UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DAVI SCHULT KOHN

FERNANDO ADELINO DA SILVA

FECHADURA ELETRÔNICA CONTROLADA POR APLICAÇÃO VIA WI-FI

CURITIBA

2022

DAVI SCHULT KOHN
FERNANDO ADELINO DA SILVA

FECHADURA ELETRÔNICA CONTROLADA POR APLICAÇÃO VIA WI-FI

ELECTRONIC LOCKER CONTROLLED BY APPLICATION VIA WI-FI

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica do Curso de Bacharelado em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná.

Orientador: Guilherme de Santi Peron

CURITIBA

2022

DAVI SCHULT KOHN
FERNANDO ADELINO DA SILVA

FECHADURA ELETRÔNICA CONTROLADA POR APLICAÇÃO VIA WI-FI

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica do Curso de Bacharelado em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 21/novembro/2022

Guilherme de Santi Peron
Doutor
Universidade Tecnológica Federal do Paraná

João Luiz Rebelatto
Doutor
Universidade Tecnológica Federal do Paraná

Luiz Fernando Copetti
Mestre
Universidade Tecnológica Federal do Paraná

CURITIBA
2022

Dedicamos este trabalho a todo o curso de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná, aos colegas e professores, a quem agradecemos as bases que deram para nos tornarmos as pessoas que somos hoje.

AGRADECIMENTOS

Agradecemos, antes de quaisquer outras pessoas, a nossos amigos, familiares e professores por todo apoio, paciência e compreensão em nossos momentos de ausência física, esgotamento psicológico e incertezas de sucesso. Agradecemos também ao professor orientador Guilherme Peron pela presença, disposição e orientação em momentos nos quais nosso conhecimento e preparo não eram suficientes para proceder no desenvolvimento deste trabalho. Por último, mas não menos importante, gostaríamos de prestar gratidões póstumas ao falecido professor Dr. Hugo Vieira Neto por sua dedicação espetacular ao ensino acadêmico, apoio psicológico e interesse pelo aprendizado de seus discentes. Tais virtudes fazem e continuarão fazendo profunda falta ao grupo docente da Universidade Tecnológica Federal do Paraná - Campus Curitiba.

A depressão é uma prisão de alta segurança construída pelos medos que insistimos em não enfrentar! (OLIVER, Sandra).

RESUMO

O constante aumento do distanciamento social causado tanto pelo avanço exponencial das tecnologias em telecomunicações e sistemas embarcados de baixíssimo consumo de energia quanto pela recente pandemia causada pelo COVID-19 têm causado um grande impacto nas relações humanas. Foi com esta base contextual que o projeto de uma fechadura eletrônica composta fisicamente por uma fonte externa alimentada pela rede elétrica foi concebido. Por meio de um microcontrolador conectado a um *boosterpack* que mostra informações através de um *display* gráfico e ainda sinaliza operações de abertura e fechamento sonoramente através de um *buzzer*. Além disso, através de uma placa auxiliar para chaveamento, um módulo *Wi-Fi* e uma trava solenóide, o usuário pode controlar o acesso, seja ele local ou remoto, a qualquer cômodo ou imóvel de sua posse independentemente de onde ele estiver, desde que tenha acesso à Internet e uma conta no *Gmail*. Utilizando uma aplicação *web*, é possível realizar um cadastro em um sistema que lista as residências do usuário bem como as suas fechaduras disponíveis. Desta forma, o usuário consegue controlar os seus dispositivos cadastrados, além de monitorar as atividades de abertura da fechadura com um sistema de *logs*. Isso possibilita ao dono da fechadura ceder, alugar ou mesmo conceder acesso de pessoas desejáveis em suas propriedades, abrindo assim um leque totalmente novo de possibilidades nas interações de inquilinismo e afins.

Palavras-chave: internet das coisas; fechadura eletrica; inquilinismo; acesso remoto.

ABSTRACT

The increase of social distancing caused both by the exponential advance of technologies in telecommunications and embedded systems with very low energy consumption and by the recent coronavirus disease pandemic has had a great impact on relationships and where new technologies are inserted in this environment. It was with this contextual basis that the project of an electronic lock physically composed of an external source fed by the electrical grid was conceived. Through a microcontroller connected to a boosterpack used to display information through a graphic display and to inform the opening and closing operation soundly through a buzzer. Besides that, using an auxiliary board as a switch, a Wi-Fi module and a solenoid lock, the user can control access, whether local or remote, to any room or property in his possession, regardless of where he is, as long as he has Internet access and a Gmail account. Using a web application, it is possible to register yourself in a system to see your residences as well as their available locks. In this way, the user can control the registered devices, and track some information about the operations with a log system. This makes it possible for the owner of the lock to yield, or even grant people's access, thus allowing a range of new possibilities for renting purposes.

Keywords: internet of things; electronic lock; renting; remote access.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

Siglas

3D	*3-Dimensional*
ADC	*Analog-to-Digital Converter*
AT	*Attention*
CAN	*Controller Area Network*
COVID-19	*Coronavirus Disease of 2019*
DC	*Direct Current*
EEPROM	*Electrically Erasable Programmable Read-Only Memory*
GCC	*GNU Compiler Collection*
GPIO	*General Purpose Input/Output*
HTTP	*Hypertext Transfer Protocol*
I2C	*Inter-Integrated Circuit*
IAR	*Ingenjörsfirman Anders Rundgren*
ICDI	*In-Circuit Debug Interface*
IP	*Internet Protocol*
LCD	*Liquid Crystal Display*
LED	*Light Emitter Diode*
NonOS	*Non Operating System*
NoSQL	*Not Only SQL*
PCI	*Peripheral Component Interconnect*
PLA	*Polylactic Acid*
PWM	*Pulse Width Modulation*
RFID	*Radio Frequency Identification*

RGB	*Red-Green-Blue*
RMS	*Root Mean Square*
RTOS	*Real-Time Operating System*
SD	*Secure Digital*
SDK	*Software Development Kit*
SHA-1	*Secure Hash Algorithm 1*
TCP	*Transmission Control Protocol*
TFT	*Thin-Film Transistor*
THT	*Through-hole-technology*
UART	*Universal Asynchronous Receiver-Transmitter*
UDP	*User Datagram Protocol*
URI	*Uniform Resource Identifier*
WEP	*Wired Equivalent Privacy*
WI-FI	*Wireless Fidelity*
WPA-PSK	*Wireless Protected Access - Pre-Shared Key*

SUMÁRIO

1 INTRODUÇÃO

Com o surgimento da internet no final da década de 60 e, posteriormente, a popularização da banda larga e redes sem fio, foi possível conectar pessoas do mundo inteiro em tempo real, facilitando a comunicação. Com esse crescimento e o aumento de acessibilidade à internet em crescente evolução, pesquisas apontam que o desejo por soluções tecnológicas conectadas à sustentabilidade, conforto e segurança doméstica está em alta no Brasil (DATAFOLHA, 2022).

A sociedade moderna está em constante mudança, evoluindo e expandindo rapidamente, assumindo um novo ritmo para suas rotinas e suas atividades do cotidiano. Conseguir administrar o estresse, gerenciar os compromissos e acima de tudo manter a qualidade de vida é o desejo da maior parte das pessoas. Essa rotina acelerada faz com que cada vez mais pessoas busquem conforto, conforme citado anteriormente, levando a um aumento da procura de casas inteligentes (DATAFOLHA, 2022).

Uma casa inteligente pode ser definida como um imóvel que possui equipamentos capazes de serem acionados e controlados por dispositivos móveis ou de forma automática. Dessa forma, ter uma casa automatizada traz muito mais conforto e praticidade no dia a dia das pessoas, além de ocasionar uma economia no gasto de energia elétrica além de um aumento da segurança, visto que dispositivos como câmeras, sensores e fechaduras inteligentes podem atuar como vigilantes de uma residência, algo de suma importância se for levado em conta os altos índices de criminalidade do Brasil.

1.1 INTERNET DAS COISAS

O conceito de internet das coisas está diretamente ligado a uma rede de objetos físicos sendo incorporados a um software, sensores e atuadores e outras tecnologias com o objetivo de estabelecer uma conexão e uma troca de dados entre os dispositivos e sistemas pela internet. Esses dispositivos compreendem desde objetos domésticos comuns para uma automação residencial, como por exemplo uma fechadura eletrônica, até ferramentas industriais mais sofisticadas, através do monitoramento da produtividade de equipamentos e até mesmo no auxilio de prevenção de acidentes.

Mesmo com a escassez de circuitos integrados causada pela crise global de semicondutores, o número de conexões globais de *Internet Of Things* (IOT) cresceu 8% em 2021, totalizando 12,2 bilhões de dispositivos conectados, representando uma queda de crescimento se comparado a anos anteriores. Apesar de uma demanda crescente por soluções de IOT, a crise de semicondutores juntamente com a recente pandemia causada pelo *Coronavirus Disease of 2019* (COVID-19) deve impactar este mercado até 2023. Em 2022 o mercado de internet das coisas deverá manter um crescimento de 18% para 14,4 bilhões de dispositivos conectados e espera-se que, até 2025, este número cresça para aproximadamente 27 bilhões de dispositivos (IOTANALYTICS, 2022).

Considerando o mercado brasileiro, algumas ações já vêm sendo tomadas "O Plano Nacional de Internet das Coisas foi instituído pelo Decreto nº 9.854, de 25 de junho de 2019, e tem como objetivo implementar e desenvolver a Internet das Coisas no País, com base na livre concorrência e na livre circulação de dados, observadas as diretrizes de segurança da informação e de proteção de dados pessoais"(BRASIL, 2019). Além disso, outro fato interessante é que "Segundo o Banco de Desenvolvimento InterAmericano, estima-se que até 2023 teremos 416 milhões de dispositivos IoT no Brasil"(FUTURECOM, 2020).

Devido à crescente alta deste mercado, é de suma importância entender que essa é uma tendência irreversível para o mercado de tecnologia, sendo essa tendência um dos principais motivos que levaram a decisão de escolher este tema para o desenvolvimento deste projeto.

1.2 OBJETIVOS GERAIS

Desenvolver um sistema de segurança residencial composto por uma ou mais fechaduras eletrônicas que podem ser acionadas manualmente utilizando uma senha cadastrada pelo usuário ou por meio de uma aplicação *web*. A abertura da fechadura de forma manual se dá mediante a utilização de um *display* e um teclado em que o usuário precisa digitar a senha corretamente. O sistema ainda possui uma aplicação *web* em que o usuário pode abrir várias fechaduras e monitorar os acessos concedidos por estes dispositivos em tempo real.

1.2.1 OBJETIVOS ESPECIFICOS

- Desenvolver um módulo auxiliar de chaveamento responsável pela ativação de uma trava solenóide que atua como uma fechadura eletrônica;
- Desenvolver um *firmware* para estabelecer uma conexão entre um microcontrolador e um módulo de conexão *Wireless Fidelity* (*Wi-Fi*);
- Desenvolver um sistema servidor capaz de receber e enviar dados para vários dispositivos conectados em uma mesma rede que utilizem um modulo de conexão *Wi-Fi*;
- Desenvolver uma interface gráfica que consiga estabelecer uma conexão com o servidor para efetuar uma troca de informações entre essa interface que recebe dados providos por um usuário e os vários dispositivos que podem estar conectados na rede;
- Exibir uma lista de fechaduras disponíveis e uma interface de monitoramento de abertura das mesmas.

1.3 JUSTIFICATIVA

O desenvolvimento deste projeto buscou atingir uma solução que fosse de fácil aprimoramento. Inicialmente o escopo compreende uma fechadura eletrônica que pode ser controlada remotamente apenas acessando uma página *web*. No entanto o sistema já foi pensado de uma maneira que consiga suportar vários dispositivos, mantendo o foco no conceito descrito anteriormente sobre casas inteligentes, onde diferentes equipamentos conectados entre si visam facilitar diversas atividades cotidianas de uma residência.

1.4 ESTRUTURA DO TRABALHO

Este trabalho possui uma organização em cinco capítulos, incluindo essa introdução. A seguir, apresenta-se uma breve explicação de cada capítulo, bem como algumas particularidades que serão tratadas.

No segundo capítulo é realizada uma fundamentação teórica e apresentados os conceitos técnicos utilizados na construção deste projeto. Neste capítulo são definidas as grandezas estudadas e alguns tópicos importantes no que diz respeito ao *hardware* e *software* da proposta em questão.

No terceiro capítulo são apresentados os detalhes da arquitetura e desenvolvimento do sistema como um todo. Definição de escopo do projeto, descrição dos módulos físicos, implementação do *firmware* e desenvolvimento do *software* são os principais temas abordados.

O capítulo quatro apresenta uma análise e discussão dos resultados com a descrição dos testes realizados e por fim o ultimo capítulo com uma conclusão de todo o trabalho realizado.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os principais conceitos que fazem parte deste projeto. Serão abordados assuntos referentes ao *software* e *hardware* do sistema, às tecnologias utilizadas, aos protocolos de mensagens e às grandezas físicas envolvidas no consumo de energia elétrica.

2.1 TENSÃO ELÉTRICA

Tensão elétrica é a força que impulsiona os elétrons através de um condutor, realizando trabalho. Sua unidade é o Volt(V), que é um indicador da quantidade de energia envolvida na movimentação de uma carga entre dois pontos de um sistema elétrico (BOYLESTAD, 2003a).

A tensão é originada em função do campo elétrico, e deste modo, pode-se separar em tensão continua (CC), cuja origem é um campo elétrico constante com um fluxo de cargas unidirecional, e tensão alternada (CA), cuja origem é um campo elétrico variante no tempo, invertendo seu sentido periodicamente.

2.2 CORRENTE ELÉTRICA

Quando um fio condutor (formado por vários átomos) é conectado a uma bateria (uma fonte de força eletromotriz), as cargas são compelidas a se mover, as cargas positivas se movem em uma direção, enquanto as cargas negativas se movem na direção oposta. A essa movimentação de cargas, dá se o nome de corrente elétrica (SADIKU, 2013).

Da mesma forma que a tensão, a corrente elétrica pode ser CC ou CA. A corrente CC é produzida por uma tensão contínua, cujos elétrons se deslocam num único sentido, enquanto a corrente CA é produzida por uma tensão alternada, cujos elétrons têm deslocamento bidirecional, acompanhando a variação de polaridade da tensão.

2.3 RESISTÊNCIA ELÉTRICA

O fluxo de carga através de qualquer material encontra a oposição de uma força semelhante, em muitos aspectos, ao atrito mecânico. Essa oposição, resultante das colisões entre elétrons e entre elétrons e átomos do material, que converte energia elétrica em uma outra forma de energia, tal como a energia térmica, é denominada resistência do material. A unidade de medida da resistência é o ohm (BOYLESTAD, 2003b).

Quando há circulação de corrente por um material condutor através da aplicação de uma diferença de potencial, pode-se observar que, para um mesmo valor de tensão aplicada

em condutores de diversos materiais, a corrente possuirá valores diferentes. Isto ocorrerá devido às características intrínsecas de cada material (GIACOMIN, 2002).

2.4 REACT JS

React, também conhecido como React.js ou ReactJS é uma biblioteca Javascript de *Front End* gratuita e de código aberto desenvolvida pelo Meta e lançada oficialmente no dia 29 de maio de 2013. Essa biblioteca facilita a criação de uma interface de usuário interativa a partir de pequenos trechos de códigos isolados chamados de componentes. Cada componente possui suas propriedades, definidas como *props*, e também uma hierarquia de elementos que pode ser exibida através do método *render*, que retorna um conjunto de instruções do que o desenvolvedor deseja ver na tela, um elemento React, que é uma descrição simplificada do que precisa ser renderizado. Este método *render* também é responsável por renderizar o componente toda vez que uma alteração em suas propriedades for detectada utilizando o conceito de estado, onde cada propriedade do componente pode receber um novo valor quando o método *setState* é chamado (REACT, 2022).

Fazendo uso dos conceitos definidos anteriormente, uma tela de *places* é um componente do tipo *place* renderizado pelo método *render* que possui como propriedade uma lista de dispositivos, que é inicializada vazia e após a busca dessa lista no banco de dados, o valor da propriedade é atualizado e o componente é renderizado novamente com os novos valores sendo apresentados na tela.

2.5 PROTOCOLOS DE MENSAGENS

Para efetuar a comunicação entre um dispositivo microcontrolado e uma aplicação, é necessário um protocolo para a troca de mensagens entre dois ou mais pontos. Este protocolo que denomina o conjunto de regras que regem a sincronização da comunicação entre os sistemas, definindo um padrão para os dados em transporte.

Os dois principais protocolos de mensagens utilizados no desenvolvimento deste projeto serão abordados a seguir:

2.5.1 HTTP

O protocolo HTTP (*Hypertext Transfer Protocol*) permite a obtenção de recursos, baseado em um modelo de requisição e resposta, sendo a base de qualquer troca de dados na *web*, onde o cliente, geralmente um navegador, faz uma requisição a um servidor e recebe uma resposta com os dados requeridos (SCHIFLETT, 2003).

Este protocolo também define um conjunto de métodos de requisição, responsáveis por indicar a ação a ser executada por um dado recurso. O método GET solicita a representação de um recurso, retornando dados, enquanto o método POST consegue submeter uma nova entidade, ou seja, sendo responsável pelo envio de dados para um determinado servidor. O método PUT substitui dados e por fim o método DELETE remove um recurso específico solicitado pelo cliente. O acesso a esses métodos se dá por meio de uma chamada direta de uma rota definida pela *Uniform Resource Identifier* (URI) (MDN, 2022).

Com essa definição, o *Front End* do sistema é o cliente que faz uma requisição para o servidor, utilizando uma URI e fornecendo um tipo de método, esperando receber informações sobre uma fechadura, informação essa que está armazenada no banco de dados.

2.5.2 TCP/IP

O protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol*), desenvolvido em 1970, é um protocolo que permite que dispositivos se comuniquem a longas distâncias. A *internet* é uma rede na qual as informações são divididas em pequenos pacotes, enviados individualmente por muitas rotas diferentes ao mesmo tempo e, em seguida, reagrupados pelo receptor. O TCP é o componente que coleta e remonta os pacotes de dados, enquanto o IP é responsável por garantir que os pacotes sejam enviados ao destino correto. Utilizando essa definição, o microcontrolador possui um IP e através do protocolo TCP, consegue trocar informações com outro dispositivo como, por exemplo, um computador que está conectado à rede (BRITANNICA, 2022).

2.6 CRIPTOGRAFIA

Criptografia é a conversão de dados de um formato legível em um formato codificado. Os dados criptografados só podem ser lidos ou processados depois de serem descriptografados. Este é um elemento fundamental na segurança de dados, sendo uma das formas de garantir que as informações não sejam roubadas na troca de dados entre um navegador e um servidor (KASPERSKY, 2022).

O algoritmo de criptografia utilizado para garantir a segurança durante a troca de mensagens entre o microcontrolador e o servidor foi o SHA-1 (Secure Hash Algorithm 1), uma função de *hash* criptográfico, criado na década de 1990, que funciona como uma espécie de chave para proteger as informações inseridas em um site ou servidor.

Uma função de *hash* criptográfico, muitas vezes conhecida simplesmente como *hash* – é um algoritmo matemático que transforma qualquer bloco de dados, como um arquivo, uma senha ou alguma informação, em um conjunto alfanumérico de comprimento fixo. Entre as características da função *hash*, destacam-se a saída de tamanho fixo independente do valor da

entrada, ou seja, os dados criptografados sempre possuem a mesma quantidade de caracteres e um valor de entrada que sempre resulta na mesma saída (SERAFIM, 2012).

3 DESENVOLVIMENTO DO SISTEMA

Este capítulo deve tratar de como se pretende realizar o trabalho/pesquisa: Para que as finalidades descritas na seção 1.2 fossem satisfeitas, o primeiro tópico abordado foi a definição do escopo do projeto.

Após a definição do projeto, foram desenvolvidas paralelamente a parte de *hardware/-firmware* e a de *software*. A primeira foi executada na seguinte sequência:

1. Definição dos módulos que fariam a composição física da fechadura eletrônica;
2. Teste de comunicação serial entre módulo central e módulo *Wi-Fi*;
3. Implementação do *firmware* oficial do módulo central;
4. Desenvolvimento do circuito auxiliar de chaveamento da trava solenoide;
5. Projeto da mecânica da fechadura eletrônica;
6. Teste de funcionamento final.

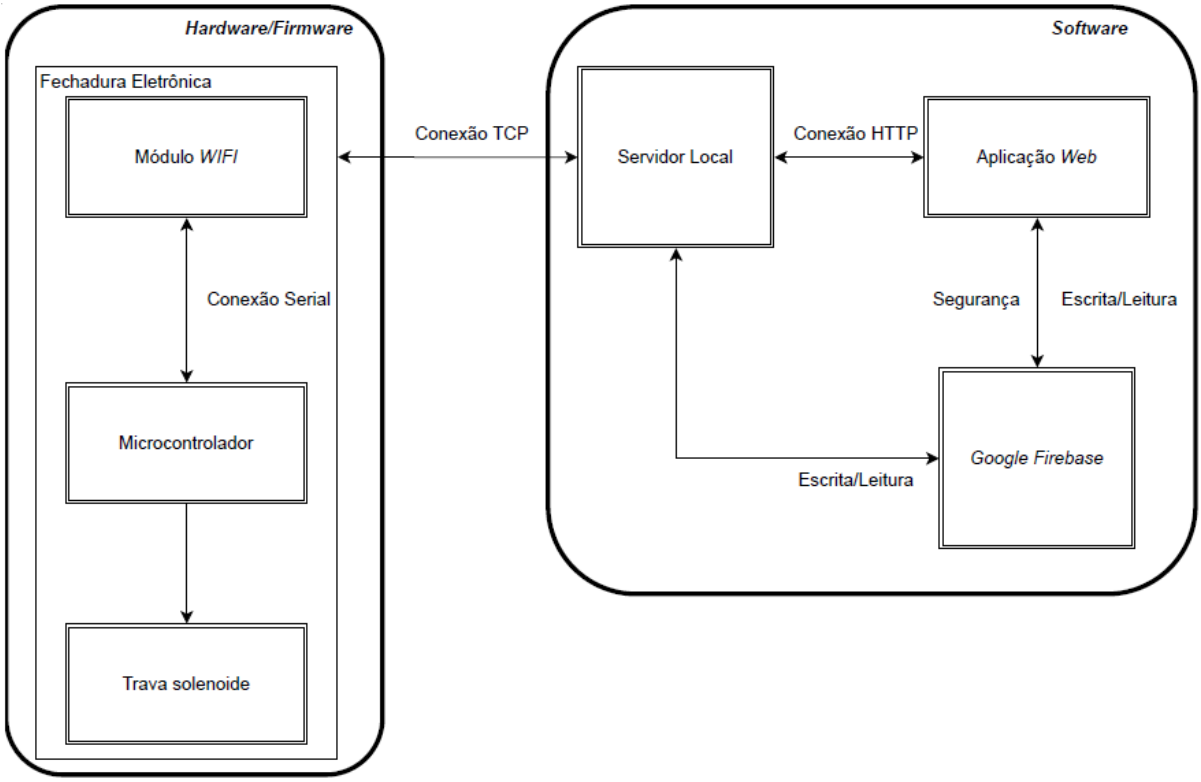
Já a parte referente ao desenvolvimento do *software* foi executada na seguinte sequência:

1. Desenvolvimento do *Back End*;
2. Desenvolvimento do *Front End*;

3.1 DEFINIÇÃO DO ESCOPO DO PROJETO

O projeto consiste em uma fechadura eletrônica, que pode ser acionada por meio de um microcontrolador conectado a um módulo *Wi-Fi*. A fechadura pode ser aberta de maneira manual através de uma senha cadastrada pelo usuário ou remotamente, por meio de protocolos de comunicação TCP e HTTP, utilizando uma aplicação *web*, que pode ser acessada diretamente pelo navegador. O *hardware* do sistema é composto por uma trava solenoide, um microcontrolador e um módulo *Wi-Fi*, enquanto a parte de *software* engloba um *Back End* que funciona como um servidor local, um *Back End* como serviço utilizando o Google Firebase e uma aplicação *web Front End*. O diagrama do escopo do projeto pode ser visualizado na Figura 1.

