

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO  
GRANDE DO NORTE  
CAMPUS NATAL - ZONA NORTE  
CURSO TÉCNICO DE NÍVEL MÉDIO INTEGRADO EM ELETRÔNICA

ALDIELYSON JORGE CAVALCANTE DE BRITO  
NICODEMOS JONATAS DE MOURA SILVA

**DISPOSITIVO DE SEGURANÇA E IDENTIFICAÇÃO PARA MORADORES DE  
PRÉDIOS E CONDOMÍNIOS (DISIMP)**

NATAL/RN  
2018

ALDIELYSON JORGE CAVALCANTE DE BRITO  
NICODEMOS JONATAS DE MOURA SILVA

**DISPOSITIVO DE SEGURANÇA E IDENTIFICAÇÃO PARA MORADORES DE  
PRÉDIOS E CONDOMÍNIOS (DISIMP)**

Trabalho de Conclusão de Curso apresentado ao Curso Integrado em Eletrônica do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte Campus Natal-Zona Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Técnico Integrado em Eletrônica.

Orientador: Dr. Érico Cadineli Braz

NATAL/RN  
2018

ALDIELYSON JORGE CAVALCANTE DE BRITO  
NICODEMOS JONATAS DE MOURA SILVA

**DISPOSITIVO DE SEGURANÇA E IDENTIFICAÇÃO PARA MORADORES DE  
PRÉDIOS E CONDOMÍNIOS (DISIMP)**

Trabalho de Conclusão de Curso  
apresentado ao Curso Integrado em  
Eletrônica do Instituto Federal de Edu-  
cação, Ciência e Tecnologia do Rio  
Grande do Norte Campus Natal-Zona  
Norte, em cumprimento às exigências  
legais como requisito parcial à obten-  
ção do título de Técnico Integrado em  
Eletrônica.

Trabalho de Conclusão de Curso apresentado e aprovado em 23/11/2018,  
pela seguinte Banca Examinadora:

**BANCA EXAMINADORA**

---

Prof Dr. Érico Cadineli Braz - Presidente  
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

---

Prof Msc. Hilário José Silveira Castro - Examinador  
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

---

Prof Msc. Jean Carlos da Silva Galdino - Examinador  
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

## **AGRADECIMENTOS**

Em primeiro lugar, quero agradecer ao Senhor Deus, pois Ele é a razão das nossas vidas, e o motivo pelo qual chegamos até aqui. Também queremos agradecer aos nossos pais Suely Albuquerque, Rosangela Moura e Francinildo Silva por sempre ter apoiado, motivado e investido em nossa educação. Deixo também meus agradecimentos as nossas famílias como um todo, como também a nossa turma XOR, a qual esteve presente conosco nessa extensa trajetória acadêmica, e se tornou essencial em nossas vidas.

Queremos agradecer ao nosso professor orientador Érico Cadineli, o qual nos instruiu no decorrer desse projeto para que obtivéssemos êxito. Ao nosso colega Anderson Wendell, por sua grande ajuda na elaboração do algoritmo. Ao Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN), por ter nos propiciado um ambiente rico em oportunidades, contribuindo para um pleno desenvolvimento intelectual. Por fim, a todos aqueles que nos ajudaram, de forma direta ou indireta, o nosso muito obrigado.

*“Não fui eu que ordenei a você? Seja forte e corajoso! Não se apavore nem desanime, pois o Senhor, o seu Deus, estará com você por onde você andar”.*

**Josué 1:9**

## RESUMO

Este presente trabalho tem como finalidade a criação de um dispositivo de controle de acesso, a partir do desenvolvimento de um protótipo de fácil utilização. Trata-se de um Dispositivo de Segurança e Identificação para Moradores de Prédios e condomínios (DISIMP), cujo funcionamento se dá através de uma plataforma microcontrolada Arduino. Para esse equipamento, utiliza-se periféricos de interface com o usuário, como o display LCD e o teclado matricial, além disso, seu comando é feito por meio de um Controle transmissor TX4R 3.0 JFL Rolling Code 433,92MHz (FS1000A) via rádio frequência (RF). O grande diferencial desse projeto é um artifício para ações coagidas, isto é, quando a entrada do usuário estiver sendo imposta por alguma ameaça externa. Dessa forma, o DISIMP possui em seu comando um botão de pânico para casualidades dessa magnitude, no qual será emitido um alerta para o porteiro tomar as decisões cabíveis. O equipamento desenvolvido, através da inserção de um cartão de memória Micro SD, possibilita a configuração e armazenamento dos dados de 160 diferentes moradores, com possibilidade de expansão. Nesse sentido, a viabilidade do projeto pode ser demonstrada pelo seu diferencial e praticidade.

Palavras-chaves: Sistema de segurança. Arduino. Controle de acesso. Identificação de moradores. Transmissor RF. Botão de pânico.

## **ABSTRACT**

This work aims to create an access control device, from the development of an easy-to-use prototype. It is a Security and Identification Device for Residents of Buildings and Condos (DISIMP), which operates through an Arduino microcontrolled platform. For this equipment, user interface peripherals are used, such as the LCD display and the matrix keyboard, in addition, its command is made by means of a radio transmitter TX4R 3.0 JFL Rolling Code 433,92MHz (FS1000A) (RF). The great differential of this project is an artifice for coerced actions, that is, when user input is being imposed by some external threat. In this way, DISIMP has in its command a panic button for coincidences of this magnitude, in which an alert will be issued for the doorman to make the appropriate decisions. The equipment developed, through the insertion of a Micro SD memory card, allows the configuration and storage of data of 160 different residents, with possibility of expansion. In this sense, the viability of the project can be demonstrated by its differential and practicality.

Keywords: Security system. Arduino. Access control. Identification of residents. RF transmitter. Panic button.

## LISTA DE FIGURAS

FIGURA 1: ARDUINO UNO. ....	17
FIGURA 2: ARDUINO MEGA2560.....	17
FIGURA 3: MEMÓRIAS EEPROM.....	18
FIGURA 4: SHIELD DA MEMÓRIA FLASH SD CARD PARA ARDUINO. ....	19
FIGURA 5: TRANSMISSOR TX4R 3.0 JFL ROLLING CODE 433,92MHZ JFL. ....	21
FIGURA 6: MÓDULO DO RECEPTOR RF 433 MHZ. ....	22
FIGURA 7: CONEXÃO INTERNA DO TECLADO MATRICIAL. ....	23
FIGURA 8: LCD 16X2. ....	25
FIGURA 9: MODULAÇÃO ASK/OOK.....	25
FIGURA 10: TRECHO DE CÓDIGO FONTE NA IDE PADRÃO DO ARDUINO. ....	27
FIGURA 11: RECEPTOR PROGRAMÁVEL RDL-250 JFL. ....	28
FIGURA 12: SIMULAÇÃO DO CIRCUITO DE TESTE DO LCD 16X2 COM MÓDULO I2C. ....	29
FIGURA 13: SIMULAÇÃO DA CONEXÃO DO MÓDULO RECEPTOR COM O ARDUINO. ....	30
FIGURA 14: SIMULAÇÃO DA CONEXÃO MÓDULO CARTÃO SD COM ARDUINO. ....	31
FIGURA 15: SIMULAÇÃO DA CONEXÃO DO TECLADO MATRICIAL E LCD COM O ARDUINO. ....	32
FIGURA 16: FLUXOGRAMA DO SEQUENCIAMENTO DE AÇÕES DO PROGRAMA. ....	34
FIGURA 17: PROTÓTIPO COMPLETO DO PROJETO DISIMP MONTADO EM PROTOBOARD. ....	38
FIGURA 18: CAIXA PLÁSTICA PATOLA.....	39
FIGURA 19: CONECTORES EMPILHÁVEIS FÊMEA. ....	39
FIGURA 20: PLACA PCI UNIVERSAL.....	40
FIGURA 21: PERIFÉRICOS ACOPLADOS NA PLACA PCI UNIVERSAL. ....	40
FIGURA 22: PARTE EXTERNA DO PROTÓTIPO FINAL.....	41
FIGURA 23: PARTE INTERNA DO PROTÓTIPO FINAL.....	41

## LISTA DE TABELAS

TABELA 1: PINAGEM DO LCD 16X2 .....	24
-------------------------------------	----

## SUMÁRIO

<b>1.INTRODUÇÃO .....</b>	<b>11</b>
1.1 JUSTIFICATIVA.....	13
1.2 OBJETIVO GERAL.....	15
<b>1.2.1 Objetivos específicos .....</b>	<b>15</b>
<b>2. FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>16</b>
2.1 Plataforma Microcontrolada Arduino.....	16
2.2 MEMÓRIAS .....	18
<b>2.2.1 EEPROM.....</b>	<b>18</b>
<b>2.2.2 Memória Flash.....</b>	<b>19</b>
2.3 MECANISMOS DOS SISTEMAS DE SEGURANÇA .....	20
2.4 TRANSMISSOR WIRELESS .....	20
2.5 RECEPTOR WIRELESS .....	21
2.6 TECLADO MATRICIAL .....	22
2.7 DISPLAY DE CRISTAL LÍQUIDO (LCD) .....	23
2.8 MODULAÇÃO ASK/OOK .....	25
<b>3.METODOLOGIA .....</b>	<b>26</b>
3.1 ESCOLHA DA PLATAFORMA MICROCONTROLADA ARDUINO .....	26
3.2 LINGUAGEM DE PROGRAMAÇÃO.....	26
3.3 ESCOLHA DOS PERIFÉRICOS DE ENTRADA E SAÍDA.....	27
3.4 TESTE COM O LCD 16X2.....	28
3.5 TESTE COM O TRANSMISSOR E RECEPTOR WIRELESS .....	29
3.6 TESTE COM A MEMÓRIA SD.....	31
3.7 TESTE COM O TECLADO MATRICIAL .....	32
3.8 PROGRAMAÇÃO .....	32
<b>3.8.1 Ideia Inicial.....</b>	<b>33</b>
<b>3.8.2 Inserção das bibliotecas no Programa .....</b>	<b>34</b>
<b>3.8.3 Implementação de funções e variáveis .....</b>	<b>35</b>
3.9 Montagem do Protótipo .....	37
<b>3.9.1 Construção da Interface.....</b>	<b>38</b>
<b>4.CONSIDERAÇÕES FINAIS.....</b>	<b>42</b>
<b>5.REFERÊNCIAS.....</b>	<b>43</b>
<b>APÊNDICE A – ALGORITMO COMPLETO DO PROGRAMA.....</b>	<b>45</b>

## 1.INTRODUÇÃO

A busca pela segurança sempre esteve presente no cotidiano das sociedades. Desde os primórdios, os povos e civilizações buscavam meios para proteger suas habitações e, com o avanço do tempo, as vilas e consequentes cidades, decorrentes dessa evolução populacional. A procura por esse meio de proteção é resultado das constantes e incisivas invasões de outros povos às posses alheias.

