



CENTRO UNIVERSITÁRIO DE BRASÍLIA – UNICEUB

FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS – FATECS

CURSO DE ENGENHARIA DE COMPUTAÇÃO

GABRIEL FERREIRA DO NASCIMENTO

**SISTEMA DE SEGURANÇA E CONTROLE DE ACESSO À ENTRADA POR
SEQUÊNCIA DE BATIDAS**

Brasília

2019

GABRIEL FERREIRA DO NASCIMENTO

**SISTEMA DE SEGURANÇA E CONTROLE DE ACESSO À ENTRADA POR
SEQUÊNCIA DE BATIDAS**

Trabalho de Conclusão de Curso apresentado à Banca examinadora do curso de Engenharia de Computação da Faculdade de Tecnologia e Ciências Sociais Aplicadas – FATECS – Centro Universitário de Brasília – UniCEUB – como requisito para obtenção do Certificado de Conclusão de Curso de Engenharia de Computação.

Orientador: Prof. MSc. Francisco Javier Obaldia Diaz

Brasília

2019

GABRIEL FERREIRA DO NASCIMENTO

**SISTEMA DE SEGURANÇA E CONTROLE DE ACESSO À ENTRADA POR
SEQUÊNCIA DE BATIDAS**

Trabalho de Conclusão de Curso apresentado à Banca examinadora do curso de Engenharia de Computação da Faculdade de Tecnologia e Ciências Sociais Aplicadas – FATECS – Centro Universitário de Brasília – UniCEUB – como requisito para obtenção do Certificado de Conclusão de Curso de Engenharia de Computação.

Orientador: Prof. MSc. Francisco Javier Obaldia Diaz

BRASÍLIA, 28 DE JUNHO DE 2019

BANCA EXAMINADORA

Prof. MSc. Francisco Javier Obaldia Diaz
Orientador

Prof. MSc. Ivandro da Silva Ribeiro

Prof. MSc. Flávio Antonio Klein

AGRADECIMENTOS

Agradeço primeiramente aos meus pais que, com muito esforço, puderam me dar uma educação de qualidade. Prezando sempre pelo meu futuro e sucesso. Aos meus amigos que, durante todos os anos de faculdade, estiveram comigo nos bons momentos e nas dificuldades. Aos professores que dividiram comigo seus conhecimentos e experiências, guiando meu caminho na formação profissional. Em especial aos professores Luciano Duque e Francisco Javier, por sua orientação sempre que necessária. Sou muito agradecido, também, a minha amada Rita por sempre estar comigo nos momentos de dificuldade e por sempre ser alguém em quem posso contar.

CITAÇÃO

“Siga, sempre nos trilhos, rumo ao suco de laranja.”

- Meluk Estúdio

SUMÁRIO

1 INTRODUÇÃO	10
1.1 Motivação	11
1.2 Objetivos	11
1.3 Justificativa e relevância do trabalho	12
1.4 Trabalhos correlatos	12
1.5 Escopo do trabalho	13
1.6 Resultados esperados	13
1.7 Estrutura do trabalho	13
2 REFERENCIAL TEÓRICO	15
2.1 Eletromagnetismo	15
2.1.1 Relé	16
2.1.2 Fecho eletromagnético	17
2.2 Microcontroladores	18
2.2.1 Arduino	19
2.3 Processamento digital de sinais	20
2.3.1 Módulo Sensor Analógico de Som	21
2.4 Bluetooth	22
2.4.1 Módulo bluetooth serial HC-06	23
2.5 Linguagem de programação	24
2.5.1 Linguagem de programação C	25
2.6 Metodologia	25
3. DESENVOLVIMENTO DO SISTEMA	28
3.1 Modelagem do sistema	29
3.1.1 Diagrama de implantação	29
3.1.2 Fluxograma de processos de negócios	30
3.2 Levantamento e obtenção de requisitos	32
3.3 Desenvolvimento do circuito	34
3.4 Desenvolvimento do software	40
3.5 Montagem do sistema	45
4 APLICAÇÃO DA SOLUÇÃO COM RESULTADOS	47
4.1 Teste do circuito digital	47
4.1.1 Teste dos LEDs	47
4.1.2 Teste do relé	48
4.1.3 Teste do módulo sensor analógico de som	50
4.1.4 Teste do módulo bluetooth HC-06	51

4.2 Teste do software	53
4.3 Teste do sistema	55
5 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS	57
5.1 Sugestões para trabalhos futuros	57
REFERÊNCIAS	59

RESUMO

Este documento apresenta a proposta de desenvolvimento de um sistema alternativo de controle de acesso capaz de permitir a entrada a um ambiente por meio de uma sequência de batidas realizada em uma porta. Dando uma opção alternativa ao meio convencional de acesso, a utilização de chaves, uma vez que a indisponibilidade das chaves em decorrência de roubos, esquecimentos entre vários outros fatores, faz com que o acesso a qualquer ambiente guardado por portas se torne impossível. Logo, usando como base os estudos sobre sistemas de controle de acesso, buscou-se desenvolver um dispositivo que seja capaz de captar uma sequência de batidas através de um sensor analógico de som e compará-las com uma senha de acesso pré-definida. No caso da sequência ser similar a senha de acesso, o dispositivo irá acionar um relé que energiza um fecho eletromagnético, liberando a trava da porta. Em paralelo, o sistema faz uso da comunicação, via conexão bluetooth, com aparelhos celulares. A qual realiza a alteração da senha de acesso e informa quando alguém tenta obter acesso, sem sucesso, a entrada. Logo, espera-se obter um dispositivo que sirva tanto como uma alternativa secundária para o acesso de ambientes ou até como o principal meio.

Palavras-chave: Automação, Arduino, Relé, Bluetooth, Sensor Analógico de Som

ABSTRACT

This document presents as a proposal the development of an alternative access control system that is capable of ensuring the entry on any place using a sequence of knocks on the door. The idea was born to give an alternative to the conventional way of access, the use of keys to open the door. Due to the fact that the keys sometimes may be unavailable because of someone forgetting where they put it, because of thefts and lots of other factors that can make the key unavailable when trying to open the door. And the lack of the key make impossible to open the door and have access to the place. So, basing on the studies about access control systems, we tried to develop a device that is capable of opening doors using a sequence of knocks on a door, the device capitate the knocks using an analog sound sensor and compare them with the pre established password. When the sequence is similar to the password the device trigger an relay tha power up an electromagnetic lock and unlock the door. Also , the system makes use of bluetooth connexion, using this technology it informs the owner of the access control system when someone tries to open the door and fails and to make possible changing the password. Given the the information presented above is expected to get a device that will serve as an alternative to accessing places or even making this product the main way for accessing the door.

Keywords: Automation, Arduino, Relay, Bluetooth, Analog Sound Sensor

1 INTRODUÇÃO

Controlar o acesso a determinado ambiente é um problema muito antigo e ainda não foi pensada uma solução 100% eficiente para ele. Desde a criação da primeira propriedade privada as pessoas têm utilizado sistemas de controle de acesso. Primeiramente foram criados sistemas rudimentares, como cercas para limitar o acesso e atualmente existem sistemas muito complexos, sendo que o sistema de controle de acesso mais utilizado mundialmente são as portas.

As portas revolucionaram o controle de acesso. Por um lado é uma forma prática de controlar o acesso a ambientes e por outro são relativamente baratas. Porém, problemas diários como chaves que quebram dentro da fechadura, esquecimento de chaves e até furtos das chaves que dariam acesso aos ambientes fazem com que a porta seja um sistema de controle de acesso muito falho, apesar de ser o mais utilizado hoje em dia.

Quando pensamos nos problemas que um sistema de controle de acesso tão prático e completo como as portas contém, percebemos que o fator núcleo deles se encontra na necessidade de existir um objeto, no caso a chave, que faria o acesso a um ambiente restringido por uma porta ser possível. Visto que grande parte da população sofre diariamente com o problema de esquecer suas chaves ou de ter algum tipo de problema com o objeto chave, o presente trabalho visa mostrar uma alternativa com custos baixos, que além de acabar com a necessidade de carregar um objeto para todos os lugares para conseguir ter acesso a um ambiente, também aumentaria e muito a própria segurança desses ambientes.

A alternativa apresentada será o acesso através de padrões de batidas realizadas na porta. Esse trabalho se trata do desenvolvimento de um dispositivo que, instalado na porta, tem a finalidade de captar batidas realizadas. Após isso, as batidas serão comparadas a um padrão pré definido e nomeado como “senha de acesso”, a fim de possibilitar o controle do acesso.

O dispositivo desenvolvido nesse projeto teria como aplicações tornar o arrombamento de uma porta muito mais difícil, uma vez que a abertura dela teria que ser por padrão de batida, o que é muito mais difícil de se descobrir do que clonar um objeto como uma chave. Além de tornar nula a possibilidade de alguém

não conseguir entrar em algum local por ter sido furtado ou esquecido a chave em algum local.

1.1 Motivação

A motivação da realização deste trabalho surgiu a partir da leitura da monografia “Sistema de Segurança de Tranca de Porta e Controle de Acesso” de Paulo Henrique de Oliveira Silva. Buscando dar continuidade ao projeto em questão, através do proposto em seu tópico de trabalhos futuros. Em que o autor sugere a integração do sistema com aparelhos smartphones.

Trabalhos acadêmicos muitas vezes não apresentam conclusões definitivas e livres de pontos sujeitos a possíveis melhorias. Por isso, o grande motivador deste trabalho é preencher as lacunas no desenvolvimento, nos resultados obtidos ou no conhecimento utilizado na monografia “Sistema de Segurança de Tranca de Porta e Controle de Acesso”. Assim alcançando um resultado mais completo.

1.2 Objetivos

O objetivo geral deste trabalho é desenvolver um dispositivo que controle o acesso a uma entrada, de maneira segura, através de uma sequência de batidas definida como correta. Essa sequência definida como correta deverá estar previamente configurada e será nomeada como “senha de acesso”. O dispositivo deve ser capaz de monitorar quando é realizada uma batida na porta da entrada, tratar esta informação capturada pelos sensores e depois executar a ação de liberar o acesso ou vetá-lo através de uma fechadura eletromagnética. Paralelo a isso um aplicativo de celular deverá enviar um relatório ao usuário como quando alguém errar a sequência de batidas e definir quando uma nova senha estiver sendo configurada.

Objetivos específicos:

- Levantar todo o material necessário para completa compreensão do desenvolvimento do sistema de controle de acesso;

- Elaborar o material escrito para transmitir o embasamento teórico do desenvolvimento prático;
- Desenvolver hardware capaz de capturar os sinais de batida, interpretar e executar o controle de acesso a porta;
- Permitir configuração do dispositivo de forma sem fio pelo celular do responsável;
- Enviar os dados de forma sem fio para o celular do responsável para acompanhamento do relatório de acesso a entrada;

1.3 Justificativa e relevância do trabalho

A AR (Automação Residencial) está cada vez mais presente em casas e apartamentos, transformando-os em residências inteligentes. A AURESIDE (Associação Brasileira de Automação Residencial) mostra, em um balanço realizado por eles, que o número de empresas associadas a ela cresceu de 6 para 50 nos últimos 20 anos. A associação também realizou uma projeção de crescimento anual global para o mercado de automação residencial de 11,36%.

A partir do momento em que você tem uma casa automatizada, é possível controlar aparelhos eletrônicos remotamente, maximizar medidas de segurança ou qualquer opção que preza o conforto. A tecnologia possibilita o aumento da qualidade de vida de uma pessoa. Podendo ser aplicada praticamente a todas as atividades que se desenvolvem dentro de uma residência. (Prudente, 2011)

O projeto se insere no contexto da domótica, com o objetivo de alcançar uma melhoria na segurança. Criando uma forma alternativa de acesso a uma entrada.

1.4 Trabalhos correlatos

A diversas aplicações que fazem uso de sons para desencadear tarefas a fim de facilitar o cotidiano das pessoas. Este grande número se deve principalmente às inúmeras combinações de sons possíveis para serem utilizados como comandos. Como por exemplo destas aplicações, uma lâmpada que acende ao bater de palmas (Monk, 2014) até um exemplo mais complexo como uma assistente digital (Schmitz, 2013).

1.5 Escopo do trabalho

O trabalho aqui desenvolvido limita-se a produção de um dispositivo. O dispositivo em questão deve ser capaz de capturar sinais sonoros através de um sensor analógico de som. Tratar estes sinais por meio de um microcontrolador posicionado em uma porta. Em que o microcontrolador deve dispor de conexão bluetooth, e uma interface de comunicação, a fim de receber comandos de um smartphone e enviar relatórios para o mesmo. Não fazendo parte do escopo deste trabalho o desenvolvimento de um aplicativo mobile. Sendo assim utilizada uma aplicação genérica para o envio de comandos e recebimento de relatórios.

Com base em todos os fatores abordados no parágrafo anterior, o dispositivo deve tomar a devida ação de acionar ou não um componente de relê responsável por energizar um fecho eletromagnético.

1.6 Resultados esperados

Com o dispositivo resultante deste trabalho instalado em uma porta de fecho eletromagnético, espera-se que um usuário configure no smartphone a sequência de batidas na porta que deve ser utilizada como “senha de acesso”. Após a configuração, o usuário estará apto a executar duas sequências de batidas na porta, uma idêntica a “senha de acesso” e outra diferente. Quando realizada a sequência idêntica a senha, a porta deve abrir. E para a sequência diferente da senha, a porta permanece fechada e um relatório alertando sobre a tentativa de acesso é enviado para o smartphone conectado.

1.7 Estrutura do trabalho

O desenvolvimento do trabalho é dividido em capítulos, expostos da seguinte maneira:

O capítulo 1 contendo a introdução aos assuntos abordados, motivação para a realização do trabalho, objetivos gerais e específicos a serem alcançados. Além da relevância do trabalho desenvolvido, seu escopo e outros trabalhos correlatos a este.