Figura 1 – Projeto - Diagrama de blocos



Fonte: Autoria própria (2022).

3.2 DEFINIÇÃO DOS MÓDULOS COMPONENTES DO *HARDWARE*

Esta seção descreve os módulos utilizados para o desenvolvimento do *hardware* deste projeto assim como as ideias que motivaram a utilização dos mesmos.

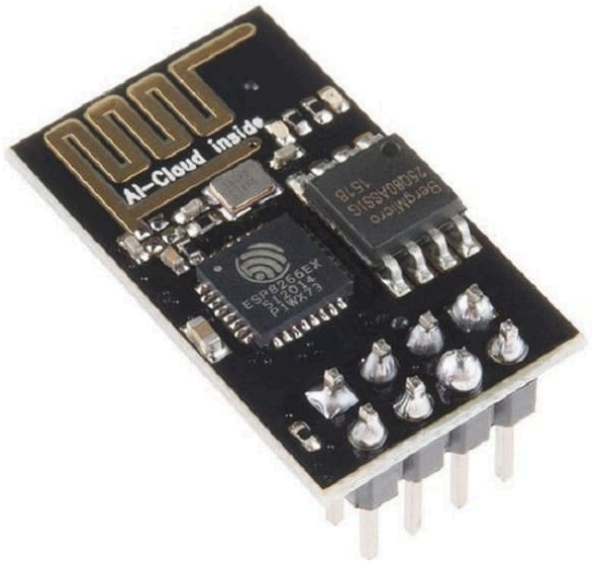
3.2.1 MÓDULO DE COMUNICAÇÃO *WI-FI*

Como um dos motivadores do projeto é poder operar uma fechadura eletrônica de modo remoto, é fundamental que se escolha um módulo *Wi-Fi*. Foram utilizados, basicamente, três filtros para escolher o módulo correto:

- Menores dimensões físicas possíveis;
- Baixo custo de aquisição;
- Opção de utilizá-lo do modo mais simples de comunicação possível, consumindo assim a menor energia de utilização;
- Alto suporte da comunidade de usuários a dúvidas e dificuldades técnicas.

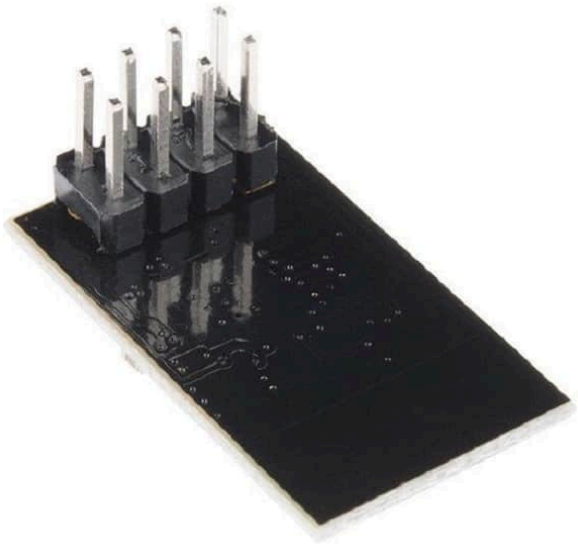
Após pesquisar sobre os itens acima, chegou-se à conclusão de que o módulo seria o ESP8266 ESP-01, uma das versões mais minimalistas de módulos de comunicação *Wi-Fi* do mercado cujo suporte, tanto via fóruns quanto vídeos de tutoriais de utilização, na *internet* é vasto. Este módulo possui somente duas *General Purpose Input/Output* (GPIO) que não foram empregadas para propósito algum, reafirmando que o módulo foi utilizado somente para troca de informações via serial com a rede.

Figura 2 – Módulo ESP8266 ESP-01 - Vista frontal



Fonte: (FILIPEFLOP, 2022).

Figura 3 – Módulo ESP8266 ESP-01 - Vista traseira



Fonte: (FILIPEFLOP, 2022).

Algumas de suas especificações constam abaixo:

- Chip: ESP8266;
- Modelo: ESP-01;
- Tensão de operação: 3,3V;
- Suporte às redes: 802.11 b/g/n;
- Alcance: Aproximadamente 90m;
- Comunicação: Serial (Transmissão/Recepção);
- Suporte às comunicações TCP e UDP (*User Datagram Protocol*);
- Conectores: GPIO, I2C, SPI, UART (*Universal Asynchronous Receiver-Transmitter*), Entrada ADC (*Analog-to-Digital Converter*), Saída PWM (*Pulse Width Modulation*) e Sensor de Temperatura interno;
- Modos de segurança: OPEN/WEP (*Wired Equivalent Privacy*)/WPA-PSK (*Wireless Protected Access - Pre-Shared Key*)/WPA2-PSK/WPA-WPA2-PSK;
- Dimensões: 25x14x1 mm;
- Peso: 7g.

As características supracitadas, baseadas nos pré-requisitos citados anteriormente, confirmam que tal dispositivo é uma escolha razoável, pois possui tensão de operação de 3,3V, cujo valor é o mesmo que é oferecido em alguns pinos do módulo central que será citado adiante. Além disso, oferece suporte a redes 802.11 b/g/n, comunicação serial UART com mais um dispositivo e comunica via protocolo TCP, o qual foi o protocolo da camada de transporte utilizado neste projeto.

3.2.2 MÓDULO DE CENTRAL DE PROCESSAMENTO

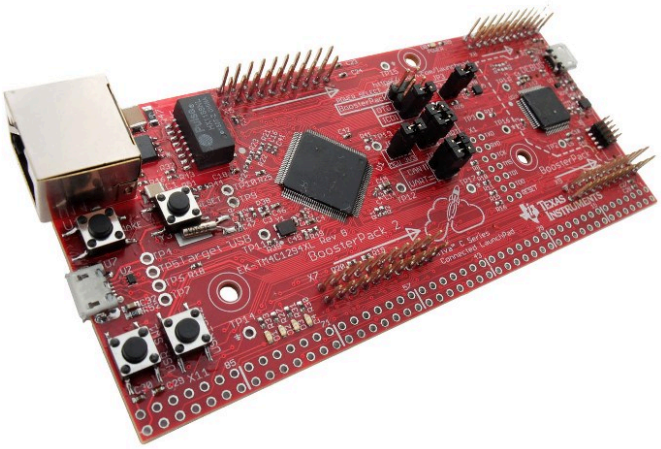
Para se comunicar com o módulo *Wi-Fi* e controlar os periféricos do sistema, escolheu-se um *kit* de desenvolvimento já utilizado anteriormente nas disciplinas de Microcontroladores e Sistemas Embarcados e de cuja utilização já era conhecida, denominado EK-TM4C1294XL. Entre suas principais funcionalidades encontram-se:

- Microcontrolador de alta performance TM4C1294NCPDT;
- CPU de 120MHz e 32-bits ARM Cortex-M4;
- Memória *Flash* de 1MB, 256kB de memória SRAM, 6kB de EEPROM (*Electrically Erasable Programmable Read-Only Memory*);

- Módulo integrado *Ethernet*;
- Saída de conexões para *kit BoosterPack* MKII;
- Dois módulos *Controller Area Network* (CAN);
- Circuito ICDI (*In Circuit Debug Interface*) integrado;
- Suporte para múltiplos ambientes de desenvolvimento, tais como CCS, Keil, IAR (*Ingenjörsfirman Anders Rundgren*) e GCC (*GNU Compiler Collection*);
- Vários exemplos de aplicações providos pela biblioteca *Tivaware Software Development Kit* (SDK).

Será mostrado adiante que o *kit* em questão foi considerado superdimensionado para as aplicações deste projeto visto que possui diversas funcionalidades e *GPIOs* de 3,3V que não foram utilizadas no processo de desenvolvimento do mesmo, porém seu uso se justifica devido à posse e experiência já adquirida em seu uso.

Figura 4 – *Kit* EK-TM4C1294XL



Fonte: (TEXASINSTRUMENTS, 2022b).

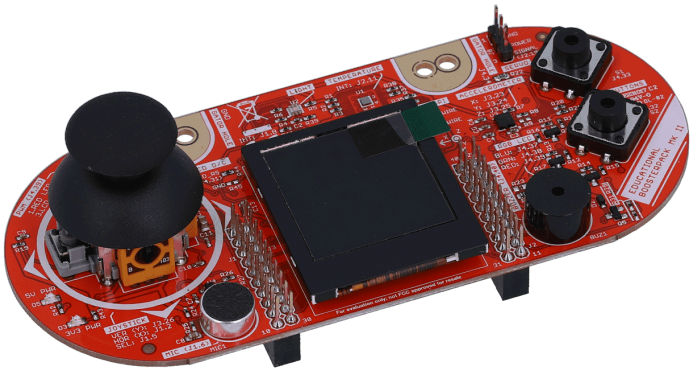
3.2.3 MÓDULO BOOSTXL-EDUMKII

Como um dos modos de aumentar a interação do usuário com a fechadura eletrônica, foi decidido utilizar um *display Liquid Crystal Display* (LCD) que pudesse exibir informações como listas de opções, ocorrência de eventos e inserção de eventuais dados do usuário quando fossem necessários. Para acelerar o desenvolvimento do projeto e não ser necessário desenvolver um circuito de controle, alimentação e roteamento de dados para o mostrador, foi utilizado um *kit* da *Texas Instruments* que, além de possuir o dispositivo com todo circuito necessário ao seu

uso, dispõe de diversas outras funcionalidades e periféricos que podem ser úteis ao projeto. Uma delas é o *buzzer* embutido que foi configurado para acionar enquanto a trava solenoide estivesse operando (fechadura aberta). O *kit* em questão é o *BOOSTXL-EDUMKII Educational BoosterPack MKII*. Algumas de suas funcionalidades se encontram abaixo:

- Sensor de luz TI OPT3001;
- Conector para servomotor;
- Acelerômetro de 3 eixos;
- Chaves tácteis de usuário;
- LED (*Light Emitter Diode*) multi-cor *Red-Green-Blue* (RGB);
- *Buzzer*;
- Conector empilhável de 40 pinos para *BoosterPack*;
- *Display* colorido *Thin-Film Transistor* (TFT) LCD;
- Microfone;
- Controlador *Joystick* de dois eixos com botões.

Figura 5 – *Kit* BOOSTXL-EDUMKII



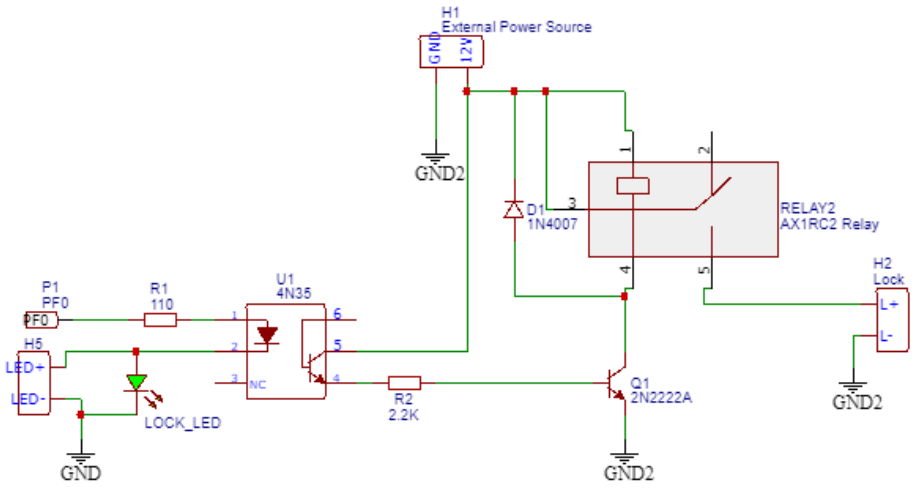
Fonte: (TEXASINSTRUMENTS, 2022a).

3.2.4 MÓDULO AUXILIAR DE CHAVEAMENTO

Como mostrado anteriormente na seção 3.2.2, a saída digital do microcontrolador EK-TM4C1294XL apresenta uma tensão contínua de 3,3V que não é suficiente para chavear a trava solenoide de 12V que foi escolhida para desenvolvimento do projeto. Portanto, foi necessário desenvolver um circuito feito, em sua maioria, para utilizar a saída digital somente como sinal de

controle e, assim, uma fonte externa servirá de alimentação para a trava. Para isso, foi utilizado um circuito de chaveamento com transistor de junção bipolar do tipo NPN 2N2222A (PHILIPS-SEMICONDUCTORS, 1997) em uma configuração de polarização fixa simples na qual a própria tensão de 12V de uma fonte de alimentação externa (que será descrita a seguir) foi utilizada para polarizar o transistor e, assim, habilitar a saída que chaveia a trava solenoide. Para não haver interação ou interferência do circuito alimentado pela fonte de alimentação externa com o circuito de saída digital do módulo de processamento central, estes mesmos foram isolados eletricamente por um circuito integrado do tipo optoacoplador 4N35 (PHILIPSSEMICONDUCTORS, 1995) cujo diodo emissor de luz localizado internamente na entrada do componente é acionado por uma das saídas digitais de 3,3V do EK-TM4C1294XL. Vale ressaltar que o valor de R1 foi arbitrariamente escolhido com um valor de resistência suficientemente baixo para limitar a corrente elétrica de saída de PF0 em situações nas quais tanto o LED interno do 4N35 quanto LOCK-LED (MULTICOMP, 2012) entram em condução para que, assim, a saída não fique conectada diretamente a GND.

Figura 6 – Esquemático do circuito de chaveamento do módulo auxiliar de chaveamento



Fonte: Autoria própria (2022).

3.2.5 CÁLCULOS DOS VALORES DE COMPONENTES

Para a simples função de atuar como um chaveador, o circuito deve controlar um transistor de modo que este componente comute entre os estados de corte e região ativa. Foi utilizada a seguinte lógica de calcular R2 partindo dos seguintes pressupostos:

- O modelo escolhido do transistor é 2N2222;
- O modelo do diodo de roda livre 1N4007 já é fixo;

- A junção Coletor-Emissor dada pelos pinos 4 e 5 do 4N35 entra em condução tendo uma queda de potencial próxima a 0V quando o LED interno (pinos 1 e 2) do mesmo componente é acionado.

Em estado de corte (quando PF0 é igual a 0V e, logo, LED de entrada de 4N35 não conduz corrente elétrica), toda a tensão de saída do módulo de alimentação externa (*Vh1*) repousa sobre os terminais 4 e 5 de 4N35, os mesmos ficam em estado aberto e, portanto, não há corrente sobre o terminal de base de 2N2222. Logo, o mesmo transistor não deixa corrente elétrica fluir entre seus terminais de coletor e emissor, toda a tensão de H1 repousa sobre os mesmos terminais e, sendo assim, não há tensão na bobina do relé AX1RC2 que chaveia a trava solenóide H2.

Já em estado ativo (quando PF0 é igual a 3,3V e, logo, LED de entrada de 4N35 conduz corrente elétrica, permitindo, assim, a passagem de corrente por seus teminais 4 e 5), pode-se escolher em quais condições o transistor deve operar. Para sobrar o máximo possível de tensão do módulo de alimentação externa sobre a bobina de AX1RC2 (METALTEX, 2022), foi escolhido trabalhar com 2N2222 quase em estado de saturação (onde Vce é igual a 1V). Para tal operação, o *datasheet* do transistor fornece a informação de que nessa condição a corrente de coletor é de 150mA e que o ganho de corrente contínua é 50. Deste modo, pode-se calcular qual a corrente de base do transistor por meio de (1):

$$Ib = \frac{Ic}{\beta}. \tag{1}$$

Substituindo os valores acima citados nos devidos campos de (1):

$$Ib = \frac{150.10^{-3}}{50} = 3mA. \tag{2}$$

A partir do valor encontrado para a corrente de base em (2) e considerando as quedas de tensão típicas de 0,7V para *Vbeq1* e 5V para V_{ceu1} dadas respectivamente pelos *datasheets* do 2N2222 e 4N35, pode-se utilizar (3) como base para o cálculo do resistor *R2*. (BOYLESTAD; Louis Nashelsky, 2002)

$$Ib = \frac{Vcc - Vbe}{Rb}. \tag{3}$$

Remodelando (3) para o cálculo de *R2* e considerando que, neste caso, *Rb* = *R2*, obtém-se:

$$R2 = \frac{Vh1 - Vceu1 - Vbeq1}{Ib}. \tag{4}$$

Substituindo os valores em (4) afim de encontrar o valor ideal para R2, obteve-se:

$$R2 = \frac{12 - 5 - 0.7}{3.10^{-3}} = 2{,}1k\Omega. \tag{5}$$

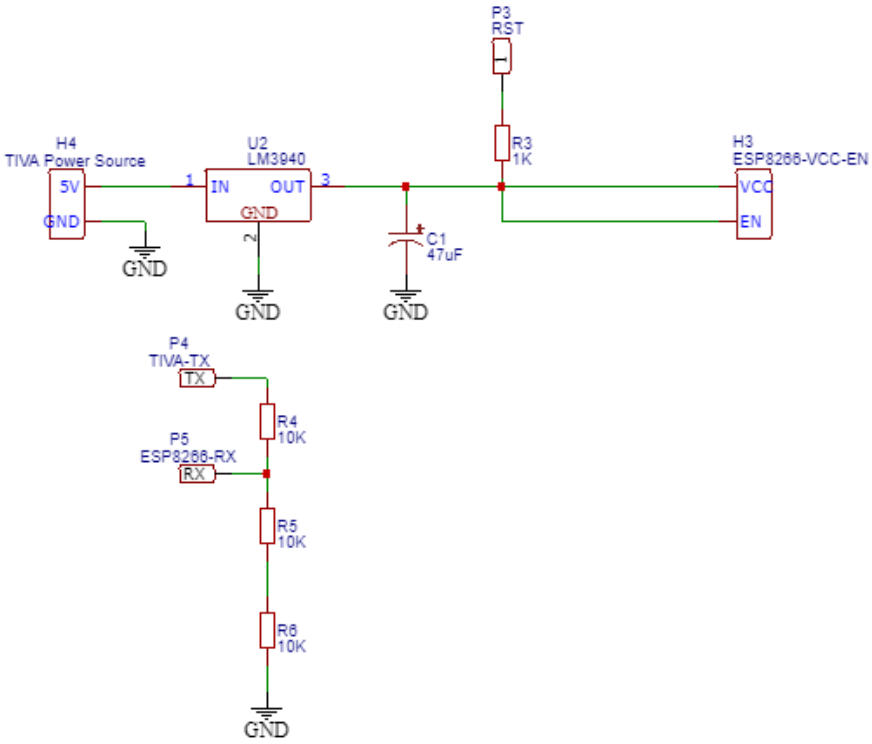
Como o valor ideal de resistência calculado para R2 em (5) não coincide com um valor comercial facilmente obtível, este valor foi alterado para o valor comercial mais próximo. Logo:

$$R2 = 2{,}2k\Omega. \tag{6}$$

3.2.6 CIRCUITOS DE ESTABILIZAÇÃO E DIVISOR RESISTIVO

O módulo auxiliar de chaveamento comporta ainda mais dois pequenos circuitos: um de estabilização da saída de tensão contínua de 5V do módulo de processamento central e outro de divisor resistivo para alinhar a tensão de saída do pino configurado como Tx do *kit* EK-TM4C1294XL com a tensão nominal de entrada do pino Rx do módulo *Wi-Fi*. O primeiro dos circuitos acima citados foi projetado devido à sobrecarga que o módulo central estava sofrendo pois, após a inclusão do módulo BOOSTXL-EDUMKII, a placa *Wi-Fi* começou a reiniciar por falta de tensão em seu pino de alimentação. Foi utilizado um LM3940 (BAYLINEAR, 2022) para estabilizar a tensão de saída do microncontrolador de 5V para 3,3V. Analisando o *datasheet* deste componente, pode-se decidir um capacitor de saída de 47uF pois este valor já ultrapassa o valor mínimo indicado pelo fabricante de 33uF cuja capacitância já é considerada suficiente para estabilidade da saída e resposta de transiente. Após a inclusão deste circuito, o mesmo problema não ocorreu.

Figura 7 – Esquemático do circuito de estabilização e divisor resistivo do módulo auxiliar de chaveamento



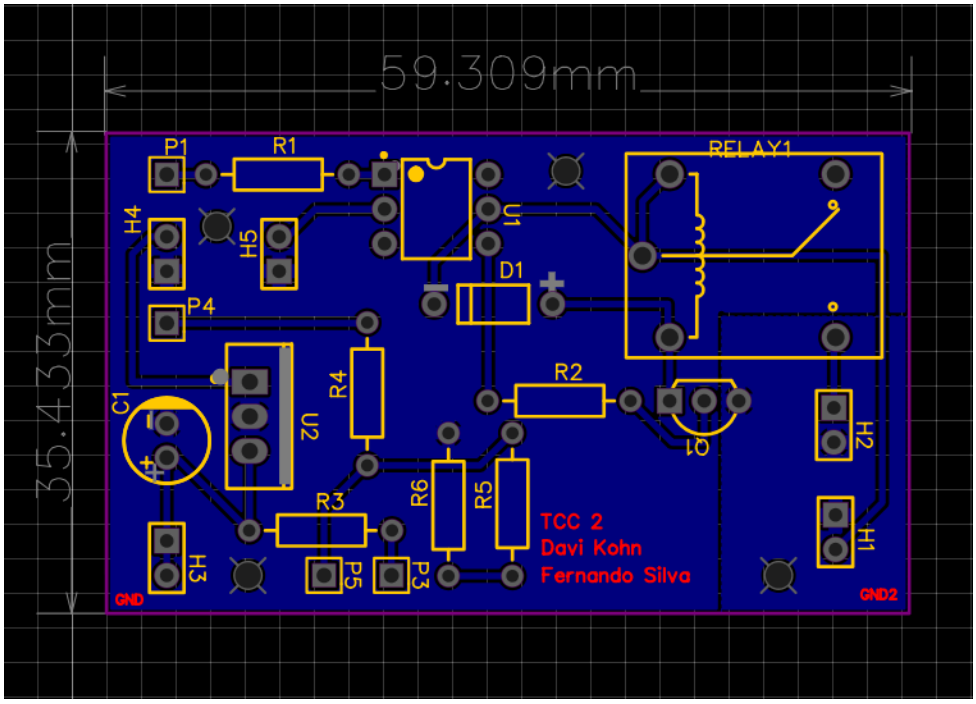
Fonte: Autoria própria (2022).

