



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

FRANTCHESCO RODOLFO NOGUEIRA

CAPTURA DE SINAL BIOMÉTRICO UTILIZANDO ARDUINO

ASSIS

2011

FRANTCHESCO RODOLFO NOGUEIRA

CAPTURA DE SINAL BIOMÉTRICO UTILIZANDO ARDUINO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito de Curso de Bacharelado em Ciência da Computação, analisado pela seguinte comissão examinadora:

Orientador: Profa. Dra. Marisa Atsuko Nitto

Área de Concentração: Informática

ASSIS

2011

FICHA CATALOGRÁFICA

NOGUEIRA, Frantchescos R.
CAPTURA DE SINAL BIOMÉTRICO UTILIZANDO ARDUINO / Frantchescos Rodolfo Nogueira. Fundação Educacional do Município de Assis – FEMA – Assis, 2011.
60p.

Orientadora: Profa. Dra. Marisa Atsuko Nitto.
Trabalho de Conclusão de Curso - Instituto Municipal de Ensino Superior de Assis – IMESA.

1. Biometria. 2. Matriz de Led. 3. Arduino.

CDD: 001.6
Biblioteca FEMA



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

FRANTCHESCO RODOLFO NOGUEIRA

CAPTURA DE SINAL BIOMÉTRICO UTILIZANDO ARDUINO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Bacharelado em Ciência da Computação, analisado pela seguinte comissão examinadora:

Orientador: Profa. Dra. Marisa Atsuko Nitto

Analisador: Prof. Dr. Luiz Carlos Begosso

Assis

2011

Dedico este trabalho aos meus pais, Marcio Ricardo Nogueira e Maria Rita Nogueira, por sempre me apoiarem, incentivarem e valorizarem meus esforços.

AGRADECIMENTOS

A Deus, por tudo, muito obrigado!;

A minha orientadora, Profa. Dra. Marisa Atsuko Nitto, pela orientação e ajuda segura durante este trabalho. Muito obrigado por todo apoio, paciência, disponibilidade, esforço e conhecimento prestados à minha carreira profissional;

A Professora Regina Fumie Eto pelo apoio e por ceder alguns materiais utilizados;

Ao meu patrão Eduardo Vitorino pelo apoio e amizade em todos os momentos. E também pela compreensão de me ausentar alguns dias do trabalho;

Ao meu irmão Franklin Tiego Nogueira por estar sempre junto nesta jornada;

A minha namorada Kéren Leal pelo apoio e companheirismo, mesmo durante os períodos de ausência;

A Fundação Educacional do Município de Assis - FEMA, por tornar-me capacitado para desenvolver este trabalho.

Aos professores do Curso de Ciência da Computação da FEMA, pelos valiosos ensinamentos durante a minha passagem pela instituição.

Aos amigos de curso e família, pelo apoio, amizade e demonstração de companheirismo.

A todos que direta ou indiretamente, contribuíram para a realização deste trabalho.

RESUMO

Neste projeto será desenvolvido um circuito para capturar um sinal biométrico utilizando o arduíno. O arduíno utilizado é baseado em um micro controlador (Atmega). O sinal biométrico capturado será o de uma impressão digital. Para isso, as impressões digitais de cada indivíduo são capturadas por um dispositivo comercial adquirido em loja e armazenadas em um banco de dados como imagens *bitmap*. Essas imagens serão binarizadas utilizando as técnicas e métodos de processamento de imagens para que possam ser marcados os pontos com características relevantes. Os métodos utilizados para a marcação desses pontos são os tradicionalmente utilizados em biometria. A comunicação do dispositivo com o arduíno será feita por um programa desenvolvido em linguagem C, onde os sinais dos pontos marcados na imagem da impressão digital serão enviados para uma matriz de *led* 8x8. Os pontos marcados da imagem correspondem aos *leds* acessos.

Palavras-chaves: Biometria, Matriz de *Led*, Arduino.

ABSTRACT

This project will develop a circuit to capture a biometric signal using Arduino. The Arduino is used based on a microcontroller (Atmega). The signal to be captured is a biometric fingerprint. For this, the fingerprints of each individual are captured by a commercial device purchased in store and stored in a database as bitmap images. These images are binarized using the techniques and methods of processing images so that the points can be marked with relevant characteristics. The methods used for marking these points are traditionally used in biometrics. The communication device with Arduino will be made by a program developed in C language, where signs of marked points on the fingerprint image will be sent to dot matrix LED 8x8. The marked points correspond to the LEDs image access.

Keywords: Biometrics, LED Dot Matrix, Arduino.

LISTA DE ILUSTRAÇÕES.

Figura 1 - Leitor de impressão digital.	13
Figura 2 - Detalhes peculiares de uma impressão digital. (KALKULAR, 2007).	21
Figura 3 - Imagens marcadas após extração de minúcias (COSTA, 2001).	22
Figura 4 - Sensor Capacitivo. (HARRIS, 2002).	25
Figura 5 - Modelos de Arduino	26
Figura 6 - Diagramas de Blocos	28
Figura 7 - Arquitetura Básica de um Arduino (SILVA, 2010).	29
Figura 8 - Dispositivos Importantes do Arduino (COSTA, 2010).	32
Figura 9 - Pinagem Atmega168.	33
Figura 10 - Diagrama de Blocos	33
Figura 11 - Ambiente Gráfico do Arduino (ENDEL B, 2008).	35
Figura 12 - Arquitetura de Programação no Arduino.	37
Figura 13 - Modelagem do problema.	44
Figura 14 - Modelo Lógico do Banco de Dados.	45
Figura 15 - Arquitetura do Sistema de Cadastro.	46
Figura 16 - Diagrama de Caso de Uso.	46
Figura 17 - Diagrama de Classe.	47
Figura 18 - Diagrama de Sequência.	47
Figura 19 - Interface Principal.	48
Figura 20 - Interface Para Cadastrar.	49
Figura 21 - Interface de tratamento da imagem.	50
Figura 22 - Interface de Pessoas cadastradas	51
Figura 23 - Arduino Modelo Program-ME (GLOBALCODE).	53
Figura 24 - Leitor zk-4000	53
Figura 25 - Diagrama do Circuito (FUTUREC, 2011).	54
Figura 26 - Integração do Arduino com Placa de Controle de Sinal(adaptado de McROBERTS, 2010).	55
Figura 27- - Diagrama de Comunicação do Computador com o Dispositivo.	56
Figura 28 - Dispositivo Montado e Funcionando.	56

LISTA DE TABELAS

Tabela 1 - Aplicações do reconhecimento de padrões (JAIN et al, 2000).....	18
Tabela 2- Principais Diferenças Entre os Arduínos (SILVEIRA, 2001).....	28
Tabela 3 - Comandos do Toolbar e Suas Funções (adaptado de ENDEL C, 2008).....	35
Tabela 4 - Comandos do Menu Sketch e suas funções (adaptado de ENDEL C, 2008).....	36
Tabela 5 - Comandos do Menu Tools e Suas Funções (adaptado de ENDEL C, 2008)	37

Sumário

1. INTRODUÇÃO	13
1.1 - OBJETIVOS	15
1.2 - JUSTIFICATIVAS	16
1.3 - MOTIVAÇÕES	16
1.4 - ESTRUTURA DO TRABALHO.....	16
2. FUNDAMENTAÇÃO TEÓRICA BÁSICA	17
2.1 – RECONHECIMENTO DE PADRÕES	17
2.2 - BIOMETRIA.....	19
2.3 – LEITORES BIOMÉTRICOS DE IMPRESSÃO DIGITAL	22
2.3.1 - Ultrasônico	23
2.3.2 - Ótico.....	23
2.3.3 - Capacitivo.....	24
2.4 – ARDUÍNO	26
2.4.1 – Hardware	27
2.4.2 – Características do arduíno.....	28
2.4.3 – Ambiente de Programação para o Arduíno	34
2.4.3.1 - IDE Arduino	34
2.4.3.2 – Linguagem de programação para arduino.....	38
2.4.3.2.1 - Linguagem de referência.....	38
2.4.3.2.2 - Funções	39
2.4.3.2.3 – Bibliotecas	39
2.4.3.2.3.1 - Comunicação (redes e protocolos)	40
2.4.3.2.3.2 - Sensoriamento.....	41
2.4.3.2.3.3 - Geração de Frequência e de Áudio	41
2.4.3.2.3.4 - Temporização	41
2.4.3.2.3.5 - Utilidades.....	41
3. DESENVOLVIMENTO DO PROJETO	43
3.1 – DESCRIÇÃO DO PROBLEMA.....	43
3.2 – MODELAGEM DO PROBLEMA.....	43
3.3 – IMPLEMENTAÇÃO	44
3.3.1 – Criação do banco de dados.....	45

3.3.2 – Sistema de Cadastramento	45
3.3.2.1 – Especificação	46
3.3.2.2 – Implementação do Sistema de Cadastro	48
3.3.3 – Desenvolvimento do Dispositivo de Captura de Sinal Biométrico.....	51
3.3.3.1 – Modelagem do dispositivo	51
3.3.3.1.1 – Material Utilizado	52
3.3.3.1.2 – Integração da matriz com o arduíno	55
4. CONCLUSÃO	57
5. REFERÊNCIAS.....	58

1. INTRODUÇÃO

As biometrias de digitais são amplamente conhecidas como um método preciso de identificação e verificação biométrica. A impressão digital é composta por vários sulcos, que em sua formação apresentam diferenças chamadas de pontos de minúcias, ou seja, aquela parte em que os sulcos se dividem (vales) ou onde terminam abruptamente (terminação) (FARIA, 2005). Cada um desses pontos tem características únicas, que podem ser medidas. Para isso é necessário um dispositivo, o sensor biométrico ou leitor de digital, capaz de efetuar o registro da impressão digital, com um elevado grau de precisão, por forma, a registrar os traços, minúcias, que definem a impressão digital. Atualmente, a biometria é usada na identificação criminal, controle de acesso, etc. A Figura 1 mostra um leitor de impressão digital.



Figura 1 - Leitor de impressão digital.

Um sistema de leitura de impressões digitais tem duas funções básicas:

- precisa obter uma imagem do dedo;
- precisa determinar se o padrão de sulcos nessa amostra confere com o padrão de sulcos em imagens previamente capturadas.

Existem muitas maneiras de se obter a imagem do dedo de alguém. Os métodos mais comuns atualmente são a leitura ótica e a leitura capacitiva. Os dois métodos resultam no mesmo tipo de imagem, mas chegam a ela de maneiras completamente diferentes. O *software* do leitor utiliza algoritmos altamente complexos para

reconhecer e analisar essas minúcias. Os sistemas que buscam evidências físicas, como leitores de impressões digitais, possuem certas vantagens sobre os outros sistemas (MAZI e DAL PINO JR, 2009), como:

- atributos físicos são muito mais difíceis de falsificar do que carteiras de identidade;
- não se pode “chutar” um padrão de impressões digitais como pode fazer com uma senha;
- não pode perder as próprias digitais, a voz ou a íris como pode acontecer com um cartão de acesso;
- não pode esquecer das impressões digitais como pode acontecer com uma senha.

Atualmente, os leitores de digitais e sistemas biométricos são uma excelente forma de identificação. No futuro, eles provavelmente irão se tornar uma parte integral da vida diária da maioria das pessoas, assim como as chaves, os cartões de banco e as senhas são hoje.

Neste projeto foi desenvolvido um dispositivo de captura de sinal biométrico utilizando arduíno, que é baseado em um micro controlador (Atmega), e dessa forma é logicamente programável, ou seja, é possível a criação de programas, utilizando uma linguagem própria baseada em C, C++ ou JAVA, que, quando implementada faz com que o hardware execute certas ações (FONSECA e BEPPU, 2010). O Arduino faz parte do conceito de *hardware* e *software* livre e está aberto para uso e contribuição de toda sociedade. O conceito Arduino surgiu na Itália em 2005, com o objetivo de criar um dispositivo para controlar projetos e protótipos construídos de uma forma menos dispendiosa do que outros sistemas disponíveis no mercado.

O grande diferencial desta ferramenta é que ela é desenvolvida e aperfeiçoada por uma comunidade que divulgam os seus projetos e seus códigos de aplicação, pois a concepção dela é *open-source*, ou seja, qualquer pessoa com conhecimento de programação pode modificar e ampliar de acordo com a necessidade, visando sempre à melhoria dos produtos que possam ser criados aplicando o Arduino (FONSECA e BEPPU, 2010).

Na verdade, o arduíno é um *kit* de desenvolvimento capaz de interpretar variáveis no ambiente e transformá-las em sinal elétrico correspondente, através de sensores

ligados aos seus terminais de entrada, e atuar o controle ou acionamento de algum outro elemento eletroeletrônico conectado ao terminal de saída.

Ele apresenta um ambiente multiplataforma, que pode ser executado em Windows, Macintosh e Linux. Ela é baseada na programação de processamento do IDE, um ambiente ao desenvolvimento fácil de ser utilizada pelos artistas e *designers*. A programação pode ser feita através de um cabo USB, e não uma porta serial. Esse recurso é útil, porque muitos computadores modernos não têm portas seriais (FERNANDES e LOPES, 2011).

