



**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

**Centro de Ciências Exatas e de Tecnologia**

**Departamento Engenharia Elétrica**

**Julia Paschoal Silva**

**Desenvolvimento e construção de um sistema de controle de  
uma fechadura elétrica para residências comandado por um  
celular**

**2021**

**São Carlos – SP**

**Julia Paschoal Silva**

**Desenvolvimento e construção de um sistema de controle de uma fechadura elétrica para residências comandado por um celular**

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia Elétrica da Universidade Federal de São Carlos como requisito para obtenção do título de Engenheira Eletricista.

Orientador: Prof. Dr. Robson Barcellos

**2021**  
**São Carlos – SP**



**Julia Paschoal Silva**

**Desenvolvimento e construção de um sistema de controle de uma fechadura elétrica para residências comandado por um celular**

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia Elétrica da Universidade Federal de São Carlos como requisito para obtenção do título de Engenheira Eletricista.

Aprovada em 26 de outubro de 2021

**BANCA EXAMINADORA**

---

Prof. Dr. Robson Barcellos – Orientador  
Universidade Federal de São Carlos

---

Prof. Dr. Celso Aparecido de França  
Universidade Federal de São Carlos

---

Prof. Dr. Samuel Lourenço Nogueira  
Universidade Federal de São Carlos

*À minha família,  
aos meus amigos,  
e aos professores.*

## **AGRADECIMENTOS**

Este trabalho de conclusão de curso é destinado à todas as pessoas que me ajudaram nessa jornada durante a graduação.

Primeiramente aos meus pais por terem me dado suporte e a oportunidade de frequentar uma universidade pública, especialmente um curso integral e complexo como a Engenharia Elétrica. Minha mãe Claudia me forneceu todos os cuidados possíveis e meu pai Rodnei me ajudou como podia, sempre disponível para o que fosse preciso.

Depois agradeço ao meu namorado/marido Gustavo que esteve presente comigo desde o início nos momentos bons e ruins. Sua presença foi de extrema importância para dar continuidade nos estudos, especialmente porque ele já havia passado por isso e não me permitiu desistir.

Agradeço também aos amigos e colegas que conquistei durante o curso: Débora, Gabriela, Raphael, Aline, Leonardo, Gustavo, Guilherme entre outros, que contribuíram cada um da sua maneira para passarmos juntos essa jornada. Foram horas de estudo compartilhando conhecimento. Ser aprovada nas disciplinas sem eles seria muito mais difícil.

Por fim, agradeço aos professores que administraram as disciplinas e ofereceram apoio para dúvidas durante o curso. Sem o conhecimento e disposição deles hoje não me tornaria a profissional e pessoa que me tornei.

Muito obrigado.

Esta monografia é dedicada a todos vocês!



*“A educação é a arma mais poderosa que você pode usar para mudar o mundo.”*

*Nelson Mandela*

## RESUMO

O presente trabalho se constitui no desenvolvimento e construção de um protótipo de uma fechadura elétrica para residências, controlada remotamente pelo celular utilizando um módulo ESP8266 NodeMcu, que é uma placa com conexão Wi-Fi. Um dos requisitos é que a fechadura elétrica fique instalada no lado de dentro da porta principal da residência, de modo que não seja possível visualizá-la externamente. O sistema contém um relé para acionamento da fechadura elétrica e um aplicativo para celular Android com botões para comando de abrir/fechar a fechadura. O módulo ESP8266 NodeMcu realiza a conexão via Wi-Fi com o aplicativo do celular e aciona o relé que aciona a fechadura. Foram instalados sensores para verificar se a fechadura se encontra aberta ou fechada e notificar o usuário.

Palavras-chave: Fechadura Elétrica via Wi-Fi, Acionamento via Celular, Wi-Fi, ESP8266, NodeMcu, Automação Residencial, Aplicativo Android para controle residencial.

## **ABSTRACT**

The present work consists in the development and construction of a prototype of an electric lock for homes, remotely controlled by cell phone using the ESP8266 NodeMcu module, a card with a Wi-Fi connection. One of the requirements is that the electric lock to be installed inside the main door of the residence, so that it is not possible to view it externally. The system contains a relay to activate the electric lock and an Android cell phone application with buttons for command to open/close the lock. The ESP8266 NodeMcu module connects via Wi-Fi to the cellphone and activates the relay through a port. Finally, sensors were installed to check whether the lock is open or closed to notify the user.

Keywords: Electric Lock using Wi-Fi, Cellular Activation, Wi-Fi, Home Automation, Android Application, ESP8266, NodeMcu.

## Lista de figuras

Figura 1 - Diagrama esquemático da solução proposta	15
Figura 2 - Etapas de desenvolvimento do sistema	18
Figura 3 - Requisitos do sistema	19
Figura 4 – NodeMcu ESP8266	21
Figura 5 - Pinagem NodeMcu ESP8266	22
Figura 6 – Módulo ESP8266	23
Figura 7 – Módulo relé 5V	23
Figura 8 - Mini fechadura elétrica	24
Figura 9 - Chave de fim de curso	25
Figura 10 - Fonte de alimentação chaveada	26
Figura 11 - Conector jack p4	26
Figura 12 - Regulador de tensão 7805	27
Figura 13 - Menu geral	28
Figura 14 - Designer	28
Figura 15 - Blocos	29
Figura 16- Diagrama geral	30
Figura 17 - Esquema elétrico	31
Figura 18 - Fluxograma do programa	35
Figura 19 - Tela inicial do aplicativo na simulação	36
Figura 20 - Tela principal do aplicativo em uma simulação	37
Figura 21 - Lógica do aplicativo	38
Figura 22 - Aparência do aplicativo na área de trabalho	39
Figura 23 - Tela inicial	40
Figura 24 - Resposta ao apertar o botão “abrir”	41
Figura 25 - Resposta ao apertar o botão “fechar”	42
Figura 26 - Notificação	43
Figura 27 - Mensagem de erro	44
Figura 28 - Cálculo tempo médio de resposta	46

## **Lista de tabelas**

Tabela 1 - Lista de materiais utilizados no projeto _____	20
Tabela 2 – Especificações do NodeMcu _____	21
Tabela 3 - Relação de portas e componentes _____	31
Tabela 4 - Requisição web _____	33
Tabela 5 - Tempo de resposta médio _____	46

# Sumário

1. Introdução	14
1.1. Objetivos	15
2. Automação residencial ou domótica	16
3. Trabalhos relacionados	17
4. Materiais e métodos	18
4.1. Etapas do projeto	18
4.2. Escopo do projeto	18
4.2.1. Requisitos do sistema	18
4.2.2. Materiais utilizados	19
4.2.3. NodeMcu ESP8266	20
4.2.4. Relé	23
4.2.5. Fechadura elétrica	24
4.2.6. Sensor de fim de curso	24
4.2.7. Fonte de alimentação	25
4.2.8. Regulador de tensão 7805	26
4.2.9. App inventor	27
5. Desenvolvimento	29
5.1. Construção do circuito	31
5.2. Programação do NodeMcu	32
5.2.1. Declarações	32
5.2.2. Configurações	33
5.2.3. Execução	33
5.3. Programação no app inventor	35
5.3.1. Designer	35
5.3.2. Blocos	37
6. Resultados e discussões	38
6.1. Falhas	43
6.1.1. Falta de conexão	43
6.1.2. Falha por tempo de resposta	44
6.1.3. Falta de energia	45
6.2. Tempo de resposta	45
6.3. Dificuldades	47
6.3.1. Hardware	47
6.3.2. Programação	48
6.4. Pontos positivos	48

7. Conclusão	50
8. Trabalhos futuros	51
8.1.1. Autenticação	51
8.1.2. Aplicativo nativo	51
8.1.3. Registro	51
8.1.4. Rede Wi-Fi	51
8.1.5. Sensor magnético na porta	51
8.1.6. Servidor Externo	51
8.1.7. Conexão IoT	52
9. Referências	53
ANEXO A – Código do NodeMcu	56
ANEXO B – Lógica em blocos do aplicativo	61
ANEXO C – Parâmetro do ESP8266	63

## 1. Introdução

De acordo com os dados estatísticos anuais do estado de São Paulo (Portal do Governo, 2020) os casos de roubo e furto passaram dos 800 mil em 2020. Em 2019 e 2018, passaram de 1 milhão de casos. Isto mostra o constante aumento da criminalidade e falta de segurança para as pessoas, seja na rua ou em suas residências. E uma das formas que está sendo explorada cada vez mais para aumentar a segurança das pessoas é o uso da tecnologia.

O avanço da tecnologia trouxe um movimento que afeta não só a indústria, mas também o setor público. É possível ver a mudança da sociedade que passa por uma inserção acelerada da tecnologia, inclusive em sua estrutura. O governo e as empresas privadas já se movimentam para adotar soluções inovadoras que beneficiem a população. O que está sendo testemunhado é a ascensão das *smart cities*, ou cidades inteligentes (Tellus, 2020).

Particularmente, equipamentos tecnológicos para residências é um fator de grande interesse para promover principalmente o bem-estar social, conforto, segurança e qualidade de vida para a população, solução essa definida como Domótica ou também conhecida como automação residencial (DOMINGUES, 2013)

Domótica é uma tecnologia responsável pela gestão de todos os recursos habitacionais. Este termo nasceu da fusão da palavra “Domus”, que significa casa, com a palavra “Robótica”, que está ligada ao ato de automatizar, isto é, realizar ações de forma automática. Este conceito de automação tem o objetivo de encontrar soluções que deem resposta à necessidade da sociedade de querer realizar o mínimo esforço nas atividades diárias e rotineiras. Assim, a domótica, além de introduzir conforto e melhoria de vida aos seus utilizadores, introduz ainda novos conceitos, tais como a comunicação e segurança. (SISLITE, 2020)

A automação residencial ainda é muito nova. Os primeiros trabalhos datam do final da década de 1970, quando surgiram nos Estados Unidos os primeiros módulos “inteligentes”, cujos comandos eram enviados pela própria rede elétrica da residência. Tratava-se de soluções simples, não integradas e que resolviam situações pontuais, como ligar remotamente algum equipamento ou luzes. (MURATORI e DAL BÓ, 2011).