O capítulo 2 faz uso de referenciais teóricos descritos na literatura, publicações técnicas, descritivos de componentes e dispositivos que serão utilizados visando a resolução do problema proposto. Para isso, são descritas as técnicas, metodologias e componentes eletrônicos usados e o descritivo da solução proposta.

O capítulo 3 aborda o desenvolvimento do projeto. Efetuando uma descrição de cada etapa envolvida, baseadas em modelos-fluxogramas. Assim como a apresentação dos elementos utilizados em cada etapa.

O capítulo 4 inicia com uma descrição sucinta das condições onde vai se realizada a aplicação do projeto. Apresentando em seguida a análise dos resultados obtidos, bem como discorrer dos problemas encontrados e recomendações de possíveis ajustes.

E por último, o capítulo 5 realiza as conclusões finais e sugestões para possíveis trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo trata dos conceitos teóricos descritos na literatura, publicações técnicas, descritivos de componentes e dispositivos que serão utilizados visando o desenvolvimento do projeto. São apresentadas também as técnicas, metodologias e componentes eletrônicos usados e o descritivo da solução proposta.

2.1 Eletromagnetismo

O eletromagnetismo é o nome do estudo das interações elétricas e magnéticas decorrentes das cargas elétricas. A carga elétrica é uma das propriedades das partículas que constituem a matéria. Muitas experiências demonstram a existência das cargas elétricas. Como por exemplo, ao pentear os cabelos com um pente de plástico num inverno seco, é possível atrair pequenos pedaços de papel com o pente. Essa atração é o resultado de forças elétricas existentes entre as cargas elétricas dos átomos do pente e dos pedacinhos de papel, que dá forma aos chamados campos eletromagnéticos. (Rego, 2010)

Os campos eletromagnéticos estão presentes em todos os lugares e estamos imersos neles. Podendo ser gerados tanto naturalmente como, por exemplo, radiação solar e raio atmosférico. Como sendo gerado por nós em estações de rádio, telefones celulares, e linhas de transmissão de energia. O escritório moderno, a cozinha e o automóvel todos estão plenos de eletricidade e campos magnéticos, atuando em qualquer lugar em que um motor elétrico esteja em operação. (Wentworth, 2006)

Figura 2.1 - Motor Elétrico



Fonte: <https://relampagomotores.com.br>

O motor elétrico é um aparelho que tem a função de transformar energia elétrica em energia mecânica, um exemplo pode ser visto na figura 2.1. O princípio básico de seu funcionamento começa conduzindo corrente elétrica por um fio. Ao conduzir corrente elétrica, o fio fica imerso em um campo eletromagnético e sofre a atuação do que é conhecido como força eletromagnética. O torque gerado pela força eletromagnética pode ser direcionado a realizar um movimento, geralmente giratório, ao moldar o formato do fio condutor, geralmente na forma de uma espira. (Rego, 2010)

Como averiguado neste tópico, são vastas as aplicações envolvendo motores elétricos. Porém, para este projeto, as aplicações se resumem a somente duas. A implementação de um relé de 5V e de um fecho de porta eletromagnético.

2.1.1 Relé

Relé é um dispositivo eletromecânico, formado por uma magneto móvel, projetado para responder a condições de entrada prescritas. Geralmente estas condições de entrada são elétricas, mas podem ser de outras grandezas físicas. Em que, após a ocorrência da entrada especificada, causa a movimentação do magneto, unindo dois contatos metálicos ou mudando o circuito elétrico. A figura 2.2 ilustra a aparência de um relé encontrado no mercado. (Silva, 2012)

O funcionamento de um relé é simples. Quando uma corrente elétrica é injetada em sua entrada, no caso das condições de entrada serem elétricas, ela circula pela bobina. A bobina por sua vez cria um campo magnético, que atrai um ou uma série de contatos. Fechando ou abrindo o circuito. Ao cessar a corrente da bobina o campo magnético também cessa, fazendo com que os contatos voltem para a posição original. Embora esse seja um processo lento, um pouco mais de um milissegundo para fechar um circuito, os relés são amplamente utilizados em aplicações que não desempenham uma atividade crítica ou que necessitem de uma resposta em tempo real. (Morimoto, 2005)

Figura 2.2 - Relé



Fonte: <http://www.cinestec.com.br>

2.1.2 Fecho eletromagnético

O fecho eletromagnético é uma das aplicações do eletromagnetismo desenvolvida para o controle de acesso de uma porta. O fecho não tem a função de abrir a porta, mas sim de destravá-la, liberando a porta para ser aberta. O modelo de fecho utilizado neste trabalho é o FEC-91 da HDL, apresentada na figura 2.3.

Figura 2.3 - Fecho Eletromagnético



Fonte: <https://www.hdl.com.br>

O funcionamento do fecho envolve um fio condutor enrolado sobre si mesmo, também denominado de bobina. Ao conduzir corrente elétrica, a bobina cria um campo eletromagnético, que movimenta a trava metálica presente no fecho de maneira a destravar a porta, e permitindo que seja aberta. Quando a bobina para de conduzir corrente, o campo eletromagnético cessa, a trava metálica volta ao seu lugar de origem e o acesso é bloqueado. No caso do fecho FEC-91 da HDL, ele possui o que é conhecido como memória mecânica. Uma lâmina metálica que segura a trava do fecho, uma vez destravado, na posição de aberto até que alguém passe pela porta.

2.2 Microcontroladores

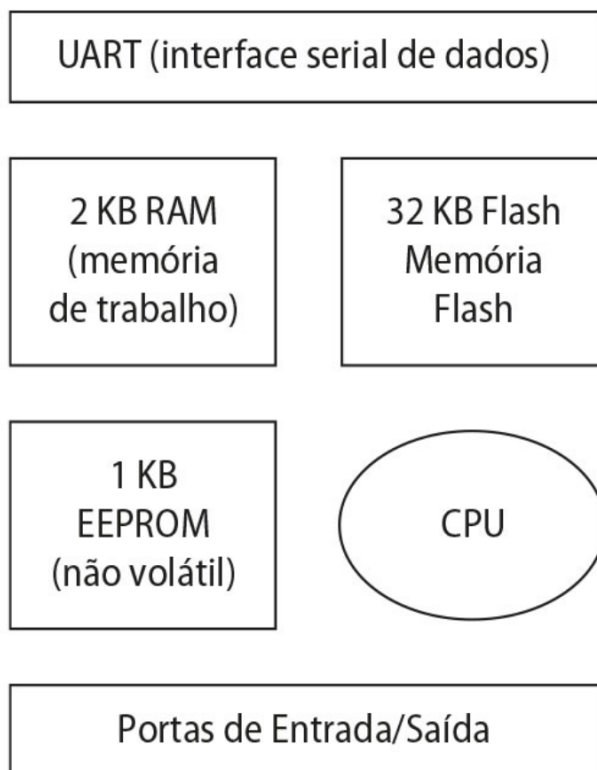
Um microcontrolador é a integração de todas as partes básicas que formam um microcomputador (microprocessador, memórias e portas de E/S) em um único circuito integrado. Normalmente, é limitado em termos de quantidade de memória, principalmente no que diz respeito à memória de dados. (Dantas, 2015)

Enquanto um microprocessador é projetado para ser utilizado em computadores de uso geral, capazes de executar programas complexos e sistemas operacionais, um microcontrolador é projetado para realizar tarefas específicas e executar programas mais simples (geralmente sem a necessidade de um sistema operacional). Apresentam um custo bastante reduzido que vem caindo bastante nos últimos anos, pois são produzidos em larga escala. E também são mais resistentes, podendo ser aplicado em condições mais extremas onde computadores normais não conseguiriam. (Pereira, 2010)

Em geral, são usados quando não é necessário manipular e armazenar altas quantidades de dados e para realizar simples tarefas de controle. Como exemplo, podem ser citados os microcontroladores usados em automação residencial. Tais como, fornos de micro-ondas, máquinas de lavar louça e de lavar roupa, controles de refrigeração de refrigeradores, ar-condicionados, telefones fixos, alarmes residenciais, equipamentos eletrônicos de controle de portões automáticos. (Dantas, 2015)

Sua estrutura básica é composta por três blocos básicos: unidade central de processamento, unidade de memória e unidade de entrada e saída de dados (E/S ou I/O), conforme apresentado na Figura 2.4.

Figura 2.4 - Esquemático de um microcontrolador ATmega328



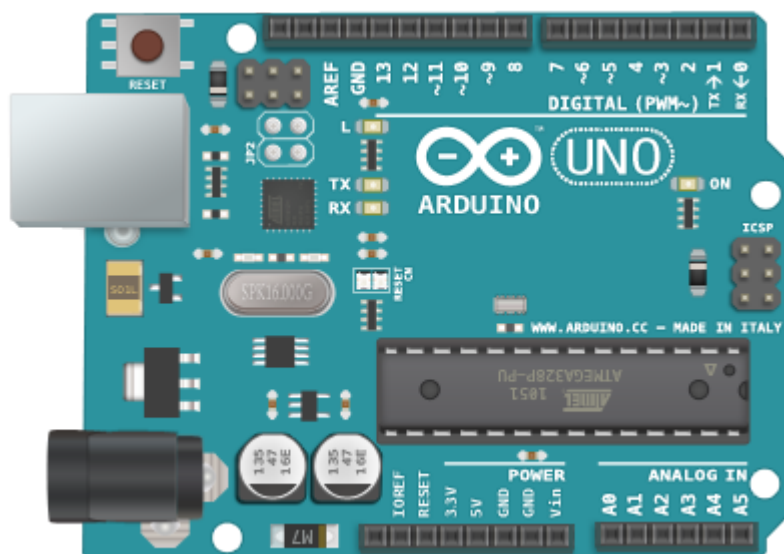
Fonte: (Monk, 2017)

2.2.1 Arduino

O Arduino é uma pequena placa de microcontrolador que contém uma conexão USB, tornando possível a ligação com um computador. Além disso, contém diversos terminais que permitem a conexão com dispositivos externos, como motores, relés, sensores luminosos, diodos a laser, alto-falantes, microfones e outros. Eles podem ser energizados pelo computador através do cabo USB, por uma bateria de 9 V ou por alguma outra fonte de alimentação. Um Arduino pode ser controlado diretamente pelo computador ou, então, pode trabalhar de forma autônoma. Neste caso, ele é primeiro programado pelo computador através da conexão USB e, em seguida, desconectado desse computador. (Monk, 2013)

O microcontrolador embutido no arduino é um dispositivo retangular preto de 28 pinos. Ele se encaixa em um soquete de modo a ser facilmente substituído, como pode ser visto na figura 2.5. No arduino UNO especificamente, o chip de 28 pinos é denominado ATmega328.

Figura 2.5 - Arduino UNO



Fonte: <https://www.arduino.cc>

2.3 Processamento digital de sinais

Sistemas em engenharia processam informações que são captadas no mundo real com o objetivo de se obter algum resultado útil. É possível associar as informações captadas relativa a qualquer instante de tempo. A variação dessas informações conforme a passagem do tempo recebe o nome de sinal, e processamento de sinais é a disciplina que estuda como os sinais se relacionam e, principalmente, como manipular os sinais de forma a se obter um resultado desejado. (Nalon, 2009)

No processamento analógico de sinais, tomamos um sinal variando continuamente. Este tipo de sinal é denominado sinal analógico e representa, da forma mais fiel possível, uma quantidade física captada do mundo real. Após obter o sinal analógico, o passamos por um sistema que o modifica com um certo propósito. Essa modificação também é, em geral, continuamente variável por

natureza, e caso não seja, estamos lidando com processamento digital de sinais. (Diniz, 2014)

O processamento digital de sinais é a disciplina que estuda as regras que governam os sinais que variam em tempo discretos, assim como os sistemas usados para processá-los. Os sinais no tempo discreto, ou sinais digitais, são sequências de números usadas no processamento realizado em algum tipo de hardware digital. O poder do processamento digital de sinais decorre do fato de que, uma vez que uma sequência de números esteja disponível para o hardware digital apropriado, podemos efetuar qualquer forma de processamento numérico sobre eles. Claramente, isso seria muito difícil de implementar usando um hardware analógico. (Diniz, 2014)

Se queremos processar um sinal analógico usando esse tipo de recurso, precisamos ter um modo de converter um sinal no tempo contínuo num sinal no tempo discreto, de tal forma que o sinal no tempo contínuo possa ser recuperado a partir do sinal no tempo discreto. (Nalon, 2009) Sendo exatamente essa conversão, uma das etapas realizadas pelo sistema desenvolvido neste trabalho. Em que convertemos um sinal sonoro no tempo contínuo, obtido por um sensor analógico de som, em sua representação no tempo discreto para possibilitar seu processamento através de um microcontrolador. Especificamente, o microcontrolador de placa arduino apresentado no tópico 2.2.1.

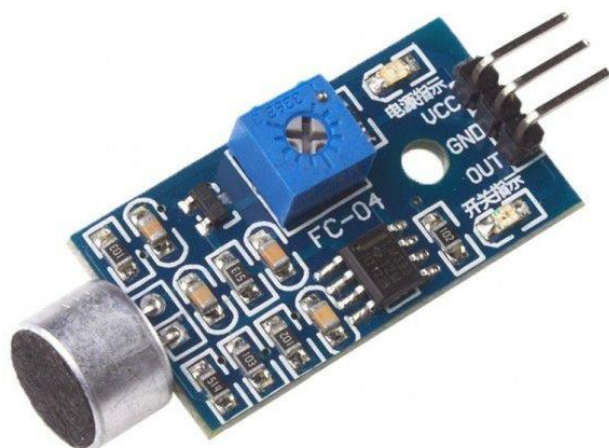
2.3.1 Módulo Sensor Analógico de Som

O Sensor Analógico de Som é um dispositivo de microfone que capta os sinais sonoros da maneira que são encontrados na natureza. Ou seja, como tratado no tópico 2.3 sobre processamento digital de sinais, os sinais se encontram em seu domínio de tempo contínuo. Porém, o sinal no domínio de tempo contínuo precisa ser transformado em um sinal digital, através do processamento digital de sinais, para ser processado por um hardware. Como por exemplo, o microcontrolador de placa arduino apresentado no tópico 2.2.1.