O desenvolvimento do dispositivo de captura de sinal biométrico utilizando o arduíno será feita com os conhecimentos adquiridos em eletrônica durante o processo de leitura de materiais relacionados com o tema do trabalho. A comunicação do arduíno com o dispositivo será feita por um programa desenvolvido em linguagem C (KERNIGHAN e RITCHIE, 1988; SCHILDT, 1996). O sinal biométrico utilizado para teste foi o de impressão digital. O arduíno recebe e envia um sinal ao dispositivo que é uma matriz de led 8x8 acoplado, onde acende e apaga os *leds*. As impressões digitais de cada indivíduo serão capturadas por um leitor adquirido em loja e cadastradas através de um sistema de cadastro para um banco de dados. As imagens originais serão processadas e marcadas (extração de minúcias) e armazenadas neste mesmo banco. Para o teste de validação do funcionamento da comunicação entre o arduíno e o dispositivo desenvolvido serão utilizadas as imagens marcadas. Os pontos marcados correspondem aos *leds* acessos na matriz de *led*.

1.1 - OBJETIVOS

O objetivo deste trabalho é desenvolver um dispositivo de captura de sinal biométrico utilizando o arduíno. O sinal biométrico utilizado será a impressão digital, devido ao baixo custo de implantação comercialmente. A comunicação entre o dispositivo e o arduíno será feita por um programa desenvolvido em linguagem C. O dispositivo de captura desenvolvido tem uma matriz de *led* 8x8 acoplada, onde cada um dos *leds* recebe instruções para acender ou apagar. As imagens utilizadas neste processo são as imagens com impressões digitais marcadas (extração de minúcias), onde os pontos marcados correspondem aos *leds* acesso na matriz de *led*.

1.2 - JUSTIFICATIVAS

A biometria surgiu recentemente por meio do avanço tecnológico, com o objetivo de controlar o acesso nas empresas com mais segurança, eliminando os meios tradicionais, e permitindo que os usuários possuam mais liberdade, sem precisar lembrar senhas ou carregar algo para ser identificadas, ganhando tempo no processo. A escolha em desenvolver este projeto é que esta tecnologia está sendo muito utilizada, pois leva em conta o conforto e agilidade dos usuários, cobrindo as necessidades das empresas e também por ser um produto muito eficaz e seguro para várias aplicações.

1.3 - MOTIVAÇÕES

A motivação para o desenvolvimento deste projeto é que o Arduino se tornou uma realidade em vários projetos de automação por ter uma fácil linguagem de desenvolvimento e um forte poder de processamento em pequenos e grandes projetos. Com isso, despertou a curiosidade em conhecer a sua programação e o funcionamento no tratamento de imagem e integração com a matriz *led* 8x8, integrado com outra tecnologia que tem sido muito utilizada que é a biometria.

1.4 - ESTRUTURA DO TRABALHO

A estrutura do trabalho é apresentada em cinco capítulos. Sendo o primeiro a introdução. No segundo capítulo, serão apresentadas as fundamentações teóricas Básicas. No terceiro capítulo, serão apresentados os Desenvolvimentos gerais. No quarto capítulo, é apresentada a conclusão, No quinto capítulo, serão apresentadas as referencias bibliográficas.

2. FUNDAMENTAÇÃO TEÓRICA BÁSICA

Neste capítulo, será feita uma descrição da fundamentação teórica das tecnologias utilizadas para o desenvolvimento do dispositivo que possa receber uma imagem com extensão *bitmap* e enviar o sinal da imagem para uma matriz *led* 8x8.

2.1 – RECONHECIMENTO DE PADRÕES

O Reconhecimento de Padrões (RP) é a ciência que tem por objetivo a classificação de objetos em categorias ou classes. Desde os primórdios da computação, a tarefa de implementar algoritmos emulando essa capacidade humana, tem se apresentado como uma das mais intrigante e desafiadora (JAIN et al, 2000).

As técnicas de reconhecimento de padrões apresentam um vasto leque de aplicações nas áreas científicas e tecnológicas, principalmente na área de informática. O interesse na área de reconhecimento de padrões tem aumentado nos últimos anos, devido a novas aplicações que são não só um desafio, mas também computacionalmente mais exigentes. Estas aplicações incluem *data mining*, ou mineração de dados que identifica um padrão ou uma relação entre milhões de modelos; a classificação de documentos, muito útil para procurar documentos de texto; previsões financeiras; organização e recuperação de bancos de dados multimídia e biometria, que é a identificação pessoal baseada em vários atributos físicos ou comportamentais. A tabela 1 mostra algumas aplicações do reconhecimento de padrões.

Domínio do Problema	Aplicação	Padrão de Entrada	Classes de Padrão
Bioinformática	Análise de Sequência	DNA/Sequência de proteínas	Tipos conhecidos de genes/padrões
Mineração de dados	Busca por padrões significantes	Pontos em um espaço multidimensional	Compactar e bem separar grupos
Classificação de documentos	Busca na Internet	Documento texto	Categorias semânticas (negócios, esportes e etc.)
Análise de documentos de imagem	Máquinas de leitura para cego	Documento de imagem	Palavras e caracteres alfanuméricos
Automação industrial	Inspeção de circuito impresso em placas	Intensidade ou alcance de imagem	Produto defeituoso/não defeituoso
Recuperação de base de dados multimídia	Busca na Internet	Vídeo clipe	Gêneros de vídeos
Reconhecimento biométrico	Identificação pessoal	Face, íris, impressão digital	Usuários autorizados para controle de acesso
Sensoriamento remoto	Prognóstico da produção de colheita	Imagem multiespectral	Categorias de aproveitamento de terra, desenvolvimento de padrões de colheita
Reconhecimento de voz	Inquérito por telefone sem assistência de operador	Voz em forma de onda	Palavras faladas

Tabela 1 - Aplicações do reconhecimento de padrões (JAIN et al, 2000).

A busca para projetar e construir sistemas capazes de realizar o reconhecimento automático de padrões de forma precisa e confiável é imensamente útil. Durante a resolução dos inúmeros problemas necessários para construir tais sistemas, a compreensão de como é realizado o reconhecimento de padrões no mundo real se faz necessária, pois algumas aplicações como o reconhecimento da fala e o reconhecimento facial, influenciam na maneira como esses sistemas são estruturados.

Existe ainda uma nova aplicação para o reconhecimento de padrão, chamado computação afetiva, que dá a um computador a capacidade de reconhecer e expressar emoções e empregar mecanismos que contribuem para a tomada de decisão racional (JAIN et al, 2000).

A técnica de comparação é realizada pelo método do qual se deseja obter o reconhecimento, que é depois conferido com o padrão armazenado. No momento da comparação devem-se levar em conta prováveis translações, rotações ou mudanças de escala que podem acontecer durante o procedimento de aquisição.

A comparação entre os modelos exige bastante processamento, mas a disponibilidade de processadores mais rápidos a preços acessíveis, tem tornado esta abordagem mais viável. Os processos de tratamento das imagens são bem complexos e exige um amplo conhecimento das técnicas e métodos de processamento digital de imagens. Esta complexidade envolve a escolha adequada das técnicas e métodos para obter uma solução razoável para o problema em questão. (SERRANO, 2010).

2.2 - BIOMETRIA

A biometria é um dos tipos utilizados nas aplicações de reconhecimento de padrões e é o ramo da ciência que estuda as medidas dos seres vivos. É uma parte da estatística que estuda os aspectos quantitativos de uma população de seres vivos (ZURADA, 1992).

A biometria, como o próprio nome define "*Bio*" que significa "vida" e "*Metria*" que significa "medida", é a medida da vida, e é através das características físicas e comportamentais que uma pessoa pode ser autenticada em um sistema de segurança (OLIVEIRA, 2002).

Segundo (MANDL, 2003), o uso de um sistema biométrico é simples. O sistema consiste no envio dos dados das características do usuário para um sistema de controle e gerenciamento dessas informações, e vale desde a impressão digital até a composição da retina da pessoa. Depois disso, existe uma comparação entre as informações coletadas com as que estão em um banco de dados e, caso exista semelhança, a autenticação é efetuada com sucesso.

De acordo com (VIGLIAZZI, 2003), qualquer característica humana, física ou comportamental pode ser usada para identificar pessoas, desde que satisfaçam os seguintes requerimentos:

- **Universalidade:** Significa que todas as pessoas devem possuir a característica;
- **Singularidade:** Indica que esta característica não pode ser igual em pessoas diferentes;

- **Permanência:** Significa que a característica não deve variar com o tempo;
- **Mensurabilidade:** Indica que a característica pode ser medida quantitativamente;
- **Desempenho:** Refere-se à precisão de identificação, aos recursos requeridos para conseguir uma precisão de identificação aceitável e ao trabalho ou fatores ambientais que afetam a precisão da identificação;
- **Aceitabilidade:** Indica o quanto as pessoas estão dispostas a aceitar os sistemas biométricos;
- **Proteção:** Refere-se à facilidade/dificuldade de enganar o sistema com técnicas fraudulentas.

É vastamente reconhecida pela sua individualidade, e seu uso surgiu no final do século XIX. Sendo hoje a tecnologia mais utilizada no mundo da biometria, (AMORIM, 2005).

As impressões digitais são ideais para a identificação pessoal sendo que elas são únicas e exclusivas. De acordo com (AMORIM, 2005), não existem duas impressões digitais iguais na face da terra. A palma das mãos e a planta dos pés, incluindo as partes internas dos dedos apresentam sulcos em padrões concêntricos. A finalidade destes sulcos é de gerar uma maior fricção para segurar objetos ou apoiar em superfícies com mais firmeza e segurança.

Tendo como características, os sulcos das impressões digitais são curvos, bifurcados e partidos, os pontos de mudança de direção e as bifurcações são conhecidos como minúcias. A figura 2 mostra as peculiaridades de uma impressão digital.



Figura 2 - Detalhes peculiares de uma impressão digital. (KALKULAR, 2007).

Como suas características são únicas, cada um desses pontos pode ser medido. As informações que podem ser obtidas são as suas posições relativas, as quantidades e o conjunto das minúcias que as impressões digitais apresentam, estes aspectos é que individualizam cada pessoa.

Os dois tipos de minúcias mais importantes são: crista final e a crista bifurcada. A crista final está definida como o ponto onde a linha termina abruptamente. A crista bifurcada está definida com o ponto onde uma linha aforquilha ou diverge em linhas ramificadas (ESPINOSA - DURÓ, 2002).

Para o processamento computacional das impressões digitais é necessário um tratamento das imagens, dependendo da aplicação que se quer fazer. No caso, de controle de acesso de pessoas, uma das técnicas mais utilizadas é a realização de marcação das minúcias para que se possa elaborar um algoritmo que faça a comparação das imagens para a autenticação (verificar se a impressão digital realmente é do indivíduo em questão).

A figura 3 mostra a imagem de uma impressão digital marcada, após extração de minúcias.



Figura 3 - Imagens marcadas após extração de minúcias (COSTA, 2001).

O reconhecimento de pessoas é feito há muitos anos por institutos oficiais de identificação de diversos países através do sistema de análise da impressão digital. Na Europa, judicialmente, são necessárias 12 minúcias para saber quem é uma pessoa. Os leitores biométricos são capazes de identificar mais de 40 minúcias de uma impressão digital (ROMAGNOLI, 2002).

2.3 – LEITORES BIOMÉTRICOS DE IMPRESSÃO DIGITAL

Antigamente os leitores de impressões digitais era um objeto fantástico, só visto em filmes de espionagem, mas hoje em dia eles estão por toda parte: em distritos policiais, autoescolas e até mesmo em teclados de computador, que ao invés de senhas, precisa de sua característica para acessar algo restrito. Um sistema de leitor de impressões digitais tem duas funções básicas:

- Precisa obter uma imagem do dedo;
- Precisa determinar se o padrão de sulcos nessa amostra confere com o padrão de sulcos em imagens previamente capturadas.

Há várias maneiras de se obter a imagem do dedo de alguém. Os métodos mais comuns atualmente são a leitura ótica e a leitura capacitiva. Os dois métodos resultam no mesmo tipo de imagem, mas chegam a ela de maneiras completamente diferentes. Para coletar a impressão digital existem três tipos de leitores: ultrassônico, ótico e capacitivo.

2.3.1 - Ultrassônico

O ultrassônico trabalha enviando sinais sonoros e analisando o retorno deles como se fosse um radar milimétrico, este produto é pouco conhecido e utilizado (HARRIS, 2002).

2.3.2 - Ótico

O coração de um leitor ótico é um dispositivo de carga acoplado (CCD), o mesmo sistema de sensor por luz utilizado em câmeras digitais e filmadoras. Um CCD nada mais é que um grupo de diodos fotossensíveis chamados *photosites*, que emitem um sinal elétrico em resposta aos fótons de luz. Cada *photosite* grava um *pixel*, um minúsculo ponto que representa a luz incidida numa determinada posição. No geral, os *pixels* escuros e claros dão forma a uma imagem capturada. Normalmente, um conversor analógico-digital no leitor processa o sinal elétrico analógico e gera uma representação digital da imagem (HARRIS, 2002).

O processo de leitura inicia quando “se coloca” o dedo sobre uma base de vidro e uma câmera CCD “tira uma foto”. O leitor possui sua própria fonte de luz, normalmente um grupo de diodos foto emissores, para iluminar os sulcos do dedo. O CCD gera na prática uma imagem invertida do dedo, com as áreas mais escuras representando mais luz refletida (os sulcos do dedo) e áreas mais claras representando menos luz refletida (vales entre os sulcos). Antes de comparar os dados digitais “com os” dados armazenados, o processador do leitor certifica-se de que o CCD obteve uma imagem bem nítida. Ele verifica a penumbra média dos pixels, ou os valores totais em uma amostra pequena, e rejeita a imagem se a média estiver muito escura ou muito clara. Se a imagem for rejeitada, o leitor ajusta o

tempo de exposição para deixar que mais ou menos luz entre e procede-se uma nova leitura.

