UNIVERSIDADE DO ESTADO DO AMAZONAS - UEA

ESCOLA SUPERIOR DE TECNOLOGIA - EST

ROBERTO CONHAGO TAVARES DE SOUSA

DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE DE ACESSO E ACIONAMENTO DA ILUMINAÇÃO E AR-CONDICIONADO DAS SALAS DE AULA, UTILIZANDO INTERNET DAS COISAS (IOT)

Manaus - Amazonas

2019

ROBERTO CONHAGO TAVARES DE SOUSA

DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE DE ACESSO E ACIONAMENTO DA ILUMINAÇÃO E AR-CONDICIONADO DAS SALAS DE AULA, UTILIZANDO INTERNET DAS COISAS (IOT)

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentado à banca avaliadora do curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletricista.

Orientadora: Prof. MSc. Ingrid Sammyne Gadelha Figueiredo.

Manaus - Amazonas

2019

Universidade do Estado do Amazonas – UEA
Escola Superior de Tecnologia - EST

Reitor:
Cleinaldo de Almeida Costa
Vice-Reitor:
Cleto Cavalcante de Souza Leal
Diretor da Escola Superior de Tecnologia:
Ingrid Sammyne Gadelha Figueiredo
Coordenador do Curso de Engenharia Elétrica:
Walfredo da Costa Lucena Filho

Banca Avaliadora composta por: *Data da defesa: 18/12/2019.*
Prof. M.Sc. Ingrid Sammyne Gadelha Figueiredo. *(Orientador)*
Prof. M.Sc. Walfredo da Costa Lucena Filho.
Prof. M.Sc. Karlo Homero Ferreira Santos.

CIP – Catalogação na Publicação

Sousa, Roberto Conhago Tavares de

Desenvolvimento de um sistema de controle de acesso e acionamento da iluminação e ar-condicionado das salas de aula, utilizando internet das coisas (IoT) / Roberto Conhago Tavares de Sousa; orientado pela Professora M.Sc. Ingrid Sammyne Gadelha Figueiredo – Manaus, 2019.
87 p. : il. (algumas color.).

Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica). Universidade do Estado do Amazonas, 2019.

1. Internet das Coisas. 2. Logística. 3. Eficiência Energética. 4. Banco de dados. 5. RFID. 6. ESP32. 7. Comunicação *Wireless*. I. Sammyne Gadelha Figueiredo, Ingrid. II. Universidade do Estado do Amazonas. III. Escola Superior de Tecnologia.

ROBERTO CONHAGO TAVARES DE SOUSA

DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE DE ACESSO E ACIONAMENTO DA ILUMINAÇÃO E AR-CONDICIONADO DAS SALAS DE AULA, UTILIZANDO INTERNET DAS COISAS (IOT)

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletricista.

Nota obtida: _____ (_____)

Aprovado em _____/_____/_____.

Área de concentração: Internet das Coisas

BANCA EXAMINADORA

_____

Prof. M.Sc. Ingrid Sammyne Gadelha Figueiredo.
Orientadora

_____

Prof. M.Sc. Walfredo da Costa Lucena Filho.
Avaliador

_____

Prof. M.Sc. Karlo Homero Ferreira Santos.
Avaliador

Manaus – Amazonas
2019

À Deus, minha família e amigos próximos que sempre me deram forças, apoio, incentivo, carinho e amor, para vencer todos os osbstáculos em minha vida.

AGRADECIMENTOS

Primeiramente à Deus, porque eu creio que sem Ele nenhuma das minhas conquistas seriam possíveis, e por sempre ter me abençoado e ajudado nas situações mais difíceis da minha vida.

Em especial um agradecimento à minha família: minha mãe, meus avós, meus irmãos e meu padrasto que sempre me deram todo apoio, carinho e amor durante toda minha vida e com certeza devo a eles poder estar finalizando a minha graduação em engenharia elétrica.

Quero agradecer também a minha professora orientadora M.Sc. Ingrid Sammyne Gadelha Figueiredo, pelo empenho dedicado ao meu projeto de pesquisa e por todo apoio e paciência ao longo da graduação. Também deixar meu agradecimento aos professores, M.Sc. Walfredo da Costa Lucena Filho e o M.Sc. Karlo Homero Ferreira Santos, que compõe a banca examinadora da defesa desse projeto que foram profissionais que fizeram toda diferença em minha graduação.

Um agradecimento especial também a minha amiga Thalyta Brito por ter sido minha parceira de estudos e grande incentivadora durante a graduação. A quem eu sempre pude contar nos momentos de maiores dificuldades do curso.

A todas as pessoas que direta ou indiretamente contribuíram para a realização da minha pesquisa.

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

(Josué 1:9)

RESUMO

Atualmente, o número de projetos que tem por finalidade resolver problemas de logística, otimizando e tornando mais eficientes processos já existentes, têm crescido significativamente. E o conceito de conectar à internet dispositivos eletrônicos utilizados no cotidiano da sociedade, conhecido como Internet das Coisas, tem sido fundamental na solução destes problemas. E no caso deste projeto não será diferente, devido ao fato dele possuir a finalidade de resolver problemas no que diz respeito à logística da utilização das salas de aula da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, onde problemas como a perda da chave da sala e a falta de controle de quem está utilizando a sala, tendem a ocorrer com uma certa frequência. Fora o fato de todo o processo de abertura das salas de aula ocorrer de forma manual, onde um funcionário da instituição é responsável por essa função, o que pode trazer como consequência a demora para abrir as salas e o fato do professor ter que muitas vezes ficar esperando para a mesma ser aberta. Além disso, o projeto tem por finalidade resolver um outro problema que também diz respeito às salas de aula da universidade, voltado para o ramo da eficiência energética, propondo uma diminuição significativa no uso desnecessário da iluminação e refrigeração. Para que esses dois problemas sejam solucionados, o projeto funcionará da seguinte forma: existirá um banco de dados onde informações sobre a data e hora que cada funcionário pode utilizar determinada sala, o nome do funcionário e um código RFID único estarão cadastrados. Onde todas essas informações foram cadastradas no banco através de um *site* criado para este projeto. Assim, um módulo chamado ESP32 utilizado no projeto se conectará a uma rede *Wi-Fi* local, dessa forma permitindo com que ele estabeleça uma comunicação *wireless* com o banco de dados, de tal forma que cada funcionário possua um cartão RFID e toda vez que este for lido pelo sistema e o mesmo possuir permissão para entrar em uma determinada sala em um determinado horário, a fechadura eletrônica, a refrigeração e a iluminação serão ativadas, da mesma forma que quando ele for embora da sala e o seu cartão RFID for lido novamente, esses objetos serão desativados.

Palavras-chaves: Internet das Coisas. Logística. Eficiência Energética. Banco de Dados. RFID. ESP32. Comunicação *Wireless*.

ABSTRACT

Currently, the number of projects aimed at solving logistics problems, optimizing and making existing procedures more efficient, has grown significantly. And the concept of connecting to the internet electronic devices used in everyday society, known as the Internet of Things, has been fundamental in solving these problems. And in the case of this project will be no different, because it has the purpose of solving logistics problems on the use of the classrooms of the Escola Superior de Tecnologia da Universidade do Amazonas, one of the issues is the loss of the key, as the inexistence access control the rooms tends to occur with some frequency. Apart from the fact that the whole process of opening the classrooms takes place manually, where an employee of the institution is responsible for this function, which can result in the delay in opening the classrooms and the fact that the teacher often has to have to wait for it to be opened. In addition, the project aims to solve another problem that also concerns university classrooms, focused on energy efficiency, proposing a significant reduction in unnecessary use of lighting and cooling. To solve these two problems, the project will work as follows: There will be a database where information about the date and time each employee can use a particular room, the employee's name and a unique RFID code will be entered. Where all this data will be registered in the database through a website created for this project. Thus, a module called ESP32 used in the project will connect to a local Wi-Fi network, thus allowing it to wirelessly communicate with the database, so that each employee has an RFID card and every time it is in place. is read by the system and is allowed to enter a particular room at a certain time, the electronic lock, cooling and lighting will be activated, just as when it leaves the room and your RFID card is read again, these objects will be disabled.

Keywords: Internet of Things. Logistics. Energy efficiency. Database. RFID. ESP32. Wireless communication.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CAN	Controller Area Network
CPU	Central Processing Unit
ENCE	Etiqueta Nacional de Conservação de Energia
EST	Escola Superior de Tecnologia
GND	Graduated Neutral Density Filter
GPIO	General-Purpose Input/Outputs
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IDE	Integrated Development Environment
INMETRO	Instituto Nacional de Metrologia, Normalização e Qualidade Industrial
IoT	Internet of Things
IP	Internet Protocol
I2C	Inter-Integrated Circuit
I/O	Input/Output
LCD	Liquid Crystal Display
LED	Light-Emitting
NA	Normalmente Aberto
NF	Normalmente Fechado
OSI	Open Systems Interconnection
PBE	Programa Brasileiro de Etiquetagem
PROCEL	Programa Nacional de Conservação de Energia Elétrica
RFID	Radio-Frequency Identification
RTC	Real Time Clock
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language
SPI	Serial Peripheral Interface
TCC	Trabalho de Conclusão de Curso

TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver/Transmitter
UEA	Universidade do Estado do Amazonas
URI	Uniform Resource Indetifier
URL	Uniform Resource Locator
USB	Universal Serial Bus

SUMÁRIO

INTRODUÇÃO

Não é de hoje que a tecnologia vem sendo utilizada para facilitar, otimizar e tornar mais eficiente as atividades realizadas diariamente pelos seres humanos. Logo, muitas dessas atividades que outrora eram realizadas manualmente e exigiam muito tempo e esforço passam a ser realizadas automaticamente de forma mais rápida e eficiente. E isso não se aplica apenas ao âmbito industrial, onde os desenvolvimentos tecnológicos estão muito presentes visando baratear e otimizar processos, mas também ao residencial e comercial, onde várias melhorias no dia-dia da humanidade são implementadas gerando através da tecnologia, de uma forma geral, um conforto para a vida de todos.

