



Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"

**CESAR HENRIQUE GUADAGNINO VELOSO**

**SISTEMA DE TRAVA POR BIOMETRIA DIGITAL COM ARDUINO.**

Assis  
2012

CESAR HENRIQUE GUADAGNINO VELOSO

## SISTEMA DE TRAVA POR BIOMETRIA DIGITAL COM ARDUINO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do curso de Análise e Desenvolvimento de Sistemas.

Orientador: Prof<sup>o</sup> Esp. Guilherme de Cleve Farto  
Área de Concentração: Informática

Assis  
2012

## FICHA CATALOGRÁFICA

VELOSO, Cesar Henrique Guadagnino

Sistema de trava por biometria digital com arduino / Cesar Henrique Guadagnino Veloso. Fundação Educacional do Município de Assis - FEMA -- Assis, 2012.

49 p

Orientador: Profº. Guilherme de Cleve Farto.

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA.

1.Fechadura biométrica. 2.Arduino. 3.Java.

CDD 001.61  
Biblioteca da FEMA

# SISTEMA DE TRAVA POR BIOMETRIA DIGITAL COM ARDUINO.

CESAR HENRIQUE GUADAGNINO VELOSO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis e Fundação Educacional do Município de Assis, como requisito do curso de graduação, analisado pela seguinte comissão examinadora:

Orientador: Prof<sup>o</sup> Esp. Guilherme de Cleve Farto

Analisador: Prof.<sup>o</sup> Ms. Osmar Aparecido Machado

Assis  
2012

## DEDICATÓRIA

Dedico este trabalho aos meus pais Benedito Cesar Veloso da Silva e Suzi dos Santos Guadagnino e a minha avó Olivia da Silvia, por sempre me apoiarem, incentivarem e valorizarem meus esforços.

## AGRADECIMENTOS

Agradeço ao papai do céu por tudo, muito obrigado!

À minha professora Regina Fumie Etó, pela ajuda segura durante este trabalho. Muito obrigado por todo apoio, paciência, disponibilidade, esforço e conhecimento prestados à minha carreira profissional;

À minha irmã Maria Angélica Guadagnino Veloso por estar sempre junto nesta jornada;

Ao meu amigo Sauro Martins Rebechine , meu amigo Erion Ricardo Barasuol e meu amigo Frederico Manoel Bertoluci Reis , pelas ajudas ;

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

Aos professores do Curso de Análise e Desenvolvimento de Sistemas da FEMA, pelos valiosos ensinamentos durante 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.

“O tempo é muito lento para os que esperam. Muito rápido para os que tem medo. Muito longo para os que lamentam. Muito curto para os que festejam. Mas, para os que amam, o tempo é eterno.”

William Shakespeare

## RESUMO

O tema aqui proposto tem por fundamento compreender o funcionamento de um sistema automatizado utilizando um dispositivo de tranca via Arduino controlado por sistema biométrico. Como fase inicial será montado um protótipo de uma tranca, baseado nos meus conhecimentos de mecânica e eletrônica, enquanto que o software será implementado com base na monografia do ex-aluno Franchesco Nogheira que fora desenvolvida a partir de um estudo sobre sistemas biométricos. Esta aplicação tem por objetivo fazer a segurança em alto nível com uso de um dispositivo biométrico acoplado, de forma que somente o dono ou a pessoa cadastrada e autenticada pelo sistema poderá abrir as trancas da residência ou local em que o projeto for instalado. Objetiva desenvolver um sistema bem simples porém muito seguro assim como de fácil entendimento para o usuário.

**Palavras-chave:** Fechadura biométrica; Arduino; Java.

## ABSTRACT

The subject considered here has for bedding to understand the functioning of an automatized system using a crossbar device saw controlled Arduino for biométrico system. As initial phase an archetype of a crossbar, based on my knowledge of mechanics and eletrônica will be mounted, whereas software will be implemented on the basis of the monograph of the former-pupil Frantchesco Nogheira that are developed from a study on biométricos systems. This application has for objective to make the security in high level with use of a connected biométrico device, of form that the owner or the person registered in cadastre and notarized for the system will only be able to open the crossbars of the residence or place where the project will be installed. Objective to develop a well simple however very safe system as well as of easy agreement for the user.

**.Keywords:** Lock biométrica; Arduino; Java

## LISTA DE ILUSTRAÇÕES

Figura 1	– Fechadura Elétrica.....	13
Figura 2	– Pontos Importantes da Digital.....	16
Figura 3	– Reconhecimento Digital.....	17
Figura 4	– Arquitetura do Hardware Arduino.....	24
Figura 5	– Ambiente Gráfico do Arduíno.....	26
Figura 6	– Arquitetura de Programação no Arduíno.....	28
Figura 7	– Caso de uso do sistema biometrico.....	33
Figura 8	– Caso de uso do administrador.....	34
Figura 9	– Caso de uso do usuario.....	35
Figura 10	– Diagrama de atividade do cadastro da digital.....	36
Figura 11	– Diagrama de sequencia do cadastro.....	37
Figura 12	– Diagrama de URL.....	38
Figura 13	– Modelagem Entidade-Relacionamento.....	38
Figura 14	– Cadastro de biometria e dados do cliente.....	39
Figura 15	– Pagina principal.....	40
Figura 16	– Login e senha.....	41
Figura 17	– Codigo de ligação do banco parte 1.....	42
Figura 18	– Codigo de ligação do banco parte 2.....	43
Figura 19	– Comunicação do arduino com aplicação java.....	44
Figura 20	– Imagem do Arduino Uno.....	45
Figura 21	– Leitor ZK4000.....	45

## SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO.....</b>	<b>11</b>
<b>2.</b>	<b>FECHADURA.....</b>	<b>12</b>
2.1	OPERAÇÃO.....	13
2.2	EQUAÇÕES.....	14
2.3	VANTAGENS.....	15
<b>3.</b>	<b>BIOMETRIA.....</b>	<b>16</b>
3.1	HISTÓRIA DA BIOMETRIA.....	16
3.2	FUNCIONAMENTO DO SISTEMA BIMETRICO.....	18
3.3	TIPOS DE SISTEMAS BIOMETRICOS.....	18
3.3.1	Metodos de captura de impressão digital.....	19
<b>4.</b>	<b>ARDUINO.....</b>	<b>21</b>
4.1	HARDWARE.....	21
4.2	AMBIENTE DE PROGRAMAÇÃO PARA O ARDUINO.....	24
<b>5.</b>	<b>JAVA.....</b>	<b>29</b>
<b>6.</b>	<b>UML.....</b>	<b>31</b>
<b>7.</b>	<b>DESENVOLVIMENTO DO PROJETO .....</b>	<b>33</b>
7.1	DIAGRAMA DE ATIVIDADE.....	36
7.2	DIAGRAMA DE SEGUENCIA.....	37
7.3	DIAGRAMA DE CLASSE .....	38
7.4	TELAS DO SISTEMA.....	39
7.4.1	SISTEMA DE CADASTRO DO USUARIO.....	39
7.4.2	TELA DE LOGIN.....	40
<b>8.</b>	<b>CONCLUSÃO.....</b>	<b>46</b>
	<b>REFERÊNCIAS.....</b>	<b>47</b>

## 1. INTRODUÇÃO

A fechadura magnética depende de alguns dos conceitos básicos do eletromagnetismo. Essencialmente, consiste de um eletroímã atraindo um condutor com uma força grande o suficiente para impedir que a porta seja aberta (SIQUEIRA, 2010).

O sistema biométrico funciona como uma ferramenta de identificação e controlar. Para atingir o objetivo ele primeiramente faz a aquisição do exemplar e através da utilização de um algoritmo que marca os pontos os transforma em binário e transforma o exemplar em um registro digital da característica física ou comportamental capturada (OLIVEIRA, 2009).

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. Nessa placa existem também alguns conectores onde podem ser ligados outros circuitos externos, como sensores, leds, chaves, relés e pequenos motores (SILVEIRA 2011).

A IDE do Arduino é escrito em Java, e pode se comunicar com a porta serial através da biblioteca Java RXTX. Essa biblioteca é muito parecida com a extensão Java Communications API. Internamente a IDE se lembra de qual porta e taxa de transmissão que utilizou pela última vez. Infelizmente essa implementação interna não pode ser considerada uma API pública que você pode utilizar com segurança.

Pelo fato das empresas não possuírem segurança apropriada, concretizamos uma forma prática e segura, que foi uma fechadura biométrica com o sistema Arduino, que gerencia a comunicação da fechadura com o sistema de armazenamento das imagens biométrica salvas no banco.

Com base nessa tecnologia de sistema open source foi feito esse sistema, por ser muito pratico de ser desenvolvido, e as vantagens dela que o uso de uma biometria vai ser mais eficaz, por não ter mais que fazer cópias de chaves e com a tecnologia Java ajuda mais no seu desempenho, por ser uma linguagem sistema open source e não ter custo para ser desenvolvido.