Se o nível de penumbra for adequado, o sistema do leitor passa a verificar a definição da imagem. O processador observa diversas linhas retas que se movem horizontal e verticalmente pela imagem. Se a imagem das impressões digitais tiver uma boa definição, uma linha em movimento perpendicular aos sulcos irá se formar a partir da alternância dos setores dos *pixels* muito claros ou muito escuros.

Se o processador achar que a imagem está clara e exibida corretamente, passa para a comparação das impressões digitais capturadas com as impressões digitais em arquivo.

2.3.3 - Capacitivo

Assim como os leitores óticos, leitores capacitivos de impressões digitais geram uma imagem dos sulcos e vales que formam as impressões digitais. Mas ao invés de captar a digital utilizando a luz, os capacitores utilizam a corrente elétrica.

A figura 4 mostra um sensor capacitivo simples, composto de um ou mais circuitos semicondutores que contêm um grupo de minúsculas células, menores que a largura de um sulco do dedo. Cada célula inclui duas placas condutoras, cobertas com uma camada isolante (HARRIS, 2002).

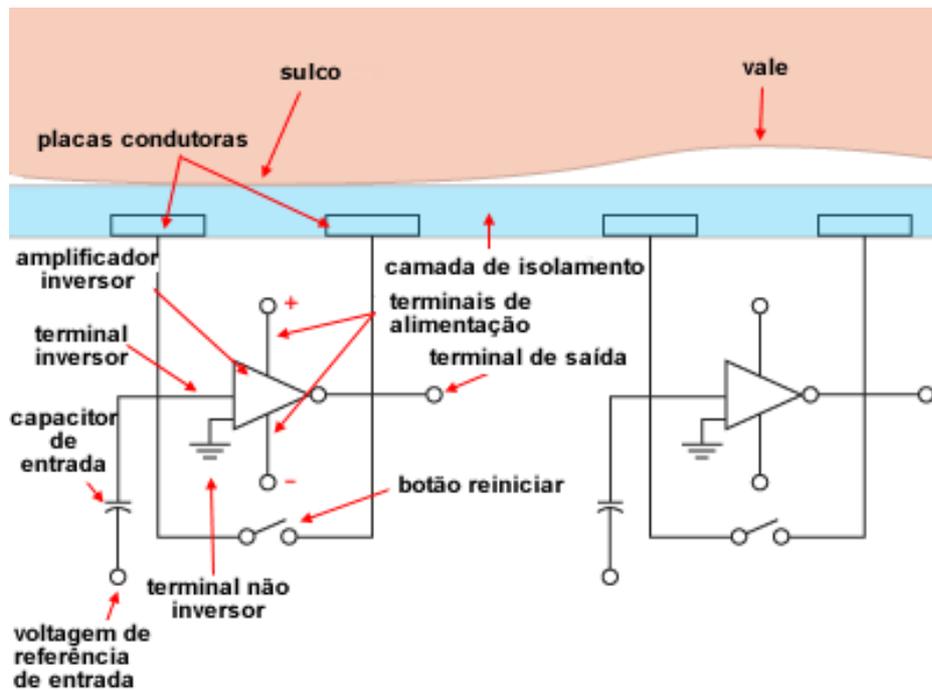


Figura 4 - Sensor Capacitivo. (HARRIS, 2002).

O sensor é conectado a um integrador, circuito elétrico construído ao redor de um amplificador operacional inversor, que é um dispositivo de semicondutores complexo, composto de grande número de transistores, resistores e capacitores.

Como todo amplificador, o inversor altera uma corrente baseada nas flutuações de outra corrente. Especificamente, o amplificador inversor altera uma tensão de alimentação, baseada na tensão relativa de duas entradas, chamadas de terminal inversor e terminal não inversor. Neste caso, o terminal não inversor é aterrado e o terminal inversor é conectado a uma fonte de tensão referencial e a um ciclo de realimentação. O ciclo de realimentação, que é conectado também à saída do amplificador, compreende duas placas condutoras (HARRIS, 2002).

Para fazer a leitura de um dedo, o processador inicialmente fecha o botão de reiniciar de cada célula, o que encurta a entrada e saída de cada amplificador para balancear o circuito integrador. Quando o botão é reaberto e o processador aplica uma carga fixa ao circuito integrador, os capacitores são carregados. A capacitância do ciclo de realimentação afeta sua tensão na entrada do amplificador, o que afeta também sua saída. Como a distância até o dedo altera a capacitância, um sulco do dedo resultará numa tensão diferente de um vale (HARRIS, 2002).

O processador do leitor lê esta saída de tensão e determina se é característica de um vale ou um sulco. Através da leitura de cada célula do sensor, o processador

pode montar uma imagem da impressão digital, semelhante à imagem capturada pelo leitor ótico (HARRIS, 2002).

2.4 – ARDUÍNO

O Arduino é um micro controlador e alguns outros componentes eletrônicos montados numa pequena placa de circuito impresso com uma interface serial para comunicação com um computador padrão PC. Nessa placa existem também alguns conectores onde podem ser ligados outros circuitos externos, como sensores, *leds*, chaves, relés e pequenos motores (SILVEIRA 2011). O Arduino é uma ferramenta para criação de protótipos de eletrônica baseada no conceito de *software* e *hardware* livres. Isso quer dizer que esses projetos podem ser copiados e modificados por outras pessoas conforme suas necessidades e depois podem ser colocados de volta ao domínio público de modo que outros usuários possam usufruir dessas mudanças em seus próprios projetos. A figura 5 mostra alguns modelos de arduino.

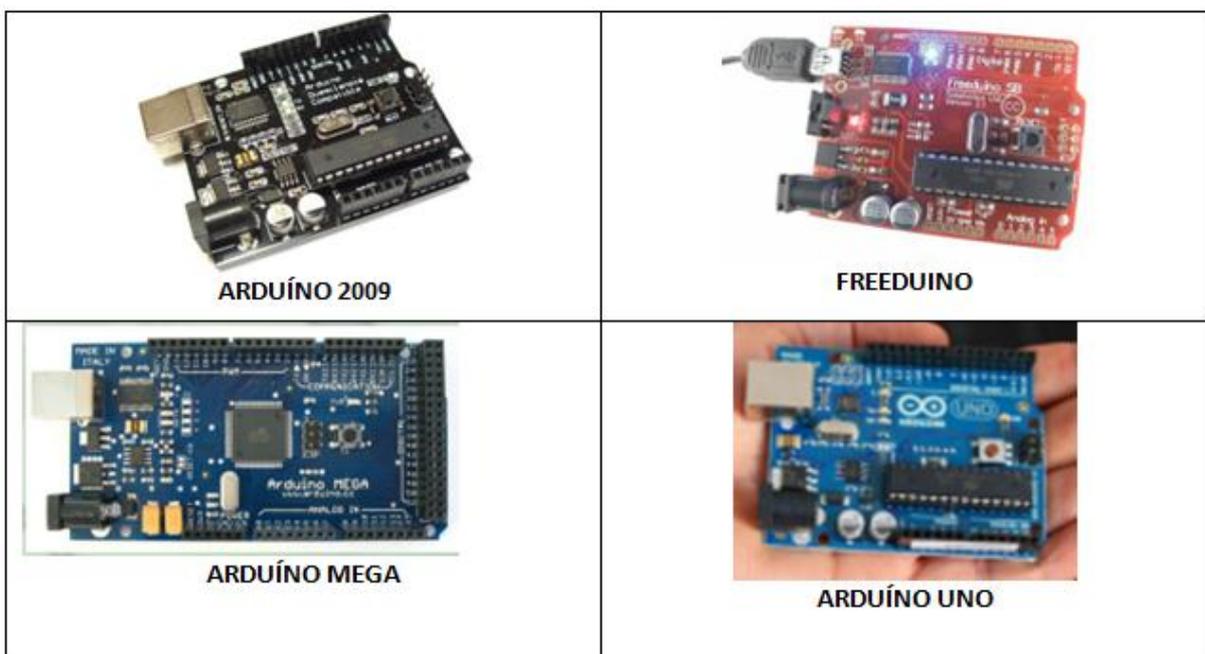


Figura 5 - Modelos de Arduino

O Arduino é composto por duas partes principais: um *hardware*, a placa de circuito impresso com o micro controlador, e um *software*, o *bootloader*, um aplicativo residente na memória de programas desse micro controlador. Externamente existe

também uma interface gráfica, um programa que roda em ambiente Windows ou Linux num computador PC ou numa máquina Apple com o Mac OS X. É nessa interface gráfica ou ambiente de desenvolvimento integrado (IDE – *Integrated Development Environment*) onde criamos os programas que vão ser carregados no Arduino. São esses programas, chamados de *sketches*, que vão dizer ao *hardware* o que deve ser feito, (SILVEIRA 2011).

Na verdade, o arduíno é um kit de desenvolvimento capaz de interpretar variáveis no ambiente e transformá-las em sinal elétrico correspondente, através de sensores ligados aos seus terminais de entrada, e atuar no controle ou acionamento de algum outro elemento eletroeletrônico conectado ao terminal de saída. Ou seja, é uma ferramenta de controle de entrada e saída de dados, que pode ser acionada por um sensor (por exemplo, um resistor dependente da luz - LDR) e que, logo após passar por uma etapa de processamento, o micro controlador, poderá acionar um atuador (um motor, por exemplo). Como podem perceber, é como um computador, que têm como sensores de entrada como o mouse e o teclado, e de saída, impressoras e caixas de som, por exemplo, só que ele faz interface com circuitos elétricos, podendo receber ou enviar informações/tensões neles.

2.4.1 – Hardware

O *hardware* do Arduino é baseado nos microcontroladores AVR da Atmel, em particular nos modelos ATmega8, ATmega168, ATmega328 e no ATmega1280. O arduíno recebe um codinome em italiano dependendo do microcontrolador utilizado.

A tabela 2 mostra as principais diferenças entre os arduínos em relação ao microcontrolador que ele recebe.

ARDUINO	Diecimila	Duemilanove168	Duemilanove328	Mega
Processador	ATmega8	ATmega168	ATmega328	ATmega1280
Memória <i>flash</i>	8 k	16 K	32 K	128 K
Memória RAM	1 K	1 K	2 K	8 K
Memória EEPROM	512 bytes	512 bytes	1 K	4 K
Pinos digitais	14	14	14	54
Pinos analógicos	6	6	6	16
Saídas PWM	3	6	6	14

Tabela 2- Principais Diferenças Entre os Arduínos (SILVEIRA, 2001).

O projeto original do arduino foi baseado no circuito básico com um microcontrolador ATmega8, cujo diagrama em blocos é mostrado na figura 6.

Entretanto, uma fonte de uma fonte de alimentação simples com o regulador LM7805, um circuito de conversão para comunicação serial RS-232 e alguns conectores para controle externo e entrada de sensores também foram incorporados a este mesmo projeto. Além disso, foi incorporado posteriormente ao projeto, um conector para a programação do microcontrolador no circuito, o ICSP ou *In-Circuit Serial Programming*.

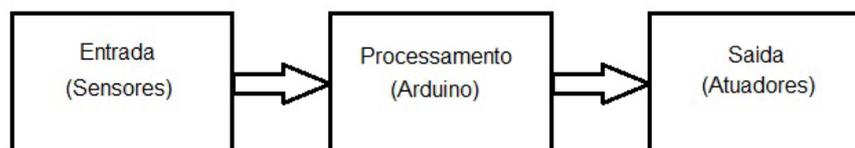


Figura 6 - Diagramas de Blocos

2.4.2 – Características do arduíno

Para a utilização do arduíno é necessário um bom conhecimento das características básicas de *hardware* para que sejam explorados ao máximo todos os recursos disponíveis. A figura 7 mostra a arquitetura básica de um arduíno.

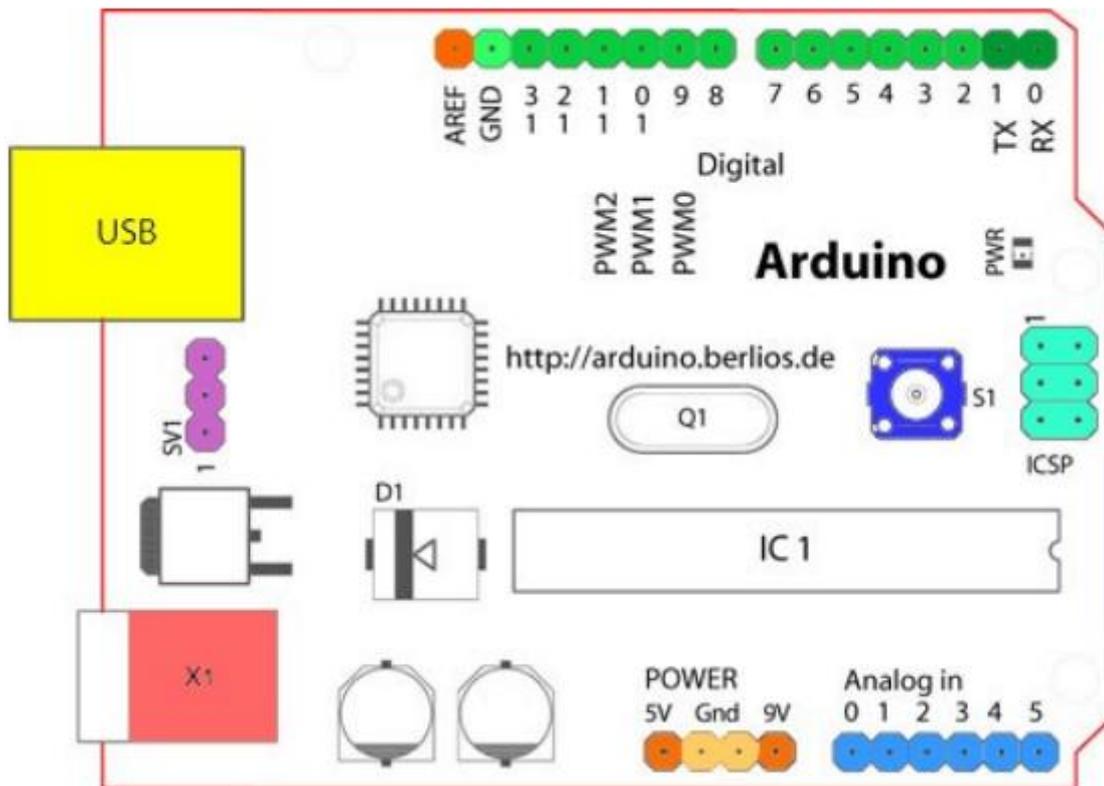


Figura 7 - Arquitetura Básica de um Arduino (SILVA, 2010).