Com o tempo, a preocupação com a preservação dos bens, e consequente integridade física, fez com que as sociedades buscassem meios mais seguros e confiáveis para se precaver contra eventuais incidentes. Com o passar dos séculos, esses meios foram evoluindo, como por exemplo a civilização romana, situada cronologicamente no século V, na qual os mesmos construíram fortes e imponentes muralhas, como também fortificações para se proteger.

Após alguns séculos a realidade é trazida para o cotidiano atual, no qual pode-se citar o Brasil, onde a segurança está com alto déficit em termos de efetivo policial, assim também como a precária estrutura de assistência às ocorrências de segurança. Ademais, também é possível trazer essa realidade para o estado do Rio Grande do Norte (RN), onde a crescente onda de violência vem trazendo sérios malefícios para a população potiguar. Com isso, algumas medidas foram tomadas, tais como a presença do exército nas ruas no final de 2017.

Segundo a Secretaria de Segurança Pública, o estado do RN registrou 2.246 mortes no ano passado – o que representa uma taxa de 64 mortes para cada grupo de 100 mil pessoas. Diante de tal situação, a instalação de sistemas de segurança cada vez mais eficazes, ajudariam a minimizar essa sensação de insegurança fornecida ao cidadão potiguar. Frente a isso, se faz necessário a compreensão da funcionalidade dos sistemas de segurança da nossa atualidade.

Os atuais sistemas de segurança seguem determinações internacionais, que têm por base reafirmar a eficácia e eficiência desses sistemas. Os mesmos têm por base as diretrizes da norma ISO/IEC 17799:2005,

influenciada pelo padrão inglês (British Standard) BS 7799. A atual norma de segurança em vigor é: ABNT NBR ISO/IEC 27002:2013. Em destaque as normas citadas, é possível destacar que existem requisitos básicos para o estabelecimento de um confiável e seguro sistema de segurança. No entanto, “para a proposição dessas normas é necessário estudar os mecanismos que serão usados; os mesmos podem ser: mecanismos de controle físico, mecanismos de controle lógico ou mecanismos de controle de acesso”. (SILVA, 2012).

Neste trabalho utiliza por base o mecanismo de controle de acesso, que será dado por sinais emitidos por um controle móvel. No entanto, dentro dos sistemas de comunicação e controle, há a possibilidade de serem utilizados outros tipos de sistemas, como por exemplo: sistemas biométricos, firewalls e cartões inteligentes.

A escolha do sistema fundamentado em controle remoto tem por base a eficaz segurança que esse dispositivo oferece, tendo em vista que o acionamento e comunicação pode ser feito a uma considerável distância, o que possibilita alguns notáveis benefícios para os usuários, como a ampla acessibilidade, haja vista que pessoas - cadeirantes, por exemplo- possuem uma grande dificuldade em termos de locomoção, e esse controle facilitará esse acesso.

Com o DISIMP, o usuário pode ter total liberdade e confiabilidade para comunicar a entrada em sua propriedade, tudo isso a uma efetiva distância que o protege de eventuais casualidades, que porventura venham a ocorrer.

## 1.1 JUSTIFICATIVA

O interesse por esse estudo surgiu em virtude do atual cenário em nossa sociedade contemporânea, na qual duas vertentes paradoxais vêm sendo largamente evoluídas: a tecnologia e a violência. No entanto, foi visto a importância da tecnologia no intuito de aumentar a segurança (FURTADO, 2002), bem como trazer mais comodidade e praticidade ao usuário.

Essa linha de pesquisa subsidiou a elaboração do projeto intitulado como “Dispositivo de segurança e identificação para moradores de prédios e condomínios - DISIMP”, o qual tem como objetivo geral criar um dispositivo que trouxesse ao usuário comodidade e facilidade na portaria de prédios e condomínios, trazendo consigo também mais segurança. A partir desse intuito, foi criado um sistema que trouxesse praticidade em relação a entrada do indivíduo, ou seja, a sua identificação, que antes era feita rotineiramente através do porteiro, agora se dá através de um cadastro acionado por um controle remoto.

Atualmente, sabe-se que todas as medidas a favor da segurança são válidas. Nesse sentido, o projeto busca minimizar o tempo pelo qual o indivíduo passa em frente a portaria, muitas das vezes se identificando. Algumas empresas realizam treinamentos especializados para seus porteiros, porém, por mais que haja um treinamento adequado, algumas ações podem passar despercebidas pelo mesmo.

Segundo a professora do departamento de psicologia clínica de Brasília Terezinha de Camargo “é necessário que o condomínio forneça ferramentas para que o porteiro possa colocar em prática as medidas de segurança cabíveis”, dessa forma, o DISIMP veio como uma ferramenta de auxílio, tanto ao porteiro como para o morador.

Golpes como: falso carteiro, falso corretor, falso policial entre outros, são comumente utilizados como formas de driblar o sistema de segurança em prédios e condomínios. De acordo com a Secretaria de Segurança Pública, roubos e furtos a condomínios cresceu 56% no estado de São Paulo em 2018, o que de certa forma mostra a importância de dedicarmos atenção a esse assunto. Dentro desse contexto, esse projeto vem como uma ferramenta, na qual cada usuário possui um cadastro específico, onde os

dados cadastrados, no dispositivo, permite a singularidade de acesso dos moradores, e conseqüentemente facilita a comunicação do porteiro ao morador, desse modo, diminui gradativamente a margem de erro e confusão de informações.

Além disso, o diferencial desse projeto é possuir em seu controle remoto um botão de pânico, o qual será acionado se o usuário se sentir coagido no acesso a portaria de seu condomínio. Esse botão de pânico, ao ser ativado, comunicará ao porteiro qual usuário está em perigo, para que o mesmo busque as medidas necessárias para aquela situação. Outro fator importante nesse projeto, como dito anteriormente, é a acessibilidade, no qual pelo fato de ser utilizado um controle remoto, qualquer usuário portador de deficiências físicas poderá utilizá-lo.

Por conseguinte, ao delimitar a criação desse projeto de pesquisa, levou-se em consideração a grande relevância em associar a segurança à comodidade, bem como compreender a importância de ferramentas tecnológicas no auxílio ao porteiro.

## 1.2 OBJETIVO GERAL

Desenvolver um protótipo para a segurança no controle de acesso e identificação de moradores de prédios e condomínios.

### 1.2.1 Objetivos específicos

- Desenvolver um sistema com interface amigável para o usuário;
- Analisar os resultados esperados e os obtidos;
- Testar o sistema em laboratório para comprovar sua eficácia.

## 2. FUNDAMENTAÇÃO TEÓRICA

Para a melhor compreensão do projeto, é necessário que haja o conhecimento prévio de alguns termos, assim também como a compreensão do uso e funcionalidade de alguns dispositivos utilizados. Objetivando o amplo entendimento do projeto, a seguir estão listadas algumas explicações de termos e instrumentos de usabilidade nesse trabalho.

### 2.1 Plataforma Microcontrolada Arduino

O Arduino é uma plataforma microcontrolada de placa única com um conjunto de software para programá-lo. O hardware é basicamente uma plataforma com um microcontrolador Atmel AVR e estruturas de entrada/saída. Em termos práticos, um Arduino é um pequeno computador, o qual você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele. Esse dispositivo, é o que chamamos de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de hardware e software (MCROBERTS, 2011).

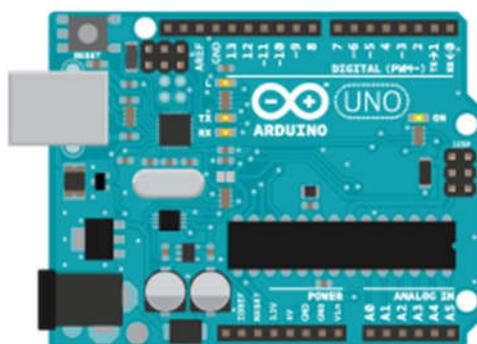
Sua composição dar-se, como relatado anteriormente, por um microcontrolador Atmel AVR, que por sua vez possui um cristal ou oscilador (relógio simples que envia pulsos de tempo em uma frequência especificada, para permitir sua operação na velocidade correta), um regulador linear de 5 volts e uma porta Universal Serial Bus (USB) - permitindo a conexão com o computador. A placa expõe os pinos de entrada/saída do microcontrolador, para que você possa conectá-los a outros circuitos ou sensores (MCROBERTS, 2011).