## 2. FECHADURA

A fechadura eletromagnética foi o primeiro método moderno direto. Saphirstein, em 1969, instalou este projeto inicialmente em todas as portas do fórum de Montreal. Pois havia preocupações da autoridade local com incêndios ao trancarem as portas do fórum, gerando solicitações de gestão para encontrar soluções de bloqueio. Para segurança durante incidentes com fogo. Saphirstein propôs inicialmente, o uso de uma pilha linear pelos titulares das portas para funcionar como bloqueio eletromagnético. Estes suportes foram tradicionalmente usados para segurar as portas abertas, mas neste projeto de aplicação, acreditava-se que poderia ser instalado e adaptado para funcionar como bloqueio de segurança contra falhas. Depois do protótipo bem sucedido de instalação no fórum, Saphirstein continuou evoluído e melhorando o design e estabelecendo que a empresa Locknetics desenvolvesse acessório e controle de circuitos para fechadura eletromagnética (SIQUEIRA,210).

Estando em condições difíceis de negócios, a empresa Locknetics foi vendida para a empresa de Hardware Ives e, depois revendida para a empresa Harrow. Muito mais tarde, essa empresa foi novamente vendida a Ingersoll Rand Security Technologies.

Os funcionários que foram associados às atividades em Locknetics formaram outras empresas de produtos eletromagnéticos projetando novos bloqueios, incluindo Dejalock Corporation e Engenharia de Segurança.

Saphirstein continuou a desenvolver tecnologias de bloqueio eletromagnéticos em outras empresas, incluindo Dortranics (mais tarde comprada pela Sag Harbor industriais), Controles Delta (comprado pela Companhia de bloqueio Lovi e depois comprado sistemas Harchett de entrada) e controles Delt Rex porta. Todas localizadas em Connecticut. Outros engenheiros também deixaram essas empresas de manufatura de fechadura eletrônica, incluindo Highpower Segurança Products Lic em Meriden, Connecticut.

Muitas outras empresas nos Estados Unidos, Canadá em todas a Ásia, foram posteriormente estabelecidas para criar oferta de produtos adicionais para a fechadura eletromagnética direta – pull (conforme a figura 1) (DINIZ, 2005).



**Figura 1 – Fechadura Elétrica (In: HDL, 2012)**

## 2.1 OPERAÇÃO

A fechadura magnética depende de alguns dos conceitos básicos do eletromagnetismo. Essencialmente, consiste de um eletroímã atraindo um condutor com uma força grande o suficiente para impedir que a porta seja aberta (SIQUEIRA,

2010). Num exame mais detalhado, o dispositivo foi criado fazendo uso de uma corrente através de um ou mais voltas de fio (conhecido como um solenoide) produz um campo magnético. Isto funciona em espaço livre, mas se o solenoide é enrolado em torno de um ferromagnético núcleo, tais como ferro macio o efeito do campo é grandemente amplificado. Isto é possível porque os domínios internos magnéticas do material de alinhado com o outro pode aumentar consideravelmente a densidade de fluxo magnético (DINIZ, 2005).

## 2.2 EQUAÇÕES

Usando o Projeto de Biot-Savart, pode ser mostrado, que a densidade de fluxo magnético  $B$  induzido por um solenoide de comprimento efetivo  $l$  com uma corrente  $I$  através de  $N$  loops é dada pela equação:

$$B = \frac{\mu_0 \mu_r I N}{l}$$

A força  $F$  entre o electromagneto e a placa de armadura com área de superfície  $S$  expostos no eletroímã é dado pela equação:

$$F = \frac{B^2 S}{2\mu_0}$$

Em ambas as equações,  $\mu_0$  são representadas a permeabilidade do espaço livre e  $\mu_r$  a permeabilidade relativa do núcleo.

Embora o desempenho real de uma fechadura magnética possa variar substancialmente, devido a perdas de vários fatores (tais como fuga de fluxo entre o electromagneto e do condutor), as equações dão uma boa visão de que é necessário para produzir um bloqueio magnético forte. Por exemplo, a força da

fechadura é proporcional ao quadrado da permeabilidade relativa do núcleo magnético. Dada à permeabilidade relativa de um material pode variar de cerca 250 de cobalto para cerca de 5000 de ferro macio e 7000 de silício - ferro . A escolha do núcleo magnético pode, portanto, ter um impacto importante sobre a força de uma fechadura magnética. Também importante é a escolha de comprimento, da corrente em número de laçadas e a eficácia do eletroímã (SIQUEIRA, 2010).

## 2.3 VANTAGENS

- Fácil instalação: fechaduras magnéticas são geralmente mais fáceis de instalar do que outros bloqueios, pois não existem peças de interconexão.
- Rápida operação: fechaduras magnéticas desbloquear instantaneamente quando a energia é cortada, permitindo a liberação rápida em comparação com outros bloqueios.
- Sturdy: é mais fácil fechaduras magnéticas sofrerem menos danos de golpes múltiplos do que as fechaduras convencionais. Se um fecho magnético for forçado a abrir com um pé de cabra, terara pouco ou nenhum dano para a porta ou fechadura (DINIZ, 2005)

Nesse capítulo foi concluído que para ser criado à fechadura magnética foi preciso de muita inovação e faria muitas equações para ela abrir e fechar uma porta; ela sofrerá menos danos que as fechaduras convencionais. Se um fecho magnético for forçado a abrir com um pé de cabra, terá pouco ou nenhum dano para a porta ou fechadura (DINIZ, 2005). E para tornar a fechadura mais completa pode-se adicionar um sistema de abertura por biometria digital.

### 3. BIOMETRIA

A biometria originada do grego Bios (vida) e metron (medida) é o uso de as características biológicas em mecanismo de identificação, como características da retina, da face, da íris, da voz, da geometria da mão, da impressão digital, etc. O uso de características biológicas para identificação é uma grande ideia, pois cada pessoa possui diferentes características um das outras (PINHEIRO, 2008).

O sistema biométrico é basicamente um método de reconhecimento dos padrões responsáveis por conhecer um usuário por meio da determinação de originalidade de uma característica específica comportamental (RIBEIRO, GAIO, 2009).



Figura 2 – Pontos Importantes da Digital (In: KIMALDI ELECTRONICS, 2012)

#### 3.1 HISTÓRICO DA BIOMETRIA

A biometria era utilizada pelos faraós do antigo Egito onde características físicas das pessoas eram usadas para distingui-las. Utilizavam como informação de

identificação: arcada dentária, cor dos olhos, cicatrizes, entre outros. O cientista Francis Galton é considerado um dos fundadores da Biometria. Em 1892, Galton inventou o primeiro sistema moderno de impressão digital, adotado pelos departamentos de polícia em todo o mundo, esta digital era a forma mais confiável de identificação, até a tecnologia do DNA no século XX (RIBEIRO, GAIO, 2009).

Os avanços comerciais na área da biometria começaram na década de setenta, com a utilização de um sistema chamado Identimat que mensurava a forma da mão e olhava principalmente para o tamanho dos dedos (CANEDO, 2010).

Outras técnicas evoluíram através da biometria nos anos 60 e 70, surgiram então o primeiro sistema a analisar o padrão único da retina em meio aos anos 80, época em que o Dr. John Daughman da Universidade de Cambridge abriu caminhos para a tecnologia de íris. Atualmente a verificação de voz possui raízes assentadas nos anos 70; enquanto reconhecimentos faciais e outros eram relativamente novos nas indústrias. A migração para desenvolvimento da pesquisa continua até os dias atuais (RIBEIRO, GAIO, 2009).



**Figura 3 – Reconhecimento Digital (In: SMAAL, 2009)**

## 3.2 FUNCIONAMENTO DO SISTEMA BIOMÉTRICO

O sistema biométrico funciona como uma ferramenta de identificação e controlar. Para atingir o objetivo ele primeiramente faz a aquisição do exemplar e através da utilização de um algoritmo que marca os pontos os transforma em binário e transforma o exemplar em um registro digital da característica física ou comportamental capturada (OLIVEIRA, 2009).

## 3.3 TIPOS DE SISTEMAS BIOMETRICOS

As características biométricas do indivíduo possibilitam as mais diversas técnicas de identificação, pois o ser humano tem características comportamentais e físicas possíveis de medição. Os sistemas biométricos se dividem de acordo com os tipos de características escolhidas para medição. Os tipos normalmente utilizados para identificação de seres humanos de acordo com (VIGLIAZZI, 2006) são:

- Impressão digital – é uma das formas de identificação mais usadas, que consiste na captura da formação de sulcos na pele dos dedos e das palmas das mãos de uma pessoa;
- Veias da palma da mão – é baseado no desenho formado pelas veias da mão;
- Reconhecimento facial – é o sistema mais utilizado para os seres humanos se reconhecerem entre si;
- Íris – é baseada na leitura dos anéis coloridos existentes em torno da pupila;
- Retina – identificação através dos vasos da retina. É uma forma muito segura, pois são únicas em cada olho é muito difícil a modificação e duplicação;
- Geometria das mãos e dedos – é baseada nas medidas das mãos, como forma, tamanho da palma, comprimento e largura dos dedos;
- Reconhecimento de voz – é a combinação de biometrias comportamental e fisiológica. Pois as características se baseiam nas formas e tamanho das

cordas vocais, da boca, dos lábios, da cavidade nasal, ou seja, nos órgãos usados na síntese do som;

- Assinatura – é a forma de assinar de um indivíduo revela sua identidade e vem sendo muito bem aceita nas transações legais e comerciais;
- DNA – possui característica biométrica única para cada indivíduo, por isto é muito utilizado em aplicações forenses;

### **3.3.1 Métodos de captura de impressão digital**

Ha dois métodos para se capturar uma impressão digital: o de tinta no papel (de tinta no papel) e por leitores de IDs. Primeiro método: Posicionar o dedo sobre a tinta e depois pressiona-lo ao papel para que seja escaneado.(TEDESCO, 2003).