O *hardware* do arduino é uma placa baseada no microcontrolador ATmega. Tem 14 pinos de entrada ou saída digital (dos quais 6 podem ser utilizados como saídas PWM), 6 entradas analógicas, um oscilador de cristal 16MHz, controlador USB, uma tomada de alimentação, um conector ICSP, e um botão de reset. Para sua utilização basta conectá-lo a um computador com um cabo USB ou ligá-lo com um adaptador AC para DC ou bateria.

O arduino pode ser alimentado pela conexão USB ou por qualquer fonte de alimentação externa. A fonte de alimentação é selecionada automaticamente. De acordo com (FONSECA e BEPPU, 2010), os pinos de alimentação são:

- 5V: a fonte de alimentação utilizada para o microcontrolador e para outros componentes da placa. Pode ser proveniente do pino 9V através de um regulador *on-board* ou ser fornecida pelo USB ou outra fonte de 5V;
- X1: suprimento externo de energia In (9-12VDC);

- 9V: entrada de alimentação para a placa arduino quando uma fonte externa for utilizada. Pode fornecer alimentação por este pino ou, se usar o conector de alimentação, acessar a alimentação por este pino;
- GND (*ground*): pino terra;
- SV1: Jumper que determina a alimentação da USB ou externa X1.

Cada um dos 14 pinos digitais do arduino pode ser usado como entrada ou saída usando as funções de *pinMode()*, *digitalWrite()* e *digitalRead()*. Eles operam com 5 V. Cada pino pode fornecer ou receber um máximo de 40 mA e tem um resistor *pull-up* interno (desconectado por padrão) de 20-50k .Além disso, de acordo com (FONSECA e BEPPU, 2010), alguns pinos têm funções especializadas:

- Digital Pins 0-1/Serial In TX/RX (Verde Escuro) – estes pinos não podem ser usados para digital i/o (*digitalRead* e *digitalWrite*) se estiver usando comunicação serial (*Serial.begin*)
- PWM: 3,5,6,9,10,e11. Fornecem uma saída analógica PWM de 8-bit com a função *analogWrite()*;
- SPI: 10(SS), 11(MOSI), 12(MISO), 13(SCK). Estes pinos suportam comunicação SPI, que embora compatível com o hardware, não está incluída na linguagem do arduino;
- LED: 13. Há um LED já montado e conectado ao pino digital13;
- AREF: Analog Reference pin-AREF, referência de tensão para entradas analógicas. Usados com *analogReference()*.

O arduino tem um fusível que protege a porta USB do seu computador contra curto circuito e sobrecarga de corrente. Apesar da maioria dos computadores possuírem proteção interna própria o fusível proporciona uma proteção extra. Se mais de 500mA forem aplicados na porta USB, o fusível irá automaticamente interromper a conexão até que o curto ou a sobrecarga seja removida.

- USB: Usada para gravar os programas; Comunicação serial entre placa e computador; Alimentação da placa (Amarelo).

Algumas versões anteriores do arduino requerem um reset físico (pressionando o botão de reset na placa) antes de carregar um sketch (o programa a ser compilado). Os arduino baseados em microcontrolador ATmega168 ou ATmega328 foram projetados de modo a permitir que isto seja feito através do software que esteja rodando no computador conectado. Uma das linhas de controle de hardware (DTR) do FT232RL está conectada ao reset do ATmega328 via um capacitor de 100microFaraday. Quando esta linha é colocada em nível lógico baixo, o sinal cai por tempo suficiente para reiniciar o chip. O software Arduino usa esta característica para permitir carregar o programa simplesmente pressionando o botão \upload no ambiente Arduino. Isto significa que o bootloader pode ter um timeout mais curto, já que a ativação do DTR (sinal baixo) pode ser bem coordenada com o início do upload.

- S1: Botão de Reset

O arduino tem 6 entradas analógicas, cada uma delas está ligada a um conversor analógico-digital de 10bits, ou seja, transformam a leitura analógica em um valor dentre 1024 possibilidades. Por padrão, elas medem de 0 a 5 V, embora seja possível mudar o limite superior usando o pino AREF e um pouco de código de baixo nível.

- Analog In Pins 0-5

Além disso, tem o ICSP que nada mais é do que um conector.

- ICSP: In-circuit Serial Program

A figura 8 mostra outros dispositivos importantes no arduino.

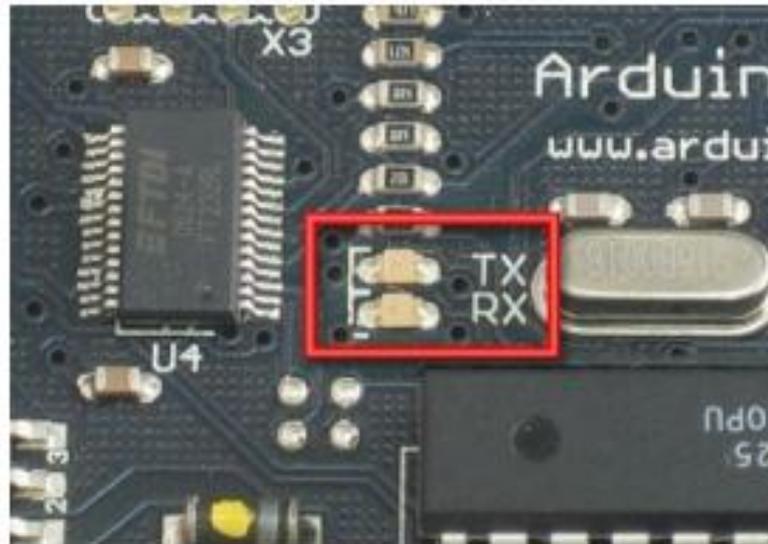


Figura 8 - Dispositivos Importantes do Arduíno (COSTA, 2010).

A comunicação com um computador, com outro arduino ou com outros microcontroladores é muito simplificada. O arduino permite comunicação serial no padrão UARTTTL (5V), que está disponível nos pinos digitais 0(RX)e1(TX). Um chip FTDI (FTDIFT232RL) na placa encaminha esta comunicação serial através da USB e os drivers FTDI (incluído no software do Arduino) fornecem uma porta virtual para o software no computador. O software Arduino inclui um monitor serial que permite que os dados de texto sejam enviados e recebidos à placa Arduino. Os LEDs RX e TX da placa piscam quando os dados estão sendo transferidos ao computador pelo chip FTDI e há conexão USB (mas não quando há comunicação serial pelos pinos 0 e 1).

- Led Serial: 0 (RX) e 1 (TX). Usados para receber (RX) e transmitir (TX) dados seriais TTL. Estes pinos são conectados aos pinos correspondentes do chip serial FTDI USB-to-TTL;
- Chip FTDI (comunicação serial USB)

A figura 9 mostra a disposição dos pinos do Atmega168:

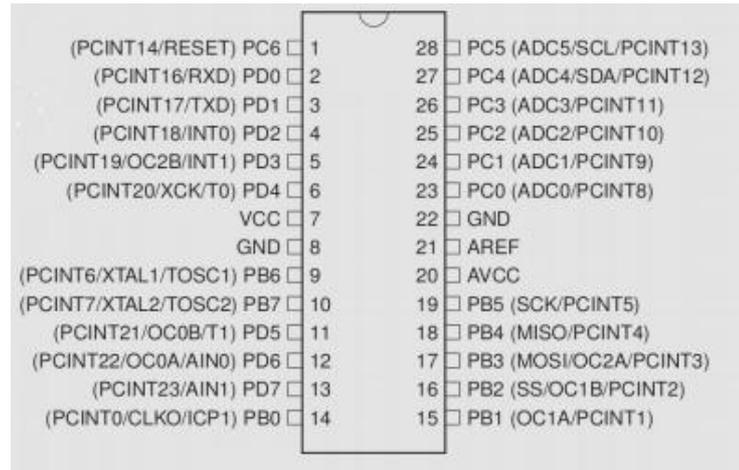


Figura 9 - Pinagem Atmega168.

A Figura 10 mostra o diagrama de blocos interno do Atmega 168:

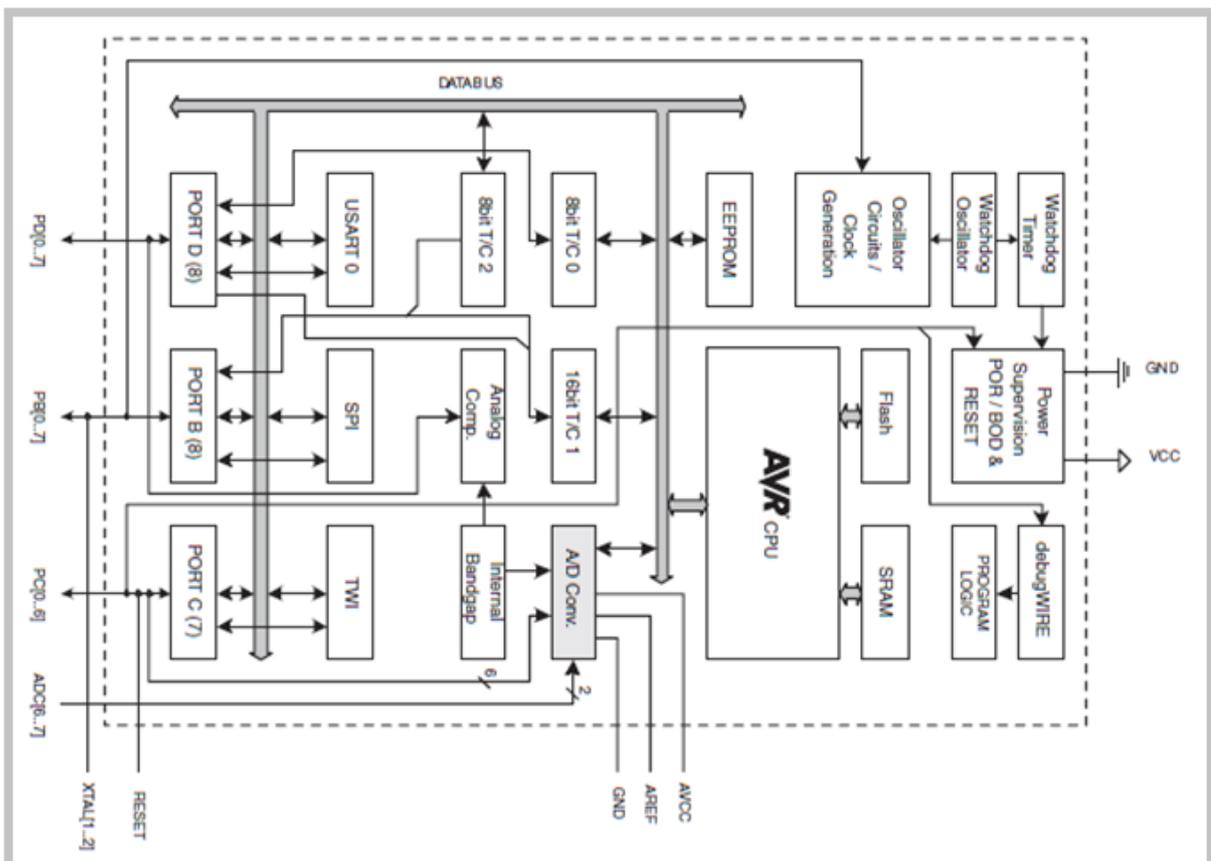


Figura 10 - Diagrama de Blocos

2.4.3 – Ambiente de Programação para o Arduino

Toda programação é baseada em uma linguagem nova, chamada *processing*. No ambiente de desenvolvimento existem também várias funções que facilitam o desenvolvimento de qualquer programa, do mais simples ao complexo, além de possuir bibliotecas prontas para facilitar o interfaceamento com outros *hardwares*.

2.4.3.1 - IDE Arduino

O arduino IDE é uma aplicação multi-plataforma escrita em Java, na qual é baseado no ambiente de programação open source Processing e a sintaxe da linguagem é baseada na biblioteca Wiring (baseado em C e C++). Na maioria dos casos os programadores que desenvolvem software para o Arduino não utilizam as linguagens C ou C++, mas a linguagem denominada Process que pode ser considerada a linguagem do Arduino. A linguagem Process é similar às linguagens C e C++, contudo possui uma série de restrições. O software arduino tem um ambiente gráfico muito semelhante ao do *Processing*.

