste projeto, portanto elas serão desconsideradas a todo momento.

Ao final de todas as configurações, é setado o início da contagem do *timer 1*, para que ele possa fazer a taxa de comunicação serial a todo instante, independente de outras configurações e rotinas.

4.1.2 Módulo do circuito micro controlador (integrado com o *smart card*)

Para a solução deste módulo, foi utilizado o *kit* do micro controlador 8051 da Unicenp e implementado com base neste *kit*, todo o resto da parte de *hardware*. A utilização da porta P1 do 80C31 está sendo utilizada para fazer toda a comunicação com o *smart card* e a liberação do acesso. Os pinos de comunicação com o cartão são os mesmos conforme projetados anteriormente, sendo eles P1.0 e P1.1, respectivamente SCL e SDA do *smart card*. Os pinos

P1.2 e P1.3 estão sendo usados para fazer a liberação e a rejeição do acesso, respectivamente.

O modo serial de comunicação com o PC foi implementado com uma velocidade de 4800bps (*baud-rate*), com o *timer 1*, responsável pela taxa de transmissão da serial, em modo 1 e com todas as outras funcionalidades com o mesmo nível de prioridade. O *timer 1* está sendo implementado com a característica de *auto-reload*, ou seja, quando a contagem chega ao limite de representação dos 16 bits, o valor inicial do *timer* é restaurado, do valor de configuração inicial, que para esta velocidade é 0xF4F4h. A serial está sendo configurada no modo 2, para que seja configurável a taxa de transmissão e controle maior da serial. Logo após o recebimento do caractere 'A' (0x41h), o circuito micro controlado faz o início da leitura do *smart card*.

O *clock* gerado pelo micro controlador para a troca das informações com o cartão é totalmente controlado pela necessidade de troca das informações com o cartão, ou seja, à medida em que o protocolo está avançando ou os bits estão sendo aguardados, o *clock* é controlado de maneira com a qual a sua frequência não é uma frequência fixa, pois em determinados instantes, como por exemplo no momento de aguardar um sinal de ACK gerado pelo cartão, o micro controlador aguarda até o instante em que o sinal for captado pelo barramento da porta P1.1, independentemente quanto tempo este sinal demore para chegar (em micro segundos).

O endereço de memória do cartão onde o identificador do usuário está sendo gravado são os 3 primeiros bytes a partir do endereço 0x0000h, mas poderia ser outro, pois existe uma combinação de 14 bits disponíveis para o endereçamento dos seus 16Kbytes de memória disponíveis, ou seja, qualquer endereço entre 0x00h e 0x3FFFh poderia ser utilizado. Os 13 bytes do biocode estão sendo armazenados a partir do endereço 0x0003h da memória do cartão.

A identificação do usuário está sendo feita, lendo-se estes bytes nas posições de memória acima especificadas, e para este projeto, não serão possíveis mais do que 999 usuários diferentes, apenas por uma questão de projeto e aplicação.

O contato elétrico feito no cartão está sendo feito com um soquete de encaixe específico para este tipo de cartão e a sua montagem está sendo feita em uma placa padrão com o tamanho do soquete de encaixe. Alguns centímetros de fios fazem a alimentação do cartão e a comunicação com o micro controlador.

4.1.3 Módulo do protocolo de comunicação com o *smart card*

O *smart card* está acoplado na placa padrão, sua alimentação e comunicação está sendo feita por alguns fios comuns de cobre. Seu protocolo de comunicação está sendo implementado com base na especificação do projeto de *hardware* anteriormente descrita.

A leitura do cartão começa a partir do momento em que o usuário ativa o comando de iniciar a leitura no computador, então o PC envia serialmente um comando para iniciar a leitura do cartão, sendo este comando a letra 'A' representado por 0x41h. Neste instante a rotina de leitura do cartão será executada e ao final da leitura do identificador do cartão, na sua respectiva posição de memória, o micro controlador envia serialmente ao PC os *bytes* lidos do cartão.