O Módulo Sensor Analógico de Som é um módulo que integra com a placa arduino, um exemplo se encontra na figura 2.6 logo a seguir. A integração permite que os dois, a placa e o módulo, trabalhem bem juntos. O módulo tem a

funcionalidade tanto de captação dos sinais sonoros quanto a sua conversão para um sinal digital equivalente. Desempenhando as duas tarefas ao mesmo tempo ele já envia para o arduino um sinal no domínio de tempo discreto pronto para ser processado pelo microcontrolador.

Figura 2.6 - Módulo Sensor Analógico de Som



Fonte: <https://arduxop.com>

2.4 Bluetooth

Bluetooth é uma tecnologia de rede pessoal sem fio que surgiu a partir da necessidade de conectar telefones móveis a PDAs, notebooks e outros equipamentos pessoais sem fios. Um grupo fabricantes de telefones móveis e computadores, liderados por L.M. Ericsson, desenvolveu uma especificação para uma rede pessoal sem fio (WPAN, Wireless Personal Area Network) para a transmissão de fluxos de voz digitalizada como dados. (Coulouris, 2013)

As redes Bluetooth diferem das redes WiFi. Seu contexto de utilização apresenta requisitos como metas de custo reduzido e baixo consumo de energia. Cumprindo o objetivo de fornecer comunicação em tempo real e com qualidade de serviço, através de um hardware compacto e largura de banda limitada. O padrão Bluetooth foi projetado para permitir comunicação entre dispositivos pequenos e de

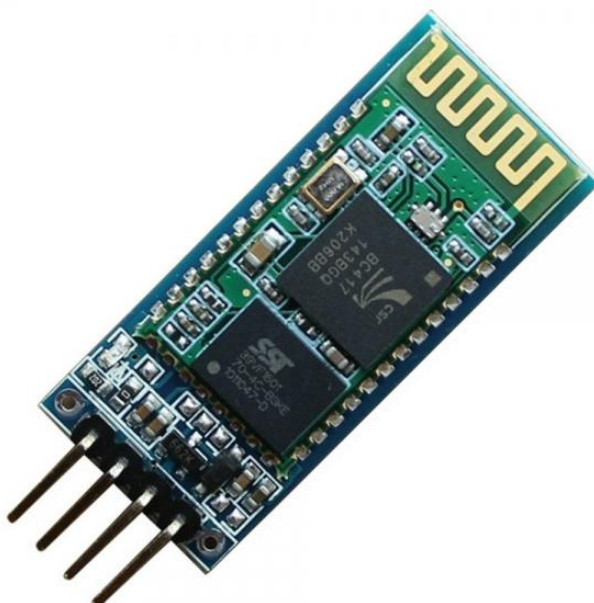
baixo custo, como os fones de ouvido sem fio adaptados à orelha, que recebem fluxos de áudio de um telefone celular, ou interconexões entre computadores, telefones, PDAs e outros dispositivos móveis. (Coulouris, 2013)

2.4.1 Módulo bluetooth serial HC-06

Uma forma de comunicação sólida é um requisito muito importante para o sucesso de um projeto. Através dela que se estabelece a interface que permite a comunicação entre o usuário e o sistema, ou até mesmo entre dois sistemas. No tópico 2.4, foi abordado a rede pessoal bluetooth, utilizada para permitir uma comunicação consistente e de curto alcance entre dispositivos pequenos.

No caso da placa de microcontrolador Arduino UNO, apresentada na figura 2.5, a conexão bluetooth não é disponibilizada nativamente. Para isso, é utilizado um módulo externo capazes de estabelecer a conexão e a troca de mensagens entre a placa Arduino e um outro dispositivo com interface Bluetooth. A uma variedade de módulos com essa funcionalidade disponíveis no mercado. Mas, para o desenvolvimento deste projeto, será utilizado o modelo HC-06, apresentado na figura 2.7.

Figura 2.7 - Módulo bluetooth serial HC-06



Fonte: <http://www.baudaeletronica.com.br/>

O módulo HC-06 é capaz de abstrair toda a parte da comunicação sem fio necessária para a comunicação bluetooth, fornecendo uma interface serial para que o microcontrolador possa enviar ou receber dados. Dispondo de apenas 4 pinos, sendo dois deles para alimentação do sistema, e os outros dois para transmissão e recepção de dados via serial.

2.5 Linguagem de programação

O funcionamento de um computador depende dele ser programado para desempenhar a tarefa que lhe é designada. O processo de programação é uma comunicação controlada entre um ser humano (tecnicamente preparado) e o computador propriamente dito. O processo de comunicação se faz com o uso de uma linguagem de programação, que é responsável por expressa informações de modo compreensível tanto para o computador quanto para os humanos. (Manzano, 2011)

As linguagens de programação possuem um domínio de expressão mais reduzido. Isso quer dizer que elas possibilitam apenas a comunicação de idéias computacionais. (Tucker, 2014) Os programas de computador, conhecidos como softwares, são formados por uma série destas expressões computacionais que são executadas no sentido de realizar as operações desejadas. Essa série de expressões é conhecida como programação. (Manzano, 2011)

A dois grupos de linguagens de programação utilizados para comunicar-se com um computador, as de baixo nível ou de alto nível. Tanto as linguagens de programação em baixo nível como em alto nível, quando em uso em um computador eletrônico, são traduzidas internamente para uma forma numérica convencional grafada em formato binário, representada pelos símbolos numéricos 1 e 0. O valor 1 indica que algum componente interno do computador encontra-se ligado, e se estiver com valor 0, indica que o componente encontra-se desligado. Não importa o tamanho ou tipo de computador em uso, pois todo computador só é operado internamente com códigos binários. Também denominados de linguagem de máquina.

2.5.1 Linguagem de programação C

As linguagens de baixo nível surgiram praticamente junto com a era dos computadores digitais e possuem diversos aspectos negativos, como a complexidade, falta de compatibilidade entre diferentes sistemas e pouca legibilidade. Então surgiu a necessidade de uma abordagem mais concisa e simplificada para comandar computadores. Essa abordagem simplificada, mais próxima da forma humana de pensamento, é chamada de linguagem de alto nível. (Pereira, 2007)

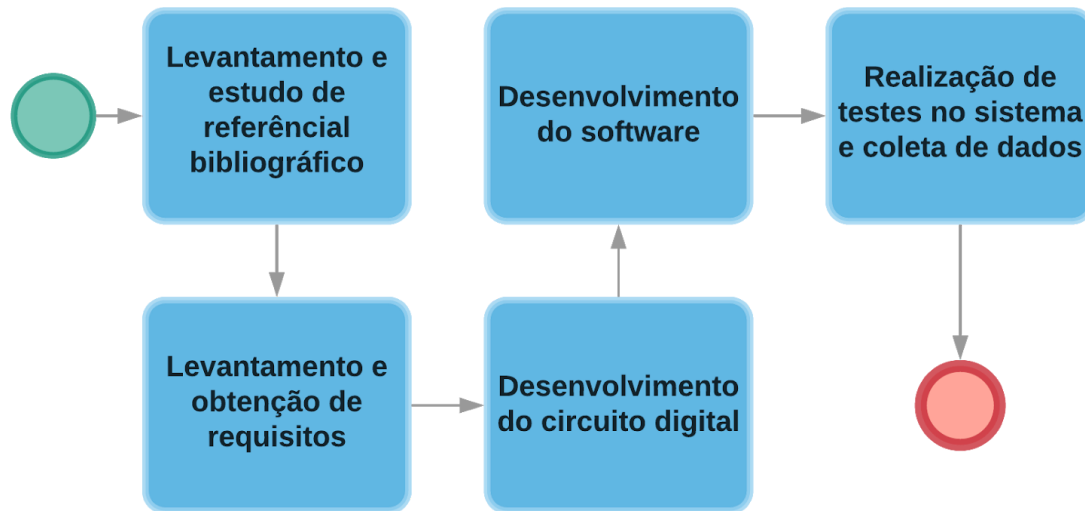
A linguagem C foi criada em 1972, por Dennis Ritchie, da Bell Laboratories, com a necessidade de reescrita do sistema operacional UNIX. Até o desenvolvimento do C, não existiam linguagens de programação adequadas à tarefa de criação de sistemas operacionais e outros softwares de baixo nível, restando aos desenvolvedores utilizar o Assembly. Sendo uma linguagem de programação genérica desenvolvida para ser tão eficiente e rápida, quanto estruturada e lógica. Consiste, na realidade, em uma linguagem de nível intermediário entre o Assembly e as linguagens de alto nível. (Pereira, 2007)

A utilização de C para a programação de microcontroladores parece uma escolha natural e realmente é. Atualmente, a maioria dos microcontroladores disponíveis no mercado contam com compiladores de linguagem C para o desenvolvimento de software. O uso de C permite a construção de programas e aplicações muito mais complexas do que seria viável utilizando apenas o Assembly. (Pereira, 2007)

2.6 Metodologia

Neste tópico abordamos a metodologia estabelecida para a criação da solução proposta, que é de suma importância para um correto desenvolvimento. A metodologia explica de forma minuciosa e detalhada todas as etapas e ações a serem desenvolvidas de acordo com o método a ser utilizado na pesquisa. Sendo assim, todas as etapas são apresentadas no fluxograma da figura 2.8 e detalhadas abaixo da figura.

Figura 2.8 - Fluxograma da metodologia utilizada



Fonte: Próprio autor

1. Na primeira etapa é realizado um levantamento bibliográfico de todo material teórico necessário para a elaboração do trabalho. O estudo do funcionamento teórico fornece uma base sólida para o desenvolvimento prático. Em vista da solução proposta, tornasse essencial os seguintes conhecimentos: a implementação da automatização em ambientes residenciais, a utilização de microcontroladores com foco nas placas arduino, o processamento digital de sinais com ênfase no processamento de sinais sonoros e a aplicação da tecnologia bluetooth.
2. A segunda etapa é constituída do levantamento de requisitos indispensáveis para a composição do trabalho, que englobam tanto necessidades de hardware quanto de software. Ainda nesta etapa, é realizada a busca e obtenção dos requisitos verificados no levantamento.
3. Na terceira etapa é elaborado um esquemático do circuito digital, que será acoplado em uma maquete de porta, responsável por receber e processar sinais sonoros vindos do ambiente. Ao mesmo tempo que possibilita a troca de informações com um dispositivo smartphone via conexão bluetooth. Em seguida tomar a devida ação de acionar, ou não, um relé que libera o acesso

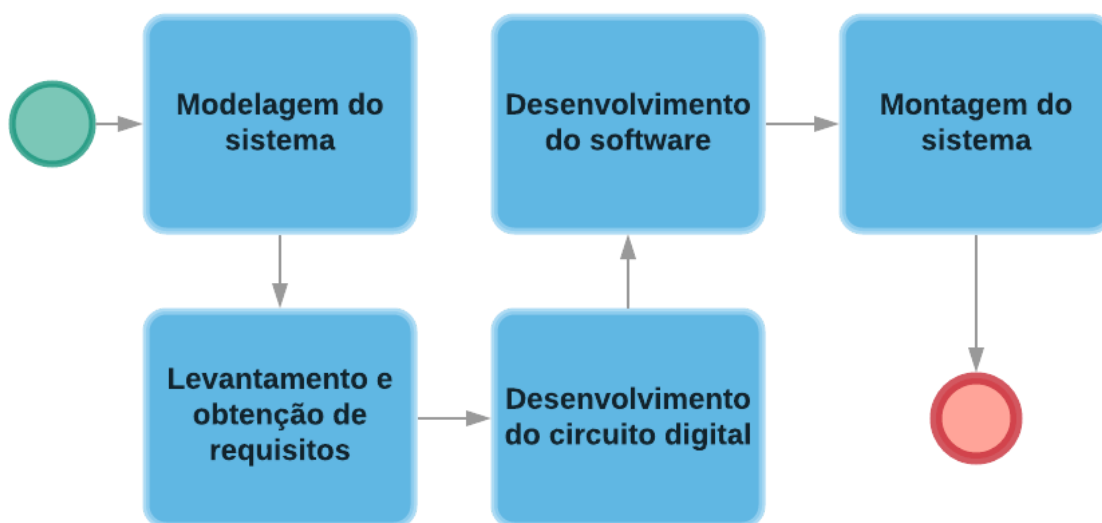
a entrada. Após a elaboração do esquemático, ainda nesta etapa, será criado o circuito digital projetado no esquemático.

4. A quarta etapa se trata da criação do software, em linguagem C, que será inserido no microcontrolador arduino. Programando o microcontrolador para desempenhar as tarefas ao qual o circuito digital elaborado é responsável.
5. Na quinta etapa, testes serão realizados no sistema como um todo em dois ambientes distintos. Sendo o primeiro, uma sala controlada para melhor isolamento das variáveis, e depois em um sala de uso casual. Os resultados serão tabulados. Fazendo relação entre uma ação exercida sobre o sistema e se ele teve a resposta esperada, ou obteve uma falha. Após a coleta dos resultados, será feito um levantamento estatístico da taxa de erro.

3. DESENVOLVIMENTO DO SISTEMA

Neste capítulo serão apresentadas as etapas de desenvolvimento do sistema proposto. Assim como a descrição detalhada das atividades realizadas em cada uma das etapas. Abordando desde a modelagem do sistema, até a montagem do mesmo. O fluxograma mostrado abaixo esboça de forma ampla as etapas que a serem detalhadas ao longo deste capítulo.