Após a escolha dos componentes para os devidos circuitos e o teste de funcionamento ainda em *protoboard*, verificou-se que tanto o circuito de chaveamento acionava a trava solenoide corretamente quanto o circuito de estabilização realmente garantiu que não houvesse oscilação na alimentação contínua do módulo ESP8266. Dito isso, foi dado início ao desenvolvimento da placa de circuito impresso deste módulo, cuja escolha de tecnologia de escapsulamento de componentes foi a *Through-hole-technology* (THT) pois a natureza e a complexidade dos circuitos a serem embarcados não geravam necessidade de se minimizar o dimensionamento da placa tanto como organizar os componentes em uma disposição *dual-layer*. Sendo assim, os componentes mostrados na Figura 7 foram projetados para caber em uma face simples de placa de circuito impresso (*single-layer*) de 1,6mm de espessura e com trilhas de cobre de 0,5mm, desde que suas posições satisfizessem alguns pré-requisitos:

- Os conectores de cabos e fios de comunicação com outras placas estivessem localizados nas bordas da placa;
- Houvesse espaço suficiente para abertura de 4 buracos para fixação na mecânica da fechadura eletrônica;
- Houvesse espaço para serigrafia básica de identificação da placa;
- Componentes organizados de modo a separar, com facilidade, os dois *pads* de GND e GND2 do esquemático do módulo auxiliar de chaveamento.

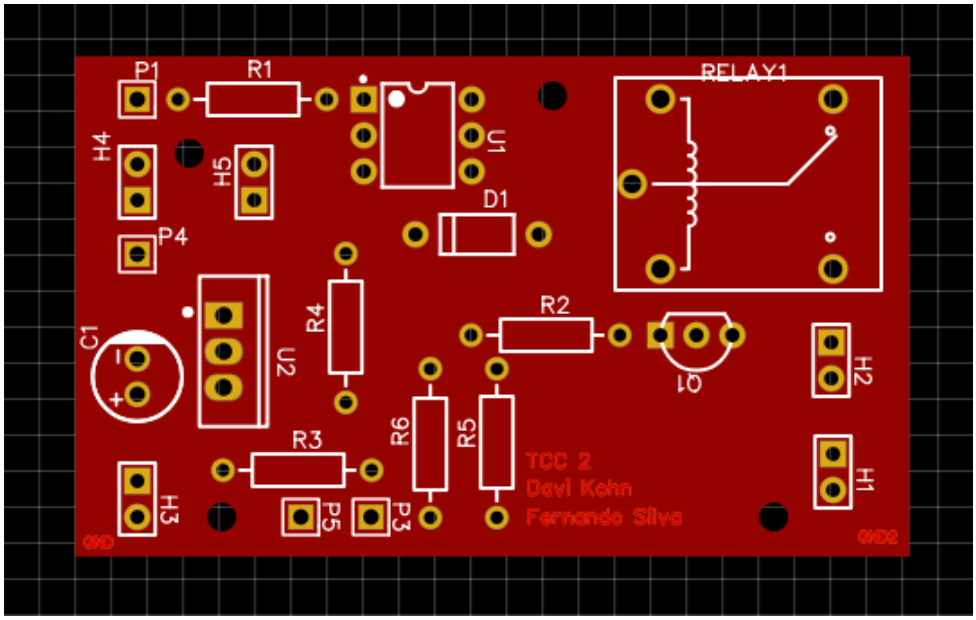
Levando em consideração estes itens acima citados, a versão final de *layout* da placa resultou no desenho da Figura 8, na qual os detalhes de cor amarela descrevem os limites que os componentes ocupam na placa, os detalhes na cor vermelha são de serigrafia onde são disponibilizadas informações como o nome da disciplina para a qual o projeto foi desenvolvido, nomes dos integrantes do grupo e descrição de onde estão localizados cada referência de tensão dos dois circuitos isolados. Por fim, os desenhos na cor azul ilustram toda a área de cobre que foi poupada da corrosão no processo de fabricação da placa e que estão localizados na face inferior da mesma.

Figura 8 – Projeto da placa do módulo auxiliar de chavemento



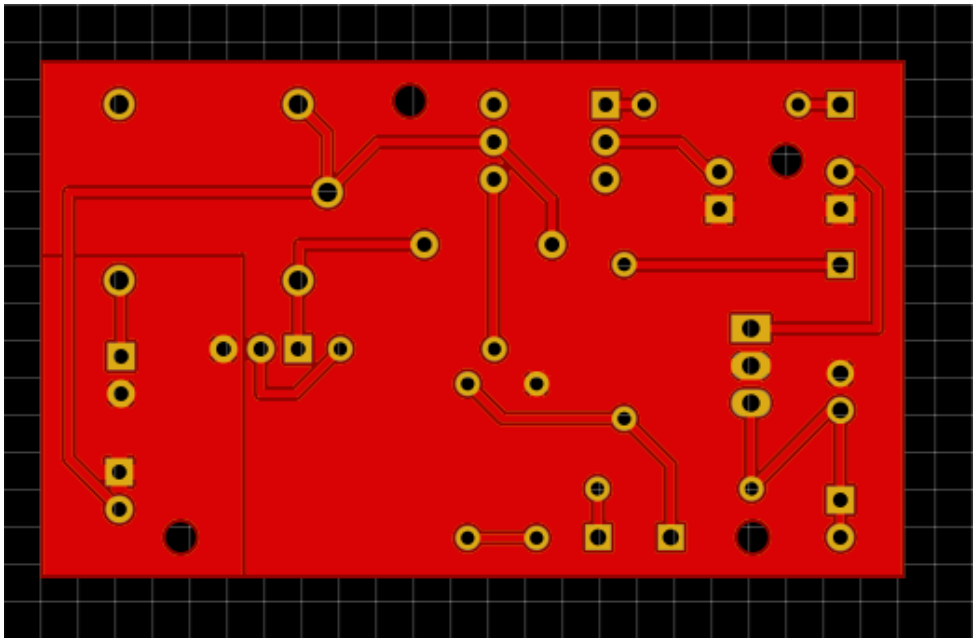
Fonte: Autoria própria (2022).

Figura 9 – Simulação da placa do módulo auxiliar de chavemento - Face superior



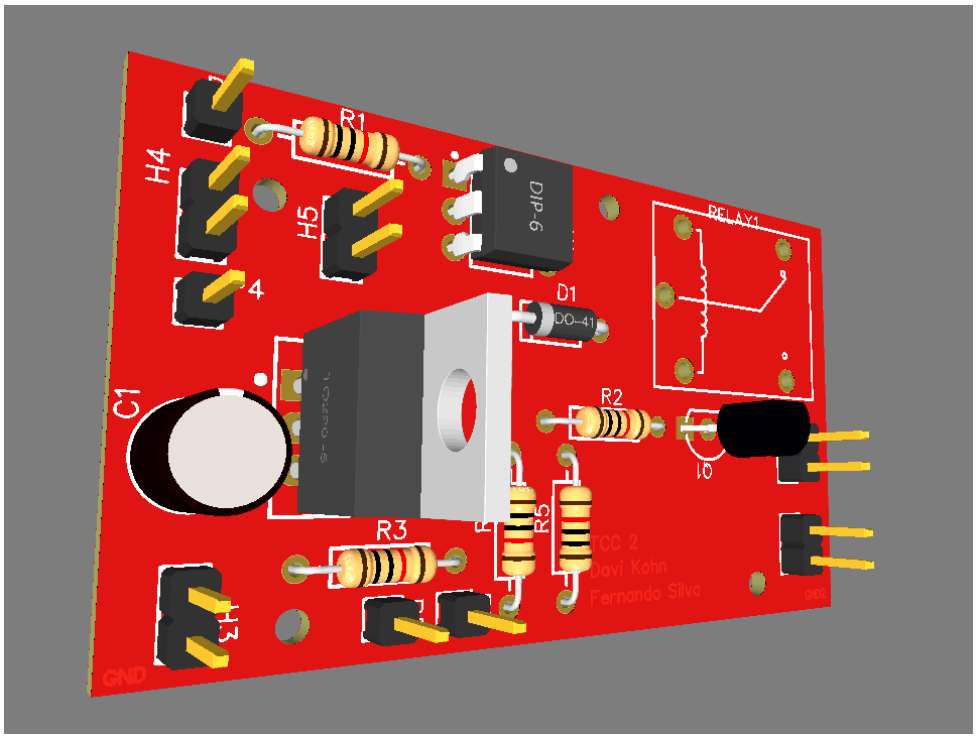
Fonte: Autoria própria (2022).

Figura 10 – Simulação da placa do módulo auxiliar de chavemento - Face inferior



Fonte: Autoria própria (2022).

Figura 11 – Simulação da placa montada do módulo auxiliar de chavemento em 3 dimensões

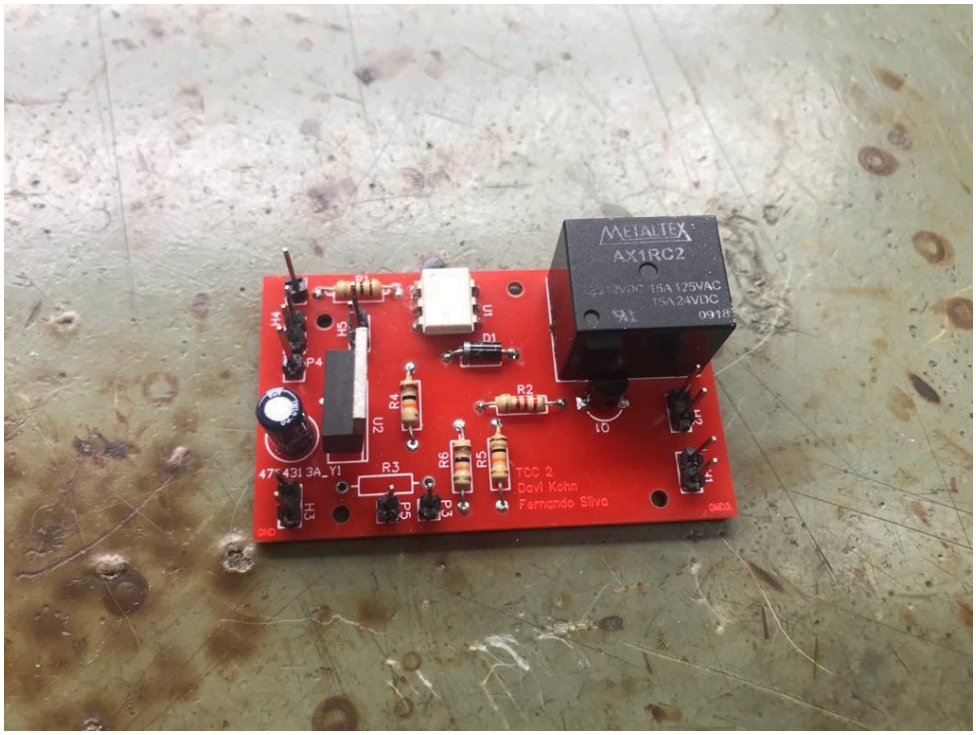


Fonte: Autoria própria (2022).

Dado o desenvolvimento do *layout* mostrado na Figura 8 e Figura 9, foi feito o orçamento e encomenda de cinco peças (quantidade mínima), sem optar pela inclusão e soldagem dos componentes na placa, pela empresa JLCPCB (2022). Com a chegada das mesmas peças,

inserção e soldagem dos componentes em uma delas, o resultado final ficou tal qual mostrado na imagem da Figura 12.

Figura 12 – Módulo auxiliar de chaveamento montado

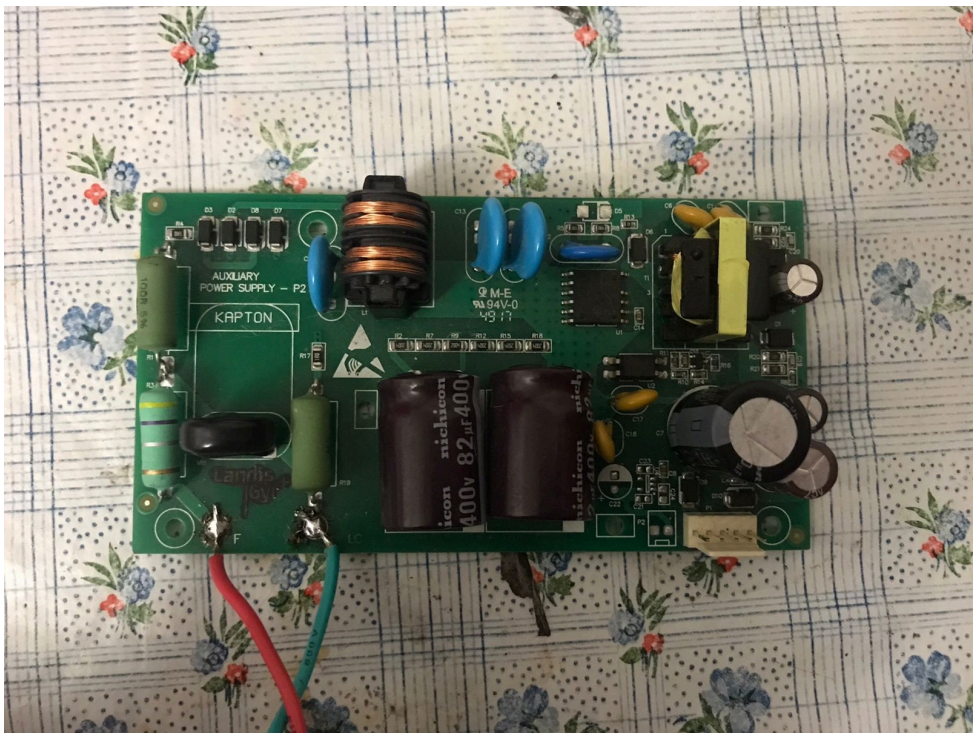


Fonte: Autoria própria (2022).

3.2.7 MÓDULO DE ALIMENTAÇÃO EXTERNA

Com a principal justificativa de chavear a trava solenoide de 12V, foi necessário obter uma fonte de alimentação que cedesse tal tensão elétrica tanto como não sobrecarregasse o módulo central de processamento, o qual já havia dado indícios de sobrecarga após a inserção do módulo BOOSTXL-EDUMKII no projeto. Com isso, foi obtido uma placa de alimentação *Direct Current* (DC) da empresa Landis+Gyr (2022) a qual foi cedida pela própria empresa por ser produto de descarte. O módulo em si possui entrada para rede 127/220V *Root Mean Square* (RMS) e quatro saídas de tensão contínua de 12, 9, -9 e -12V, portanto a primeira destas foi a única saída utilizada no projeto. A existência de uma fonte externa para um projeto no qual se pode adicionar variados periféricos (seja uma câmera ou módulo de biometria, por exemplo) é interessante pois assim ela consegue suprir a energia necessária para alimentá-los em situações nas quais nem sempre o próprio *kit* microcontrolador consegue prover.

Figura 13 – Placa do módulo de alimentação externa



Fonte: Autoria própria (2022).

3.3 DESENVOLVIMENTO DO *FIRMWARE*

Primeiramente, há de se elucidar quais módulos dos quais compõem o *hardware* possuem *firmware* embarcado. São eles o módulo central de processamento (EK-TM4C1294XL), módulo BOOSTXL-EDUMKII e o módulo *Wi-Fi* (ESP8266 ESP-01). Destes acima citados, foi escolhido manter o código-fonte original de fábrica de seu respectivo fabricante apenas no segundo dispositivo.

3.3.1 *FIRMWARE* DO MÓDULO DE COMUNICAÇÃO *WI-FI*

Para o módulo Wi-Fi, foi escolhido fazer atualização de seu *firmware* para garantir que, além do código-fonte embarcado fosse o último lançado pelo seu fabricante *Espressif*, a confiabilidade da comunicação e a variedade de comandos fossem a maior possível visto que a tendência, ao longo das atualizações, é da melhoria da qualidade do uso do dispositivo. Pàra isso, foi utilizado a última versão de *firmware* sem sistema operacional embarcado existente no momento do desenvolvimento do projeto cujo nome é "ESP8266-NonOS-AT-V1.7.4"e de cujas partições do título significam:

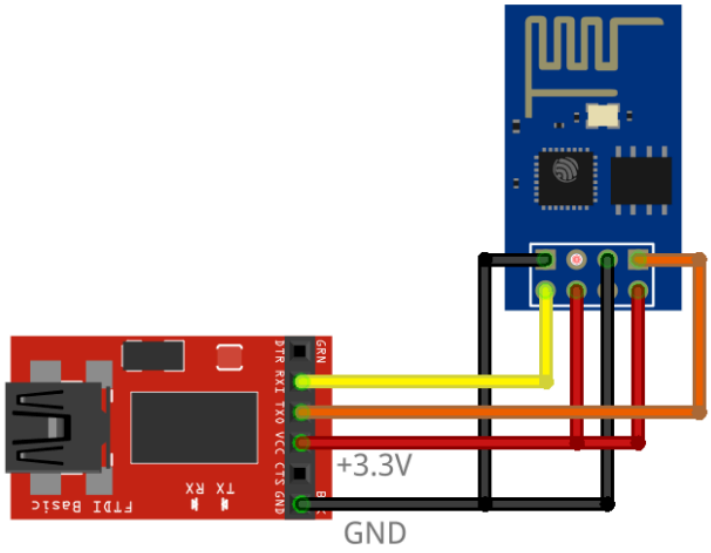
- ESP8266: Nome do dispositivo a embarcar o código do *firmware*;

- NonOS (*Non Operating System*): Versão de *firmware* sem recursos de sistemas operacionais em tempo real;
- AT (*Attention*): Padrão de comandos de rede do tipo "Attention";
- V1.7.4: Identificador númerico da versão do *firmware*.

3.3.2 PROCESSO DE ATUALIZAÇÃO DE *FIRMWARE* DO MÓDULO DE COMUNICAÇÃO *WI-FI*

A atualização da versão de *firmware* do módulo *Wi-Fi* não pode ser considerada uma tarefa trivial, porém teve de ser executada de modo a garantir uma padronização no uso dos comandos de rede do dispositivo. Tal processo consiste na conexão física dos terminais do ESP8266 a um conversor USB-Serial para que o código binário do *firmware* novo seja transferido ao módulo por meio do computador. A conexão pode ser visualizada na Figura 14:

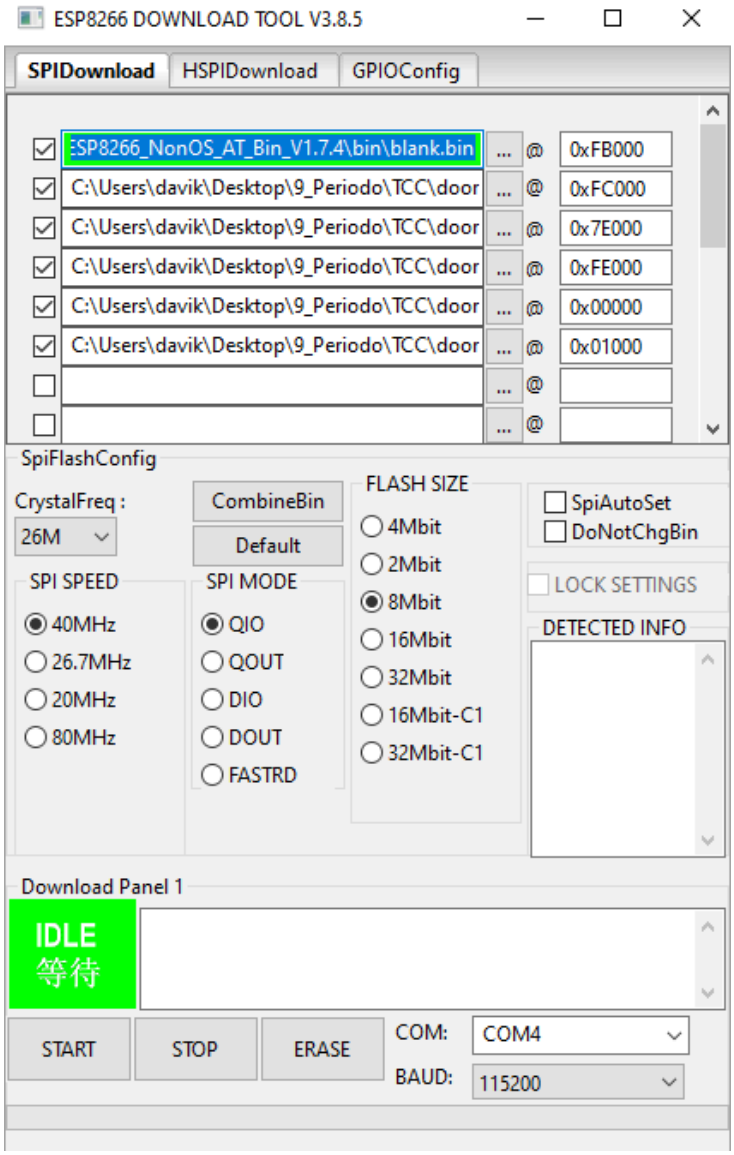
Figura 14 – Conexão do módulo *Wi-Fi* com o conversor USB-Serial



Fonte: (ALASDAIRALLAN, 2022).

O *software* utilizado para transferência da imagem do *firmware* para o módulo em questão é chamado de "ESP8266 DOWNLOAD TOOL V3.8.5", programa desenvolvido e distribuído pelo próprio fabricante *Espressif* e de cuja interface do usuário se mostra como na Figura 15:

Figura 15 – Interface do usuário do programa ESP8266 DOWNLOAD TOOL V3.8.5



Fonte: Autoria própria (2022).

Antes de realmente fazer esta transferência, há de se identificar qual a capacidade da memória *flash* do ESP8266 pois existem algumas variantes deste módulo produzidas mas difundidas com o mesmo nome. Após este processo de identificação, que não será abordado deste documento por estar fora do escopo, e selecionar a opção que corresponde ao tamanho da memória em questão, foi selecionada uma série de arquivos binários componentes do *firmware* que são apontados para diversos endereços específicos da memória *flash* do módulo e, finalmente, é feita a transferência do código embarcado para o ESP8266.