Hoje têm-se sistemas completos de automação residencial, que facilitam a rotina das pessoas. Este trabalho propõe trazer mais uma solução para os âmbitos da automação residencial e segurança.

Têm-se como fonte de motivação para a realização deste trabalho, o fato de que pessoas próximas passaram por furtos em suas residências. Tendo como consequência a invasão da privacidade e risco de vida dos mesmos.

Por isso, a proposta é ter uma fechadura instalada na porta, dentro da residência, de forma que reforce a segurança contra furto/roubo e que esteja invisível aos olhos de quem esteja fora da residência. Assim, não há como saber da existência da fechadura e nem de como abri-la.

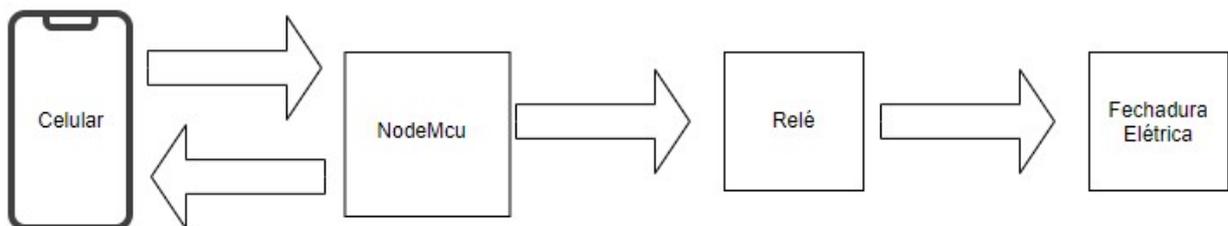
### 1.1. Objetivos

Este trabalho tem como objetivo desenvolver um protótipo de uma fechadura elétrica que será controlada remotamente utilizando um celular. Para atingir este objetivo, os seguintes passos foram realizados:

- Desenvolvimento de uma aplicação que chama um servidor local e privado, e que realiza a comunicação com o celular via Wi-Fi. Para isso, utilizou-se o módulo ESP8266 NodeMcu.
- Construção de uma estação local que contém a fechadura elétrica, o módulo ESP8266, um interruptor eletromecânico (um relé), e uma fonte de alimentação.
- Desenvolvimento de um aplicativo para celular com sistema operacional Android.

A Figura 1 apresenta um diagrama esquemático da solução proposta.

Figura 1 - Diagrama esquemático da solução proposta



Fonte: Autoria Própria.

## 2. Automação residencial ou domótica

Segundo Jorge Roberto e Paulo Henrique (2011, pág. 70):

Automação Residencial ou Domótica é um conjunto de serviços proporcionados por sistemas tecnológicos integrados como o melhor meio de satisfazer as necessidades básicas de segurança, comunicação, gestão energética e conforto de uma habitação.

O objetivo principal da automação residencial, e que define uma instalação nas residências, é a integração entre os sistemas presentes que podem ser controlados, aliados à capacidade de executar funções e comandos mediante instruções programáveis que foram previamente definidas. Essa integração pretende englobar todos os sistemas tecnológicos da residência, como:

- Instalação Elétrica: que engloba o controle da iluminação, a gestão da energia, dados de uso rotineiro, e otimização deste uso baseado nos dados arrecadados.
- Sistemas de segurança: alarmes contra invasão, fechaduras inteligentes, câmeras de segurança, controle de acesso e monitoramento.
- Sistemas multimídia: controle dos equipamentos eletrônicos da casa como som, televisão, videogames e aparelho de vídeo e condicionadores de ar.
- Sistemas de comunicação: como telefones celulares, interfone, internet e serviços de assinatura.
- Utilidades gerais: aquecimento de água, bombas d'água, irrigação de plantas e outros serviços.

### 3. Trabalhos relacionados

A preocupação em aprimorar a segurança das pessoas e desenvolver sistemas inteligentes baseados em IoT é comum. Existem diversos trabalhos já realizados com o conceito de fechadura inteligente, que buscam prover não só segurança, mas conforto para as pessoas.

Grosmann (2018) propôs um sistema de segurança pensando nos bens pessoais, como carteira, documentos, entre outros que podem ser guardados em uma gaveta qualquer. Neste trabalho, ele desenvolveu uma aplicação de automação e segurança residencial que serve como fechadura para gavetas e armários de pequeno porte e integrou conceitos de IoT, como: IPv6, 6LowPan, MQTT e banco de dados na nuvem. Dessa forma, obteve-se uma fechadura que trava e destrava por aplicativo mobile via Bluetooth ou NFC (comunicação por campo de proximidade), possibilidade de cadastro de fechaduras e compartilhamento de chaves entre os usuários que utilizam o aplicativo. Por fim, todos os dados são armazenados.

Gusmão, Castro e Zampolo (2016) propôs um sistema de fechadura inteligente que identifica o usuário que utiliza o sistema através de reconhecimento facial. Ele se baseia em OpenCV e *eigenfaces*, e o processamento é realizado em dispositivos móveis que portem o sistema operacional Android, cujo comando de ativação é enviado por Bluetooth. Esse comando só é realizado se na aplicação para Android que for feita a captura da imagem do rosto do usuário, for identificado. A aplicação irá se comunicar com o circuito da fechadura que resultará ou não na autorização ao portador do aparelho.

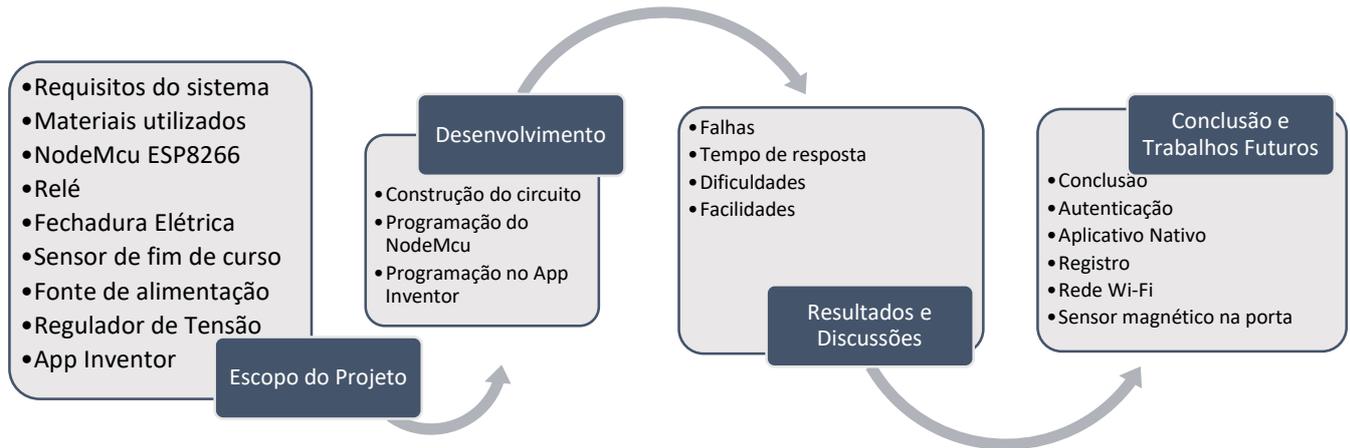
Pereira, Dias e Queiroz (2019) propuseram um sistema de monitoramento e controle de acesso de múltiplos ambientes, onde a arquitetura é baseada na computação em névoa, e os dados podem ser processados na borda da rede em vez de enviá-los para a nuvem, proporcionando uma resposta rápida e de qualidade. Para controlar e monitorar os ambientes com mais eficácia e segurança, foi projetado uma trava inteligente usando um microcontrolador ESP8266. Ela possui um módulo RFID para leitura de cartões de acesso, que de acordo com o código de acesso desse cartão, irá ou não conceder acesso à sala.

## 4. Materiais e métodos

### 4.1. Etapas do projeto

No diagrama da Figura 2 são apresentadas as etapas de desenvolvimento do sistema proposto. Cada um dos itens mencionados, serão explicados a seguir.

Figura 2 - Etapas de desenvolvimento do sistema



Fonte: Elaborado pela autora

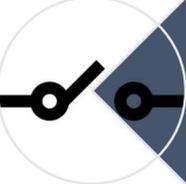
### 4.2. Escopo do projeto

#### 4.2.1. Requisitos do sistema

Para o desenvolvimento de um software ou um sistema é necessário definir os requisitos, que são as descrições dos serviços propostos pelo sistema e suas restrições operacionais (SOMMERVILLE, 2011). Os requisitos definem como o sistema deve se comportar, o que deve fazer, e quais os recursos providos.

Este trabalho têm-se os requisitos mostrados na Figura 3.

Figura 3 - Requisitos do sistema

-  A instalação da fechadura deve ser dentro da residência, de forma que quem esteja fora não saiba que existe uma fechadura ali.
-  A estação local e o celular precisam estar conectados na mesma rede Wi-Fi local.
-  A fechadura elétrica deve ser acionada por um relé.
-  O aplicativo deve notificar ao usuário do estado da fechadura.
-  O sistema deve notificar quando não há conexão à rede Wi-Fi e apresentar uma solução.
-  O aplicativo deve notificar quando a fechadura tiver problemas de acionamento.

Fonte: Elaborado pela autora

#### 4.2.2. Materiais utilizados

Para desenvolvimento do sistema de segurança foram utilizados os materiais

presentes na Tabela 1.