O protocolo de comunicação implementado é o mesmo descrito na fase de projeto do módulo.

A gravação de dados no cartão está sendo feita, enviando-se serialmente o identificador 'B', representado por 0x42h e logo após, o micro-controlador aguarda do PC o byte a ser gravado. No final da gravação, o micro-controlador retorna ao PC um byte sinalizador, representado pela letra 'G' (gravado).

4.2 Implementação do módulo de Software

A modularização das etapas do projeto está fazendo com que todos os métodos pudessem ser testados quase que independentes um do outro. A integração e a seqüência de chamada dos métodos é definida pela funcionalidade de cada método

4.2.1 Aquisição da Imagem

A aquisição da imagem está sendo feita normalmente, como descrito em fase de projeto, através de um *scanner* de mesa padrão, com uma qualidade de 100dpi. A imagem está sendo digitalizada normalmente colorida em formato *True Color*, ou seja, com 24 bits de representação. A imagem original está sendo mantida em escala de cinza, para que pudessem ser feitas algumas comparações com a imagem original, depois de todo o processamento.

A imagem está sendo carregada em memória e a partir de então, o programa abre a imagem e inicia o tratamento. Uma imagem, para ser considerada válida, precisa estar com toda palma da mão encostada no vidro do *scanner* com uma leve pressão no vidro, para ser uma imagem de fácil tratamento.

Uma máscara de cartolina de dimensões 30cm x 22cm, com um quadrado no centro, cuja medida lateral é de 6,5cm, está sendo aplicada ao projeto, para facilitar a digitalização e seleção da região de interesse. A inclusão desta máscara torna o pré-processamento e a seleção da região de interesse bem mais rápida para a abordagem deste projeto, tornando uma saída viável em busca da solução.

O uso da cartolina está sendo utilizado, em virtude do não sucesso na implementação de um método confiável de segmentação da imagem da palma da mão.

A escolha da cartolina de cor preta também elimina qualquer variação de luminosidade que possa ser encontrada durante a digitalização da imagem no *scanner*, o que torna a qualidade da imagem obtida muito maior.

4.2.2 Processamento da Imagem

O processamento da imagem se inicia no momento em que a imagem está sendo carregada para a visualização na tela, pois a imagem é digitalizada com

cores e todo o processamento é baseado apenas nos tons de cinza. Portanto, no momento já da leitura, o endereço de memória que contém a imagem original já é iniciado com o tom de cinza respectivo para cada *pixel*. Todo o processamento da imagem se baseia em identificar na imagem a parte correspondente à palma da mão e a extração das características (análise de padrões).

A identificação da região de interesse, a partir da imagem adquirida, é feita levando-se em consideração que toda a palma da mão está contida dentro do retângulo da máscara de *scanner*. A partir de então, a imagem é convertida para a escala de cinza, conforme o projeto especifica.

O método de Sobel é invocado, depois da etapa anterior, para fazer a extração das linhas da palma da mão na imagem. A Figura 26 mostra o resultado da extração direta do algoritmo de Sobel, e em seguida, o resultado sendo selecionado pelo limiar automático de Otsu.

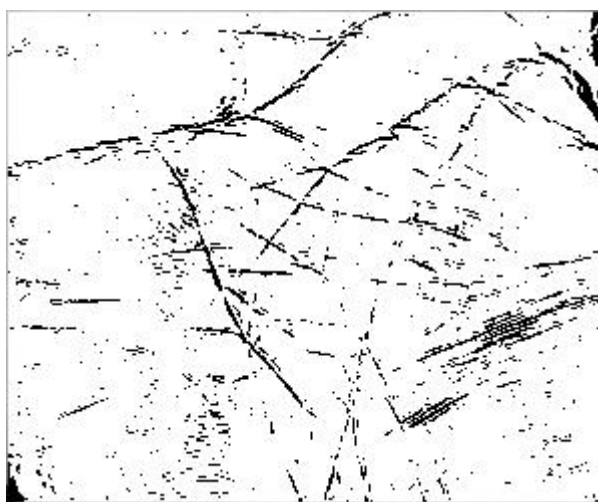


Figura 26 - Palma da mão com as linhas evidenciadas.