Figura 3.1 - Fluxograma das etapas de desenvolvimento



Fonte: Próprio autor

O fluxograma inicia com a modelagem do sistema para a obtenção de um esboço definitivo. Ou seja, uma visão geral de como o sistema deve ser para ser usada como referência no decorrer de todas as outras etapas do processo. A próxima etapa passa pelo levantamento e obtenção dos requisitos necessários para a construção do que foi modelado na etapa anterior. Depois, em posse de todos os requisitos do projeto, a próxima etapa compreende a elaboração e construção do circuito digital. Já a última etapa compreende o desenvolvimento do software que controlará o acesso a entrada e estabelecerá uma interface para a conexão bluetooth.

3.1 Modelagem do sistema

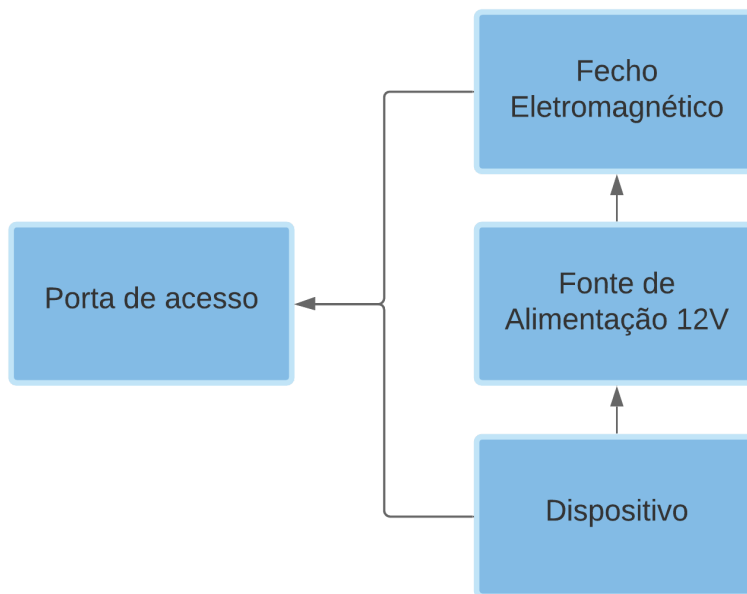
A modelagem de sistema é a prática de construção de modelos abstratos de um dado sistema. Modelos são formas de representação da realidade, ou de objetos reais, sendo que cada modelo demonstra uma perspectiva diferente do sistema. Os modelos são usados como base durante a etapa de levantamento de requisitos para ajudar a extrair os requisitos do sistema ainda em sua fase de planejamento. (Souza, 2013)

A modelagem serve também para descrever e validar o projeto para aqueles que vão executá-lo, testando suas reações sob diversas condições para certificar que seu funcionamento atenderá aos requisitos estabelecidos. E após isso, é usada para documentar a estrutura e a operação do sistema. (Souza, 2013) Por isso, a criação de um modelo é a primeira etapa no desenvolvimento do projeto para facilitar as outras etapas subsequentes. O modelo confeccionado neste trabalho utiliza um diagrama de implantação e um fluxograma de processos de negócio.

3.1.1 Diagrama de implantação

O diagrama de implantação faz parte do conjunto de diagramas contemplados pela linguagem de modelagem UML. A principal função do diagrama de implantação é modelar o relacionamento entre recursos de estrutura. Descrevendo os elementos estruturais que compõem o sistema de maneira a representar suas partes e suas ligações. A figura 3.2 apresenta o diagrama de implantação do sistema proposto neste trabalho.

Figura 3.2 - Diagrama de implantação do sistema



Fonte: Próprio autor

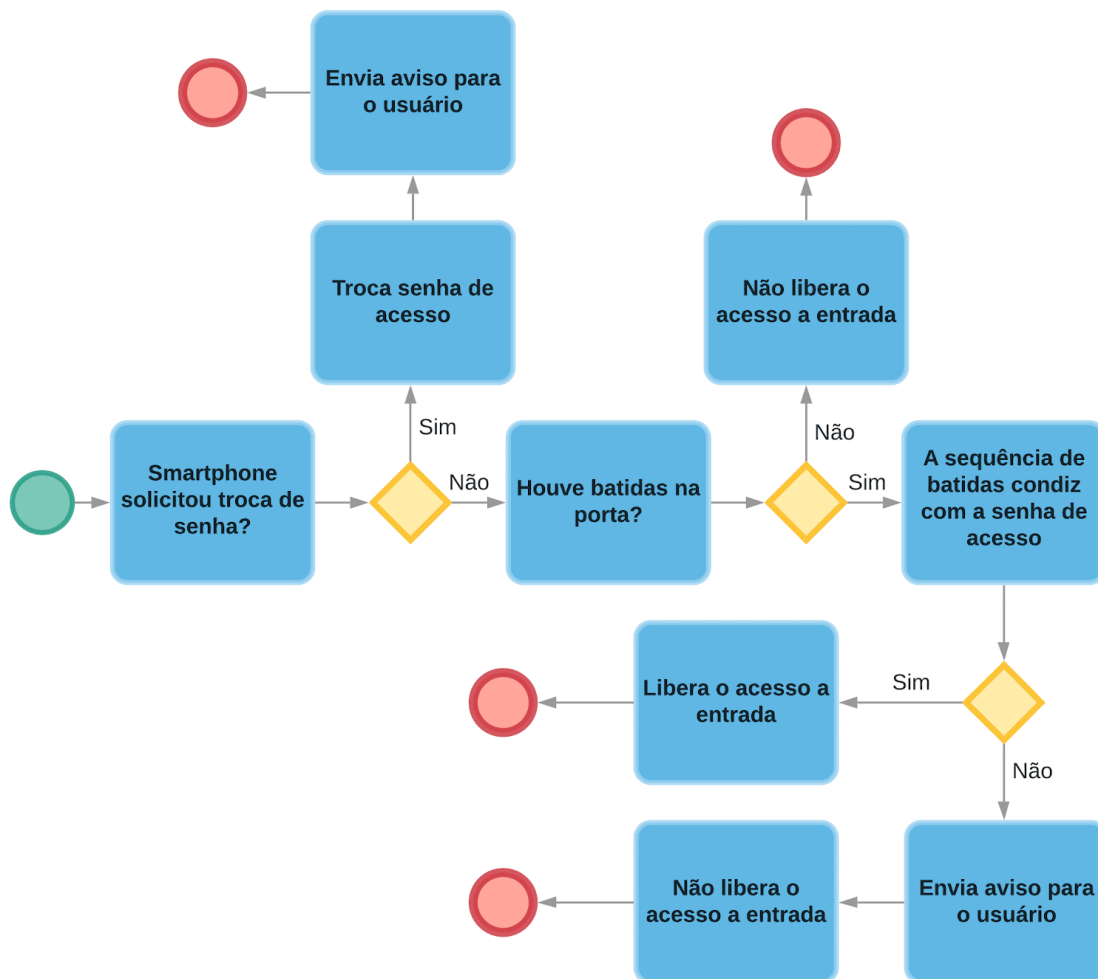
O diagrama acima descreve, de forma simplificada, que a porta de acesso deve conter o dispositivo desenvolvido no tópico 3.3 e um fecho eletromagnético instalados nela. Sendo assim, a porta deve ser pequena para facilitar os testes, mas ter tamanho suficiente para acomodar estes dois elementos visando a praticidade. O diagrama também estabelece uma ligação entre o dispositivo e o fecho eletromagnético através de uma fonte de alimentação de 12V. A fonte de alimentação tem a função de fornecer corrente elétrica ao fecho eletromagnético, sendo este fornecimento de corrente elétrica controlado pelo dispositivo.

3.1.2 Fluxograma de processos de negócios

O fluxograma de processos de negócios é um modelo que descreve o fluxo dos processos no sistema. Ou seja, as atividades e procedimentos que o sistema realiza em um dado cenário buscando atingir um resultado. (Souza, 2013) O fluxograma cria uma compreensão geral do funcionamento do sistema. E também serve como base para a produção do código fonte, tendo em vista cada tarefa que o software está responsável por realizar. A figura 3.3 abaixo apresenta o fluxograma

de processos de negócios do sistema na notação BPMN 2.0, uma popular notação para desenvolver modelos de processos de negócio.

Figura 3.3 - Fluxograma de processos de negócios do sistema



Fonte: Próprio autor

O fluxograma se inicia verificando a ocorrência da interação do dispositivo com um smartphone na tentativa de trocar a senha de acesso. Durante a tentativa de trocar a senha, o dispositivo fica inabilitado de permitir o acesso a entrada e, assim que concluída a troca, envia uma mensagem confirmando a troca de senha para o usuário. Caso não esteja ocorrendo nenhuma tentativa de trocar a senha, o dispositivo segue verificando a ocorrência de batidas na porta.

O dispositivo encerra seu fluxo de atividade, sem liberar o acesso a entrada, na ausência de batidas na porta. Entende-se por ausência de batidas na porta, a falta de contato do usuário com a porta ou a realização de contato que não gere barulho. Caso um usuário realize batidas na porta, o dispositivo verifica se a sequência de batidas condiz com o padrão estabelecido na senha de acesso.

Após a verificação da sequência de batidas, a duas possibilidades que podem ocorrer. Primeiro, se porventura a sequência de batidas condizer com o padrão da senha de acesso, o dispositivo libera o acesso a entrada. No caso da sequência de batidas não condizer com a senha de acesso, o dispositivo envia um alerta ao smartphone conectado, avisando da tentativa falha, e não libera o acesso a entrada.

3.2 Levantamento e obtenção de requisitos

Baseado nos modelos de sistema apresentados no tópico 3.1, e nos objetivos a serem alcançados do tópico 1.2, foi feito o levantamento dos requisitos necessários para o desenvolvimento do projeto. No quadro 3.1, são mostrados os itens associados e levantamento dos requisitos, juntamente com seu custo unitário, quantidade obtida e necessidade ao qual ele preenche.

Quadro 3.1 - Levantamento de requisitos

Itens	Custo Unitário	Quantia	Necessidade
Arduino	R\$56,00	1	Fornecer um microcontrolador de fácil uso
Fecho Eletromagnético	R\$79,90	1	Servir como atuador no controle de acesso a porta
Sensor Analógico de Som	R\$49,90	1	Captar quando uma batida for realizada na porta
Maquete de Porta	R\$79,90	1	Simular uma porta real em tamanho reduzido
LED Difuso (5mm)	R\$0,25	2	Enviar mensagens simples ao usuário por meio de sinais luminosos

Resistor de Filme de Carbono 1K Ω - 1/4W	R\$0,15	2	Limitar a quantidade de corrente elétrica fornecida aos LEDs.
Resistor de Filme de Carbono 2K2 Ω - 1/4W	R\$0,15	2	Servir como resistor de base do transistor e criar um divisor de tensão para o Módulo Bluetooth Serial HC-06 (explicado no tópico 3.3)
Resistor de Filme de Carbono 1K2 Ω - 1/4W	R\$0,15	1	Criar um divisor de tensão para o Módulo Bluetooth Serial HC-06 (explicado no tópico 3.3)
Transistor BC547	R\$0,20	1	Utilizar uma pequena fonte de energia para controlar uma fonte maior
Placa Universal Perfurada	R\$5,00	1	Servir como base para a construção do circuito digital
Diodo 1N4148	R\$0,25	1	Evitar a tensão reversa do relé (explicada no tópico 3.3)
Relé 5V	R\$6,00	1	Controlar o fornecimento de energia ao fecho eletromagnético
Fonte de Alimentação - 12V/1A	R\$12,00	1	Fornecer energia para o fecho eletromagnético
Módulo Bluetooth Serial HC-06	R\$39,90	1	Estabelecer uma conexão bluetooth
Kit de jumpers unifilares macho-macho	R\$5,00	1	Conectar o arduino com os demais componentes eletrônicos
Kit de jumpers unifilares macho-femea	R\$5,00	1	Conectar o arduino com os demais componentes eletrônicos

Fonte: Próprio autor

O orçamento do projeto foi estipulado em 340,30 reais a partir das informações fornecidas pelo quadro.

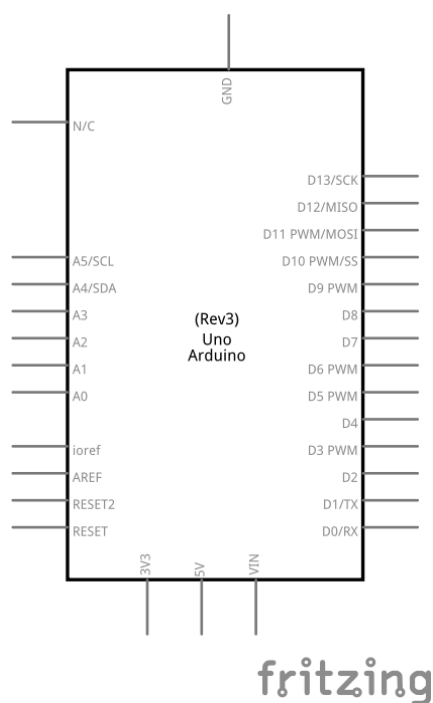
3.3 Desenvolvimento do circuito

O circuito foi primeiramente desenhado na forma de esquemático, através de um software, antes de ser construído em escala real. O software utilizado para desenhar o circuito foi o FRITZING, um software online simples, mas que permite esboçar as ligações entre os componentes. Desta forma, estabelecendo um modelo prévio e minimizando o risco de falhas ao construir o circuito digital.

Ao término do capítulo, será apresentada uma foto do circuito produzido em escala real. Porém, nas fases de construção, serão esboçados esquemáticos produzidos pelo software FRITZING com o intuito de facilitar a compreensão através de imagens mais sucintas.

O componente principal do circuito é o microcontrolador arduino. Sendo assim, o primeiro componente esboçado conforme a figura 3.4.

