



**CENTRO UNIVERSITÁRIO DE BRASÍLIA - UniCEUB**

**CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**PROJETO FINAL**

**PAULO HENRIQUE DE OLIVEIRA SILVA**

**SISTEMA DE SEGURANÇA DE TRANCA DE PORTA E CONTROLE  
DE ACESSO**

**Orientadora: MSc. Maria Marony Sousa Farias**

**Brasília**

**2013**

**PAULO HENRIQUE DE OLIVEIRA SILVA**

**SISTEMA DE SEGURANÇA DE TRANCA DE PORTA E CONTROLE DE  
ACESSO**

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

Orientadora: Prof<sup>a</sup>. MSc.  
Maria Marony Sousa  
Farias

Brasília

2013

**PAULO HENRIQUE DE OLIVEIRA SILVA**  
**SISTEMA DE SEGURANÇA DE TRANÇA DE PORTA E CONTROLE DE**  
**ACESSO**

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

Orientadora: Prof<sup>a</sup>. MSc.  
Maria Marony Sousa Farias

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

---

**Prof. Abiezer Amarilia Fernandes**  
**Coordenador do Curso**

**Banca Examinadora:**

---

Prof<sup>a</sup>. Mestre Maria Marony Sousa Farias  
Orientadora

---

Prof. Mestre Luciano Henrique Duque

---

Prof. Doutor William Oliveira Bessa

---

Prof. João Marcos Souza Costa

## DEDICATÓRIA

*Aos meus pais que me proporcionaram uma educação de qualidade; aos meus amigos e familiares que me apoiaram em mais uma etapa.*

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, por me permitir todas as oportunidades. Aos meus pais, que com muito esforço puderam me dar uma educação digna apoiando e investindo no meu futuro. A minha família pelo apoio e pela boa convivência durante esses 22 anos de vida. Aos meus amigos que dividiram durante os 5 anos de curso todas as provas, trabalhos e contratempos que o curso de Engenharia de Computação proporciona. Aos professores que através dos seus conhecimentos me ensinaram a teoria e experiências de suas carreiras profissionais. Em especial ao amigo José Carlos e aos professores Maria Marony e Francisco Javier, pela orientação sempre que necessário.

A todos um sincero agradecimento, sem vocês eu não conseguiria obter êxito nessa etapa tão importante da minha vida.

*“A alegria está na luta, na tentativa, no sofrimento envolvido e não na vitória propriamente dita.”*

## SUMÁRIO

DEDICATÓRIA .....	iv
AGRADECIMENTOS.....	v
SUMÁRIO.....	vii
LISTA DE FIGURAS.....	x
LISTA DE QUADROS E TABELAS .....	xii
LISTA DE SIGLAS E ABREVIATURAS.....	xiii
RESUMO .....	xiv
ABSTRACT .....	xv
CAPÍTULO 1 - INTRODUÇÃO .....	16
1.1 – Apresentação do Problema.....	16
1.2 – Motivação.....	16
1.3 – Objetivos .....	17
1.4 – Importância .....	17
1.5 – Escopo .....	18
1.6 – Estrutura.....	18
CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA .....	20
2.1 – Automação Residencial .....	20
2.2 – Benefícios da Automação .....	21
2.3 – Histórico .....	22
2.4 – Benefícios do Dispositivo Proposto e suas Restrições .....	24
CAPÍTULO 3 – REFERENCIAL TEÓRICO .....	26
3.1 – Microcontroladores.....	26
3.1.1 – Arduino.....	28
3.2 – Eletromagnetismo .....	28

3.2.1 – Relés .....	30
3.3 – Linguagens de Programação .....	32
3.3.1 – Linguagem C .....	32
3.4 – Sensor de Som .....	34
3.5 – Visão do Projeto .....	35
CAPÍTULO 4 – DESCRIÇÃO DO HARDWARE E DO SOFTWARE ...	36
4.1 - Arduino .....	36
4.1.1 – Especificação .....	37
4.1.2 – Pinagem .....	39
4.2 – Arduino IDE .....	40
4.3 – Relé .....	41
4.3.1 – Especificações .....	42
4.3.2 – Funcionamento .....	43
4.4 – Sensor de Som .....	43
4.4.1 – Especificações .....	44
CAPÍTULO 5 – DESENVOLVIMENTO E RESULTADOS OBTIDOS ..	45
5.1 – Modelagem do Sistema .....	45
5.2 – Fluxograma Geral do Dispositivo .....	46
5.2 – Escrita do Código Fonte .....	47
5.2.1 – Compilação e Gravação do Código Fonte .....	51
5.3 – Elaboração do Protótipo e dos Circuitos .....	52
5.4 – Testes e simulações .....	55
5.5 – Problemas Encontrados .....	59
5.6 – Orçamento do Projeto .....	60
5.7 – Protótipo Final .....	61
CAPÍTULO 6 – CONSIDERAÇÕES FINAIS .....	62

6.1 – Conclusões .....	62
6.2 – Propostas para Trabalhos Futuros .....	62
REFERÊNCIAS BIBLIOGRÁFICAS .....	64
APÊNDICE A – CÓDIGO FONTE DO PROJETO .....	68

## LISTA DE FIGURAS

Figura 2.1- Automação Residencial e Automação Industrial .....	23
Figura 3.1– Esquema Interno de um Microcontrolador.....	27
Figura 3.2 – Motor Elétrico .....	29
Figura 3.3 – Relés Comerciais .....	31
Figura 3.4 – Relés Industriais .....	31
Figura 4.1– Arduino Duemilanove .....	36
Figura 4.2.a) – Pinos Digitais.....	39
Figura 4.2.b) – Pinos Analógicos .....	39
Figura 4.2.c) – Pinos de Alimentação .....	40
Figura 4.3 – Relé 5V .....	41
Figura 4.4.a) – Especificações Relé 5V.....	42
Figura 4.4.b) – Especificações Relé 5V .....	42
Figura 4.5 – Sensor de Som .....	44
Figura 4.6 – Esquema do Sensor de Som.....	44
Figura 5.1 – Componentes do Projeto .....	46
Figura 5.2 – Fluxograma Geral do Dispositivo.....	47
Figura 5.3 – Código Fonte Diretivas Pré-Compliação.....	48
Figura 5.4 – Código Fonte Inicial .....	49
Figura 5.5.a) – Código Fonte Função Parte I .....	50
Figura 5.5.b) – Código Fonte Função Parte II .....	51
Figura 5.6 – Teste de Compilação.....	51
Figura 5.7 – Circuito LEDs.....	52
Figura 5.8 – Circuito Push Button.....	53

Figura 5.9 – Circuito Relé + Fecho +VCC .....	54
Figura 5.10 – Circuito Final.....	54
Figura 5.11 – Protótipo .....	55
Figura 5.12 – Teste LEDs.....	56
Figura 5.13– Teste Push Button .....	57
Figura 5.14 – Teste Sensor de Som .....	58
Figura 5.15 – Protótipo Final .....	61

## LISTA DE QUADROS E TABELAS

Quadro 4.1-Especificações Arduino Duemilanove .....	37
Quadro 5.1– Custo do Projeto.....	60

## LISTA DE SIGLAS E ABREVIATURAS

$\Omega$  - Ohm

mA – Miliampere

V – Volts

AR – Automação Residencial

DAC – Conversor Analógico Digital

DC – Corrente Contínua

EEPROM - Erasable Electronically Programmable Read Only Memory

GND - Terra

IDE – Ambiente de Desenvolvimento Integrado

I/O – Entrada / Saída

ICSP - In-Circuit Serial Programming

LED – Diodo Emissor de Luz

PLC – Power Line Communication

SRAM - Static Random Access Memory

## RESUMO

Este trabalho apresenta um dispositivo de controle de acesso em uma porta baseando-se nos intervalos dos sons de batidas na porta. O projeto integra um Arduino Duemilanove com o microcontrolador ATmega328, um Fecho Eletromagnético da marca HDL, um relé 5V para acionar o fecho e um botão responsável pela mudança das batidas sem ter que mexer no código-fonte. O controle do circuito é feito utilizando-se da linguagem de programação C, onde o dispositivo permite a configuração dos intervalos entre as batidas utilizando o *timer* nativo do microcontrolador. A abertura/bloqueio do fecho eletromagnético será realizada a partir de um sinal enviado pelo Arduino para o relé, que funcionará como um interruptor liga/desliga. O protótipo é composto por uma porta de madeira contendo um fecho e uma lingueta, uma caixa contendo o circuito montado e uma fonte de alimentação.

Palavras chave: Domótica, Microcontrolador, Automação, Arduino, Relé, Sensor de Som.

## ABSTRACT

This paper presents a device access control on a door based on the intervals of the sounds of knocking at the door. The project integrates an Arduino Duemilanove with ATmega328 microcotrolador a closing Electromagnetic brand HDL, 5V relay to trigger a button closure and responsible for the change of beats without touching the source code. The control circuit is done using the C programming language, which allows the device to configure the intervals between beats using the timer native microcontroller. The opening / closing of the electromagnetic lock will be held from a signal sent from the Arduino to the relay, which will act as an on / off switch. The prototype consists of a wood door having a latch and a pawl, a box containing the assembled circuit and a power supply.

Keywords: Home Automation, Microcontroller, Automation, Arduino, Relay, Sensor Sound

# CAPÍTULO 1 – INTRODUÇÃO

## 1.1 – Apresentação do Problema

A automação residencial (AR), também conhecida como Domótica, tem mostrado cada vez mais que a integração dos dispositivos eletrônicos e microcontrolados com elementos do nosso cotidiano aumentam consideravelmente os benefícios quando comparados com situações sem eles. Esse conceito deriva-se da necessidade do homem em querer realizar o mínimo de esforço possível durante algumas atividades. Quando se fala em modernidade, é normal a associação com conforto, pois tudo ligado à tecnologia basicamente está relacionado à melhoria da qualidade de vida. O conteúdo desse projeto detalha o desenvolvimento de um sistema aplicado à segurança, uma parte importante da Domótica.

## 1.2 – Motivação

O propósito da realização desse projeto surgiu a partir da leitura do artigo “Desmistificando a Domótica” de Caio Augusto Morais Bolzani e do projeto “Secret Knock Detecting Door Lock” do Grathio, disponível no sítio ([www.instructables.com/id/Secret-Knock-Detecting-Door-Lock/](http://www.instructables.com/id/Secret-Knock-Detecting-Door-Lock/)).

Além disso, motivado pelo crescente aumento de violência ocorrida nos últimos anos, principalmente as relacionadas à invasão de residências e locais restritos, o desenvolvimento de uma forma alternativa aos padrões convencionais de tranca de porta surge como uma possível solução. Minimizar as preocupações geradas quando as pessoas saem e não sabem o que pode acontecer durante sua ausência é a grande aposta desse projeto.

Levando esse problema de falta de segurança à outra esfera sem ser a residencial, inúmeras são as notícias de invasão ou vazamento de provas no Brasil. Em 2009 um acontecimento que teve repercussão nacional foi o vazamento de informações sigilosas sobre o Exame Nacional do Ensino Médio (ENEM), devido a falhas no processo de segurança. Na época, foi alegado que

um funcionário não autorizado entrou na sala onde se armazenavam as provas e se apossou de uma delas junto com o gabarito. Já em 2010, a polícia federal prendeu um grupo de fraudadores de concursos públicos que tiveram acesso ilegal à provas do concurso de Agente da Polícia Federal e da segunda etapa do 3º Exame da Ordem da OAB.

Analisando esses dois casos, questionamentos como: “Será que se houvesse maior segurança isso teria conhecido? Será que se houve um sistema de controle de acesso à abertura das portas minimizaria os prejuízos?”

### **1.3 – Objetivos**

O objetivo geral desse projeto é desenvolver um sistema capaz de fornecer maior segurança quanto à forma de acesso a um determinado local. Para obter o êxito nesse propósito, objetivos específicos foram traçados:

- Criar um sistema que automatize a forma de acesso a um local através do desenvolvimento de uma solução integrada em hardware e software.
- A base do sistema será a placa de prototipagem Arduino.
- O sistema irá detectar as batidas utilizando um sensor analógico de som. Esse sensor utiliza um microfone de eletreto para captar a intensidade sonora e amplificar o som.
- 

### **1.4 – Importância**

Seguindo a forte tendência de casas inteligentes, a AR pode ser vista cada vez mais em casas e apartamentos modernos. Seja através da integração entre áudio e vídeo, medidas de segurança ou opções que prezam pelo conforto, o fato é que não existe nada melhor do que contar com uma ajuda da tecnologia para melhorar a qualidade de vida.

Hideki Hattori, um executivo da Nobel Home, empresa especializada no ramo, explicou que a automação residencial é a integração entre diversos

equipamentos motorizados e automatizados interagindo entre si. A partir do momento em que você tem uma casa automatizada, é possível controlar as luzes de toda a residência, fechar cortinas, desligar pontos de ar condicionado e etc.

A segurança também é um dos grandes objetivos da Domótica, já que ela apresenta o que há de melhor na parte de alarmes, permissões de acesso, controle de câmeras de segurança.

Dessa maneira, o projeto insere-se no dia-a-dia criando uma forma alternativa e pessoal para o trancamento de uma porta.

### **1.5 – Escopo**

O projeto aqui desenvolvido limita-se a criar um dispositivo que através de batidas secretas programadas pelo usuário, seja capaz de permitir ou bloquear o acesso de uma porta por intermédio de um dispositivo microcontrolado devidamente posicionado na porta. Para isso, esse dispositivo será construído com base no funcionamento de um Arduino e de um Sensor analógico de som.

### **1.6 – Estrutura**

O trabalho conta com seis capítulos expostos da seguinte maneira:

Capítulo 1 – Introdução – Apresenta, de forma sucinta, os objetivos e importância do desenvolvimento do projeto bem como os resultados esperados.

Capítulo 2 – Apresentação do problema - Motivação da escolha do projeto, justificando a sua importância através dos benefícios trazidos pela automação.