O histograma direcional é criado com base nos operadores de gradiente característicos da imagem, aonde, a partir da análise vetorial, sabe-se que o vetor gradiente aponta na direção de mudanças mais rápida em um determinado ponto da imagem. Em detecção de bordas, a magnitude deste vetor é uma quantidade importante, geralmente chamada apenas de gradiente, aonde o seu valor é

resultado da raiz quadrada da soma do quadrado do gradiente de cada direção (x e y).

Com o histograma direcional em mãos e o algoritmo de Otsu feito neste histograma, tem-se uma imagem com as linhas da palma da mão e ainda alguns pontos de alta frequência resultantes na imagem. Neste ponto agora, para eliminar estes ruídos, um filtro gaussiano será aplicado para retirar estes pontos.

Um detalhe que está acontecendo no momento do processamento da imagem, é com relação ao limiar automático de Otsu. A imagem está ficando muito poluída com o valor encontrado automaticamente pelo algoritmo, conforme mostra a Figura 27, a imagem da esquerda é a imagem que possui um incremento de 13 unidades no valor encontrado pelo algoritmo e a imagem da direita é a imagem que está resultando normalmente, com o uso do limiar automático de Otsu.

O valor de incremento aqui implementado como 13, veio de alguns testes realizados com uma base de imagens já adquiridas. Valores maiores do que este, fazem com que a imagem resultante perca informações importantes, enquanto que valores menores do que este, fazem com que a imagem fique com muitos ruídos desnecessários para o processamento da imagem e que não são considerados informações importantes para o processamento das linhas da palma da mão.

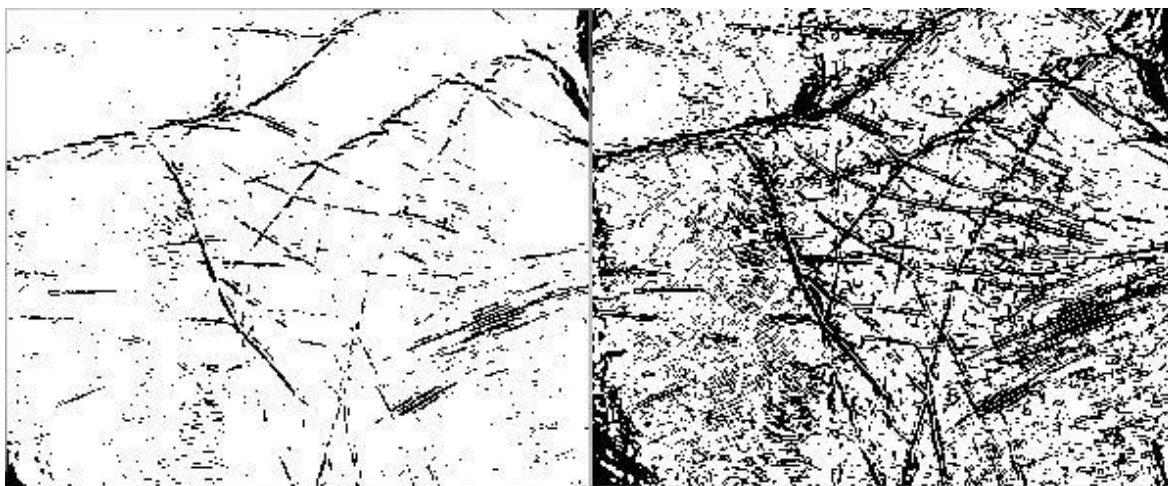


Figura 27 - Resultados do processamento da imagem.

4.2.3 Geração do *Biocode*

Conforme descrito em fase de projeto, o método de verificação da palma da mão está sendo implementado depois de todo o processamento da imagem, quando a imagem possui apenas as linhas evidenciadas.