A diferença entre uma ID adquirida pelo método de tinta no papel e pelo leitor pode compromete a eficácia do método de captura através de tinta no papel.

O segundo método, por leitores de IDs (figura), utiliza um sistema eletrônico de geração de dados, e por esta razão é muito mais eficiente. O leitor biométrico captura a imagens, das características retiradas da impressão digital que em seguida será processado pelo algoritmo Existem diferentes tipos de leitores de ID, (CANEDO 2003).

- Leitor Ótico: o dedo é colocado sobre uma das faces de um prisma a luz é projetada de outra face e na terceira face uma câmera capta a imagem, as partes do dedo em contato com o vidro causam uma alta diferença de contraste que é detectada pela câmera. Um sistema de lentes para correção de imagem é necessário;
- Leitor Emissor de luz: as papilas do dedo entram em contato com uma pequena camada de um material especial que com o contato emite uma pequena quantidade de fótons que são detectados por um sensor CMOS localizado logo a baixo;
- Leitor Térmico: possui sensores térmicos que detectam o contato das papilas;

- Leitor eletroluminescente: possui fina camada de material eletroluminescente que é excitado com o contato das papilas, a camada de sensores detecta esta luz;
- Leitor Capacitivo: é formado de pequenos capacitores que são descarregados seu nível de carga e medido pelos circuitos lógicos e enviado para um conversor analógico para Digital que transforma essa informação em nível de cinza.

Finalizando esse capítulo podemos ver que biometria tem vários métodos, para poder ver os seus pontos único; conforme o leitor biométrico captura a imagem, das características retiradas da impressão digital que em seguida será processado pelo algoritmo Existem diferentes tipos de leitores de ID, (CANEDO 2003).

## 4. ARDUINO

Nesse capítulo tem objetivo falar os motivos de usar essa placa que é o Arduino, ele irá ajudar a aplicação conversar com a fechadura magnética.

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. 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 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 (SILVEIRA 2011).

O primeiro Arduino foi criado em 2005 no Instituto de Interatividade e Design na Itália, foi criada a partir de uma idéia dos professores David Cuartielles e Massimo Banzi. O objetivo era criar uma ferramenta de hardware única que fosse facilmente programável por qualquer pessoa e também que não fosse muito cara (SILVEIRA 2011).

Arduino é uma plataforma de computação física (são sistemas digitais ligados a sensores e atuadores, que permitem construir sistemas que percebam a realidade e respondem com ações físicas), baseada em uma simples placa de Entrada/Saída micro-controlada e desenvolvida sobre uma biblioteca que simplifica a escrita da programação em C/C++ (FONSECA; BEPPU, 2010).

### 4.1 HARDWARE

O hardware do Arduino é baseado nos microcontroladores AVR da Atmel, em modelos AT mega8, AT mega168, AT mega328 e no AT mega1280. O Arduino recebe um codinome em italiano dependendo do microcontrolador utilizado (francesco).

Para a utilização do Arduino é necessário um bom conhecimento das características básicas de hardware, para que possa explorar ao máximo todos os recursos disponíveis.

O hardware do Arduino é uma placa baseada no microcontrolador ATmega. Tem quatorze pinos de entrada ou saída digital (podendo ser utilizados seis como saídas PWM), seis entradas analógicas, um conector ICSP, controlador USB, um oscilador de cristal 16 MHz, uma tomada de alimentação, e um botão de reset. Para sua utilização basta conectar 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 autores 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);
- Cada um dos 14 pinos digitais pode ser usado como entrada ou saída utilizando 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 (desligado 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 poderão ser utilizados para digital i/o (`digital Read` e `digital Write`) 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 que 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 protetor que protege a porta USB do seu computador contra sobrecarga de corrente e curto circuito. Apesar dos computadores possuírem proteção interna própria o fusível protetor proporciona uma proteção extra. Se mais de 500 mA forem aplicados na porta USB, o fusível irá automaticamente interromper a conexão até que o curto ou a sobrecarga seja removida (NOGUEIRA, 2011).

USB: Usada para gravar os programas; Comunicação serial entre placa e computador; Alimentação da placa (NOGUEIRA, 2011).

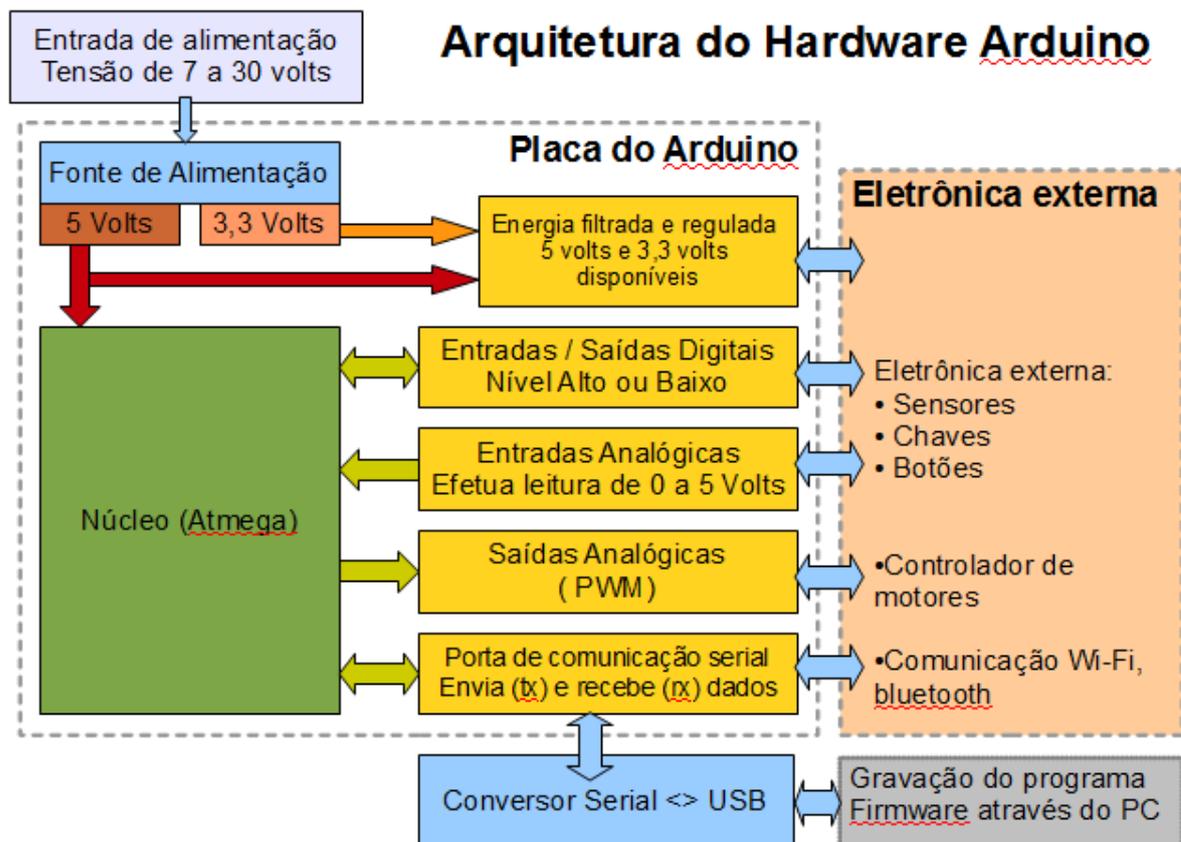
O Arduino contem 6 entradas analógicas, cada uma está ligada a um conversor analógico-digital de 10bits, ou seja, transformam a leitura analógica em um valor dentre 1024 possibilidades. Por padrão, elas medem de 0 a 5 V, embora seja possível mudar o limite superior usando o pino AREF e um pouco de código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

A numeração das portas analógica vai do pino 0 - 5

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

□ ICSP: em circuito Serial Programado

(Na figura 4 esta mostra como funciona a arquitetura do hardware do Arduino)



**Figura 4 – Arquitetura do Hardware Arduino (In:FILHO,2012)**

## 4.2 AMBIENTE DE PROGRAMAÇÃO PARA O ARDUINO

Toda programação é baseada na linguagem 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 a interface amento com outros hardwares.

Os programas para o Arduino são implementados tendo como referência a linguagem C++.