Capítulo 3 – Referencial Teórico – Procura-se tratar da aplicação dos conteúdos vistos no decorrer do curso visando à resolução do problema proposto no Capítulo 1 e fundamentado no Capítulo 2. Para isso, são descritas as técnicas, metodologias e componentes eletrônicos usados na solução do problema.

Capítulo 4 – Descrição do Hardware e do Software– Descreve as especificações dos dispositivos utilizados, detalhando o Arduino e seu microcontrolador, o Sensor de som, o Relé e os demais dispositivos e softwares.

Capítulo 5 – Desenvolvimento e Resultados Obtidos – Aborda o desenvolvimento do projeto, como a integração entre o hardware e software, escrita do código-fonte e apresenta os resultados provenientes dos testes e os ajustes dos problemas encontrados.

Capítulo 6 – Considerações Finais – São colocadas as conclusões, embasadas nas dificuldades encontradas para atingir os objetivos detalhados no Capítulo 1. Também é sugerida alguma forma de implementação futura desse protótipo.

## CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA

Este capítulo tem o objetivo de detalhar as questões que motivaram o trabalho, como: o que é a automação residencial, histórico no Brasil e no mundo e quais os principais benefícios, principalmente em relação à segurança.

São apresentadas também algumas soluções existentes para estas questões, no que diz respeito à automatização de portas. E por fim, são apresentados os benefícios do dispositivo proposto pelo trabalho e suas restrições.

### 2.1 – Automação Residencial

*“A automação engloba qualquer sistema, apoiado em computadores, que substitua o trabalho humano e que vise a soluções rápidas e econômicas para atingir os complexos objetivos das indústrias e dos serviços.”*

*A automação residencial é um ramo que está em constante crescimento juntamente ao avanço tecnológico e o aumento das possibilidades e diversas aplicações. Criada para aplicar os conceitos provenientes da automação industrial das fábricas num ambiente mais cotidiano, é responsável por grande parte da comodidade e conforto existentes no meio doméstico. (HARPER, 2003)*

A automação é o conceito de tornar atividades repetitivas em automáticas, utilizando dispositivos que coletam dados e atuam nos processos, diminuindo a necessidade da interferência humana, resultando em maior velocidade nas operações, redução de erros e fidelidade de informações, que são elementos cruciais para melhor administração. (MORAES, CASTRUCCI, 2007)

Quando o termo AR é mencionado, a primeira reação de muitas pessoas é no sentido do desconhecimento do significado e da real importância de tal palavra. Para definir o conceito de um ambiente doméstico automatizado, é possível imaginar um conjunto de máquinas que teriam a função de trabalhar, sem nenhum repouso, tentando sanar todas as necessidades que os moradores

de uma casa possam ter. Dessa forma, pode-se afirmar uma casa automatizada deve ter uma aparência tão normal como qualquer outra casa, mas contando com ajuda de sistemas embarcados.

Durante muito tempo, os sistemas que eram encontrados em casas automatizadas tinham funções independentes e sem muita interação entre eles. Pode-se citar, por exemplo, os sistemas por trás do monitoramento de câmeras de segurança, sistemas de iluminação e sistemas de abertura/fechamento de portões. No entanto, tais sistemas sempre desempenhavam suas funções de forma isolada.

Atualmente, existem sistemas que fornecem soluções para essa interação e para vincular todos os elementos de uma casa, proporcionando aos usuários uma gama de soluções e alto grau de funcionalidade. No Brasil ainda é pouco comum, mas na Europa, existem muitos edifícios onde a domótica já é realidade e é responsável por gerenciar a segurança, bem-estar, iluminação e lazer. Estas tecnologias estão sendo progressivamente implantadas principalmente nas capitais, onde a tecnologia é bastante conciliada com atividades do dia-a-dia.

## **2.2 – Benefícios da Automação**

A tecnologia trouxe diversos avanços na vida cotidiana. Pequenas doses de facilidade foram introduzidas ao dia a dia, facilitando aos poucos tarefas que seriam mais complexas ou mais trabalhosas sem o apoio de aparatos modernos. Desde a Revolução Industrial, a automação vem lentamente tomando espaço em atividades rotineiras, começando nas linhas de montagem, onde a automação industrial se tornou essencial para a produção. Percebeu-se assim, a grande vantagem ao mecanizar processos repetitivos. (GROOVER, 1987)

Com a popularização da tecnologia e maior acessibilidade às soluções mais sofisticadas para problemas simples, a automação deixa as indústrias para solucionar problemas caseiros. AR é o termo que define a utilização dessa tecnologia para meios residenciais. Os exemplos são inúmeros, sendo alguns citados abaixo:

- Segurança automatizada (Sensores, Alarmes, Travas Eletrônicas, Câmeras de Segurança)
- Controle de iluminação de ambientes
- Controle de temperatura
- Cuidados com animais de estimação automatizados (Alimentação, Iluminação dentre outras necessidades)
- Jardim automatizado (irrigação e iluminação)
- Dentre outras.

A busca por comodidade é algo sempre presente, e as facilidades que são proporcionadas por esses dispositivos tornam a vida em si mais agradável, diminuindo o stress e a preocupação de quem os possui. Isso impulsiona o crescimento do mercado de serviços de automação, visando garantir que essa tecnologia se expanda e possa abranger cada vez mais áreas.

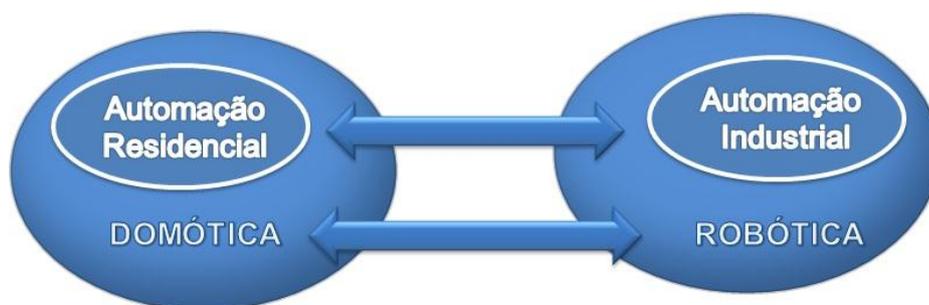
### **2.3 – Histórico**

O marco inicial para os diversos segmentos da automação foi o surgimento da máquina a vapor com a Revolução Industrial ocorrida na Inglaterra, em 1750. Foi nesse período que ocorreu a transição do modo de produção feudal, conhecido como um sistema de subsistência, para o modelo atual, o modelo de produção capitalista. A partir daí outros eventos importantes deram continuidade à expansão tecnológica. Pode-se citar:

- 1876: Alexander Graham Bell com a realização da comunicação entre dois cômodos utilizando o telefone.
- 1879: Thomas Alva Edison com a invenção da lâmpada com filamento de carbono incandescente
- 1904: John Ambrose Fleming com a válvula que permitia o desenvolvimento da eletrônica no século XX, precursora dos equipamentos de rádio e televisão;
- 1946: John Mauchly e John Eckart com o ENIAC, o primeiro computador.

- 1947: William Bradford, John Bardeen e Walter Houser com o desenvolvimento do transistor, um dispositivo que vira por fim na era das válvulas. (SRA ENGENHARIA BLOG)

A AR surge decorrente do processo iniciado por esses eventos mencionados, que teve nos dispositivos CLPs (sigla, Controladores Lógicos Programáveis, datados da década de 60, uma grande revolução graças aos avanços da microeletrônica. É nessa época, também, que surge o termo Domótica, uma equivalência ente os dois tipos de automação, conforme ilustrado na figura 2.1.



**Figura 2. 1 - Automação Residencial e Automação Industrial. (Fonte: SRA Engenharia Blog)**

A década de 70 é considerada o marco inicial da AR, quando as primeiras práticas foram realizadas principalmente no Japão e nos Estados Unidos. O protocolo X-10, os primeiros módulos “inteligentes”, utilizavam a rede elétrica como canal de comunicação entre o dispositivo e a residência. Os comandos eram enviados pela rede elétrica utilizando a tecnologia PLC. (SRA ENGENHARIA BLOG)

Mais adiante, em meados dos anos 80, com a popularização dos PCs (Computadores Pessoais), pôde-se pensar em um PC como central de automação, em detrimento aos mainframes. Entretanto, as grandes desvantagens desse sistema são o elevado consumo e a centralização do controle, que pode vir a ser falho e comprometer o funcionamento de todo o

sistema automatizado. Devido a isso, parte-se para o desenvolvimento de dispositivos embarcados através da utilização de microprocessadores e microcontroladores.

Inúmeras outras tecnologias foram sendo incorporadas à automação residencial como os controles remotos infravermelhos e radiofrequência. Os controles remotos infravermelho são capazes de interpretar diferentes protocolos utilizados por diferentes fabricantes. A tecnologia de radiofrequência (RF) difere da infravermelha por não necessitar visada direta entre o controle remoto e o dispositivo controlado. (SRA ENGENHARIA BLOG)

A internet banda larga concedeu ao usuário a possibilidade de controle e monitoramento da residência de qualquer lugar que disponha do serviço. Acrescenta-se a este fato a convergência tecnológica intensificada a partir do século XXI, na qual um mesmo dispositivo (celular, smarphone, tablet, etc.) pode incorporar diferentes serviços (telefonia, internet, monitoramento, controle da residência e assim por diante). (SRA ENGENHARIA BLOG, modificado)

No Brasil também se pode observar uma rápida absorção das novas tecnologias pelos usuários no dia-a-dia. Porém, esta tendência ainda não se transferiu para o mercado de construção civil na mesma intensidade. Aqui, a domótica pode ser um fator decisivo para atingir consumidores com necessidades específicas, das quais se pode destacar a segurança, a acessibilidade, o entretenimento, o conforto, a conveniência e a economia de energia. Sendo assim, pode-se esperar uma evolução dos lançamentos imobiliários com diferenciais tecnológicos, apesar do custo elevado quando comparado com equipamentos que também contam com eletrônica embarcada. (SRA ENGENHARIA BLOG, modificado)

#### **2.4 – Benefícios do Dispositivo Proposto e suas Restrições**

A função do projeto é apresentar um dispositivo para controle de abertura de uma porta, integrando o sensor de som com a plataforma Arduino.

O principal benefício do dispositivo será a possibilidade do usuário abrir/fechar uma determinada porta através de batidas previamente gravadas no microcontrolador. Paralelamente, outro dispositivo está fazendo o papel de segurança, de modo que a cada 3 tentativas erradas o usuário receberá um *feedback* do dispositivo, informando se ocorreu uma tentativa de abertura da porta.

O projeto não tem como funcionalidade de tratar erros do sensor ou até armazenar os dados recebidos pelo mesmo. O protótipo somente fará a leitura do nível do gás e dependendo deste valor terá um comportamento.

Por uma questão de melhor demonstração do funcionamento do projeto, será utilizada uma maquete com o dispositivo acoplado, simulando uma aplicação real.

## **CAPÍTULO 3 – REFERENCIAL TEÓRICO**

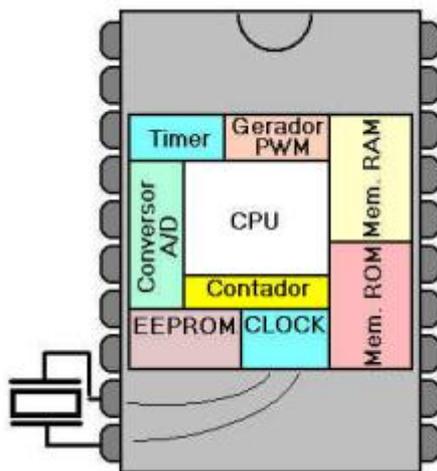
Neste capítulo são abordados termos e conceitos teóricos necessários para um melhor entendimento do projeto e suas funcionalidades. Alguns desses temas, por serem extensos e visando o foco no projeto descrito neste trabalho, são apresentados somente as características e conceitos pertinentes.

É apresentada também uma visão geral do protótipo construído. O entendimento deste capítulo é essencial para um melhor aproveitamento do restante do trabalho

### **3.1 – Microcontroladores**

Um microcontrolador é um dispositivo semicondutor em forma de circuito integrado, que integra as partes básicas de um microcomputador e um microprocessador, memórias não voláteis e voláteis e portas de entrada e saída. Por ser limitado em termos de quantidade de memória, principalmente no que diz respeito à memória de dados, é utilizado em aplicações específicas, isto é, naquelas que não necessitam armazenar grandes quantidades de dados. (GIMENEZ, 2005)

Além de receber entradas do usuário, o microprocessador também recebe informações do próprio produto no qual está embarcado, e controla o aparelho mandando sinais para diversos componentes do mesmo. É também, de certa forma, resistente, podendo ser aplicado em condições mais extremas onde computadores normais não conseguiriam. (GROOVER, 1987)



**Figura 3. 1 - Esquema de um Microcontrolador. (Fonte: eletronicagratis.blogspot.com.br)**

Um bom exemplo do funcionamento do microcontrolador seria observando um aparelho de micro-ondas. Além de receber informações das teclas e exibi-las no display, controla o relé que liga ou desliga o funcionamento do aparelho. Pode-se observar a sua funcionalidade quando aplicado ao controle de motores de carros onde, dependendo da localização e clima, podem funcionar a temperaturas de  $-34^{\circ}\text{C}$  até  $49^{\circ}\text{C}$ . Se for considerado o calor gerado pelo próprio motor, as temperaturas podem subir até  $80^{\circ}\text{C}$ , sem que o controlador pare de funcionar ou perca suas capacidades. (GROOVER, 1987)

Muitos componentes foram lançados no mercado com essas características. Com a evolução da microeletrônica, empresas como a Microchip, ATMEL, Itachi, Intel são alguns dos principais fornecedores de microcontroladores atualmente.

O modelo adotado nesse projeto é o ATmega328, componente da placa Arduino Duemilanove, devido a abundância de pinos I/O digitais e analógicas e pela facilidade na comunicação serial com o computador. Suas principais características são discutidas no tópico a seguir.