Entretanto, um novo conceito que envolve conectar objetos que são utilizados diariamente pela sociedade à rede através da internet, vem sendo muito utilizado, tal conceito é conhecido como internet das coisas (em inglês, *Internet of Things* - IoT). Cada vez mais o número de projetos desenvolvidos, no ramo científico e de desenvolvimento, nessa área tem aumentado significativamente, e pelo fato de existirem objetos conectados à rede, surge a possibilidade de fazer o acionamento e monitoramento remoto de equipamentos de acordo com a necessidade do usuário.

Com essa facilidade de acionamento e monitoramento remoto possibilitado pela internet das coisas e tendo em vista que tal otimização pode trazer economias financeiras consideráveis, suas aplicações estão muito voltadas para tonar mais eficientes as atividades de logística, tanto no que diz respeito ao âmbito residencial e comercial quanto ao industrial.

O crescimento da utilização da internet das coisas deve-se bastante por ser uma aplicação com custo relativamente baixo levando em consideração seu custo-benefício. E pelo fato da acessibilidade da internet para as pessoas ter aumentado consideravelmente tanto no que diz respeito às redes móveis quanto às fixas.

O problema da pesquisa deste trabalho é a atual logística de abertura e fechamento das salas de aula da Escola Superior de Tecnologia da Universidade do Estado do Amazonas que são feitas por meio de funcionários da instituição, o que caracteriza um trabalho manual, além de causar uma ineficiência no sistema, tendo em vista que o próprio professor que for utilizar o local pode realizar esse procedimento.

Além disso, pode-se abordar que em um considerável número de vezes, mesmo tendo-se encerrado a aula e não havendo mais alunos e nem professores dentro da sala, ela permanece com a iluminação e/ou refrigeração ligada, o que gera um gasto de energia desnecessário, algo

inaceitável ecologicamente, e que traz como consequência o aumento significativo da conta de energia elétrica da universidade.

Para conhecimento da dimensão deste problema na universidade, basta imaginar o número de salas que esta contém e que de segunda a sábado, durante o período letivo, estas são utilizadas com uma frequência constante de sete e trinta da manhã até dez e dez da noite, que são os horários em que são ministradas as aulas na faculdade. Tendo em vista que vários professores começam e terminam suas aulas no mesmo horário, pelo fato dos horários serem padrões para todos os cursos da universidade. O funcionário que realiza o procedimento de abertura e fechamento das salas é muito requisitado em determinados horários, portanto, haverá um atraso de abertura das salas de aula que serão abertas por último nesses momentos de pico. Sem levar em consideração que algumas vezes os professores terminam suas aulas antes do horário previsto, e pelo fato de não possuírem controle e de ser inconveniente ter que procurar o funcionário para desligar o ar-condicionado, deixam o mesmo ligado. E isso com certeza acontece várias vezes por dia, tanto no que diz respeito a refrigeração quanto a iluminação.

Logo, é possível desenvolver um sistema que faça o controle automatizado do acesso, iluminação e refrigeração das salas de aula da EST. Utilizando um banco de dados para armazenar informações cadastradas, referentes aos dias e horários das aulas de cada disciplina de todos os cursos pertencentes a universidade, com o nome dos professores que ministraram essas disciplinas, onde cada professor possuirá um cartão RFID com um código único. Logo, será possível através deste projeto fazer o controle de entrada e saída das salas de aula, bem como extinguir os momentos em que o uso da iluminação e da refrigeração são desnecessários.

O objetivo do trabalho é desenvolver e implementar um protótipo de um sistema de controle automatizado, utilizando Internet das coisas (IoT), nas salas de aula da EST, capaz de realizar remotamente a abertura e fechamento das salas, além de realizar remotamente o desligamento e acionamento da refrigeração e iluminação das mesmas, utilizando Internet das Coisas (IoT). A unidade de processamento foi composta por um leitor de cartão RFID juntamente com um sistema embarcado que foi conectado à um banco de dados via *Wireless*.

Vários sistemas são desenvolvidos utilizando internet das coisas para solucionar problemas de logísticas, ou seja, são projetos que visam facilitar tarefas o dia-a-dia das pessoas. O projeto desenvolvido tem justamente esse intuito, entretanto, além de facilitar determinado serviço, ele também o otimizou, tornando-o mais eficiente.

Além de que, este projeto permitirá a diminuição de transtornos referentes a perdas e esquecimentos de devolução das chaves das salas de aula e facilitará a abertura e fechamento das mesmas. Além de tornar desnecessário o funcionário da universidade ficar se locomovendo

dentro da mesma para ligar ou desligar o ar-condicionado e abrir ou fechar as portas das salas de aulas.

Outro ponto, é que este projeto irá fazer a automatização da iluminação e refrigeração das salas de aula de tal forma que as mesmas só ficarão ligadas enquanto o professor estiver em sala de aula, o que gerará vantagens financeiras e ecológicas, pois a conta de energia diminuirá e o uso desnecessário de energia elétrica reduzirá significativamente, tendo em vista que a refrigeração e iluminação frequentemente ficam ligadas sem ninguém estar utilizando e em alguns casos chegam a ficar ligadas de um dia para o outro.

Foi necessário utilizar vários conhecimentos, que foram adquiridos durante o curso de engenharia elétrica, para desenvolver o projeto. Logo, existiu a experiência de colocar boa parte deste conhecimento teórico obtido em uma aplicação prática.

As matérias estudadas durante o curso de engenharia elétrica que mais contribuíram para o desenvolvimento do projeto foram: Linguagem de Programação I e II, Circuitos Elétricos I, Eletrônica Digital I e II, Princípios de Comunicações I e II, Sinais e Sistemas, Sistemas Microprocessados, Processamento Digital de Sinais, Sistemas de Telecomunicações, Redes de Comunicações de dados I, Microcontroladores e Comunicações Digitais.

Diante disso, este plano de pesquisa sugere o desenvolvimento e implementação de um protótipo de um sistema de controle de acesso e acionamento da iluminação e ar-condicionado das salas de aula, utilizando internet das coisas.

Onde o tema do trabalho é: Desenvolvimento de um sistema de controle de baixo custo de acesso e acionamento da iluminação e ar-condicionado das salas de aula, utilizando internet das coisas (IoT).

O capítulo 1, referencial teórico, destina-se para esclarecimento de ferramentas e conceitos necessários para o desenvolvimento da solução. Consiste em uma revisão breve sobre: Internet das Coisas, sistemas embarcados, banco de dados, sensores e atuadores.

O capítulo 2 apresenta a metodologia, no qual é descrita o tipo de pesquisa realizada, os métodos de programação dos algoritmos utilizados, tanto no que diz respeito ao *software* quanto ao *hardware*.

O capítulo 3, desenvolvimento do projeto, tem a finalidade de descrever detalhadamente cada etapa apresentada na metodologia. Neste tópico é detalhado o desenvolvimento do circuito protótipo e a elaboração do código-fonte.

O capítulo 4, mostra os resultados obtidos, depois de concluir o desenvolvimento do sistema proposto.

1 REFERENCIAL TEÓRICO

Neste trabalho foram apresentados um problema e uma possível solução. Entretanto para o desenvolvimento desta solução é necessária uma pesquisa bem fundamentada teoricamente, realizando-se a ligação entre a bibliografia pesquisada e o projeto que está sendo desenvolvido. Logo, nesta parte do trabalho será apresentada o referencial teórico que foi estudado desde a escolha do tema do projeto até o estudo de como desenvolvê-lo.

1.1 INTERNET DAS COISAS

Desde 2009 a utilização da IoT vem crescendo muito, e pelas tendências dos projetos de desenvolvimento ela tende a crescer muito mais nos próximos anos. A quantidade de dispositivos conectados à internet das coisas será de aproximadamente 30 bilhões em 2020, segundo Balaguer (2014).

Portanto, pode se afirmar que o mercado de desenvolvimento de projetos utilizando a internet das coisas ainda está em grande ascensão e tende a permanecer assim por bastante tempo. Tendo em vista que o número de objetos conectados à internet das coisas só tende a aumentar no futuro.

A IoT nada mais é que a revolução tecnológica que tem como intuito conectar elementos que são utilizados no cotidiano da humanidade à rede mundial de computadores, e gradualmente maçanetas, roupas, tênis, meios de transporte e eletrodomésticos são conectados à internet e a outros dispositivos, como smartphones e computadores, segundo Zambarda (2014).