Figura 3.4 - Esquemático placa arduino

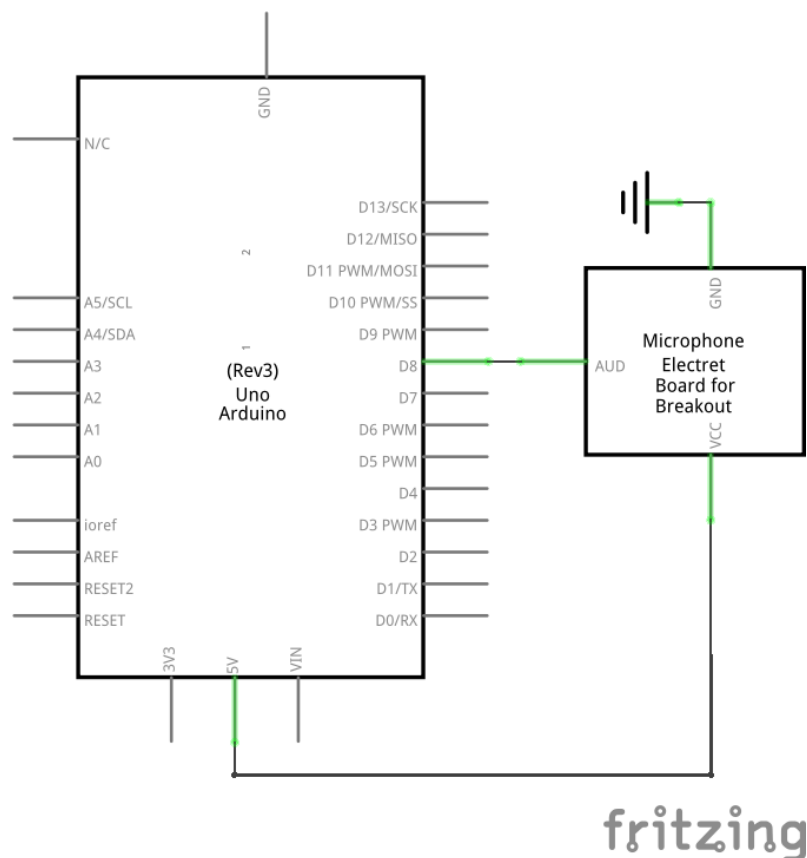


Fonte: Próprio autor

Após estabelecer a placa arduino UNO como componente central, o circuito está pronto para realizar sua primeira integração da placa com outro componente. A

figura 3.5 ilustra as ligações entre a placa arduino e o módulo sensor analógico de som.

Figura 3.5 - Esquemático arduino + módulo sensor analógico de som



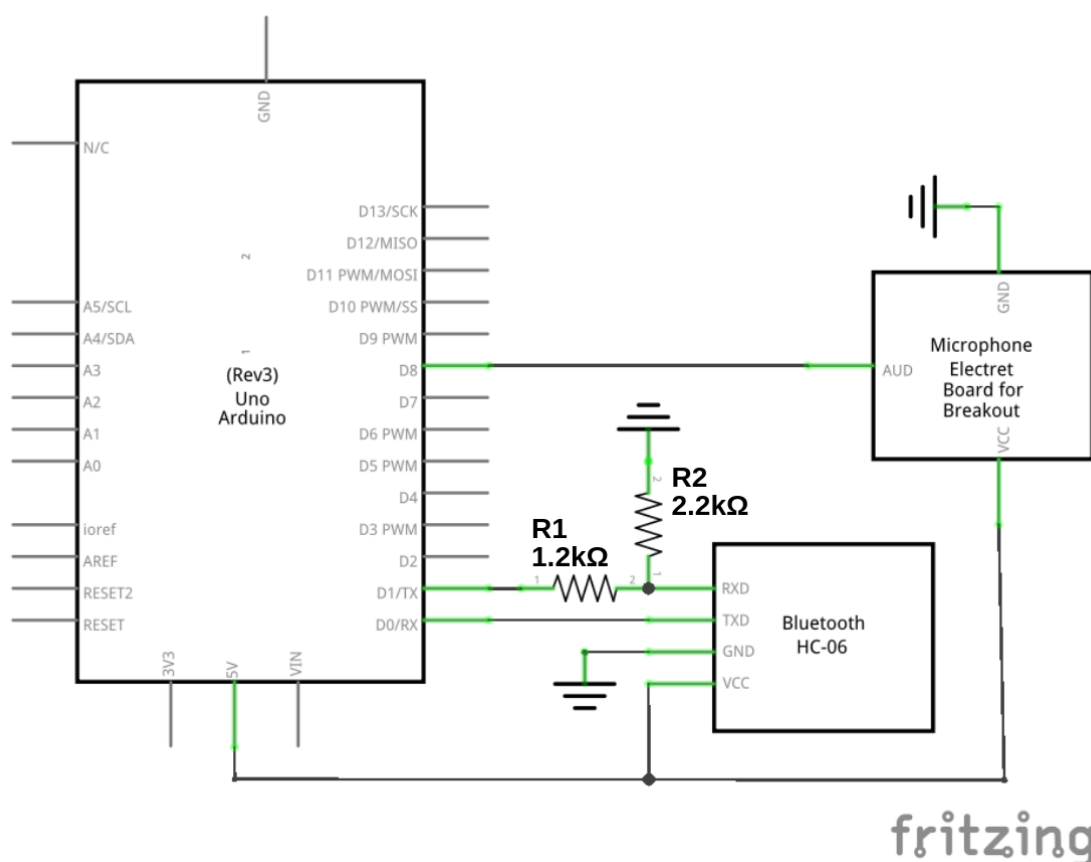
Fonte: Próprio autor

O módulo sensor analógico de som, conforme a figura 3.5, possui 3 pinos, sendo estes: VCC, GND e AUD. O módulo é alimentado através de seu pino VCC por uma fonte de 5V vinda da placa arduino. E também conectado ao terra por seu pino GND. Já o pino AUD fornece a saída digital do módulo baseada no sinal analógico captado pelo sensor analógico de som. O sinal digital é por fim enviado até a porta D8, que por sua vez está configurada como porta de entrada para o recebimento do sinal.

A figura 3.6 esboça a integração entre a placa arduino e módulo bluetooth HC-06. Conforme a figura, o módulo bluetooth possui 4 pinos. Sendo destes, os pinos VCC e GND de mesma função que o módulo sensor analógico de som,

conectados respectivamente na porta 5V de alimentação do arduino e ao terra. Já os pinos RXD e TXD são responsáveis pela recepção e transmissão de dados via serial.

Figura 3.6 - Esquemático arduino + módulo sensor analógico de som + módulo bluetooth HC-06



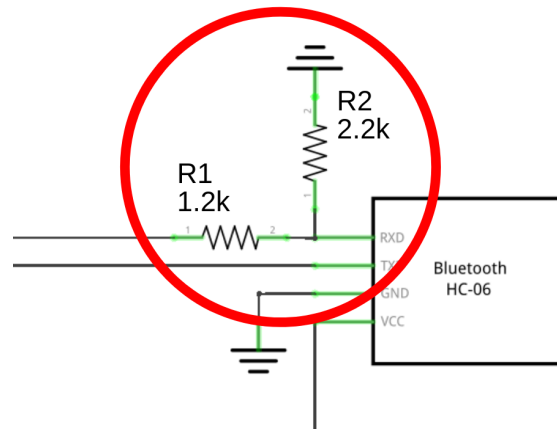
Fonte: Próprio autor

O pino TXD transmite dados seriais, recebidos de um smartphone através do módulo bluetooth, até a porta de entrada D0/RX presente no arduino. O pino RXD recebe dados seriais da porta de saída D1/TX do arduino, em seguida transmitindo os mesmos dados para qualquer dispositivo smartphone conectado ao módulo bluetooth.

Um fator crítico na confecção desta parte do circuito é que o módulo HC-06 utiliza o nível lógico de 3,3 volts. Então, já que o arduino UNO possui o nível lógico de 5 volts, é necessário adicionar ao pino RXD, que recebe informações direto do

arduino, um divisor de tensão para evitar que o componente seja danificado. Na figura 3.7, pode se visualizar a implementação do divisor de tensão com mais destaque.

Figura 3.7 - Esquemático circuito divisor de tensão



Fonte: Próprio autor

O circuito usa um resistor R1 de 1,2 K ohms e outro R2 de 2,2 K ohms, o que gerou um nível de sinal na resistência R2 de aproximadamente 3,235 volts. Foi considerada também a utilização de divisores de tensão profissionais, porém o divisor de tensão normal já fornece o nível de sinal suficiente para o funcionamento seguro do módulo. O cálculo utilizado para chegar ao resultado de 3,235 volts se encontra na figura 3.8. Em que R1 e R2 são os valores dos resistores, V o valor da tensão de entrada e V2 o valor da tensão de saída.

Figura 3.8 - Cálculo do divisor de tensão

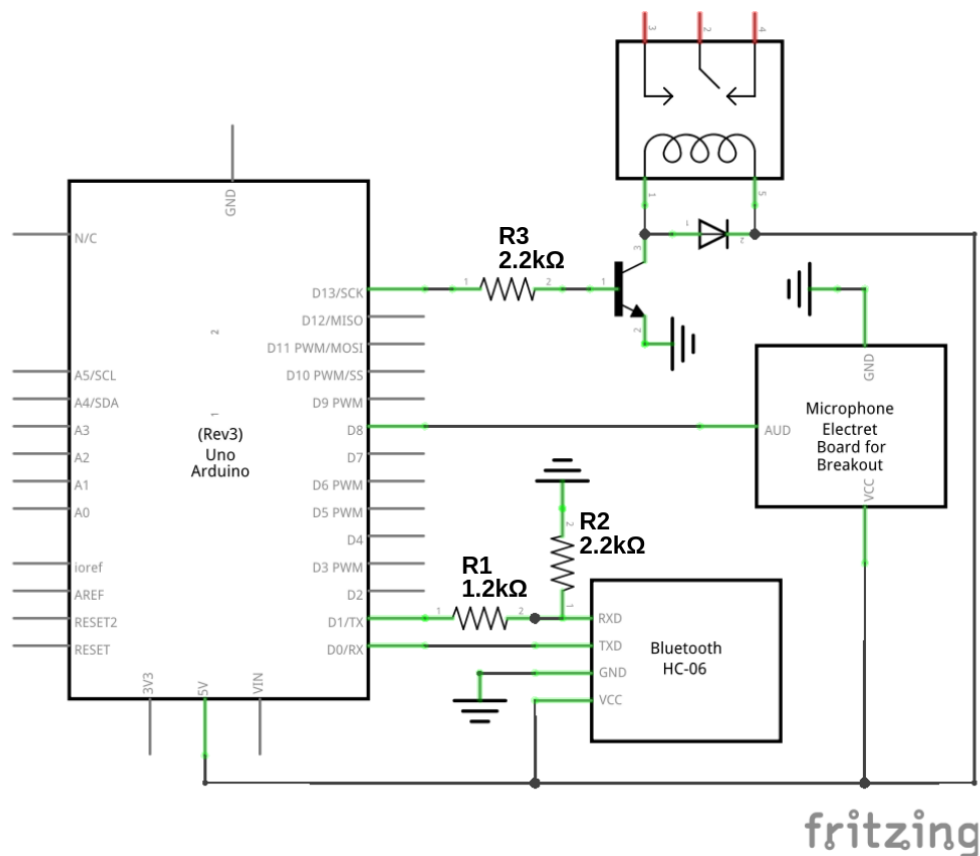
$$V_2 = V \cdot \frac{R_2}{R_2 + R_1}$$

$$V_2 = 5 \cdot \frac{2,2}{2,2 + 1,2} = 3,235 \text{ Volts}$$

Fonte: (DUARTE, 2017)

Em seguida, foi instalado um relé 5V junto a placa arduino como apresentado na figura 3.9. Para fazer a alimentação do relé 5V com o Arduino, é necessário o uso de dois outros componentes: um transistor BC547 e um diodo 1N4148. O transistor é utilizado para enviar um sinal de 5 volts, comandado pela porta de saída D13 do arduino, com o intuito de acionar o relé. O transistor também é responsável por garantir que o nível do sinal permaneça em zero na ausência de atividade na porta D13 do arduino. Já o diodo é responsável por evitar que a tensão reversa do relé alcance o arduino e danifique o componente. Quando o relé desliga, o indutor dentro dele gera uma corrente invertida. A tensão reversa é quando ocorre da corrente elétrica do circuito ser aplicada no sentido inverso do que foi projetado, e nem todo componente do circuito está apto a lidar com essa situação.