### 3.1.1 – Arduino

O Arduino é uma plataforma de prototipagem eletrônica de software livre, projetada para circuitos de entrada/saída para um microcontrolador Atmel AVR, um ambiente de desenvolvimento e o bootloader que já vem gravado no microcontrolador, ou que pode ser gravado facilmente caso construa seu próprio Arduino a partir dos esquemas disponíveis.

Pode ser utilizada para desenvolver objetos que sejam independentes ou objetos que podem ser conectados a um computador, uma rede ou até mesmo à internet para recuperar e enviar dados. (JUCA, modificado)

Os circuitos integrados da Atmel AVR embutidos na placa são categorizados como microcontroladores por possuir:

- CPU: Para interpretar as instruções de programa
- Memória EPROM: Para gravar permanentemente as instruções de programa
- Memória RAM: Para o armazenamento de variáveis utilizadas pelo código
- Barramentos de I/O: Para conexão de dispositivos externos, recepção de pulsos, sinais além de uma série de dispositivos auxiliares.

Outra importante parte essencial é seu *clock* ou circuito oscilador. Ele é responsável por ditar a velocidade em que as instruções serão executadas. Por estarem todos esses dispositivos num só circuito integrado, há uma maior facilidade e rapidez de aplicação poupando o custo e espaço que todos esses dispositivos trariam, vantagens de um sistema microcontrolado. (JUCA, 2010)

### 3.2 – Eletromagnetismo

Eletromagnetismo é o nome que se dá ao conjunto de teorias desenvolvidas para explicar a relação existente entre a eletricidade e o magnetismo. Desde da Grécia Antiga que fenômenos eletromagnéticos são conhecidos, no entanto, apenas nos séculos XVIII e XIX através de físicos como Ampère, Ohm, Faraday e Maxwell que essas teorias obtiveram evidência através do chamado fenômeno eletromagnético.

Para produzir energia elétrica, é necessário o consumo de uma forma de energia qualquer. No século XIX, somente a energia química, obtida através das pilhas e baterias, era transformada em energia elétrica através de um processo químico espontâneo. No entanto, a conversão desse tipo de energia para energia elétrica não era adequado para alimentar as grandes indústrias da época. Foi assim que no ano de 1831, Michael Faraday descobriu o fenômeno da indução eletromagnética, o qual revolucionou o estudo do eletromagnetismo. Graças a essa descoberta, foi possível construir aparelhos que funcionam através da indução eletromagnética e que transformam energia mecânica em energia elétrica, como os motores elétricos, ilustrado na figura 3.2. (BRASIL ESCOLA, modificado)



**Figura 3. 2 - Motor Elétrico.**

(Fonte: <http://www.orientalreduutores.com.br>)

O assunto do eletromagnetismo é muito vasto, e seu estudo possibilita o entendimento de uma variedade de instrumentos e coisas que fazem parte de nosso cotidiano, como por exemplo, a automação de portas, fenômenos parecidos com os que são adotados em supermercados e *shopping centers*.

No entanto, sua variedade de características e o seu princípio de funcionamento exigem cuidados especiais para que componentes

eletromagnéticos possam ser utilizados corretamente. Para esse projeto, foi utilizado um relé de 5 V, explicado no tópico a seguir e detalhado no capítulo 4.

### 3.2.1 – Relés

Relés (derivado do inglês Relais) é um interruptor eletromecânico destinado a produzir modificações em contatos elétricos. A movimentação física deste interruptor ocorre quando a corrente elétrica percorre as espiras da bobina, criando assim um campo magnético que por sua vez atrai a alavanca responsável pela mudança do estado dos contatos. (INSTITUTO NEWTON C. BRAGA, modificado)

A história dos relés começou nos anos de 1820 e 1830, quando o cientista norte-americano Joseph Henry estudava a ação dos eletroímãs, dispositivos que utilizam correntes elétricas para gerar campos magnéticos a exemplo dos campos existentes nos ímãs naturais. Entretanto, apenas em 1878 que esse dispositivo passou a ser utilizado em larga escala. O uso comercial dos relés foi iniciado pelo inventor americano Samuel Morse, através da criação do telégrafo, em 1937. (O SETOR ELÉTRICO, modificado)

Embora sejam componentes lentos e em alguns casos menos confiáveis que os dispositivos semicondutores de comutação, os relés eletromecânicos oferecem uma infinidade de recursos para aplicações em projetos de Robótica e Mecatrônica. Na verdade, existem situações em que o uso do relé pode ser até mais simples do que um circuito eletrônico que utilize um componente de estado sólido. As figuras 3.2 e 3.3 ilustram relés encontrados no mercado. (INSTITUTO NEWTON C. BRAGA, modificado)



Figura 3. 3 - Relés Comerciais.

(Fonte: <http://http://seriallink.com.br/loja/images/relés.jpg>)



Figura 3. 4 – Relé industrial.

(Fonte: [www.labdegaragem.org](http://www.labdegaragem.org))

### **3.3 – Linguagens de Programação**

Uma linguagem de programação é um meio de expressar informações de modo compreensível tanto para o computador quanto para as pessoas. A sintaxe específica como uma combinação de frases (podendo ser comandos, declarações e etc) pode gerar um programa. (HARPER, 2003)

Este tipo de comunicação entre computadores e programadores é o que resulta em diversos sistemas com diferentes finalidades, sejam eles sistemas operacionais, indispensáveis para qualquer computador, ou mesmo para microcontroladores, visando aplicações menores e mais pontuais.

Ao todo, as linguagens de programação se dividem em dois grupos: as de baixo nível e as de alto nível. Os computadores interpretam tudo como números em base binária, ou seja, só entendem combinações entre 0 e 1. As linguagens de baixo nível são interpretadas diretamente pelo computador, tendo um resultado rápido, apesar de ser muito difícil e incômodo de se trabalhar com ela. Exemplos desse tipo de linguagens são a linguagem binária e a linguagem Assembly.

Já as linguagens de alto nível são mais fáceis de trabalhar e de entender, as ações são representadas por palavras de ordem (exemplo faça, imprima, pare) geralmente em inglês, foram feitas assim para facilitar a memorização e a lógica. Elas não são interpretadas diretamente pelo computador, sendo necessário traduzi-las para linguagem binária utilizando-se de um compilador. Exemplo desse tipo de linguagem é a linguagem C, que foi utilizada nesse projeto e será mais detalhada no tópico 3.3.1. (INFOESCOLA, modificado)

#### **3.3.1 – Linguagem C**

A linguagem C é uma linguagem de alto nível, genérica. Foi desenvolvida por programadores para programadores tendo como meta características de flexibilidade e portabilidade. O C é uma linguagem que nasceu juntamente com o advento da teoria de linguagem estruturada e do computador pessoal. Assim tornou-se rapidamente uma linguagem “popular” entre os programadores. O C

foi usado para desenvolver o sistema operacional UNIX e hoje está sendo usada para desenvolver linguagens modernas, como C++ e Java. (PEREIRA, 2007)

Entre as principais características do C, pode-se citar:

- O C é uma linguagem de alto nível com uma sintaxe bastante estruturada e flexível tornando sua programação bastante simplificada.
- Programas em C são compilados, gerando programas executáveis.
- O C compartilha recursos tanto de alto quanto de baixo nível, pois permite acesso e programação direta do microprocessador. Com isto, rotinas cuja dependência do tempo é crítica, podem ser facilmente implementadas usando instruções em Assembly. Por esta razão o C é a linguagem preferida dos programadores de aplicativos.
- O C é uma linguagem estruturalmente simples e de grande portabilidade. O compilador C gera códigos mais enxutos e velozes do que muitas outras linguagens.
- Embora estruturalmente simples (poucas funções intrínsecas) o C não perde funcionalidade pois permite a inclusão de uma farta quantidade de rotinas do usuário. Os fabricantes de compiladores fornecem uma ampla variedade de rotinas pré-compiladas em bibliotecas.

A escolha de C para programação em microcontroladores se torna óbvia ao observar a grande velocidade na criação de novos projetos devido às facilidades de programação fornecidas pela linguagem, sua portabilidade que permite adaptar programas de um sistema para o outro com um mínimo de esforço e sua proximidade à linguagem de máquina, proporcionando extrema eficiência, o colocam como linguagem de alto nível mais eficiente atualmente disponível (PEREIRA, 2007).

### 3.4 – Sensor de Som

Sensor é um termo empregado para designar dispositivos sensíveis a alguma forma de energia do ambiente. Um sensor nem sempre tem as características elétricas necessárias para ser utilizado em um sistema de controle. Geralmente o sinal de saída é manipulado antes de sua leitura no sistema de controle. (THOMAZINI e ALBUQUERQUE, 2005).

É um termo utilizado para indicar dispositivos sensíveis a alguma forma de energia do ambiente que pode ser luminosa, térmica, cinética, relacionando informações sobre uma grandeza que pode ser medida, como: temperatura, pressão, velocidade, corrente, aceleração, posição, etc. (THOMAZINI e ALBUQUERQUE, 2005).

O som é uma das formas de energia que o universo apresenta, e como tal, obedece às leis fundamentais da física. Dessa forma, BONJORNIO (1992) e SOARES (1984) descrevem o som como aumentos e reduções periódicas da densidade do ar, ou seja, compressão e rarefação do mesmo. Suas principais características são:

- Intensidade: relativa à força do som. Distingue sons mais fracos de sons mais fortes.
- Timbre: geralmente definida como a “cor” do som, pois através dele é possível identificar um mesmo som produzido por diferentes fontes.
- Duração: referente ao tempo, isto é, se os sons são mais longos ou mais curtos.
- Altura: referente ao “alcance” do som. Classifica os sons mais altos como agudos e os mais baixos como graves.

Assim, para detectar essencialmente a intensidade das batidas na porta, é utilizado um sensor de som analógico que funciona a partir de um microfone de eletreto e um amplificador operacional, que permite converter o som detectado em energia elétrica. No tópico 4.4 estão as especificações e os princípios físicos que regem o funcionamento do microfone e do amplificador para a detecção do som.

### 3.5 – Visão do Projeto

O protótipo construído é constituído basicamente de um microcontrolador Atmega328, embutido em uma placa Arduino Duemilanove, um relé 5V, um sensor analógico de som e demais componentes elétricos, como resistores, diodos e transistor.

No próximo capítulo é feito um detalhamento sobre o hardware e software que este projeto abrange e seu funcionamento será descrito no capítulo 4 – Descrição de Hardware e Software.

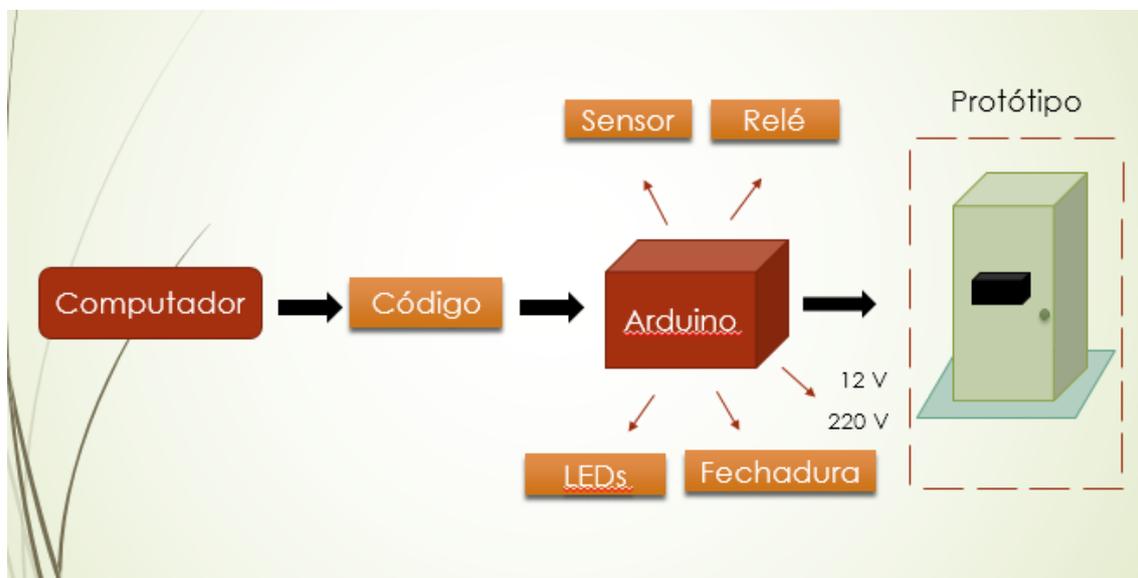


Figura 3. 5 - Visão Geral do Projeto.

(Fonte: Autor)

## CAPÍTULO 4 – DESCRIÇÃO DO HARDWARE E DO SOFTWARE

Este capítulo é responsável pelo detalhamento dos componentes utilizados na construção do protótipo e seu funcionamento, englobando a parte física (hardware) e a parte lógica (software).

### 4.1 - Arduino

O Arduino utilizado nesse projeto é o modelo Duemilanove, conforme especificado no capítulo 3. Ele é uma plataforma de prototipagem eletrônica de software livre, projetada com um microcontrolador, com suporte I/O embutida e uma linguagem de programação padrão baseada em C++. Ele é o responsável por fazer a comunicação com o sensor de som e com o motor DC, outras peças fundamentais do circuito.