Logo, a internet das coisas surgiu com o intuito de facilitar o dia a dia das pessoas, seja no trabalho, em casa, no trânsito, entre outros lugares, onde a conexão das coisas com a internet permite o manuseio de itens a longa distância através de computadores ou smartphones controlados por um usuário, ou até mesmo cem por cento automatizados, sem a necessidade de um usuário para manusear o controle destes objetos. Além do fato de possibilitar o acionamento e monitoramento remoto. Entre outros tantos benefícios que o provento dessa nova tecnologia pode trazer.

Portanto, a Internet das Coisas está envolvida na interligação de objetos, sejam eles físicos ou virtuais, em redes que estejam conectadas à internet, possibilitando que dispositivos troquem, coletem e armazenem dados, para gerir informações e serviços por meio da análise e processamento desses dados. O que tornou possível a miniaturização e popularização de sensores, tornando viável a coleta e transmissão de dados, segundo Almeida (2015).

Geralmente o termo Internet das coisas é utilizado para o conjunto de sistemas, técnicas de design e tecnologias, que são a abordagem de desenvolvimento de “coisas” conectadas à internet que se baseiam no ambiente físico. Para que os dispositivos usados na rotina diária da sociedade possuam identificadores e conectividade à internet, onde esses dispositivos se comunicam uns com os outros. No geral, IoT é a situação em que dispositivos e sensores são fornecidos com conectividade de rede e capacidade de computação para consumir, gerar e trocar dados e se comunicar com a intervenção humana mínima, segundo Rose, Eldridge e Chapin (2015).

Segundo Aguiar (2018), a visão de conectar sensores e outros dispositivos ao sistema de tecnologia de informação e comunicação por meio das redes com fio ou sem fio pode ser aplicado em uma série de áreas no mundo como o setor de saúde, residencial, e cidades inteiras que, por sua vez, podem se tornar conhecidos como sistema inteligente de saúde, casa inteligente ou cidades inteligentes.

Segundo Martins (2018), a logística é um campo onde existem um elevado número de processos em que são fundamentais ponderações com a comercialização, produção e distribuição. Como nas indústrias as tarefas em sua maioria são feitas por meio de máquinas, possuir um sistema que permita monitoramento dinâmico e remoto consegue retratar um aumento significativo na produtividade e qualidade dos serviços.

Ou seja, os processos de logística tem sido alvo de incessantes projetos utilizando internet das coisas, sempre buscando otimizar as atividades, isto é, tonar mais eficientes processos já existentes, tendo em vista como um dos principais fatores a redução de custo. E sistemas utilizando internet das coisas são muito procurados para tornar tais atividades mais eficientes.

1.2 EFICIÊNCIA ENERGÉTICA

Segundo Abesco (2017), eficiência energética é uma atividade que busca melhorar o uso das fontes de energia, que consiste em utilizar de modo eficiente a energia para obter-se um determinado resultado. O conceito de eficiência energética consiste da relação entre a quantidade de energia empregada em uma atividade e aquela disponibilizada para sua realização.

Segundo INEE (2014), atualmente todas as atividades em uma sociedade moderna só são possíveis com uso intensivo de uma ou mais formas de energia. A energia é utilizada em aparelhos simples, como lâmpadas e motores elétricos, ou em sistemas mais complexos, como

são os casos das geladeiras, automóveis ou até mesmo das fábricas. Estes equipamentos e sistemas transformam formas de energia. Certa quantia dela sempre é perdida para o meio durante esse processo. Como exemplo, pode-se citar uma lâmpada que transforma a eletricidade em luz e calor. Logo, como o objetivo de uma lâmpada é iluminar, uma medida da sua eficiência é obtida pela razão entre a energia da luz e a energia elétrica usada pela lâmpada.

Ainda segundo INEE (2014), outro modo de desperdício vem do uso inadequado dos aparelhos e sistemas. No caso de uma sala de aula com as lâmpadas acesas sem ninguém estar dentro, também é um desperdício, devido ao fato de a luz não estar servindo ao seu propósito de iluminação. Outros fatores mais sutis explicam muitos desperdícios. Mas vale notar que esses efeitos aumentam significativamente conforme a energia vai migrando por todos os setores da economia.

O INEE (2014) também aponta que se desperdiça energia, porque uma lâmpada incandescente comum tem uma eficiência de 8% (ou seja, 8% da energia elétrica usada é transformada em luz e o restante aquece o meio ambiente). Já a eficiência de uma lâmpada fluorescente compacta, que produz a mesma iluminação, é da ordem de 32%. Mas como o preço de uma lâmpada eficiente é entre 10 a 20 vezes maior do que a comum, para saber qual delas comprar dependerá de fatores econômicos que levam em consideração a vida útil das lâmpadas e a economia proporcionada na conta de energia elétrica.

O Programa Nacional de Conservação de Energia Elétrica (PROCEL), foi criado para promover o uso eficiente de energia elétrica no Brasil, combatendo o desperdício, reduzindo os custos e os investimentos no setor (ELETROBRAS, 2018).

O PROCEL possui um selo, ilustrado na figura 1, que tem como objetivo ser uma ferramenta simples e eficaz que possibilita ao consumidor conhecer, entre os equipamentos e eletromagnéticos à disposição no mercado, os mais eficientes e que consomem menos energia. Para um produto conseguir o selo PROCEL, a Eletrobrás, junto com o INMETRO, submete o produto a ensaios específicos em um laboratório idôneo, indicado pelo PROCEL, de acordo com os parâmetros do programa. Cada linha de produto tem um conjunto de parâmetros a ser analisado.

Figura 1– Selo PROCEL

Fonte: ELETROBRAS (2018).

Os produtos que possuem o Selo PROCEL são caracterizados pela faixa “A” da Etiqueta Nacional de Conservação de Energia (ENCE), ilustrada a figura 2. A ENCE faz parte do Programa Brasileiro de Etiquetagem (PBE) e é concedida pelo Instituto Nacional de Metrologia, Normalização e Qualidade Industrial (INMETRO). A ENCE foi criada para prestar informações sobre a eficiência energética dos equipamentos disponíveis no mercado nacional e contribui para a racionalização de energia no país estimulando o consumidor a fazer uma compra mais consciente.

Figura 2 – Etiqueta Nacional de Conservação de Energia (ENCE)

Fonte: ELETROBRAS (2018).

1.3 SISTEMAS EMBARCADOS

Segundo Garcia (2018), a tecnologia está cada vez mais presente em todas as atividades realizadas no dia a dia do ser humano. Foi-se comtemplado nas últimas décadas um grande avanço dos computadores pessoais, da eletrônica, dos equipamentos eletrônicos, entre outros. Nesta realidade os sistemas embarcados tornam-se cada vez mais essenciais e presentes na vida da humanidade. Devido essa necessidade bilhões de sistemas computacionais são produzidos por ano e aplicados em produtos de vários ramos diferentes.

Segundo Chase (2007), para projetar em sistemas embarcados o primeiro passo é a escolha do sistema embarcado, podendo ele ser um microcontrolador ou um microprocessador. Depois são projetadas as suas dimensões físicas, tendo em vista o tamanho e peso do sistema em desenvolvimento, o recomendado é sempre buscar os menores possíveis, pois com a tendência dos equipamentos, em sua maioria, serem cada vez menores, o tamanho e o peso se tornam decisivos tanto na locomoção quanto na competitividade caso se torne um produto. Depois o projetista deve levar em consideração o consumo de energia elétrica do sistema, pois quanto maior a autonomia do sistema e menor for sua necessidade de recarga, troca de sistema de alimentação ou baixo consumo elétrico, mais competitivo será o produto. E o outro ponto que o projetista deve levar em consideração é a resistência e durabilidade, tendo em vista, que vários sistemas embarcados são feitos para trabalhar em ambientes com condições adversas (poeira, calor, variações na tensão de alimentação, vibrações, interferências eletromagnéticas, umidade, entre outros).

Ainda segundo Chase (2007), na escolha do microcontrolador que irá compor o sistema embarcado é fundamental estudar se ele é capaz de atender a necessidade do projeto, por exemplo, num projeto onde existe a necessidade de um sensor de temperatura como o LM35 com sinal de saída analógico de 10 mV por grau centígrado é preciso que o microcontrolador possua um conversor A/D (Analógico/Digital) para o recebimento e processamento de sinais e, enviar o sinal convertido para um display LCD, por exemplo.

Segundo Aguiar (2018), inicialmente as funções dos microcontroladores se resumiam a interface de entrada e saída (I/O), posteriormente foram acrescentando, a cada nova versão ou produto, memória RAM, memória EPROM para programas e dados e circuito de oscilador (*clock*), interfaces de comunicação (serial, USB) e, mais recentemente, interfaces de rede, *Ethernet*, WI-FI e *Bluetooth*. A junção de todas essas funcionalidades em um único componente torna a sua integração muito barata e acessível, o que justifica sua implementação em dispositivos de baixo custo, como uma fechadura, por exemplo.