Tabela 1 - Lista de materiais utilizados no projeto

<b>Item</b>
<b>NodeMcu ESP8266</b>
<b>Celular</b>
<b>Relé</b>
<b>Fechadura elétrica</b>
<b>Sensor de fim de curso</b>
<b>Fonte de alimentação</b>
<b>Regulador de tensão L7805</b>
<b>Componentes eletrônicos: (Leds, Conectores, Capacitores, Resistores)</b>

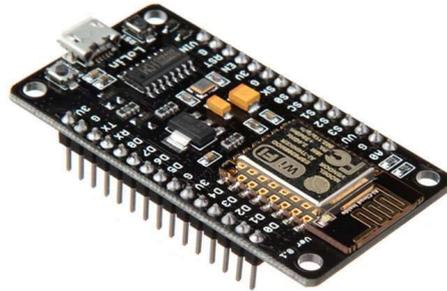
Fonte: Elaborada pela Autora.

Também foram utilizados os softwares Arduino IDE e App Inventor, para programação do NodeMcu e do aplicativo para celular, respectivamente.

#### **4.2.3. NodeMcu ESP8266**

O NodeMcu como o da Figura 4, é uma placa de desenvolvimento que combina o chip ESP8266, uma interface usb-serial e um regulador de tensão 3.3V. A programação do dispositivo pode ser feita com a IDE do Arduino, utilizando a comunicação via cabo micro-usb. Ele possui uma antena embutida e conector micro-usb para conexão ao computador, além de 11 pinos de I/O e conversor analógico/digital.

Figura 4 – NodeMcu ESP8266



Fonte: (Bau da Eletrônica Componentes Eletrônicos)

Ele é um componente completo para projetos pequenos que necessitem de conexão com a internet, inclusive por Wi-Fi. Algumas das especificações do NodeMcu se encontram na Tabela 2 e no Anexo C:

Tabela 2 – Especificações do NodeMcu

<b>Especificações</b>
<b>(datasheet)</b>
<b>Wireless padrão 802.11 b/g/n</b>
<b>Alcance de frequência: 2.4GHz – 2.5GHz</b>
<b>Tensão de operação: 3.0~3.6V</b>
<b>Corrente de operação: 80mA</b>
<b>Alcance de temperatura de operação: -40°~125°</b>
<b>Número de GPIOs: 11</b>
<b>Barramento periférico: UART/HSPI/12C/12S/IR Remote Control ou GPIO/PWM</b>
<b>Tamanho: 16mmx24mmx3mm</b>
<b>Modo Wi-Fi: station/softAP/SoftAP+station</b>

Fonte: (MORAIS, 2017)

Apesar da pinagem reduzida quando comparado a outros Arduinos, existem no mercado outros circuitos integrados que podem ser utilizados para expansão de entradas



Figura 6 – Módulo ESP8266



Fonte: (MORAIS, 2017)

Para este projeto, o NodeMcu foi utilizado como a placa de desenvolvimento de todo o sistema. Ele recebe dados do celular e sensores, processa o que será feito a seguir, envia os comandos novamente para o celular, via conexão Wi-Fi, e aciona o relé.

#### 4.2.4. Relé

O relé que foi utilizado para este projeto (Figura 7), é um módulo de 5V com 1 canal, já adaptado de forma que o circuito a ser alimentado fica completamente isolado do circuito do microcontrolador. Ele pode acionar cargas de até 250 VAC ou 30 VDC, suportando uma corrente máxima de 10A. Possui LED indicador de energia, 2 pinos de energia e 1 de controle, além de borne de saída com parafusos.

Figura 7 – Módulo relé 5V



Fonte: (Bau da Eletrônica Componentes Eletrônicos)

Neste projeto, o relé foi utilizado como o componente que abre ou fecha a fechadura

elétrica, de acordo com o sinal recebido do NodeMcu. Além de oferecer o controle e a facilidade para instalação, ele também suporta a tensão da carga acionada.

#### 4.2.5. Fechadura elétrica

É muito comum ver fechaduras elétricas em portões ou portas principais contendo interfone. Elas possuem um mecanismo que oferece a ligação entre esses dois componentes de forma que a pessoa possa abrir o portão ou a porta utilizando apenas o interfone.

Por questões de custo do protótipo, foi utilizado um modelo mini de fechadura elétrica como o da Figura 8. Mas a melhor opção seria utilizar as fechaduras elétricas como aquelas utilizadas em portões e portas principais. E assim garantir a segurança proposta neste trabalho.

Figura 8 - Mini fechadura elétrica



Fonte: (FILIPEFLOP)

#### 4.2.6. Sensor de fim de curso

Um sensor de fim de curso ou chave de fim de curso (Figura 9), é um dispositivo eletromecânico que consegue determinar que a estrutura ligada a seu eixo, chegou ao fim do seu campo de acionamento. Ele consegue ser atuado por uma pequena força externa, tem baixo custo de aquisição e tem diversos tamanhos e modelos.

É composto por uma caixa, um contato e um atuador. A caixa é uma estrutura que

pode ser metálica ou plástica e abriga o contato e o atuador, deixando o conjunto unido. O contato pode ser NF (normalmente fechada) ou NA (normalmente aberta) e será usado dentro do circuito como um indicador. Por fim, o atuador fica anexado na caixa e que recebe a força externa.

Portanto, quando o atuador recebe alguma ação de força externa, este abre ou fecha o contato, energizando ou não o circuito.

Figura 9 - Chave de fim de curso



Fonte: (FILIPEFLOP)

Neste trabalho utilizou-se o sensor mostrado na Figura 9, com tensão máxima de 250VAC e corrente máxima de 5A.

O sensor foi instalado junto a fechadura elétrica para identificar quando ela está aberta ou fechada. Por exemplo: quando a fechadura está fechada, ela faz uma força no atuador do sensor, de forma que ele fique acionado e envia ao NodeMcu um sinal.

O NodeMcu interpreta o sinal, envia ao aplicativo uma resposta e o aplicativo mostra uma mensagem no celular informando ao usuário sobre o estado da fechadura elétrica. O mesmo processo ocorre quando a fechadura está aberta.

#### **4.2.7. Fonte de alimentação**

Para este projeto, utilizou se uma fonte chaveada (Figura 10), com uma tensão fixa de 12V na saída, e corrente de 1A.

Figura 10 - Fonte de alimentação chaveada



Fonte: (FILIPEFLOP)

A saída da fonte de alimentação é um conector plug P4 o que dificulta a conexão dela com o circuito. Por isso, utilizou-se um conector Jack P4 (Figura 11) para fazer a ligação da fonte com o circuito.

Figura 11 - Conector jack p4



Fonte: (FILIPEFLOP)

#### 4.2.8. Regulador de tensão 7805

Reguladores de tensão (Figura 12) são muito utilizados em circuitos que precisam garantir uma tensão de alimentação fixa em algum ponto do circuito.

A série 78xx, fornece tensões de saída de 5 a 24V, a partir de uma tensão de entrada de até 35V e com 1A de saída.

Figura 12 - Regulador de tensão 7805



Fonte: (FILIPEFLOP)

O NodeMcu necessita de uma tensão de alimentação de 5V, e a fechadura elétrica de uma tensão de alimentação de 12V. A tensão de alimentação do NodeMcu é então obtida através do regulador de tensão L7805, a partir da tensão de 12V.

#### 4.2.9. App inventor

Para fazer o aplicativo sem que fosse necessário conhecer e saber programar em uma linguagem para Android, utilizou-se o App Inventor.

O App Inventor é um aplicativo desenvolvido pelo MIT (Massachusetts Institute of Technology) Media Lab, com uma aplicação de código aberto originalmente criado pela Google, e atualmente mantida pelo MIT. Ele permite desenvolver aplicativos para equipamentos com o sistema operacional Android, usando apenas um navegador na web.

É possível acompanhar o trabalho em desenvolvimento no site através de um telefone ou um emulador. Os servidores do App Inventor armazenam o trabalho pelo navegador Web, utilizando um e-mail. Então, para mexer no projeto tudo que se precisa fazer é logar por este e-mail.

No menu geral da Figura 13, têm-se algumas funções como a de conectar, que foi utilizado para simular todo o trabalho construído. E na fase final, utilizou – se a função compilar para gerar um arquivo .apk que pode ser instalado em dispositivos Android.

Figura 13 - Menu geral



Fonte: (LAB, 2012)

No menu principal, é possível selecionar os módulos: designer e blocos.

Figura 14 - Designer



Fonte: (LAB, 2012)

Todo aplicativo precisa de uma interface que o usuário possa entender e mexer, e isso foi feito no designer.

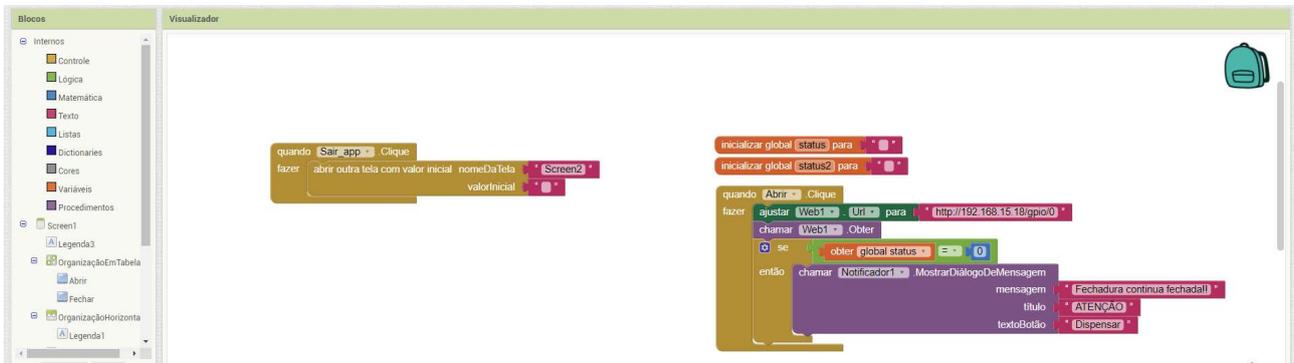
Dentro de designer (Figura 14), existem algumas funções como: paleta, visualizador, componentes, propriedades e mídia. Todas essas funções foram utilizadas para construir a parte visual do aplicativo.