Preservando sua sintaxe clássica na declaração de variáveis, nos operadores, nos ponteiros, nas matrizes, nas estruturas são muitas outras características da linguagem. Com isso temos as referências da linguagem, essas são divididas em

três partes principais: As estruturas, os valores (variáveis e constantes) e as funções. As estruturas de referências da linguagem são:

- Estruturas de controle (if, else, break, ...)
- Sintaxe básica (de ne, include, ; , ...)
- Operadores aritméticos e de comparação (+, -, =, ==, !=, ...)
- Operadores booleanos (, jj, !)
- Acesso a ponteiros (\*, )
- Operadores compostos (++ , {, +=, ...)
- Operadores de bits (j, ; ; ::)
- Os valores de referências são:
- 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)

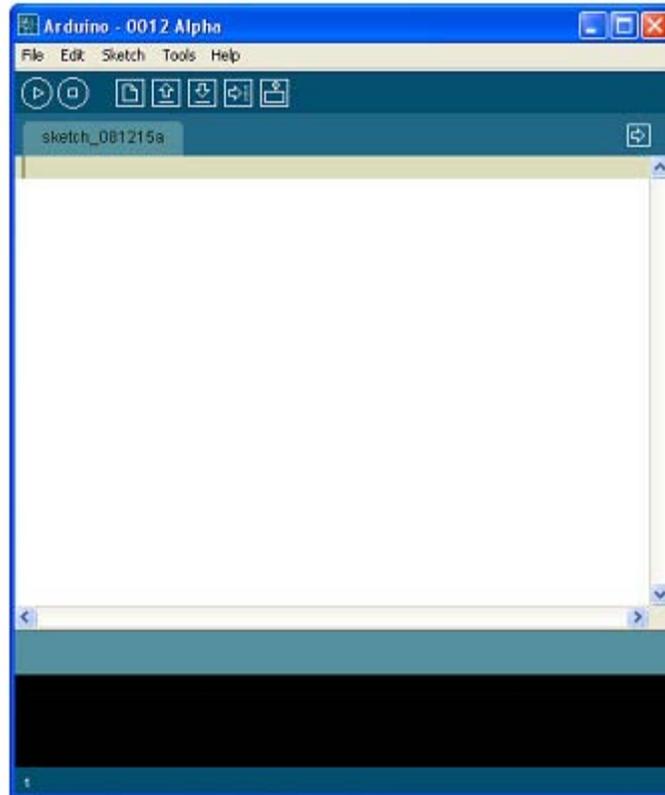
## 8\_E

Bom citar que o software que vem no Arduino já prove varias funções e constantes para facilitar a programação, como:

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

O sistema é compilado efetivamente pelo compilador da GNU (FONSECA; BEPPU, 2010).

A figura 5 mostra o ambiente gráfico do Arduino.



**Figura 5 - Ambiente Gráfico do Arduíno (ENDEL B, 2008).**

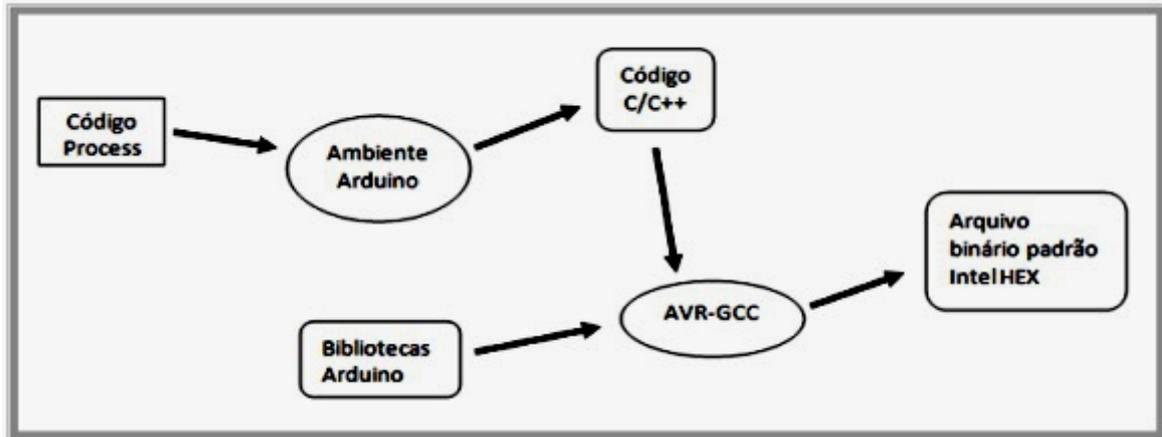
O ambiente é constituído pelo Toolbar, Tab Menu e Menus (File, Edit, Sketch, Tools e Help). A tabela 1 mostra os vários botões com funções distintas.

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

**Tabela 1 - Comandos do Toolbar e Suas Funções (In: NOGUEIRA, 2011).**

O Tab Menu permite gerar documentos com mais de 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 figura 6 mostra a arquitetura de programação em Arduino.



**Figura 6 - Arquitetura de Programação no Arduino.**

Neste ambiente de prototipagem do Arduino são realizadas algumas transformações, no código: “Processe-o resultado é um código” C/C++. No processo GCC é compilado o código C/C++ e junta às bibliotecas para controle dos recursos do microcontrolador, tais como;

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

## 5. JAVA

Em 1991, na Sun Microsystems, o Green Project foi iniciado, o berço do Java uma linguagem de programação orientada a objetos. Os mentores eram Patrick Naughton, Mike Sheridan, e James Gosling. O objetivo do projeto não era a criação de uma nova linguagem de programação, mas antecipar e planejar a “próxima onda” do mundo digital. Acreditavam que em algum tempo haveria uma convergência dos computadores com os equipamentos e eletrodomésticos comumente usados (MATTOS, 2007).

A tecnologia Java teve uma enorme utilização, e logo grandes empresas como a IBM, iriam rodar aplicativos feitos em Java, Estimativas apontam que a tecnologia Java foi a mais rapidamente incorporada na historia da informática. Em 2003 o Java já tinha mais de 4 milhões de desenvolvedores. A ideia inicial do Green Project começou a se concretizar; a linguagem deles passou a ser utilizada em dezenas de produtos diferentes: computadores, celulares, palmtops, e a maioria dos produtos da Apple. Há muitos aplicativos e sites que funcionam somente com o Java instalado, e muitos outros aplicativos e sites são desenvolvidos e disponibilizados com o suporte dessa tecnologia todos os dias. O Java é rápido, seguro e confiável. A tecnologia Java está em todo lugar! Ela pode ser encontrada em laptops, datacenters, consoles de jogo, supercomputadores científicos, telefones celulares e até na Internet (JAVA, 2011).

A mais nova tecnologia de Java é o JavaFX. O JavaFX estende a capacidade do Java permitindo que desenvolvedores usem qualquer biblioteca Java dentro das aplicações JavaFX. Desta forma, desenvolvedores podem expandir seus recursos no Java e aproveitarem a tecnologia de apresentação revolucionária que o JavaFX fornece para criar experiências visuais envolventes (MATTOS, 2007).

A IDE do Arduino é escrito em Java, e pode se comunicar com a porta serial através da biblioteca Java RXTX. Essa biblioteca é muito parecido com à extensão Java Communications API. Internamente a IDE se lembra de qual porta e taxa de transmissão que utilizou pela última vez. Infelizmente essa implementação interna não pode ser considerada uma API pública que você pode utilizar com segurança.

Então você terá que manter suas próprias configurações para lembrar qual porta com seu cartão de Arduino está usando (FONSECA; BEPPU, 2010).

## 6. UML

O UML (Unified Modeling Language) é uma linguagem padrão para elaboração de estrutura de projeto de software (BOOCH; JACOBSON; RUMBAUGH, 2000).

O UML providencia as seguintes particularidades principais [OMG99]:

- Semântica e notação para tratar um grande número de tópicos atuais de modelação.
- Semântica para tratar tópicos de modelação futura, relacionados em particular com a tecnologia de componentes, computação distribuída, frameworks e Internet.
- Mecanismos de extensão de modo a permitir que futuras aproximações e notações de modelação possam continuar a ser desenvolvidas sobre o UML.
- Semântica e sintaxe para facilitar a troca de modelos entre ferramentas distintas.

O UML alarga o âmbito de aplicações alvo comparativamente com outros métodos existentes, designadamente porque permite, por exemplo, a modelação de sistemas concorrentes, distribuídos, para a Web, sistemas de informação geográficos, etc.

A ênfase do UML é na definição de uma linguagem de modelação standard, e, por conseguinte, o UML é independente das linguagens de programação, das ferramentas CASE, bem como dos processos de desenvolvimento. O objectivo do UML é que, dependendo do tipo de projeto, da ferramenta de suporte, ou da organização envolvida, devem ser adaptados diferentes processos/metodologias, mantendo-se, contudo a utilização da mesma linguagem de modelação (BOOCH; JACOBSON; RUMBAUGH, 2000).

O UML é independente das ferramentas de modelação. Apesar da especificação do UML incluir sugestões para os fabricantes de ferramentas adaptarem na apresentação dessas notações (e.g., tópicos como o desenho de diagramas, cor, navegação entre esquemas, etc) (BOOCH; JACOBSON; RUMBAUGH, 2000).