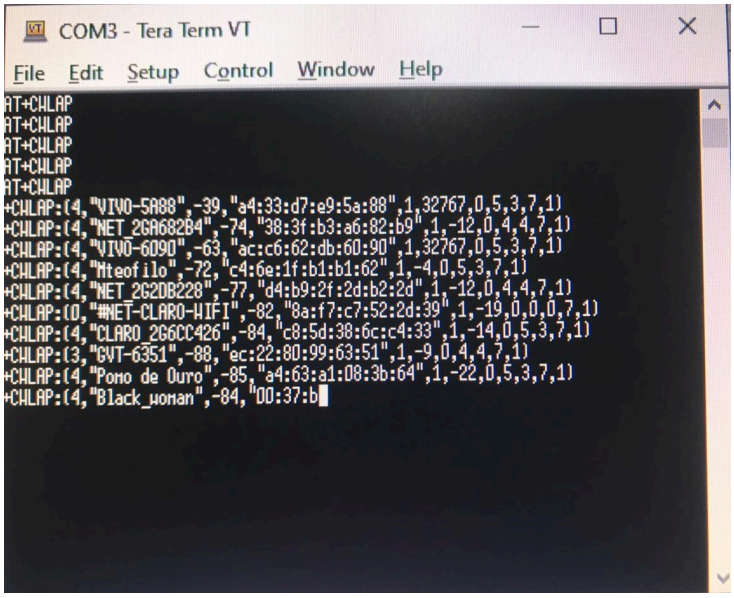
3.3.3 *FIRMWARE* DO MÓDULO DE CENTRAL DE PROCESSAMENTO

De todos os processos de implementação deste projeto, o desenvolvimento do código embarcado do módulo central de processamento foi o que exigiu mais tempo pois quaisquer

equívocos, tanto na fase de planejamento quanto escrita, poderiam causar um retrabalho considerável para resolvê-los.

Antes de planejar a arquitetura do código, foi feito um teste simples de comunicação serial via protocolo UART entre módulo central de processamento e módulo *Wi-Fi* para, antes de quaisquer outros motivos, garantir que a última versão de *firmware* do módulo ESP8266 tivesse sido realmente transferida e também que o próprio módulo reconhecesse sua versão. Com este intuito, foi construído um código-fonte básico procedural na linguagem C que estabelecia duas conexões, uma para comunicação entre módulo central e módulo *Wi-Fi* e outro para comunicação entre módulo central e computador. Este último serviria com uma espécie de monitor para a primeira conexão citada, no qual toda comunicação entre os módulos seria exibida na tela do computador via terminal para facilitar a implementação do desenvolvedor. Um exemplo da informação sendo mostrada neste terminal está na Figura 16 e o *software* utilizado como monitor da comunicação serial entre computador e módulo central é o Tera Term.

Figura 16 – Monitor utilizado para depurar comunicação entre computador e módulo central



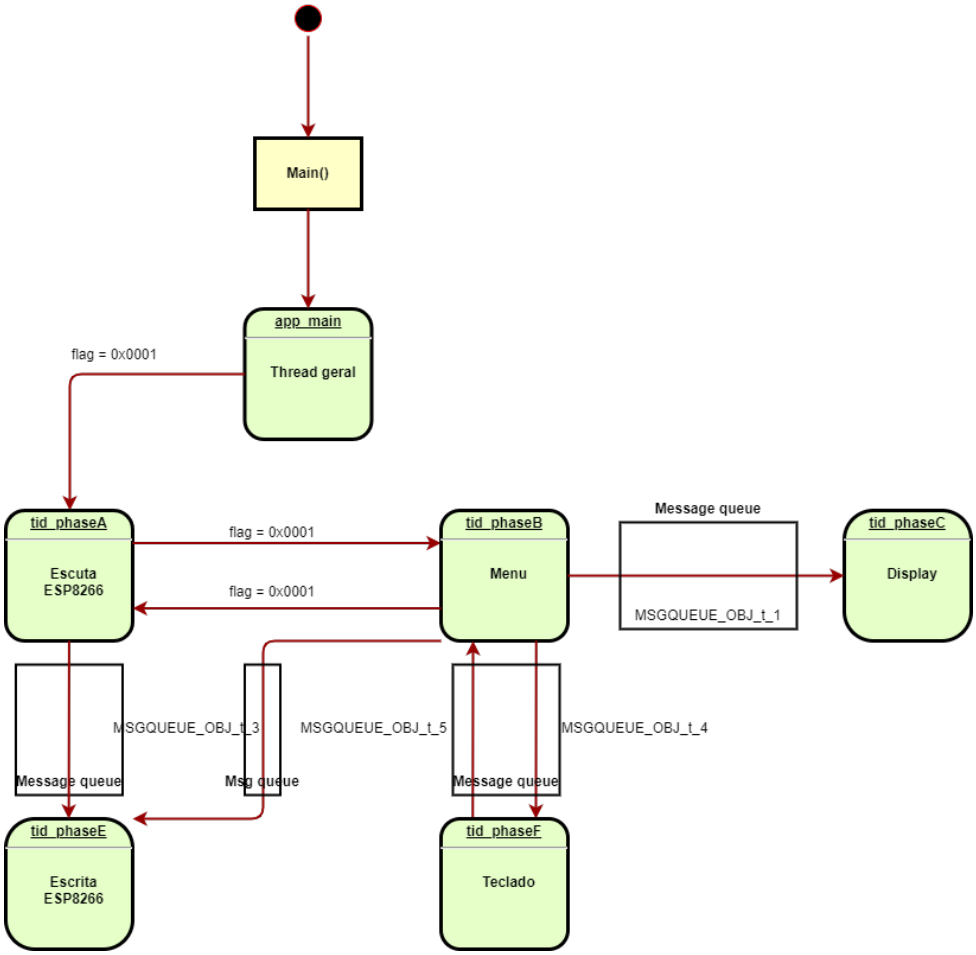
Fonte: Autoria própria (2022).

Após a verificação de que os comandos de rede AT estavam sendo identificados e executados corretamente pelo módulo ESP8266, partiu-se para a etapa de planejamento de como a versão de *firmware* final funcionaria tendo em vista que o uso do *software* Tera Term continuaria sendo utilizado para envio de comandos e monitoramento da comunicação até que fosse inserido um teclado no *hardware* do projeto para interação física com o usuário.

O módulo EK-TM4C1294XL possui suporte para implementação de código embarcado com *Real-Time Operating System* (RTOS) e, sabendo que o projeto possui vários módulos e periféricos a serem controlados de forma eficiente, foi escolhido o CMSIS-RTOS API v2 para administrar os recursos do processador TM4C1294NCPDT embarcado no *kit*. Tendo como base

estas informações, foi planejado o funcionamento do *firmware* baseado no diagrama de classes ilustrado abaixo.

Figura 17 – Diagrama de classes do *firmware* do módulo central



Fonte: Autoria própria (2022).

A partir da imagem acima, pode-se explicar o que cada componente faz concretamente logo após a implementação do código propriamente dito.

- **Main**: Primeira função do código a ser chamada. Inicializa todos os módulos e os configura antes de sua utilização. Colocando em sequência as chamadas, inicializa o *display* LCD do módulo *BoosterPack* MKII e escreve a tela inicial; configura pinos e informações da comunicação UART com o módulo ESP8266 ESP-01; configura GPIOs dos periféricos, sendo eles os pinos de entrada e saída do teclado matricial e a saída digital de controle da trava solenoide; chama função de biblioteca nativa do *BoosterPack* MKII que inicializa o *buzzer* do *kit*; envia os comandos AT iniciais que configuram o modo de funcionamento do módulo *Wi-Fi*, conectam o módulo à rede desejada e o conectam ao servidor do projeto via conexão TCP; configura as credenciais do usuário; estabelece o método de criptografia de mensagens que será utilizado; inicializa

o Kernel do CMSIS RTOS v2 que é responsável, entre outras funções, por chamar a primeira *thread* do código, a *app-main*.

- **app-main**: *Thread* de configuração do *firmware*, tem como função principal criar e configurar todos os recursos do CMSIS RTOS v2 como *threads*, filas de mensagens (*message queues*), *timers*, *mutex* e chamar a *thread tid-phaseA*.

- **tid-phaseA**: *Thread* responsável por verificar se existe mensagem armazenada no *buffer* reservado para dados vindos do módulo ESP8266 ESP-01 e, se caso a mensagem for um comando válido do servidor para operar a trava, chama a *thread tid-phaseE* via fila de mensagens que a operação. Se não houver nada no *buffer*, a *thread* habilita uma flag para o *Kernel* executar a *tid-phaseB* enquanto mantém o estado da *tid-phaseA* como *blocked*.

- **tid-phaseB**: *Thread* responsável pela lógica de programação do menu e gerenciamento da comunicação por meio de filas de mensagens com as outras *threads* que se relacionam com o menu. Uma explicação mais detalhada de como a lógica desta *thread* funciona será passada a seguir.

- **tid-phaseC**: Responsável por esperar a mensagem da *thread* tid-phaseB com dados como *payload* e posicionamento da mensagem na matriz do *display* para, assim, imprimir o *payload* no *display* LCD do *BoosterPack* MKII.

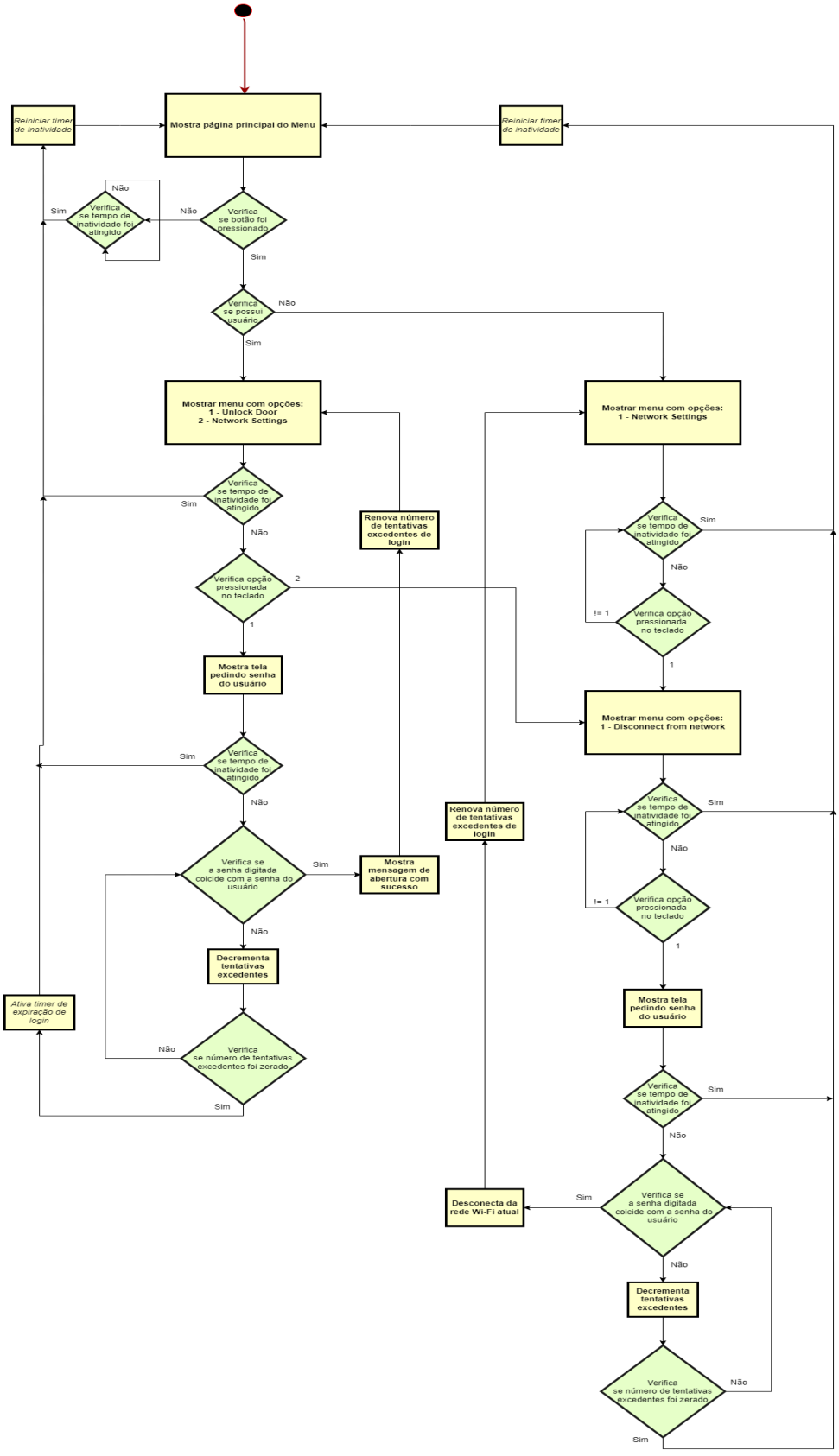
- **tid-phaseF**: *Thread* acionada pela *thread* tid-phaseB via fila de mensagens. Sua função é fazer a varredura do teclado matricial buscando pelo pressionamento de alguma tecla. Em caso positivo e se a *thread* tid-phaseB esperar mais de um dígito, a *thread* entra em *loop* e só retorna quando o número máximo de dígitos for atingido ou quando a tecla '*' for pressionada. Independentemente se houver somente um caractere, vários caracteres ou mesmo nenhum digitado, a *thread* tid-phaseF retorna o valor para a *thread* tid-phaseB via fila de mensagens.

- **tid-phaseE**: *Thread* responsável por acionar a trava solenoide e enviar a mensagem de abertura ao servidor via TCP ao mesmo tempo pois os dois eventos necessitam de ocorrer juntamente. Ela pode ser chamada somente pelas *threads* tid-phaseA e tid-phaseB (Escuta ESP8266 e Menu) via fila de mensagens e, neste caso, não retorna valor nenhum para estas pois parte-se do pressuposto que, a partir do acionamento da saída digital a responsabilidade do acionamento da trava é exclusiva do *hardware* que compõe o módulo citado na seção Módulo Auxiliar de Chaveamento.

3.3.4 FLUXO DE OPERAÇÃO DA *THREAD* TID-PHASEB (MENU)

A lógica por trás do funcionamento da *thread* tid-phaseB responsável pelo menu do usuário merece uma seção isolada pois houve todo um planejamento teórico para que ela funcionasse sem nenhum *deadlock*. Neste caso, foi projetado um diagrama de estados somente para o fluxo de telas e interação do menu com o usuário.

Figura 18 – Fluxograma do comportamento do menu (*thread* tid-phaseB)



Fonte: Autoria própria (2022).

3.3.5 CRIPTOGRAFIA *Secure Hash Algorithm 1* (SHA-1)

A fim de garantir segurança adicional na troca de mensagens entre fechadura eletrônica e servidor, foi estabelecido um algoritmo de criptografia imediatamente anterior ao envio de mensagens propriamente dito. Ao invés de implementar o algoritmo de algum método de criptografia desde o seu início, optou-se por utilizar um dos métodos de envelopamento de mensagens já implementados na biblioteca oficial TivaWare que já suporta os algoritmos MD5, SHA-1, SHA-224 e SHA-256. Para isso, escolheu-se o algoritmo SHA-1 que camufla a mensagem original em 160 *bits* antes de ser colocada para envio ao servidor via *thread* tid-phaseE. O mesmo ocorre logo após a recepção da mensagem de comando do servidor (que já chega criptografada com o mesmo algoritmo), a qual é descriptografada utilizando funções de descriptografia disponíveis na mesma biblioteca acima citada imediatamente anterior à leitura de seu *payload*.

3.4 DESENVOLVIMENTO DA MECÂNICA DA FECHADURA

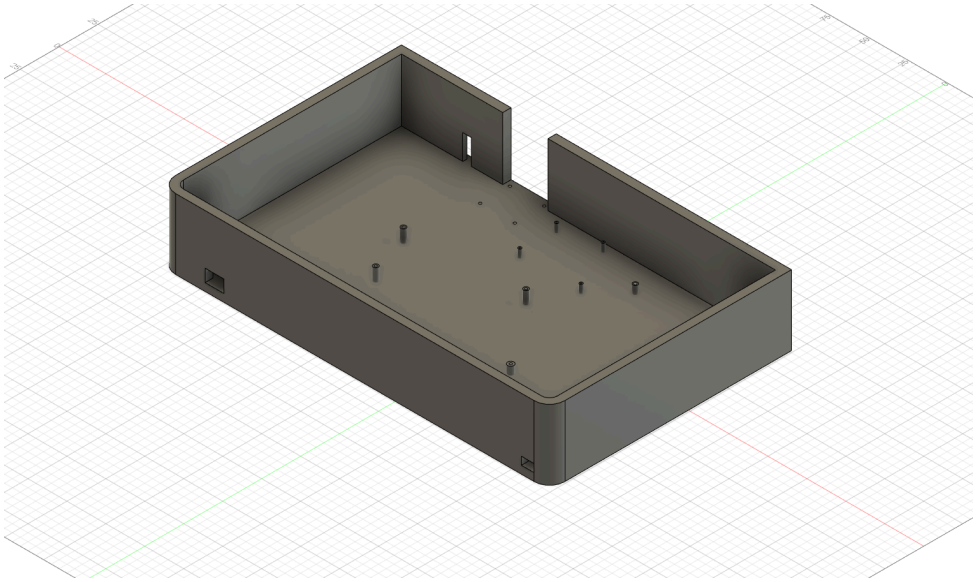
Para comportar fisicamente todos os módulos citados na seção Definição dos módulos componentes do *hardware* foi dado início ao planejamento dos pré-requisitos e desenvolvimento do formato, dimensões, encaixes e aberturas de uma mecânica plástica. Tal mecânica deveria ser projetada de modo a satisfazer algumas condições, são elas:

- Deve ser dividida em duas partes: base e tampa;
- Comportar todos os módulos, placas e periféricos descritos na seção 3.2;
- Possuir abertura frontal de mesmas dimensões das do *display* do módulo Boosterpack MKII para que o usuário tenha contato visual com o mesmo;
- O módulo *Wi-Fi* deve, de alguma maneira, estar localizado na parte externa da mecânica de modo que a face de sua *Peripheral Component Interconnect* (PCI) exiba sua antena para fora da fechadura e seus pinos para a parte interna;
- Deve possuir duas aberturas para passagem do cabo USB do módulo central de processamento e dos cabos de alimentação 127/220V RMS do módulo de alimentação externa;
- Deve possuir abertura circular para posicionamento do LED LOCK-LED ilustrado na Figura 6, o qual será ativado no momento em que a trava operar;
- A trava solenoide deverá ser posicionada com sua trava para a parte externa da mecânica em uma das faces laterais;

- A face lateral na qual a trava solenoide será posicionada deve ser totalmente plana e suas arestas e vértices não devem ser arredondados;
- A base da mecânica deve possuir furos para parafusos de cujas posições devem corresponder aos furos das placas dos módulos citados na seção 3.2.

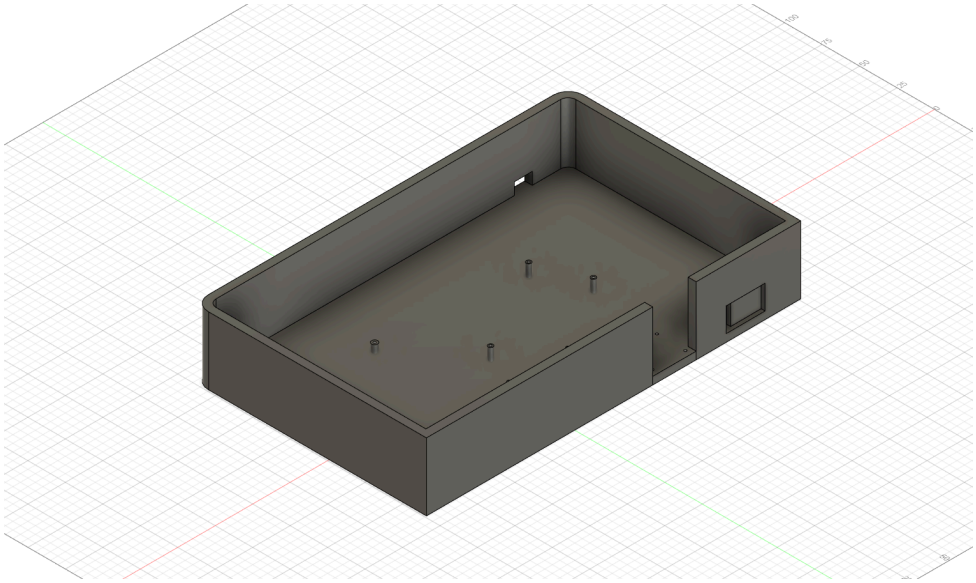
Tendo como base estes estas premissas, foi desenvolvido com a ajuda do *software* Autodesk Fusion 360 uma mecânica que se assemelha a uma fechadura eletrônica eletrônica convencional. As Figuras 19 a 28 ilustram a versão final da mesma, assim como o rascunho de todos os módulos que compõem o projeto.

Figura 19 – Base da mecânica - Vista diagonal 1



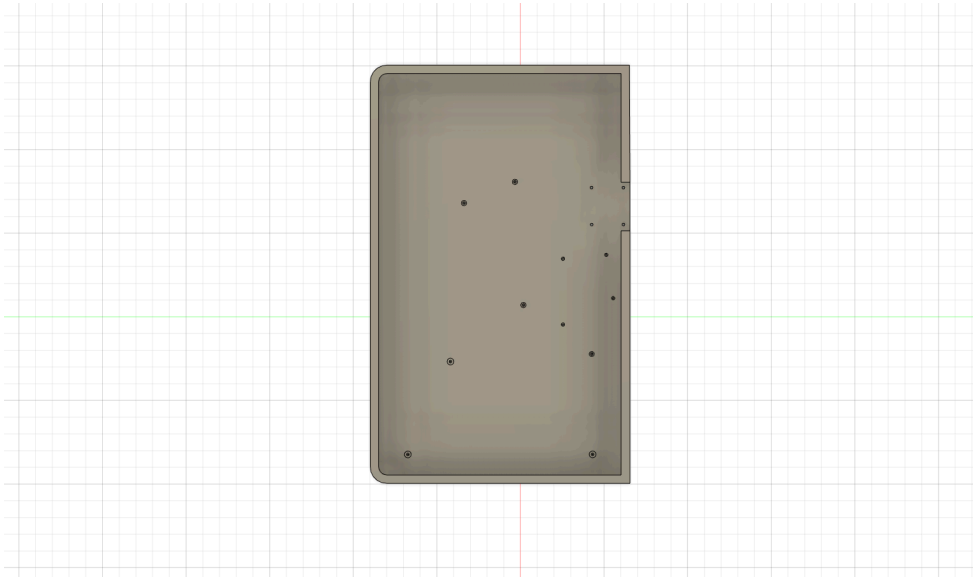
Fonte: Autoria própria (2022).