Para a determinação deste código, a seguinte estratégia está sendo usada: a imagem está sendo dividida em 16 partes iguais, como mostra a Figura 28 e, em cada uma destas partes, está sendo contado o número de pontos pretos, a magnitude resultante de todas as partes das linhas presentes nesta parte e o ângulo resultante desta região.

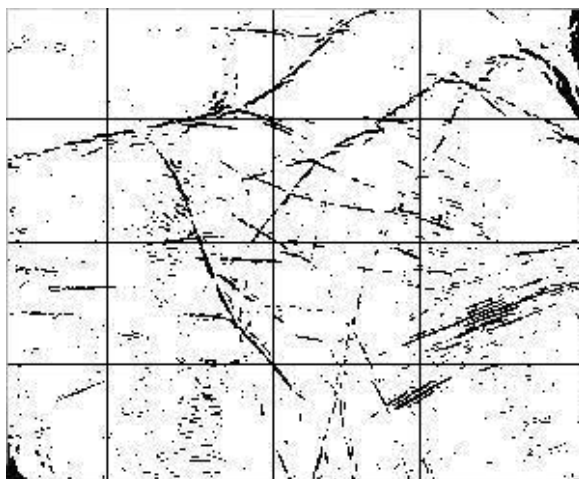


Figura 28 - Palma da mão dividida em 16 regiões.

Com estes 16 valores de pontos, uma média ponderada dará como resultado um número real para a quantidade de pontos nesta janela (junto, o cálculo do desvio padrão para cada uma das 16 partes, para estas três medidas). Com os 16 valores das magnitudes, uma outra média ponderada será calculada e o resultado destas três partes (pontos, magnitudes e ângulos resultantes de cada parte destas 16 regiões) será ainda calculado um valor seguindo a seguinte fórmula: raiz quadrada da soma do quadrado dos pontos, da magnitude e do ângulo.

$$\text{Biocode} = \sqrt{(\text{soma pontos})^2 + (\text{soma magnitudes})^2 + (\text{soma ângulos})^2}$$

Este valor real será armazenado no banco de dados do sistema e servirá de verificador (contra-senha) para o código do *smart card* lido anteriormente.

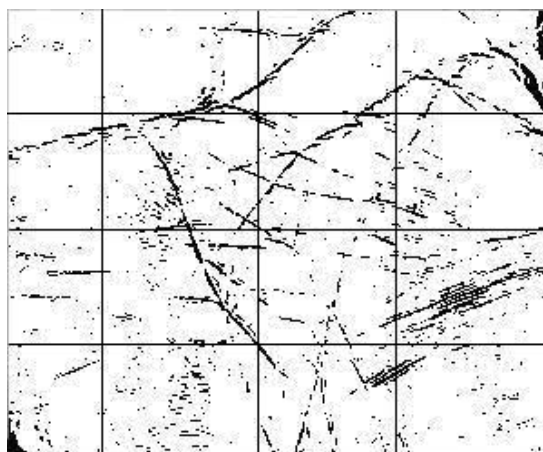


Figura 29 - Palma da mão separada em 16 regiões.

Todo o cálculo do biocode leva em consideração estas 16 regiões que a mão é dividida, pois para cada uma delas, um desvio padrão é calculado no momento do cadastro e no momento da verificação de acesso, se a mão a ser verificada não estiver dentro do padrão cadastrado, certamente, o acesso será rejeitado.

4.3 Testes e Validação do Projeto

Após concluída toda a fase de implementação o projeto foi submetido ao protocolo de teste e validação definido anteriormente. Ele foi capaz de atender a todos os requisitos definidos anteriormente.

4.3.1 Testes do Módulo de cadastro de usuário

O módulo de cadastro se mostrou funcional, pois foram realizados diversos cadastros, e o módulo cumpriu com a obrigação de manter a informação armazenada. Durante os testes do *software* foram realizados diversos testes de cadastro, pesquisa e exclusão de usuário. O módulo atendeu às expectativas mencionadas anteriormente.