Sua placa consiste em um microcontrolador ATmega328, um microcontrolador de 8 bits, 14 pinos I/O digitais, 6 entradas analógicas, um regulador linear de 5 volts, um oscilador de cristal de 16 MHz, uma conexão USB, uma entrada para alimentação, um cabeçalho ICSP e um botão de reset. A figura 4.1 ilustra o modelo Duemilanove utilizado nesse projeto. (McROBERTS, 2011)

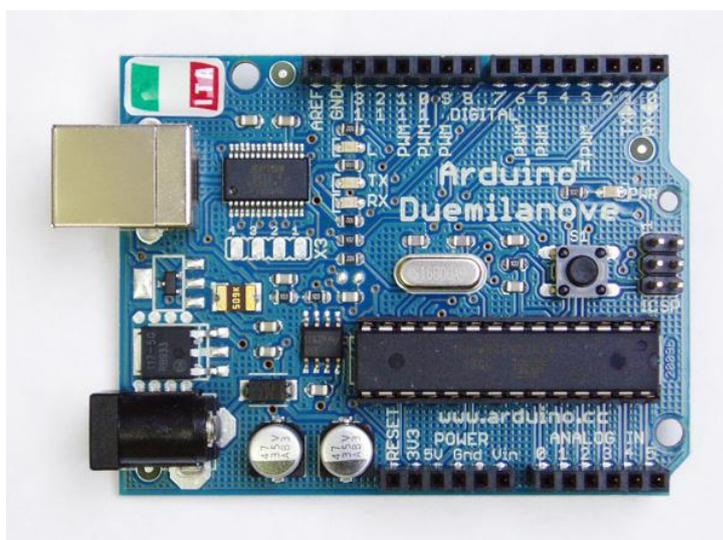


Figura 4. 1 - Arduino Duemilanove. (Fonte: Multilógica)

### 4.1.1 – Especificação

Quadro 4. 1 - Especificações Arduino Duemilanove. (Fonte: Multilógica)

Características	
Microcontrolador	ATmega328
Voltagem operacional	5V
Voltagem de alimentação (recomendada)	7-12V
Voltagem de alimentação (limites)	6-20V
Pinos I/O digitais	14 (dos quais 6 podem ser saídas PWM)
Pinos de entrada analógica	6
Corrente DC por pino I/O	40 mA
Corrente DC para o pino 3.3V	50 mA
Memória flash	32 KB (sendo que 2 KB são usados para o bootloader)
SRAM	2 KB
EEPROM	1 KB
Velocidade de clock	16 MHz

#### Alimentação

Com relação à alimentação, o Arduino pode ser alimentado pela conexão USB (cabo não incluso) ou por qualquer fonte de alimentação externa. A fonte de alimentação é selecionada automaticamente. Alimentação externa (não-USB) pode ser tanto de uma fonte quanto de uma bateria.

Pode operar com uma alimentação externa de 6 a 20 volts. Entretanto, se a alimentação for inferior a 7 volts o pino 5V pode fornecer menos de 5 volts e a placa pode ficar instável. Se a alimentação for superior a 12 volts o regulador de voltagem pode superaquecer e avariar a placa. A alimentação recomendada é de 7 a 12 volts. Os pinos de alimentação são:

- VIN: Entrada de alimentação para a placa Arduino quando uma fonte externa for utilizada.
- 5V: A fonte de alimentação utilizada para o microcontrolador e para outros componentes da placa. Pode ser proveniente do pino Vin através de um regulador on-board, ser fornecida pelo USB ou por outra fonte de 5 volts.

- 3V3: Alimentação de 3,3 volts fornecida pelo chip FTDI. A corrente máxima é de 50 mA
- GND: Pino terra.

### Memória

O microcontrolador ATmega328 tem 2 KB de memória flash para armazenar código (dos quais 2 KB são do bootloader, conforme mencionado na tabela 4.1), além de 2 KB de SRAM e 1 KB de EEPROM.

### I/O

Cada um dos 14 pinos digitais do Duemilanove pode ser usado como entrada ou saída e operam com 5 volts. Cada pino pode fornecer ou receber um máximo de 40 mA e tem um resistor pull-up de 20-50 k $\Omega$ . Além disso, alguns pinos têm função específica, que podem ser visualizadas no tópico 4.1.2.

### Comunicação

A comunicação com um computador, com outro Arduino ou com outros microcontroladores é muito simples. O ATmega328 permite comunicação serial no padrão UART TTL (5V), que está disponível nos pinos digitais 0 (RX) e 1 (TX). Um chip FTDI FT232RL na placa encaminha esta comunicação serial através do USB e os drivers FTDI (incluído no software do Arduino) fornece uma porta COM virtual para o software no computador. O software Arduino inclui um monitor serial que permite que dados simples de texto sejam enviados à placa Arduino. (MULTILÓGICA, 2013)

### 4.1.2 – Pinagem

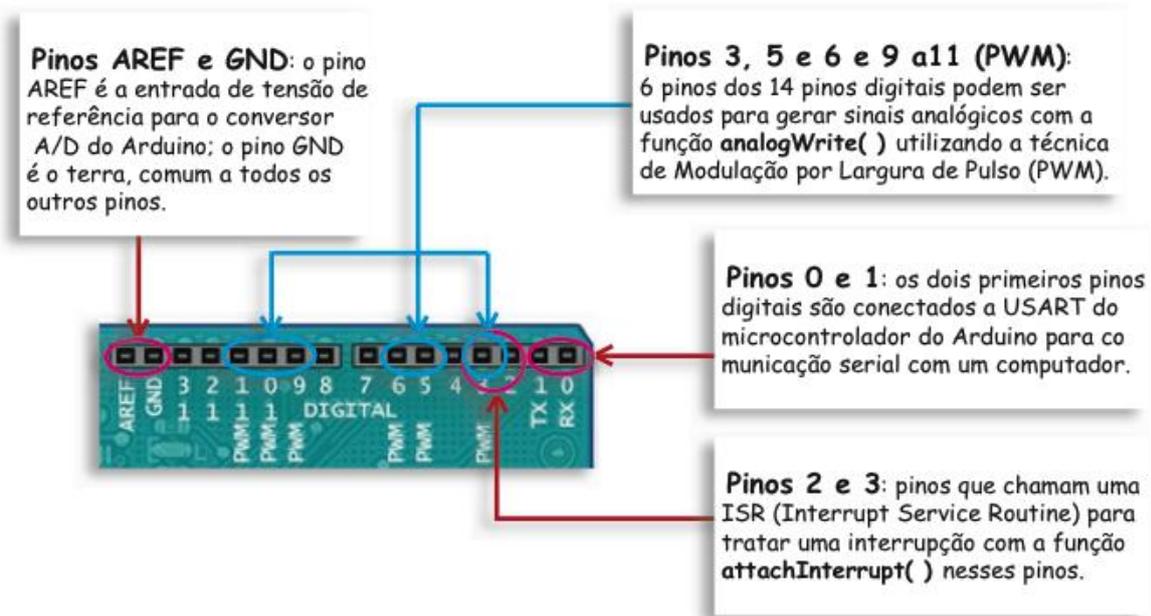


Figura 4. 2. a) - Pinos Digitais.

(Fonte: <http://meetarduino.wordpress.com>)

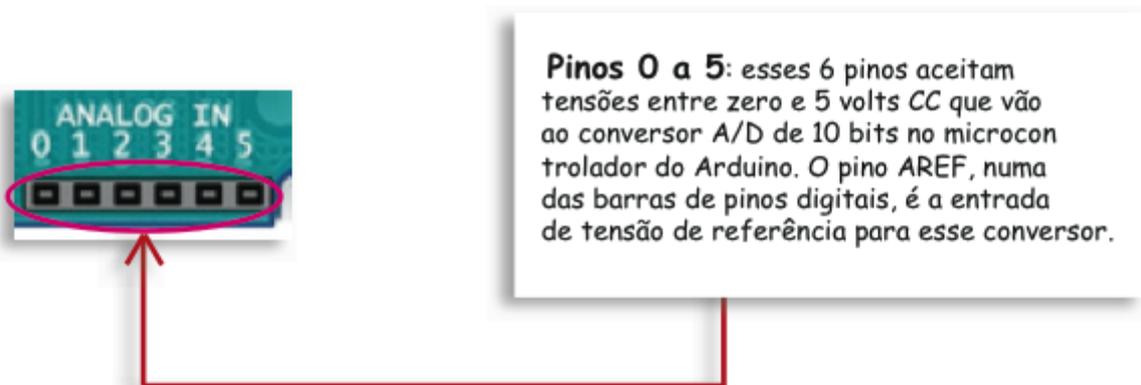
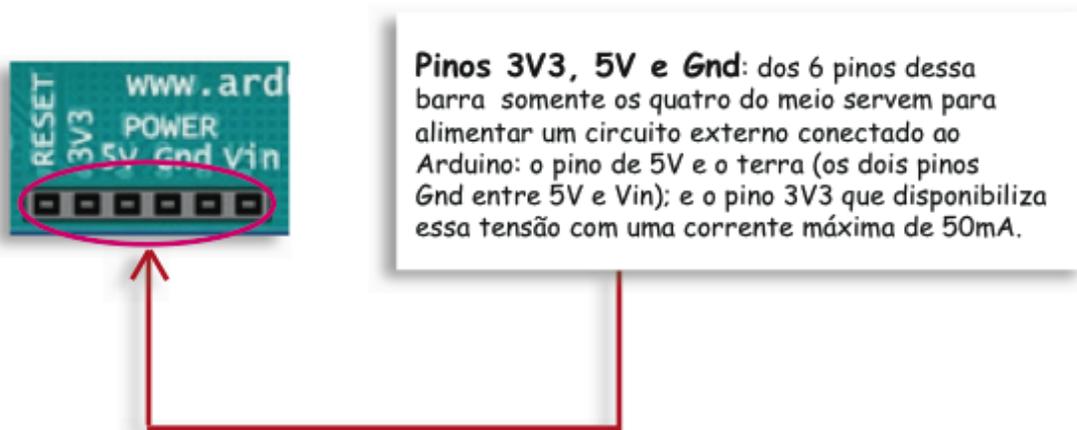


Figura 4. 2. b) - Pinos Analógicos.

(Fonte: <http://meetarduino.wordpress.com>)



**Figura 4. 2. c) - Pinos de Alimentação.**

(Fonte: <http://meetarduino.wordpress.com>)

## 4.2 – Arduino IDE

Para programar o Arduino é necessário utilizar a interface chamada Arduino IDE, um software livre no qual você escreve um código na linguagem de máquina que o Arduino compreende (baseada na linguagem C/C++). O IDE permite que o usuário escreva um algoritmo, que é um conjunto de instruções passo a passo, e faça o upload para o Arduino. O Arduino, então, executará essas instruções, interagindo com o que estiver conectado a ele. No mundo do Arduino, programas são conhecidos como sketches (rascunho, ou esboço). (McROBERTS, 2011)

O grande diferencial desta ferramenta é que ela é desenvolvida e aperfeiçoada por uma comunidade que divulga os seus projetos e seus códigos de aplicação, pois a concepção dela é open-source, ou seja, qualquer pessoa com conhecimento de programação pode modificá-lo e ampliá-lo de acordo com a necessidade, visando sempre a melhoria dos produtos que possam ser criados aplicando o Arduino. ((McROBERTS, 2011, modificado)

O microcontrolador ATmega 328 no Arduino Duemilanove vem pré-gravado com um bootloader, que permite enviar novos programas sem o uso de um programador de hardware externo. Ele se comunica utilizando o protocolo

original STK500. Alternativamente, é possível programar através do ICSP, *In-Circuit Serial Programming header*. (MULTILÓGICA)

### 4.3 – Relé

Relé é um dispositivo eletromecânico/magnético que funcionam como interruptores ou chaves comutadores eletromecânicas. O modelo utilizado no projeto é o relé 5V, ilustrado na figura 4.3 e que tem como função a “comunicação” entre o microcontrolador com a fechadura da porta. Após a detecção da batida correta, o responsável por movimentar a fechadura é o contato feito pelo relé. Descrito abaixo se encontra o funcionamento detalhado desse componente.



Figura 4. 3 - Relé 5V.

(Fonte: <http://www.http://seriallink.com.br>)



### 4.3.2 – Funcionamento

Os relés trabalham da seguinte forma: quando uma corrente circula pela bobina, esta cria um campo magnético que atrai um ou uma série de contatos fechando ou abrindo circuitos. Ao cessar a corrente da bobina o campo magnético também cessa, fazendo com que os contatos voltem para a posição original.

Os relés podem ter diversas configurações quanto aos seus contatos: podem ter contatos NA, NF ou ambos, neste caso com um contato comum ou central (C). Os contatos NA (normalmente aberto) são os que estão abertos enquanto a bobina não está energizada e que fecham, quando a bobina recebe corrente. Os NF (normalmente fechado) abrem-se quando a bobina recebe corrente, ao contrário dos NA. O contato central ou C é o comum, ou seja, quando o contato NA fecha é com o C que se estabelece a condução e o contrário com o NF.

A principal vantagem dos relés em relação a outros componentes é que o circuito de carga está completamente isolado do de controle, podendo inclusive trabalhar com tensões diferentes entre controle e carga. A desvantagem é o fator do desgaste, pois em todo o componente mecânico há uma vida útil, o que não ocorre nos Tiristores (dispositivos semicondutores), por exemplo. Devem ser observadas as limitações dos relés quanto à corrente e tensão máxima admitida entre os terminais. Caso contrário, a vida útil do relé estará comprometida, ou até a do circuito controlado como um todo.

### 4.4 – Sensor de Som

O sensor de som analógico mostrado na figura 4.4 tem um microfone de eletreto e um amplificador operacional com ganho de 100x e permite sensibilidade sonora através de um conversor DAC de um microcontrolador.

Para a funcionalidade do projeto, esse sensor é ideal, pois além de atender as necessidades tem um custo benefício acessível.



Figura 4. 5 – Sensor de Som.

(Fonte: <http://www.dfrobot.com>)

#### 4.4.1 – Especificações

Estes microfones de eletreto, também conhecidos como microfones de condensador, precisam de um resistor e de um capacitor para poderem ser conectados ao circuito. Sua alimentação típica vai de 3 a 10 V, sendo que a partir de 1,5 V já possuem um bom funcionamento e, com o aumento do resistor, possam funcionar com tensões até maiores do que 10 V (12 e 15 V são normais). Na ligação desse microfone é necessário observar sua polaridade e, além disso, usar cabo blindado para conexão à entrada do amplificador, para que não ocorra a captação de zumbidos durante a operação. (INSTITUTO NEWTON C. BRAGA)

A figura 4.5 indica o circuito interno da placa utilizada.