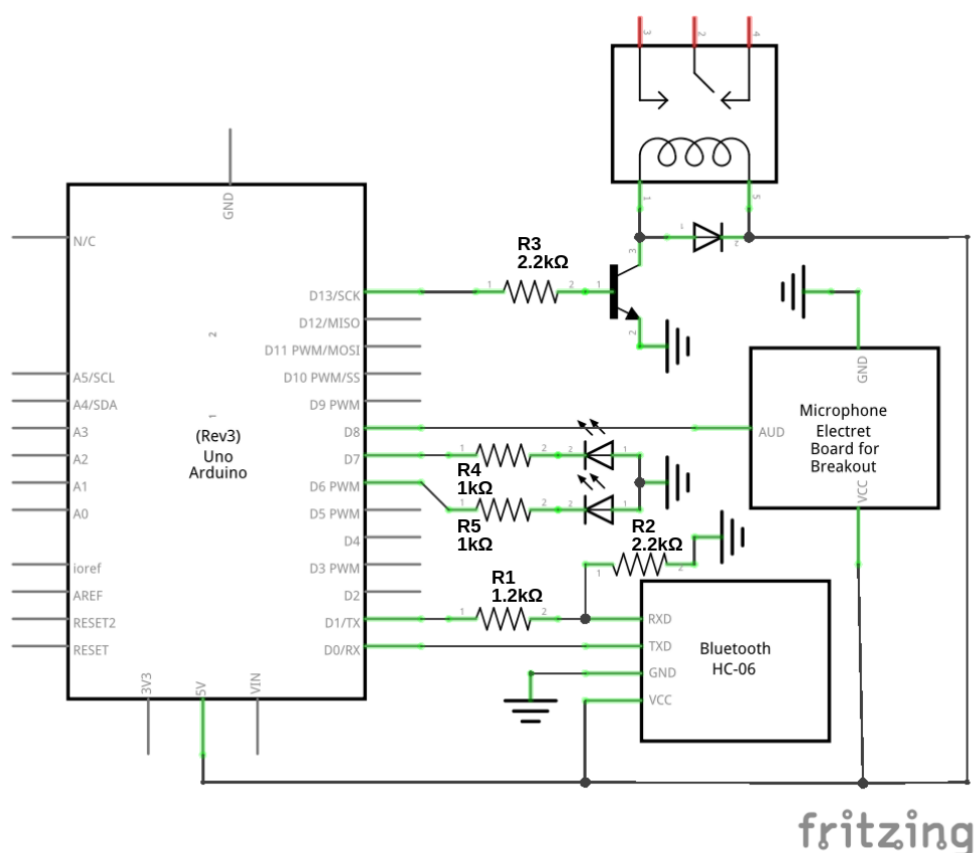
Figura 3.9 - Esquemático arduino + módulo sensor analógico de som + módulo bluetooth HC-06 + relé



Fonte: Próprio autor

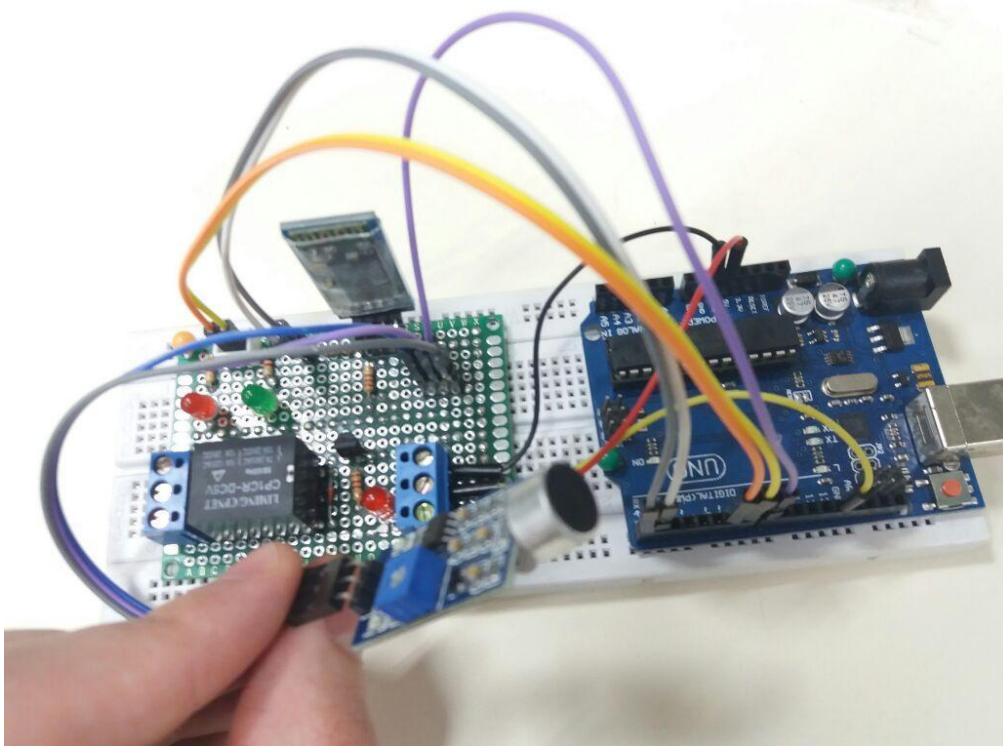
Até o presente momento o circuito já se encontra apto para desempenhar sua tarefa. Porém, serão acrescentados dois leds que serviram de resposta informativa para que o usuário acompanhe o funcionamento do sistema. Os leds irão acender quando os sinais de 5 volts, com intuito informativo, são enviados através das portas de saída D6 e D7 do arduino. A versão final do esquemático do circuito se encontra na figura 3.10, e uma foto do circuito em escala real é apresentada na figura 3.11.

Figura 3.10 - Esquemático completo do circuito



Fonte: Próprio autor

Figura 3.11 - Esquemático completo do circuito



Fonte: Próprio autor

3.4 Desenvolvimento do software

A última etapa antes do dispositivo estar pronto para ser implantado na porta, corresponde a codificação do código fonte, também denominado de software. Ou seja, os comandos lógicos, dados em uma linguagem de programação, responsáveis pelas ações executadas pelo microcontrolador. A linguagem de programação utilizada é baseada em C, apresentada no tópico 2.5.1, com pequenas alterações para facilitar seu uso na placa arduino.

Na primeira parte do código fonte, realizamos a definição das variáveis iniciais. A definição das variáveis iniciais, mostrada na figura 3.12, determina as variáveis, e seus valores, com as quais o software vai começar. Fornecendo toda informação necessária para o seu funcionamento em seu primeiro momento.

Figura 3.12 - Código fonte definição das variáveis iniciais

```

/*===== INÍCIO - DEFINIÇÃO INICIAL DE VARIÁVEIS =====*/
//Definição das portas de input e output
const int RELE = 13;
const int microfone = 8;
const int led_vermelho = 6;
const int led_verde = 7;

//Definição das Variáveis de Configuração
const int reject_value = 25;
const int average_reject_value = 15;
const int intervalo_batidas = 350;
const int intervalo_rele = 1000;
const int max_qtd_batidas = 10;
const int batida_gravada = 10000;

// Definição das Variáveis de Estado
int batida_secreta[max_qtd_batidas] = {100, 100, 100};
int sensor_leitura[max_qtd_batidas];
int button = 48;

/*===== FIM - DEFINIÇÃO INICIAL DE VARIÁVEIS =====*/

```

Fonte: Próprio autor

O primeiro bloco de declaração é formado por 4 constantes, que definem as portas do arduino que serão utilizadas. Como são declaradas como constantes, esperasse que seu valor não mude ao longo de seu funcionamento.

O segundo bloco é reservado para valores de configuração, encontrados por meio de experimentação e ajustes. Em que são determinadas: as margens de erros aceitáveis, o intervalo mínimo entre as batidas, o intervalo de ativação do relé, a quantidade máxima de batidas em uma sequência e o tempo máximo que uma sequência de batidas pode durar. Os valores neste bloco de código também são declarados como constantes, em vista do sistema não fornecer suporte a troca destas configurações.

Por fim, o último bloco de declaração de variáveis, determina as variáveis de estado flexíveis do sistema. Ou seja, os valores que podem mudar ao longo do funcionamento do sistema. Sendo 3 variáveis presentes nesta parte: a senha de

acesso, os intervalos de tempo da sequência de batidas sendo executada e o estado do botão que dita se está ocorrendo uma tentativa de trocar a senha.

Na figura 3.13 é apresentada a função setup. Função ao qual é executada, uma única vez, no momento da inicialização do programa. O uso da função setup é particularmente simples, mas de extrema importância. Estabelecendo se as portas do arduino são de entrada ou saída, ligando o led verde e iniciando a saída serial do arduino.

Figura 3.13 - Código fonte da função setup

```
void setup() {  
    pinMode(RELE, OUTPUT);  
    pinMode(microfone, INPUT);  
    pinMode(led_vermelho, OUTPUT);  
    pinMode(led_verde, OUTPUT);  
  
    digitalWrite(led_verde, HIGH);  
    Serial.begin(9600);  
    Serial.println("Iniciando");  
}
```

Fonte: Próprio autor

A figura 3.14 apresenta a função principal de todo código fonte, a função loop, que estabelece um loop de repetição infinito cuja as atividades podem ser divididas em duas partes. A primeira parte da função verifica e determina o valor da variável de estado button. Variável responsável por identificar se a aplicação está trocando a senha de acesso ou está em seu funcionamento normal.

Figura 3.14 - Código fonte da função loop

```
void loop() {  
  if(Serial.available() > 0){  
    if(Serial.read() == 49){  
      Serial.println("Modo trocar senha ativado!");  
      digitalWrite(led_verde, LOW);  
      button = 49;  
    }else{  
      Serial.println("Modo trocar senha desativado!");  
      digitalWrite(led_verde, HIGH);  
      button = 48;  
    }  
  }  
  
  if(digitalRead(microfone) == LOW) {  
    leituraNovaBatida();  
  }  
}
```

Fonte: Próprio autor

Já a segunda parte da função verifica se houve alguma batida na porta. No caso de ter ocorrido alguma batida, a função loop chama a função leituraNovaBatida, apresentada nas figuras 3.15, 3.16 e 3.17. O código fonte da função leituraNovaBatida também pode ser conferido no apêndice A.

Figura 3.15 - Código fonte da função leituraNovaBatida parte 1

```

/*=====INÍCIO - Função para gravar o tempo das batidas =====*/
void leituraNovaBatida(){
    for (int n = 0; n < max_qtd_batidas; n++){
        sensor_leitura[n] = 0;
    }

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

    digitalWrite(led_vermelho, HIGH);
    delay(intervalo_batidas);
    digitalWrite(led_vermelho, LOW);

```

Fonte: Próprio autor

Figura 3.16 - Código fonte da função leituraNovaBatida parte 2

```

do{
    if(digitalRead(microfone)== LOW){
        digitalWrite(led_vermelho, HIGH);
        tempo_real = millis();
        sensor_leitura[incremento] = tempo_real - tempo_inicial;
        incremento = incremento + 1;
        tempo_inicial = tempo_real;
        delay(intervalo_batidas);
        digitalWrite(led_vermelho, LOW);
    }
    tempo_real = millis() - tempo_primeiro;
}while((tempo_real < batida_gravada) && (incremento < max_qtd_batidas));

```

Fonte: Próprio autor

Na primeira parte da função leituraNovaBatida, o array contendo os dados dos intervalos de tempo das batidas captados pelo sensor de som tem todos os seus valores zerados para que não haja resquícios de informações antigas e já obsoletas. Depois de zerar o array sensor_leitura, ainda na primeira parte do código, são declaradas as variáveis necessárias para o funcionamento da função. Então, na segunda parte da função leituraNovaBatida, são calculados os intervalos entre uma

batida e outra para definir uma nova sequência de batidas. Os intervalos da nova sequência são armazenados no array `sensor_leitura`.

Figura 3.17 - Código fonte da função `leituraNovaBatida` parte 3

```

if(button == 49){
  if(changeSenha() == true){
    Serial.println("senha trocada com sucesso");
  }else{
    Serial.println("falha ao trocar a senha");
  }
}else{
  if(batidaValidacao() == true){
    destrancarPorta();
  }else{
    Serial.println("Erro! Tentativa de entrar falha!");
    failFeedback();
  }
}
}
}
/*=====FIM - Função para gravar o tempo das batidas =====*/

```

Fonte: Próprio autor

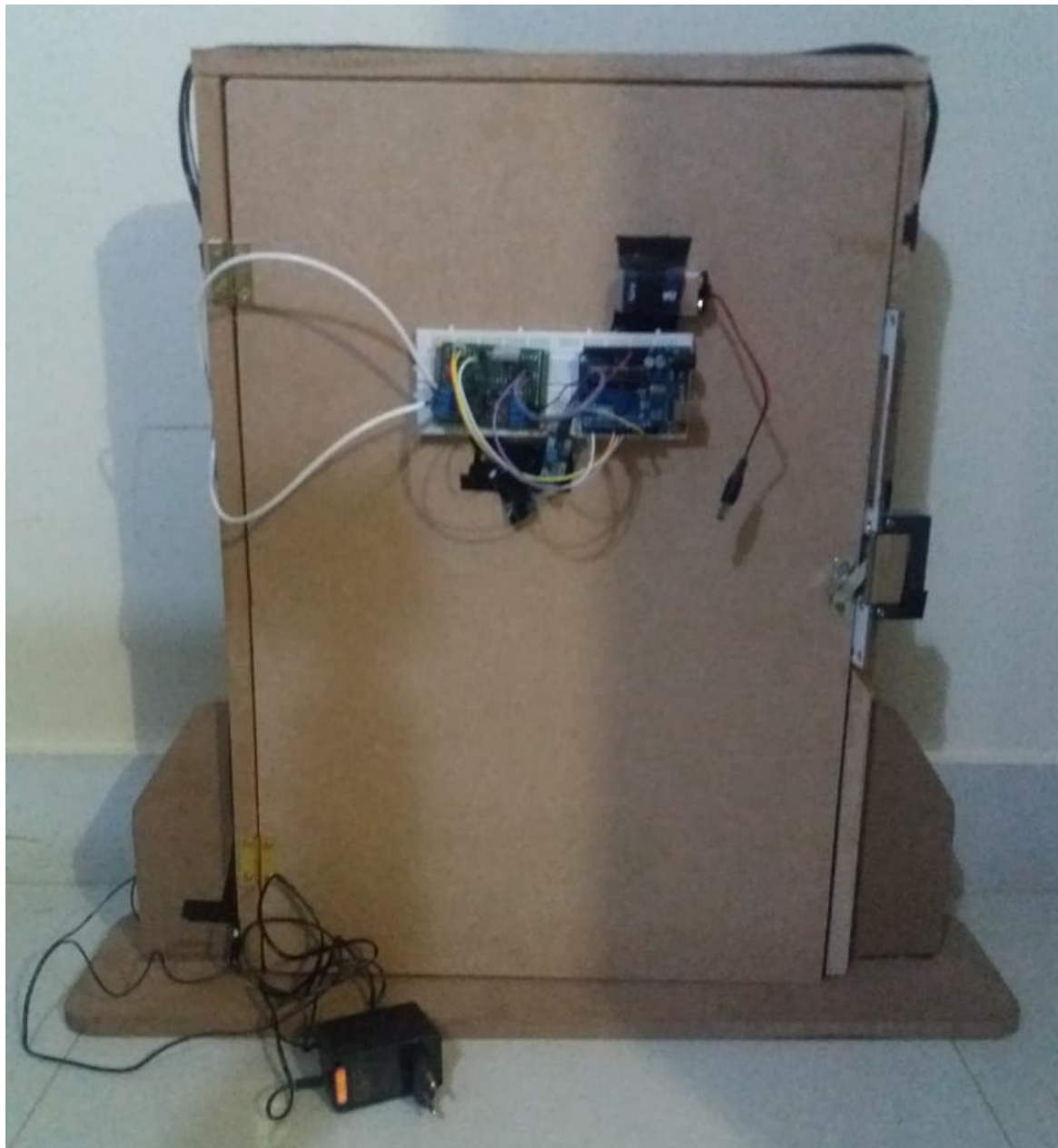
Por fim, a função `leituraNovaBatida` verifica se o usuário está tentando trocar a senha ou abrir a porta através do estado da variável `button`. Se usuário estiver querendo trocar a senha, a função `changeSenha` é chamada, substituindo o valor da senha de acesso pelo da nova sequência de batidas recém realizadas. No caso do usuário querer acesso a entrada, a sequência de batidas, que acabou de ser processada, é comparada com a senha de acesso. Se a sequência de batidas e a senha de acesso forem iguais, respeitando as margens de erro, então o relé é acionado. O código fonte do software pode ser conferido na íntegra no apêndice A.