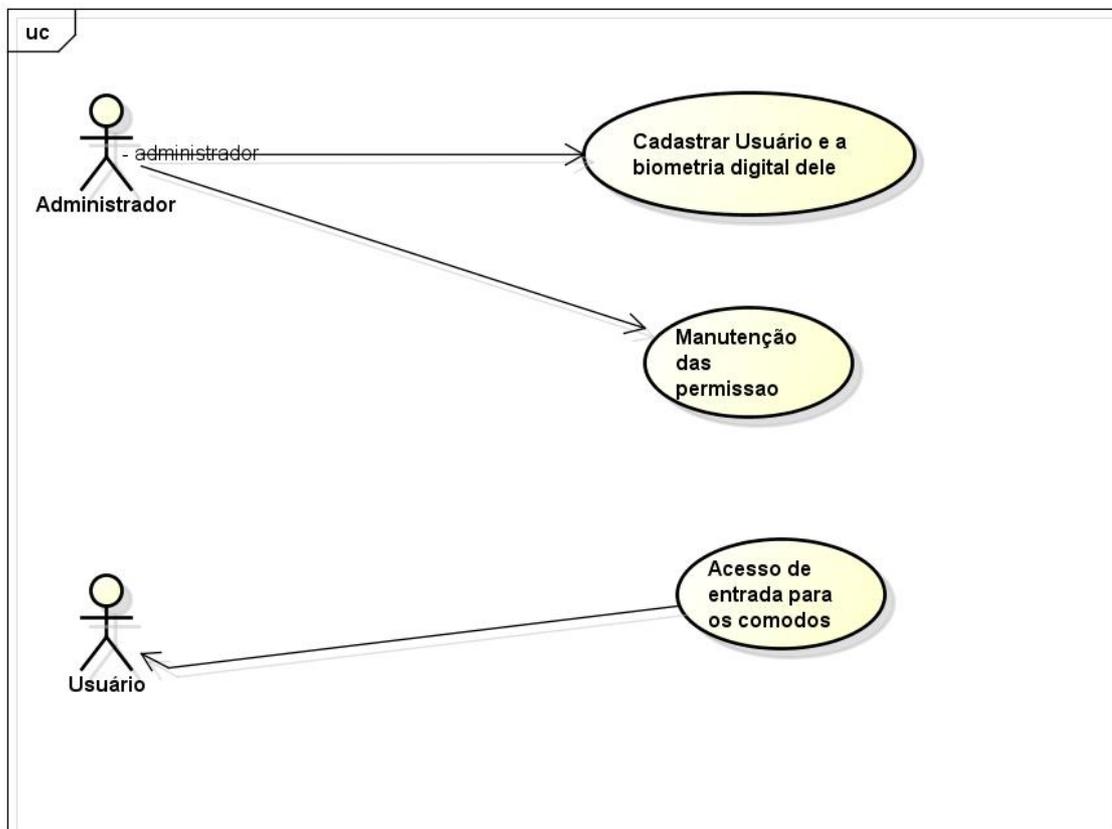
Os diagramas de UML usado nesse trabalho foi:

- Casa de uso
- Atividade
- Sequencia
- URL; e por ultimo o de classe

## 7. DESENVOLVIMENTO DO PROJETO

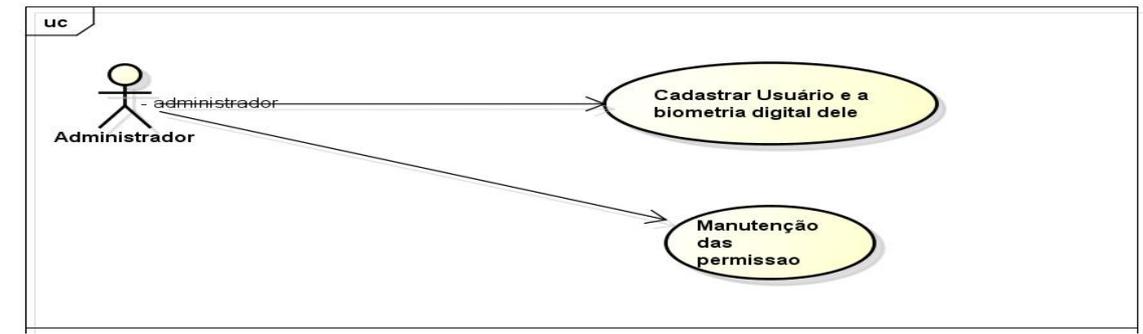
Neste capítulo será apresentada a elaboração do protótipo para reconhecer a impressão digital marcada e salva no computador, utilizando os conceitos obtidos durante a revisão bibliográfica. É apresentada também a descrição e outros fatores do problema, bem como os métodos utilizados para obter a solução.

O desenvolvimento do projeto será dividido em módulos para facilitar a implementação. (A figura 7 mostra os casos de uso geral do sistema)



powered by Astah

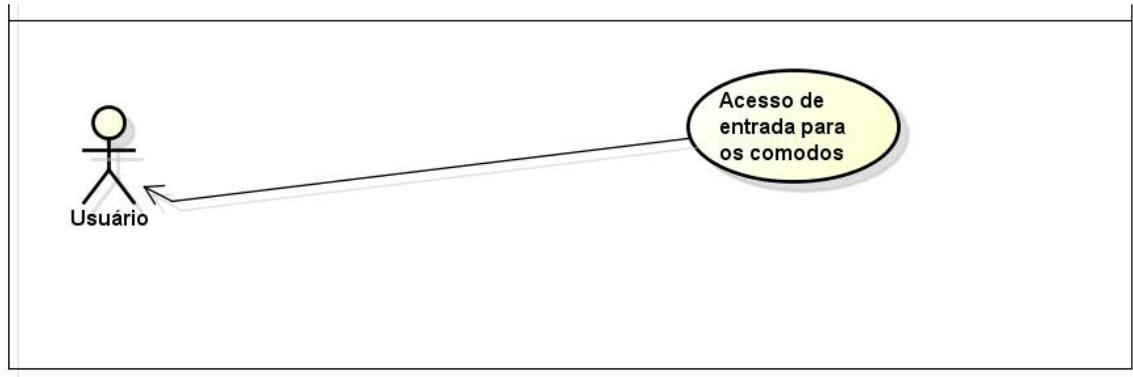
**Figura 7 - Caso de uso do sistema biométrico**



**Figura 8 – Caso de uso do administrador**

Nome do caso uso 1	Cadastrar Clientes
Atores	Administrador
Pré-Condição	Cadastrar a digital e permissão de entrada devido a hora
Cenário Principal	<p>1- O administrado cadastro e faz a captura da biometria do usuário</p> <p>2- O administrador faz a permissão de horário para entrada do funcionário na empresa</p> <p>3- O Administrador configura os relatórios de entrada e saída do funcionário pelo sistema</p>
Cenário alternativo	Caso o sistema não valide o funcionário não terá acesso para acessar dentro da empresa
Caso de teste	<p>1.1 O sistema verifica se o usuário esta cadastrado</p> <p>1.2 O sistema verifica se esta no horário permitido de acesso</p> <p>1.3 O sistema verifica se o usuário foi embora ou esta dentro da empresa</p>

**Tabela 2 – Cadastro de cliente**



**Figura 9 - Caso de uso do usuário**

Nome de caso de uso	Acesso para abrir a porta
Atores	Funcionário ou usuário
Pre-condição	Abrir e fechar a porta
Cenário principal	<ol style="list-style-type: none"> <li>1- O Funcionário informa seus dados e abre a porta</li> <li>2- O usuário tem o seu banco de horas de acesso a entrada dele na empresa</li> </ol>
Cenário alternativo	Caso o sistema não valide a entrada do funcionário ele não ponderar ter acesso dentro da empresa e vai marcar no banco de horas que o usuário (Cesar ) não foi possível entrar na tal Hora
Casos de teste	<ol style="list-style-type: none"> <li>1.1 O Sistema verifica os dados e a digital do funcionário para ele ter acesso na empresa</li> <li>1.2 O sistema grava a hora de entrada e a saída do funcionário no banco de dados</li> </ol>

**Tabela 3 – Liberação de porta**

## 7.1 DIAGRAMA DE ATIVIDADE

Os diagramas de atividades são um dos principais diagramas disponíveis na UML para a modelagem de aspectos dinâmicos de sistema. Ele é essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra (BOOCH; JACOBSON; RUMBAUGH, 2000).