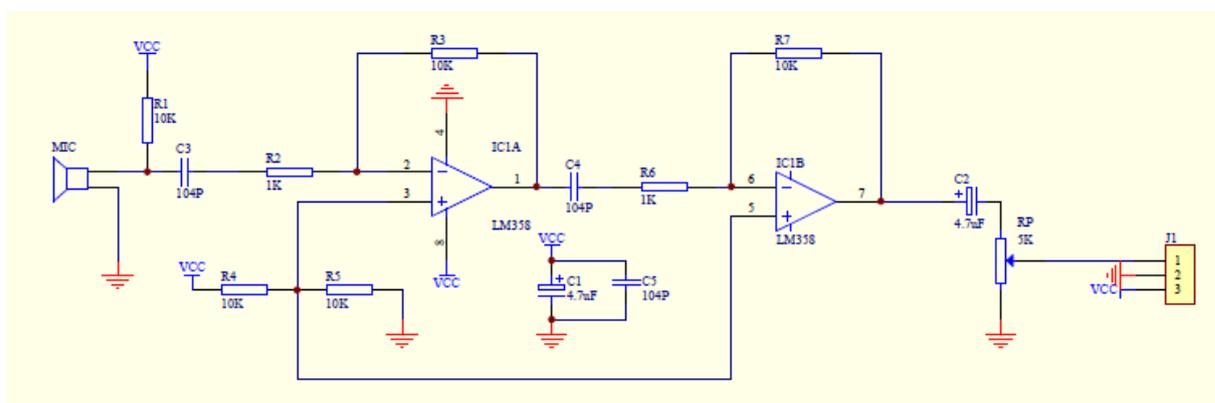


Figura 4. 5 - Esquema do Sensor de Som.

(Fonte:[http://www.dfrobot.com/image/data/DFR0034/DFR0034\\_SoundSensor\\_SC.pdf](http://www.dfrobot.com/image/data/DFR0034/DFR0034_SoundSensor_SC.pdf))

## **CAPÍTULO 5 – DESENVOLVIMENTO E RESULTADOS OBTIDOS**

Neste capítulo é descrito como o projeto foi planejado, como foi feito e suas etapas para que fosse concluído. Por isso, é considerado como o capítulo principal.

Visado facilitar o entendimento dos procedimentos realizados, ele é dividido em quatro tópicos:

1. Modelagem do sistema;
2. Elaboração dos circuitos;
3. Elaboração do código fonte para o Arduino;
4. Montagem do protótipo;

### **5.1 – Modelagem do Sistema**

A primeira etapa na implementação do projeto, foi a realização de um esboço do sistema definitivo, ou seja, a disposição em que se encontrariam os componentes em um ambiente real e que pudesse ser escolhida a melhor forma para dispor os componentes visando maior comodidade ao usuário e garantindo melhor desempenho do projeto.

Para isso foi pensado em como o dispositivo ficaria acoplado a porta ocupando o mínimo de espaço possível e como seria seu funcionamento. Esta etapa auxiliou na visualização do projeto como um todo e foi extremamente importante para que alterações necessárias fossem feitas. O projeto consiste em uma porta de madeira contendo o equipamento, que engloba o Arduino e as placas perfuradas, que servem de junção entre o Arduino com os demais componentes. Para montagem do protótipo foi utilizado a maior quantidade de material reutilizável e reciclável possível, visando menor custo na produção do

produto final. Na figura 5.1, consta os componentes adquiridos na confecção do projeto.



**Figura 5. 1 - Componentes do Projeto.**

**(Fonte: Autor)**

Com a modelagem da ideia do sistema, a próxima etapa foi a elaboração do código fonte e testes utilizando a porta Serial do Arduino e gravação no microcontrolador. Para a escrita das linhas de comando e compilação do código foi utilizado a Arduino IDE.

## **5.2 – Fluxograma Geral do Dispositivo**

Após a definição dos componentes, foi elaborado o fluxograma geral do dispositivo, conforme ilustrado na figura 5.2. Este fluxograma foi fundamental na

escrita do código fonte, fazendo a integração entre a teoria e a prática do projeto proposto.

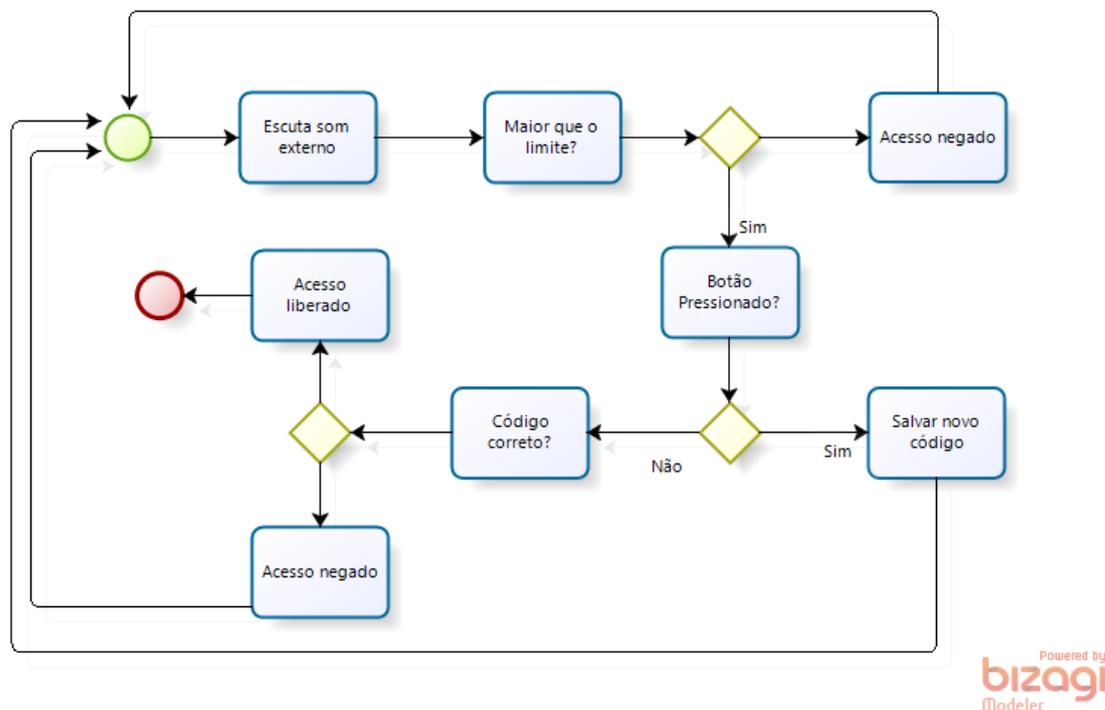


Figura 5. 2 - Fluxograma Geral do Dispositivo.

(Fonte: Autor)

## 5.2 – Escrita do Código Fonte

O código fonte foi escrito utilizando a linguagem Wiring, uma linguagem baseada em C/C++, adotada por sua facilidade e alto nível de programação, vantagens sobre a programação em ASSEMBLY. A linguagem foi escolhida também pela grande biblioteca de exemplos e informações disponíveis em livros e sites pela *internet*.

*“... a utilização de uma linguagem de alto nível permite que o programador preocupe-se mais com a programação da aplicação em si, já que o compilador assume para si tarefas como o controle e localização das variáveis, verificação de bancos de memória, etc.” (PEREIRA, 2007).*

A seguir é explicada a escrita do código-fonte, com as funções do programa para o devido funcionamento do projeto com todos seus componentes.

```

/* INICIO - Diretivas Pré-Compilação */

// Pin Definitions
const int microfone = 0;
const int botao = 2;
const int motor = 3;
const int led_vermelho = 4;
const int led_verde = 5;

// Tuning constants
const int min_sinal = 3;
const int reject_value = 25;
const int average_reject_value = 15;
const int intervalo_batidas = 150;
const int intervalo_motor = 650;
const int max_qtd_batidas = 20;
const int batida_gravada = 1200;

// Variáveis
int batida_secreta[max_qtd_batidas] = {50, 25, 25, 50, 100, 50, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
int sensor_leitura[max_qtd_batidas];
int ultima_batida_lida = 0;
int botao_status = false;

/* FIM - Diretivas Pré-Compilação*/

```

**Figura 5. 3 - Código Fonte Diretivas Pré-Compilação.**

**(Fonte: Autor)**

Na figura 5.3 é mostrado o início da escrita, com as chamadas Diretivas Pré-Compilação, onde por definição da linguagem, são descritas as constantes, variáveis e bibliotecas necessárias para o funcionamento do programa. Nesse caso as constantes e variáveis declaradas são, basicamente, a posição dos componentes nos pinos do Arduino e alguns arrays para armazenar os sons que podem ser considerados como relevantes pelo sistema.

```

Codigo_fonte_final
/*===== INÍCIO - Função Principal=====*/
void setup(){
  pinMode(motor, OUTPUT);
  pinMode(led_vermelho, OUTPUT);
  pinMode(led_verde, OUTPUT);
  pinMode(botao, INPUT);
  digitalWrite(led_verde, HIGH);
  Serial.begin(9600);
  Serial.println("Iniciando");
}

void loop(){
  // Escuta qualquer som na porta
  ultima_batida_lida = analogRead(microfone);

  if(digitalRead(botao) == HIGH){
    botao_status = true;
    digitalWrite(led_vermelho, HIGH);
  }else{
    botao_status = false;
    digitalWrite(led_vermelho, LOW);
  }

  if(ultima_batida_lida >= min_sinal){
    leituraNovaBatida();
  }
}
/*===== FIM - Função Principal=====*/

```

**Figura 5. 4 – Código Fonte Função Inicial.**

**(Fonte: Autor)**

Na figura 5.4 são apresentadas as chamadas funções principais do Arduino, as funções “setup ()” e “loop ()”. A função setup é a primeira função a ser chamada quando o programa inicia. Ela é executada uma única vez em toda a compilação do código. Ela é dita como uma função de preparação, pois é nela em que são descritos os comportamentos dos pinos e inicializa a porta serial. A função loop é chamada logo a seguir e todas as funções embarcadas nela são repetidamente executadas. Essa função fica “lendo” os pinos de entrada do Arduino e comandando os pinos de saída e a porta serial. Nesse caso específico, na função setup são descritos o relé e os LED’s como sendo pinos de saída digital e o push button como um pino analógico. A função loop inicia fazendo a

leitura de qualquer ruído que o sensor de som detectar na porta e faz um jogo de luzes com os LED's para dar um *feedback* do status do sistema para o usuário. Se esses ruídos detectados for maior ou igual à variável “min\_sinal”, variável que estipula um valor mínimo de intervalo em milisegundos para um ruído ser considerado uma batida, o sistema entra na função “leituraNovaBatida()” e assim por diante, conforme indicado nas figuras 5.5 a) e b). Todo o código-fonte pode ser conferido no apêndice A.

```

/*===== INÍCIO - Função para gravar o tempo das batidas =====*/
void leituraNovaBatida(){
  Serial.println ("Batida detectada");

  int n = 0;
  for (n = 0; n < max_qtd_batidas; n++){
    sensor_leitura[n] = 0;
  }

  int incremento = 0;
  int tempo_inicial = millis();
  int tempo_real = millis();

  digitalWrite(led_verde, LOW);
  if(botao_status == true){
    digitalWrite(led_vermelho, LOW);
  }
  delay(intervalo_batidas);
  digitalWrite(led_verde, HIGH);
  if(botao_status == true){
    digitalWrite(led_vermelho, HIGH);
  }
}

do{
  // Escuta a nova batida
  ultima_batida_lida = analogRead(microfone);
  if(ultima_batida_lida >= min_sinal){
    // Grava o intervalo
    Serial.println("Batidas.");
    tempo_real = millis();
    sensor_leitura[incremento] = tempo_real - tempo_inicial;
    incremento = incremento + 1;
    tempo_inicial = tempo_real; // reseta a variável de tempo para a próx. batida
    digitalWrite(led_verde, LOW);
    if(botao_status == true){
      digitalWrite(led_vermelho, LOW);
    }
  }
}

```

**Figura 5. 5. a) – Código Fonte Função Parte I.**

**(Fonte: Autor)**

```

}while((tempo_real < batida_gravada) && (incremento < max_qtd_batidas));

// Com a batida gravada, agora ela é validada
if(botao_status == false){
  if(batidaValidacao() == true){
    destrancarPorta();
  }else{
    Serial.println("Erro!");
    digitalWrite(led_vermelho, LOW);
    for(n = 0; n < 4; n++){
      digitalWrite(led_vermelho, HIGH);
      delay(100);
      digitalWrite(led_vermelho, LOW);
      delay(100);
    }
    digitalWrite(led_verde, HIGH);
  }
}else{
  batidaValidacao();
  // O piscamento dos LEDs é pra mostrar que a gravação/validação foi completada
  Serial.println("Nova batida gravada!");
  digitalWrite(led_vermelho, LOW);
  digitalWrite(led_verde, HIGH);
  for(n = 0; n < 3; n++){
    delay(100);
    digitalWrite(led_vermelho, HIGH);
    digitalWrite(led_verde, LOW);
    delay(100);
    digitalWrite(led_vermelho, LOW);
    digitalWrite(led_verde, HIGH);
  }
}
}
}
/*===== FIM - Função para gravar o tempo das batidas =====*/

```

**Figura 5. 5. b) – Código Fonte Função Parte II.**

(Fonte: Autor)

### 5.2.1 – Compilação e Gravação do Código Fonte

Com o código fonte escrito, o próximo passo é a sua compilação, com a utilização da IDE do Arduino Duemilanove, para que seja gerado o arquivo necessário para a simulação e a gravação na placa. A figura 5.6 ilustra o momento final de compilação onde o software dá algumas informações sobre o código-fonte. É descrito o tamanho do arquivo em bytes e a quantidade de linhas.