Um uso simples de utilização do Arduino seria, por exemplo, no acendimento de uma luz por certo intervalo de tempo, no qual se estabeleceria um tempo de 30 segundos, logo após o pressionamento de um botão (nesse exemplo, o Arduino teria uma lâmpada e um botão conectados a ele). Sendo assim, o Arduino iria aguardar o pressionamento do botão, no qual - uma vez pressionado - iria acender a lâmpada e iniciaria a contagem. Após os 30 segundos, apagaria a lâmpada, aguardando um novo comando

de mesma magnitude. Portanto, seria possível a utilização dessa configuração para controlar uma lâmpada em um closet, por exemplo.

Outra característica explorada, é a utilização do Arduino para desenvolver objetos interativos independentes, podendo ser conectado a um computador, redes, ou até mesmo a Internet, no intuito de recuperar e enviar dados do Arduino e atuar sobre eles, como também ele pode ser conectado a LEDs, motores, interruptores, sensores de distância e/ou qualquer outro dispositivo que emita dados ou possa ser controlado. O Arduino pode ser encontrado em diversos modelos e formas, a princípio o processador pode ser diferente, como também a quantidade de portas de entrada/saída, a velocidade de processamento, a tensão máxima e mínima suportada, o tamanho, preço entre outros. O Arduino Uno como mostrado na Figura 1, e o modelo Mega2560 (Figura 2), por exemplo, necessita de um cabo USB regular, enquanto o Arduino Nano necessita de um cabo USB A para Mini-B.

Figura 1: Arduino Uno.



Fonte: Autoria própria.

Figura 2: Arduino Mega2560.



Fonte: Disponível em: <<https://egypt.souq.com/eg-en/arduino-mega-2560-r3-6567186/i/>>

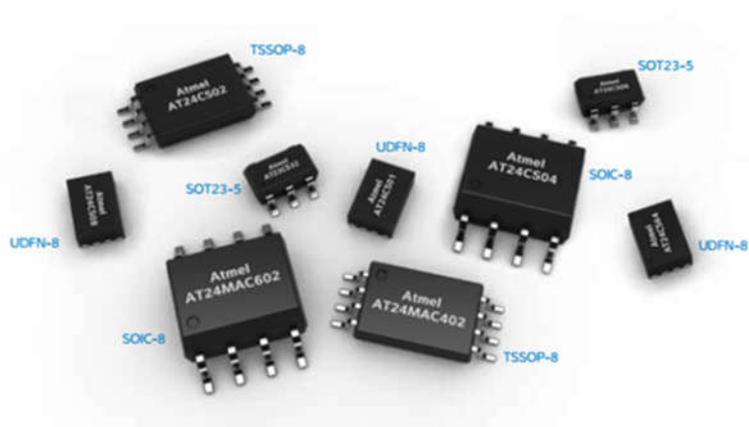
## 2.2 MEMÓRIAS

### 2.2.1 EEPROM

A memória EEPROM (Electrically-Erasable Programmable Read-Only Memory) possui uma característica de ser não volátil, ou seja, os dados não se apagam após o cessamento de energia. Os dados dessa memória podem ser excluídos e reescritos, porém sua exclusão é total, sendo impossível uma seleção de dados para uma nova escrita (Techtarget, 2010).

A maioria dos microcontroladores AVR possuem a memória EEPROM para o armazenamento dos dados. Nas diversas arquiteturas AVR, essa memória interna não é mapeada para o espaço de endereçamento da MCU. Dessa forma, ela só pode ser acessada de uma forma análoga a um dispositivo periférico externo, através da utilização de registradores de ponteiro especiais de leitura/escrita, deixando o seu acesso mais lento se comparado a outra RAM interna. Contudo, alguns dispositivos pertencentes a família SecureAVR (AT90SC) ou até mesmo o Xmega usam, dependendo da configuração, um mapeamento de EEPROM especial para a memória de programas. É importante salientar que o número de escritas na EEPROM é limitado, desse modo, a Atmel especifica 100.000 ciclos de escrita (GOMES, 2011).

Figura 3: Memórias EEPROM.



**Fonte:** Disponível em: <<https://atmelcorporation.wordpress.com/2013/07/30/a-closer-look-at-atmels-at24cs-serial-eprom/>>

## 2.2.2 Memória Flash

A memória flash possui características semelhantes a memória EEPROM, sendo não volátil e apagável, porém possuindo uma maior velocidade para apagar e escrever dados. Nesse sentido, ela é capaz de ser escrita em blocos utilizando o campo elétrico, o qual atua em todo o circuito ou sobre um determinado bloco (CODÁ, 2013, p.1). No entanto, a memória EEPROM só pode ser editada um byte de cada vez, possuindo uma vida útil e capacidade de edição menor em relação a flash.

A capacidade de armazenamento da memória flash varia de acordo com o modelo do Arduino, como por exemplo: o Arduino Uno possui 32kB, no qual 5kB é dedicado ao bootloader (primeiro software executado por um microcontrolador após um reset). No entanto, no Arduino Mega2560 a capacidade de armazenamento é superior, sendo de 256kB, e 8kB dedicado ao bootloader (arduino.cc).

Tanto as memórias não-voláteis já citadas quanto algumas voláteis, como a memória RAM, são memórias que fazem parte do circuito esquematizado do Arduino, porém às vezes essas memórias não são capazes de atender a necessidade de quem está programando, sendo necessário a utilização de shields, que além de possuir bibliotecas prontas, também possibilitam a expansão de memória no Arduino (CAMARGO, 2013). A Figura 4 a seguir, mostra um Shield para memória flash SD card.

Figura 4: Shield da memória Flash SD card para Arduino.



**Fonte:** Disponível em: <<https://tecnosant.com.br/modulo-leitor-cartao-sd-shield-card>>

## 2.3 MECANISMOS DOS SISTEMAS DE SEGURANÇA

Diante das normas dos sistemas de segurança, é necessário estabelecer os três tipos de mecanismos dos mesmos, que dão total subsídio à eficiência das suas utilizações, são eles: mecanismos de controle físico, mecanismos de controle lógico ou mecanismos de controle de acesso.

No mecanismo de controle de acesso, o controle de acesso é responsável por limitar as ações ou operações que um usuário de um sistema computacional pode executar, restringindo o que ele pode fazer diretamente, como também os programas que podem ser executados em seu nome. Tal mecanismo é a base desse projeto, pois dá o poder ao usuário, no que se refere à tomada de ação desejada, na qual o morador irá comunicar sua entrada no condomínio, assim também como pode informar sobre qualquer imprevisto.

O mecanismo de controle lógico consiste em barreiras que impedem ou limitam o acesso à informação, que está em ambiente controlado, geralmente eletrônico, e que, de outro modo, ficaria exposta à alteração não autorizada por elemento mal-intencionado. Já o mecanismo de controle físico conceitua-se como barreiras que limitam o contato ou acesso direto à informação ou à infraestrutura (que garante a existência da informação) que a suporta.

## 2.4 TRANSMISSOR WIRELESS

O transmissor é um dispositivo eletrônico que emite sinais eletromagnéticos e se comunica com o receptor, porém devido às suas características internas e a influência do ambiente externo, a propagação das ondas de rádio frequência sofre variações (GIACOMIN, 2006).

O transmissor sincronizado na mesma frequência do receptor pode mandar dados e sinais, e assim comunicar-se a distância. Nesse projeto iremos utilizar o transmissor de FS1000A, através do Controle TX4R 3.0 JFL Rolling Code 433,92MHz da marca JFL, o qual é mostrado na Figura 5. Esse transmissor funciona com uma tensão de 3V até 12V, quanto maior

tensão, maior a potência e assim maior o alcance dele, além disso ele possui 3 pinos, um sendo o vcc, e os outros são terra e o Data, esse último é responsável pela transmissão de dados do dispositivo. Quando a entrada estiver em nível alto, esse transmissor trabalha em uma frequência de 433 Mhz.

Figura 5: Transmissor TX4R 3.0 JFL Rolling Code 433,92MHz JFL.



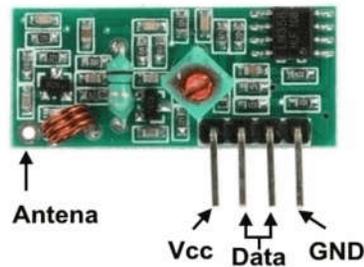
Fonte: Disponível em: <<https://www.dedcomponentes.com.br/control-re-moto-de-port-o-tx4r-433mhz-jfl.html>>

## 2.5 RECEPTOR WIRELESS

Os receptores têm a capacidade de receber o sinal do transmissor se eles estiverem na mesma frequência, isto é, em casos de receptores de rádio frequência. Nesse sentido, nesse projeto será usado um Receptor RF-5v, o qual possui um alcance nominal na faixa de 45cm, podendo ser expandido com a implementação de uma antena.

O receptor RF possui uma alimentação de 5V, no qual o mesmo possui uma largura de banda de 2 Mhz, o pino Data, vcc e o ground (GND). Esse modelo de receptor é largamente usado em placas microcontroladas, pois são de fácil manuseio para programação, e compatíveis com várias plataformas microcontroladas, nesse caso, o Arduino. A Figura 6 a seguir mostra o módulo receptor utilizado.