Em seguida, nos capítulos 1.3.1 e 1.3.2, são abordados brevemente sobre duas plataformas onde será possível desenvolver o projeto deste trabalho. Entretanto, será utilizada apenas uma dessas plataformas que é o ESP32, porém essa possui muitas semelhanças com o outro sistema embarcado que será citado que é o Arduino. A semelhança e a compatibilidade entre ambos são tamanhas que a IDE do Arduino, um software livre onde se escreve o código na linguagem que o Arduino compreende é compatível com o ESP32, ou seja, é possível utilizar a IDE do Arduino para programar o ESP32 na mesma linguagem em que o Arduino é programado. Por isso no referencial teórico desta pesquisa é abordado sobre essas duas plataformas.

1.3.1 Arduino

Segundo Roberts (2011), a principal vantagem da plataforma de desenvolvimento Arduino sobre as outras é a facilidade de sua utilização, mesmo pessoas que estudam áreas diferentes do segmento de desenvolvimento de projetos tecnológico utilizando hardwares conseguem aprender o básico e desenvolver seus projetos em um curto espaço de tempo. Além disso, existe um grupo muito grande de pessoas que utilizam Arduinos, e compartilham gratuitamente seus códigos e diagramas de circuitos, assim permitindo que outras pessoas copiem e modifiquem o que ela já fez. E essas pessoas estão muito dispostas a auxiliar outros desenvolvedores. Porém, apesar do grande número de informações disponíveis na internet, grande parte delas estão espalhadas em várias fontes, fazendo com que seja difícil encontrar as informações desejadas.

Ainda segundo Roberts (2011), para realizar a programação no Arduino utiliza-se a IDE, *Integrated Development Environment*, do Arduino, um software livre onde se escreve o código na linguagem que o Arduino compreende, ela é baseada na linguagem C, C++ e Java.

Ou seja, o Arduino é uma plataforma de desenvolvimento de sistemas embarcados de baixo custo aberta e livre. As referências a ele consideram, normalmente, uma placa integrada com um microcontrolador e suas interfaces de entrada e saída, alimentação e comunicação. Antes dessa plataforma, desenvolver um sistema embarcado exigia compra de um kit de desenvolvimento menos acessível, com recursos para programação e gravação dos microcontroladores, além da necessidade de construir uma placa para acomodar os componentes. As placas desse embarcado vêm prontas para utilização e integração e tem baixo custo e o ambiente de desenvolvimento torna a tarefa mais fácil. Essa plataforma é muito utilizada em aplicações que envolvem diretamente controle sobre periféricos, sem exigir muito

do processamento, como realizar leituras de sensores e controlar determinada carga, tendo em vista que seu processador não é tão potente.

Uma placa Arduino, em geral, é composta por um microprocessador Atmel AVR, um cristal ou oscilador, que envia pulsos para permitir a operação na velocidade correta, um regulador de tensão de 5 V, entradas e saídas digitais e analógicas e uma entrada USB, por onde a programação é realizada. Diversos dispositivos são usados de modo a complementar a aplicação, são eles: display, sensores, modo bluetooth, etc. Além disso, utilizando os Shields, que são placas de circuitos contendo outros dispositivos, é possível combinar um ou mais Arduinos.

Como pode-se perceber, utilizando-se o Arduino existirá uma certa facilidade para o desenvolvimento do projeto, tendo em vista que já existem vários projetos que foram desenvolvidos utilizando o Arduino, e estes estão disponíveis na internet, e podem ser usados como base para o que se planeja desenvolver neste trabalho. Além do fato da linguagem para programar o Arduino, ser uma linguagem baseada em outras que são muito utilizadas no que diz respeito a programação, o que acaba se tornando mais um facilitador para a utilização do Arduino. A figura 3 é a imagem de um modelo do Arduino, o Arduino Uno.

Figura 3 – Placa Arduino Uno

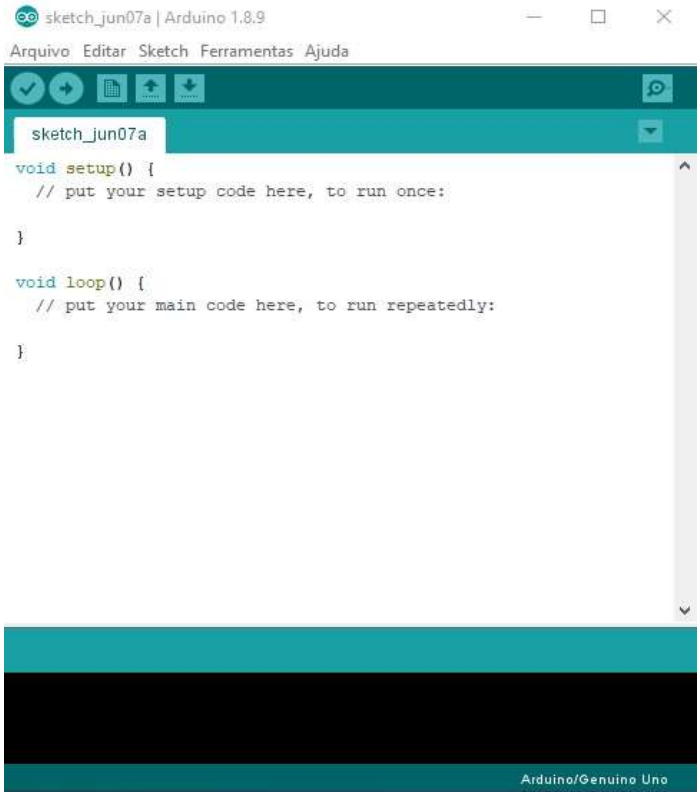
Fonte: Própria.

O Arduino uno pode ser considerado o mais versátil, pois utiliza um chip padrão de 28 pinos, ligado a um soquete de circuito integrado. A vantagem desse sistema é que é possível retirar o chip da placa e inseri-lo em uma placa de circuito personalizado, caso deseje

transformar alguma coisa do Arduino em algo permanente. Assim, é possível obter um dispositivo embarcado personalizado.

O Arduino possui uma IDE própria, como foi citado anteriormente, que está sendo mostrada na Figura 4, ela é 100% virtual e nela será feita a digitação do algoritmo de programação. A IDE possui acesso a destaque de sintaxe, envio de código, correção de erros, inclusão de bibliotecas (conjuntos de funções prontas), monitor serial (usado para a comunicação com a placa), entre outros. Através da IDE é possível editar o programa na linguagem de alto nível, compilá-lo para a linguagem do microcontrolador, carregar o código compilado para a memória e, por meio das interfaces de comunicação, realizar testes.

Figura 4 – Interface da IDE do Arduino



Fonte: Própria.

1.3.2 ESP32

O ESP32 possui o microprocessador Xtensa 32-Bits LX6 de baixo consumode energia, que tem dois núcleos de CPU (processador dual core) podendo ser controlados individualmente com frequência de *clock* ajustável de 80 MHz à 240 MHz. Também é possível desligar a CPU e utilizar o coprocessador de baixa potência, sendo possível monitorar constantemente os

periféricos (sensores, botões, teclados, interface I2C, SPI, UART, CAN, entre outros) com baixo consumo de energia.

Além da poderosa capacidade de processamento, a integração com o *Wi-Fi* e *Bluetooth* LE (baixo consumo de energia) garante a aplicação do módulo em uma variedade de projetos, principalmente aos projetos voltados à Internet das Coisas. O *Wi-Fi* suporta uma taxa de 150 Mbps e potência de 20,5 dBm na antena, permitindo uma ampla faixa física e conexão direta à internet. O *Bluetooth* LE permite que qualquer celular se conecte a ele enviando diversos tipos de informação com baixo consumo de energia.

A placa DOIT ESP32 DevKit V1, ilustrada na figura 5, com o módulo ESP32 é de fato muito poderosa, tento em sua capacidade de processamento, dispensando assim o uso de um microcontrolador extra como o Arduino, quanto na sua capacidade de memória (RAM e Flash), interfaces de comunicação, quantidade de GPIO, quantidade de canais de conversor analógico digital, sensores internos, por possuir *Wi-Fi* e *Bluetooth* BL, além é claro de já possuir o conversor USB/Serial CP2102, LED's de indicação de funcionamento, regulador de tensão AMS1117 para que seja possível alimentar a mesma com tensão de 12 V. Ou seja, a placa DOIT ESP 32 é uma plataforma completa para desenvolvimento de novos projetos voltados à Internet das Coisas. Suas principais características são:

- Módulo ESP-WROOM-32 (Datasheet ESP-WROOM-32);
- Chip Base: ESP32-D0WDQ6 (Datasheet ESP32);
- Processador: Xtensa 32-Bit LX6 Dual Core;
- Clock: 80 à 240 MHz (Ajustável);
- Memória ROM: 448 Kb;
- Memória SRAM: 520 Kb;
- Memória Flash Externa: 32-Bit de acesso e 4 Mb;
- Tensão de Alimentação: 4,5 à 12,0 VDC (Pino V_{in});
- Tensão de nível lógico: 3,3 VDC (não tolera 5 V);
- Corrente de consumo: 80 mA (típica);
- Corrente de consumo: 500 mA (máxima);
- Interfaces: Cartão SD, UART (3 canais), SPI (3canais), SDIO, I2C (2 canais) e IR.
- Tipos GPIO: Digital IO (36), ADC 12-Bits (16 canais), DAC 8-Bits (2 canais), Sensor Capacitivo (10 canais); LNA pré-amplificador;
- WiFi 802.11b/g/n: 2.4 à 2.5 GHz;
- Segurança WiFi: WPA /WPA2 /WPA2-Enterprise/WPS;

- Criptografia WiFi: AES/ RSA /ECC / SHA;
- *Bluetooth* 4.2 BR / EDR e BLE (Bluetooth Low Energy);
- RTC Integrado de 8 Kb (Slow / Fast);
- Sensor integrado: Temperatura e Hall;
- Temperatura de trabalho: -40º à +85ºC;
- Compatível com a IDE do Arduino;
- Dimensões: 27,5 x 51,0 x 7,0 mm;

Figura 5 – DOIT ESP32 DevKit V1

Fonte: Schwartz (2017).