Done compiling.

Binary sketch size: 4.324 bytes (of a 30.720 byte maximum)

215

**Figura 5. 6 - Teste do Compilador.**

(Fonte: Autor)

Pressionando o botão para compilar (F9), o compilador começa uma série de verificações sintáticas, semânticas e lógica no código, verificando possíveis erros que necessitem ser corrigidos antes da compilação de fato. No caso da existência de algum problema, o programa indica o tipo de erro assim como a linha em que esse erro possivelmente ocorreu. Caso esteja tudo correto, o compilador gera um arquivo com extensão ".INO" que será utilizada tanto para a simulação quanto para a gravação no Arduino.

### 5.3 – Elaboração do Protótipo e dos Circuitos

Inicialmente o projeto foi planejado contendo uma maquete de uma porta contendo um fecho simples e com o dispositivo construído acoplado na parte posterior da mesma. No entanto, no decorrer do desenvolvimento algumas mudanças estruturais foram importantes para melhorar a implementação prática. Essas diferenças e alterações são tratadas tópico 6.2 do capítulo 6.

O software utilizado para desenhar os circuitos foi o CIRCUITLAB, um software online simples, mas que permite esboçar as ligações entre os componentes de forma a prever como devem ser feitas as ligações, minimizando o risco de queimar ou danificar algo. O primeiro circuito esboçado foi o dos LED's, conforme ilustrado na figura 5.7.

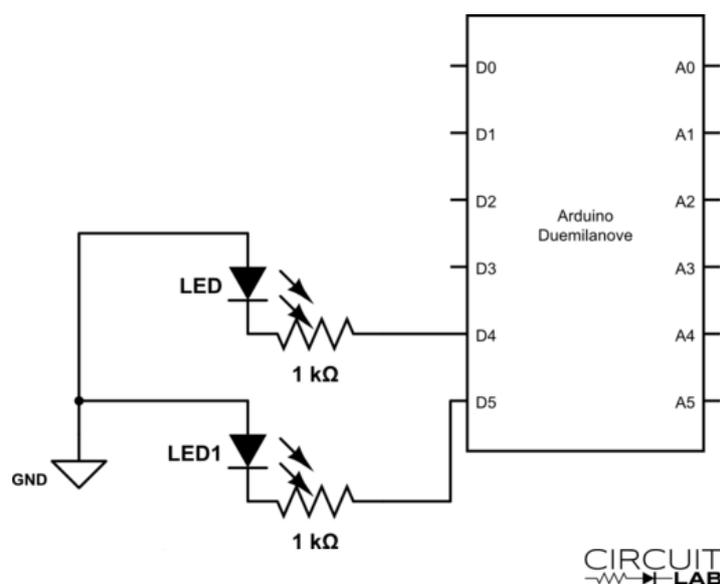


Figura 5. 7 - Circuito LEDs.

(Fonte: Autor)

O circuito contendo os LED's é o mais simples do projeto, tendo somente os 2 LED's verde e vermelhos, os resistores 1 k $\Omega$  e o Arduino. Na figura 5.4, o catodo do LED é o lado aonde tem uma barra horizontal enquanto que o anodo é o lado contrário.

Na figura 5.8, pode-se visualizar as ligações feitas entre o push button e o Arduino. Conforme indicado, para o botão ser alimentado, ele precisa necessariamente ser ligado em uma fonte de + 5V, no Arduino e no GND.

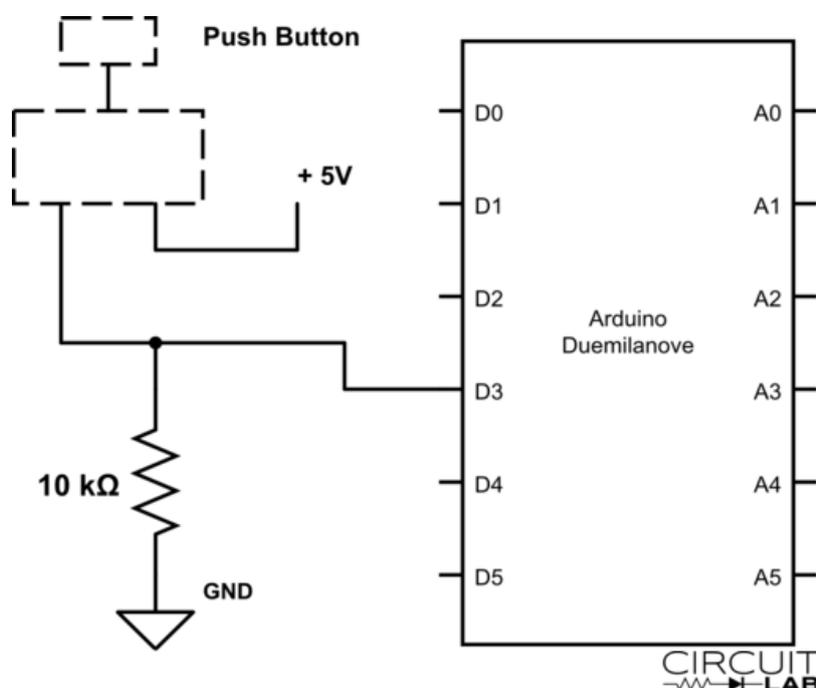


Figura 5. 8 - Circuito Push Button.

(Fonte: Autor)

A figura 5.9 ilustra a ligação feita entre Arduino, o relé e o fecho eletromagnético. Para fazer a alimentação do relé 5V com o Arduino, é necessário o uso de um diodo e um transistor. Nesse caso foi utilizado o diodo 1N4148 e o transistor BC547, conforme explicitado no tópico 5.5. A figura 5.10 indica todos esses circuitos juntos formando o circuito final.

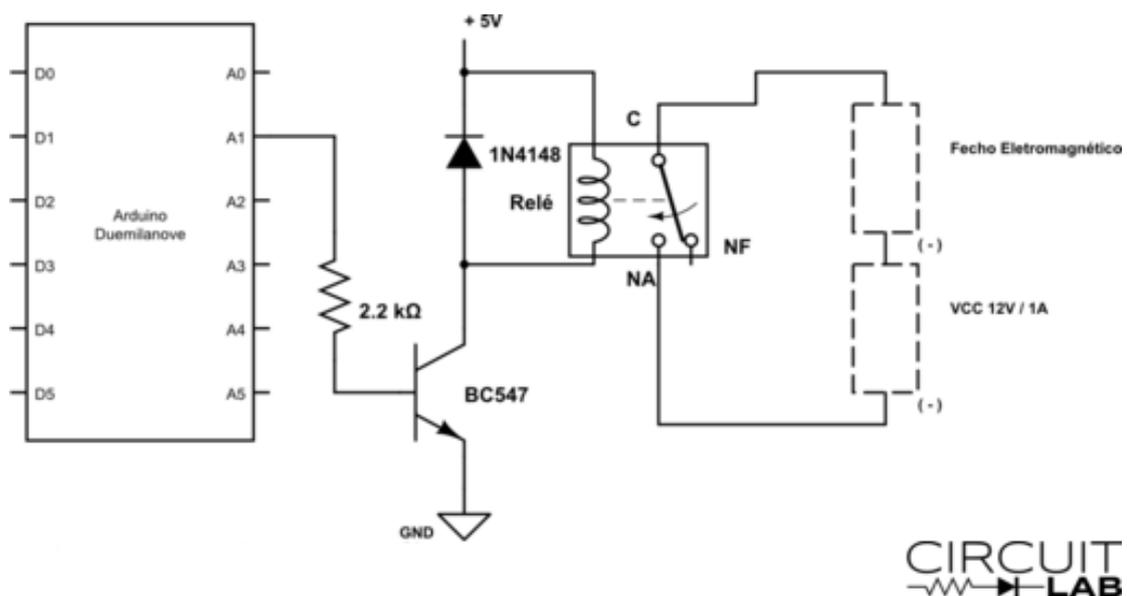


Figura 5. 9 - Circuito Relé + Fecho + VCC.

(Fonte: Autor)

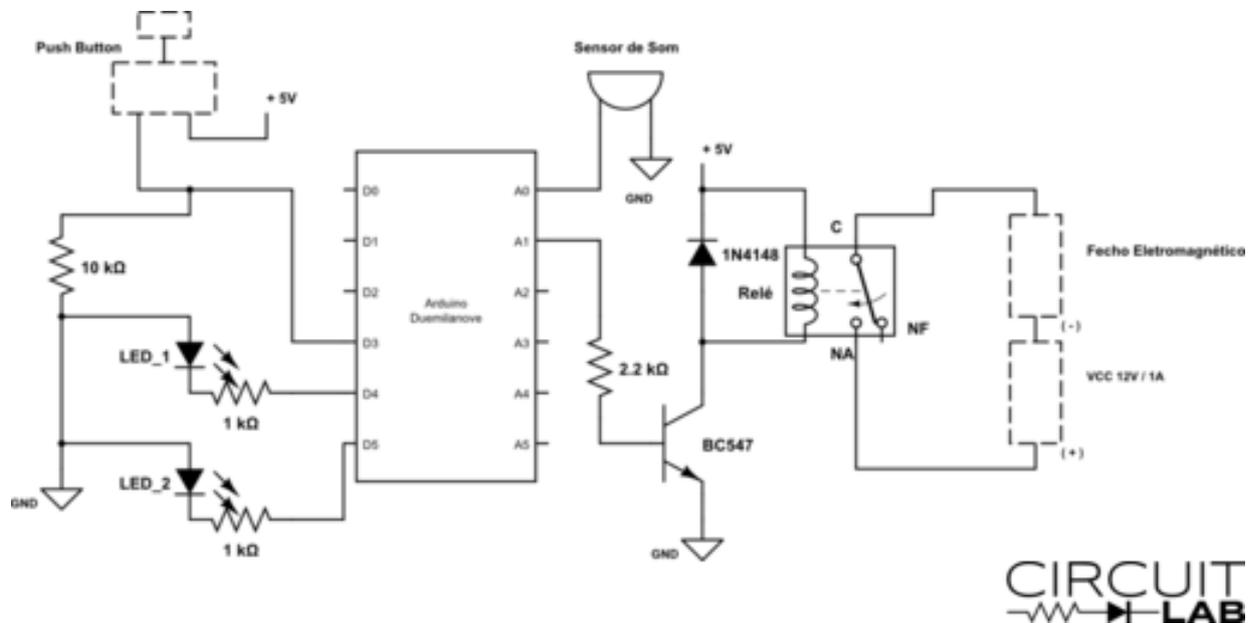
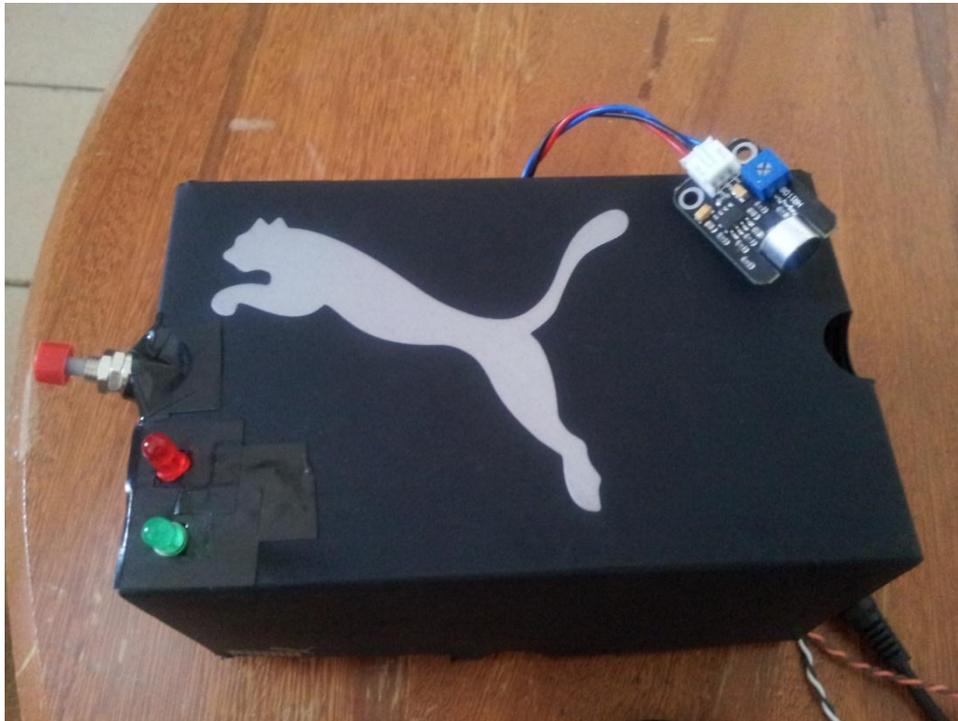


Figura 5. 10 - Circuito final.

(Fonte: Autor)

Por fim, a caixa onde o protótipo foi armazenado tem um tamanho de 22 cm x 12 cm x 3 cm e foi fixada na parte posterior da porta através de um velcro, colado na porta e na caixa.