Figura 6: Módulo do receptor RF 433 MHz.



arduinoocia.com.br

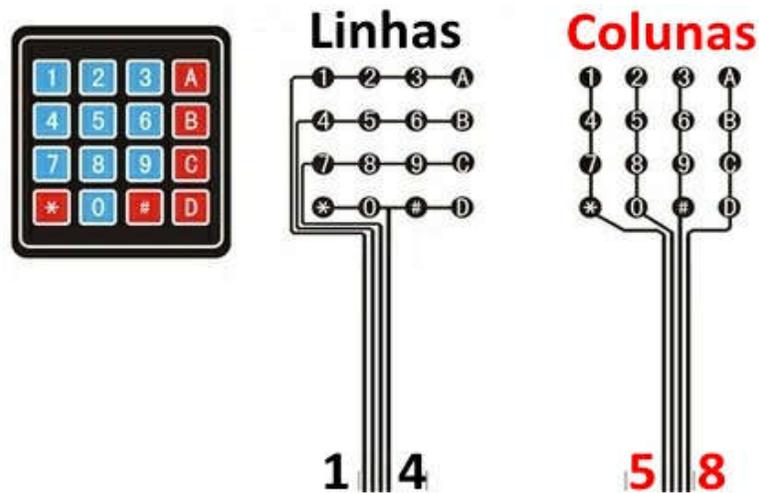
**Fonte:** Disponível em: <<https://www.arduinoocia.com.br/2015/04/comunicacao-sem-fio-modulo-rf-433-mhz.html>>

## 2.6 TECLADO MATRICIAL

É um dispositivo eletrônico, no qual tem como uma das principais finalidades obter saídas digitais organizadas em matrizes, dessa forma, facilitou significativamente o programa em questão, o qual contém múltiplas saídas. Esse apetrecho foi fundamental, pois além da organização das saídas, deixou a interface do protótipo mais simples e prática.

As conexões do teclado matricial são organizada em linhas e colunas. Nesse sentido, a pinagem se dá interligando os quatro primeiros pinos com as 4 linhas, ou seja, o pino 1 liga-se a primeira linha, o pino 2 liga-se a segunda linha e assim segue sucessivamente até o pino 4. Por conseguinte, a partir do pino 5 até o 8 (ou 7 em alguns modelos), é obtida a ligação com as colunas, a qual é tida de forma análoga a organização das linhas. Nesse projeto, utilizamos as 16 teclas, sendo 8 entradas, a Figura 7 a seguir mostra as conexões explicadas anteriormente.

Figura 7: Conexão interna do teclado matricial.



**Fonte:** Disponível em: <<https://www.filipeflop.com/blog/teclado-matricial-4x4-arduino/>>

Ao pressionar as teclas, é estabelecido as conexões entre as linhas e as colunas. Desse modo, se apertarmos a tecla B no teclado matricial, vão se estabelecer as ligações entre o pino 2 (linha 2) e o pino 8 (coluna 4), seguindo assim a mesma linha de raciocínio para as demais teclas.

## 2.7 DISPLAY DE CRISTAL LÍQUIDO (LCD)

Em 1888, o botânico Friedrich Reinitzer observou pela primeira vez a fase líquido-cristalina utilizando o benzoato de colesterila. Desse modo, notou-se que tal substância apresentava dois diferentes pontos de fusão, uma vez que ela se fundia a 146,6 C, tendo como resultado um colorido fluido turvo, no qual posteriormente se fundia outra vez a 180,6 C, originando um líquido isotrópico claro (ASADA, 1990).

Nesse sentido, em 1889, Otto Lehman observou as substâncias oleato de amônia e o p-azoxi-fenetol, e constatou que as mesmas se fundiam passando por um estado intermediário, no qual o líquido era birrefringente. Como Lehman presumia que a distinção entre cristais sólidos e cristais líquidos era o grau de fluidez, ele denominou assim o termo cristal líquido para essas substâncias, as quais apresentavam um estado de orientação intermediário (ASADA, 1990).

Em face do exposto, o LCD, como o próprio nome sugere, baseia-se justamente no estado líquido-cristal como base de funcionamento. A partir daí, configura-se um display de ampla importância para qualquer projeto que necessite visualizar a leitura de uma variável. Esse dispositivo possui módulos gráficos e de caracteres, nos quais os módulos gráficos são encontrados no mercado com resoluções de 122x32, 128x64, 240x64 e 240x128 pixels, e geralmente estão disponíveis com 20 pinos para conexão. Os LCDs comuns do tipo character são diferenciados em número de linhas por colunas (BARBACENA; FLEURY, 1996, p.1).

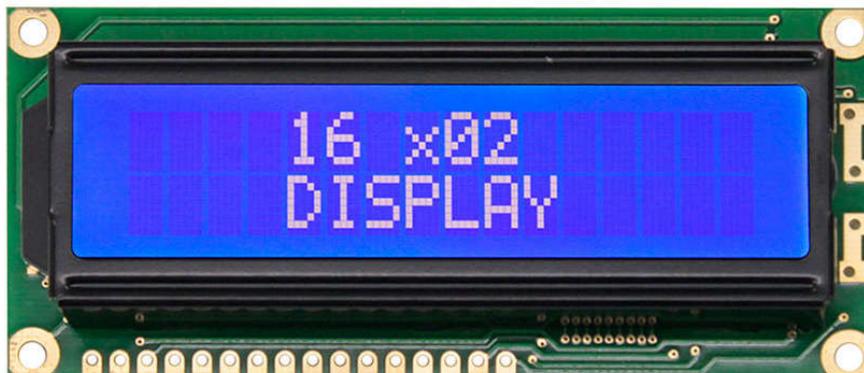
O LCD é conectável a diversos tipos de plataforma microcontrolada. Desse modo, O modelo utilizado foi o LCD 16x2 aderido à plataforma Arduino, esse tipo de display apresenta 16 caracteres dispostos em 2 colunas, no qual a conexão com plataforma microcontrolada se dá através dos 16 pinos, sendo 12 para interligação básica, 2 para o backlight e 1 para o contraste. Na tabela 1 a seguir é mostrada a configuração da pinagem, e o LCD 16x2 é mostrado na Figura 8.

Tabela 1: Pinagem do LCD 16x2.

Conexões LCD 16x2 - HD44780		
Pino LCD	Função	Ligação
1	Vss	GND
2	Vdd	Vcc 5V
3	V0	Pino central potenciômetro
4	RS	Pino 12 Arduino
5	RW	GND
6	E	Pino 11 Arduino
7	D0	Não conectado
8	D1	Não conectado
9	D2	Não conectado
10	D3	Não conectado
11	D4	Pino 5 Arduino
12	D5	Pino 4 Arduino
13	D6	Pino 3 Arduino
14	D7	Pino 2 Arduino
15	A	Vcc 5V
16	K	GND

**Fonte:** Disponível em: <<https://www.arduinoecia.com.br/2013/12/lcd-16x2-arduino-hd44780.html>>

Figura 8: LCD 16x2.

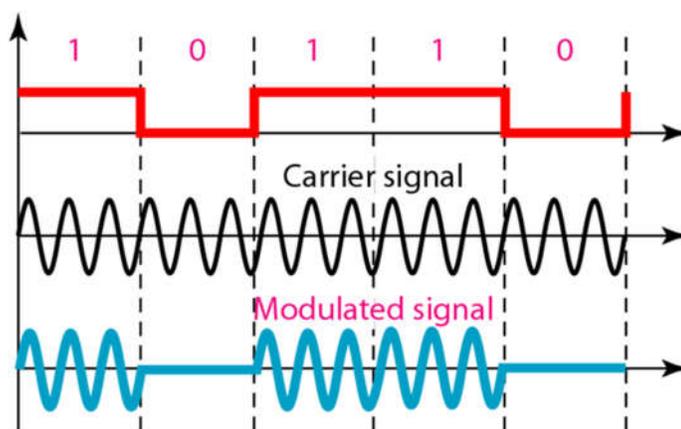


**Fonte:** Disponível em: <<http://www.winstar.com.tw/products/character-lcd-display-module/wh1602j.html>>

## 2.8 MODULAÇÃO ASK/OOK

Esse tipo de modulação é uma técnica utilizada quando se pretende transmitir uma informação digital, mas utilizando uma portadora analógica. Desse modo, sua modulação é concebida de acordo com os desvios de amplitude, permitindo ou não a transmissão da portadora em função da sequência dos bits digitais (HUANG, 2010). A Figura 9 a seguir exemplifica o processo de modulação ASK/OOK.

Figura 9: Modulação ASK/OOK.



**Fonte:** Disponível em: <<http://www.myreadingroom.co.in/notes-and-studymaterial/68-dcn/749-digital-to-analog-conversion-techniques.html>>

### 3.METODOLOGIA

Os instrumentos eletrônicos utilizados neste projeto foram: Arduino Mega2560, Teclado Matricial, LCD 16x2, Transmissor, receptor e uma memória flash. O projeto foi dividido em várias etapas, dentre as quais descreveremos a seguir.

#### 3.1 ESCOLHA DA PLATAFORMA MICROCONTROLADA ARDUINO

Inicialmente, foi realizado uma pesquisa acerca do funcionamento do Arduino, como também seus componentes e suas respectivas conexões. Foi analisado a facilidade que esse dispositivo traria no que diz respeito ao andamento do projeto, verificando a grande diversidade de funções e ferramentas de hardware e software que ele possui, as quais são essenciais para a prototipagem do sistema.

Além disso, a facilidade de acesso e inserções de bibliotecas foi um dos motivos pelo qual optamos pela plataforma Arduino, pois otimiza muito o tempo do projeto, visto que as comunicações e comandos relativos aos periféricos inseridos já estão prontas. Portanto, após ter um bom embasamento dessa plataforma microcontrolada, passamos a construção do projeto iniciando sua montagem e testes em protoboard. Esses testes foram feitos em etapas, pois a medida que íamos progredindo no algoritmo, também era visto os resultados na prática, através da inserção dos dispositivos periféricos no Arduino.