Figura 20 – Base da mecânica - Vista diagonal 2



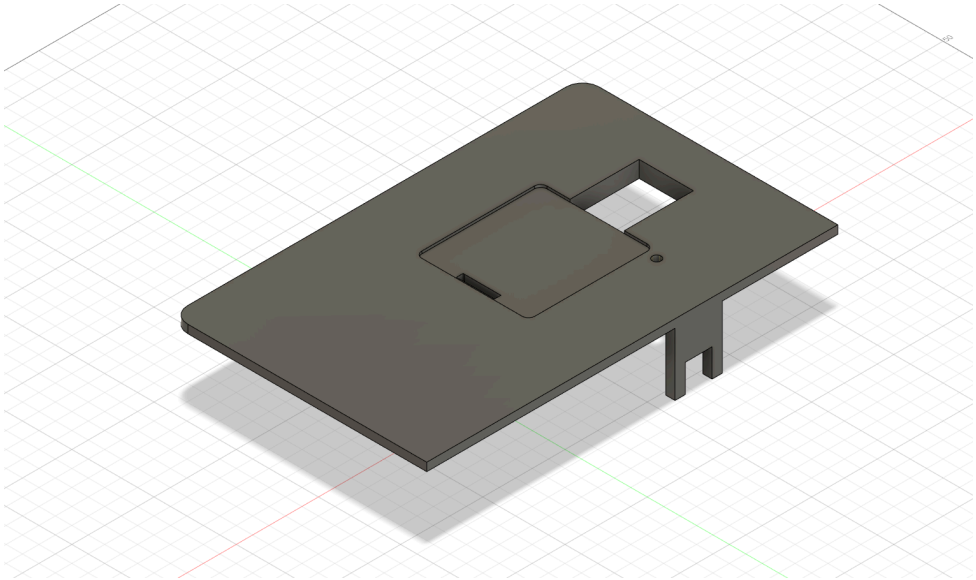
Fonte: Autoria própria (2022).

Figura 21 – Base da mecânica - Vista superior



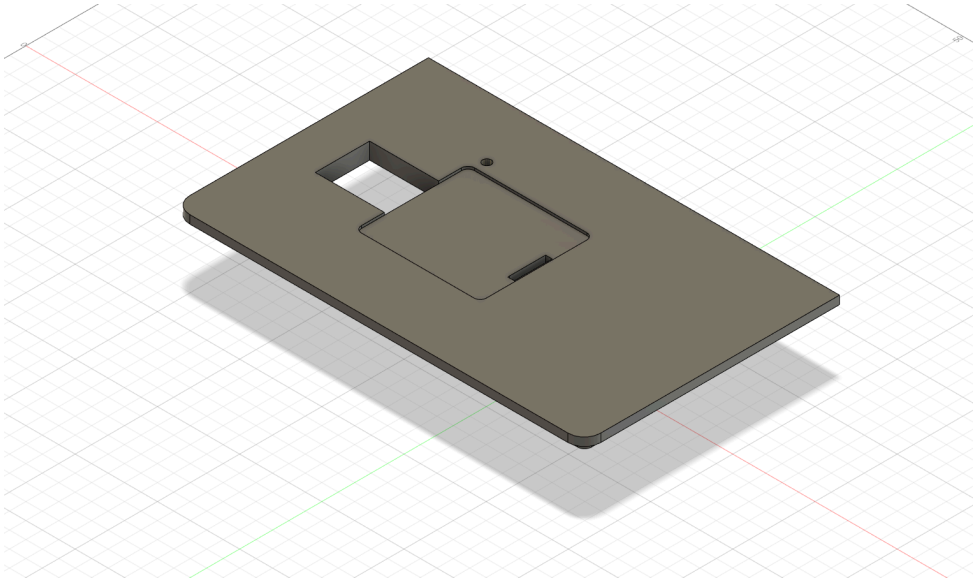
Fonte: Autoria própria (2022).

Figura 22 – Tampa da mecânica - Vista diagonal 1



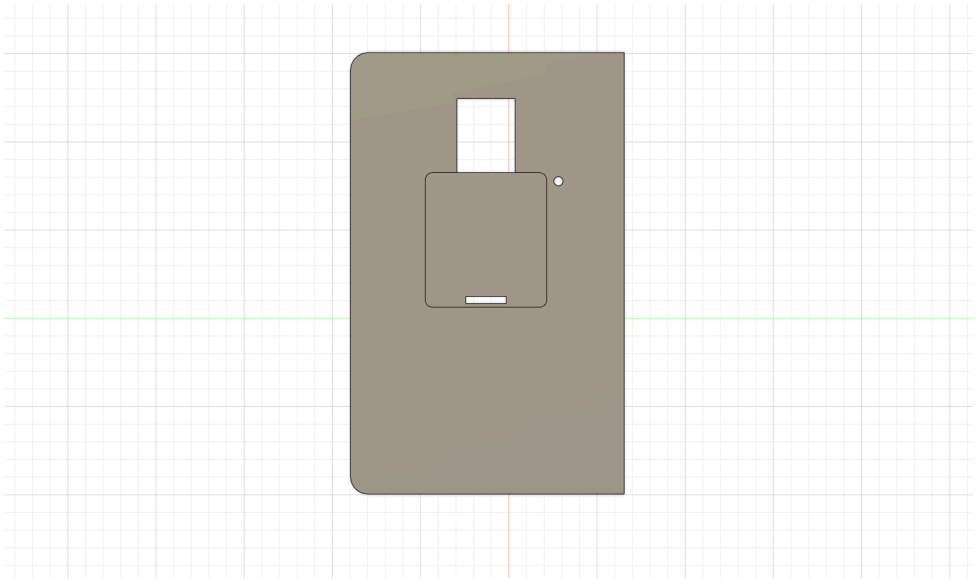
Fonte: Autoria própria (2022).

Figura 23 – Tampa da mecânica - Vista diagonal 2



Fonte: Autoria própria (2022).

Figura 24 – Tampa da mecânica - Vista superior

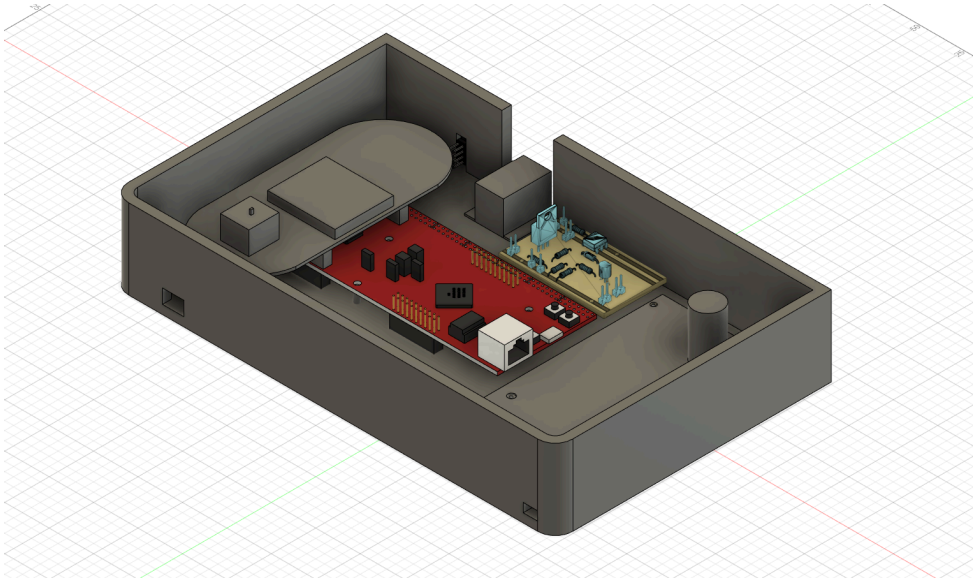


Fonte: Autoria própria (2022).

Figura 25 – Mecânica com módulos sem tampa - Vista diagonal 1

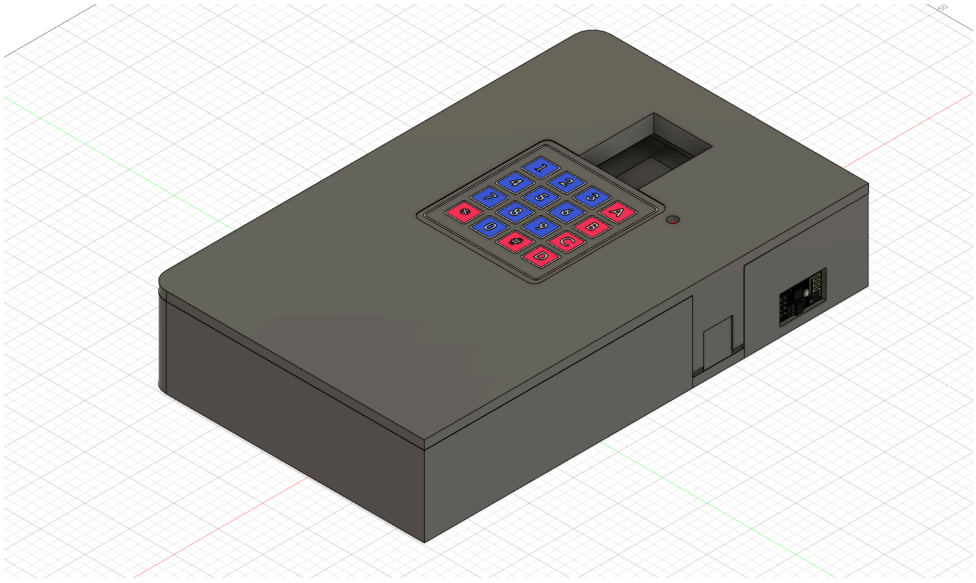
Fonte: Autoria própria (2022).

Figura 26 – Mecânica com módulos sem tampa - Vista diagonal 2



Fonte: Autoria própria (2022).

Figura 27 – Mecânica completa - Vista diagonal 1



Fonte: Autoria própria (2022).

Figura 28 – Mecânica completa - Vista diagonal 2

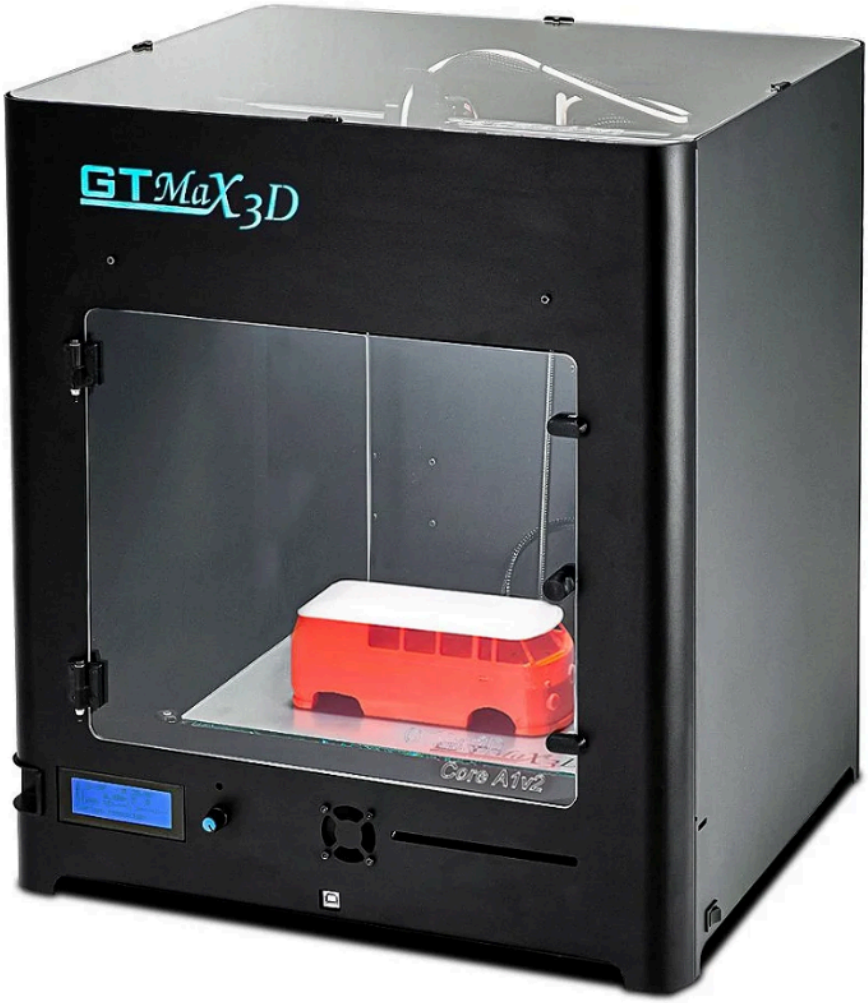
Fonte: Autoria própria (2022).

Após projetar a mecânica descrita acima, foi dado início à pesquisa por um impressora *3-Dimensional* (3D) para imprimi-la e, após a recomendação do professor orientador deste projeto, escolheu-se utilizar o Laboratório MEI-U do Departamento Acadêmico de Eletrônica da Universidade Tecnológica Federal do Paraná (*Campus* Curitiba) para tal atividade. Foi utilizado um filamento de material *Polylactic Acid* (PLA) da cor cinza e a impressora do laboratório em questão é uma GTMAX3D CORE A1V2 cujas especificações incluem:

- Área de Impressão: 300x200x300 mm;
- Cabeça de Impressão: 1;

- Qualidade de Impressão: 0,05mm à 0,32mm;
- Automação: Detecção de fim de filamento e troca automática de filamento;
- Velocidade de Impressão: Até 150mm/s;
- Velocidade de Deslocamento: 300mm/s;
- Mesa: Alumínio aquecida com tampo de vidro;
- Nivelamento: Automático;
- Gabinete: Aço carbono com pintura eletrostática;
- Dimensões da impressora: 490x455x555 mm;
- Alimentação: Bivolt (127V/220V) automático;
- Controle: Display LCD;
- Conectividade USB e Cartão *Secure Digital* (SD).

Figura 29 – Impressora utilizada para impressão da mecânica do projeto



Fonte: (GTMAX3D, 2022).

Figura 30 – Filamento cinza de PLA da marca 3N3 utilizado para compôr a mecânica do projeto

Fonte: (3DCURITIBA, 2022).

Ao fim da impressão das duas peças componentes da mecânica (tampa e base), todos os módulos foram inseridos e parafusados em suas devidas posições e, então, feitas as conexões elétricas entre eles. Por fim, o resultado final pode ser observado pelas Figuras 32 a 35:

Figura 31 – Fechadura eletrônica finalizada - Vista geral

Fonte: Autoria própria (2022).

Figura 32 – Fechadura eletrônica finalizada - Vista lateral direita

Fonte: Autoria própria (2022).

Figura 33 – Fechadura eletrônica finalizada - Vista lateral esquerda

Fonte: Autoria própria (2022).

Figura 34 – Fechadura eletrônica finalizada - Vista do topo

Fonte: Autoria própria (2022).

Figura 35 – Fechadura eletrônica finalizada - Vista do fundo

Fonte: Autoria própria (2022).

3.5 DESENVOLVIMENTO DO *BACK END*

O *Back End* do sistema é dividido em duas partes. A primeira é um servidor local que estabelece a comunicação entre a página web e as fechaduras cadastradas no sistema que estão conectadas na mesma rede. Já a segunda parte consiste em um serviço, com uma estrutura na nuvem responsável pelo armazenamento de informações e segurança da aplicação.

3.5.1 IMPLEMENTAÇÃO DO SERVIDOR LOCAL

O servidor local é a camada do sistema responsável pela comunicação entre o microcontrolador e a aplicação web. Ele é composto por dois servidores menores que se comunicam entre si por meio de diferentes protocolos.

Para execução do servidor local, fez se o uso do *software* de código aberto Node.js, que permite a execução de códigos Javascript fora de um navegador web, juntamente com uma coleção de módulos de desenvolvimento. Para este caso, a aplicação desenvolvida em Javascript utliza o módulo *net* do Node.js para criar um servidor baseado no protocolo de conexão TCP para enviar e receber dados para o microcontrolador e o módulo HTTP também do Node.js, este responsável por criar um servidor baseado no protocolo HTTP para receber e enviar dados para a página web.

O servidor local também possui conectividade com o Google Firebase para registrar alterações de estado e usabilidade das fechaduras no banco de dados Cloud Firestore.

3.5.2 IMPLEMENTAÇÃO DO *BACK END* COMO SERVIÇO

Como parte da estrutura em nuvem do nosso sistema, o Google Firebase foi escolhido por ser uma ferramenta de fácil acesso e que provê diversas facilidades para o desenvolvimento de aplicações, principalmente na parte de segurança, que é feita através da autenticação e autorização do usuário com seu *email* do Google, e também pelo armazenamento de dados, fazendo-se o uso do banco de dados *Not Only Structured Query Language* (NoSQL) Firestore.

3.5.3 BANCO DE DADOS

Utilizando o banco de dados não relacional Firestore, foi levantado um modelo que representa o sistema, sendo este modelo utilizado para armazenar e consultar os dados dos dispositivos e seus estados. Como o banco de dados escolhido é do tipo não relacional, não existem tabelas e colunas, sendo assim, os dados são armazenados em documentos, organizados em várias coleções de dados. É possível ver o modelo do banco na Figura 36.

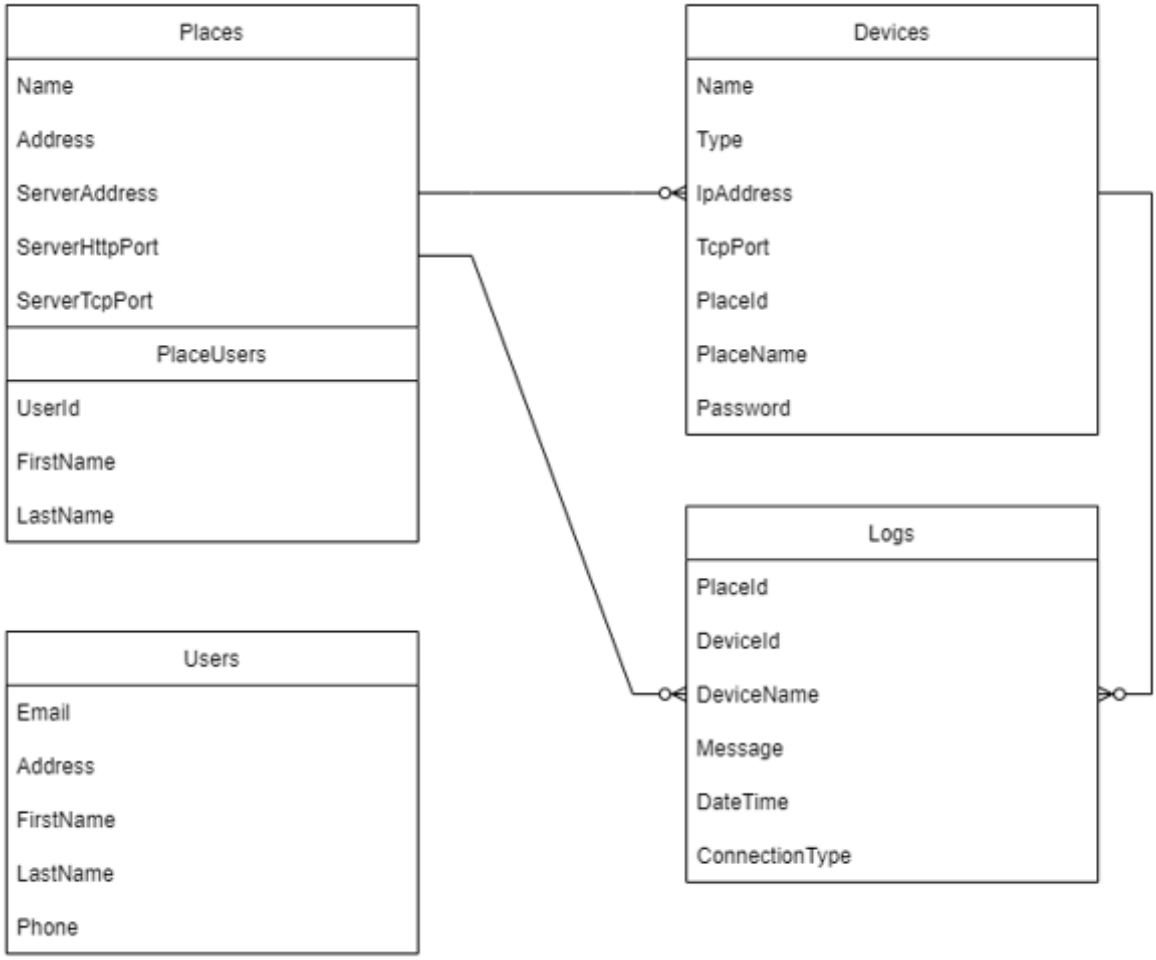
A coleção de *Users* contém a informação de todos os usuários que possuem um cadastro no aplicativo, sendo esses dados coletados na etapa inicial.

A coleção de *Devices* abrange a lista de dispostivos que o usuário possui em seu local cadastrado, além de conter informações de configuração do disposito e conectividade com a rede, como seu endereço de IP (*IpAddress*) e sua porta TCP (*TcpPort*). Por fim cada dispositivo contém um campo *Password*, que é a senha necessária para realizar uma operação manual no dispositivo. Essa coleção possui uma relação de um para muitos com os *Logs*, pois um dispositivo pode conter vários logs associados a ele.

A coleção de *Logs* contém os dados de ações realizadas pelo dispositivo. O campo *Message* armazena a mensagem do tipo *string* que foi enviada para o dispositivo. Já o campo *ConnectionType* descreve se a ação foi realizada manualmente no dispositivo, como é o caso da fechadura sendo aberta por senha, ou se a fechadura foi aberta remotamente pelo aplicativo. Por fim o campo *DateTime* é do tipo data e nele é armazenado o horário que a ação foi executada.

A coleção de *Places* descreve os dados referentes aos locais cadastrados no aplicativo, sendo que um local cadastrado pode ter muitos dispositivos e este mesmo local possui vários logs de atividades. O campo *ServerAddress* possui a informação referente ao IP do servidor local e os campos *ServerHttpPort* e *ServerTcpPort* armazenam as informações da porta HTTP e a porta TCP do servidor, respectivamente. Por fim, dentro da coleção de *Places* existe uma coleção de *PlaceUsers*, que é uma lista de usuários que possuem acesso aos dispositivos deste local.

Figura 36 – Modelo do banco de dados



Fonte: Autoria própria (2022).

3.6 DESENVOLVIMENTO DO *FRONT END*

O *Front End* do sistema consiste em uma aplicação web em que o usuário, através de uma interface gráfica, consegue interagir diretamente com um ou vários dispositivos cadastrados no banco de dados, seja através do envio de dados entre a aplicação e os dispositivos, ou através do monitoramento do estado dos dispositivos em tempo real.