3.5 Montagem do sistema

Após a conclusão de todas as etapas anteriores do desenvolvimento do projeto, devesse ter como resultado o dispositivo de controle de acesso. Faltando apenas sua instalação em uma porta para que ele possa desempenhar sua tarefa de controlar o acesso a entrada. Sendo que, para fins de teste, será utilizada uma

maquete de porta ao invés de uma porta em escala real, como apresentado na figura 3.18.

Figura 3.18 - Maquete de porta com o dispositivo instalado



Fonte: Próprio autor

4 APLICAÇÃO DA SOLUÇÃO COM RESULTADOS

Nesta fase do trabalho, considera-se que o sistema está inteiramente ou parcialmente desenvolvido e fornecendo algum tipo de resultado. Com o sistema concluído, o próximo passo é a condução de um conjunto de testes para averiguar seu funcionamento.

Neste capítulo serão descritas as condições em que foram realizados esses testes e os métodos utilizados no processo. Assim como, a análise dos resultados. Levando em consideração o atingimento dos objetivos e a os resultados esperados.

4.1 Teste do circuito digital

A primeira fase de testes é dada na construção do circuito digital, onde cada componente tem seu funcionamento avaliado separadamente dos demais. Os testes foram realizados em uma sala fechada contendo apenas um indivíduo, o responsável pelo procedimento. Vale lembrar também, que apesar do circuito final se encontrar em uma placa universal perfurada, preso por solda, no momento dos testes seus componentes foram montados em uma protoboard. Obtendo maior flexibilidade para toda e qualquer alteração necessária.

4.1.1 Teste dos LEDs

Os primeiros componentes testados foram os LEDs por conta da simplicidade do procedimento de teste necessário. Os LEDs foram ligados às portas D6 e D7 do arduino, tomando o devido cuidado de utilizar resistores para que não sejam danificados. Após isso, foi executado o código de teste, apresentado na figura 4.1, que configura as duas portas como portas de saída e manda um sinal alto para os LEDs em intervalos de 1 segundo.

Figura 4.1 - Código fonte para teste de LEDs

```
const int led_vermelho = 6;
const int led_verde = 7;

void setup() {
    pinMode(led_vermelho, OUTPUT);
    pinMode(led_verde, OUTPUT);
}

void loop() {
    digitalWrite(led_vermelho, HIGH);
    digitalWrite(led_verde, HIGH);
    delay(1000);
    digitalWrite(led_vermelho, LOW);
    digitalWrite(led_verde, LOW);
    delay(1000);
}
```

Fonte: Próprio autor

Com o objetivo de simplesmente apagar e acender os LEDs no intervalo de tempo definido. Sempre mostrando o estado que cada LED deveria se encontrar no monitor serial. Desta forma, observou-se o acender e apagar de LEDs na mesma sequência mostrada no monitor serial. Conferindo o funcionamento adequados dos LEDs.

4.1.2 Teste do relé

O relé possui um procedimento de teste muito similar ao dos LEDs. Porém, sua estrutura de funcionamento é muito maior. Tendo a necessidade de um resistor, um transistor e um diodo. Neste sentido, o circuito mais complexo exige uma maior atenção. Considerando que a função do transistor é evitar inconsistência no funcionamento do relé e a função do diodo é evitar que o relé danifique algum

componente devido sua tensão reversa, como abordado no tópico 3.3. Qualquer falha na utilização do transistor ou do diodo, pode acarretar em danos permanentes no circuito.

Para o teste a base do transistor foi ligada à porta D13 do arduino. Após isso, foi utilizado um código de teste, apresentado na figura 4.2, que configura a porta como porta de saída e manda um sinal alto para a base do transistor em intervalos de 1 segundo.

Figura 4.2 - Código fonte para teste de relé

```
const int RELE = 13;

void setup() {
    pinMode(RELE, OUTPUT);
}

void loop() {
    digitalWrite(RELE, HIGH);
    delay(1000);
    digitalWrite(RELE, LOW);
    delay(1000);
}
```

Fonte: Próprio autor

O transistor repassa o sinal para o relé, com o objetivo de acioná-lo no intervalo de tempo definido. Sempre mostrando o estado que o relé deveria se encontrar no monitor serial. Desta forma, observou-se o abrir e fechar dos contatos metálicos dentro do relé na mesma sequência mostrada no monitor serial. Conferindo o funcionamento adequado do relé.

4.1.3 Teste do módulo sensor analógico de som

O terceiro teste realizado foi no componente módulo sensor analógico de som. Para este teste, a presença de uma única pessoa no ambiente do procedimento é de extrema importância. Restringindo ao máximo a quantidade de sons que possam ocorrer perto do sensor. Para o teste, o sensor foi conectado a porta D8 do arduino e teve o seguinte código de teste, presente na figura 4.3, executado.

Figura 4.3 - Código para teste do módulo sensor analógico de som

```
const int microfone = 8;

void setup() {
  pinMode(microfone, INPUT);
  Serial.begin(9600);
  Serial.println("Iniciando");
}

void loop() {
  if(digitalRead(microfone) == LOW) {
    Serial.println("batida");
  }
}
```

Fonte: Próprio autor

Vale lembrar que o próprio módulo já faz a conversão do sinal analógico para digital, resultando em maior simplicidade no código implementado. Então, o código configura a porta do arduino como porta de entrada e verifica se o sensor enviou algum sinal baixo. Indicando a ocorrência de barulho. Com o objetivo de captar todo e qualquer som que ocorrer perto do sensor. No caso da ocorrência de um sinal baixo, o código mostra no monitor serial a palavra “batida”. Desta forma, observou-se o funcionamento adequado do sensor ao verificar a identificação, de quando foi realizado um barulho, no monitor serial.

4.1.4 Teste do módulo bluetooth HC-06

O último componente testado foi o módulo bluetooth HC-06 que, apesar de ser um componente crítico na confecção deste trabalho, tem seu teste relativamente simples. Devido ao módulo HC-06 camuflar toda implementação do protocolo de comunicação bluetooth. Podendo assim nos concentrar apenas em sua utilização.

Para este teste, o módulo teve seus pinos de transmissão e leitura conectados às portas seriais do arduino. Tomando cuidado para que o pino de leitura passe por um divisor de tensão, como abordado no tópico 3.3, devido ao fato do módulo operar em 3,3 volts. Depois, foi executado um código de teste, presente na figura 4.4 logo abaixo.

Figura 4.4 - Código para teste do módulo bluetooth HC-06

```
void setup() {  
  Serial.begin(9600);  
  Serial.println("Iniciando");  
}  
  
void loop() {  
  if(Serial.available() > 0){  
    Serial.print("Mensagem recebida com sucesso! Mensagem: ");  
    Serial.println(Serial.read());  
  }  
}
```

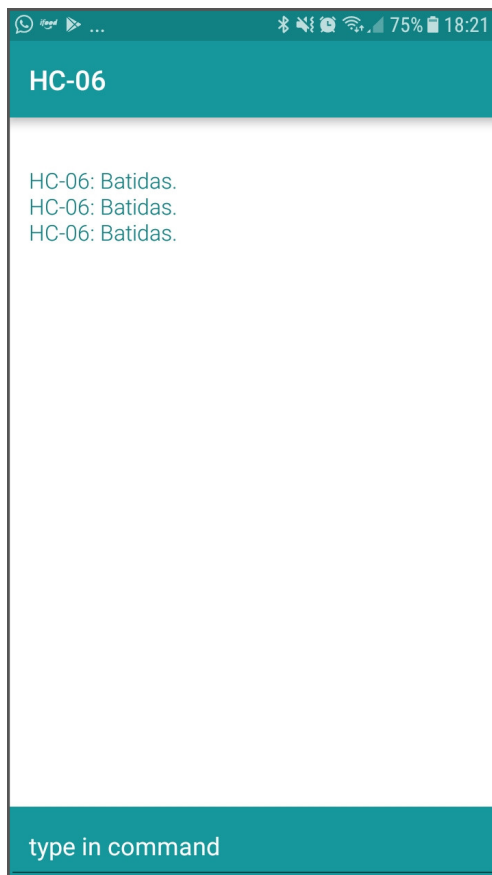
Fonte: Próprio autor

O código verifica se há presença de alguma mensagem na entrada serial do arduino. Caso exista, ele transmite uma resposta na saída serial do arduino, alertando que a mensagem foi recebida com sucesso e informando qual foi a mensagem. O objetivo deste teste é realizar a troca de informações nos dois sentidos, tanto na transmissão quanto no recebimento, entre o arduino e um smartphone através do módulo bluetooth.

Para realizar a troca de informações entre o arduino e um smartphone, é necessário o uso de uma aplicação mobile com tal finalidade. Porém, como dito no

tópico 1.5, não faz parte do escopo deste trabalho o desenvolvimento de tal aplicação. Então, se faz uso de uma aplicação genérica chamada “Arduino Bluetooth Controller”, que possui um conjunto de interfaces pré-definidas para comunicação bluetooth com o arduino. Dentre essas interfaces, há um monitor serial, como mostrado na figura 4.5.

Figura 4.5 - Monitor serial aplicação “Arduino Bluetooth Controller”



Fonte: Próprio autor

Sendo assim, com o aplicativo instalado em um smartphone conectado ao bluetooth do arduino, estamos aptos a realizar a troca de mensagens entre os dois. Observa-se que ao enviar o número 1 pelo monitor serial do smartphone, o texto recebido é “Mensagem recebida com sucesso! Mensagem: 49”. Note que o número 1 na realidade foi reconhecido no arduino como 49. Isto ocorre pois mensagens enviadas, nas circunstâncias descritas neste tópico, são recebidas de acordo com a tabela ASCII pelo arduino.

4.2 Teste do software

Após a construção do circuito digital, juntamente com a confecção dos testes para averiguar o funcionamento do circuito, são realizados os testes referentes ao software desenvolvido. Para os testes do software, considera-se que todo o circuito está inteiramente desenvolvido e já foi testado. Pois o software é executado no microcontrolador do circuito completo e o teste avalia o dispositivo de acesso como um todo.

O ambiente de teste do software é idêntico do circuito, uma sala fechada. Porém, ao invés de conter apenas um único indivíduo, o responsável pelo procedimento, o teste possui duas etapas. Sendo a primeira etapa com um único indivíduo no ambiente e a segunda com cinco pessoas no mesmo ambiente. O objetivo é testar o funcionamento do dispositivo em um ambiente controlado e outro com maior quantidade de barulho.

Durante cada etapa de testes, o microfone do sensor analógico de som é posicionado em cima de uma mesa de madeira para que batidas sejam realizadas logo abaixo da mesa. Simulando a tábua de madeira de uma porta. Dado que um ponto crítico, a ser analisado nesse teste, é a capacidade do sensor de som captar as batidas e ignorar todo e qualquer outro som presente no ambiente.

Na primeira etapa de testes, foram realizadas 40 sequências de batidas logo abaixo da mesa. Batidas similares a senha de acesso estabelecida, com o intuito de ter o acesso liberado. Considerando o acesso liberado como a ativação do relé, que posteriormente irá energizar o fecho eletromagnético instalado na porta. Em seguida, foram realizadas outras 40 sequências de batidas. Porém, diferentes da senha de acesso estabelecida, com o intuito de ter o acesso negado. E o dispositivo teve sua senha de acesso alterada a cada intervalo de 5 sequências de batidas. Verificando o funcionamento da troca de senha.

Os resultados dos testes são apresentados no quadro abaixo. Os resultados bem-sucedidos são aqueles que tiveram o resultado esperado com a realização do teste, e não necessariamente ter o acesso liberado.

Quadro 4.1 - Resultados dos testes da primeira etapa

	Quantidade de testes realizados	Quantidade de resultados bem sucedidos	Porcentagem de resultados bem sucedidos
Sequências de batidas similares a senha de acesso	40	36	90%
Sequências de batidas diferentes da senha de acesso	40	29	72,5%
Tentativas de trocar a senha	16	16	100%

Fonte: Próprio autor

Avaliando os resultados acima, estima-se a porcentagem de sucesso do dispositivo no ambiente com um único indivíduo. Tendo, para as tentativas com a sequência similar a senha de acesso, a porcentagem muito satisfatória de 90% de sucesso. Todavia, para as tentativas com a sequência diferente da senha de acesso, a porcentagem de sucesso foi mais baixa, com 72,5%. E as tentativas de trocar a senha tiveram a taxa de sucesso de 100%.

Ao analisar a baixa porcentagem de sucesso na tentativa de ter o acesso negado, julga-se que a possível causa é a longa margem de tolerância ao erro presente nas regras de validação da senha. Porém, a redução da tolerância ao erro acarretaria na diminuição da porcentagem de sucesso na tentativa de ter o acesso liberado. Uma solução mais viável seria projetar regras de validação mais robustas, que melhorasse a taxa de sucesso, sem prejudicar outras áreas da aplicação.