Uma das principais vantagens desse módulo é realizar comunicação sem fio através de uma rede local disponível. E outro ponto é que a implementação da placa DOIT ESP32 DevKit V1 possui uma logística simples devido ela ser pequena.

1.4 HTTP

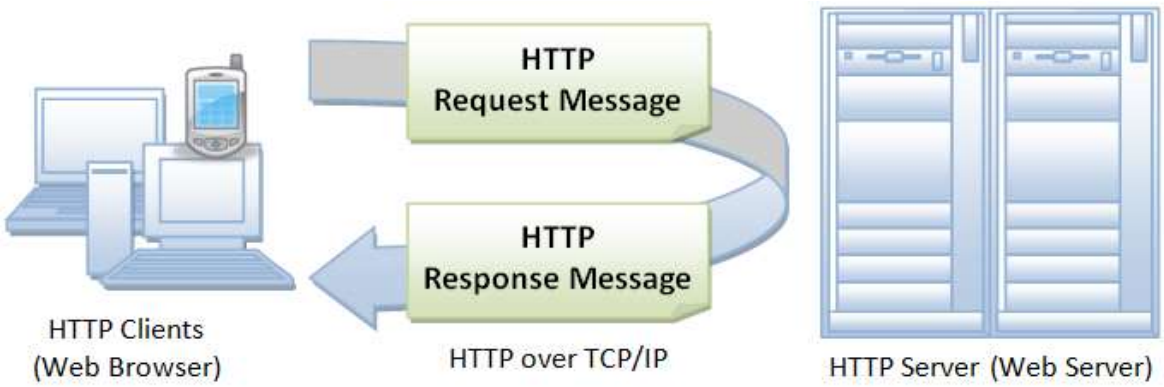
O HTTP, *Hypertext Transfer Protocol*, é um protocolo de comunicação. Por meio dele o cliente e o servidor conseguem se comunicar, obedecendo um conjunto de regras. Além disso, é o método utilizado para enviar e receber informações na *Web*. Por exemplo, em uma aplicação *Web*, o cliente é o navegador, ele envia um pedido para o Servidor *Web* usando o protocolo HTTP, com base nesse pedido, se tudo estiver correto, o servidor responde também usando o mesmo protocolo o conteúdo solicitado.

O protocolo HTTP é baseado em requisições e respostas entre clientes e servidores. O cliente fará a requisição solicitando um determinado recurso, enviando um pacote de

informações contendo alguns cabeçalhos a um URI ou, mais especificamente, URL. O servidor recebe estas informações e envia uma resposta, que pode ser um recurso ou simplesmente um outro cabeçalho. O protocolo HTTP é *stateless*, ou seja, ele não é capaz por si só de reter informações entre requisições diferentes. Para persistir informações você precisa utilizar cookies, sessões, campos de formulário ou variáveis na própria URL.

A Request é o pedido que um cliente realiza ao servidor. Esse pedido contém uma série de dados que são usados para descrever exatamente o que o cliente precisa. A Request que o cliente envia ao servidor possui todas as informações acerca do que o cliente espera receber de volta. O servidor Web ao receber essas informações precisa enviar uma resposta ao cliente, nesse ponto entra a Response. A Response nada mais é do que a resposta que o servidor envia ao cliente. Essa resposta pode conter os dados que realmente o cliente esperava receber a uma resposta informando que alguma coisa deu errado. Para que a transferência de dados na internet seja realizada, o protocolo HTTP necessita estar associado a outros dois protocolos TCP (*Transmission Control Protocol*) e IP (*Internet Protocol*). Estes dois últimos protocolos foram o modelo TCP/IP, necessário para a conexão entre computadores clientes-servidores. A figura 11 ilustra o princípio de funcionamento do protocolo HTTP.

Figura 6 – Protocolo HTTP



Fonte: Vieira (2007).

Quando uma requisição irá ser feita, é preciso especificar qual método será utilizado. Os métodos HTTP identificam qual ação deve ser executada em um determinado recurso, dois desses métodos são: o GET e o POST. Onde o GET é utilizado quando o cliente deseja obter recursos do servidor. E o POST é utilizado quando o cliente deseja enviar dados para processamento ao servidor, como os dados de um formulário, por exemplo.

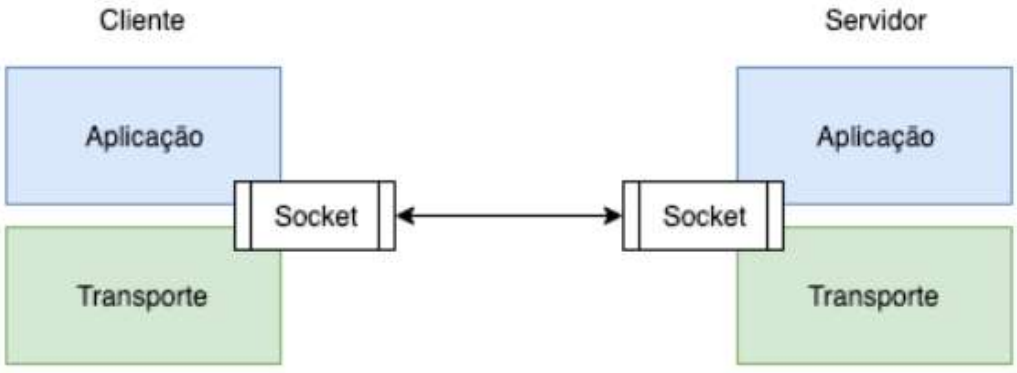
1.5 SERVIDOR WEB

Segundo Andrade (2019), tendo em vista o *hardware*, um servidor *Web* é um computador que armazena arquivos que justos compõe os sites, como documentos HTML, e os entrega para o dispositivo do usuário final. Tomando como base o *software*, determinado servidor *Web* inclui diversos componentes que controlam a maneira como os usuários acessam os arquivos hospedados, armazenados para disponibilização, no mínimo um servidor HTTP, que é um *software* que compreende URLs, endereços *Web*, e HTTP, o protocolo que seu navegador utiliza para visualizar páginas *Web*. Para publicar um *Website* é preciso um servidor *Web* estático ou um dinâmico. Um servidor *Web* estático consiste em um computador com um servidor HTTP, e é chamado de "estático" devido o servidor enviar seus arquivos tal como foram criados e armazenados ao navegador. Um servidor *Web* dinâmico nada mais é que um servidor *Web* estático com *software*, mais comumente um servidor de aplicações e um banco de dados, e é chamado de dinâmico devido o servidor de aplicações utilizar os arquivos hospedados antes de enviá-los ao navegador através do servidor HTTP.

O Socket provê a comunicação entre duas pontas (fonte e destino) entre dois processos que estejam na mesma máquina ou na rede (TCP/IP Sockets). Na rede, a representação de um Socket se dá por ip:porta, por exemplo: 127.0.1:447 (IPv4). Um Socket que usa rede é um Socket TCP/IP.

O navegador, que é utilizado no dia a dia pela sociedade, utiliza Sockets para requisitar as páginas, quando o servidor é acessado pelo protocolo de aplicação SSH também se abre e utiliza-se um Socket. Considerando o modelo de rede OSI, os Sockets estão entre a camada de aplicação e a de transporte, como está ilustrado na figura 6.

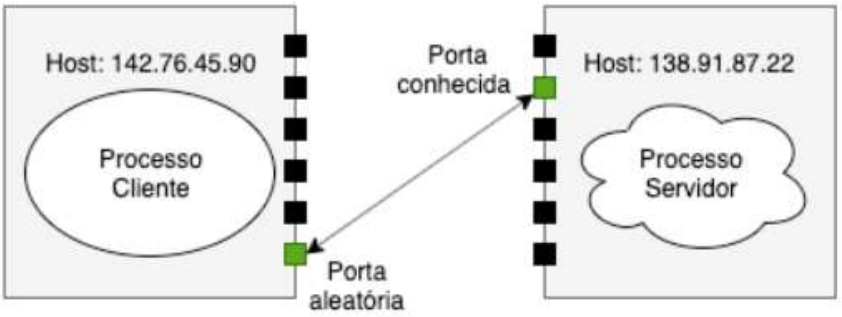
Figura 7 – Posição Socket no modelo de rede OSI



Fonte: Monk (2016).

Para os processos envolvidos a sensação é que a comunicação está acontecendo diretamente entre eles, porém, ela está passando pelas camadas de rede. Essa abstração provida pelos Sockets é o que se chama de comunicação lógica. Qualquer cliente deve conhecer o Socket do servidor (conjunto ip e porta) para se comunicar, mas o servidor só vai conhecer o socket do cliente quando este realizar uma conexão com ele, ou seja, a conexão no modelo cliente-servidor é sempre indicada pelo cliente. O diagrama, ilustrado na figura 7, mostra que a porta do servidor precisa ser previamente conhecida pelo cliente, enquanto que para o servidor não importa qual é a porta do cliente, ele vai conhecê-la quando a conexão dele com o cliente for estabelecida.