Um programa escrito em process é denominado sketch e deve ser escrito no Ambiente de desenvolvimento Arduino. O ambiente Arduino é escrito na linguagem Java e assim está disponível para diversos sistemas.

Quando um sketch é compilado, o ambiente Arduino varre o código fonte Process fazendo algumas substituições e transforma o código Process em linguagem C/C++ que é compilado efetivamente pelo compilador da GNU.

A figura 11 mostra o ambiente gráfico do arduino.

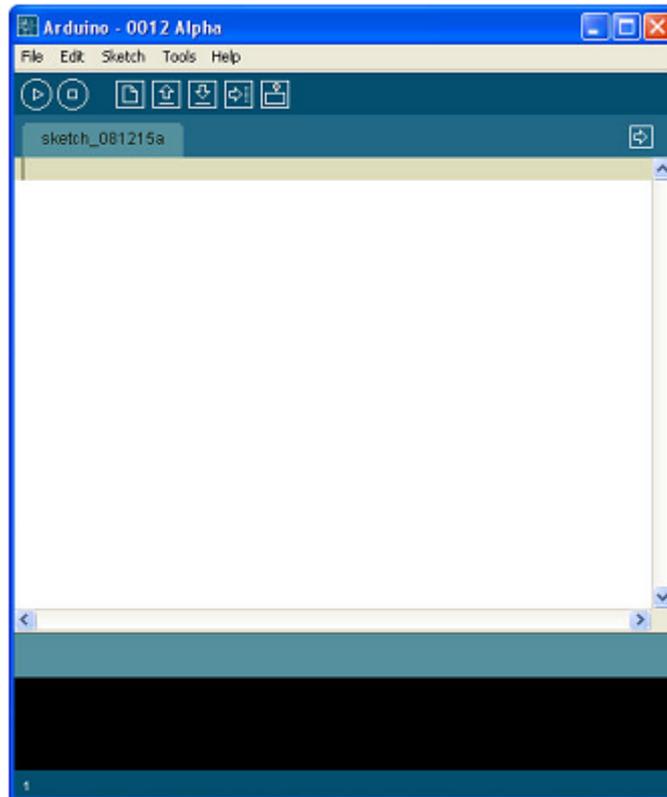


Figura 11 - Ambiente Gráfico do Arduíno (ENDEL B, 2008).

O ambiente é constituído pelo Toolbar, Tab Menu e Menus (*File*, *Edit*, *Sketch*, *Tools* e *Help*). A tabela 3 mostra os vários botões com funções distintas.

	Verifica se o código tem erros
	Para o serial monitor ou desativa outros botões
	Cria um novo documento
	Abre uma lista de documentos salvos e abre o que for selecionado
	Salva o documento
	Envia código para o Arduino
	Mostra a informação enviada pela placa do arduíno
	Cria nova Tab

Tabela 3 - Comandos do Toolbar e Suas Funções (adaptado de ENDEL C, 2008).

O Tab Menu  permite gerir documentos com mais do que um ficheiro, cada um aberto num tab independente. Esses ficheiros podem ser ficheiros normais de código Arduino (sem extensão), ficheiros C (extensão .c), C++ (.cpp) ou *header files* (.h). Os Menus *File*, *Edit* e *Help* são semelhantes em todos os programas, e não será feita uma descrição detalhada de cada uma delas. A tabela 4 mostra o menu *Sketch*.

Verify/Compile	Verifica se o código tem erros
Import Library	Adiciona livrarias ao código, o que lhe confere novas possibilidades
Show Sketch Folder	Abre a pasta do ficheiro no Ambiente de Trabalho
Add File...	Adiciona outro ficheiro ao ficheiro inicial. Esse ficheiro aparece numa nova tab

Tabela 4 - Comandos do Menu Sketch e suas funções (adaptado de ENDEL C, 2008).

O menu *Tools* é composto por outras funções que são intrínsecas a este programa. A tabela 5 mostra os comandos do menu tools e suas funções.

Auto Format	Formata o código para uma melhor leitura, alinhando as chavetas e indentando o seu conteúdo.
Copy for Discourse	Copia o código para o clipboard, de forma que seja possível colocá-lo num fórum e manter a distinção da sintaxe por cores.
Board	Seleciona a placa que estamos a usar, o que controla a forma como o código é compilado e o modo como é feito o upload, assim como o comportamento dos itens do menu Burn Bootloader
Serial Port	Contém todas as portas série que o computador possui, permitindo escolher a que está a ser usada. Para usar portas USB é necessário instalar um driver ³ . No Mac, a porta série USB tem uma apresentação semelhante a esta: <code>/dev/tty.usbserial-1B1</code> e no Windows a esta: <code>COM1</code> , variando os números finais.
Burn Bootloader	Os itens deste menu permitem gravar um bootloader na placa <i>Arduino</i> com variados programas, mas não é necessário para um uso normal do <i>Arduino</i> .

Tabela 5 - Comandos do Menu Tools e Suas Funções (adaptado de ENDEL C, 2008)

A figura 12 mostra a arquitetura de programação em arduino.

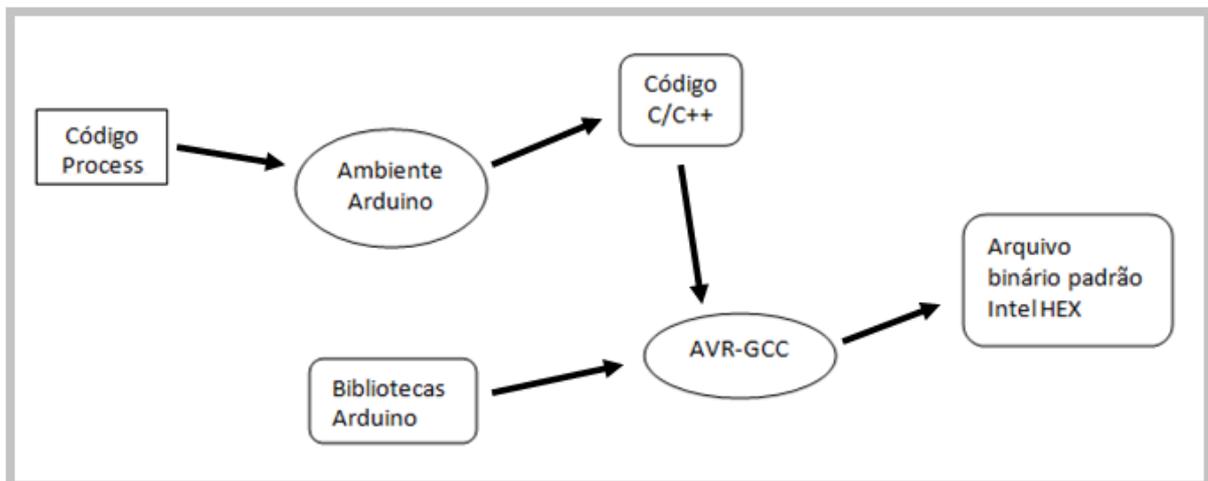


Figura 12 - Arquitetura de Programação no Arduino.

No Ambiente Arduino são realizadas algumas transformações no código Process, o resultado é um código C/C++. No Processo GCC é compilado o código C/C++ e junta as bibliotecas para controle dos recursos do microcontrolador, tais como

Serial.printf(), digitalWrite()). No fim de todo esse processo ele gera um arquivo binário que será gravado na memória do microcontrolador.

2.4.3.2 – Linguagem de programação para arduino

Neste capítulo será apresentada toda a parte de linguagem de programação, citando os comando e especificações.

2.4.3.2.1 - Linguagem de referência

As funções são referências essenciais para o desenvolvimento de um projeto sendo o Arduino, principalmente para os iniciantes no assunto. Essas funções já implementadas e disponíveis em bibliotecas direcionam e exemplificam as funcionalidades básicas do microcontrolador. Segundo (FONSECA e BEPPU, 2010) as funções básicas e de referências são:

- **Estruturas de referências**

- Estruturas de controle (if, else, break, ...);
- Sintaxe básica (define, include, ; , ...);
- Operadores aritméticos e de comparação (+, -, =, ==, !=, ...);
- Operadores booleanos (, ||, !);
- Acesso a ponteiros (*,);
- Operadores compostos (++ , - , += , ...);
- Operadores de bits (|, ^ ; ...).

- **Valores de referências**

- Tipos de dados (byte, array, int , char , ...);
- Conversões(char(), byte(), int(), ...);
- Variável de escopo e de qualificação (variable scope, static, volatile, ...);
- Utilitários (sizeof(), diz o tamanho da variável em bytes).

É bom citar que o software que vem no Arduino já provê várias funções e constantes para facilitar a programação que são:

- `setup()`;
- `loop()`;
- Constantes (HIGH | LOW , INPUT | OUTPUT , ...);
- Bibliotecas (Serial, Servo, Tone, etc.).

2.4.3.2.2 - Funções

As funções são referências essenciais para o desenvolvimento de um projeto usando o Arduino, principalmente para os iniciantes no assunto. Essas funções já implementadas e disponíveis em bibliotecas direcionam e exemplificam as funcionalidades básicas do microcontrolador. Segundo (FONSECA e BEPPU, 2010) as funções básicas e de referências são:

- Digital I/O - `pinMode()` `digitalWrite()` `digitalRead()`;
- Analógico I/O - `analogReference()` `analogRead()` `analogWrite()` – PWM;
- Avançado I/O - `tone()` `noTone()` `shiftOut()` `pulseIn()`;
- Tempo - `millis()` `micros()` `delay()` `delayMicroseconds()`;
- Matemática - `min()` `max()` `abs()` `constrain()` `map()` `pow()` ***só do C/C++ `sqrt()`
***só do C/C++;
- Trigonometria - `sin()` ***só do C/C++ `cos()` ***só do C/C++ `tan()` ***só do C/C++;
- Números aleatórios - `randomSeed()` `random()`;
- Bits e Bytes - `lowByte()` `highByte()` `bitRead()` `bitWrite()` `bitSet()` `bitClear()` `bit()`;
- Interrupções externas -`attachInterrupt()` `detachInterrupt()`;
- Interrupções - `interrupts()` `noInterrupts()`;
- Comunicação Serial.

2.4.3.2.3 – Bibliotecas

O uso de bibliotecas nos proporciona um horizonte de programação mais amplo e diverso quando comparado à utilização apenas de estruturas, valores e funções. Isso é perceptível quando analisamos os assuntos que são abordados por cada biblioteca em específico. Lembrando sempre que, para utilizar uma biblioteca esta já deve estar instalada e disponível na sua máquina. As bibliotecas de referencias são:

- EEPROM - leitura e escrita de "armazenamento" permanente;
- Ethernet - para se conectar a uma rede Ethernet usando o Arduino Ethernet Shield;
- Firmata - para se comunicar com os aplicativos no computador usando o protocolo Firmata;
- LiquidCrystal - para controlar telas de cristal líquido (LCDs);
- Servo - para controlar servo motores;
- SPI - para se comunicar com dispositivos que utilizam barramento Serial Peripheral Interface (SPI);
- SoftwareSerial - Para a comunicação serial em qualquer um dos pinos digitais;
- Stepper - para controlar motores de passo;
- Wire - Dois Wire Interface (TWI/I2C) para enviar e receber dados através de uma rede de dispositivos ou sensores.

Além dessas referencias, existem algumas bibliotecas mais específicas. Isso é de extrema importância, pois é possível utilizar o arduino com um enfoque em uma determinada área. De acordo com (FONSECA e BEPPU, 2010) essas bibliotecas são:

2.4.3.2.3.1 - Comunicação (redes e protocolos)

- Messenger - Para o processamento de mensagens de texto a partir do computador;
- NewSoftSerial - Uma versão melhorada da biblioteca SoftwareSerial;
- OneWire - Dispositivos de controle que usam o protocolo One Wire;
- PS2Keyboard - Ler caracteres de um PS2 teclado;
- Simple Message System - Enviar mensagens entre Arduino e o computador;

- SSerial2Mobile - Enviar mensagens de texto ou e-mails usando um telefone celular;
- Webduino - Biblioteca que cria um servidor Web (para uso com o Arduino Ethernet Shield);
- X10 - Envio de sinais X10 nas linhas de energia AC;
- XBee - Para se comunicar via protocolo XBee;
- SerialControl - Controle remoto através de uma conexão serial.

2.4.3.2.3.2 - Sensoriamento

- Capacitive Sensing - Transformar dois ou mais pinos em sensores capacitivos;
- Debounce - Leitura de ruídos na entrada digital.

2.4.3.2.3.3 - Geração de Frequência e de Áudio

- Tone - Gerar ondas quadradas de frequência de áudio em qualquer pino do microcontrolador.

2.4.3.2.3.4 - Temporização

- DateTime - Uma biblioteca para se manter informado da data e hora atuais do software;
- Metro - Ajuda ao programador a acionar o tempo em intervalos regulares;
- MsTimer2 - Utiliza o temporizador de 2 de interrupção para desencadear uma ação a cada N milissegundos.

2.4.3.2.3.5 - Utilidades

- TextString (String) - Manipular strings
- PString - uma classe leve para imprimir em buffers.
- Streaming - Um método para simplificar as declarações de impressão.

2.4.3.3 – Comunicação Entre o Arduíno e o Processing

Para fazer o Arduino se comunicar com o *Processing* é preciso pegar a biblioteca que é responsável por criar o caminho de comunicação entre os dois. Para que a biblioteca funcione corretamente é necessário ter rodando no Arduino o *firmware* Firmata. É possível realizar a comunicação entre os dois fazendo uso da biblioteca serial do *processing* e do Arduino, mas isso dá um pouco mais de trabalho e não me parece elegante.