Para que o aplicativo tenha ações, como abrir ou fechar a fechadura, precisou-se programar essa interface e isso foi feito em blocos. É possível fazer a programação em blocos (Figura 15) dos itens inseridos no designer, informando ao sistema o que ocorrerá quando o usuário utilizar qualquer um dos itens na tela inicial.

Por exemplo, neste projeto foi programado que ao apertar o botão “Abrir” ele irá enviar uma resposta ao NodeMcu para que ele abra a fechadura, e mostre uma mensagem

na tela ao usuário indicando que a fechadura está aberta.

Figura 15 - Blocos



Fonte: (LAB, 2012)

Não foi necessário conhecer as linguagens de programação para Android, apenas utilizar a lógica para informar o que precisa ser feito. E os conceitos são transmitidos de forma visual e menos complexa do que um código em uma linguagem de programação como Python ou Java, entre outras.

## 5. Desenvolvimento

O projeto aqui desenvolvido deve funcionar conforme o diagrama geral da Figura 16.

Figura 16- Diagrama geral



Fonte: Elaborada pela autora.

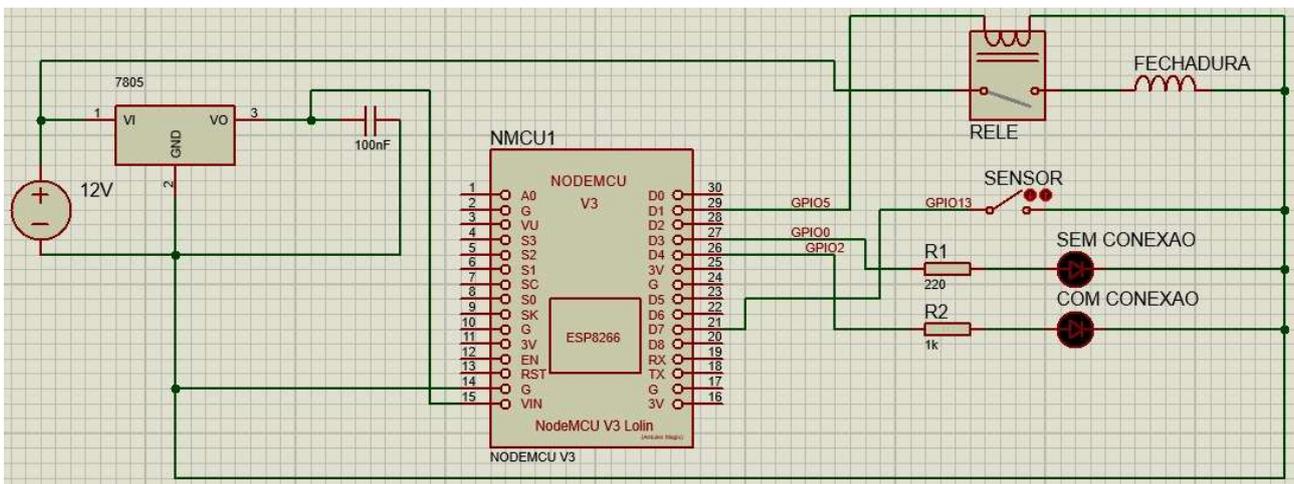
O desenvolvimento do projeto deu-se em três etapas: construção da estação local, programação do NodeMcu e programação no app inventor. Cada uma dessas etapas serão explicadas a seguir.

### 5.1. Construção do circuito

A fim de realizar testes e ir construindo o circuito conforme as necessidades surgiam, ele foi montado primeiro em uma protoboard. Posteriormente, os componentes foram soldados em uma placa de circuito impresso para garantir segurança no funcionamento.

O esquema elétrico do circuito e a relação de portas e componentes utilizados no NodeMcu, se encontram na Figura 17 e na Tabela 3, respectivamente.

Figura 17 - Esquema elétrico



Fonte: Elaborada pela autora.

Tabela 3 - Relação de portas e componentes

Porta	Componente
GPIO5	Relé
GPIO0	Led vermelho (sem conexão à internet)
GPIO2	Led verde (com conexão à internet)
GPIO13	Chave de Fim de Curso

Fonte: Elaborada pela autora.

## 5.2. Programação do NodeMcu

### 5.2.1. Declarações

Para programar o NodeMcu, utilizou-se a plataforma Arduino IDE. O código escrito encontra-se no Anexo A. A programação do NodeMcu foi feita baseada na sua comunicação com o celular.

Toda e qualquer informação é enviada via endereço web e por isso ambos os dispositivos precisam estar conectados à internet e na mesma rede.

Posteriormente, foram configuradas as bibliotecas necessárias: WiFiClient, ESP8266WiFi, ESP8266WebServer, DNSServer e WiFiManager. Em cada uma delas, foram utilizadas as seguintes funções

- **WiFiClient:** Cria uma variável “client” que se conecta ao endereço IP especificado. Clients (ou Clientes) são os dispositivos que irão se conectar ao ESP8266. Neste trabalho o “cliente” é o celular.
- **ESP8266WiFi:** fornece funções e propriedades para configurar e operar o módulo ESP8266 no modo de estação e/ou ponto de acesso. No modo estação, o módulo se conecta a uma rede Wi-Fi, e no modo ponto de acesso é criada uma rede Wi-Fi pelo ESP8266.
- **ESP8266WebServer:** Configura um servidor web no ESP8266, que atende um cliente por vez e sabe como lidar com solicitações HTTP como GET e POST.
- **DNSServer:** Implementa um servidor DNS simples que pode ser usado nos modos estação ou ponto de acesso. Um DNS (Domain Name System) é um sistema que traduz os nomes de domínios inseridos em um navegador, para os endereços IP necessários para acessar esses sites.
- **WiFiManager:** Esta biblioteca permite gerenciar a conexão Wi-Fi. A partir dela é possível configurar ponto de acesso e/ou modo estação.

Primeiro foram definidas as variáveis que serão utilizadas neste projeto e posteriormente definiu-se um IP fixo e criou-se o servidor.

### 5.2.2. Configurações

Configurou-se a primeira função do programa que é executada apenas quando o circuito é ligado. A função serve para configurar os pinos da placa e estabelecer a comunicação com os dispositivos (NodeMcu e App Inventor).

Foram então configuradas as entrada e saída. Iniciou-se a conexão Wi-Fi, e por fim, montou-se uma lógica para verificar se há conexão com a internet e informar ao usuário através de leds.

Quando não houver conexão com a internet o led vermelho estará aceso e o led verde estará apagado. O programa roda a função “WiFiManager”, que gera um ponto de acesso onde o usuário consegue se conectar à internet com qualquer dispositivo que possibilite essa conexão via Wi-Fi.

Quando o NodeMcu se conecta com este ponto de acesso, a função gera uma página na web e então tudo o que o usuário precisa fazer é: escolher a opção “Configure WiFi”, escolher a rede Wi-Fi da residência, inserir login e senha, e conectar o dispositivo (NodeMcu + relé + fechadura) na rede Wi-Fi escolhida. Dessa forma o NodeMcu se conecta a internet da residência e ainda grava esses dados para uso futuro.

O próximo passo foi iniciar o servidor, para realizar a comunicação do NodeMcu com o celular.

### 5.2.3. Execução

Após concluída a configuração inicial, é feita a programação da segunda etapa do programa.

Primeiro, o NodeMcu fica em modo de aguardo até que o aplicativo se conecte com ele. Quando conectados, é enviado um comando através de uma URL, que será interpretado pelo NodeMcu de acordo com a Tabela 4:

Tabela 4 - Requisição web

URL	Significado
<a href="http://192.168.15.18/gpio/0">http://192.168.15.18/gpio/0</a>	abrir fechadura
<a href="http://192.168.15.18/gpio/1">http://192.168.15.18/gpio/1</a>	fechar fechadura

Fonte: Elaborada pela autora.