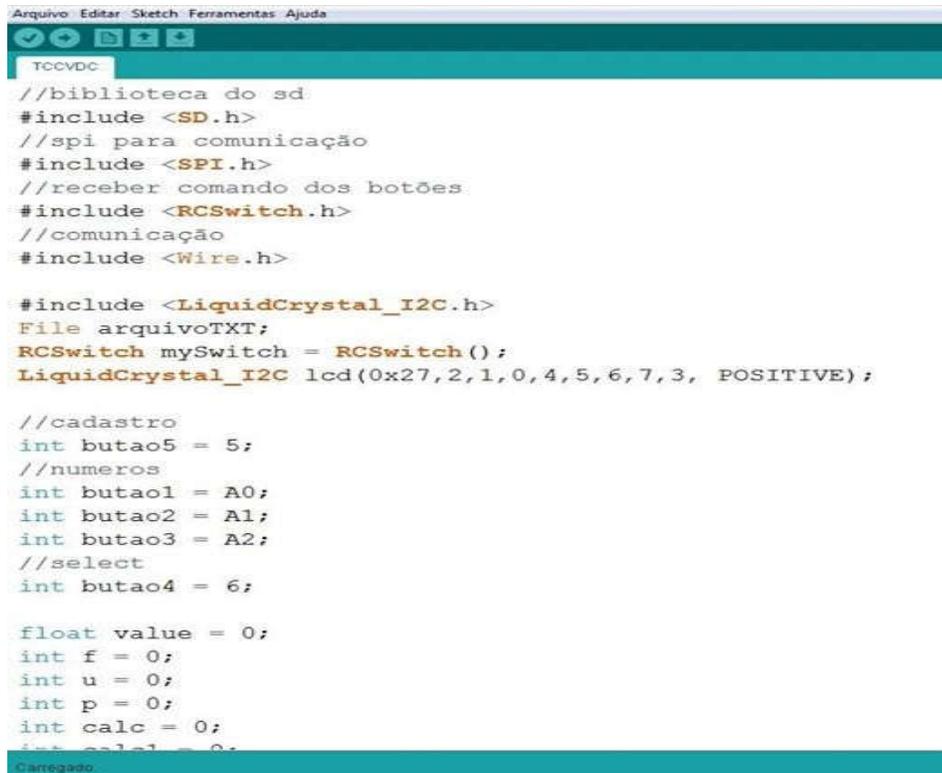
#### 3.2 LINGUAGEM DE PROGRAMAÇÃO

No desenvolvimento do algoritmo do projeto, utilizou-se a linguagem de programação referente ao Arduino. Tal código se baseia em Wiring, o que por sua vez, trata-se de um framework de código aberto, o qual é utilizado para programar microcontroladores (Wiring, 2017).

Como pode ser visto na Figura 10, a programação do projeto deu-se no ambiente conhecido como Arduino Software IDE (Integrated Development Environment, traduzindo: Ambiente de Desenvolvimento

Integrado), o qual constitui o ambiente de desenvolvimento padrão da plataforma microcontrolada Arduino, que por sua vez, é escrita em java e baseado em Processing (Processing, 2017).

Figura 10: Trecho de código fonte na IDE padrão do Arduino.



```

Arquivo Editar Sketch Ferramentas Ajuda
TCCVDC
//biblioteca do sd
#include <SD.h>
//spi para comunicação
#include <SPI.h>
//receber comando dos botões
#include <RCSwitch.h>
//comunicação
#include <Wire.h>

#include <LiquidCrystal_I2C.h>
File arquivoTXT;
RCSwitch mySwitch = RCSwitch();
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

//cadastro
int butao5 = 5;
//numeros
int butao1 = A0;
int butao2 = A1;
int butao3 = A2;
//select
int butao4 = 6;

float value = 0;
int f = 0;
int u = 0;
int p = 0;
int calc = 0;
int calcul = 0;
Carregado

```

**Fonte:** Autoria Própria.

### 3.3 ESCOLHA DOS PERIFÉRICOS DE ENTRADA E SAÍDA

Na escolha dos dispositivos de entrada e saída que iria compor esse projeto, foi tomado como base um projeto já existente no mercado, que foi o Receptor programável RDL-250 da marca JFL, como mostra a Figura 11. Ao analisar esse dispositivo, foi visto que o mesmo possui uma interface simples, composta por um teclado matricial e um display LCD.

Figura 11: Receptor programável RDL-250 JFL.



**Fonte:** Disponível em: <<http://www.jfl.com.br/produto-alarmes-receptor-rdl-250>>

Porém, com as instruções da matéria de Comunicação Eletrônica, foi analisado que para haver uma comunicação entre pelo menos dois pontos se faz necessário possuir uma fonte de informação, a qual pode ser em forma de voz, imagens ou dados; um terminal fonte (transmissor) necessário para transmitir a informação; um terminal destino(receptor) responsável por receber a informação; como também um canal de comunicação para transmitir os dados. Nesse sentido, foi facilitada a compreensão acerca do funcionamento interno do dispositivo citado anteriormente, e escolhido os periféricos transmissor e receptor para compor esse projeto.

Dentro desse contexto, foi necessário também a inserção de um shield de cartão de memória SD, pois nesse projeto visava-se o armazenamento de 160 apartamentos. Sendo assim, tivemos que realizar a troca da memória EEPROM que estávamos usando inicialmente, pelo fato da mesma não ter capacidade suficiente para abarcar essa quantidade de informações, como também pelo rápido poder de processamento da memória flash SD em relação a EEPROM.

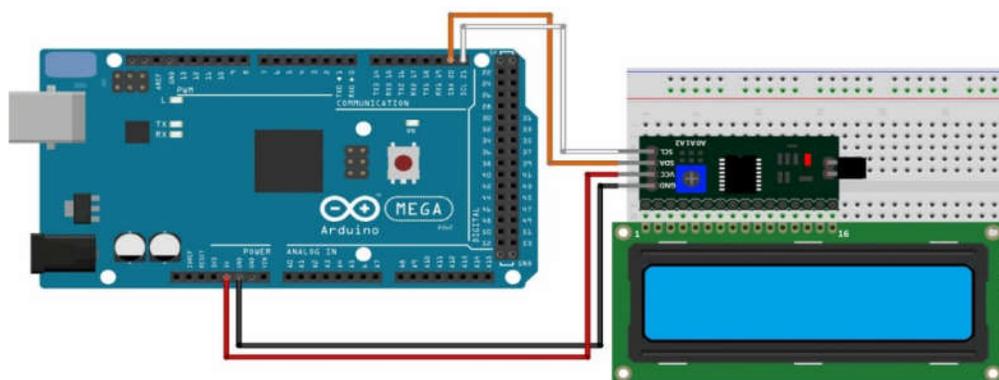
### 3.4 TESTE COM O LCD 16X2

Como dito anteriormente, foi escolhido o display LCD 16x2 para compor a interface do projeto. Sendo assim, o próximo passo foi estudar acerca do seu funcionamento, e iniciar sua implementação no algoritmo. Desse modo, foi inserido a biblioteca <LiquidCrystal\_I2C.h>, a qual além

de fornecer os comandos para execução do programa, também traz consigo as funcionalidades específicas do módulo I2C, que utilizamos para otimizar a conexão do display LCD com o Arduino.

Vale ressaltar que foi fundamental ter embasamento de alguns comandos importantes do LCD, como o `lcd.print("BEM VINDO")`, o qual manda para o LCD a instrução de mostrar uma mensagem em seu display; o `lcd.begin(16,2)` que estabelece a configuração do LCD disposto em dezesseis colunas e duas linhas; como também o `lcd.setCursor(0,0)`, o qual posiciona o cursor na linha e coluna pretendida; e o `lcd.clear()` responsável por limpar qualquer informação que porventura tenha sido ou não utilizado em outro dado momento, e possa vir a interferir no resultado esperado no display. Na Figura 12 a seguir, é mostrado a simulação de como foi feita a montagem adequada do circuito, feito no intuito de realizar os testes com o LCD de acordo com as modificações no programa.

Figura 12: Simulação do circuito de teste do LCD 16x2 com módulo I2C.



Fonte: Autoria Própria.

### 3.5 TESTE COM O TRANSMISSOR E RECEPTOR WIRELESS

Nesse tópico será abordado também o processo de testes, porém com o transmissor e receptor. Nesse sentido, foi visto que suas implementações foram fundamentais, pois através deles foi-se possível realizar uma comunicação sem fio, numa frequência de 433MHz. Vale salientar que foi uma fase bastante importante no tocante à segurança do sistema em

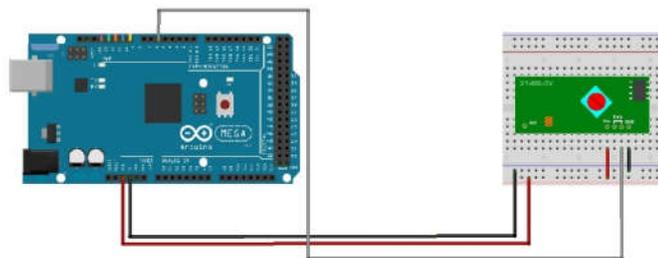
questão, pelo fato de ter que lidar com os testes de 3 controles transmissores da modelo TX4R 3.0 JFL Rolling Code JFL, objetivando a verificação e eficácia do algoritmo.

Durante essa fase, foi possível concluir que cada controle tem uma codificação específica de 28 bits, desse modo, quando houvesse a realização do processo de cadastro do controle transmissor, automaticamente seria salvo no programa esse código. Dessa forma, seria improvável outro controle com sua respectiva codificação conseguir burlar o sistema em contexto, uma vez que seu cadastro não fora realizado.

O teste do receptor, primeiramente, consistiu na implementação da biblioteca <Virtual.Wire>, a qual usa a modulação ASK/OOK. No entanto, percebeu-se que essa biblioteca não abarcava as instruções necessárias à comunicação do modelo do transmissor que utilizamos. Nesse sentido, foi necessário optarmos pela biblioteca <RC.switch>, como também entender a diferença dessas duas bibliotecas. Sendo assim, analisamos que a <Virtual.Wire> está voltada a funcionalidade de modem, replicando assim a comunicação serial do Arduino, através de ondas RF.