4.3.2 Testes do Módulo de Leitura do Smart Card

Todos os testes realizados neste módulo foram feitos utilizando somente três smart cards. Em cada um dos cartões, foi gravado um identificador do sistema e um biocode. Sempre, em todas as ocasiões de leitura do cartão, o processo de identificação foi realizado com sucesso. Como o esperado, o cartão manteve gravado todos os dados, até que o cartão fosse novamente gravado e atualizado os seus campos.

4.3.3 Testes do Módulo de Reconhecimento Biométrico

O módulo de reconhecimento biométrico se apresentou estável e funcional, e para todas as imagens submetidas ao processamento, o método retornou sempre um biocode. Um fator que atrapalhou muito durante a validação deste módulo foi a diferença de pressão da palma da mão aplicada sob o vidro do

scanner de mesa. A solução adotada para este problema foi o de se utilizar uma espuma pequena (densidade 33) para que o administrador do sistema pudesse aplicar a mesma força em cima da mão do usuário (no momento do cadastro) e o usuário era responsável por aplicar esta mesma força no momento da utilização do sistema. O sistema conta com uma base de dados de 40 usuários cadastrados e todas as imagens foram processadas para gerar os valores de falsa aceitação.

4.3.4 Testes do Sistema Completo

O sistema como um todo se apresentou de maneira muito segura, com a utilização simultânea de todos os módulos. O cadastro foi realizado com sucesso, utilizando os módulos de reconhecimento biométrico, o de gravação do smart card, e todos os procedimentos que envolvem o cadastro. O processamento do biocode se mostrou eficaz durante toda a fase de testes do algoritmo.

Os testes tiveram início, adquirindo cinco (05) amostras da palma da mão do usuário e manualmente, escolhidas apenas três (03) para que o desvio padrão e a média de todos os biocodes fossem calculados para serem gravados no banco de dados. Em seguida, um número identificador (ID) foi atribuído a este usuário e gravados no smart card estes dados (identificador e biocode), e os outros dados, junto com estes foram instanciados no banco de dados do sistema. Neste ponto, o módulo de cadastro encerrou para este usuário. Em seguida, para efeito de validação, o usuário cadastrado inseriu seu cartão e o sistema fez a sua busca no banco de dados e em seguida este usuário inseriu sua palma da mão (a mesma cadastrada) e o sistema calculou seu biocode e fez a comparação se o biocode do cartão está dentro do desvio padrão gravado para este usuário. Em caso afirmativo, o circuito com o micro-controlador libera o acesso, caso contrário, rejeita o acesso.

5 RESULTADOS E CONCLUSÕES

O desenvolvimento deste projeto permitiu demonstrar que é possível a utilização de um método computacional que faça a extração das características biométricas da palma da mão para se fazer um sistema baseado em identificação do usuário para controlar acessos e/ou informações restritos.

Com a base de dados coletada de 40 usuários, sendo três amostras válidas por usuário, a taxa de Falsa Aceitação obtida foi de 5,59%. É uma taxa relativamente alta, se formos comparar com a situação ideal que é de zero. Esta é uma grande preocupação no uso deste sistema como sendo um sistema de segurança, pois esta taxa permite que pessoas diferentes daquelas que são identificadas possam ser aceitas pelo sistema. Alguns outros componentes biométricos podem ser incorporados junto a esta análise da palma da mão, para aumentar a confiabilidade do sistema, com mais parâmetros na geração do *biocode*, como por exemplo um algoritmo de cálculo da pressão exercida sobre a superfície do vidro, para que isso não seja mais um fator extremamente variável na entrada do sistema.

Ainda com a mesma base, a taxa de Falsa Rejeição encontrada no sistema não pode ser calculada, pois com base em apenas cinco amostras coletadas e utilizando sempre as três mais próximas para se fazer o cálculo do desvio padrão, não se pode fazer um cálculo de Falsa Rejeição. Seria ideal, uma base de dados de aproximadamente 20 mãos por usuário, para podermos ter um índice comparativo.