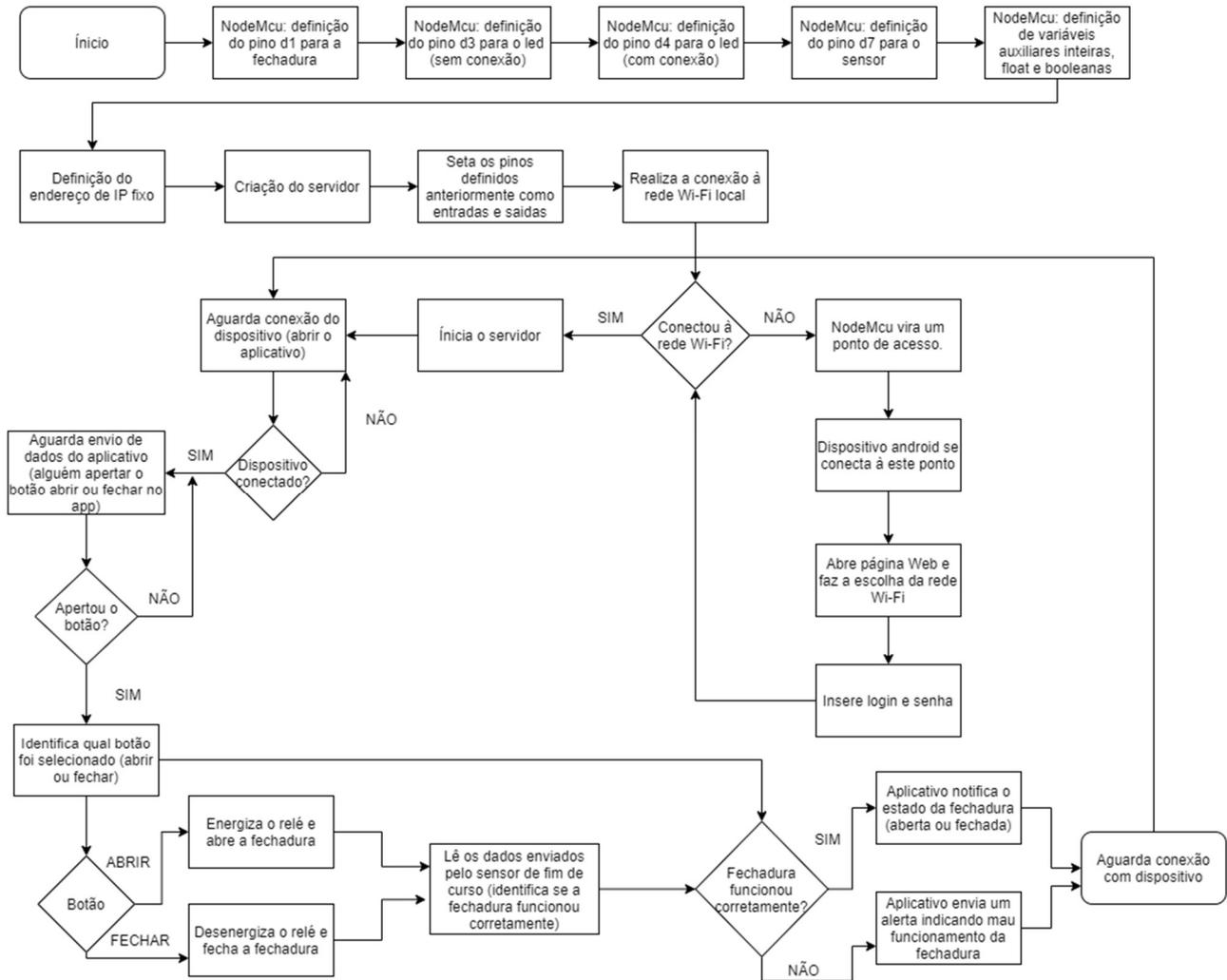
Ao receber esta URL o NodeMcu irá interpretar e enviar uma resposta ao aplicativo e um comando ao relé, que deve abrir ou fechar a fechadura elétrica. No meio desse processo, é realizada a leitura do sensor magnético.

O programa interpreta o valor lido no sensor, que pode ser fechadura aberta ou fechada, e verifica se o estado da fechadura está de acordo com a solicitação do usuário. Se corresponder, significa que a fechadura está respondendo corretamente aos comandos do aplicativo. Caso contrário, é enviado uma resposta para o aplicativo, informando o usuário do mau funcionamento

Observe que neste programa só é possível conectar um dispositivo por vez, o que evita que múltiplas pessoas tentem abrir a fechadura ao mesmo tempo, contribuindo para a segurança da residência.

O fluxograma do programa, encontra-se na Figura 18.

Figura 18 - Fluxograma do programa



Fonte: Elaborada pela autora.

### 5.3. Programação no app inventor

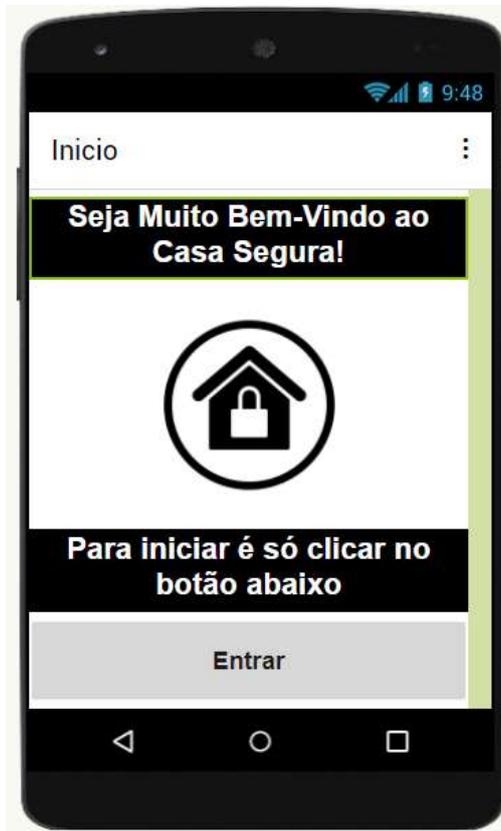
A programação no App Inventor foi realizada em duas etapas e cada uma delas é apresentada nas próximas seções.

#### 5.3.1. Designer

Como já detalhado na seção 4.2.9, o módulo designer programa a parte visual do aplicativo. Tudo que for inserido neste módulo será visualizado pelo usuário que está utilizando o app. Nessa fase, foram construídas duas telas: uma inicial e uma principal.

A tela Inicial (Figura 19) contém um texto de boas-vindas, instruções de como acessar o app e um botão para entrar.

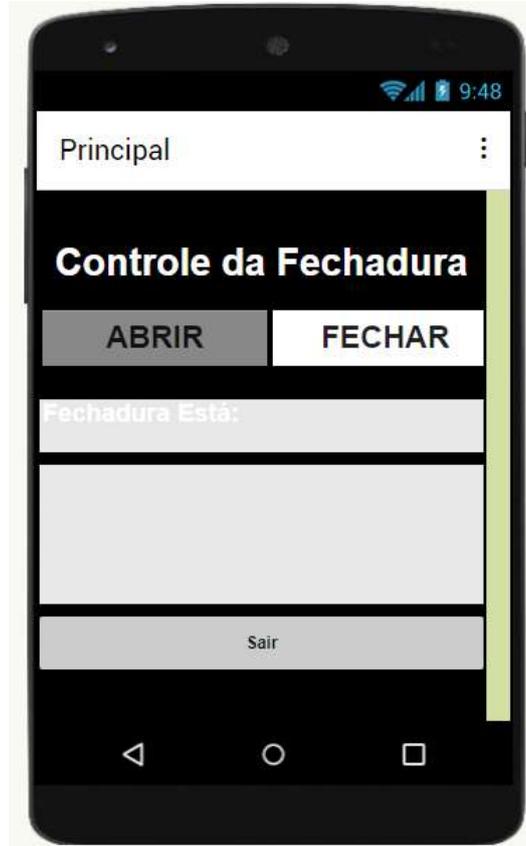
Figura 19 - Tela inicial do aplicativo na simulação



Fonte: Elaborada pela autora.

Na tela principal (Figura 20) os comandos “Abrir”, “Fechar” e “Sair do Aplicativo” foram construídos com botões, e há um espaço para imprimir a mensagem do status da fechadura.

Figura 20 - Tela principal do aplicativo em uma simulação



Fonte: Elaborada pela autora.

### 5.3.2. Blocos

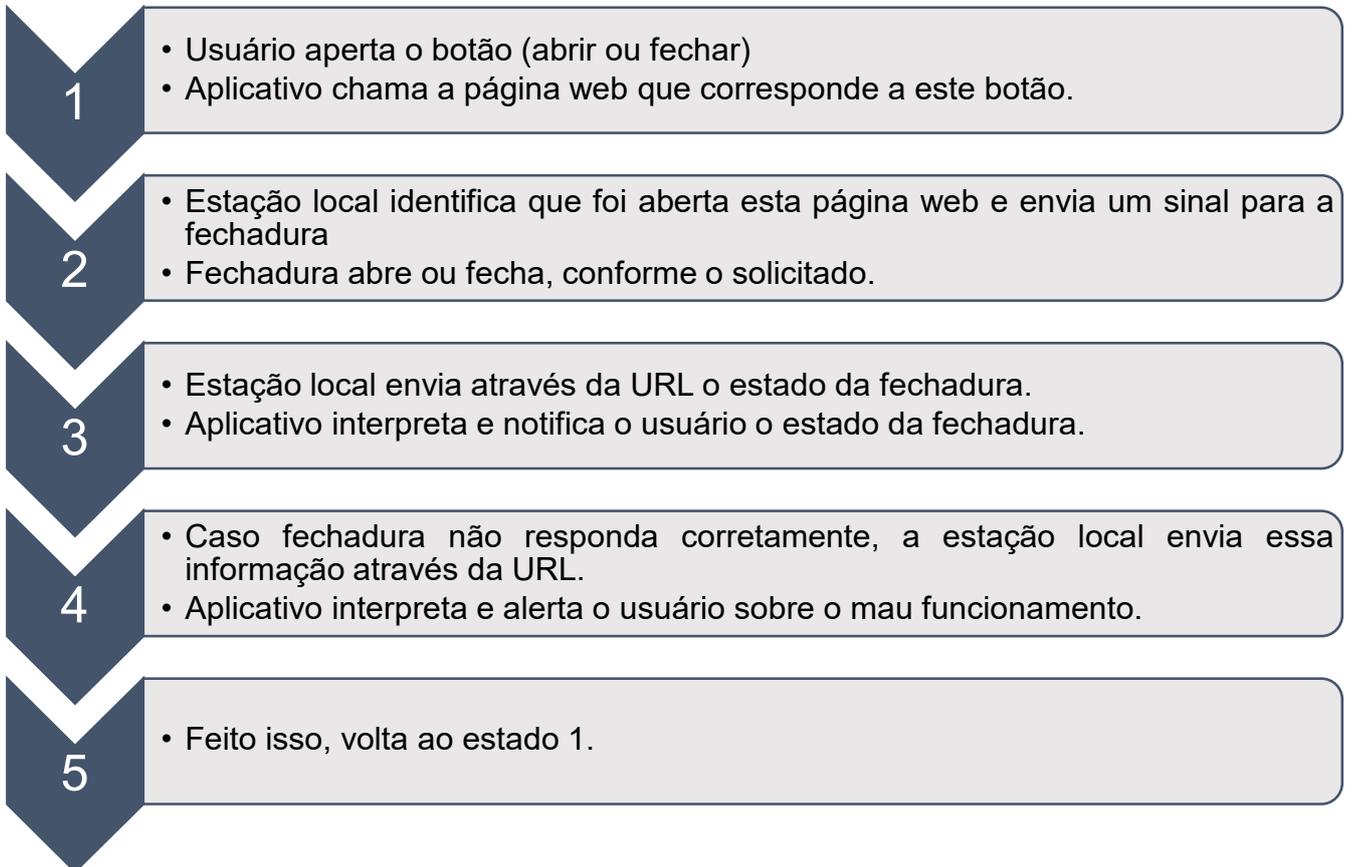
Como já detalhado na seção 4.2.9, no módulo blocos é programada a parte funcional do aplicativo e ele define como cada ícone do designer irá funcionar quando acionado. O código desenvolvido no módulo blocos se encontra no Anexo B.