Figura 8 – Comunicação entre cliente e servidor



Fonte: Monk (2016).

1.6 RFID

Segundo Pais e Couto (2009), ainda que o RFID não seja uma tecnologia nova, ele teve um elevado desenvolvimento no que diz respeito aos últimos anos, e acredita-se que para os próximos, ele estará muito presente na vida de usuários comuns. Os avanços recentes na miniaturização de chips e nos processos produtivos possibilitou o barateamento dessa tecnologia a níveis toleráveis. Diversas aplicações já se fazem presentes no cotidiano da sociedade e várias outras já estão ajudando empresas a reduzir seus custos.

Ainda segundo Pais e Couto (2009), basicamente, o RFID é um procedimento sem fio que possibilita atribuir uma identidade única a um objeto através de uma etiqueta. Os sistemas são formados: por um *transponder*, que é a própria etiqueta, por dispositivos de leitura que implementam a comunicação e por um sistema de dados que permite o acesso a todas as informações relacionadas ao objeto identificado. Tal comunicação ocorre por ondas de rádio que levam a informação que portam os dados uni ou bidirecionalmente, dependendo da

tecnologia a ser usada. E o *transponder* quando entra na zona de leitura de um leitor, captura sua identidade, e esta é passada para o dispositivo de interesse para a ação.

A utilização do RFID se encaixa muito bem neste projeto tendo e vista a necessidade do sistema a ser desenvolvido que é de atribuir uma identidade única a cada professor, visando conseguir identificar cada professor com um código diferente, e saber exatamente qual deles está utilizando o cartão RFID. A seguir, será mostrada, na Figura 8, a imagem de um modelo de leitor RFID o módulo RFID MFRC522, com os seus respetivos conectores e cartão e chaveiro RFID.

Figura 9 – Módulo leitor RFID MFRC522, e cartão e chaveiro RFID

Fonte: Cruzes (2015).

1.7 DIODO EMISSOR DE LUZ (LED) INFRAVERMELHA

Um diodo emissor de luz (LED) infravermelha, ilustrado na figura 9, é um tipo de dispositivo eletrônico que emite luz infravermelha não visível a olho nu. Um LED infravermelho funciona como um diodo emissor de luz normal, mas pode utilizar diferentes materiais para produzir a luz infravermelha. Esta luz infravermelha pode ser utilizada em um controle remoto, na transferência de dados entre os dispositivos.

Figura 10 – LED Infravermelho TIL32 5mm

Fonte: Adafruit (2018).

Um LED infravermelho é, como todos os LED's, um tipo de diodo ou um semicondutor simples. Os diodos são projetados de modo que a corrente elétrica só possa fluir em uma direção. Como os fluxos de corrente, os elétrons saem de uma parte do diodo para outra parte. Para isso, os elétrons devem lançar energia na forma de fótons, que produz luz.

O comprimento da onda e a cor da luz emitida dependem do material utilizado no diodo. LED's infravermelhos utilizam o material que produz luz na parte infravermelha do espectro, ou seja, um pouco abaixo do que o olho humano pode ver. Diferentes LED's infravermelhos podem produzir luz infravermelha de comprimento de onda diferentes, como inúmeros LED's produzem luz de cores variadas.

1.8 RELÉ

O relé, ilustrado na figura 10, é um dispositivo eletromecânico, formado por um magneto móvel, que se desloca e une dois contatos metálicos. Seu funcionamento consiste basicamente em uma corrente que circula pela bobina, gerando um campo magnético que atrai um ou mais contatos, abrindo ou fechando circuitos. Segundo Santos (2015), em geral, a bobina de um relé é formada por um fio em torno do núcleo de aço maciço e esse núcleo fornece um caminho de baixa relutância para o fluxo magnético. Enquanto se manter desenergizada, a força das molas mantém os contatos na sua posição de repouso.

Figura 11 – Módulo Relé 5 V

Fonte: Tectronics (2019).

Ainda segundo Santos (2015), o estado inicial é definido de acordo com a configuração do relé, seja normalmente aberto (NA), ou normalmente fechado (NF). Quando a bobina recebe uma corrente elétrica, a armadura do relé movimenta-se em direção ao núcleo, atraída pelo campo magnético que foi gerado permitindo a movimentação do contato. No momento em que a força magnética proveniente da circulação da corrente se torna superior à força das molas, o contato é atraído, assim saindo da sua posição de repouso e mudando seu estado. Logo, o contato que é normalmente aberto comuta para o contato fechado e o contato que era normalmente fechado passa a ser contato aberto. Ao interromper a corrente da bobina o campo magnético também é interrompido, permitindo que os contatos voltem para a posição inicial.

A maioria dos relés são fabricados para funcionar rapidamente. Isso ocorre para diminuir o ruído em aplicações de baixa tensão e para reduzir a formação de arcos em aplicações de alta tensão ou corrente elevada.

1.9 BANCO DE DADOS

Segundo Oliveira (2017), sistemas de gerenciamento de bancos de dados são *softwares* especiais que gerenciam armazenamento de dados. Os quais são normalmente organizados em tabelas, onde são feitos relacionamentos de acordo com a lógica de organização dos dados. O projeto de bancos de dados envolve várias fases: modelagem entidade-relacionamento, álgebra relacional e descrição em linguagem SQL. A modelagem entidade-relacionamento permite identificar quais informações devem ser armazenadas e como elas se relacionam, como é sua cardinalidade e quais informações caracterizam unicamente um elemento em um conjunto de

dados. Essa etapa gera como resultado um diagrama chamado entidade-relacionamento, que indica a estrutura geral de organização dos dados. A outra etapa do projeto de banco de dados é construir a lógica de organização e localização de dados. Essa fase facilita a representação dos comandos em linguagem SQL, que seria a próxima etapa.

Ainda segundo Oliveira (2017), A linguagem SQL (*Structured Query Language* – Linguagem de Consulta Estruturada) é usada para acessar os sistemas de gerenciamento de bancos de dados. Conta com subconjuntos da linguagem para funções específicas, como a DML (Linguagem de Manipulação de Dados) e a DDL (Linguagem de Definição de Dados). A DML possui a parte da linguagem que possibilita localizar, inserir ou atualizar os dados. A DDL possui a parte da linguagem que define a estrutura dos dados e das tabelas.

A plataforma Arduino contém bibliotecas para acesso ao MySQL, um SGBD gratuito e disponível em diversos provedores. Isso é uma grande vantagem dessa plataforma, que pode ser usada também para programar o ESP32. Usando a biblioteca disponível, é possível enviar comandos diretamente a um servidor MySQL.

2 METODOLOGIA

O trabalho apresentado é uma pesquisa aplicada, onde o objetivo é a realização de uma pesquisa exploratória tendo como base o material bibliográfico e de laboratório apanhado sobre o assunto. Os procedimentos técnicos utilizados foram de pesquisa bibliográfica e experimental. Os métodos utilizados foram os de: abordagem hipotético-dedutivo e o de procedimento monográfico em sua elaboração. Para coleta de dados foi utilizada a documentação direta intensiva, a documentação indireta e a análise e interpretação de seus dados, qualitativos, ocorreu globalmente.

Foram realizadas pesquisas bibliográficas nas áreas de Linguagem de Programação, Eletrônica Digital e Analógica, Sistemas de Telecomunicações, Redes de Computadores, Microcontroladores e Comunicações Digitais. Pesquisas em artigos, *sites* da internet e em outros TCCs realizados na área de Internet das Coisas, Eficiência Energética, Sistemas Embarcados e a utilização de RFID para otimização de atividades.

O sistema foi composto por três tipos de cargas, sendo elas as lâmpadas, os aparelhos de ar-condicionado e as fechaduras eletrônicas. Tendo em vista que ao entrar na sala o professor necessita acionar primordialmente essas três cargas, que dizem respeito a iluminação, refrigeração e fechadura eletrônica, respectivamente, e ao sair ele precisa desativar as mesmas.

Essas cargas neste projeto foram controladas por um sistema embarcado, o DOIT ESP32 Dev Kit V1, ele foi escolhido por possuir vários periféricos e características que foram utilizadas para o desenvolvimento do projeto, como será abordado no decorrer do trabalho, em seus pinos digitais foram conectados os seguintes sensores e atuadores: leitor RFID, diodo emissor de luz infravermelha e módulo relé de 2 canais. E estes pinos por sua vez podem ser programados para serem de entrada ou saída e são responsáveis por receber o sinal provido da leitura do cartão RFID, quando atuam como porta de entrada e de enviar um sinal de nível lógico alto ou baixo, quando operam como porta de saída, que aciona ou desativa o ar-condicionado, lâmpada e a fechadura eletrônica. Vale salientar que não necessariamente o ESP lê ou escreve um único sinal analógico através de suas portas digitais, ele também pode receber ou enviar por meio destas uma sequência de sinais digitais.