O *smart card* utilizado no sistema cumpriu com perfeição a missão de guardar os dados de identificação e o *biocode* do usuário, porém, com toda a sua capacidade, poderiam ser gravados juntos os dados de desvio padrão, ângulo resultante de cada quadrante, enfim, todos os dados que são necessários para o cálculo do *biocode*.

Futuramente, melhorias poderiam ser implementadas, como por exemplo, utilizar alguma outra característica biométrica, como a geometria da palma da mão junto para diminuir a porcentagem de erros no sistema como um todo, e

ainda mudar o sistema de aquisição de imagens, usando uma câmera digital, pois esta não tem problemas com relação à diferença de pressão exercida sobre o vidro do scanner e apresenta uma velocidade de captura da imagem bem maior, comparada com o scanner de mesa comum utilizado neste projeto e também implementar uma criptografia de dados armazenados no *smart card*, pois no sistema atual isso não foi necessário e para um uso mais contínuo, isso é extremamente interessante.

Os LEDs indicativos de acesso liberado ou rejeitado poderiam ser substituídos no futuro por alguma chave eletrônica ou por uma catraca eletrônica, acionada por um relé, de modo que o acesso fosse realmente liberado.

A seguir, na Figura 30 está o gráfico de dispersão das pessoas e seus respectivos *biocodes*. Através deste gráfico, foi calculado a taxa de Falsa Aceitação, mencionada anteriormente, para a base de dados atual.