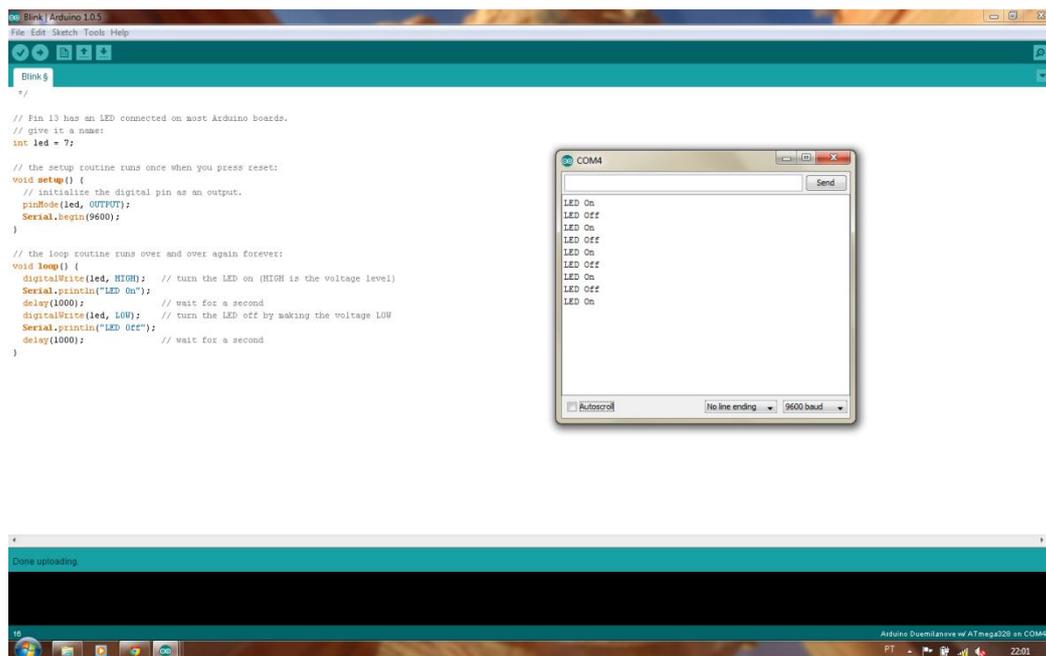
**Figura 5. 11 - Protótipo.**

**(Fonte: Autor)**

#### **5.4 – Testes e simulações**

Esta fase do trabalho teve como objetivo o teste e comprovação do funcionamento do projeto. As simulações descrevem todas as possibilidades de configuração e funcionalidades que o projeto possui para atender as necessidades do usuário. Os testes foram divididos em duas partes: os testes de cada componente separado e teste do protótipo completo.

Na primeira parte, os teste foram realizados através de 3 códigos-fontes, em que visavam obter respostas de cada componente separadamente do conjunto completo.



**Figura 5. 12 - Teste LEDs.**

**(Fonte: Autor)**

Os primeiros componentes testados foram os LEDs. O objetivo deste teste era bem simples, a cada intervalo de tempo definido, o LED iria acender e apagar, sempre mostrando seu status para o usuário pelo o Serial Monitor. O código-fonte utilizado pode ser visualizado na figura 5.12.

O próximo componente testado foi o push button. Nesse teste, foi utilizado um código em que quando o botão fosse pressionado, um LED acenderia e quando solto, o LED apagava. Entre essas transições entre botão/LED, era escrito na Serial Monitor "Botão On/LED On" para quando o botão estivesse pressionado e "Botão Off/LED Off" para quando o botão estivesse solto.

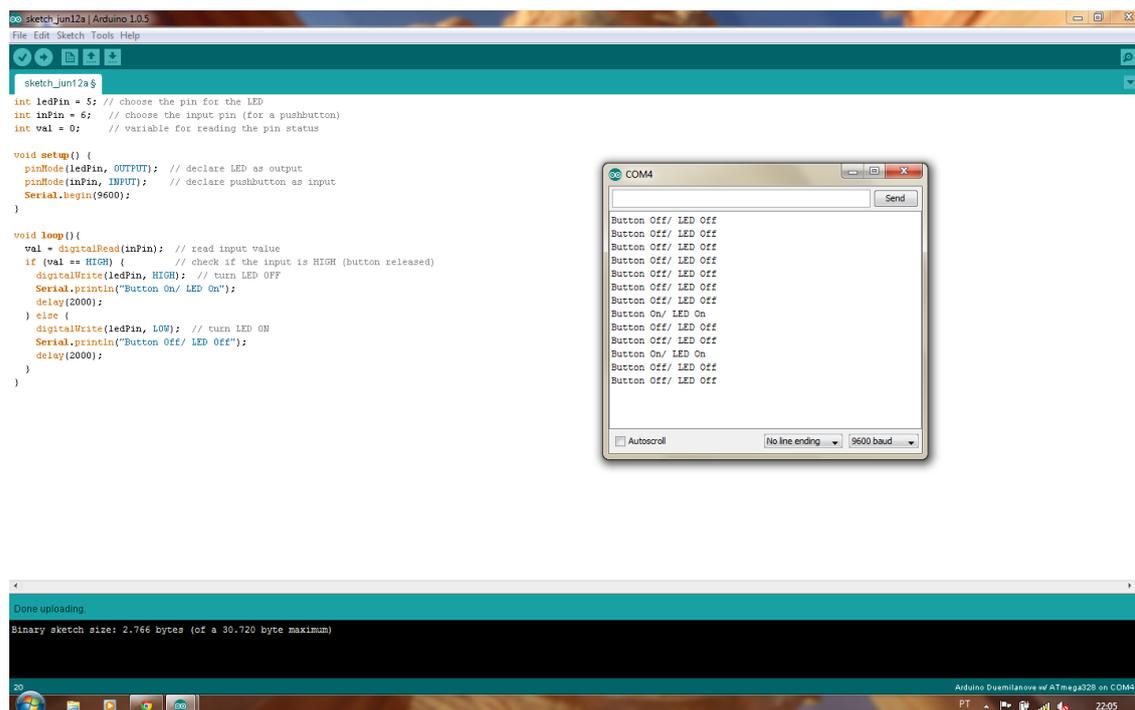
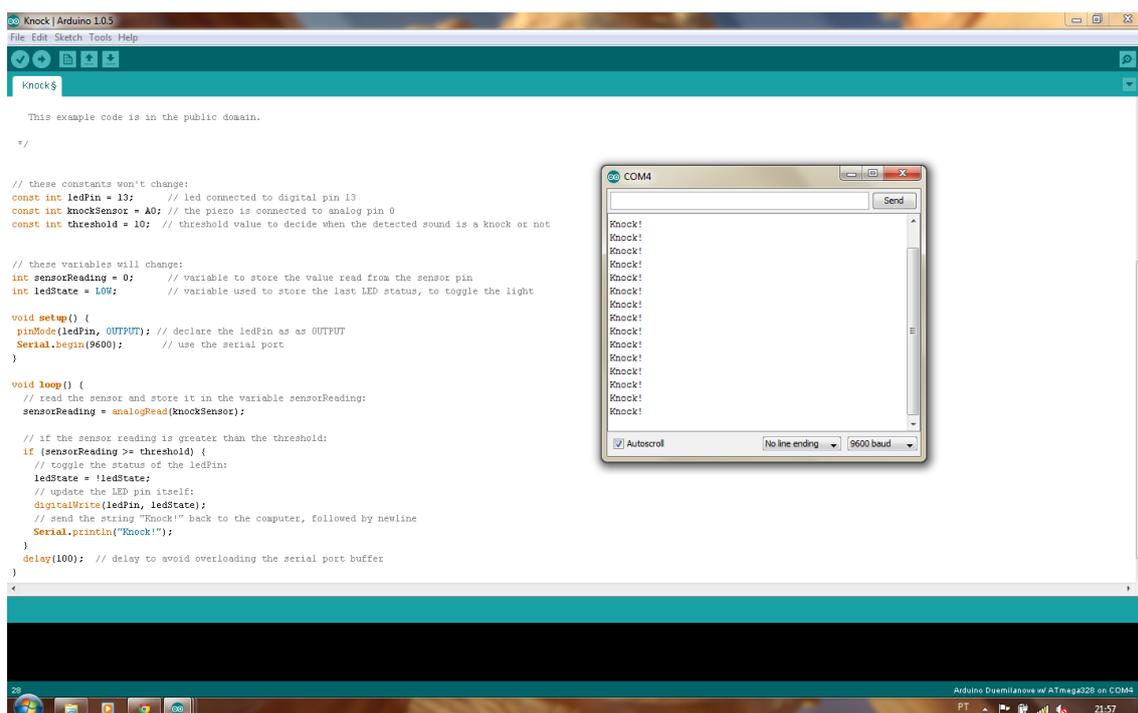


Figura 5. 13 - Teste Push Button.

(Fonte: Autor)

O terceiro teste foi o sensor de som. Como é um componente bastante sensível, esse teste precisou ser realizado quando não houvesse ninguém por perto, de forma a captar apenas a intensidade sonora provida pelo autor. Para tanto, foi realizado um código em que utilizava o sensor junto com um LED. Assim que a placa era ligada, o sensor fica à espera de uma intensidade sonora maior que o mínimo definido no código. Quando o som fornecido era maior ou igual à intensidade mínima definida, o LED alterava seu status e uma mensagem escrita “Knock!” era indicada na Serial Monitor.



**Figura 5. 14 - Teste Sensor de Som.**

(Fonte: Autor)

Após realizar os testes separados, o próximo passo foi testar o funcionamento do protótipo completo. Como a funcionalidade do projeto passa pelo bom funcionamento dos componentes, basicamente o código-fonte completo é a junção de todos os testes realizados anteriormente. Ao todo, foram realizados 4 testes separados, para comprovar o funcionamento de cada componente, e 1 teste completo, com todos os equipamentos juntos e acoplados a porta.

Dessa forma, como todos os testes iniciais se mostraram consistentes, as únicas alterações necessárias foram nas variáveis que limitam o reconhecimento dos ruídos externos como batidas na porta e nas variáveis que indicam a máxima quantidade de sons que o sistema detecta como sendo apenas uma batida.

## 5.5 – Problemas Encontrados

Inicialmente, o projeto foi planejado utilizando uma porta com um fecho simples, daqueles encontrado em portas de banheiro, uma placa de prototipagem Arduino, um piezo elétrico e um motor com caixa de rotação. Entretanto, no decorrer do desenvolvimento houve alguns problemas quanto à escolha desses componentes. A primeira mudança ocorreu com a substituição do piezo elétrico com o sensor de som analógico, apenas pela facilidade de mercado. A segunda alteração foi no uso de um relé 5V ao invés do motor com caixa de redução. Essa troca foi necessária pela dificuldade em acoplar o motor junto ao fecho comum instalado na porta. No embalo dessa troca do motor pelo relé, foi necessária, também, a troca do fecho simples para um fecho eletromagnético, que pudesse ser estimulado pelo relé.

Devido o problema inesperado descrito acima, o circuito teve que ser praticamente todo refeito, pois a placa implementada para as ligações entre o Arduino e o motor não era adequada para utilização com o relé.

Outra parte que deu trabalho foi à aquisição dos componentes. Apesar de utilizar componentes simples, os fornecedores não possuíam todos eles, e com isso alguns componentes tiveram que ser comprados fora de Brasília.

Como é um protótipo que foi ligado basicamente todo através de solda, uma das dificuldades também foi no momento de soldar os componentes do relé na placa perfurada e fazer a conexão do circuito com a fonte de alimentação, de modo a utilizar apenas uma fonte tanto para o fecho eletromagnético quanto para o circuito.

## 5.6 – Orçamento do Projeto

Quadro 5. 1 - Custo do Projeto. (Fonte: Autor)

Item	Custo Unitário	Quantidade	Custo Total
Arduino Duemilanove	R\$ 69,00	1	R\$ 69,00
Fecho Eletromagnético	R\$ 56,00	1	R\$ 56,00
Sensor Analógico de Som	R\$ 79,00	1	R\$ 79,00
Maquete da porta	R\$ 80,00	1	R\$ 80,00
LED difuso (5mm)	R\$ 0,11	4	R\$ 0,44
Resistor de Filme de Carbono 1K $\Omega$ - 1/4W	R\$ 0,04	4	R\$ 0,16
Resistor de Filme de Carbono 2K2 $\Omega$ - 1/4W	R\$ 0,04	2	R\$ 0,08
Resistor de Filme de Carbono 10K $\Omega$ - 1/4W	R\$ 0,04	2	R\$ 0,08
Transistor BC547	R\$ 0,11	2	R\$ 0,44
Placa Universal Perfurada Lig. (2,8x12,9cm)	R\$ 1,99	1	R\$ 1,99
Chave Push Button 2P (vermelho)	R\$ 2,37	1	R\$ 2,37
Diodo 1N4148	R\$ 0,25	2	R\$ 0,50
Relé 5 V	R\$ 5,00	2	R\$ 5,00
Fonte de Alimentação 12 V/ 1 A	R\$ 20,00	1	R\$ 20,00
<b>Total</b>			R\$ 315,06 US\$ 139,57

## 5.7 – Protótipo Final



Figura 5. 15 - Protótipo Final.

(Fonte: Autor)

## **CAPÍTULO 6 – CONSIDERAÇÕES FINAIS**

Neste capítulo são apresentadas as considerações finais e as propostas para futuros projetos, que podem usar este como referencial teórico.

### **6.1 – Conclusões**

Foi desenvolvido neste projeto um dispositivo para controle de abertura de uma porta. Com os testes realizados simulando uma aplicação real, o projeto funcionou conforme o previsto, e assim, pode ser usado em situações reais.

O compromisso desse projeto é criar um equipamento em que o usuário possa restringir o acesso. Para isso, o sistema ficará monitorando 24 horas todo o tipo de som que acontecer na porta.

Acredito que o objetivo foi alcançado, pois o equipamento funcionou conforme o planejado. Ocorreram alguns imprevistos durante o processo de confecção, mas nada que comprometesse o resultado final.

O Arduino atendeu completamente as funções do protótipo e ainda poderiam ser adicionados outros componentes caso futuramente seja realizado um projeto similar. O circuito foi projetado com sucesso e o funcionamento do mesmo foi excelente atendendo as expectativas criadas no início do projeto.

### **6.2 – Propostas para Trabalhos Futuros**

Uma modificação a ser feita no projeto seria adicionar mais um sensor para a o envio de mensagem SMS para o celular do usuário cadastrado no sistema. Com isso, cada vez que ocorra uma batida errada, o usuário receberá um aviso sobre a tentativa frustrada.

O modem sugerido para isso seria o GSM SIM900. Além dessa função básica, ele possui outras funções que contribuiriam ainda mais, tais como: caso

o usuário queira monitorar a sua casa em um determinado momento, enviaria uma SMS para o dispositivo e receberia a resposta sobre o status do momento.