A <RC.switch>, por sua vez, foi criada com o propósito específico de automação residencial. Em virtude dessa última característica, a biblioteca <RC.switch> possui em seu código os algoritmos responsáveis pela comunicação de transmissores, lâmpadas e interruptores que estão voltados para esse contexto. Na Figura 13 a seguir, é mostrada a simulação de como foi realizada a conexão do módulo receptor RF-5V com o Arduino, o qual foi necessário para a realização dos testes.

Figura 13: Simulação da conexão do módulo receptor com o Arduino.



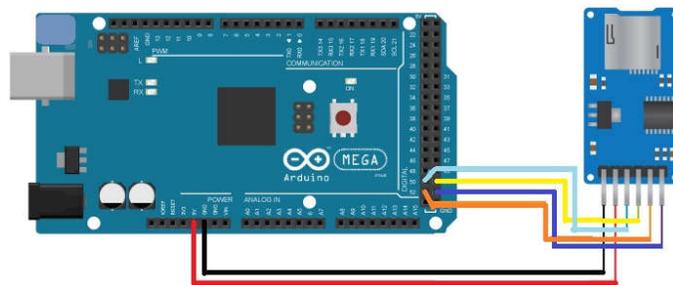
Fonte: Autoria Própria.

### 3.6 TESTE COM A MEMÓRIA SD

A inserção da memória Flash SD foi um passo essencial no programa, pois como dito anteriormente, essa memória possibilita o armazenamento seguro dos dados dos controles. De base na premissa destacada, sua escolha foi realizada com base no algoritmo em questão, e conhecimentos acerca das memórias internas do Arduino e suas respectivas funcionalidades. Sendo assim, foi analisado que existem três locais de memória usado nas plataformas microcontroladas Arduino, as quais são: a memória flash, dedicada ao armazenamento do programa; a SRAM, local onde durante a execução é criada e manipulada as variáveis; e a EEPROM, destinado ao armazenamento de informações de longo termo, como também dos valores do programa.

Com o conhecimento acerca do funcionamento das memórias internas do Arduino, foi possível analisar que as conexões estão todas interligadas, dessa forma, qualquer possível falha no sistema poderia acarretar perdas de informações. Nesse sentido, visando a integridade dos dados do usuário, optamos pela expansão da memória flash, utilizando o módulo cartão SD. Além disso, as informações cadastradas poderiam ser acessadas externamente. Na Figura 14 a seguir, é mostrado a simulação da conexão que realizamos para o teste do Shield do cartão SD com o Arduino.

Figura 14: Simulação da conexão módulo cartão SD com Arduino.



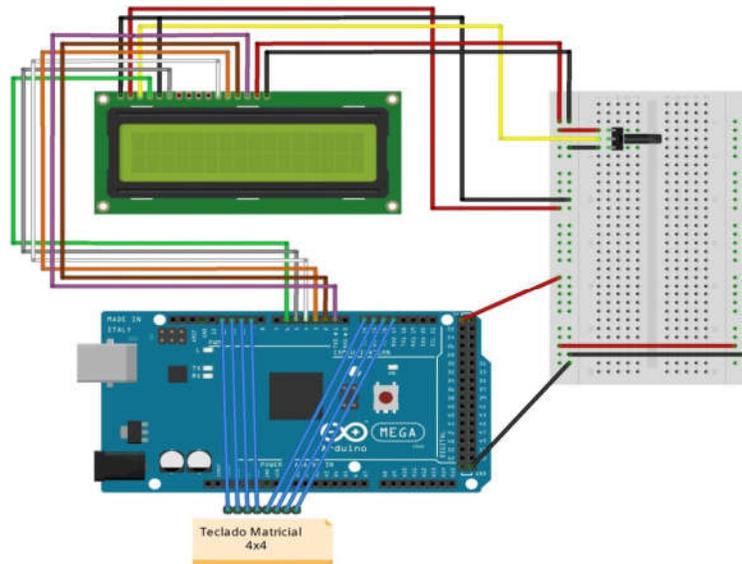
Fonte: Autoria Própria.

### 3.7 TESTE COM O TECLADO MATRICIAL

Nessa fase, o teste realizado foi em virtude da implementação do teclado matricial 4x4. Nessa etapa, verificamos de antemão a saída dos números pressionados, analisando se suas respectivas saídas apareciam no monitor serial da IDE do Arduino. Nesse sentido, após essa análise, foi verificado também suas saídas expressas no LCD com suas devidas conexões, como mostrado na simulação da Figura 15.

A biblioteca inserida no algoritmo foi a “Keypad.h”, a qual é responsável pelos comandos referentes ao teclado matricial. Desse modo, vale salientar que a declaração das teclas deve ser feita por meio da variável: `char teclas [Linhas][colunas] = {}`. Por conseguinte, o espaço entre os “{}” aparecem todas as teclas que o teclado possui.

Figura 15: Simulação da conexão do Teclado Matricial e LCD com o Arduino.



Fonte: Autoria Própria.

### 3.8 PROGRAMAÇÃO

Esse tópico visa mostrar a ideia descrita do projeto, como também algumas funcionalidades e junções dos periféricos já comentados

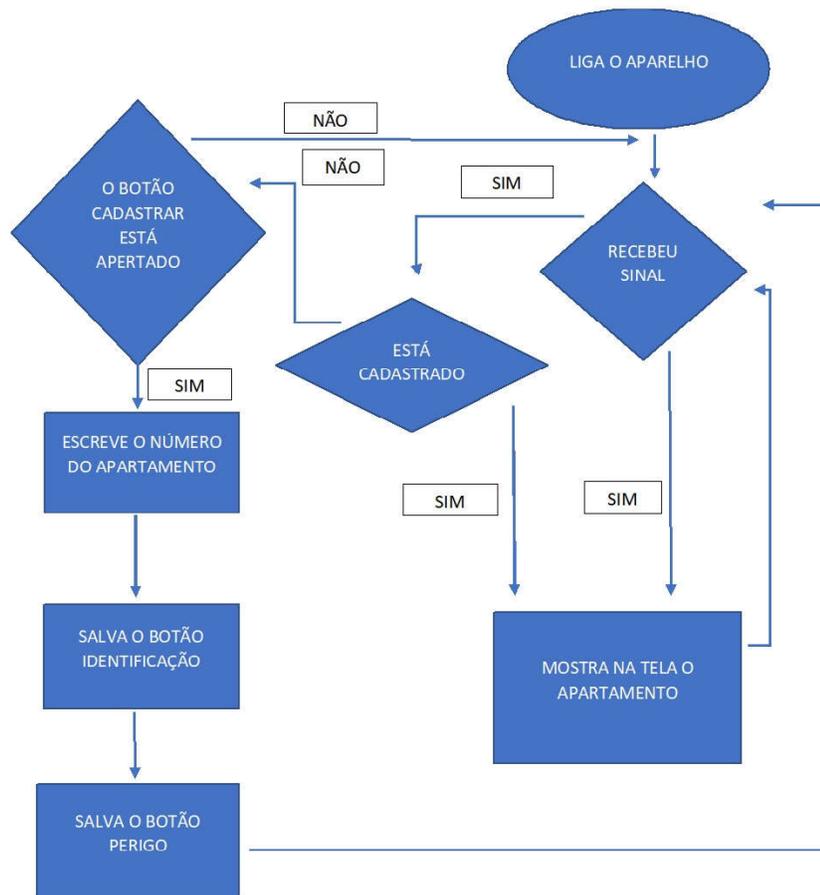
anteriormente, servindo como subsidio para o tópico responsável pelo algoritmo propriamente dito.

### 3.8.1 Ideia Inicial

A pretensão ao construir o algoritmo, é de cadastrar cerca de 160 apartamentos, conseqüentemente 160 controles, porém como o projeto tem o diferencial de um botão extra para casos de emergências, o número de botões duplicou. Para esses cadastramentos, foi pensando na maneira mais simples e otimizada, ou seja, foi cogitada a ideia de um sistema que possuísse um botão para entrar em modo “Cadastro”, e nesse estágio seria digitado o número do apartamento.

No projeto inicial foi pensado em utilizar três botões para digitar esse número, onde cada um deles correspondente a um valor diferente, um para a centena, outro para dezena e o último para a unidade, porém mudamos para o teclado matricial como veremos posteriormente. Após o valor digitado, o usuário pressionaria o seu controle remoto de forma a enviar o código numérico ao sistema, o aparelho receberia esse código, e então seria apertado o próximo botão do controle de forma a registrar o código de perigo. Em conjunto com os botões, foi utilizado um display de 16x2 para aparecer as informações necessárias na tela, como o apartamento, as instruções para o cadastramento de controle e a mensagem de perigo. Além disso, foi utilizado a memória flash SD para garantir o acesso externo dos dados e salva-los com maior segurança. O Fluxograma sequenciando as etapas do algoritmo é mostrado na Figura 16 a seguir.

Figura 16: Fluxograma do sequenciamento de ações do Programa.



Fonte: Autoria Própria.

### 3.8.2 Inserção das bibliotecas no Programa

Ao iniciar o programa, foi necessário declarar algumas bibliotecas pertinentes aos periféricos de entrada e saídas que utilizamos. Nesse sentido, usamos as bibliotecas <SD.h>, <SPI.h>, <RCSwitch.h>, <Wire.h>, <LiquidCrystal\_I2C.h>, <Keypad.h>, as quais suas especificidades serão descritas a seguir.

A biblioteca <SD.h> é responsável por incluir os comandos relacionados à memória SD, desse modo, ela também suporta sistemas de arquivos FAT16 e FAT32 (nomes de sistemas de arquivos utilizados por padrão em versões ultrapassadas, como do sistema operacional windows 98) em cartões SD e SDHC padrão. Essa biblioteca usa um short 8.3 para nomes de arquivos, as quais passam as funções quando se inclui caminhos

separados por barras, como: "diretório / nome\_do\_arquivo.txt", porém como o diretório está sempre contido no cartão SD, utilizamos tanto "/file.txt", como "file.txt".