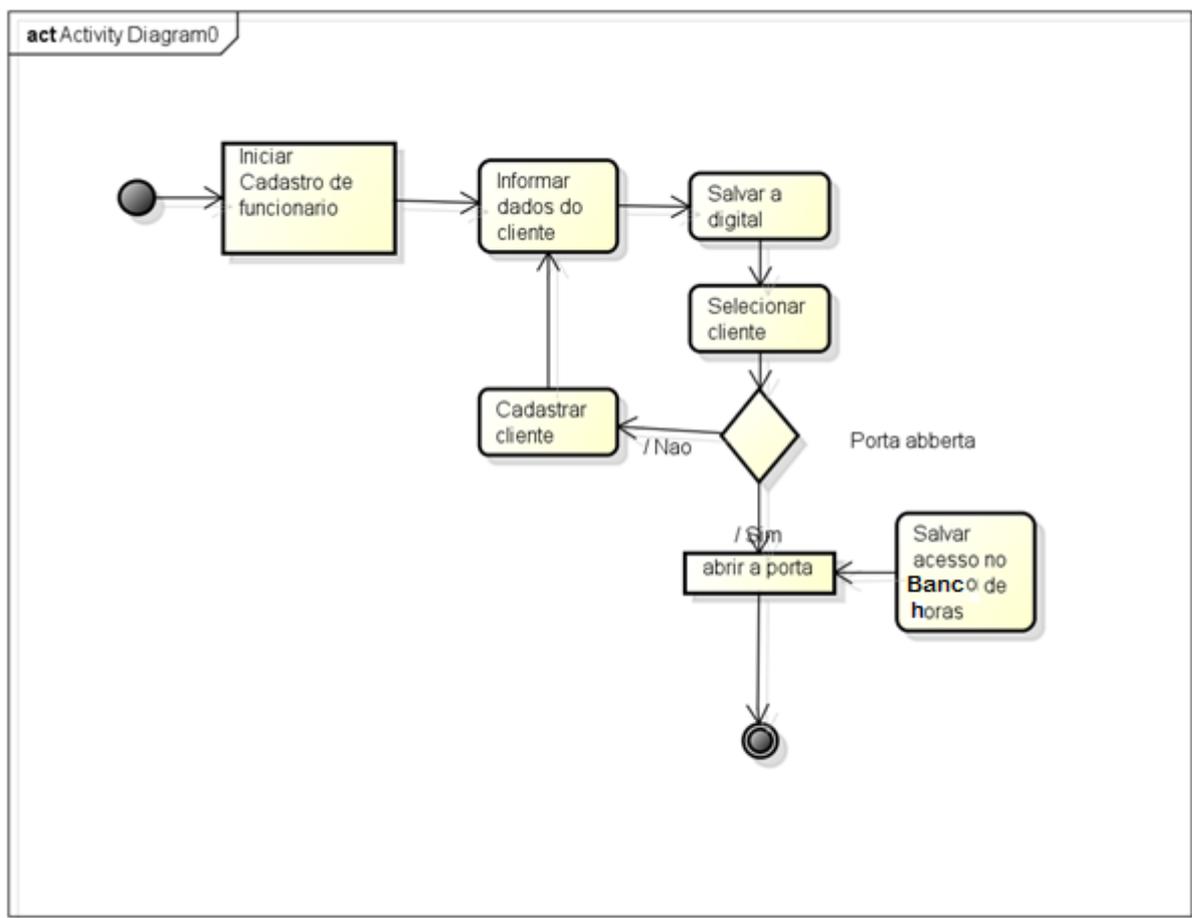
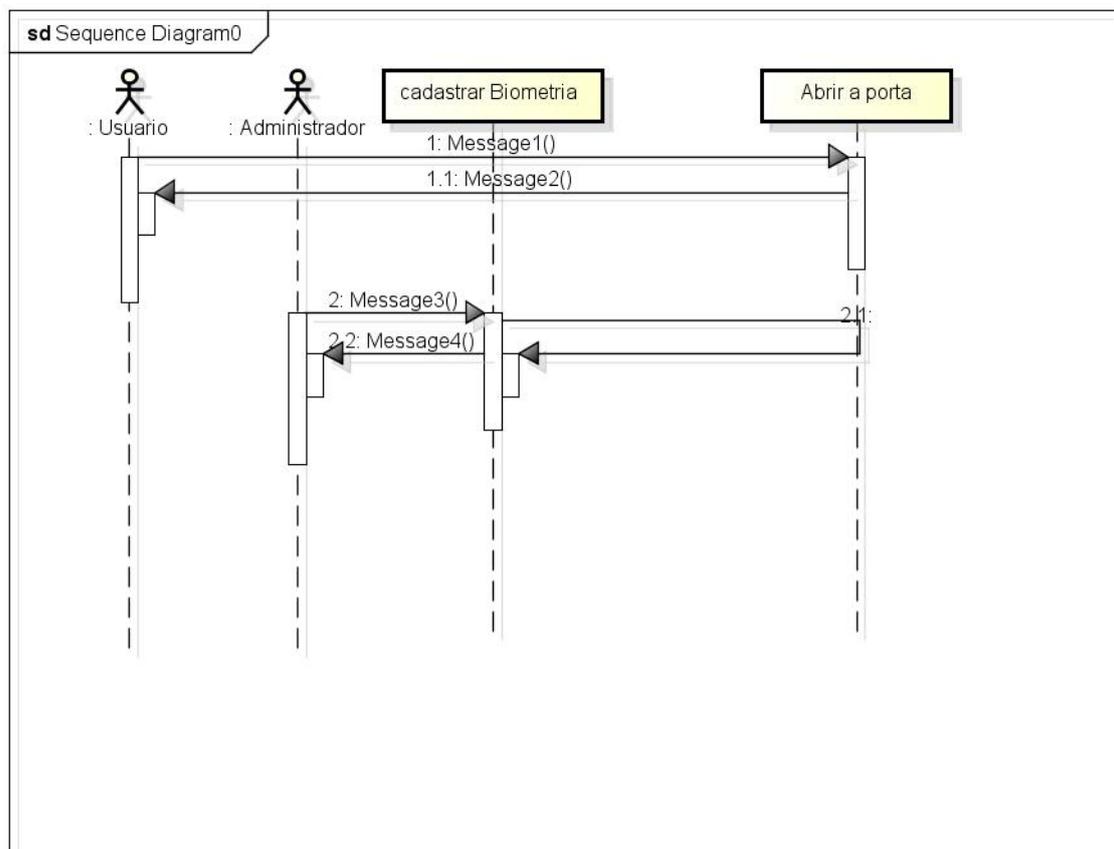


Figura 10 - Diagrama de atividade do cadastro de digital.

## 7.2 DIAGRAMA DE SEQUENCIA

Um diagrama de sequência mostra o fluxo de controle por ordenamento e enfatiza a passagem de mensagens à medida que novos serviços ou transações são implementados (LEE; TEPFENHART, 2001)

Na próxima figura mostrará a sequência que cada tipo de usuário faz no sistema



powered by Astah

**Figura 11 - Diagrama de sequencia – cadastro**

Na próxima figura será mostrado o diagrama de url

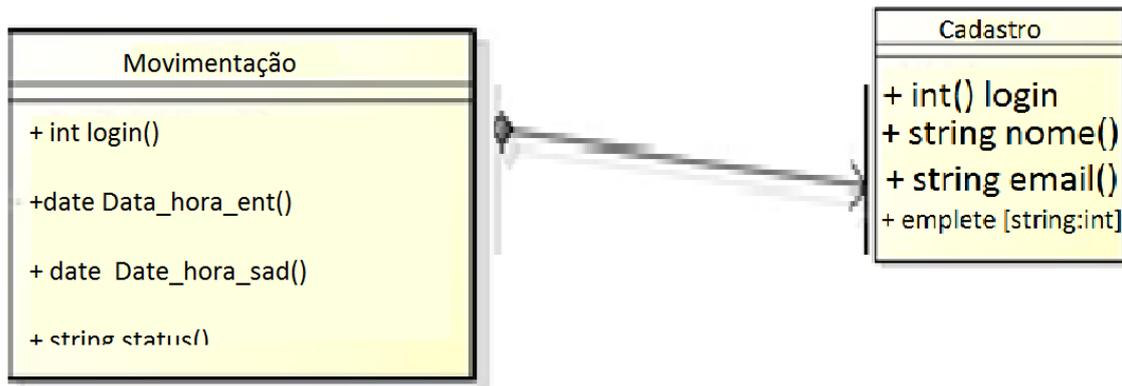


Figura 12 - Diagrama de URL

### 7.3 DIAGRAMA DE CLASSE

O diagrama de classe apresenta uma visão estática do projeto de um sistema. Eles são importantes não só apenas para a visualização, a especificação e a documentação de modelos estruturais, mas também para a construção de sistema executáveis por intermédio de engenharia de produção e reversa (BOOCH; JACOBSON; RUMBAUGH,2000).

Na próxima figura será mostrados as entidades que teve o banco de dados

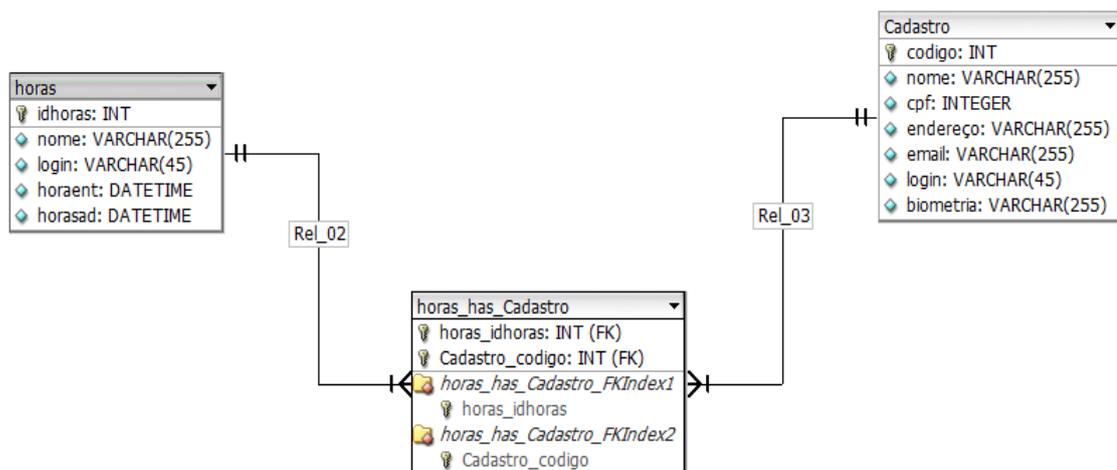
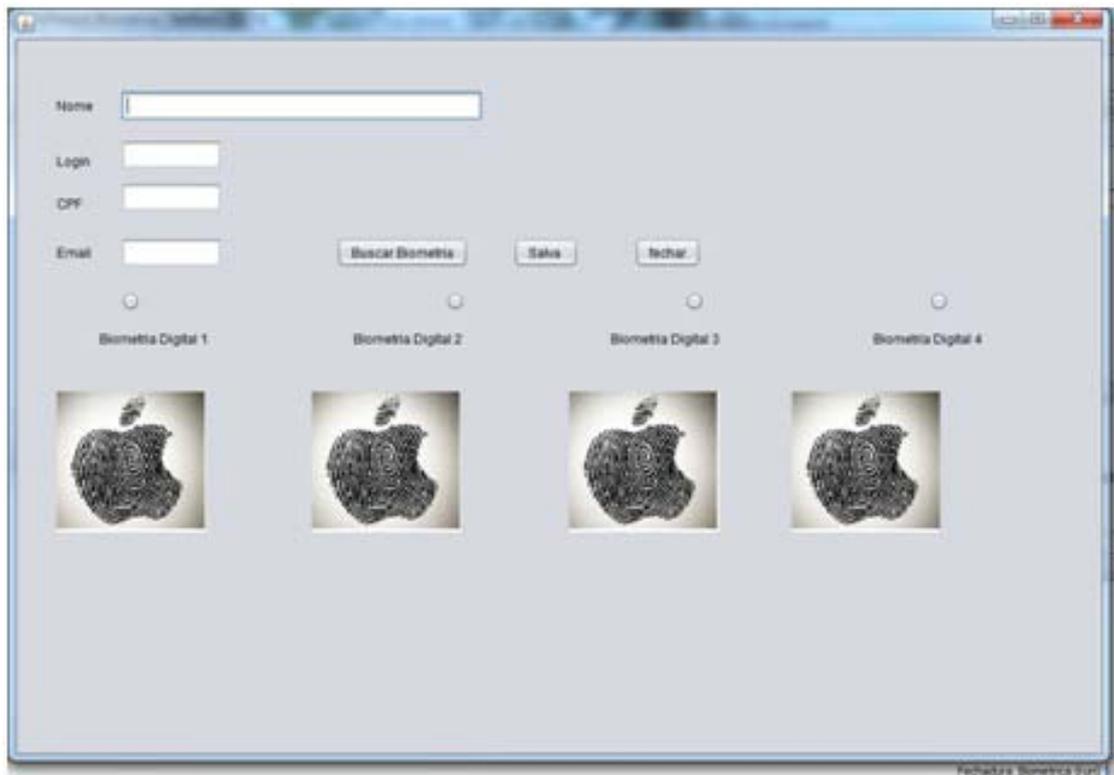