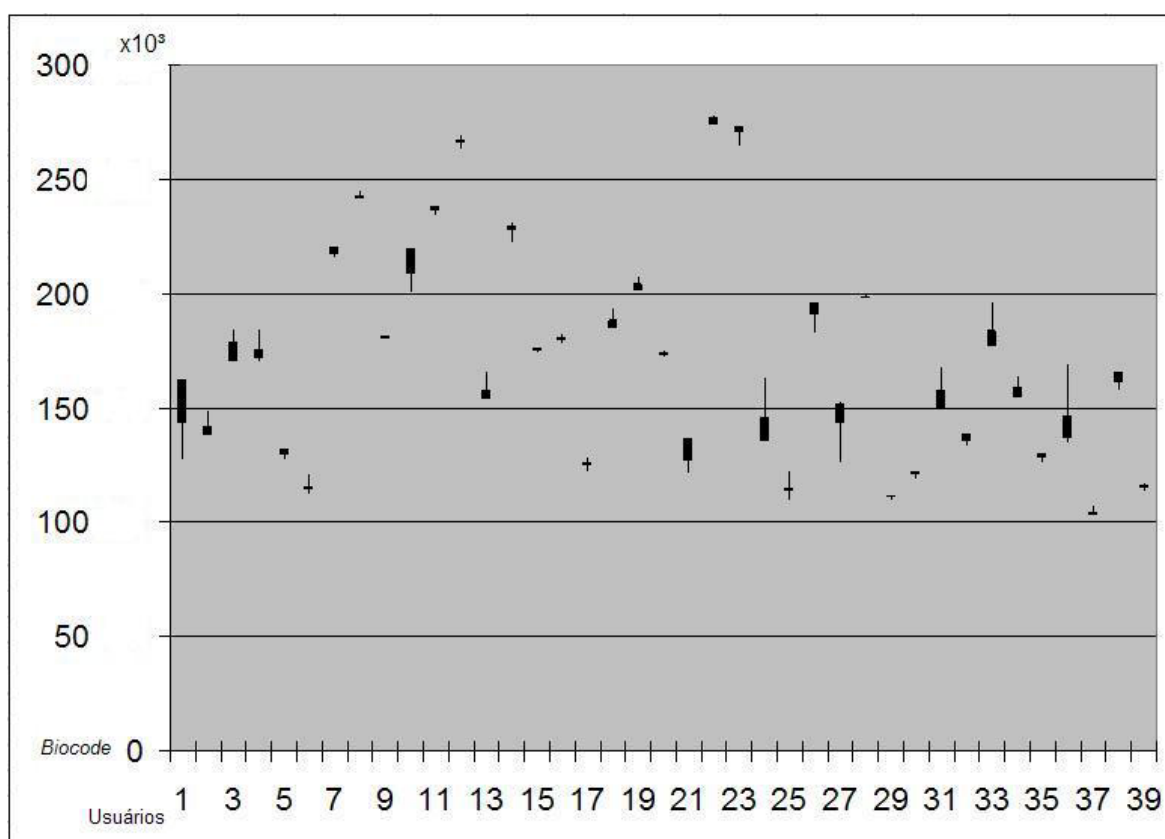


Figura 30 - Gráfico de Dispersão dos usuários cadastrados.

Pelo gráfico, analisamos os valores de *biocode* (eixo 'y') dos usuários (eixo 'x') e a conclusão que podemos tomar é de que apesar de alguns valores de *biocode* de usuários poderem estar em uma mesma posição, o resultado do sistema como um todo foi satisfatório, pois o sistema conseguiu separar os usuários pelo *biocode*, conforme vemos no gráfico. Supondo que o usuário número 1 esteja querendo entrar no sistema, seu *biocode* resultante não vai ser confundido com o usuário 3, pois eles estão em regiões distintas de *biocode* resultante.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Boreki, G., Sistema de controle de acesso por *iButton* com verificação biométrica da geometria da mão, Núcleo de Ciências Exatas e Tecnológicas, UnicenP – Centro Universitário Positivo – 2003.
- [2] Espinosa, M., Um sistema biométrico de identificação pessoal via internet com ênfase em assinaturas estáticas, Departamento de Comunicações, Unicamp, 2000.
- [3] Jain, A.; Hong L.; Pankanti, S., "Biometrics: promising frontiers for emerging identification market", *Communication of ACM*, pp. 91 - 98, February, 2000.
- [4] *International Biometric Group*. Disponível em www.ibgweb.com - 2004.
- [5] Castleman, K., "Digital Image Processing", Prentice-Hall, Inc., 1996.
- [6] *NIST Computer Security Handbook. Identification and Authentication*. Disponível em www.bilbo.edu/security/isl/hk_i&a.html.
- [7] Matos, R., Mecanismos de Autenticação de Usuários com utilização de dados biométricos – 1998.
Disponível em www.inf.ufrgs.br/pos/SemanaAcademica/Semana98/renata.html.
- [8] Dougman, J., *Biometric Decision Landscapes*, University of Cambridge.
- [9] Gonzalez, R., Woods, Richard., Processamento de imagens digitais, Editora Edgard Blücher LTDA, 2000.
- [10] Hajn, V., *Less common dermatoglyphic patterns on the palm in moravian population*, Department of Zoology and Anthropology Faculty of Natural Science, Palacký University, 2002.
- [11] Reed, T., Meier, R., Department of Medical Genetics, Indiana University School of Medicine and Department of Anthropology, 1990.
- [12] Otsu N., "A Threshold Selection Method from Gray-level Histograms", *IEEE Transactions on Systems, Man and Cybernetics*, v. SMC 9, no 1, pp.62-66, 1979.
- [13] M. Worring, A. W. M. Smeulders, *The Accuracy and Precision of Curvature Estimation Methods*, University of Amsterdam, 1992.
- [14] L. O' Gorman, *An Analysis of Feature Detectability from Curvature Estimation*, AT&T Bell Laboratories, Murray Hill, New Jersey – 1988.
- [15] Green, B., *Edge Detection Tutorial* – 2002.
Disponível em www.pages.drexel.edu/~weg22/can_tut.html.

[16] *AT24C128SC, 2-wire Serial EEPROM Smart Card Modules.*

Disponível em www.atmel.fi/dyn/resources/prod_documents/doc1661.pdf.

[17] *Twain – Standard for image acquisition devices.* Disponível em www.twain.org
Driver versão 2.0 de julho de 2004.