Outra proposta viável seria trocar o Arduino por um microcontrolador da família PIC. O 16F877A é um microcontrolador que possui vários pinos que facilitaria muito para aumentar o leque de funções que o microcontrolador pode assumir.

A implementação de um sistema de alarme também seria muito interessante, pois aumentaria ainda mais a segurança tanto do lado interno quanto do externo da porta. Poderia ser feita uma integração entre o dispositivo construído, o modem GSM e o sistema de alarme. Em uma aplicação real, um exemplo viável seria a aplicação em residências ou em ambientes corporativos.

## REFERÊNCIAS BIBLIOGRÁFICAS

ABREU, R. S. **Automação Residencial: um pouco da história.** Disponível em: < [http://www.aureside.org.br/publicacoes/download/automacao\\_residencial.zip](http://www.aureside.org.br/publicacoes/download/automacao_residencial.zip) > Acessado em: 23 de mar. 2013.

ARDUINO. **Arduino Duemilanove.** Disponível em: < <http://arduino.cc/en/Main/arduinoBoardDuemilanove> > Acessado em: 09 de abr. 2013.

BRASIL ESCOLA. **Eletromagnetismo.** Disponível em: < <http://www.brasilecola.com/fisica/eletromagnetismo.htm> > Acessado em: 03 de jun. 2013.

CESPE/UnB. **Polícia Federal prende fraudadores de concursos públicos.** Disponível em: < <http://www.cespe.unb.br/NoticiasHTML/LerNoticia.asp?IdNoticia=341> > Acessado em: 21 de mar. 2013.

CIRCUITLAB. **Sketch, simulate and share schematics.** Disponível em: < <http://www.circuitlab.com> > Acessado em: 04 de jun. 2013.

BONJORNO, J. R.; RAMOS, C. M. **Física 2: Termologia, Óptica Geométrica Ondulatória.** 1ª. Ed. São Paulo: FTD, 1992.

GIMENEZ, S. P. **Microcontroladores 8051.** 1ª. Ed. São Paulo: Pearson, 2005.

GROOVER, M. P. **Automation, Production Systems and Computer Integrated Manufacturing Systems.** 2ª. Ed. Prentice Hal, 1987.

HARPER, R. **Inside the Smart Home.** Bristol: Springer Verlag, 2003.

INFOESCOLA. **O que são Linguagens de Programação.** Disponível em: < <http://www.infoescola.com/informatica/o-que-sao-linguagens-de-programacao/> > Acessado em: 11 de maio. 2013.

INSTITUTO NEWTON C. BRAGA. **Uso de Relés em Robótica e Mecatrônica.** Disponível em: < <http://www.newtoncbraga.com.br/index.php/robotica/1733-mec037> > Acessado em: 03 de jun. 2013.

INSTITUTO NEWTON C. BRAGA. **Tudo sobre Relés.** Disponível em: < <http://www.newtoncbraga.com.br/index.php/como-funciona/597-como-funcionam-os-reles> > Acessado em: 04 de jun. 2013.

JUCA, S. **Apostila de Microcontroladores e Periféricos.** Disponível em: < <http://pt.scribd.com/doc/38537364/ApostilaCPIC> > Acessado em: 09 de jun. 2013.

MORAES, C. C. de; CASTRUCCI, P. L. **Engenharia de Automação Industrial.** 2ª. Ed. Rio de Janeiro: LTC, 2007.

MULTILÓGICA-SHOP. **Arduino Duemilanove com ATmega328.** Disponível em: < <http://multilogica-shop.com/Arduino-duemilanove> > Acessado em: 12 de abr. 2013.

McROBERTS, M. **Arduino Básico.** 1ª. Ed. São Paulo: Novatec Editora Ltda, 2011.

O SETOR ELÉTRICO. **Relés e contadores.** Disponível em: < <http://www.osetoreletrico.com.br/web/a-revista/edicoes/169-reles-e-contadores.html> > Acessado em: 04 de jun. 2013.

PATSKO, L. F. **Tutorial Controle de Relés.** Disponível em: < [http://www.maxwellbohr.com.br/downloads/robotica/mec1000\\_kdr5000/tutorial\\_eletronica\\_-\\_rele.pdf](http://www.maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_eletronica_-_rele.pdf) > Acessado em: 04 de jun. 2013.

PEREIRA, F. **Microcontroladores PIC: Programação em C.** 7ª. Ed. São Paulo: Érica, 2007.

ROSÁRIO, J. M. **Princípios de Mecatrônica.** 7ª. Ed. São Paulo: Pearson Brasil, 2005.

SABER ELETRÔNICA. **Placas de Circuito Impresso.** Disponível em: < <http://www.sabereletronica.com.br/secoes/leitura/1494> > Acessado em: 29 de abr. 2013.

SILVEIRA, J. A. **Arduino Cartilha para Programação em C.** Disponível em: < [http://www.ordemnatural.com.br/pdf-files/CartilhadoArduino\\_ed1.pdf](http://www.ordemnatural.com.br/pdf-files/CartilhadoArduino_ed1.pdf) > Acessado em: 06 de jun. 2013.

SOARES, P. A. T; FERRANO, N. G; SANTOS, J. I. C. **Aulas de Física.** 1ª Ed. São Paulo: Atual, 1984.

SRA ENGENHARIA BLOG. **Histórico da Automação Residencial.** Disponível em: < [http://sraengenharia.blogspot.com.br/2013/01/historico-da-automacao-residencial\\_10.html](http://sraengenharia.blogspot.com.br/2013/01/historico-da-automacao-residencial_10.html) > Acessado em: 10 de abr. 2013.

THOMAZINI, D; ALBURQUERQUE, P. U. B. de. **Sensores Industriais: Fundamentos e Aplicações.** 4ª Ed. Érica, 2005.

UOL EDUCAÇÃO. **MEC divulga na internet provas do ENEM 2009 que vazaram.** Disponível em: < <http://educacao.uol.com.br/ultnot/2009/10/01/ult1811u397.jhtm> > Acessado em: 21 de mar. 2013.

WEG. **Motores Elétricos.** Disponível em: <  
<http://www.coe.ufrj.br/~richard/Accionamentos/Catalogo%20de%20Motores.pdf>.  
> Acessado em: 08 de abr. 2013.

WIKIPEDIA. **Arduino.** Disponível em: < <http://pt.wikipedia.org/wiki/Arduino> >  
Acessado em: 12 de abr. 2013.

## APÊNDICE A – CÓDIGO FONTE DO PROJETO

```
/*=====
Nome: Paulo Henrique de Oliveira Silva
Matrícula: 2081763/0
Curso: Engenharia de Computação
Disciplina: Projeto Final
Data: 1º semestre/2013
=====*/

/*===== INÍCIO - DIRETIVAS PRÉ-COMPILAÇÃO=====*/

// Definições de constantes

const int microfone = A5;

const int botao = 6;

const int rele = A4;

const int led_vermelho = 5;

const int led_verde = 7;

const int min_sinal = 350;

const int reject_value = 25;

const int average_reject_value = 15;

const int intervalo_batidas = 150;

const int intervalo_rele = 650;
```

```
const int max_qtd_batidas = 20;

const int batida_gravada = 1200;

// Definições de Variáveis

int batida_secreta[max_qtd_batidas] = {100, 100, 100};

int sensor_leitura[max_qtd_batidas];

int ultima_batida_lida = 0;

int botao_status = false;

/*===== FIM - DIRETIRVAS PRÉ-COMPILAÇÃO=====*/

/*=====INÍCIO - Função Principal=====*/

void setup(){

  pinMode(rele, OUTPUT);

  pinMode(led_vermelho, OUTPUT);

  pinMode(led_verde, OUTPUT);

  pinMode(botao, INPUT);

  digitalWrite(led_verde, HIGH);

  Serial.begin(9600);

  Serial.println("Iniciando");

}

void loop(){

  // Escuta qualquer som na porta
```

```
ultima_batida_lida = analogRead(microfone);

if(digitalRead(botao) == HIGH){

    botao_status = true;

    digitalWrite(led_vermelho, HIGH);

}else{

    botao_status = false;

digitalWrite(led_vermelho, LOW);

}

if(ultima_batida_lida >= min_sinal){

    leituraNovaBatida();

}

}

/*=====FIM - Função Principal=====*/

/*=====INÍCIO - Função para gravar o tempo das batidas =====*/

void leituraNovaBatida(){

    Serial.println ("Batida detectada");

    int n = 0;

    for (n = 0; n < max_qtd_batidas; n++){

        sensor_leitura[n] = 0;
```

```
}

int incremento = 0;

int tempo_inicial = millis();

int tempo_real = millis();

digitalWrite(led_verde, LOW);

if(botao_status == true){

    digitalWrite(led_vermelho, LOW);

}

delay(intervalo_batidas);

digitalWrite(led_verde, HIGH);

if(botao_status == true){

digitalWrite(led_vermelho, HIGH);

}

do{

// Escuta a nova batida

    ultima_batida_lida = analogRead(microfone);

    if(ultima_batida_lida >= min_sinal){

        // Grava o intervalo

        Serial.println("Batidas.");

        tempo_real = millis();
```

```
sensor_leitura[incremento] = tempo_real - tempo_inicial;

incremento = incremento + 1;

tempo_inicial = tempo_real; // reseta a variável de tempo para a próx. batida

digitalWrite(led_verde, LOW);

if(botao_status == true){

    digitalWrite(led_vermelho, LOW);

}

delay(intervalo_batidas);

digitalWrite(led_verde, HIGH);

if(botao_status == true){

    digitalWrite(led_vermelho, HIGH);

}

}

tempo_real = millis();

}while((tempo_real < batida_gravada) && (incremento < max_qtd_batidas));

// Com a batida gravada, agora ela é validada

if(botao_status == false){

    if(batidaValidacao() == true){

destrancarPorta();

    }else{

        Serial.println("Erro!");

    }

}

}
```

```
digitalWrite(led_vermelho, LOW);

    for(n = 0; n < 4; n++){

        digitalWrite(led_vermelho, HIGH);

        delay(100);

        digitalWrite(led_vermelho, LOW);

        delay(100);

    }

    digitalWrite(led_verde, HIGH);

}

}else{

batidaValidacao();

    // O piscamento dos LEDs é pra mostrar que a gravação/validação foi
completada

    Serial.println("Nova batida gravada!");

digitalWrite(led_vermelho, LOW);

    digitalWrite(led_verde, HIGH);

for(n = 0; n < 3; n++){

    delay(100);

digitalWrite(led_vermelho, HIGH);

    digitalWrite(led_verde, LOW);

    delay(100);

    digitalWrite(led_vermelho, LOW);

digitalWrite(led_verde, HIGH);
```

```
    }  
  }  
}  
  
/*=====FIM - Função para gravar o tempo das batidas =====*/  
  
/*=====INÍCIO - Função para destrancar a porta=====*/  
void destrancarPorta(){  
  Serial.println("Porta destrancada!");  
  
  int n = 0;  
  
  // Estimula o relé  
  digitalWrite(rele, HIGH);  
  digitalWrite(led_verde, HIGH);  
  delay(intervalo_rele);  
  digitalWrite(rele, LOW);  
  
  // Pisca o led verde por uns segundos  
  for (n = 0; n < 5; n++){  
    digitalWrite(led_verde, LOW);  
    delay(100);  
    digitalWrite(led_verde, HIGH);  
  }  
  delay(100);  
}  
  
/*=====FIM - Função para destrancar a porta=====*/
```

```
/*=====INÍCIO - Função para validação da batida=====*/
```

```
boolean batidaValidacao(){
```

```
    int n = 0;
```

```
    // faz uma simples verificação inicial.
```

```
    int count = 0;
```

```
    int batida_secreta_count = 0;
```

```
    int intervalo_max_qtd_batidas = 0;
```

```
    for(n = 0; n < max_qtd_batidas; n++){
```

```
        if(sensor_leitura[n] > 0){
```

```
            count++;
```

```
        }
```

```
        if(batida_secreta[n] > 0){
```

```
            batida_secreta_count++;
```

```
        }
```

```
        if(sensor_leitura[n] > intervalo_max_qtd_batidas){
```

```
            intervalo_max_qtd_batidas = sensor_leitura[n];
```

```
        }
```

```
    }
```

```
    if(botao_status == true){
```

```
        for(n = 0; n < max_qtd_batidas; n++){
```

```
batida_secreta[n] = map(sensor_leitura[n], 0, intervalo_max_qtd_batidas, 0,
100);

}

digitalWrite(led_verde, LOW);

digitalWrite(led_vermelho, LOW);

delay(100);

digitalWrite(led_verde, HIGH);

digitalWrite(led_vermelho, HIGH);

delay(50);

for(n = 0; n < max_qtd_batidas; n++){

digitalWrite(led_verde, LOW);

    digitalWrite(led_vermelho, LOW);

// Se houver delay:

    if(batida_secreta[n] > 0){

        delay(map(batida_secreta[n], 0, 100, 0, intervalo_max_qtd_batidas));

digitalWrite(led_verde, LOW);

        digitalWrite(led_vermelho, LOW);

    }

    delay(50);

}

return false; // Não foi possível destrancar a porta quando foi gravada uma
nova batida

}

if(count == batida_secreta_count){
```

```
    return true;
}
else{
    return false;
}

/* Agora é feita uma comparação entre os intervalos das batidas.
*/

int diferenca_intervalo = 0;

int tempo = 0;

for(n = 0; n < max_qtd_batidas; n++){

    sensor_leitura[n] = map(sensor_leitura[n], 0, intervalo_max_qtd_batidas, 0,
100);

    tempo = abs(sensor_leitura[n] - batida_secreta[n]);

if(tempo > reject_value){

    return false;

}

    diferenca_intervalo = diferenca_intervalo + tempo;

}

if((diferenca_intervalo/batida_secreta_count) > average_reject_value){

    return false;

}

return true;

}
```

/\*=====FIM - Função para validação da batida=====\*/