Para listar os locais cadastrados pelo usuário, bem como seus dispositivos associados, foi desenvolvida uma aplicação utilizando React JS juntamente com a biblioteca Material-UI, que possui uma vasta gama de componentes funcionais que auxiliam no desenvolvimento e também na estilização das páginas.

A aplicação é capaz de estabelecer uma conexão HTTP com o *Back End* local por meio da biblioteca *socket.IO.client*, que possui um conjunto de instruções para facilitar uma conexão via socket entre a aplicação e o servidor local. Sendo assim, é possível enviar dados para o servidor que fica responsável por enviar a resposta para o dispositivo selecionado. Além disso, o

servidor também consegue enviar mensagens para a aplicação, desta forma é possível verificar se a mensagem foi enviada.

Para execução do *Front End*, fez se o uso do *software* de código aberto NodeJS juntamente com o Node Package Manager, que permitem a execução de códigos Javascript criando uma versão compilada de um software ou parte dele que contém um conjunto de recursos que integram o produto final. Após a existência desta versão compilada, o seu computador passa a operar como um servidor de hospedagem para a aplicação, sendo possível acessar a versão final através do seu navegador pelo endereço de IP 127.0.0.1 conhecido como *loopback address* ou *localhost* seguido da porta 80, ambos considerados um padrão HTTP para um servidor local.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

4.1 TESTES DE *FIRMWARE*

Quanto aos testes básicos de validação de *firmware*, pode-se estabelecer resumidamente dois tipos de verificações: uma para verificar a operabilidade do menu da fechadura e outro para validar o funcionamento da fechadura como um todo, no qual verifica-se a operação da trava solenoide tanto manualmente (através do manuseio do menu) como remotamente utilizando-se um *software* de simulação e gerenciamento de redes isolando, assim, a parte de *hardware* e *firmware* da parte de *software*.

4.2 VERIFICAÇÃO DE FUNCIONAMENTO DO MENU

O processo de validação do diagrama de estados descrito na seção Fluxo de Operação da Thread tid-phaseB (Menu) é ilustrado nas Figuras 37 a 41:

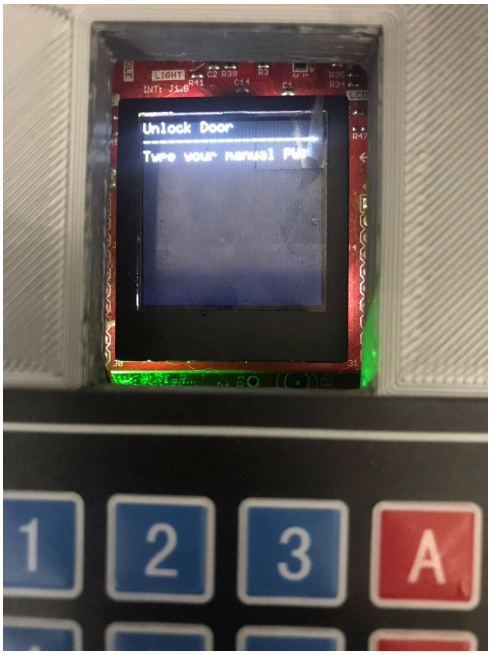
Figura 37 – Tela principal do menu

Fonte: Autoria própria (2022).

Figura 38 – Tela de opções de comando

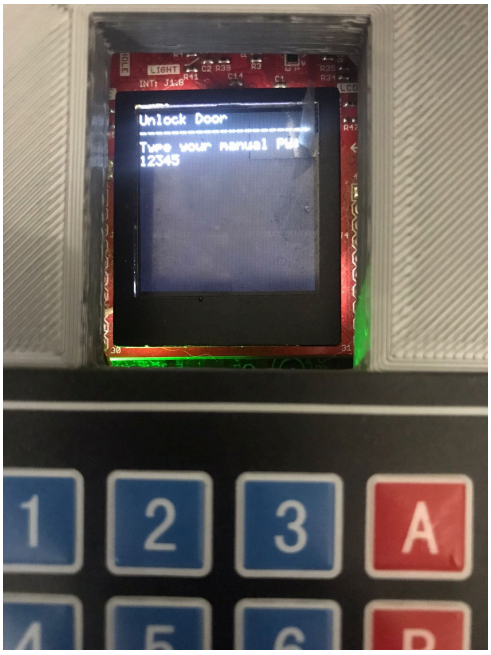
Fonte: Autoria própria (2022).

Figura 39 – Tela de entrada de senha do usuário para desbloquear trava solenoide



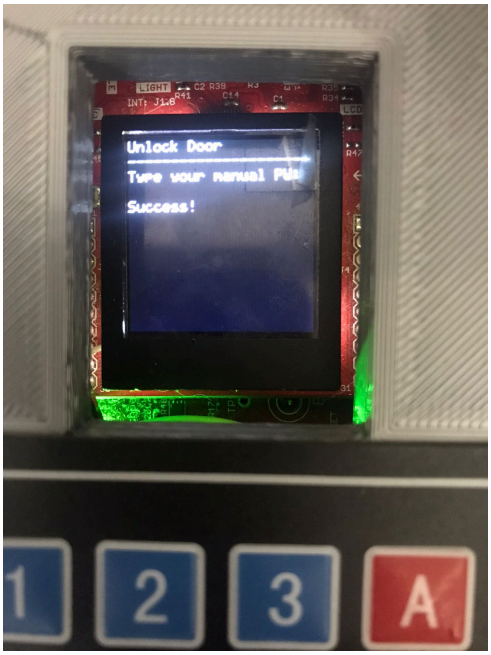
Fonte: Autoria própria (2022).

Figura 40 – Tela de entrada de senha do usuário com senha correta digitada



Fonte: Autoria própria (2022).

Figura 41 – Tela de mensagem de senha digitada com sucesso

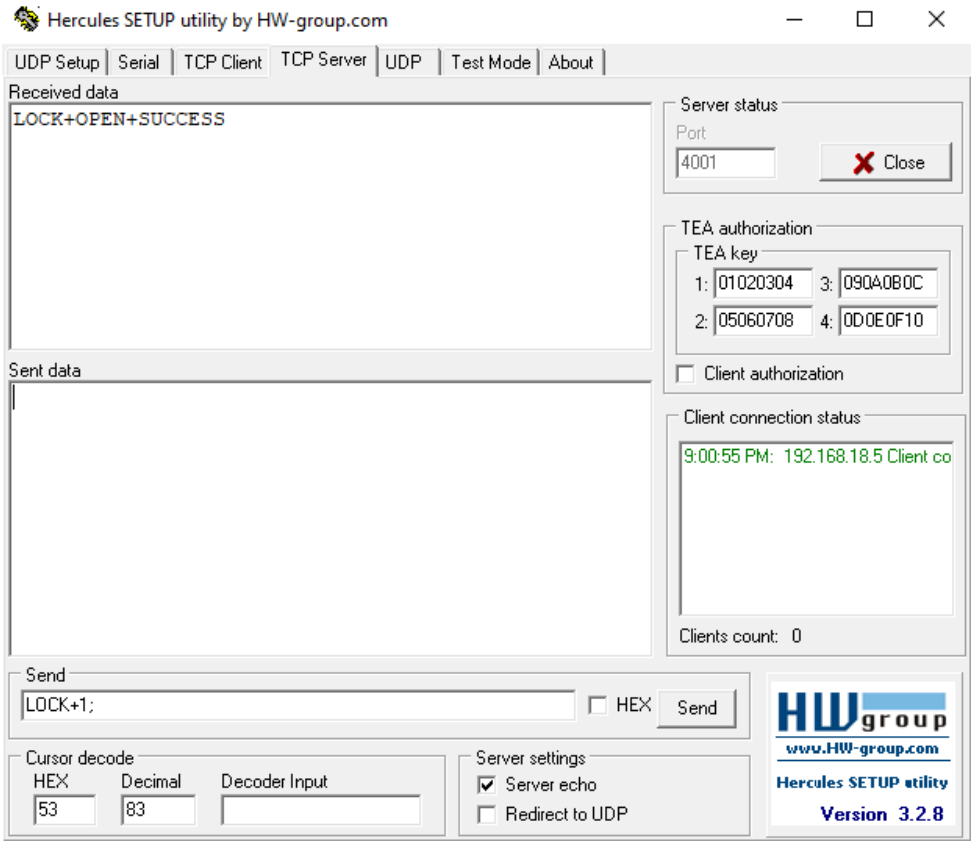


Fonte: Autoria própria (2022).

A fim de testar o envio da mensagem de evento de abertura da trava para o servidor ao qual a fechadura está conectada, foi utilizado o *software* Hercules SETUP Utility ilustrado na Figura 42, que é uma ferramenta de teste de conexões de rede, o qual foi configurado como um servidor TCP escutando quaisquer mensagens que eventualmente cheguem através de uma porta de número conhecido. Vale lembrar que, para efeitos de ilustrar o *payload* da mensagem

durante este teste, a função nativa de criptografia de mensagens SHA-1 foi desabilitada no código-fonte do *firmware* da fechadura.

Figura 42 – Tela de mensagem do *software* Hercules SETUP Utility



Fonte: Autoria própria (2022).

A seguir, as Figuras 43 a 50 demonstram um exemplo do fluxo do menu quando a senha digitada pelo usuário não corresponde à senha cadastrada ao mesmo:

Figura 43 – Tela de entrada de senha do usuário com senha incorreta digitada após primeira tentativa

Fonte: Autoria própria (2022).

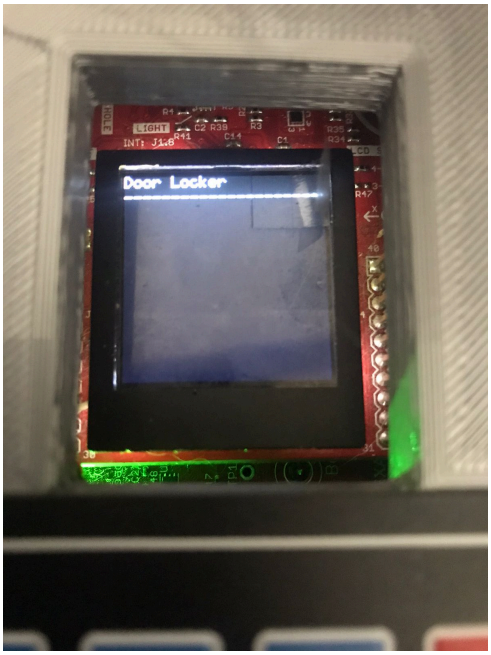
Figura 44 – Tela de entrada de senha do usuário com senha incorreta digitada após segunda tentativa

Fonte: Autoria própria (2022).

Figura 45 – Tela de entrada de senha do usuário com senha incorreta digitada antes da terceira tentativa

Fonte: Autoria própria (2022).

Figura 46 – Tela inicial após terceira tentativa incorreta

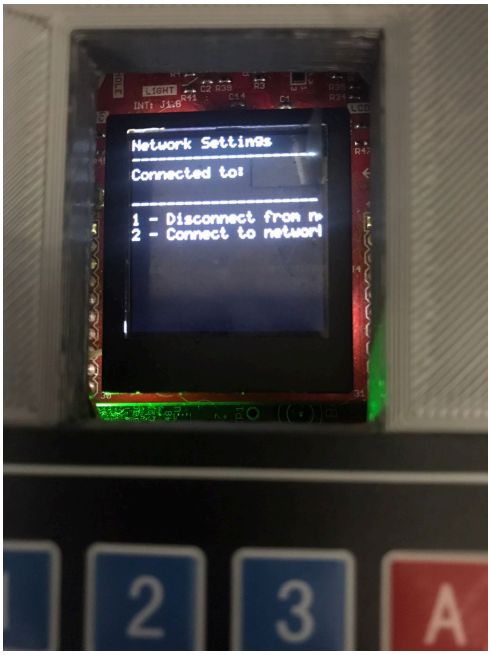


Fonte: Autoria própria (2022).

Figura 47 – Tela de opções de comando

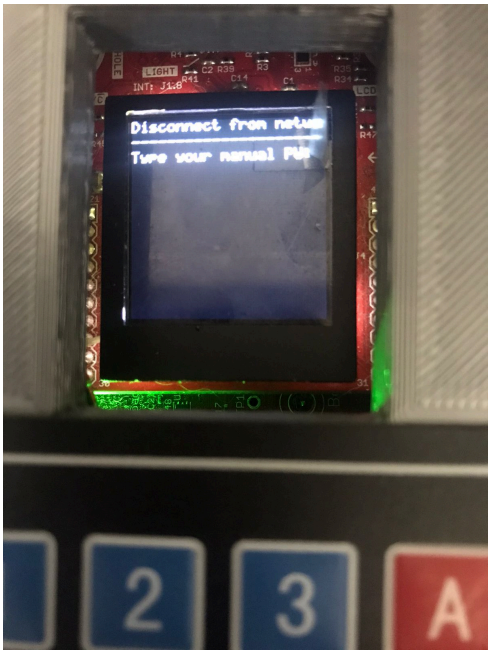
Fonte: Autoria própria (2022).

Figura 48 – Tela de configurações de rede



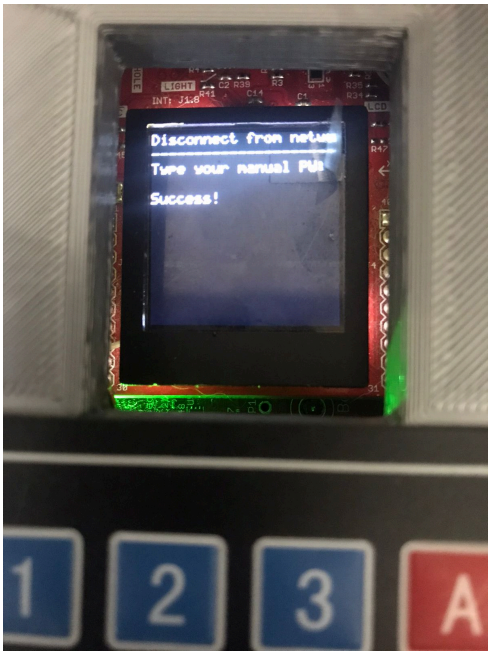
Fonte: Autoria própria (2022).

Figura 49 – Tela de entrada de senha do usuário para desconectar da rede atual



Fonte: Autoria própria (2022).

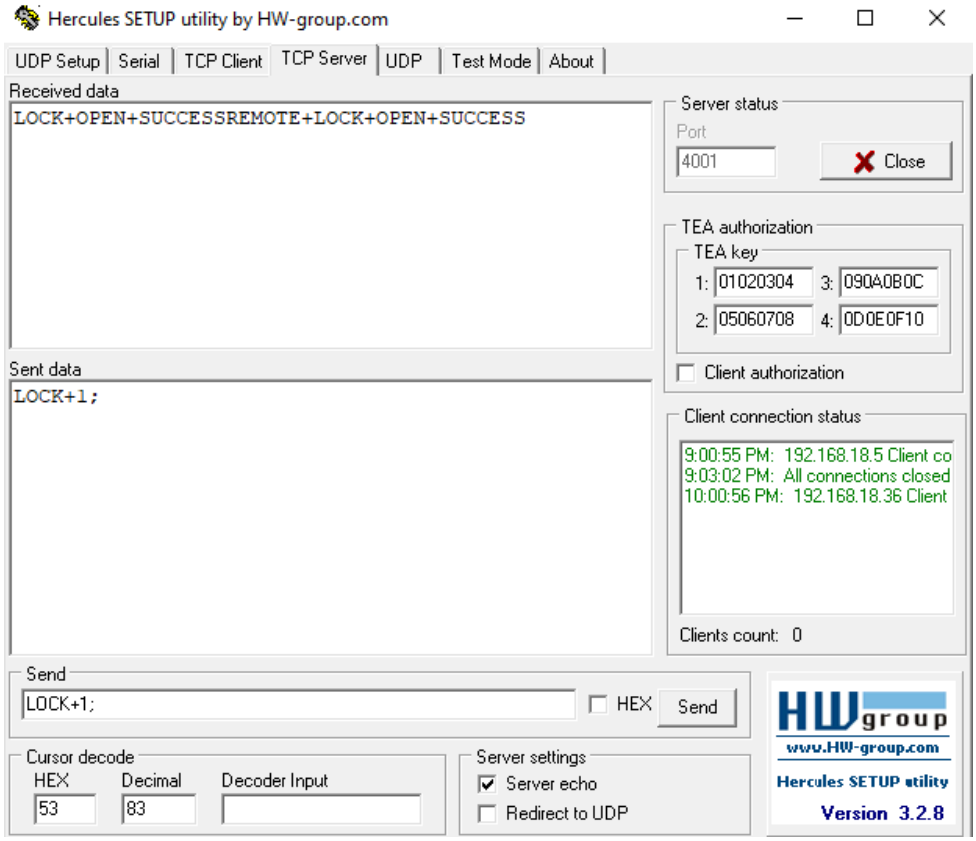
Figura 50 – Tela de mensagem de senha digitada com sucesso



Fonte: Autoria própria (2022).

Finalmente, após reconectar na rede padrão, foi feito o teste de operação remota da trava solenoide a partir do mesmo *software* utilizando o comando válido "LOCK+1;"para que a fechadura o identifique, opere e envie de volta ao servidor a mensagem especificando que a operação foi feita remotamente, conforme mostra a Figura 51:

Figura 51 – Tela de mensagem do *software* Hercules SETUP Utility após operação remota válida



Fonte: Autoria própria (2022).

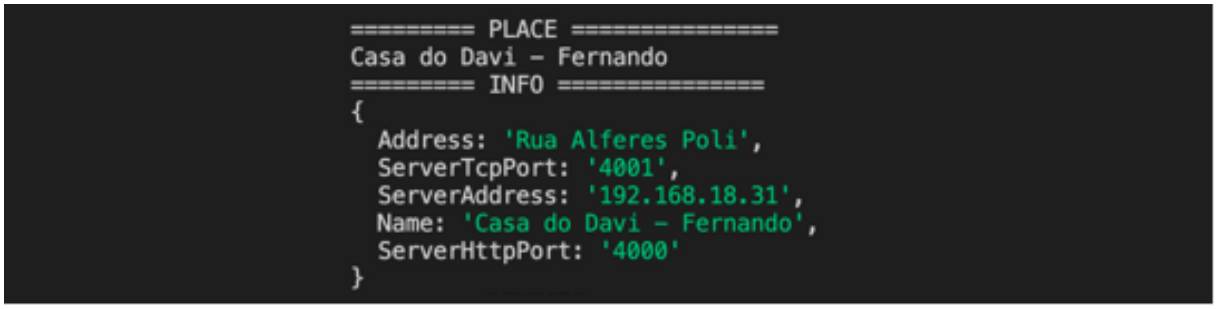
4.3 TESTES DE *SOFTWARE*

Para a realização de testes, foi desenvolvido um código em Javascript para inserir os dados no banco de dados do Google Firebase a fim de simular uma situação real de utilização da aplicação. Neste caso um usuário possui um cadastro no sistema com duas residências sendo que uma delas possui três dispositivos que conseguem efetuar a troca de dados com o servidor.

4.3.1 TESTES DO SERVIDOR

Com o servidor ligado, primeiramente foi verificada a conexão com o Google Firebase, sendo esta conectividade a primeira necessária já que o servidor busca no banco de dados as informações referentes à rede do sistema para efetuar as configurações dos servidores TCP e HTTP que serão utilizados. A Figura 52 ilustra os logs com as informações adquiridas pelo servidor:

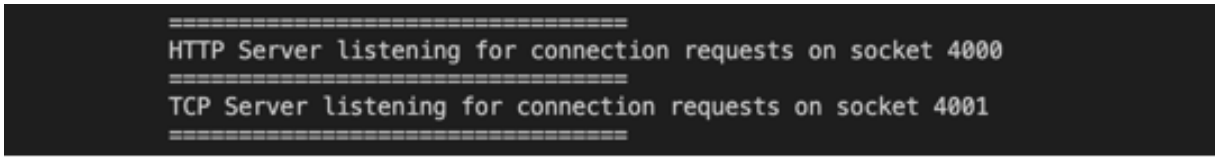
Figura 52 – Informações adquiridas pelo servidor



Fonte: Autoria própria (2022).

Com o acesso as informações, as configurações dos sub servidores TCP e HTTP foram efetuadas, conforme mostra a Figura 53:

Figura 53 – Informações sobre sub servidores



Fonte: Autoria própria (2022).

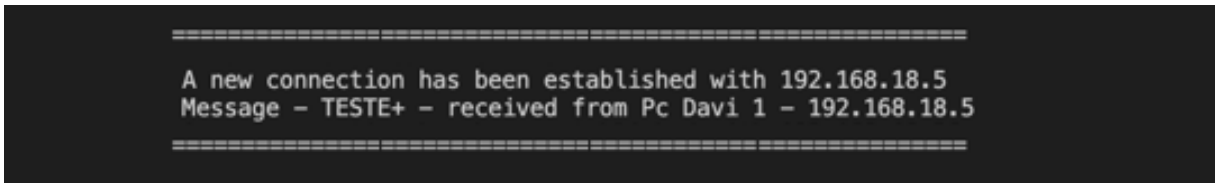
Após todas as configurações, a aplicação realiza uma conexão novamente com o banco de dados para encontrar os dispostivos cadastrados naquela residência. A Figura 54 ilustra os dispositivos que foram encontrados pelo servidor que podem efetuar uma troca de mensagens.

Figura 54 – Dispositivos disponíveis

Fonte: Autoria própria (2022).

Para os testes de conectividade com os dispositivos, primeiramente foram realizados testes com o servidor TCP, utilizando um *software* de simulação de TCP/IP Cliente Servidor, sendo possível verificar o envio e recebimento de mensagens em outro computador. A Figura 55 mostra as informações recebidas pelo servidor vindas de um computador utilizando um software cliente.

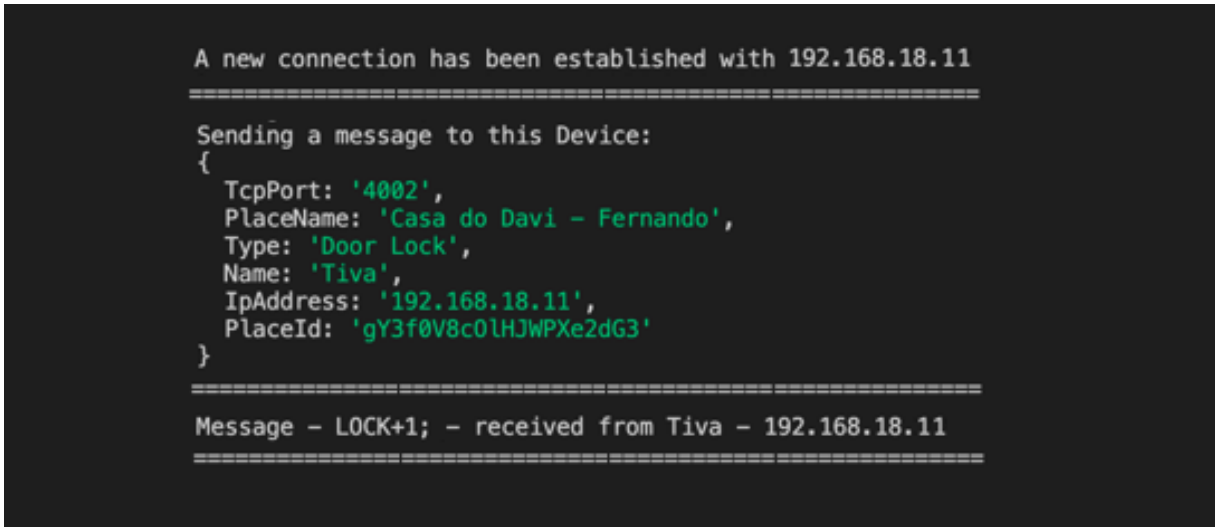
Figura 55 – Testes de Conectividade



Fonte: Autoria própria (2022).