Figura 13 - Modelagem Entidade-Relacionamento

## 7.4 TELAS DO SISTEMA

A imagem a baixo esta representando o cadastro das biometria no banco de dados

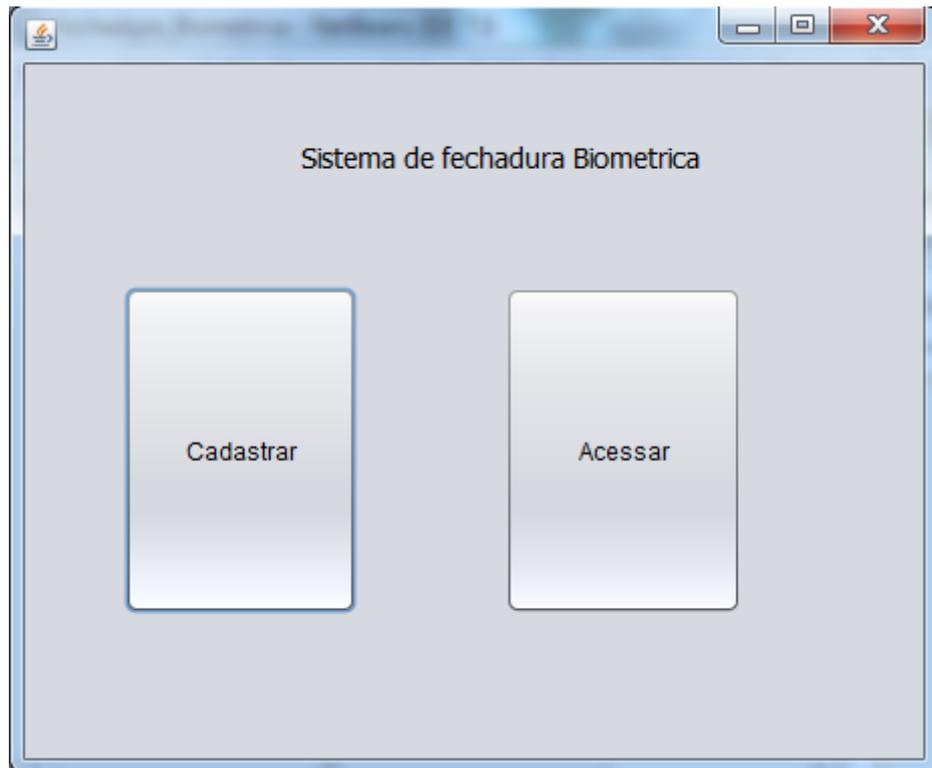


The screenshot shows a web application window with a light gray background. On the left side, there are four input fields stacked vertically, labeled 'Nome', 'Login', 'CPF', and 'Email'. To the right of these fields are three buttons: 'Buscar Biometria', 'Salvar', and 'Fechar'. Below the input fields, there are four radio buttons, each labeled 'Biometria Digital 1', 'Biometria Digital 2', 'Biometria Digital 3', and 'Biometria Digital 4'. Underneath each radio button is a square image of the Apple logo, which appears to be a grayscale or textured version of the logo. The window has a standard Windows-style title bar at the top with minimize, maximize, and close buttons.

**Figura 14 – Cadastro de biometria e dados do cliente**

### 7.4.1 SISTEMA DE CADASTRO DE USUÁRIO

O sistema de cadastro é implementado por meio do Java Configuration, sendo assim o banco do cadastro será manipulado por uma conta de administrador. Seu cadastro terá mais segurança, por que só uma pessoa terá acesso dificultando alguma tentativa de invasão no sistema. O usuário terá que entrar em contato com o administrador para solicitar o cadastramento de um *login* e senha.



**Figura 15 – Pagina principal**

#### 7.4.2 TELA DE LOGIN

Nesta tela o usuário do sistema poderá escolher se quer cadastrar um novo funcionário ou destravar a porta.



**Figura 16 – Login e senha.**

### Login e Senha

Neste local usuário poderá fazer seu login e senha para poder cadastrar um novo usuário

Nas próximas duas imagens será mostrado como foi a ligação do banco mysql com o Java

```

1 package conexao;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.SQLException;
7 import java.util.logging.Level;
8 import java.util.logging.Logger;
9
10 /**
11  *
12  * @author Cesar
13  * @version 1.0
14  */
15 public class GerenciaConexoes {
16
17     // Instância da própria classe estática para emprego de Singleton
18     private static GerenciaConexoes gerenciaConexoes = null;
19     // Objeto que instancia uma conexão
20     private static Connection conn = null;
21     // Objeto que executa instruções SQL
22     private static PreparedStatement stmt = null;
23     // Driver JDBC
24     private static final String DRIVER = "org.mysql.Driver";
25     // IP do servidor da base de dados
26     private static final String IP = "localhost";
27     // URL MySQL
28     private static final String PostgreSQL = "jdbc:mysql://";
29     // Base de dados
30     private static final String database = "fechadura";
31     // Usuário e senha
32     private static final String user = "biometrica";
33     private static final String password = "cesarVeloso";
34
35     /**
36     * Método responsável pelo controle da existência de apenas uma instância
37     * desta classe.
38     *
39     * @return GerenciaConexoes
40     *
41     */
42     public static GerenciaConexoes getInstance() {
43         // Testa se já existe uma instância
44         if (gerenciaConexoes == null) {
45             // Cria a instância
46             gerenciaConexoes = new GerenciaConexoes();
47         }
48         // retorno da função

```

Figura 17 – Código de ligação com o banco parte 1

```

47
48     // retorno da função
49     return gerenciaConexoes;
50 }
51
52 /**
53  * Construtor private, garante que o único método que poderá criar uma nova
54  * instância desta classe será o método getInstance()
55  *
56  */
57 private GerenciaConexoes() {
58 }
59
60 /**
61  * Método responsável por conectar à base de dados
62  *
63  */
64 public void conectar() {
65     // Testa se a conexão já foi iniciada
66     if (conn == null) {
67         try {
68             // Monta a URL de conexão
69             String dbPostgree = PostgreSQL + IP + "/" + database;
70             // Prepara uma instância do Driver JDBC
71             Class.forName(DRIVER);
72             // Cria a conexão
73             conn = DriverManager.getConnection(dbPostgree, user, password);
74             // Auto commit
75             conn.setAutoCommit(true);
76             // Imprime uma mensagem de conexão
77             System.out.println("Conectou!!");
78         } catch (ClassNotFoundException e) {
79             e.printStackTrace();
80         } catch (SQLException e) {
81             e.printStackTrace();
82         }
83     } else {
84         // se já estiver conectado apenas imprime uma mensagem
85         System.out.println(conn + " esta conectado!!!");
86     }
87 }