3. DESENVOLVIMENTO DO PROJETO

Neste capítulo será apresentada a elaboração do protótipo para reconhecer a impressão digital marcada e salva no computador, utilizando os conceitos obtidos durante a revisão bibliográfica. É apresentada também a descrição e outros fatores do problema, bem como os métodos utilizados para obter a solução.

3.1 – DESCRIÇÃO DO PROBLEMA

Foi desenvolvido um protótipo utilizando o Arduino para processar o reconhecimento de uma imagem gerada pelo leitor de impressão digital e redimensionada em uma matriz 8x8, para que o mesmo possa enviar os sinais exatos para Matriz de leds 8x8. As imagens feitas pelo leitor foram armazenadas em um banco de dados a partir de um sistema de cadastro de pessoa.

3.2 – MODELAGEM DO PROBLEMA

Nesta seção, será apresentada a modelagem do sistema geral do projeto. O modelo tem o leitor de impressão comprado, o arduino será programado para emitir o sinal correto para a matriz de *leds* 8x8, e no computador serão armazenadas as imagens capturadas pelo leitor de impressão digital através de um sistema de cadastro o computador também é onde se programa o Arduino para reconhecer e enviar o sinal. A figura 13, mostra a modelagem do problema.

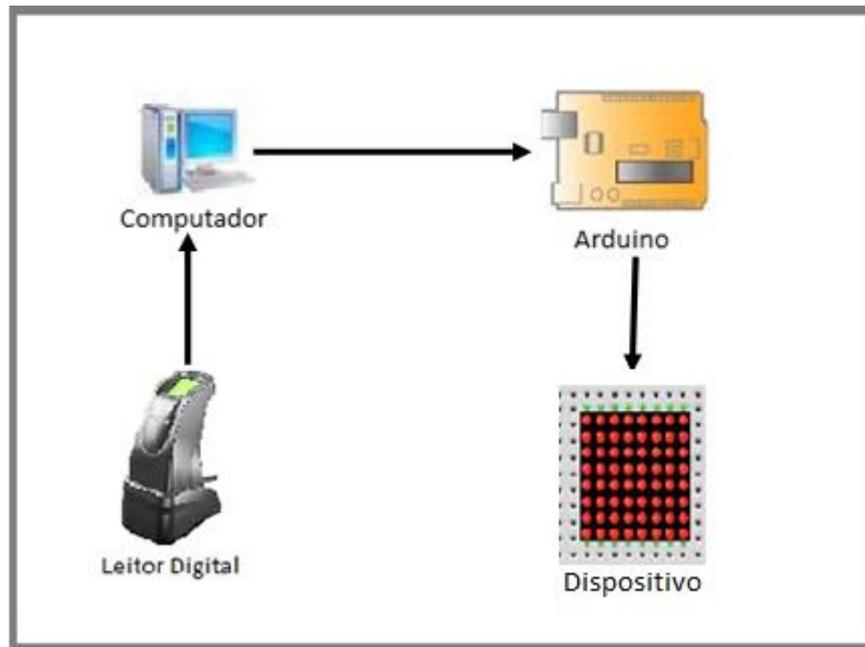


Figura 13 - Modelagem do problema.

O desenvolvimento do projeto foi dividido em módulos para facilitar a sua implementação.

Módulo 1: Criação do Banco de Dados;

Módulo 2: Sistema de Cadastramento;

Módulo 3: Desenvolvimento do Dispositivo de Captura de Sinal Biométrico;

Módulo 4: Integração do Dispositivo Com o Arduíno.

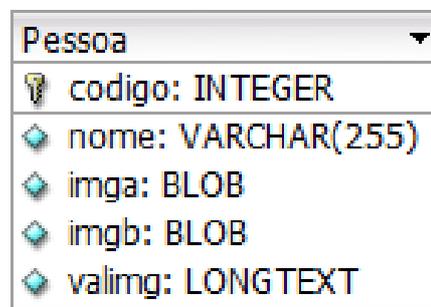
3.3 – IMPLEMENTAÇÃO

Será apresentado a implementação de cada módulo do projeto para captura de sinal biométrico utilizando o arduíno.

3.3.1 – Criação do banco de dados

Para a criação do banco de dados foi necessária para armazenar as informações de cada pessoa e suas respectivas impressões digitais.

A figura 14 mostra o modelo lógico do banco de dados usado para o gerenciamento das informações.



Pessoa	
🔑	codigo: INTEGER
🔹	nome: VARCHAR(255)
🔹	imga: BLOB
🔹	imgb: BLOB
🔹	valimg: LONGTEXT

Figura 14 - Modelo Lógico do Banco de Dados.

A tabela **Pessoa** é responsável por armazenar o nome da pessoa, com a sua impressão digital original, redimensionada e sua respectiva marcação.

3.3.2 – Sistema de Cadastramento

O cadastramento é realizado capturando alguma característica fisiológica ou comportamental, através de um *scanner*, microfone ou câmera. Através desta característica é criado um registro biométrico ou *template* no qual possui as informações quantitativas a serem levadas em consideração num processo de reconhecimento. Essas informações são armazenadas em um banco de dados.

A figura 15 mostra a arquitetura do sistema de cadastramento.

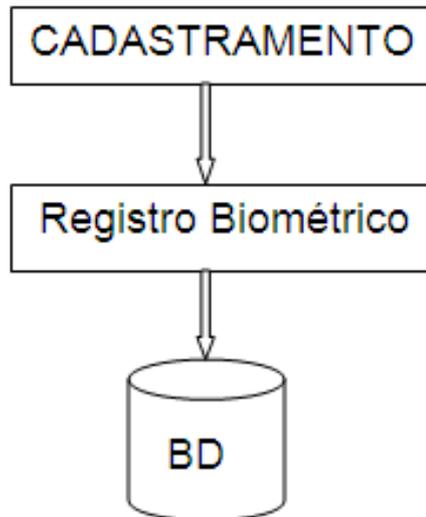


Figura 15 - Arquitetura do Sistema de Cadastro.

3.3.2.1 – Especificação

Para a especificação do sistema de cadastro foi desenvolvido apenas o diagrama de caso de uso. A figura 16 mostra o diagrama de caso de uso de acordo com o modelo lógico do banco de dados.



Figura 16 - Diagrama de Caso de Uso.

A figura 17 mostra o diagrama de classes do sistema, que tem apenas para cadastrar o código, nome, as duas imagens da impressão digital sendo elas: a original, binária, e também sua respectiva marcação.

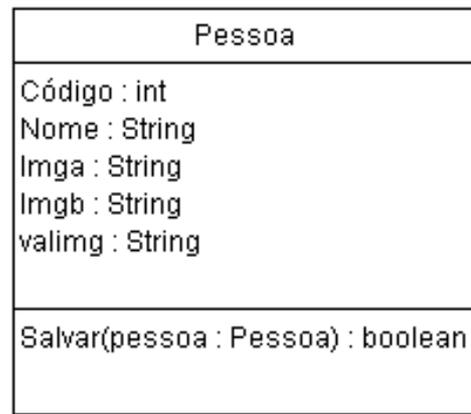


Figura 17 - Diagrama de Classe.

A figura 18 mostra o diagrama de seqüência do sistema de cadastro a partir do diagrama de classe.

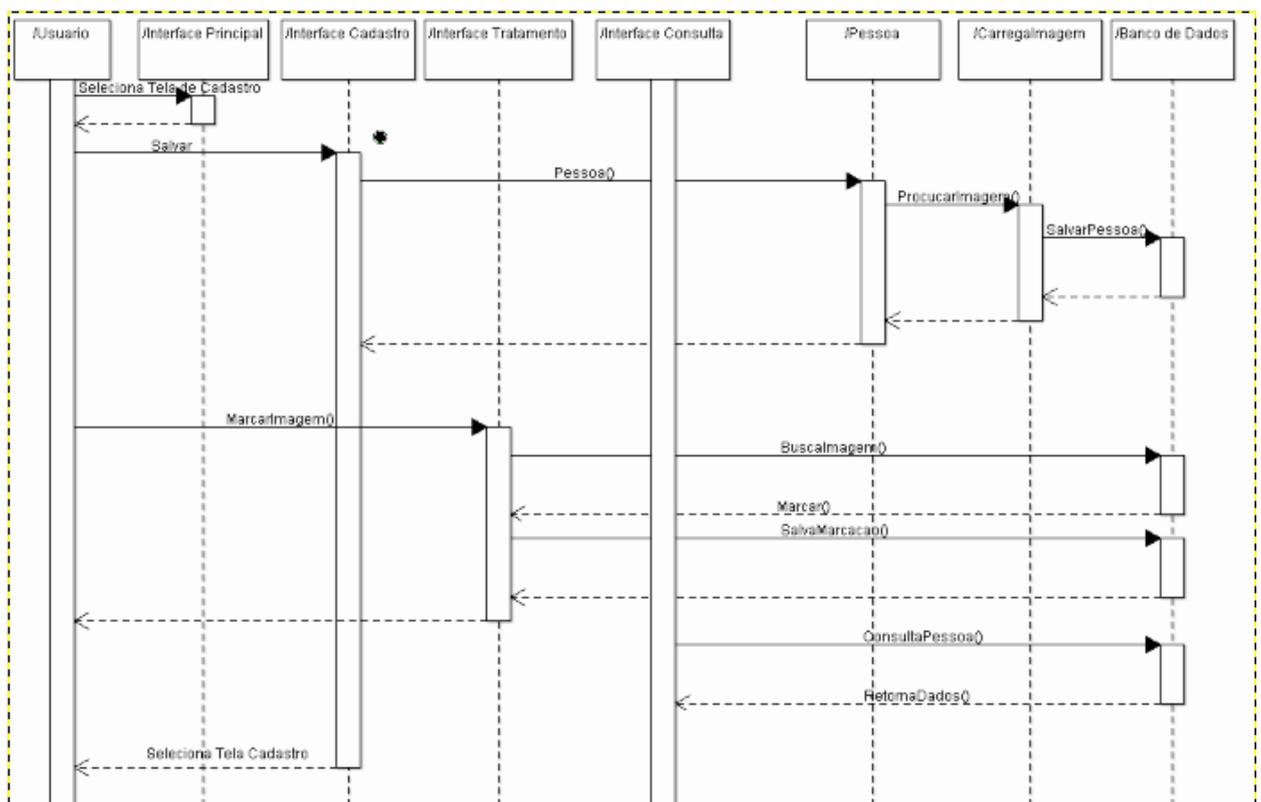


Figura 18 - Diagrama de Sequência.

3.3.2.2 – Implementação do Sistema de Cadastro

Neste módulo serão apresentadas as interfaces do aplicativo para armazenamento das pessoas e suas impressões digitais. A figura 19 mostra a interface principal do aplicativo para cadastro.

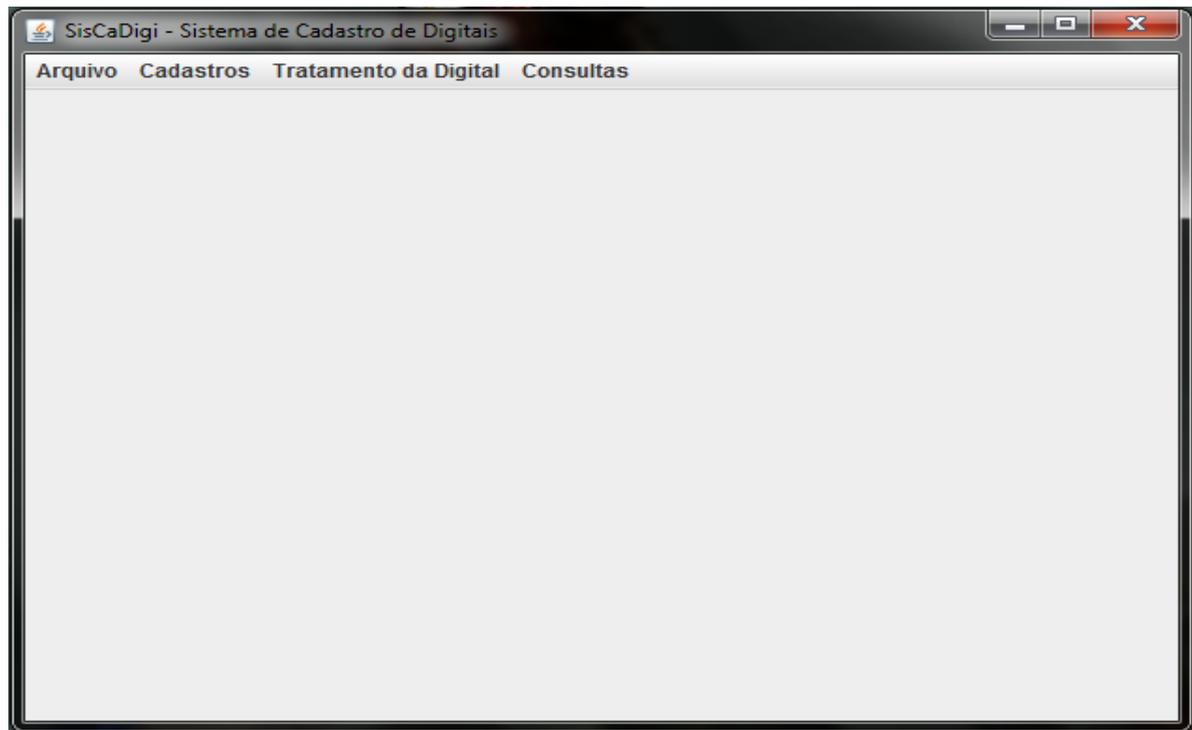


Figura 19 - Interface Principal.