A utilização do RFID se encaixa muito bem neste projeto tendo em vista a necessidade do sistema a ser desenvolvido que é de atribuir uma identidade única a cada professor, visando conseguir identificar cada professor com um código diferente, e saber exatamente qual deles está utilizando o cartão RFID.

Mas o ESP32 não teve no projeto apenas a função de controlar as cargas, mas também foi responsável por estabelecer uma comunicação com o servidor, através de uma rede local. E para estabelecer as regras dessa comunicação foi-se escolhido o protocolo HTTP, devido ao fato dele ser um protocolo padrão para a navegação *Web*, através dele os navegadores solicitam páginas na internet e tratam do seu recebimento. E como é citado posteriormente o banco de dados, o servidor do projeto, é online, ou seja, quando o ESP envia a informação para o banco de dados, ele está enviando a informação para uma página web que possui um link, onde esse link é o "endereço" do banco de dados online que está sendo utilizado no projeto.

A plataforma na qual foi desenvolvida o banco de dados é a Back4App que fornece as ferramentas para desenvolver um *back-end* completo incluindo um banco de dados com API, que foi exatamente o que foi desenvolvido no projeto, um banco de dados, onde foram armazenadas as informações necessárias para o desenvolvimento do sistema. Como foi citado anteriormente o ESP32 estabelece uma comunicação com o banco de dados na qual se foi utilizado o protocolo HTTP, onde o ESP32 foi o cliente e o banco de dados o servidor. E para enviar informações do ESP32 para o banco de dados é feita uma requisição, e o servidor envia de volta uma resposta com o que foi solicitado pelo cliente ou uma mensagem de erro caso a comunicação falhe. Vale salientar que a comunicação sempre ocorre com o cliente enviando uma requisição ao servidor, e o servidor enviando uma resposta ou mensagem de erro em reação a requisição enviada. Ou seja, quem sempre começa a comunicação cliente-servidor é o cliente, e o servidor somente envia uma resposta quando é feita uma requisição pelo cliente. Através dessa comunicação é possível enviar informações do ESP para o banco de dados, e essas informações serem comparadas com informações já cadastradas no banco de dados, e o ESP obter uma resposta do banco de dados, positiva ou negativa em relação a essa comparação, o que foi bem útil para o desenvolvimento do sistema.

Para serem cadastradas as informações no banco de dados foi feito todo um trabalho de *front-end*, de tal forma que o cliente não precise cadastrar informações referentes ao nome do professor, código RFID, número da sala, horário e o dia da semana em que ocorrerá a aula diretamente no banco de dados, ao invés disso criou-se um *site* que tornou mais interativo e intuitivo o cadastro de todas essas informações que são necessárias para o sistema. Portanto, toda vez em que forem cadastradas informações no site desenvolvido para este projeto, as mesmas serão automaticamente e instantaneamente cadastradas também no banco de dados. Tendo em vista que da mesma forma que foi criada uma comunicação entre ESP e banco de dados, também se criou uma comunicação entre banco de dados e *site*, onde o banco de dados é o servidor e o site o cliente, e essa comunicação também foi criada utilizando o protocolo

HTTP. A comunicação cliente-servidor neste caso se inicia quando o site faz uma requisição, enviando os dados a serem cadastrados no banco de dados, o banco de dados por sua vez recebe a informação, a processa e faz o cadastramento delas, e posteriormente envia uma resposta ao site positiva ou negativa, no caso da resposta positiva significa que foi possível cadastrar todas as informações enviadas na requisição que o *site* fez, e no caso de ser negativa significa que não foi feito o cadastramento de todas as informações com sucesso. E de uma forma geral é assim que acontece a comunicação entre *site* e banco de dados neste projeto, respeitando as regras do protocolo HTTP.

A criação do *site* aumenta significativamente a aplicabilidade do projeto, tendo em vista que poucas pessoas tem a habilidade de manusear um banco de dados mesmo no que diz respeito a fazer um simples cadastro de informações diretamente no banco de dados. Entretanto com a enorme acessibilidade e desenvolvimento da internet ultimamente, a maior parte da população possui a habilidade de manusear *sites*, ainda mais quando estes são de fácil manuseio e bem interativos, como é o caso do site desenvolvido para esse projeto. A preocupação em fazer um *site* interativo para o usuário foi tamanha que na hora de cadastrar informações no *site*, o usuário não precisa digitar nenhuma informação, todas elas já estão pré-cadastradas no próprio código-fonte do *site*, e basta o usuário escolher e clicar na informação que deseja cadastrar. Esse procedimento além de tornar o *site* mais intuitivo, também diminui significativamente a possibilidade de informações erradas serem cadastradas pelo usuário e implicar no funcionamento correto do sistema desenvolvido para este projeto.

Para entender como funciona o sistema deste projeto, posteriormente será abordado um exemplo de uma situação prática, onde o sistema está atuando. O início do processo do sistema acontece, por exemplo, quando um professor está tentando entrar em uma sala de aula e encosta o seu cartão RFID no leitor e então será feita a leitura do cartão RFID, onde essa informação será processada pelo ESP32 e este enviará uma requisição para o servidor contendo informações referentes ao nome do professor, código RFID do cartão que está sendo lido, a sala na qual a operação está ocorrendo, horário (hora e minuto) e dia da semana em que a leitura do cartão RFID está acontecendo. Onde o banco de dados receberá essa mensagem e irá compara-las com informações já cadastradas no banco de dados, por exemplo, se um professor que ministra aulas na sala A4 às 16:20 às 18:00 horas nas segundas e quartas-feiras, esses dados de data, hora, minuto e número de sala estarão cadastrados no banco de dados juntamente com o nome do professor e o seu código RFID, cada professor possui um código RFID único. Essa comparação irá gerar uma resposta positiva ou negativa, caso as informações cadastradas coincidam a resposta será positiva, senão ela será negativa. Ou seja, caso esse professor ministre aula nessa

sala, nesse dia e horário, a resposta que o banco de dados enviará para o ESP será positiva, caso contrário ela será negativa. Caso esta resposta seja positiva o ESP irá mandar um nível lógico baixo para o módulo relé de dois canais, tendo em vista que esses relés são ativos em níveis lógicos baixo e desativados em níveis lógicos alto, que por consequência irão destrancar a fechadura eletrônica e ativar a iluminação, e uma sequência de níveis lógicos para o diodo emissor de luz infravermelha que ocasionará a ativação da refrigeração. Caso a resposta seja negativa o ESP não mandará nenhum nível lógico ou sequência de nível lógico para estes componentes eletrônicos, citados anteriormente, e consequentemente a fechadura continuará trancada e a refrigeração e iluminação desativadas. E na hora em que a aula acabar e o professor for embora, basta ele encostar o cartão RFID dele no leitor que o ESP identifica que o professor está saindo da sala de aula e manda um nível lógico alto para o módulo relé de dois canais, fazendo com que a fechadura eletrônica fique trancada e com que a iluminação seja desativada e envia uma sequência de níveis lógicos para o diodo emissor de luz infravermelha, dessa forma fazendo com que a refrigeração seja desativada.

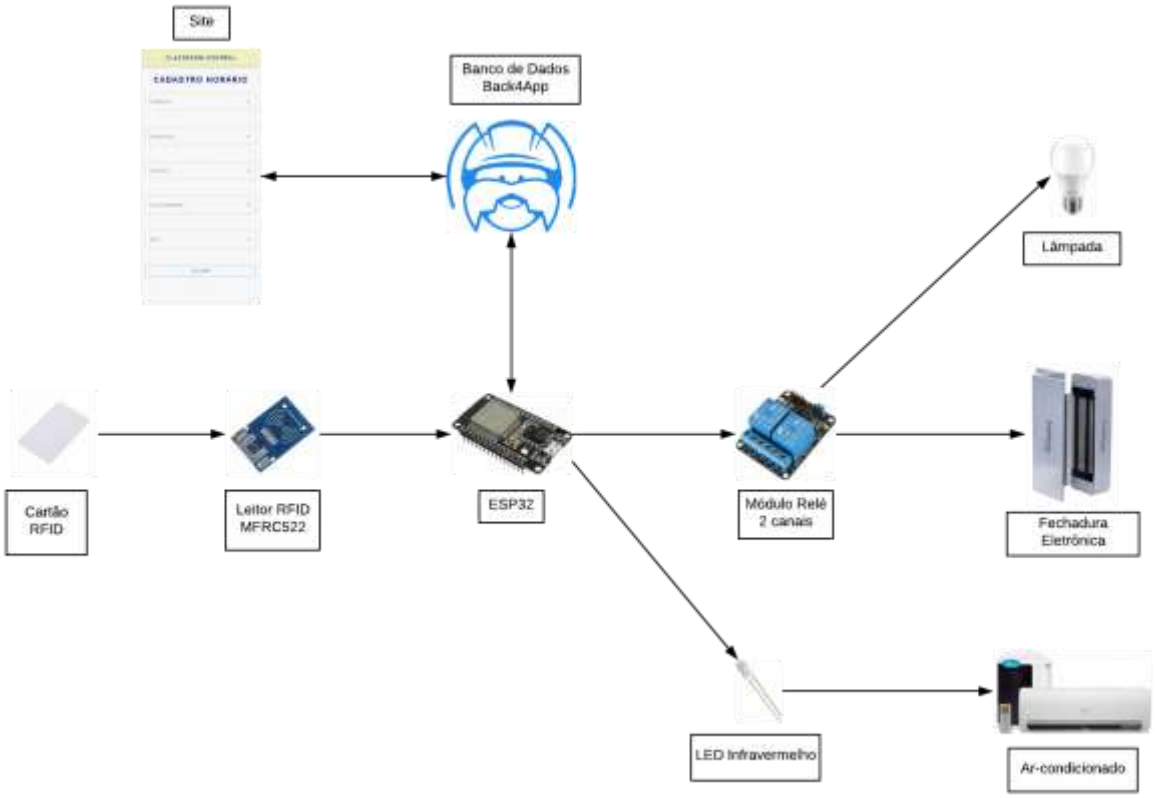
Para entender melhor o funcionamento do projeto elaborado neste trabalho, vamos supor o caso citado anteriormente do professor que ministra aulas na sala A4 às 16h20 às 18h nas segundas e quartas-feiras. Considerando que ele chegou a sala às 16h22 em uma segunda-feira, ele irá botar o seu cartão RFID no leitor e consequentemente a iluminação e refrigeração serão ativadas e a fechadura eletrônica destrancada. Então, as 18h o professor encerra a aula, ele bota o seu cartão RFID no leitor, e fazendo isto será identificado que ele está saindo da sala e automaticamente a iluminação, refrigeração serão desativadas e a fechadura eletrônica trancada.