```

Figura 18 – Código de ligação com o banco parte 2

```

48     // retorno da função
49     return gerenciaConexoes;
50 }
51
52 /**
53  * Construtor private, garante que o único método que poderá criar uma nova
54  * instância desta classe será o método getInstance()
55  *
56  */
57 private GerenciaConexoes() {
58 }
59
60 /**
61  * Método responsável por conectar à base de dados
62  *
63  */
64 public void conectar() {
65     // Testa se a conexão já foi iniciada
66     if (conn == null) {
67         try {
68             // Monta a URL de conexão
69             String dbPostgree = PostgreSQL + IP + "/" + database;
70             // Prepara uma instância do Driver JDBC
71             Class.forName(DRIVER);
72             // Cria a conexão
73             conn = DriverManager.getConnection(dbPostgree, user, password);
74             // Auto commit
75             conn.setAutoCommit(true);
76             // Imprime uma mensagem de conexão
77             System.out.println("Conectou!!");
78         } catch (ClassNotFoundException e) {
79             e.printStackTrace();
80         } catch (SQLException e) {
81             e.printStackTrace();
82         }
83     } else {
84         // se já estiver conectado apenas imprime uma mensagem
85         System.out.println(conn + " esta conectado!!!");
86     }
87 }

```

**Figura 19 – Comunicação do Arduino com aplicação Java**

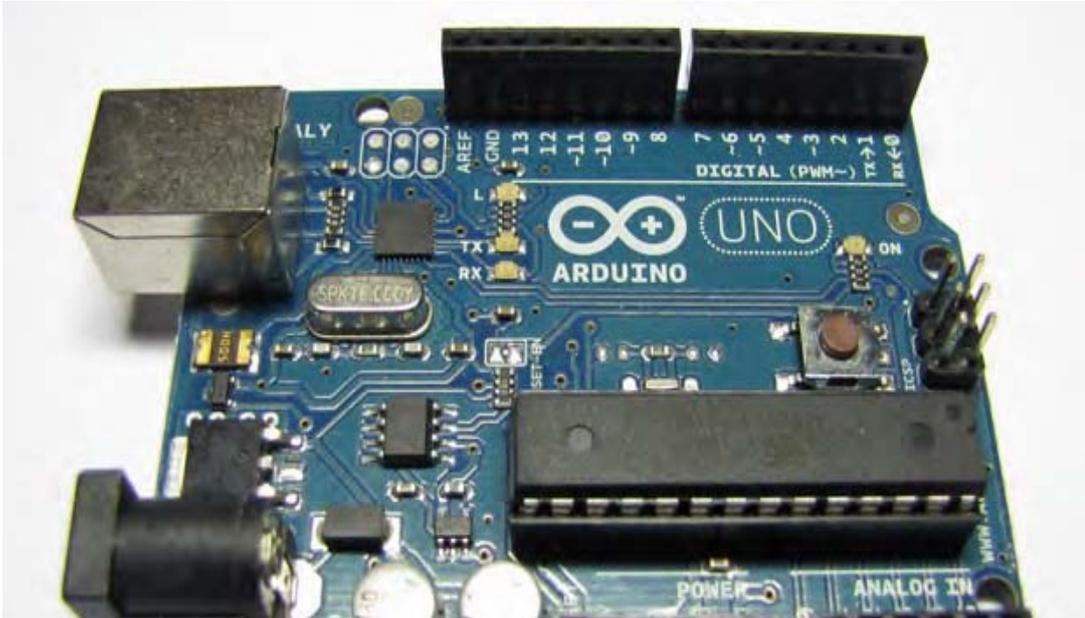
Aqui será criada uma classe onde será declarada a porta serial, fazendo a comunicação do programa do Arduino que ficará encarregado de mandar o sinal da abertura da porta

Será feita uma descrição dos materiais utilizados para o desenvolvimento do sistema de fechadura biométrica

Arduino Uno (REVER MARCADOR)

Para a construção de um sistema vou usar um notebook na onde serão efetuados os códigos de ligação do banco de dados com o leitor na onde se reconhecer o funcionário vai ser mandado para o Arduino à liberação da porta

O Arduino pode ser usado para desenvolver artefatos interativos stand-alone ou conectados ao computador através de Adobe Flash, Processing, Max/MSP, Pure Data ou Supercollider. (FONSECA; BEPPU, 2010 )



**Figura 20 - Imagem do Arduino uno**



**Figura 21 – Leitor ZK4000**

## 8. CONCLUSÃO

Apesar do pouquíssimo conhecimento nesta área foi possível desenvolver um dispositivo que pode servir de base para novos projetos, principalmente na área de segurança. Com a utilização de uma fechadura biometrica somente pessoas realmente autorizadas poderão ter acesso à abertura da fechadura. O objetivo é desenvolver um sistema em java com a ligação do arduino para poder abrir a porta em um novo sistema de segurança e conhecer melhor sobre Arduino e aplicação Java. Ao longo do desenvolvimento deste projeto foi obtido muitos conhecimentos.

Para obter sucesso houve a necessidade de utilizar uma aplicação Java com varias bibliotecas, sendo uma para reconhecer o leitor e outra para o arduino.

Tive grande aprendizado, com novos conceitos e pude criar uma fechadura que poderar fazer as casa e as empresas mais seguras com um menor custo. O desenvolvimento deste projeto trouxe novos conhecimentos profissionais, abrindo um novo mercado de atuação.

## REFERENCIAS

BOOCH, Grady; JACOBSIN, Ivar; RUMBAUGH, James. **UML Guia do Usuário**. 2 ed. Tradução Fabio Freitas da Silva e Cristiana de Amorim Machado. Rio de Janeiro: Elsevier, 2005.

CANEDO, José Alberto. **História da Biometria**. 2012. Disponível em: <<http://www.forumbiometria.com/fundamentos-de-biometria/118-historia-da-biometria.html>>. Acesso em: 23 jun. 2012.

DINIZ, Bruno Leite. **Controle de Acesso**. 2005. 26p. Universidade Federal Fluminense – Niterói-RJ, 2005.

ENDEL B, **Arduino Language Reference**, 2008. Disponível em: <<http://arduino.cc/en/Reference/HomePage,%202008>>. Acesso 25 jun 2012.

FILHO, Daniel O. Basconcello. O Hardware. 2012. Disponível em: <[http://www.robotizando.com.br/curso\\_arduino\\_hardware\\_pg1.php](http://www.robotizando.com.br/curso_arduino_hardware_pg1.php)>. Acesso em: 24 jun 2012.

FONSECA, Erika G. P. da.; BEPPU, Mathyan Motta. **Apostila Arduino**. 2010. 23p. Universidade Federal Fluminense – Niterói-RJ, 2010.

HDL. **Fechaduras Elétricas**. 2012. Disponível em: <<http://www.hdl.com.br/v2/fechadurasDetalhes.php?id=5>>. Acesso em: 24 jun 2012.

JAVA. **O que é a tecnologia Java e por que é necessária?**. 2011. Disponível em: <[http://www.java.com/pt\\_BR/download/faq/whatis\\_java.xml](http://www.java.com/pt_BR/download/faq/whatis_java.xml)>. Acesso em: 26 out 2012.

KIMALDI ELECTRONICS. **Biometria**. 2012. Disponível em: <[http://www.kimaldi.com/kimaldi\\_por/area\\_de\\_conhecimento/biometria/por\\_que\\_a\\_impresao\\_digital](http://www.kimaldi.com/kimaldi_por/area_de_conhecimento/biometria/por_que_a_impresao_digital)>. Acesso em: 24 jun 2012.

LEE, Richard C; TEPFENHART, William M. **UML e C++ Guia prático de desenvolvimento orientado a Objeto**. 1 ed. Tradução de Celso Roberto Paschoa. São Paulo: MAKRON Books, 2001.

MATTOS, Érico C. Tavares de. **Programação de software em Java**. Editora Universo dos Livros: São Paulo, 2007.

NOQUEIRA, Franchesco Rodolfo. **Captura de Sinal Biométrico Utilizando Arduino**. 2011. 60p. FEMA – Fundação Educacional do Município de Assis – Assis, 2011.

OLIVEIRA, Alexandre Nunes de. **Proposta de Utilização da Biometria Aplicada na Segurança das Transações Bancárias**. 2009. 97p. Centro Universitário Feevale – Novo Hamburgo, 2009.

PINHEIRO, José Mauricio. **Biometria nos Sistemas Computacionais – Você é a Senha**. Rio de Janeiro: Ciência Moderna, 2008.

RIBEIRO, Alan de Carvalho; GAIO, Leandro C. **Biometria e seu impacto na Sociedade**. 2009. 11p. Faculdade Presbiteriana Gammon – Lavras, 2009.

SILVEIRA, João Alexandre. **Experimentos com o Arduino**. 1. ed. Editora Ensino Profissional, 2011.

SIQUEIRA, Felipe M.B.L.; IAMAGUTI, Akira S.; SIQUEIRA, Haniel J.B.L.; BRITO, Luis F. R. Fechaura com Modulo Bluetooth. **Revista Eletronica de Educação e Tecnologia do SENAI-SP**, v.4, n.9, outubro, 2010. p. 1-19.

SMAAL, Beatriz. **Os computadores e a emoção**. 2009. Disponível em: <<http://www.tecmundo.com.br/picasa/2872-os-computadores-e-a-emocao.htm>>. Acesso em: 24 jun 2012.

TEDESCO, Cláudia; COUTO, Daniel Lucena. **Desenvolvimento de Algoritmos para Análise de Imagens de IDS Rotacionadas Utilizando Grafos**. Salvador: Artigo. Faculdade Ruy Barbosa, 2003.

VIGLIAZZI, Douglas. **Biometria – Medidas de Segurança**. 2. ed. Florianópolis: Editora Visual Books, 2006.