Para realizar as ações de comunicação entre o microcontrolador e o cartão SD, foi necessária usar a biblioteca <SPI.h>, a qual se constitui como um protocolo de dados serial síncrono, no intuito de fazer com que a comunicação de microcontroladores com outros dispositivos em curtas distâncias, seja rápida.

No objetivo de incluir os comandos que operam tanto o emissor com o receptor numa frequência de 433Mhz, foi utilizada a biblioteca <RCSwitch.h>. Ela foi criada justamente com o propósito de interação com dispositivos de automação residencial, e nesse contexto, foi utilizada para realizarmos as funções relacionadas com o receptor e o transmissor, que por sua vez estão responsáveis pelas informações oriundas do controle RF.

Na finalidade de controlar o LCD com o Arduino, foi preciso implementar a biblioteca <LiquidCrystal\_I2C.h>, a qual se baseia no chipset Hitachi HD44780 (encontrado na maioria dos LCDs que são baseados em textos). A biblioteca em questão funciona em dois modos, sendo eles o de 4 bits, ou o de 8 bits, no qual nesse projeto, utilizamos a de 4 bits pelo fato de reduzir o número de portas e ter uma melhor otimização do projeto.

Nesse projeto foi necessário utilizar o módulo I2C, no intuito de estabelecer as conexões como o LCD de forma mais otimizada. Em face do exposto, foi necessário a inserção da biblioteca <Wire.h>, a qual permite a comunicação com dispositivos I2C / TWI. Porém, vale salientar que a implementação dessa biblioteca usa um buffer de 32 bytes, logo, qualquer tentativa, em uma transmissão, que exceder esse limite, será descartado. Além disso, também inserimos a biblioteca <Keypad.h>, que por sua vez, é responsável pelo comandos e atribuições do teclado matricial.

### **3.8.3 Implementação de funções e variáveis**

Após declararmos as bibliotecas a serem utilizadas, declaramos o nome do cartão de memória a ser utilizado com "*File arquivoTXT*",

iniciamos a biblioteca de comunicação RF através de `"RCSwitch mySwitch = RCSwitch()"`. Foi iniciado também o LCD com `"LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE)"`, declaramos quantas linha e quantas colunas tem o teclado matricial utilizado, identificamos os pinos em que cada uma das linhas e das colunas estão, e declaramos o que cada posição na matriz significa. Depois da inicialização dos periféricos, declaramos os dois botões externos ligados aos pinos A5 e A7 do Arduino mega2560, depois as variáveis globais do código, as quais totalizam 15, que serão explicados ao longo da explicação da lógica do programa.

Iniciamos o void `setup()` para configurações e ações que o programa realiza. Ao iniciar, definimos os botões como saídas declarando `"pinMode(butao5, INPUT)"`, configuramos a velocidade de comunicação serial entre o Arduino e o receptor, como também entre Arduino e o LCD usando a variável `"Serial.begin(9600)"`, iniciamos também o LCD declarando quantas colunas e linhas ele possui. Nesse sentido, declaramos o pino que aciona o módulo SD `"SD.begin(53)"` e habilita a recepção de sinais com o receptor, o qual está ligado ao pino de interrupção "0", sendo este o pino digital 2 configurado como `"mySwitch.enableReceive(0)"`.

Iniciamos o void `loop()`, que é a parte do código que fica em constante repetição, logo após, acendemos a luz de fundo do LCD com `lcd.setBacklight(HIGH)`. Posteriormente a isso, temos um "if" que verifica se o botão de cadastro foi pressionado, se não, ele vai para a rotina de verificação de sinais, em que verifica-se usando o `"if (mySwitch.available())"`, desse modo, analisa se há algum sinal no receptor, se não ele, volta ao começo do loop, se sim ele salva esse valor na variável "value". Após isso, verifica se "value" é igual a 0, se sim, manda uma mensagem de erro ao LCD, se não, ele abre o arquivo desejado que nomeamos "texto.txt", e entramos em um laço de repetição, no qual é verificado cada uma das linhas do arquivo de texto. Por conseguinte, salva o valor da linha em um vetor de caracteres nomeado "letra" que foi declarado no início do código como `"char letra[13]"`, e então converte o valor salvo em um valor numérico, e verifica se ele é igual ao valor recebido, se não for, ele pula uma linha do arquivo de texto e lê a próxima linha até encontrar um valor correspondente entre os valores salvos no cartão de memória.

Desse modo, se o valor é encontrado, o programa lê a linha seguinte para identificar o número do apartamento, se o número for menor que 160, ele irá salvar o valor em um vetor de caracteres com 4 espaços "char comp[4]", posteriormente transformará esses caracteres salvos em um valor numérico com a variável "rec", em que "rec" é responsável por salvar o valor numérico com a linha "rec = atof(comp)", e mostra a seguinte mensagem no LCD "apartamento número", como também mostra o número salvo. Se o número salvo for maior que 160, ele é subtraído 160 dele e mostra-se no LCD "perigo apt". Por fim, mostra-se o novo número salvo, e após mostrar a mensagem, ele zera as variáveis utilizadas e volta ao começo do loop.

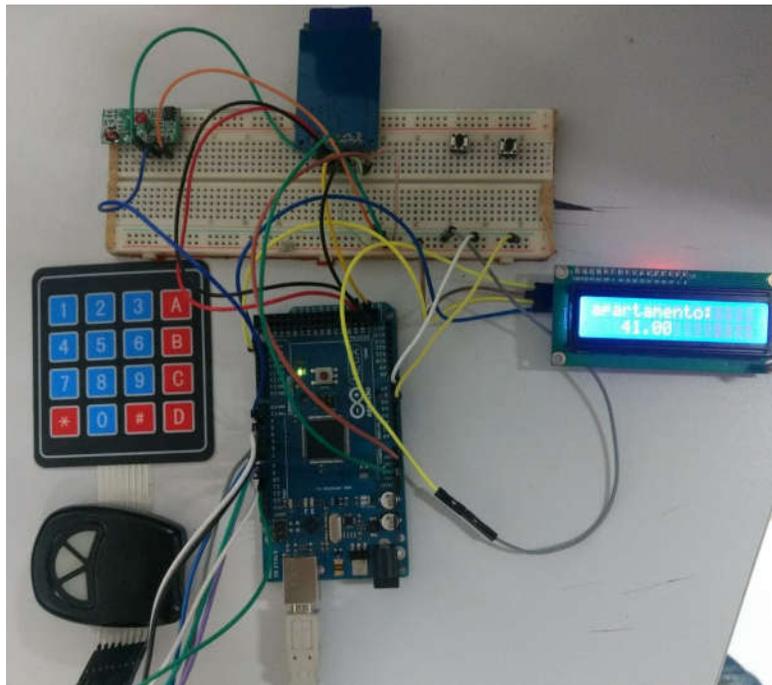
Se o botão for pressionado, ele entra na rotina de cadastro, nessa rotina ele espera o botão ser desapertado, e então entra num laço de repetição em que o usuário digita o número escolhido no teclado matricial, esse número é salvo na variável "unis" quando o botão '#' do teclado matricial é pressionado, e repete esse ciclo mais duas vezes para salvar as dezenas e centenas quando o usuário pressiona o botão select. Posteriormente, ele segue para a rotina de gravação de valores no cartão, espera um valor ser recebido pelo receptor, e salva esse valor na variável "value", logo após verifica se "value>0", se não, ele mostra uma mensagem de erro, se sim, ele abre o arquivo do cartão de memória, salva o valor do apartamento e mostra uma mensagem "código recebido", e espera o usuário apertar o botão de select para habilitar o salvamento do código de perigo daquele apartamento. Quando o botão select é pressionado, o código adiciona 160 ao valor do apartamento e espera receber algo no receptor, e repete o mesmo processo novamente, e aguarda o usuário apertar o select novamente para encerrar o processo, e voltar ao começo do loop.

### 3.9 Montagem do Protótipo

A montagem do dispositivo se dividiu em duas partes, as quais consistem no teste em protoboard, e posteriormente a construção da sua interface. Na construção do circuito em protoboard, utilizamos todos os periféricos de entrada e saída mencionados, simulando seu funcionamento

real, e realizando pequenos ajustes. Esses ajustes foram de suma importância nesse projeto, pois mudamos os botões de cadastro, os quais eram separados, e implementamos suas funções no próprio teclado matricial, deixando a interface do dispositivo amigável e mais estética. A Figura 17 a seguir, mostra o circuito completo montado em protoboard.

Figura 17: Protótipo completo do Projeto DISIMP montado em protoboard.



Fonte: Autoria Própria.

### 3.9.1 Construção da Interface

Para a construção desse estágio do projeto, foram necessários alguns materiais, tais como: uma caixa patola, uma placa PCI universal, conectores empilháveis fêmea, e fios. Foi escolhido uma caixa de plástico da marca patola, pois além de ser um material próprio para construção de dispositivos eletrônicos, traz consigo também mais robustez para o projeto em relação a uma caixa de madeira MDF. A Figura 18 a seguir, mostra a caixa utilizada para a construção do protótipo.

Figura 18: Caixa plástica patola.