Por fim, foi realizada a segunda etapa de testes. Seguindo os mesmos parâmetros da primeira etapa, com exceção das cinco pessoas interagindo dentro do ambiente. Os resultados dos testes são apresentados no quadro abaixo.

Quadro 4.2 - Resultados dos testes da segunda etapa

	Quantidade de testes realizados	Quantidade de resultados bem sucedidos	Porcentagem de resultados bem sucedidos
Sequências de batidas similares a senha de acesso	40	33	82,5%
Sequências de batidas diferentes da senha de acesso	40	31	77,5%
Tentativas de trocar a senha	16	12	75%

Fonte: Próprio autor

Analisando os resultados da segunda etapa, observa-se a queda da quantidade de vezes que o acesso foi liberado. Principalmente na ocorrência de sons muito altos, vindo das cinco pessoas no ambiente. Como por exemplo, um grito ou objetos caindo no chão. Baseado nesta análise, julga-se que o sensor analógico de som possui capacidade limitada de ignorar sons do ambiente e captar apenas as batidas. Uma possível solução seria o isolamento acústico do sensor.

Outro resultado divergente, entre as etapas de testes, é a taxa de 75% de sucesso na tentativa de trocar a senha de acesso. No início da aplicação dos testes, a funcionalidade de trocar a senha se mostrou indisponível. Mas, após reiniciar o sistema algumas vezes, a função de trocar a senha voltou ao normal. Sendo então considerada como um problema isolado e sem motivo aparente.

4.3 Teste do sistema

Por fim, na última fase de testes, considera-se que foi conferido o funcionamento do dispositivo de controle de acesso, averiguando tanto seu software quanto seu hardware. E que o dispositivo foi instalado com sucesso na maquete de porta junto com o fecho eletromagnético. Formando o sistema de controle de acesso de acordo com o diagrama de implantação apresentado na figura 3.2.

Os testes realizados no sistema de controle de acesso foram mais simplificados que os demais. Levando em consideração que os testes de uso e o cálculo das taxas de sucesso já foram feitos no tópico 4.2. O ambiente de teste do sistema é novamente uma sala fechada com um único indivíduo dentro, o responsável do procedimento de teste.

Após ter em posse o sistema de controle de acesso pronto, foi verificado se o sensor analógico de som captava corretamente as batidas na maquete de porta. Depois, foram realizadas algumas tentativas de uso, averiguando se o comportamento do sistema seguia de acordo com o que foi presenciado no tópico 4.2. Então, avaliando o comportamento do sistema, chegou-se a conclusão que o sistema como um todo funcionou exatamente como esperado. Identificando corretamente as batidas quando realizadas, e possuindo taxas de sucesso em seus resultados semelhantes ao que foi visto no tópico 4.2.

5 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

As portas são amplamente utilizadas em todo o mundo para fornecer acesso aos mais diversos ambientes. Tendo como vantagens sua praticidade e seu baixo custo. Porém, apresenta alguns problemas em seu sistema de controle de acesso, baseado na existência de uma chave que lhe permite o acesso através de uma fechadura. Pensando nos problemas que um sistema de controle de acesso tão prático e completo como as portas contém, o trabalho proposto procurou apresentar uma nova solução de controle de acesso. Uma solução que resolvesse os problemas do meio antigo, mantendo suas vantagens com relação a praticidade e baixo custo.

Este trabalho atingiu seu objetivo de criar um sistema de controle de acesso por meio de padrões de batidas realizadas na porta. O novo sistema possui um baixo custo, como visto no tópico 3.2 sobre levantamento e obtenção de requisitos. Porém, seu uso é pouco prático. Essa falta de usabilidade é confirmada ao longo de todo o capítulo 4, onde o sistema apresenta uma taxa de sucesso de 72,5% em seus resultados. Deixando o sistema sujeito a possíveis melhorias.

Ainda ao que tange o objetivo geral deste trabalho, o sistema cumpriu o objetivo específico que se refere a troca de mensagens entre o dispositivo de acesso e um aparelho celular. Com uma alta taxa de sucesso em seus resultados de interação com aparelhos celulares por meio de conexão bluetooth, confirmada nos testes realizados no tópico 4.2. Permitindo que o dispositivo tenha sua senha de acesso configurada remotamente e envie mensagens informativas ao celular do usuário.

5.1 Sugestões para trabalhos futuros

Em vista da melhoria da solução do problema proposto neste trabalho, algumas sugestões foram feitas buscando resolver falhas no sistema desenvolvido:

- Produzir código fonte mais robusto e inteligente, focado em melhorar a experiência do usuário e aumentar a taxa de sucesso do sistema.
- Criar uma aplicação para celular específica do sistema de controle de acesso, a fim de melhorar a usabilidade do sistema.

- Confeccionar isolamento acústico, a fim de reduzir o número de sons captados que não são batidas na porta.
- Adicionar um meio de enviar mensagens ao usuário em longas distâncias. Em vista de que a conexão bluetooth funciona apenas a curto alcance.

REFERÊNCIAS

AURESIDE. A Automação Residencial Alavanca a Demanda por Eficiência.

Disponível em:

<http://www.aureside.org.br/noticias/a-automacao-residencial-alavanca-a-demanda-por-eficiencia>. Acessado em: 26 de Março de 2019.

AURESIDE. Vinte Anos de Automação Residencial no Brasil: Um balanço.

Disponível em:

<http://aureside.blogspot.com/2017/11/vinte-anos-de-automacao-residencial-no.html>.

Acessado em: 26 de Março de 2019.

COULOURIS, George E Outros. Sistemas Distribuídos: Conceitos e Projetos.

Porto Alegre: Bookman, 2013.

DANTAS, Leandro Polini; GIMENEZ, Salvador Pinillos. Microcontroladores PIC18: Conceitos, Operação, Fluxogramas e Programação. 1. ed. São Paulo: Érica, 2015.

DINIZ, Paulo S. R.; SILVA, Eduardo A. B. da; NETTO, Sergio L. Processamento Digital de Sinais: Projeto e Análise de Sistemas. 2. ed. Porto Alegre: Bookman, 2014.

DUARTE, Marcelo de Almeida. Eletrônica Analógica Básica. 1. ed. Rio de Janeiro: LTC, 2017.

GUANGZHOU HC INFORMATION TECHNOLOGY. Módulo Bluetooth Serial HC-06 Data Sheet.

HDL LEGRAND. Manual Fecho Eletromagnético FEC-91.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. Algoritmos: Lógica para Desenvolvimento de Programação de Computadores. 28. ed. São Paulo: Érica, 2011.

MONK, Simon. Projetos com Arduino e Android: Use seu Smartphone ou Tablet para Controlar o Arduino. Porto Alegre: Bookman, 2013.

MONK, Simon. 30 Projetos com Arduino. Porto Alegre: Bookman, 2014.

MONK, Simon. Programação com Arduino: Começando com Sketches. 2. ed. Porto Alegre: Bookman, 2017.

MORIMOTO, Carlos. Dicionário Técnico de Informática. 3. ed. Centro de Ciências Exatas e Tecnologia - CCE, Universidade Federal de Viçosa - UFV, Viçosa, 2005.

NALON, José Alexandre. Introdução ao Processamento Digital de Sinais. Rio de Janeiro: LTC, 2009.

PEREIRA, Fábio. Microcontrolador PIC18 Detalhado: Hardware e Software. 1. ed. São Paulo: Érica, 2010.

PEREIRA, Fábio. Microcontroladores PIC: Programação em C. 7. ed. São Paulo: Érica, 2007.

PRUDENTE, Francesco. Automação Predial e Residencial: Uma Introdução. Rio de Janeiro: LTC, 2011.

REGO, Ricardo Affonso do. Eletromagnetismo Básico. Rio de Janeiro: LTC, 2010.

SCHMITZ, Eber Assis; ALENCAR, Antonio Juarez; CRUZ, Leôncio Teixeira. Assistentes Virtuais Inteligentes: Conceitos e Estratégias. Rio de Janeiro: Brasport, 2013.

SILVA, Márcio Gabriel Melo. Avaliação de Desempenho de Relés de Proteção Digitais. 2012. 114 f. Monografia (Graduação). Curso de Engenharia Elétrica, Universidade Federal do Rio de Janeiro - UFRJ, Rio de Janeiro, 2012.

SILVA, Paulo Henrique de Oliveira. Sistema de Segurança de Tranca de Porta e Controle de Acesso. 2013. 78 f. Monografia (Graduação). Faculdade de Tecnologia e Ciências Sociais Aplicadas - FATECS, Curso de Engenharia de Computação, UniCeub, Brasília, 2013.

SOUZA, Adriana Casagrande Motta e. et al. Análise e Modelagem de Processos de Negócio: Foco na notação BPMN. 1. ed. São Paulo: Atlas, 2013.

TUCKER, Allen B.;NOONAN, Robert E. Linguagens de Programação: Princípios e Paradigmas. 2. ed. Porto Alegre: AMGH, 2014.

WENTWORTH, Stuart M. Fundamentos de Eletromagnetismo: Com Aplicações em Engenharia. Rio de Janeiro: LTC, 2006.

Apêndice A - Código fonte do microcontrolador

```
/*=====*/  
Nome: Gabriel Ferreira do Nascimento  
Matrícula: 21316499  
Curso: Engenharia de Computação  
Disciplina: Projeto Final - 1º semestre/2019  
/*=====*/  
  
/*===== INÍCIO - DEFINIÇÃO INICIAL DE VARIÁVEIS =====*/  
//Definição das portas de input e output  
const int RELE = 13;  
const int microfone = 8;  
const int led_vermelho = 6;  
const int led_verde = 7;  
  
//Definição das Variáveis de Configuração  
const int reject_value = 25;  
const int average_reject_value = 15;  
const int intervalo_batidas = 350;  
const int intervalo_rele = 1000;  
const int max_qtd_batidas = 10;  
const int batida_gravada = 10000;  
  
// Definição das Variáveis de Estado  
int batida_secreta[max_qtd_batidas] = {100, 100, 100};  
int sensor_leitura[max_qtd_batidas];  
int button = 48;  
  
/*===== FIM - DEFINIÇÃO INICIAL DE VARIÁVEIS =====*/  
  
/*=====INÍCIO - Função Principal=====*/
```

```

void setup() {
  pinMode(RELE, OUTPUT);
  pinMode(microfone, INPUT);
  pinMode(led_vermelho, OUTPUT);
  pinMode(led_verde, OUTPUT);

  digitalWrite(led_verde, HIGH);
  Serial.begin(9600);
  Serial.println("Iniciando");
}

void loop() {
  if(Serial.available() > 0){
    if(Serial.read() == 49){
      Serial.println("Modo trocar senha ativado!");
      digitalWrite(led_verde, LOW);
      button = 49;
    }else{
      Serial.println("Modo trocar senha desativado!");
      digitalWrite(led_verde, HIGH);
      button = 48;
    }
  }
}

if(digitalRead(microfone)== LOW){
  leituraNovaBatida();
}
}

/*=====FIM - Função Principal=====*/

/*=====INÍCIO - Função para gravar o tempo das batidas =====*/
void leituraNovaBatida(){

```

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

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

digitalWrite(led_vermelho, HIGH);
delay(intervalo_batidas);
digitalWrite(led_vermelho, LOW);

do{
  if(digitalRead(microfone)== LOW){
    digitalWrite(led_vermelho, HIGH);
    tempo_real = millis();
    sensor_leitura[incremento] = tempo_real - tempo_inicial;
    incremento = incremento + 1;
    tempo_inicial = tempo_real;
    delay(intervalo_batidas);
    digitalWrite(led_vermelho, LOW);
  }
  tempo_real = millis() - tempo_primeiro;
}while((tempo_real < batida_gravada) && (incremento < max_qtd_batidas));

if(button == 49){
  if(changeSenha() == true){
    Serial.println("senha trocada com sucesso");
  }else{
    Serial.println("falha ao trocar a senha");
  }
}
```



```

    }
}else{
    if(batidaValidacao() == true){
        destrancarPorta();
    }else{
        Serial.println("Erro! Tentativa de entrar falha!");
        failFeedback();
    }
}
}
}
/*=====FIM - Função para gravar o tempo das batidas =====*/

/*=====INÍCIO - Função para validação da batida=====*/
boolean batidaValidacao(){
    int n = 0;
    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(count != batida_secreta_count){

```

```

    return false;
}

int diferenca_intervalo = 0;
int tempo = 0;

for(n = 0; n < count; 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=====*/

/*=====INÍCIO - Função trocar senha=====*/
boolean changeSenha(){
    int n = 0;
    int intervalo_max_qtd_batidas = 0;

    for(n = 0; n < max_qtd_batidas; n++){
        if(sensor_leitura[n] > intervalo_max_qtd_batidas){

```

```

    intervalo_max_qtd_batidas = sensor_leitura[n];
  }
}
for(n = 0; n < max_qtd_batidas; n++){
  batida_secreta[n] = map(sensor_leitura[n], 0, intervalo_max_qtd_batidas, 0,100);
}
return true;
}
/*===== FIM - Função Converter valor ASCII
=====*/

/*=====INÍCIO - Função para destrancar a porta=====*/
void destrancarPorta(){
  Serial.println("Porta destrancada!");
  digitalWrite(RELE, HIGH);
  delay(intervalo_rele);
  digitalWrite(RELE, LOW);
  delay(intervalo_rele);
}
/*=====FIM - Função para destrancar a porta=====*/

/*=====INÍCIO - Função para feedback de falha=====*/
void failFeedback(){
  digitalWrite(led_vermelho, HIGH);
  delay(350);
  digitalWrite(led_vermelho, LOW);
  delay(350);
  digitalWrite(led_vermelho, HIGH);
  delay(350);
  digitalWrite(led_vermelho, LOW);
  delay(350);
  digitalWrite(led_vermelho, HIGH);
}

```

```
delay(350);  
digitalWrite(led_vermelho, LOW);  
}  
/*=====FIM - Função para feedback de falha=====*/
```