Esta interface é a parte inicial do cadastro, onde o usuário escolhe os itens para cadastrar pessoa, marcar imagem, e fazer as respectivas consultas das pessoas cadastradas.

A Figura 20 mostra a interface onde será feito o cadastramento do indivíduo e suas impressões digitais.



Figura 20 - Interface Para Cadastrar.

A interface de cadastro possui os campos CÓDIGO e NOME que serão completados manualmente, possui também o botão de CARREGAR IMAGEM ORIGINAL e GARREGAR IMAGEM DIMENSIONADA, que importarão as imagens salvas em uma pasta, possui os botões INCUIR, SALVAR e CANCELAR, onde respectivamente fazem a inclusão de uma nova pessoa, salvamento da pessoa no banco de dados e cancelamento do cadastro.

A Figura 21 mostra a interface onde será feito marcação da impressão digital do indivíduo.

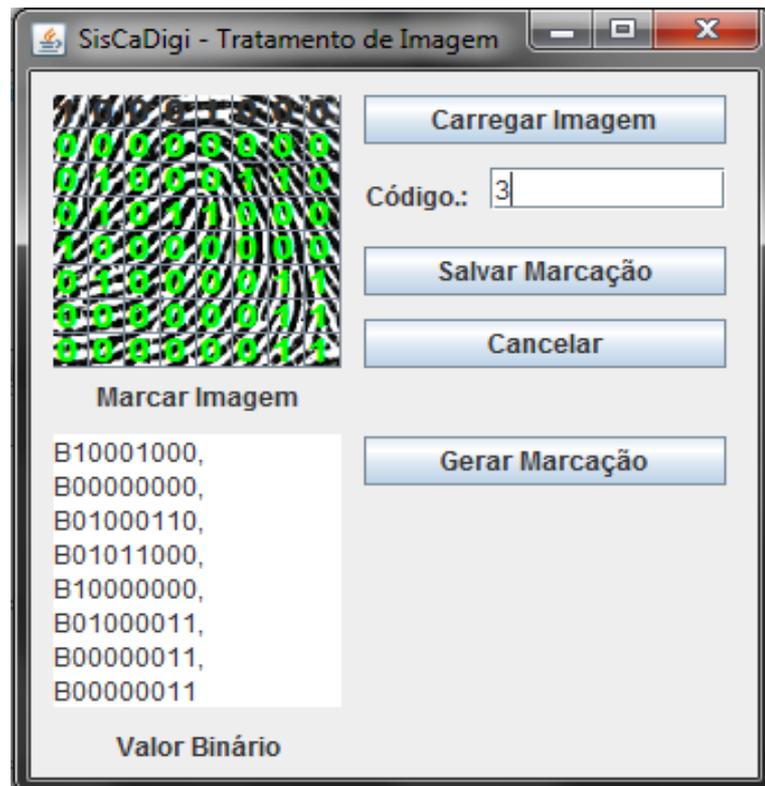


Figura 21 - Interface de tratamento da imagem

A interface de tratamento possui o botão CARREGAR IMAGEM que importara a imagem salva do banco de dados a partir do código. Possui também o botão de GERAR MARCAÇÃO onde gera o código binário a partir da marcação feita manualmente com 0s e 1s. E o botão SALVAR MARCAÇÃO é onde salva o binário gerado, no banco de dados de acordo com código da imagem carregada.

A Figura 22 mostra a interface onde será feito a busca dos cadastros realizados com suas respectivas impressões e marcação.

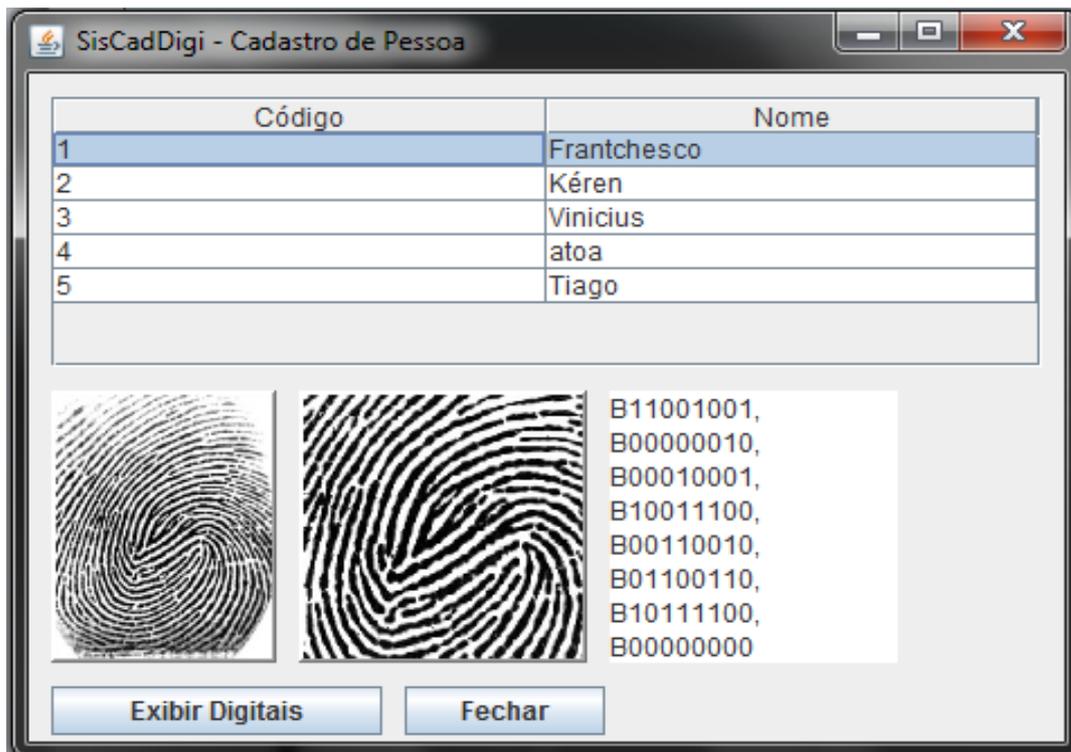


Figura 22 - Interface de Pessoas cadastradas

A interface de Pessoas cadastradas possui o botão EXIBIR DIGITAIS que exibe as imagens e sua marcação a partir do código e nome selecionado.

3.3.3 – Desenvolvimento do Dispositivo de Captura de Sinal Biométrico

Neste módulo será desenvolvido o dispositivo de captura de sinal biométrico. Para isso será necessário desenvolver os circuitos para reconhecimento e impressão em uma matriz 8x8.

3.3.3.1 – Modelagem do dispositivo

O modelo utilizado para o desenvolvimento do circuito para o reconhecimento da impressão digital será adaptada de (McROBERTS, 2010), que utiliza o arduino para realizar a impressão de imagens montadas dentro da programação. A diferença é que no projeto será utilizada uma imagem capturada por um dispositivo de leitura de digitais armazenadas em um banco de dados. Essas imagens serão marcadas de acordo com as técnicas de busca de minúcias e serão dimensionais em uma matriz.

O dispositivo desenvolvido vai capturar o sinal emitido através desta marcação nas imagens de impressões digitais e enviar para o arduíno interpretar esses sinais na matriz de led 8x8. A interpretação do sinal pelo arduíno será feita através de um programa que foi desenvolvido para esta finalidade. Os pontos marcados nas imagens correspondem aos leds acessos na matriz de leds.

3.3.3.1.1 – Material Utilizado

Será feita uma descrição dos materiais utilizados para o desenvolvimento do dispositivo de captura de sinal biométrico.

- **Arduino Program-ME**

O Arduino utilizado será o *Program-ME da GlobalCode* que é um computador de pequeno porte, baseado em uma plataforma de prototipagem eletrônica *open-souce*. Sendo um computador de pequeno porte ele utiliza um microcontrolador (computador completo em um chip) e não um microprocessador tradicional que precisa de memórias e demais recursos externos para funcionar.

Ele pode ser utilizado para controlar diversos tipos de componentes eletrônicos digitais e analógicos permitindo a criação, prototipação e até mesmo produção de hardwares e projetos eletrônicos e invenções em geral. A figura 23 mostra o Arduino modelo Program-Me.

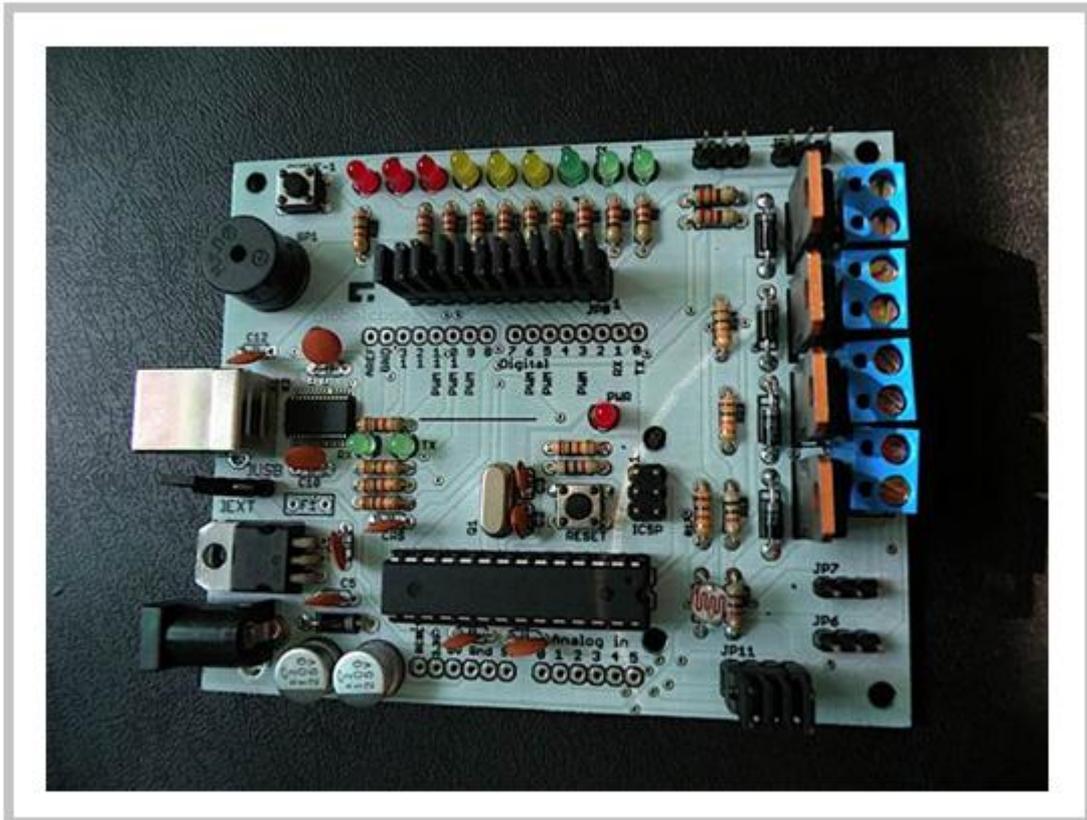


Figura 23 - Arduino Modelo Program-ME (GLOBALCODE).

- **Leitor de Impressão Digital**

O leitor de impressão digital utilizado foi o zk-4000, como mostra a figura 24.



Figura 24 - Leitor zk-4000 .

As especificações do leitor de impressão são:

- Sensor de impressão digital: ótico
- Resolução: 500 DPI / 256 cinza

- Área de detecção: 15 * 18 milímetros
- Tamanho da imagem: 280 * 360pixel
- Interface: USB
- Suporte O/S: Windows XP e Vista, Windows 7 (32 bits)
- Temperatura de Operação: 0 ° -55 ° C / 32-131 ° F
- Umidade de operação: 20-80%
- Cor: preto • Cabo USB: 150 cm • Peso: 0,20 kg
- Dimensões (LxAxP): 65.5 * 49 * 79,8

- **Matriz Led 8x8**

A matriz de led utilizada é o modelo LEDM88RGCC (Red-Green 8x8 Common Cathode Led Matrix Display). Eles são muito utilizados em projeto para imprimir os pontos marcados na impressão digital binária, na sua respectiva linha/coluna. A figura 25 mostra o diagrama interno do circuito da matriz LEDM88RGCC.