**Fonte:** Autoria Própria.

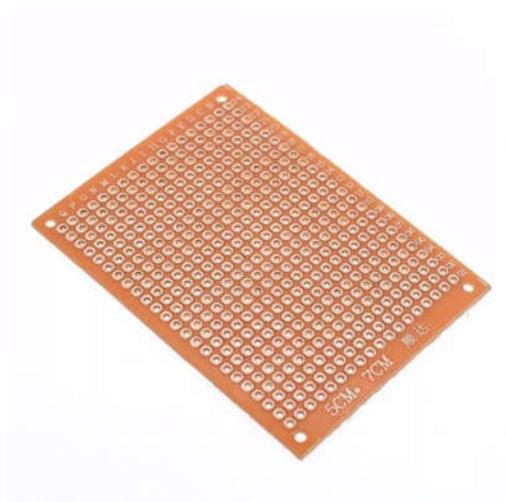
Durante a montagem do equipamento, foi analisado a importância de construir um dispositivo com uma facilidade de manutenção. Dentre desse contexto, utilizamos os conectores empilháveis fêmea, os quais tiveram seus pinos soldados na placa PCI universal. Desse modo, os pinos dos periféricos utilizados ficaram conectados na entrada fêmea dos conectores, mas disponíveis para uma fácil remoção e troca, caso ocorresse algum incidente com o circuito. Na Figura 19 e 20 a seguir são mostrados os conectores utilizados e a placa PCI universal, respectivamente.

Figura 19: Conectores empilháveis fêmea.



**Fonte:** Disponível em: <<https://bit.ly/2S7Z6AZ>>

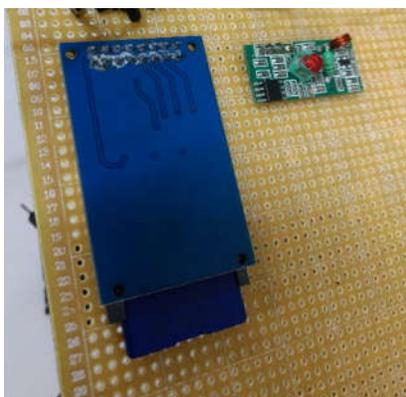
Figura 20: Placa PCI universal.



Fonte: Disponível em: <<https://bit.ly/2Bt0b00>>

Como relatado anteriormente, soldamos os conectores fêmeas direto na placa, no intuito de fixar e obter uma conexão confiável do Shield de memória SD, como também do módulo receptor RF-5v (Figura 21). Foi realizado também uma ilha de solda nos pinos dos conectores, no objetivo de colocar um fio para fazer a conexão com o Arduino.

Figura 21: Periféricos acoplados na Placa PCI universal.



Fonte: Autoria Própria.

Por fim, fizemos três cortes na caixa no objetivo de inserir o display LCD, o teclado matricial, e um corte para a saída da alimentação do Arduino. Por conseguinte, foi colado os periféricos na caixa com fita dupla

face, e estabelecido as conexões necessárias para o funcionamento do protótipo.

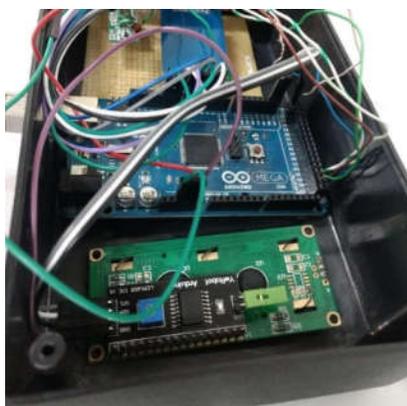
Após uma jornada árdua de dedicação e pesquisa, foi obtido com êxito o resultado esperado, como mostrado na Figura 22 e 23. O equipamento possui um mecanismo de controle de acesso que visa maior segurança na entrada dos usuários a seus condomínios, como também possui um fácil manuseamento das suas funções tanto para o porteiro, quanto para o morador, possuindo uma interface robusta e amigável.

Figura 22: Parte externa do Protótipo final.



Fonte: Aatoria Própria.

Figura 23: Parte interna do Protótipo final.



Fonte: Aatoria Própria.

#### 4. CONSIDERAÇÕES FINAIS

Ao final desse projeto, foi conseguido o resultado esperado ao construir um sistema de fácil manuseamento e acessibilidade para todos os usuários. Nesse sentido, o DISIMP veio como uma ferramenta de controle de acesso com maior eficácia no tocante à segurança, em virtude de todo seu procedimento metodológico, como também do seu botão de perigo.

Outro fator que vale ressaltar, é a possibilidade de aperfeiçoamento do projeto para trabalhos futuros. Desse modo, o projeto pode ser modificado futuramente para um equipamento mais aprimorado, com a implementação do seu circuito através de uma placa de circuito impresso, e a inserção de uma antena para maior alcance. Além disso, também pode ser modificado o controle transmissor, isto é, a construção de um controle transmissor próprio, com suas respectivas funcionalidades.

Porém, é válido citar algumas dificuldades enfrentadas durante a realização do trabalho. Uma das primeiras dificuldades encontradas no início do projeto foi realizar o armazenamento dos dados do usuário, o qual abarca a codificação do controle juntamente com as informações referentes ao número do apartamento e botão perigo. Sendo assim, não foi possível realizar o armazenamento dessas informações, em virtude da memória insuficiente do Arduino Uno que utilizamos inicialmente. Porém, após a inserção do Shield de memória SD foi possível sanar esse problema, como também analisamos a importância dela no tocante à segurança de armazenamento, fato que explica a sua permanência mesmo após a mudança para o Arduino mega2560.

Outro imprevisto enfrentado foi realizar a junção dos códigos de cada periférico, pois no ato da programação foi feito cada um de forma isolada. Desse modo, ao colocar todos no algoritmo principal tivemos que estudar as relações e conexões entre eles, no intuito de chegar ao resultado esperado. Esse fato demandou algum tempo e um significativo grau de dificuldade, porém, após ajuda mútua e pesquisa, tornou-se possível chegar ao êxito nesse quesito.

## 5.REFERÊNCIAS

MENDES, Elzilaine Domingues; VIANA, Terezinha de Camargo. Os condomínios fechados: uma questão de segurança ou um estilo de vida. In: **IN: Anais do II Congresso Sul-Americano de Psicanálise sobre Violência, Culpa e Ato: causas e efeitos subjetivos**. 2009.

MACEDO, Diego. Disponível em: <<http://www.diegomacedo.com.br/mecanismos-de-controle-de-acesso/>>. Acesso em 24 abril de 2018.

McROBERTS, Michael, **Arduino Básico**. Novatec, 2011.

FURTADO, Vasco. **Tecnologia e gestão da informação na segurança pública**. Editora Garamond, 2002.

CAVALCANTE, Luiz Augusto Gonçalves et al. ANÁLISE MULTIVARIADA DA UTILIDADE DA CONSOLIDAÇÃO DE SERVIÇOS ESPECIALIZADOS EM CONDOMÍNIOS LOGÍSTICOS. **Concurso de Monografia CBTU**, v. 5, 2009.

GIACOMIN, João Carlos; VASCONCELOS, Flávio Henrique. Qualidade da medição de intensidade de sinal nas comunicações de uma rede de sensores sem fios: Uma abordagem da camada física. **INFOCOMP Journal of Computer Science**, v. 5, n. 2, p. 83-92, 2006.

TECHTARGET. Disponível em: <<http://whatis.techtarget.com/definition/EEPROM-electrically-erasable-programmable-read-only-memory>>. Acesso em 24 abril de 2018.

CAMARGO, DANIEL. **Arduino: Introdução à Prototipagem Eletrônica**, 2013.

CODÁ, Luiza Maria Romeiro **Introdução à organização de computadores**. Departamento de Engenharia Elétrica e de Computação, 2013.

CASTRO, Cel. **Segurança de condomínios - POLÍCIA MILITAR DO ESTADO DE SP** (2012).

GODOY, José E. **Técnicas de Segurança em Condomínios**, São Paulo, 2ª Edição, SENAC, 2009.

BUCCUCCI, Giuliano. **Sistema de Monitoração e Controle de Acesso para Condomínios Utilizando a Tecnologia de Identificação por Rádio Freqüência (RFID)** (2010).

BEZERRA, Júnior. **RFID em sistemas de segurança em prédios** (2010).

MARCHESAN, Marcelo. **SISTEMA DE MONITORAMENTO RESIDENCIAL UTILIZANDO A PLATAFORMA ARDUINO** (2012)

PEREIRA, Daniela. **Brasileiros investem em segurança residencial**. Tribuna da Bahia. Bahia, 25 fev. 2012.

MORAIS, Filho. **ORIENTAÇÃO À SEGURANÇA EM CONDOMÍNIOS: Normas e procedimentos de segurança** (2016).

PUHLMANN, Henrique Frank W. **Sistemas Operacionais de Tempo Real-Teclados Matriciais**.

DE COMPUTACAO, ENGENHARIA; GOMES, Victor Kaleb Leite. **Desenvolvimento De Um Jogo De Memorizacao Luminosa Na Plataforma Arduino**.

ASADA, A. Electronic Displays: A Revealing Look at the Latest in LCDs. **Display Devices Dempa Publications**. 30, jul. 1990.

BARBACENA, Ilton L.; FLEURY, Claudio Afonso, **Display LCD**. In-Tech, 1996.

Wiring. **Wiring**: Wiring Framework. Disponível em: <<https://github.com/WiringProject/Wiring>>. Acesso em: 10 outubro de 2018.

Processing. **Processing.org**. Disponível em: <<https://processing.org/>>. Acesso em: 10 outubro de 2018.

Violência no Rio Grande do Norte. Disponível em: <<https://g1.globo.com/rn/rio-grande-do-norte/noticia/rn-e-o-estado-com-a-maior-taxa-de-crimes-violentos-de-2017.ghtml>>. Acesso em: 5 de dezembro de 2018.

Roubos e Furtos a Condomínios em São Paulo. Disponível em: <<https://g1.globo.com/sp/sao-paulo/noticia/roubos-e-furtos-a-condominios-crescem-56-no-estado-de-sp-em-2018.ghtml>>. Acesso em: 5 de dezembro de 2018.

HUANG, **Xiongchuan et al**. A 0dBm 10Mbps 2.4 GHz ultra-low power ASK/OOK transmitter with digital pulse-shaping. In: Radio Frequency Integrated Circuits Symposium (RFIC), 2010 IEEE. IEEE, 2010. p. 263-266.