Por fim, os próximos testes foram realizados diretamente com um microcontrolador, sendo que o servidor primeiramente verifica se o microcontrolador está na lista de dispositivos por uma questão de segurança, lista o dispositivo na tela e, após o envio da mensagem, o microcontrolador envia uma resposta informando que a mensagem foi recebida com sucesso pelo dispositivo. A Figura 56 ilustra a troca de mensagens entre um dispositivo e o servidor.

Figura 56 – Conectividade com o Microcontrolador

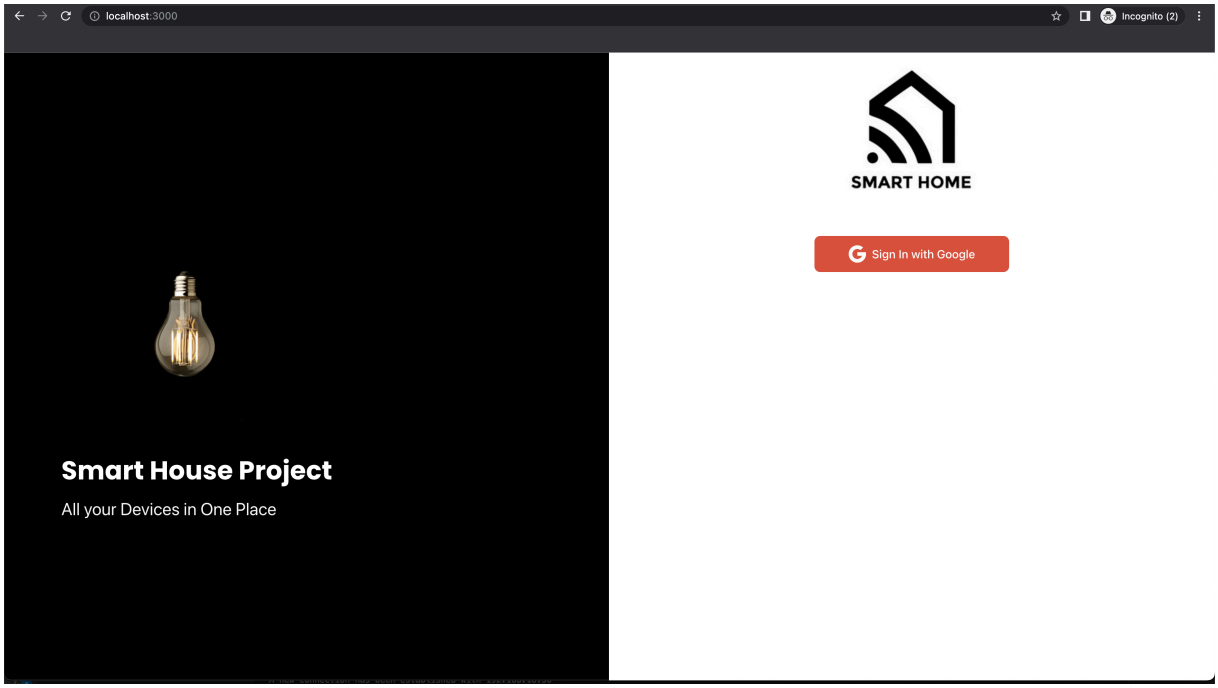


Fonte: Autoria própria (2022).

4.3.2 TESTES DA APLICAÇÃO WEB *FRONTEND*

Com a aplicação em execução, primeiramente foi testada a exibição da tela inicial, juntamente com a opção de *login*, sendo possível realizar o *login* clicando no botão e utilizando a sua conta do Google, conforme ilustra a Figura 57.

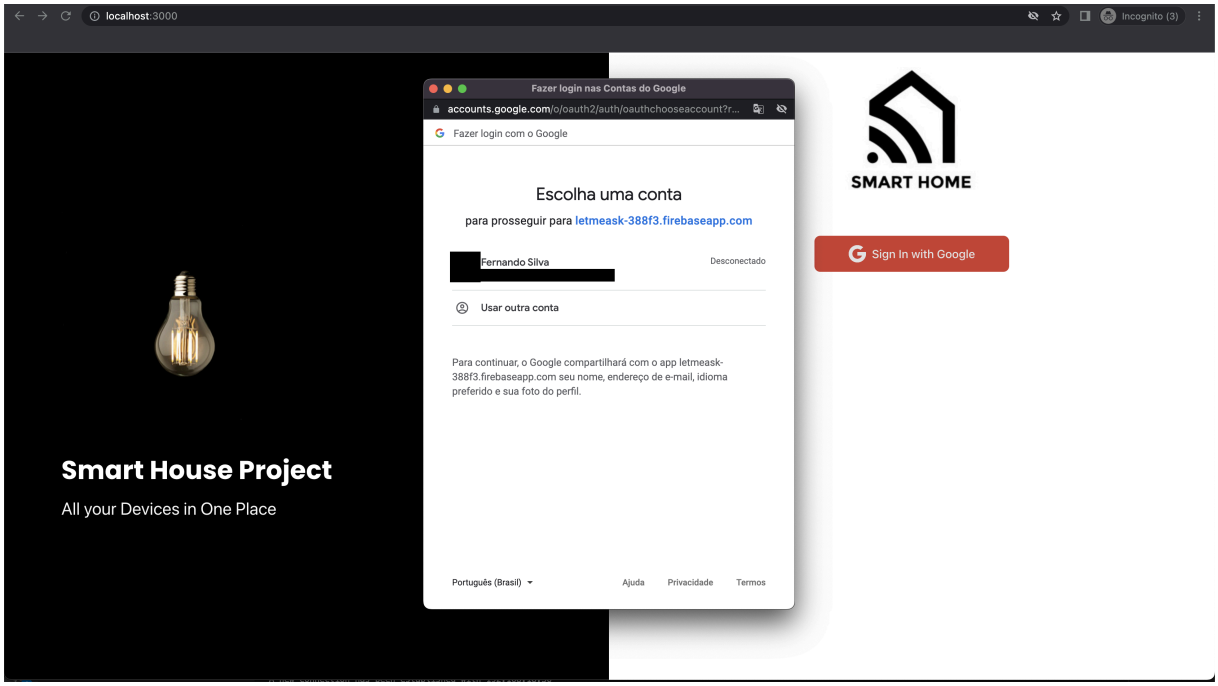
Figura 57 – Tela Inicial



Fonte: Autoria própria (2022).

Após clicar no botão de *Sign In*, uma nova janela aparece para o usuário inserir seu *email* da sua conta do Google, como mostra a Figura 58.

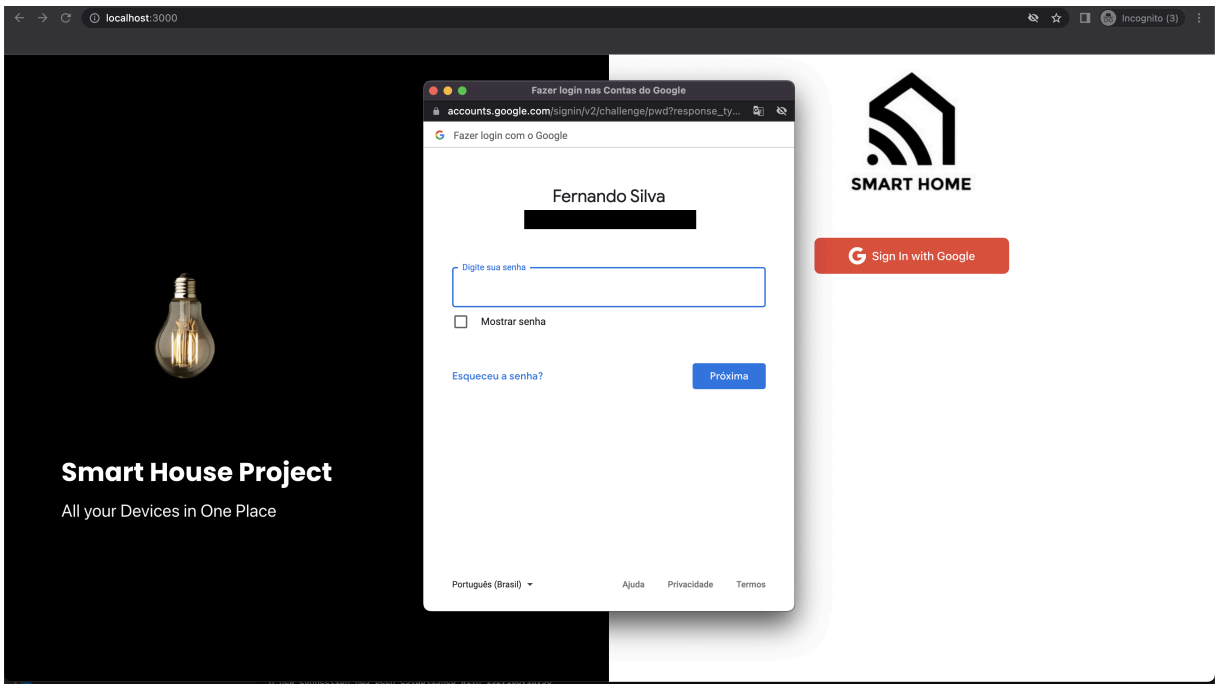
Figura 58 – Janela de *login* - Email



Fonte: Autoria própria (2022).

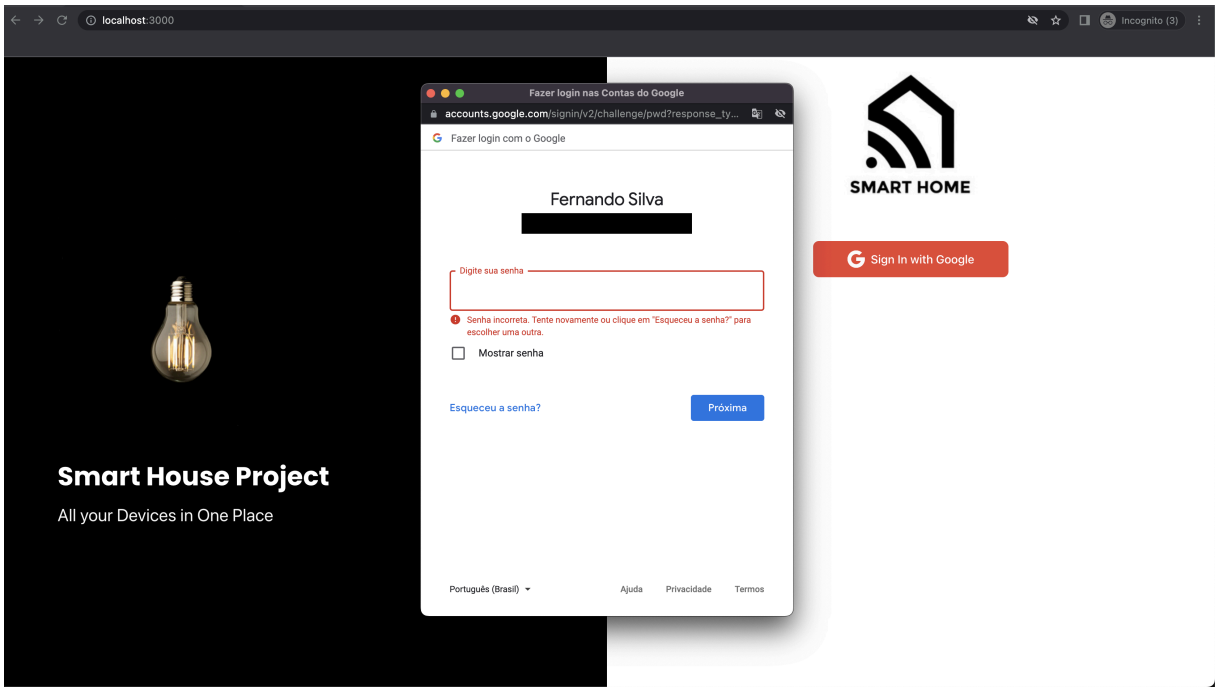
Com a existência da conta verificada, o usuário deve informar sua senha. Com isso, foram realizados testes com diferentes senhas, para verificar a segurança da aplicação, conforme mostrado na Figura 59 e Figura 60.

Figura 59 – Janela de *login* - Senha do usuário



Fonte: Autoria própria (2022).

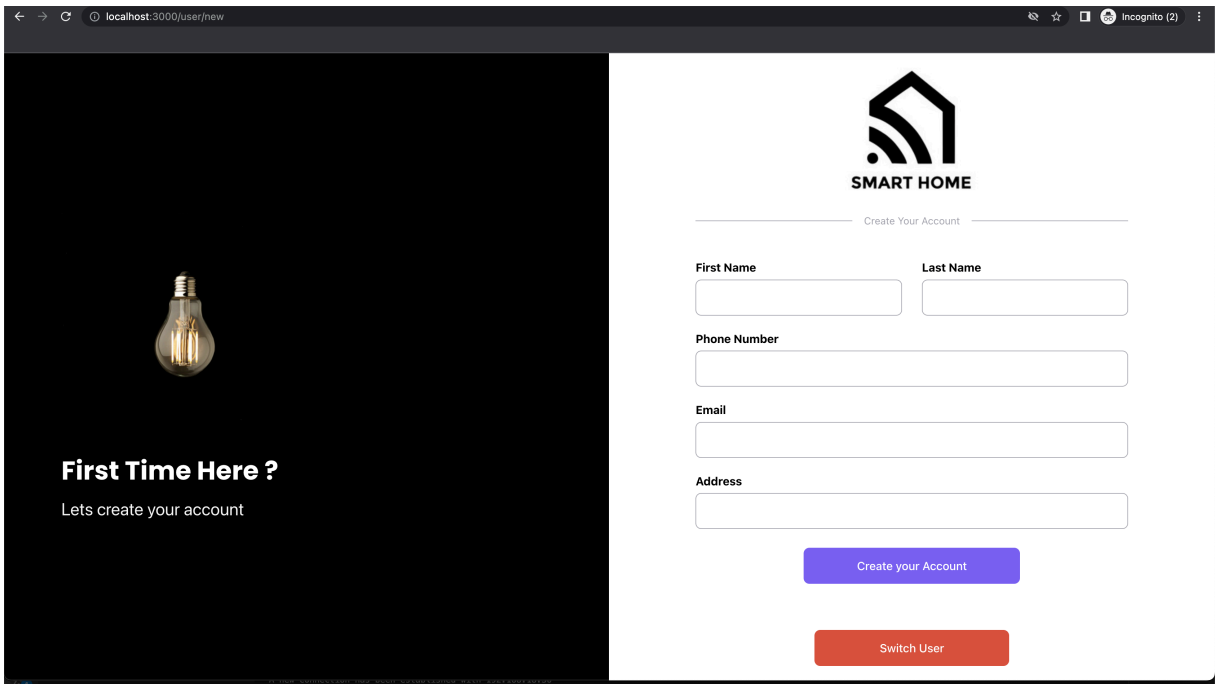
Figura 60 – Janela de *login* - Validação de senha



Fonte: Autoria própria (2022).

Caso o usuário não possua um cadastro no sistema, ele será redirecionado para uma tela de cadastro, em que pode inserir seus dados ou trocar de conta. Estes dados inseridos são armazenados no banco de dados do sistema, conforme mostra a Figura 61.

Figura 61 – Tela de cadastro



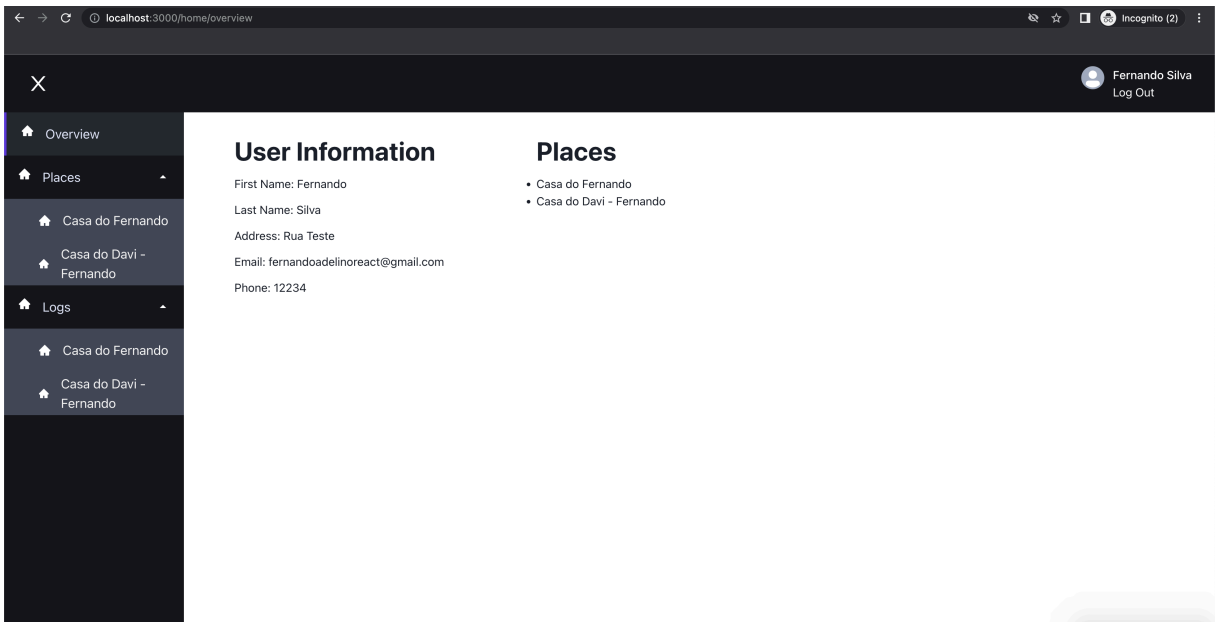
Fonte: Autoria própria (2022).

Após essa etapa inicial, o usuário é redirecionado para a tela de *Overview* da aplicação, ilustrada na figura 62. Nesta tela, o usuário pode verificar os dados inseridos por ele na etapa de cadastro, bem como verificar suas residências cadastradas no sistema, que foram inseridas com ajuda do código Javascript descrito no início da sessão de Testes de *Software*.

No canto esquerdo, é possível visualizar um menu e navegar entre as diferentes janelas do sistema. Na aba *Overview*, o usuário volta para a tela de *Overview* da aplicação. A aba *Places* contém uma lista das diferentes residências que o usuário cadastrou e, pela aba *logs*, é possível verificar os dados referentes a ações realizadas pelos dispositivos de uma residência selecionada.

Por fim, no canto superior direito, o usuário pode realizar o *logout* e retornar para a tela inicial.

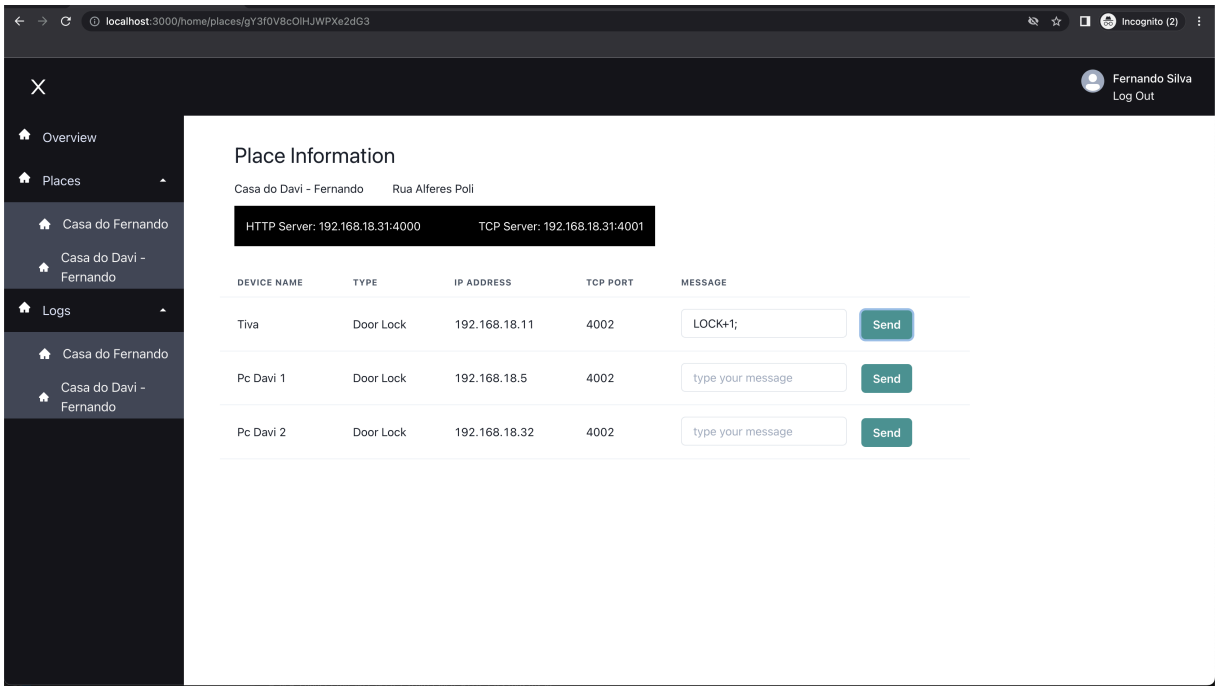
Figura 62 – Tela de *overview*



Fonte: Autoria própria (2022).

Ao selecionar uma das residências na aba *Places*, o sistema faz o redirecionamento para uma nova página, onde é possível verificar algumas informações sobre as configurações de rede, nome e endereço daquele local. Os dispositivos são listados nesta página, sendo possível enviar um comando para a fechadura selecionada, como uma *string* do tipo LOCK+1; para efetuar a abertura da mesma de maneira remota. Este processo é demonstrado na Figura 63.