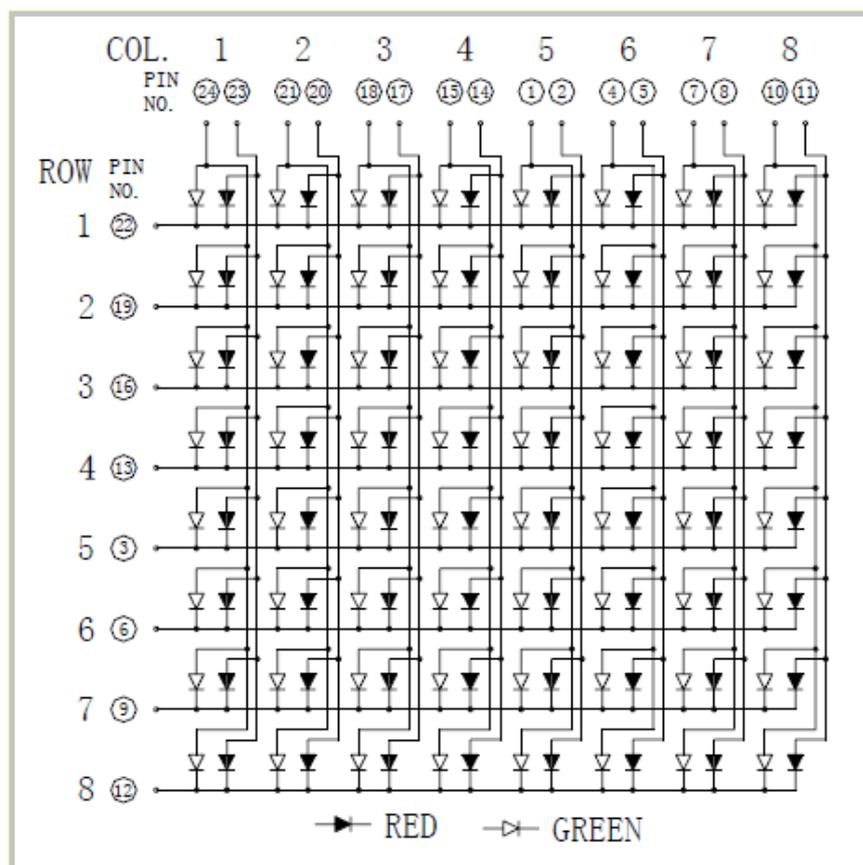


Figura 25 - Diagrama do Circuito (FUTUREC, 2011).

Esta matriz possui 24 pinos sendo 8 para linha e 16 para coluna. Para o desenvolvimento do projeto será utilizado somente os leds verdes da matriz, deixando de ser utilizados os oito pinos da coluna (2, 4, 8, 11, 23, 20, 17 e 14).

3.3.3.1.2 – Integração da matriz com o arduíno

A integração do dispositivo de captura de sinal biométrico com o arduíno será mostrada na figura 26.

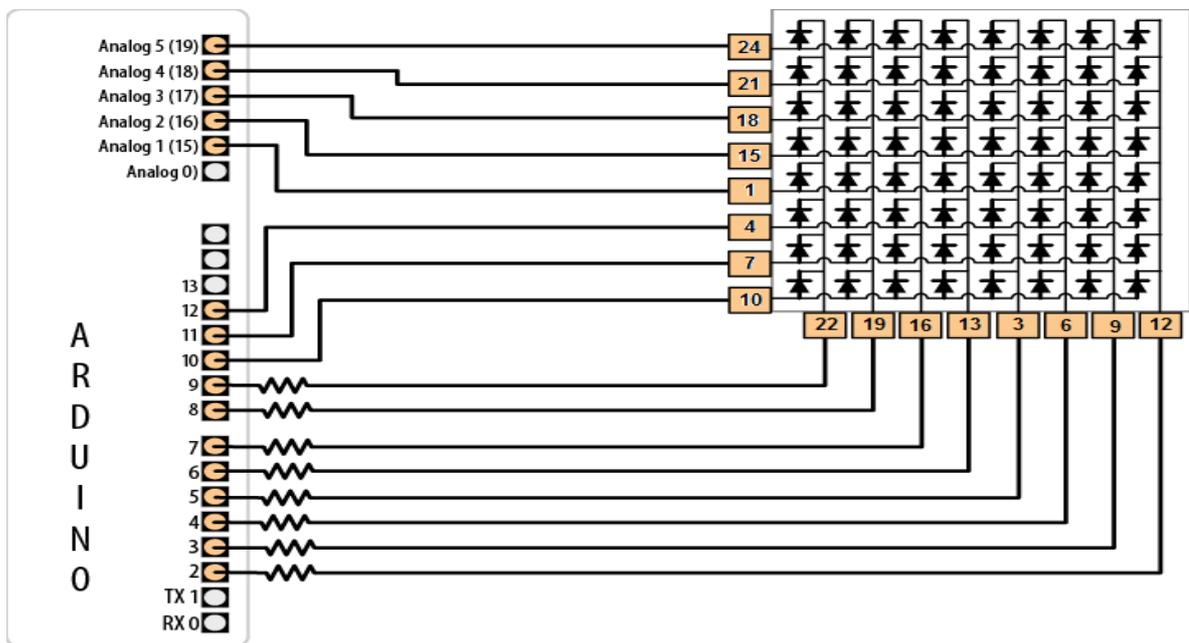


Figura 26 - Integração do Arduino com Placa de Controle de Sinal(adaptado de McROBERTS, 2010).

A comunicação do arduíno com o dispositivo é feita por um programa que foi desenvolvido em C, tendo em vista que a sintaxe da linguagem é baseada na biblioteca *Wiring* (baseado em C e C++). A figura 27 mostra o diagrama de integração do computador com o dispositivo de captura de sinal biométrico utilizando o arduíno.

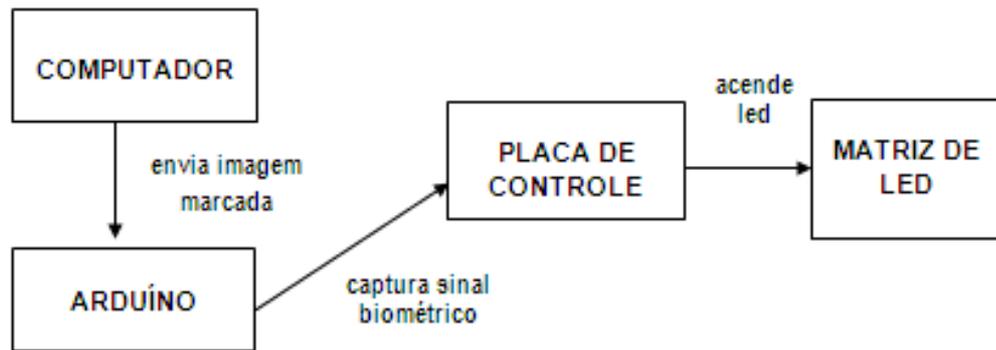


Figura 27- - Diagrama de Comunicação do Computador com o Dispositivo.

A figura 28 mostra o dispositivo montado e em funcionamento. Os leds acesos correspondem aos pontos marcados na imagem da impressão digital. Esses pontos marcados são as minúcias extraídas da imagem original. A localização destas minúcias é que faz com que uma impressão digital seja diferente de uma das outras.

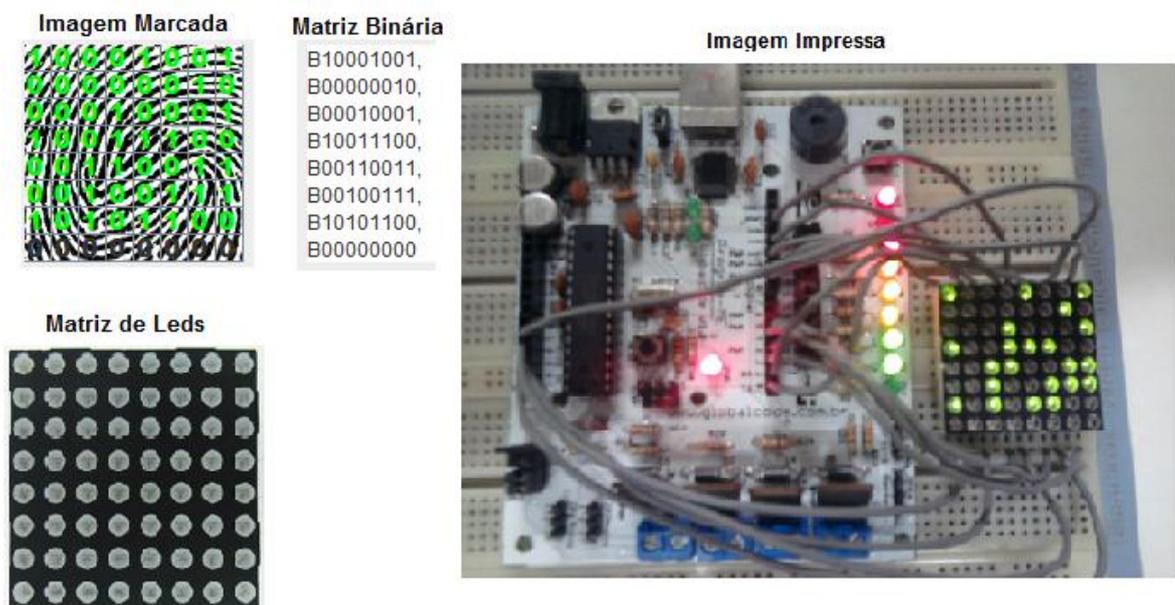


Figura 28 - Dispositivo Montado e Funcionando.

4. CONCLUSÃO

Ao longo do desenvolvimento deste projeto foram feitas muitas descobertas e com isso muito aprendido, aumentando os conhecimentos teóricos e práticos aprendidos na faculdade. A cada dia um novo desafio, ajudando em minha formação e meus conhecimentos.

O desenvolvimento deste projeto trouxe novos alentos profissionais, abrindo um novo mercado de atuação. Apesar do pouquíssimo conhecimento nesta área foi possível desenvolver um dispositivo que pode servir de base para novos projetos, principalmente na área de segurança. A experiência foi muito boa e com a qual adquiri um bom conhecimento para o desenvolvimento de futuros projetos.

5. REFERÊNCIAS

AMORIM, P.R.F., **Biometria**, Universidade Federal de Pernambuco, 2005.

CANEDO, J.A.F., **Terminal de controle de ponto e acesso usando biometria e integrado a web**, Trabalho de Conclusão de Curso, Universidade Federal de Goiás, 2003.

CAPUANO, G.F. e IDOETA, I.V., **Elementos de eletrônica digital**, Editora Erica, 2001.

COSTA, S.M.F., **Classificação e verificação de impressões digitais**, Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo, 2001.

De La ROCHA, F.R., **Curso Introdutório de Microcontroladores – Plataforma Arduíno**, 2010.

ENDEL A, http://www.lojasmartsec.com.br/produto.php?cod_produto=51184.

ENDEL B, **Arduino Language Reference**, <http://arduino.cc/en/Reference/HomePage>, 2008. Acesso em maio de 2011.

ENDEL C, <http://produto.mercadolivre.com.br/MLB-199915725-max7219-driver-de-display-7-segmentos-pic-8051-arduino-avr- JM>. Acesso em setembro de 2011.

ENDEL D, <http://www.ptc.gda.pl/english/products-services/products.catalogue.product/biometric-fingerprint-sensor-usb-zk4000.html>. Acesso em março de 2011.

ENDEL E, <http://www.sparkfun.com/datasheets/Components/General/COM-09622-MAX7219-MAX7221.pdf>. Acesso em setembro de 2011.

ENDEL F, <http://multilogica-shop.com/driver-para-display-de-led-max7219cng>. Acesso em setembro de 2011.

ESPINOSA-DURÓ, Virginia. **Minutiae Detection Algoritmo for FingerPrint Recognition**. IEEE Aerospace and Electronics System Magazine, Vol. 17, 2002.

FARIA, D.R., **Reconhecimento de Impressões Digitais com Baixo Custo Computacional para um Sistema de Controle de Acesso**, Dissertação de Mestrado, UFP, 2005.

FERNANDES, C.C. e LOPES, G.T., **Introdução ao Arduino**, <http://pt.scribd.com/doc/35379935/Introducao-ao-Arduino-Get-Starter-com-arduino> acesso em maio de 2011.

FONSECA, E.G.P. e BEPPU, M.M., **Apostila Arduino**, Universidade Federal Fluminense, Rio de Janeiro, 2010.

FUTURLEC, <http://www.futurlec.com/LED/LEDM88RGCC.shtml>. Acesso em setembro de 2011.

HARRIS, T., **HowStuffWorks - Como Funcionam os Leitores de Impressões Digitais**. <http://informatica.hsw.uol.com.br/leitores-de-impressoes-digitais2.htm>. Acesso em maio de 2011.

JAIN, Anil; DUIN, Robert; MAO, Jianchang. **Statistical Pattern Recognition: A Review**. 2000. 34p. IEEE Transactions on Pattern Analysis and machine Intelligence, Vol. 22, 2000.

KERNIGHAN B.W. e RITCHIE, D.M., **The C programming language**, 2th Edition, 1988.

LEMAY, L. e PERKINS, C.L., **Teach yourself – Java in 21 days**, Sams.net Publishing, 1996.

MAZI, R.C. e DAL PINO JR, A., **Identificação Biométrica Através da Impressão Digital Usando Redes Neurais Artificiais**, Anais do XIV ENCITA, 2009.

POTTS, A. e FRIEDEL JR, D. **Java programming language handbook**, CH., Coriolis Group Books, 2004.

ROMAGNOLI, G.S. **Biometria: Você é sua senha**. In TEMATEC, 61, Ano VIII, Tema 161, 2002, <http://www.serpro.gov.br/publicacao/tematec/2002/ttec61>, Acesso em março 2011.

ROMAGNOLI, GIUSEPPE DOS SANTOS. **Biometria: Você é sua senha.** In TEMATEC, 61, Ano VIII, Tema 161, 2002. Disponível em: <http://www.serpro.gov.br/publicacao/tematec/2002/ttec61>. Acesso em abril de 2011.

SCHILDT, H., **C Completo e Total**, 3ª Edição, Makron Books do Brasil Editora Ltda, 1996.

SILVA, L.M.G., Mini-Curso de Arduíno, 2010. Disponível em WWW.brasilrobotics.blogspot.com. Acesso em maio de 2011.

SILVEIRA , J.A., **O Arduíno**, <http://www.ordemnatural.com.br>, Acesso em maio 2011.

ZURADA, Jackek M., **Introduction to Artificial Neural System** West Publishing Company, 1992 110p.