Mas basicamente a lógica (Figura 21) por trás dos botões mostrados na Figura 19 é a seguinte: quando o usuário clica no botão “abrir”, o aplicativo seta uma variável com a URL: <http://192.168.15.18/gpio/0> e abre ela na rede conectada (ou seja, abre uma página web). Ao mesmo tempo, a estação local (especificamente o ESP8266) identifica através do servidor que foi requisitado essa URL e com isso ele interpreta que é para abrir a fechadura. Após abrir essa URL, o aplicativo notifica através de mensagem de texto que a fechadura está aberta. O mesmo procedimento é realizado ao apertar o botão “fechar”.

Caso, o sensor de fim de curso identifique que a fechadura não funcionou

corretamente, o servidor através da mesma URL, envia uma mensagem ao aplicativo indicando o mau funcionamento. O aplicativo por sua vez, alerta o usuário de que a fechadura não respondeu como o esperado.

Figura 21 - Lógica do aplicativo



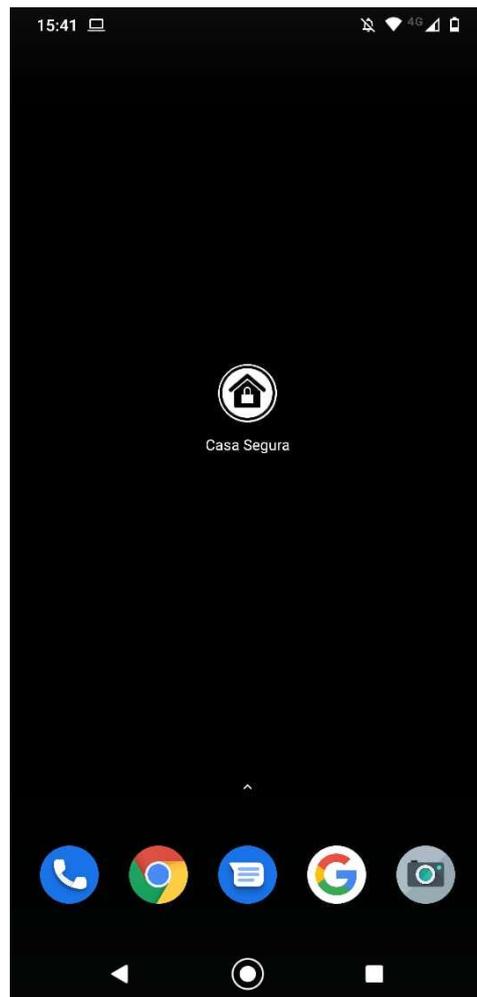
Fonte: Elaborada pela autora.

## 6. Resultados e discussões

Foram realizados testes com um celular e os resultados obtidos foram:

- Após a instalação, o aplicativo ficou como na Figura 22 na área de trabalho do celular.

Figura 22 - Aparência do aplicativo na área de trabalho



Fonte: Elaborada pela autora.

Ao acessar o aplicativo, ele abriu a tela inicial da Figura 23. Clicando em entrar, ele abriu a tela principal (Figura 24) com os comandos para controlar a fechadura elétrica. Ao apertar o botão “abrir”, foi obtida a resposta “fechadura aberta” mostrada na Figura 24:

Figura 23 - Tela inicial



Fonte: Elaborada pela autora.

Figura 24 - Resposta ao apertar o botão “abrir”



Fonte: Elaborada pela autora.

E ao apertar o botão “fechar”, foi obtida a resposta “fechadura fechada” mostrada na Figura 25:

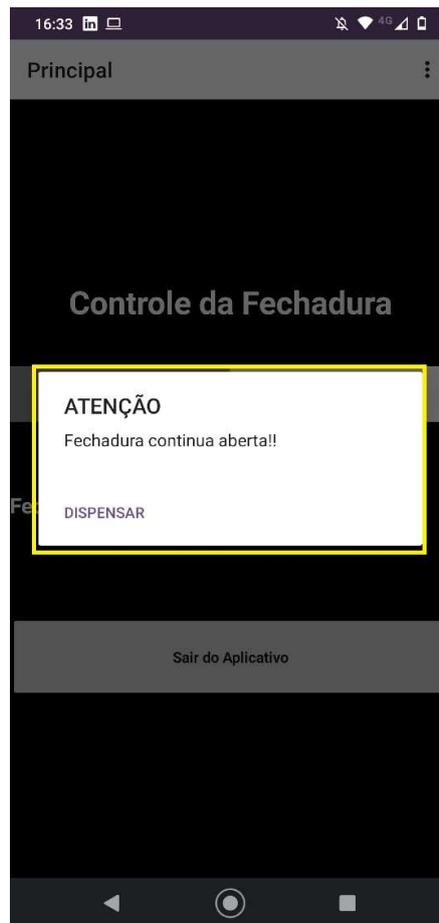
Figura 25 - Resposta ao apertar o botão “fechar”



Fonte: Elaborada pela autora.

A fechadura ficou aberta logo após apertar o botão “fechar”, e por isso o dispositivo mostrou a notificação da Figura 26:

Figura 26 - Notificação



Fonte: Elaborada pela autora.

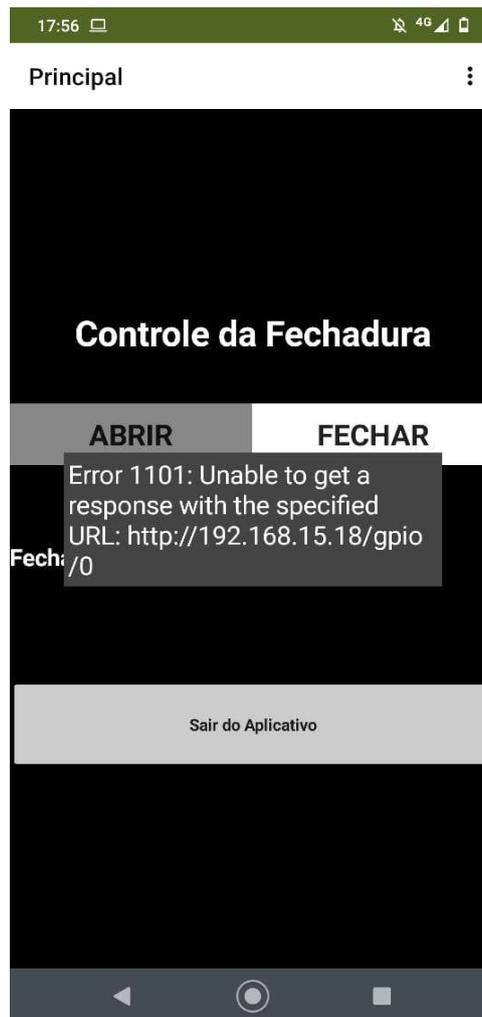
O sistema também notificou-o usuário quando a fechadura se manteve fechada logo após ter apertado o botão “abrir.”

## 6.1. Falhas

### 6.1.1. Falta de conexão

Caso aconteça do NodeMcu, do celular ou ambos estarem sem internet já não é possível utilizar o sistema para controlar a fechadura. Se ambos estiverem conectados à internet em redes Wi-Fi diferentes será possível entrar no aplicativo, mas não será possível controlar a fechadura, pois ambos precisam estar na mesma rede. Quando isso ocorre, o aplicativo responde com a mensagem de erro da Figura 27.

Figura 27 - Mensagem de erro



Fonte: Elaborada pela autora.

Uma solução para ambas as falhas aqui descritas, seria a de construir um aplicativo nativo que é melhor explicado na seção 6.1.2.

### 6.1.2. Falha por tempo de resposta

Nesse sistema de conexão entre NodeMcu e dispositivo que foi utilizado, só é possível atender uma única requisição por vez. E cada requisição tem um tempo de resposta que será mostrado na seção 5.2.

Se ocorrer de o usuário apertar o botão de abrir e fechar repetidas vezes, de forma que o tempo de solicitar a requisição seja menor do que o tempo de resposta da requisição, acontece uma falha no dispositivo. Ele não consegue responder, perde a conexão e mostra a mensagem da Figura 27 para o usuário.

Uma solução para este problema, seria a de alocar o aplicativo construído em um servidor que permite respostas rápidas e múltiplas requisições.

### **6.1.3. Falta de energia**

Caso aconteça de a residência ficar sem energia, o sistema da fechadura elétrica também não funcionará, e dessa forma ele irá se manter em seu último estado (aberto ou fechado).

Uma solução para isso é o uso de baterias para manter ligado a estação local e o roteador Wi-Fi da residência, pois o sistema funciona sem internet, ele só precisa estar conectado a rede Wi-Fi.

## **6.2. Tempo de resposta**

Podemos considerar dois tempos de resposta para o sistema:

- O tempo de resposta da fechadura, que é o quanto a fechadura demora para abrir e fechar após o usuário apertar o botão abrir/fechar.
- O tempo de resposta do celular, que é o quanto demora para aparecer a mensagem no celular após o usuário apertar o botão abrir/fechar.
- Portanto, foi calculado cada um desses tempos de resposta. E para isso, só foi necessário apertar o botão abrir e fechar 100 vezes, como na Figura 28.