Figura 63 – Informações da residência



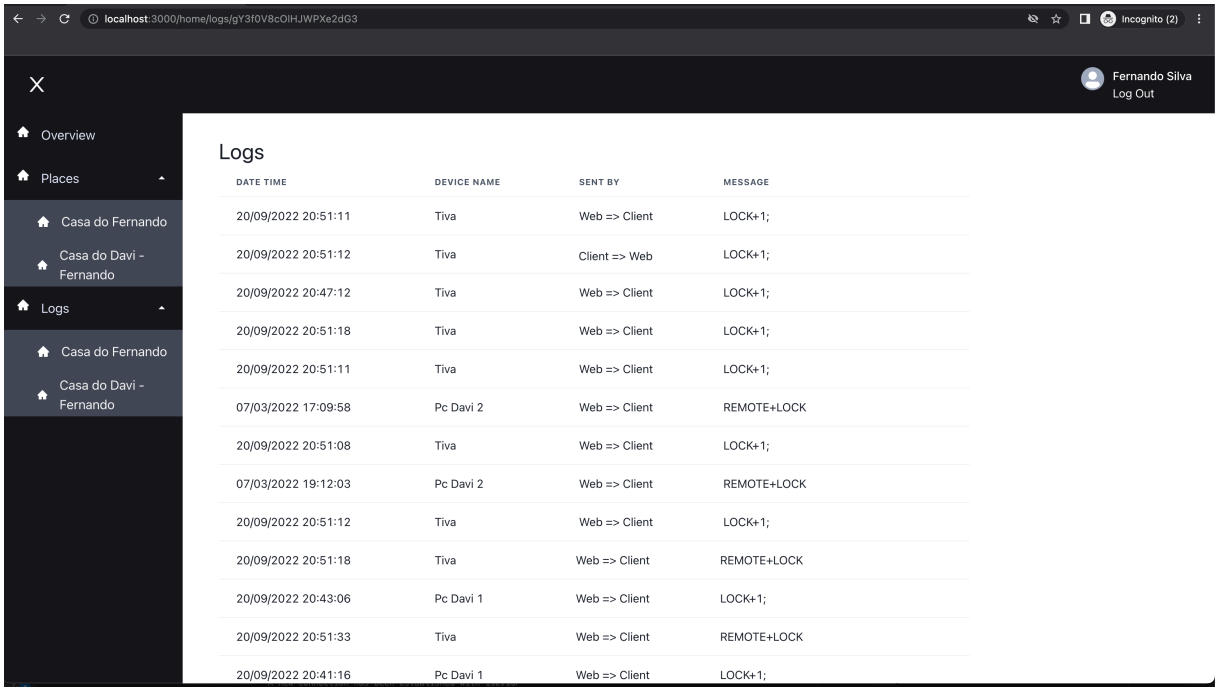
Fonte: Autoria própria (2022).

Os últimos testes de software foram realizados na aba *Logs*. Nesta aba o usuário também pode selecionar uma de suas residências, e um novo redirecionamento é feito pelo sistema e uma nova janela com informações sobre o monitoramento dos dispositivos pode ser visualizada.

As informações são disponibilizadas em um formato de tabela, com cada coluna sendo descrita a seguir:

- *Date Time*: data e a hora que a mensagem foi enviada ou recebida pelo dispositivo listado;
- *Device Name*: nome do dispositivo;
- *Sent By*: campo que possui somente dois valores possíveis, sendo eles:
  - *Web => Client* – significa que a mensagem foi enviada pela aplicação *web* para o dispositivo em questão;
  - *Client => Web* – significa que a mensagem foi enviada pelo dispositivo para a aplicação *web*, como por exemplo um comando de abertura da fechadura sendo feito de forma manual.
- *Message*: valor do tipo *string* que o dispositivo recebeu ou enviou para a aplicação.

Figura 64 – Tela de logs



Fonte: Autoria própria (2022).

5 CONCLUSÃO

O sistema teve como proposta a criação de uma fechadura eletrônica que pudesse ser acionada manualmente ou de maneira remota por meio de uma conexão com a internet e com uma interface de usuário. Desde o início deste projeto o tema central foi o conceito de casas inteligentes. Cada vez mais hoje em dia as pessoas querem ter o controle de tudo que acontece, e isso faz ainda mais sentido se for considerada a infinidade de possibilidades e alterações no cotidiano que um dispositivo móvel pode proporcionar em nossa rotina. Sendo assim, o primeiro objetivo atingido com sucesso foi o de criar uma página *web* em que um usuário pudesse ter controle das várias fechaduras que possui em sua residência, criando um protótipo inicial de casa inteligente utilizando este conceito de vários dispositivos conectados.

Analisando pelo ponto de vista de *software*, o servidor implementado necessita apenas do nome da residência como uma chave primária para buscar todas as informações necessárias no banco de dados. Dessa forma, o servidor realiza as configurações e estabelece todas as conexões necessárias com a rede local. Por fim, de maneira similar, o mesmo ocorre com as conexões entre o servidor e os dispositivos, buscando as informações utilizando essa mesma chave. O *Front End web* funciona da mesma maneira, buscando a lista de residências de um determinado usuário e logo após, a lista de dispositivos e dados referentes a residência que é selecionada na interface. Tudo isso corrobora com a ideia, a nível de software, de que podem ser utilizados diferentes dispositivos com esta mesma implementação, tendo uma central de controle de dispositivos em um só aplicativo.

5.1 TRABALHOS FUTUROS

Com o intuito de aprimorar o que foi realizado neste trabalho, existem alguns horizontes possíveis que devem ser considerados para aprimorar o escopo do projeto. Sobre o ponto de vista de *hardware*, há várias melhorias que podem ser implementadas, sendo elas:

- Diminuição das especificações do microprocessador, no caso um TM4C1294NCPDT, para se adequar aos requisitos e recursos utilizados pelo periféricos. Atualmente sabe-se que sobram recursos do processador, tais como GPIOs, conversores, interfaces e processamento, podendo até se diminuir o *clock* do processador fazendo com que ele entre em um estado de consumo de energia menor do que o atual;
- Utilizar a saída DC do módulo de alimentação externa para alimentar o módulo central de processamento, descartando assim a necessidade de utilização de circuito USB para que este módulo funcione;
- Implementação de um circuito de baterias ou supercapacitores que, durante eventuais ocorrências de quedas de energia da rede elétrica domiciliar, substituam temporari-

amente o módulo de alimentação externa para manter a operabilidade da fechadura eletrônica;

- Inclusão de uma fechadura convencional para abertura manual da porta em condições extremas nas quais a queda de energia superou o tempo de operação do circuito de baterias/supercapacitores;

- Implementação de novos módulos que possam expandir as possibilidades de desbloqueio da fechadura e aumentam a segurança da mesma. Alguns exemplos de tais aplicações podem incluir biometria, câmera para reconhecimento facial, dispositivos *Radio Frequency Identification* (RFID), entre outros;

- Descartar o uso do *kit* EK-TM4C1294XL, aproveitando dele somente seu microprocessador. Isto acarretaria na necessidade de se projetar uma placa que irá utilizar todos os recursos do processador necessários somente ao projeto diminuindo, assim, o custo financeiro e as dimensões físicas da placa de circuito impresso;

- Implementação de um circuito de *driver* para um *display* gráfico maior sem que haja necessidade de uso do *kit* Boosterpack MKII;

- Desenvolvimento de outros dispositivos IOT, tais como lâmpadas, tocadores de música, motores de cortina, portões, que possam se comunicar com o mesmo servidor e serem controlados pelo dono da fechadura do domicílio. Logo, o dono desta poderá ter acesso e controlar todo dispositivo inteligente que há dentro do mesmo imóvel.

No que diz respeito ao âmbito de arquitetura de *software*, há a possibilidade de mover a parte do sistema para a nuvem, removendo a dependência que hoje a aplicação possui de ter um servidor rodando localmente na mesma rede da residência para estabelecer as comunicações necessárias.

No quesito acessibilidade, existe a possibilidade de desenvolver um aplicativo para dispositivos móveis (Android e iOS). Esse foi um dos pontos na escolha do React JS como a biblioteca de desenvolvimento da aplicação *web Front End*, já que existe outra biblioteca também desenvolvida pelo Facebook chamada React Native, focada para dispositivos móveis e que possui diversas similaridades com o projeto já executado, diminuindo curva de aprendizado e assim facilitando o desenvolvimento.

No ponto de vista de interface da aplicação *web*, há a possibilidade de criar uma nova janela em que o usuário final consiga adicionar e configurar novos dispositivos sem a necessidade de uma configuração inicial para isso, bem como um sistema de hierarquia, onde um usuário pode ser um administrador que possui diferentes acessos no sistema e ter a capacidade de incluir diferentes usuários que podem utilizar os dispositivos de sua residência.

REFERÊNCIAS

3DCURITIBA. *Filamento PLA Prata 1.75mm (3N3)*. 2022. Disponível em: <https://www.3dcuritiba.com.br/filamento-pla-prata-175mm-3n3>. Acesso em: 8 de setembro de 2022.

ALASDAIRALLAN. **Getting Started with the ESP8266**. 2022. Disponível em: https://aallan.medium.com/getting-started-with-the-esp8266-270e30feb4d1. Acesso em: 7 de junho de 2022.

BAYLINEAR. **LM3940 - Voltage Regulator for Convert 5V & 3.3V**. 2022. Disponível em: https://pdf1.alldatasheet.com/datasheet-pdf/view/91825/ETC/LM3940.html. Acesso em: 16 de agosto de 2022.

BOYLESTAD. **Introdução a Análise de Circuitos**. 10. ed. [*S.l.*]: Pearson Universidades, 2003.

BOYLESTAD. **Introdução a Análise de Circuitos**. 10. ed. [*S.l.*]: Pearson Universidades, 2003.

BOYLESTAD, R. L.; Louis Nashelsky. **Dispositivos Eletrônicos e Teoria dos Circuitos**. 8. ed. [*S.l.*]: Pearson, 2002.

BRASIL. **Plano Nacional de Internet das Coisas - IoT - Ministério da Ciência e Tecnologia**. 2019. Disponível em: https://www.gov.br/mcti/pt-br/acompanhe-o-mcti/transformacaodigital/internet-das-coisas. Acesso em: 14 de outubro de 2022.

BRITANNICA, T. E. O. E. **TCP/IP**. 2022. Disponível em: https://www.britannica.com/technology/TCP-IP. Acesso em: 13 de outubro de 2022.

DATAFOLHA. **A relaçao dos brasileiros com sua casa**. 2022. Disponível em: https://censodemoradia.quintoandar.com.br/a-relacao-dos-brasileiros-com-a-sua-casa/. Acesso em: 28 de outubro de 2022.

FILIPEFLOP. **ESP8266 ESP-01**. 2022. Disponível em: https://www.filipeflop.com/produto/modulo-wifi-esp8266-esp-01/. Acesso em: 2 de julho de 2022.

FUTURECOM. **Plano Nacional de Internet das Coisas - IoT**. 2020. Disponível em: https://digital.futurecom.com.br/transformacao-digital/iot-no-brasil-quais-sao-barreiras-e-como-supera-las. Acesso em: 14 de outubro de 2022.

GIACOMIN, J. C. **Circuitos Elétricos**. 2002. Disponível em: http://algol.dcc.ufla.br/~giacomin/Com145/Eletricidade.pdf. Acesso em: 23 de outubro de 2022.

GTMAX3D. **GTMAX3D CORE A1V2**. 2022. Disponível em: https://www.gtmax3d.com.br/impressora-3d-pro/gtmax3d-core-a1v2. Acesso em: 7 de setembro de 2022.

IOTANALYTICS. **Number of connected IoT devices growing 18% to 14.4 billion globally**. 2022. Disponível em: https://iot-analytics.com/number-connected-iot-devices/. Acesso em: 14 de outubro de 2022.

KASPERSKY. **O que é criptografia de dados? Definição e explicação**. 2022. Disponível em: https://www.kaspersky.com.br/resource-center/definitions/encryption. Acesso em: 22 de outubro de 2022.

MDN. **Métodos de requisição HTTP**. 2022. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/HTTP. Acesso em: 13 de outubro de 2022.

METALTEX. **Miniature Relay AX Type**. 2022. Disponível em: https://www.metaltex.com.br/assets/produtos/pdf/ax.pdf. Acesso em: 15 de agosto de 2022.

MULTICOMP. **LED Yellow / Green, 5mm**. 2012. Disponível em: https://www.farnell.com/datasheets/1671521.pdf. Acesso em: 15 de agosto de 2022.

PHILIPSSEMICONDUCTORS. **6-Pin DIP Optoisolators Transistor Output**. 1995. Disponível em: https://pdf1.alldatasheet.com/datasheet-pdf/view/2848/MOTOROLA/4N35.html. Acesso em: 15 de agosto de 2022.

PHILIPSSEMICONDUCTORS. **DATASHEET 2N2222; 2N2222A NPN switching transistors**. 1997. Disponível em: https://pdf1.alldatasheet.com/datasheet-pdf/view/15067/PHILIPS/2N2222.html. Acesso em: 14 de agosto de 2022.

REACT. **React**. 2022. Disponível em: https://pt-br.reactjs.org/. Acesso em: 22 de outubro de 2022.

SADIKU, M. **Fundamentos de Circuitos Elétricos**. 5. ed. [*S.l.*]: AMGH, 2013.

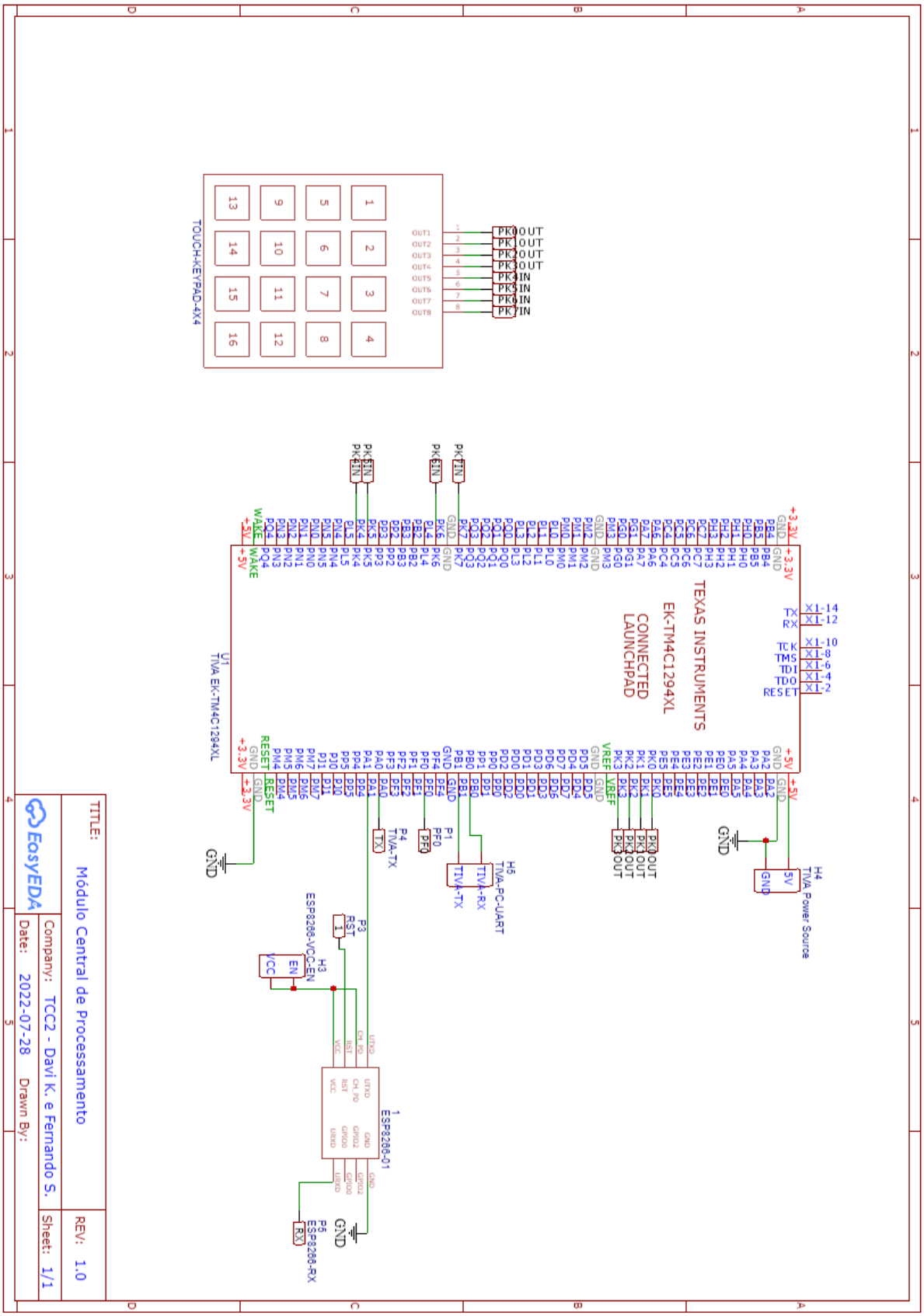
SCHIFLETT, C. **HTTP Developer's Handbook**. 1. ed. [*S.l.*]: Sams, 2003.

SERAFIM, V. D. S. **Introdução à Criptografia: Funções Criptográficas de Hash**. 2012. Disponível em: http://www.serafim.eti.br/academia/recursos/Roteiro_08-Funcoes_de_Hash.pdf. Acesso em: 23 de outubro de 2022.

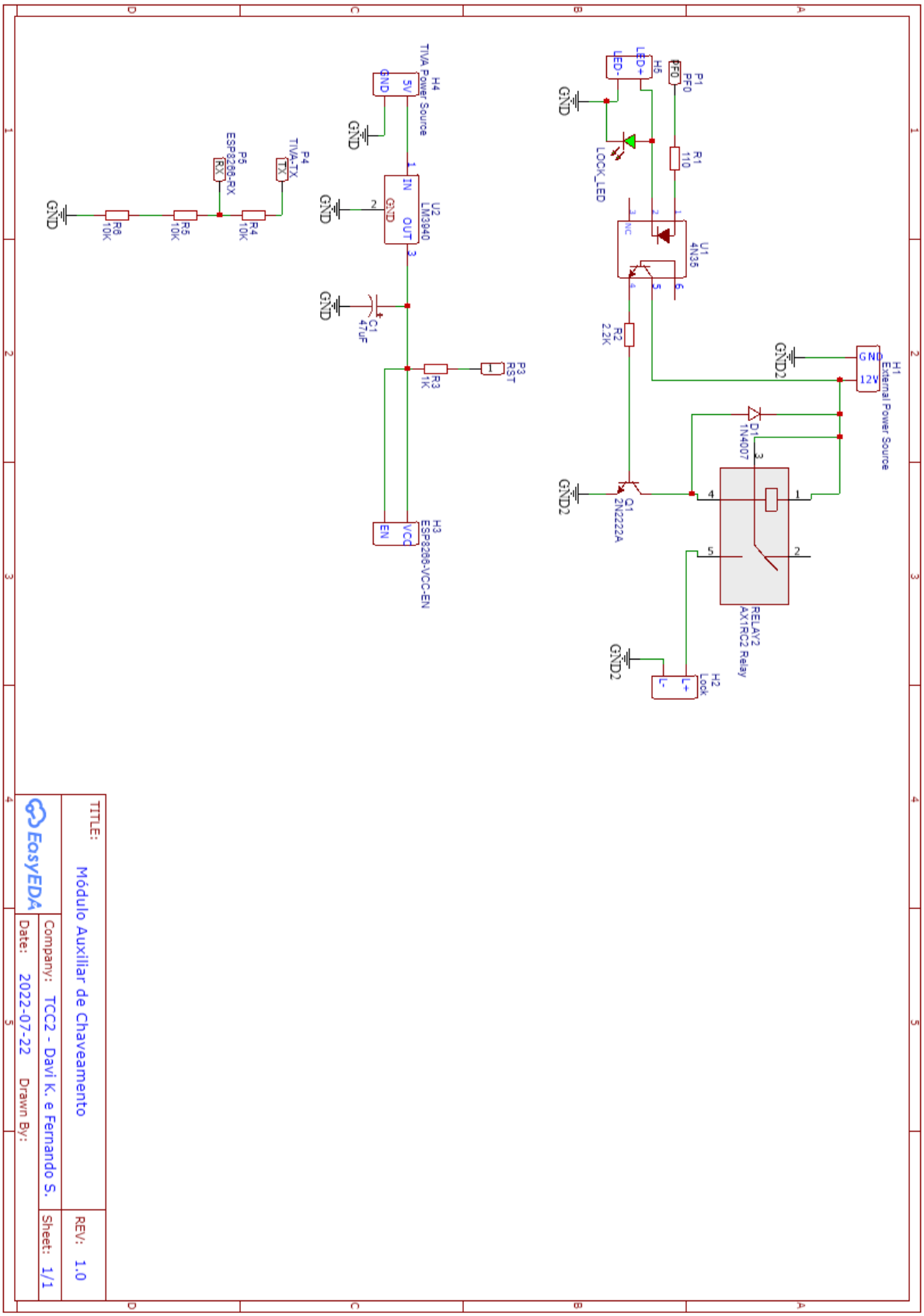
TEXASINSTRUMENTS. **BOOSTXL-EDUMKII**. 2022. Disponível em: https://www.ti.com/tool/BOOSTXL-EDUMKII. Acesso em: 31 de julho de 2022.

TEXASINSTRUMENTS. **EK-TM4C1294XL**. 2022. Disponível em: https://www.ti.com/tool/EK-TM4C1294XL. Acesso em: 2 de julho de 2022.

APÊNDICE A – Esquemático do Módulo Central de Processamento


TOUCH-KEYPAD-4X4
TEXAS INSTRUMENTS
EK-TM4C1294XL
CONNECTED
LAUNCHPAD
U1
TIVA EK-TM4C1294XL
H4
TIVA Power Source
H5
TIVA-PC-UART
P1
TIVA-TX
P4
TIVA-TX
P3
ESP8266-VCC-EN
P5
ESP8266-RX
1
ESP8266-01
TITLE:
Módulo Central de Processamento
EasyEDA
Company: TCC2 - Davi K. e Fernando S.
Date: 2022-07-28
Drawn By:
REV: 1.0
Sheet: 1/1

APÊNDICE B – ESQUEMÁTICO DO MÓDULO AUXILIAR DE CHAVEAMENTO


TITLE:
Módulo Auxiliar de Chaveamento
EasyEDA
Company: TCC2 - Davi K. e Fernando S.
REV: 1.0
Date: 2022-07-22
Drawn By:
Sheet: 1/1
H1
External Power Source
GND2
12V
H2
Lock
L+
L-
H3
ESP8266-VCC-EN
VCC
EN
H4
TIVA Power Source
5V
GND
H5
LED+
LED-
LOCK_LED
P1
PF0
R1
110
R2
2.2K
R3
1K
R4
10K
R5
10K
R8
10K
P3
RST
P4
TIVA-TX
TX
P5
ESP8266-RX
RX
U1
4N35
NC
U2
LM3940
IN
OUT
C1
47uF
D1
1N4007
Q1
2N2222A
RELAY2
AX1RC2 Relay