Figura 28 - Cálculo tempo médio de resposta

```

COM3
Tempo médio de resposta da fechadura:490.04
Tempo médio de resposta do celular:1080.52
Dispositivo encontrado
tempo1=148871076
GET /gpio/1 HTTP/1.1
Fechadura Fechada
tempo3=148872148

Amostra:100.00
Tempo médio de resposta da fechadura:489.76
Tempo médio de resposta do celular:1080.43
Dispositivo encontrado
tempo1=151071106
GET /gpio/0 HTTP/1.1
tempo3=151072199
Dispositivo desconectado
Amostra:101.00
Tempo médio de resposta da fechadura:489.49
Tempo médio de resposta do celular:1079.96
Dispositivo encontrado
tempo1=151845719
GET /gpio/1 HTTP/1.1
Fechadura Fechada
tempo3=151846798
Dispositivo desconectado
Amostra:102.00
Tempo médio de resposta da fechadura:489.26
Tempo médio de resposta do celular:1079.95

 Auto-rolagem  Show timestamp
Nova-linha 19200 velocidade Deleta e saída

```

Fonte: Elaborada pela autora.

O tempo de resposta da fechadura e do celular obtido, se encontra na Tabela 5.

Tabela 5 - Tempo de resposta médio

Item	Tempo de Resposta Médio (mS)
Fechadura	0,490
Mensagens no Celular	1,080

Fonte: Elaborada pela autora.

De acordo com o livro *Usability Engineering* (NIELSEN, 1993), existem 3 limites de tempo principais que são determinados pelas habilidades de percepção humana, e que devem ser considerados ao otimizar o desempenho de páginas da web e aplicativos:

- **0,1 segundo** é o limite de tempo para que o usuário entenda que o sistema está carregando e irá abrir em breve, e por isso ele espera.

- **1,0 segundo** é o limite de tempo para que o usuário permaneça esperando o sistema carregar sem pensar em fazer outras coisas, mesmo que ele perceba o atraso.
- **10 segundos** é o limite de tempo para conseguir manter o usuário esperando o sistema carregar, mostrando a ele mensagens explicativas em relação a demora. Para atrasos mais longos, os usuários vão querer realizar outras tarefas enquanto esperam o sistema carregar.

Conclui-se que o sistema responde de forma rápida. E como não há necessidade de abrir e fechar a fechadura muitas vezes em um curto intervalo de tempo, mais o fato dele responder uma solicitação por vez em no máximo 1mS, não atrapalha o funcionamento correto do aplicativo.

### 6.3. Dificuldades

#### 6.3.1. Hardware

A primeira dificuldade encontrada foi a de como construir um aplicativo sem que precisasse saber linguagem de programação para Android ou iOS. Por isso optou pelo App Inventor.

Tendo uma forma de construir o aplicativo, a próxima dificuldade foi entender como seria construído um circuito que acionasse a fechadura. Em um primeiro momento, foi decidido utilizar o Raspberry Pi pela facilidade de programá-lo e por já ter em sua configuração a conexão à internet via Wi-Fi. Porém, ele encareceu o projeto por conta do seu preço no mercado, além de ter outras funções que não iriam serem utilizadas.

Por isso, foi decidido procurar dispositivos mais baratos e simples, encontrou-se então o NodeMcu e um módulo com relé Wi-Fi. Ambos são dispositivos com menor poder de processamento, se conectam à internet via Wi-Fi, e compactos.

Por fim, resolveu-se utilizar o NodeMcu por ter portas que poderiam ser utilizadas para sensores e leds.

Além disso, teve-se dificuldade em encontrar uma solução para a alimentação dos dispositivos, pois o NodeMcu precisa de 5V e a fechadura de 12V, e não é viável utilizar duas fontes no sistema. Para resolver este problema, utilizou-se o regulador de tensão L7805.

### 6.3.2. Programação

Entender a comunicação entre o NodeMcu e o aplicativo foi a maior dificuldade da programação do dispositivo e que mais levou tempo para desenvolver.

Foi necessário encontrar uma forma de trocar informações entre dois dispositivos que são programados em plataformas diferentes: App Inventor e Arduino. Estudando as bibliotecas e aplicações do Arduino, encontrou-se a documentação “web server” (Arduino, 2018)

É uma documentação do Arduino que cria um servidor de web simples. É possível realizar solicitações e respostas através de um HTTP (também conhecidos como documentos em HTML), linguagem utilizada hoje para construção de páginas na web e que é possível acessar através de uma URL (conhecido também como link).

Portanto, a comunicação entre os dispositivos foi resolvida através de solicitações HTTP. Onde diferentes URL's significam diferentes solicitações.

A segunda dificuldade enfrentada, foi a necessidade de encontrar uma biblioteca que auxiliasse na escolha da rede Wi-Fi, quando o NodeMcu não encontrasse conexão. Após alguns testes, utilizou-se a biblioteca WiFi Manager.

E no fim, a própria biblioteca já traz toda uma página em HTML construída para essas ocasiões. Precisou entender como funciona e como adaptá-la ao circuito nas ocasiões em que o usuário não tivesse conexão com a internet. E essa dificuldade, trouxe benefícios ao sistema: a implementação de leds que notificam o usuário quando há ou não conexão à internet.

Já na programação do App Inventor, a dificuldade encontrada refere-se à construção do design do aplicativo. Ele possui uma interface de design limitada, o que não favorece a construção de um aplicativo moderno.

### 6.4. Pontos positivos

Entre os pontos positivos deste projeto, podemos citar: a satisfação de realizá-lo sem precisar de conhecimentos técnicos além do que foi aprendido durante a graduação.

A construção do circuito NodeMcu/fechadura foi feita de forma rápida, especialmente depois que ficou claro o que precisava ser feito.

O datasheet do L7805 e do módulo do relé já trouxe formas de como o circuito deve

ser ligado conforme a aplicação.

A chave de fim de curso também foi facilmente implementada pelo fato de ser uma aplicação simples com respostas digitais 1 ou 0. Tudo que precisou foi entender como ligá-la no circuito e fazer uma lógica condicional no sistema.

## 7. Conclusão

Com os resultados obtidos é possível concluir que o sistema aqui proposto funciona de acordo com o especificado. Porém, são necessários ajustes para aprimorar a experiência do usuário com o aplicativo e tornar o sistema ainda mais seguro.

O aplicativo abre e fecha a fechadura elétrica em um tempo médio de 0,5mS, o que é bem rápido. O usuário recebe uma mensagem de confirmação no aplicativo em um tempo médio de 1mS.

Para utilizar este sistema em residências será necessário realizar testes com fechaduras elétricas maiores e robustas, pois a fechadura utilizada no protótipo não garante a segurança apenas o funcionamento do trabalho aqui proposto.

Esse trabalho mostrou que é possível criar um sistema com uma fechadura elétrica para aprimorar a segurança dentro de residências.

## **8. Trabalhos futuros**

Apesar de funcionar corretamente, o sistema da fechadura elétrica pode ser melhorado em diversos pontos, contribuindo para seu funcionamento e segurança da residência.

### **8.1.1. Autenticação**

Para evitar que qualquer pessoa acesse o aplicativo no celular do usuário e controle a fechadura elétrica, é indicado construir um sistema de autenticação para entrar no aplicativo.

### **8.1.2. Aplicativo nativo**

Neste trabalho utilizou uma ferramenta de terceiro (App Inventor) para construir o aplicativo, mas para garantir segurança de dados, o ideal seria construí-lo utilizando uma linguagem de programação para Android e iOS e posteriormente, aloca-lo em um servidor para melhorar os tempos de resposta e aumentar o número de requisições possíveis sem que aconteça uma falha no sistema.

### **8.1.3. Registro**

A implementação de um sistema que registre quem utilizou o aplicativo com hora e data e assim quando for preciso saber esses dados eles estarão lá registrados.

### **8.1.4. Rede Wi-Fi**

A construção de uma página dentro do aplicativo para que a pessoa consiga escolher a rede Wi-Fi para conectar o NodeMcu, melhoraria a experiência do usuário.

### **8.1.5. Sensor magnético na porta**

A instalação de um sensor magnético na porta para informar ao usuário quando a porta estiver aberta ou não, complementando o sistema de segurança.

### **8.1.6. Servidor Externo**

De forma a aprimorar a segurança e utilizar um domínio próprio para a comunicação

entre os dispositivos aqui mostrados (estação local e aplicativo), utilizar um servidor externo.

#### **8.1.7. Conexão IoT**

Realizar a comunicação entre os aplicativos através da conexão IoT, e dessa forma poder implementar o sistema.

## 9. Referências

ATHOS ELECTRONICS. s.d. Disponível em: <https://athoselectronics.com/regulador-de-tensao/>. Acessado em 23 de Fevereiro de 2021.

Bau da Eletrônica Componentes Eletrônicos. s.d. Disponível em: <https://www.baudaeletronica.com.br/modulo-wifi-ESP8266-NodeMcu-ESP-12e.html>. Acessado em 2021 de Fevereiro de 22.

Bau da Eletrônica Componentes Eletrônicos. s.d. Disponível em: <https://www.baudaeletronica.com.br/modulo-rele-5v.html>. Acessado em 22 de Fevereiro de 2021.

CLUBE DO HARDWARE. 16 de Janeiro de 2018. Disponível em: <https://www.clubedohardware.com.br/topic/1280610-conector-jack-p4-como-usar>. Acessado em 25 de Fevereiro de 2021.

DOMINGUES, Ricardo Gil. **A Domótica Como Tendência na Habitação. Rio de Janeiro, 2013.**

FILIFELOP. FilipeFlop Componentes Eletrônicos. s.d. Disponível em: <https://www.filipeflop.com/>. Acessado em 21 de Fevereiro de 2021.

GROSSMANN, Gustavo Henrique. **IoT Smart Lock: Sistema De Fechadura. Curitiba, 2018.**

GUSMAO, Arthur Trindade; ABREU DE, Agostinho Luiz da Silva; CASTRO, Ronaldo de Freitas Zampolo. **Fechadura Baseada em Reconhecimento Facial Via. XXXIV Simposio Brasileiro De Telecomunicações, 2016.**

I DO CODE. s.d. Disponível em: <https://idocode.com.br/blog/programacao/programacao-em-blocos/>. Acessado em 23 de Fevereiro de 2021.

LAB, MIT Media. App Inventor. 2012. Disponível em: <https://appinventor.mit.edu/>. Acessado em 12 de Março de 2021.

MORAIS, José. Portal Vida de Silício. 9 de Junho de 2017. Disponível em: <https://portal.vidadesilicio.com.br/o-que-ESP8266-NodeMcu/>. Acesso em 22 de Fevereiro de 2021.

MURATORI, José Roberto, e Paulo Henrique Dal Bó. Automação residencial: histórico, definições e conceitos. Março de 2011. Disponível em: [https://www.osetoreletrico.com.br/wp-content/uploads/2011/04/Ed62\\_fasc\\_automacao\\_capl.pdf](https://www.osetoreletrico.com.br/wp-content/uploads/2011/04/Ed62_fasc_automacao_capl.pdf). Acesso em 20 de Julho de 2020.

MURTA Gustavo. Eletrogate. 01 de Março de 2018. Disponível em: <https://blog.eletrogate.com/NodeMcu-ESP12-introducao-1/>. Acesso em 22 de Fevereiro de 2021.

NIELSEN, Jakob. **Usability Engineering. Morgan Kaufmann, 1993.**

PORTAL DO GOVERNO, 2020. Disponível em: <https://www.ssp.sp.gov.br/estatistica/mapas.aspx>. Acessado em 20 de Julho de 2020.

SEMICONDUCTOR, Fairchild. FilipeFlop. s.d. Disponível em: [https://img.filipeflop.com/files/download/Datasheet\\_7805.pdf](https://img.filipeflop.com/files/download/Datasheet_7805.pdf). Acessado em 25 de Fevereiro de 2021.

SISLITE. s.d. Disponível em: <http://www.sislite.pt/domus.htm>. Acessado em 28 de Julho de 2020.

SOMMERVILLE, Ian. **Engenharia de Software, 9 ed. Pearson, 2011.**

SYSTEMS, ESPressif. “ESPressif.” ESPressif. 2021. Disponível em: [https://www.ESPressif.com/sites/default/files/documentation/0a-ESP8266ex\\_datasheet\\_en.pdf](https://www.ESPressif.com/sites/default/files/documentation/0a-ESP8266ex_datasheet_en.pdf). Acessado em 25 de Maio de 2021.

TELLUS. 2020. Disponível em: <https://tellus.org.br/instituto/tecnologias-seguranca-publica/>. Acessado em 20 de Julho de 2020.

V. PEREIRA, Vitor; SAMAHERNI, M. Dias; KURIOS, I. P. M. de Queiroz. **“Sistema Inteligente Para Controle de Acesso e Monitoramento de Múltiplos Ambientes (Class Control).”** *Sociedade Brasileira de Computação*, 19 de Novembro de 2019: 25-30.

## ANEXO A – Código do NodeMcu

```
//Limpar as comunicações para evitar problemas de envio
//sequencias de dados e gastos de memória RAM
struct tcp_pcb;
extern struct tcp_pcb* tcp_tw_pcbs;
extern "C" void tcp_abort (struct tcp_pcb* pcb);

void tcpCleanup ()
{
  while (tcp_tw_pcbs != NULL)
  {
    tcp_abort(tcp_tw_pcbs);
  }
}

#include <WiFiClient.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <DNSServer.h>
#include <WiFiManager.h>

// definição do pino que será usado na saída
#define ativarTranca 5
#define semConexao 0
#define comConexao 2
#define pinosinal 13
int leitura;
int cont2;
int leiturasensor;
boolean x;
float aux1=0;
float aux2=0;
float aux3=0;
```

```
IPAddress ip(192,168,15,18);
IPAddress gateway(192,168,15,1);
IPAddress subnet(255,255,255,0);

char ssid[] = "Gustavo";
char password[] = "Thunder_54";
WiFiServer server(80);

void setup() {
  // setar pino do input do relé como saída
  pinMode(ativarTranca, OUTPUT);
  digitalWrite(ativarTranca, 0);
  pinMode(semConexao,OUTPUT);
  pinMode(comConexao,OUTPUT);

  //Define o pino do sensor hall como entrada
  pinMode(pinosinal, INPUT_PULLUP);

  // iniciar wi fi com senha e password
  WiFi.begin(ssid, password);
  WiFi.config(ip, gateway, subnet);

  // iniciar conexão serial (pra debugar)
  Serial.begin(19200);

  //verificar estado do wi fi
  if(WiFi.status() != WL_CONNECTED) {
    digitalWrite(semConexao, 1);
    digitalWrite(comConexao, 0);
    delay(500);
    Serial.println("...conectando");
    ESP.wdtFeed();
    delay (10000);
  }
  if(WiFi.status() == WL_CONNECTED){
    Serial.println("conectado :D");
    digitalWrite(semConexao, 0);
    digitalWrite(comConexao, 1);
```

```
}
else if(WiFi.status() != WL_CONNECTED){
  WiFiManager wifiManager;
  wifiManager.autoConnect("AutoConnectAP");
}

// Iniciar o servidor
server.begin();
Serial.println("Servidor iniciado em: ");

//Mostrar endereço de IP
Serial.println(WiFi.localIP());
}

void loop() {
tcpCleanup();

//Verifica se tem algum dispositivo conectado
WiFiClient client = server.available();
if(!client) {
  return;
}

//Aguarda até o dispositivo enviar dado
Serial.println("Dispositivo encontrado");
while(!client.available()){
  delay(1);
}

int tempo1 = micros();
Serial.println(tempo1);
//Le os dados que recebeu
String dado = client.readStringUntil('\r');
Serial.println(dado);
client.flush();

//Compara os dados

int cont;
```

```

if(dado.indexOf("/gpio/0") != -1)
  cont = 1
;
else if(dado.indexOf("/gpio/1") != -1)
  cont = 0;
else {
  Serial.println("Comando Inválido");
  client.stop();
  return;
}

//Acionar relé de acordo com o dado
digitalWrite(ativarTranca, cont);
int tempo2 = micros();

leitura = digitalRead(pinosinal);

if(leitura == 1 & cont == 0)
{
  x = 1;
  Serial.println("Tranca Fechada");
}
if (leitura == 0 & cont == 1)
{
  x = 0;
  Serial.println("Tranca Aberta");
}

client.flush();
tcpCleanup();

//Prepara a rESPosta para o dispositivo
String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HTML>\r\n<html>\r\nTranca esta
";
s += (cont)?"aberta":"fechada";
s += "</html>\n";

//Manda a rESPosta para o dispositivo

```

```
client.print(s);
client.print(x);
int tempo3 = micros();

delay(1);
Serial.println("Dispositivo desconectado");

int rESPostafechadura = (tempo2 - tempo1);
aux1 = (aux1+1);
Serial.println(aux1);
aux2 = (aux2 + rESPostafechadura);
float media1 = (aux2/aux1);

Serial.print("Tempo médio de rESPosta da fechadura:");
Serial.println(media1);

int rESPostacelular = (tempo3 - tempo1);
aux3 = (aux3 + rESPostacelular);
float media2 = (aux3/aux1);

Serial.print("Tempo médio de rESPosta do celular:");
Serial.println(media2);

}
```

## ANEXO B – Lógica em blocos do aplicativo

```

quando Sair_app .Clique
fazer
  abrir outra tela com valor inicial nomeDaTela " Screen2 "
  valorInicial " 0 "

```

```

inicializar global status para " 0 "
inicializar global status2 para " 0 "

quando Abrir .Clique
fazer
  ajustar Web1 . Url para " http://192.168.15.18/gpio/0 "
  chamar Web1 .Obter
  se obter global status = 0
  então
    chamar Notificador1 .MostrarDiálogoDeMensagem
      mensagem " Tranca continua fechada!! "
      titulo " ATENÇÃO "
      textoBotão " Dispensar "

```

```

quando Fechar .Clique
fazer
  ajustar Web1 . Url para " http://192.168.15.18/gpio/1 "
  chamar Web1 .Obter
  se obter global status = 1
  então
    chamar Notifier1 .MostrarDiálogoDeMensagem
      mensagem " Tranca continua aberta!! "
      titulo " ATENÇÃO "
      textoBotão " Dispensar "

```

+

```
quando Web1 .RecebeuTexto
  url códigoDeResposta tipoDaResposta conteúdoDaResposta
  fazer
    se contém texto obter conteúdoDaResposta
      parte "Tranca esta aberta"
      então ajustar Legenda1 . Texto para "Tranca Aberta"
    se contém texto obter conteúdoDaResposta
      parte "Tranca esta fechada"
      então ajustar Legenda1 . Texto para "Tranca Fechada"
    se contém texto obter conteúdoDaResposta
      parte "0"
      então ajustar global status para 0
    se contém texto obter conteúdoDaResposta
      parte "1"
      então ajustar global status para 1
    se contém texto obter conteúdoDaResposta
      parte "2"
      então
    se contém texto obter conteúdoDaResposta
      parte "3"
      então
```

## ANEXO C – Parâmetro do ESP8266

### C.1 – Parâmetros ESP8266

Categories	Items	Values
WiFi Paramters	WiFi Protocdes	802.11 b/g/n
	Frequency Range	2.4GHz-2.5GHz (2400M-2483.5M)
Hardware Paramaters	Peripheral Bus	UART/HSPI/I2C/I2S/Ir Remote Contorl
		GPIO/PWM
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40°~125°
	Ambient Temperature Range	Normal temperature
	Package Size	16mm*24mm*3mm
	External Interface	N/A
Software Parameters	Wi-Fi mode	station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network) / download and write firmware via host
	Ssoftware Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App