E caso esse professor que ministra aulas na sala A4 às 16h20 às 18h horas nas segundas e quartas-feiras, tente abrir essa sala de aula, botando o seu cartão RFID no leitor da sala, em um horário ou em um dia da semana diferente dos que foram citados anteriormente, a fechadura eletrônica continuará trancada e a refrigeração e iluminação não serão ativadas.

2.1 LAYOUT DO PROJETO

Para montar o *layout* do projeto, ilustrado na figura 12, foi utilizado um software online para construção de diagramas.

Figura 12 – Layout do projeto



Fonte: Própria.

Através do *layout* busca-se ilustrar o passo a passo do funcionamento do projeto, desde o momento em que a leitura do cartão RFID é feita até o momento em que a fechadura eletrônica, a iluminação e a refrigeração são acionadas. Além disso, também é possível observar a comunicação entre o ESP32 e o banco de dados, que ocorre de forma *wireless*, seguindo as normas do protocolo HTTP. E a comunicação entre o banco de dados e o *site*, que também segue as normas do protocolo HTTP. Onde ambas foram citadas anteriormente. Vale salientar, que esse *layout* ilustra o sistema desenvolvido para uma sala de aula, na prática cada sala precisará de um sistema desse.

2.2 DESENVOLVIMENTO DOS CÓDIGOS-FONTES

Os desenvolvimentos dos códigos-fontes possibilitaram fazer o controle seguindo toda a lógica do projeto, explicada anteriormente, de ativação e desativação da iluminação, refrigeração e fechadura eletrônica. Além de também terem sido utilizados para criar a

comunicação wireless entre o ESP32 e o banco de dados e criar uma interface onde são cadastradas por um usuário as informações que ficam armazenadas no banco de dados.

2.2.1 Desenvolvimento da lógica do código-fonte do ESP32

O código-fonte do ESP32 foi desenvolvido na IDE do Arduino, devido a compatibilidade entre as duas plataformas que permite que o código fonte do ESP seja desenvolvido na interface padrão de desenvolvimento de algoritmos do Arduino.

A primeira parte do desenvolvimento do código fonte do ESP 32, diz respeito a conectá-lo a uma rede *Wi-Fi* local para poder enviar e receber informações ao banco de dados. Posteriormente serão feitas a leitura do cartão RFID e o processamento dos dados lidos pelo leitor RFID. Ou seja, nesta etapa será lida a informação contida no cartão RFID através do leitor RFID que está conectado ao ESP e então essas informações serão processadas pelo ESP.

Após isso, o ESP estará com a informação do código RFID pronta para ser enviada, entretanto além desse dado, ele também tem que enviar ao banco de dados o número da sala, o dia da semana e o exato horário (hora e minuto) em que a operação está ocorrendo. Como cada sala de aula possuirá um ESP instalado, cada um deles terá a informação do número da sala a qual ele está instalado, logo a informação referente ao número da sala de aula já está presente no código fonte precisando apenas ser enviada. Referente a informação de horário e dia da semana foi utilizado um protocolo que sincroniza o relógio interno que o ESP possui com o horário da rede *Wi-Fi* a qual o ESP está conectado, assim o ESP obtém a informação de data e horário.

Assim, o ESP possuirá todas as informações necessárias a serem enviadas ao banco de dados. Então chegou o momento de enviar estes dados. Como foi citado anteriormente ESP está conectado a uma rede *Wi-Fi* local, logo as informações serão enviadas ao banco de dados através dessa rede, onde o protocolo utilizado para essa comunicação foi o HTTP. Depois do envio das informações o ESP receberá uma resposta do banco de dados que pode ser positiva, negativa ou até mesmo uma mensagem de erro dizendo que a comunicação falou.

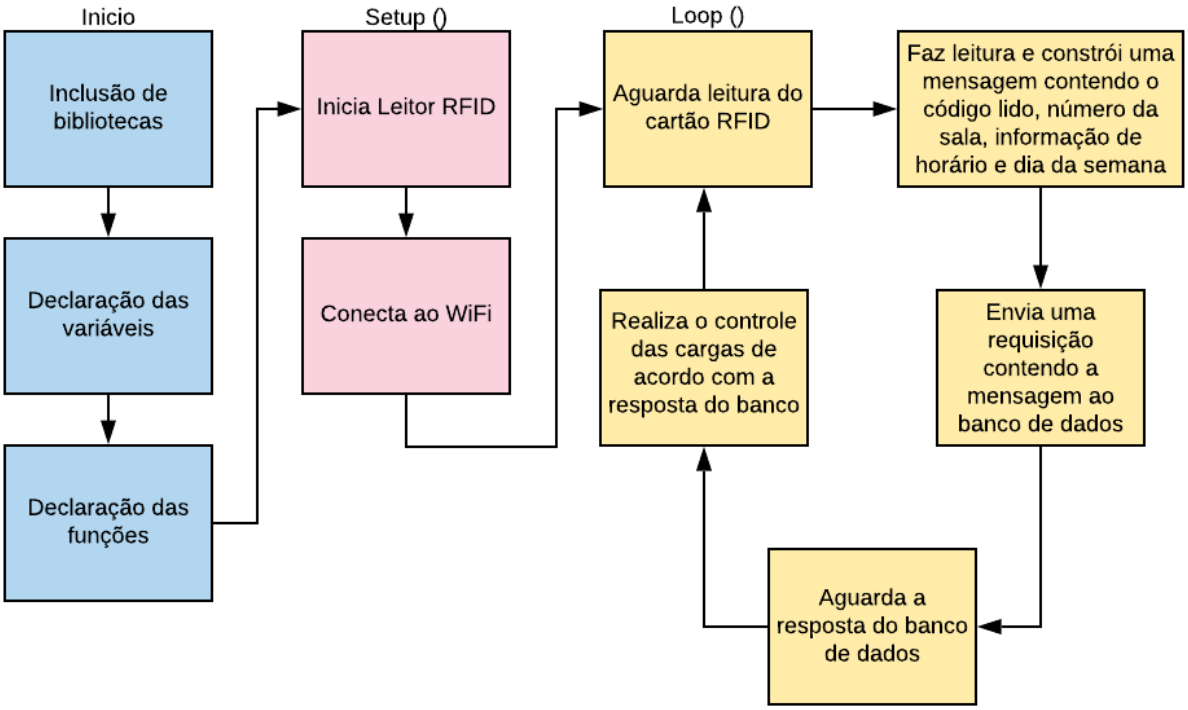
De posse da resposta recebida pelo banco de dados, o ESP processa as informações contidas na resposta e caso ela seja positiva, ele enviará um nível lógico baixo para o módulo relé de dois canais, que ocasionará o acionamento do mesmo, fazendo com que a fechadura eletrônica seja destrancada e a iluminação ativada, e enviará ao o LED infravermelho uma sequência de níveis lógicos que fará com que a refrigeração seja ativada.

Caso a resposta enviada pelo banco de dados ao ESP seja negativa, O ESP não enviará nenhum nível lógico ou sequência de níveis lógicos para o módulo relé de dois canais e nem para o LED infravermelho, o que fará com que a iluminação e refrigeração continuem desligadas e com que a fechadura eletrônica permaneça trancada. E caso a resposta enviada pelo banco de dados seja uma mensagem de erro, toda a operação terá que ser refeita, ou seja, o cartão RFID terá que ser posto pelo usuário novamente no leitor RFID do sistema do projeto.

Na hora de sair da sala de aula o usuário deverá pôr o seu cartão RFID no leitor, o mesmo procedimento que foi realizado na entrada. E quando esse procedimento for feito o ESP entenderá que o usuário está saindo da sala e enviará um nível lógico alto para o módulo relé de dois canais, assim desativando o mesmo, e fazendo com que a iluminação seja desligada e com que a fechadura eletrônica passe a ficar trancada, e enviará uma sequência de níveis lógicos para o LED infravermelho fazendo com que o ar-condicionado seja desligado. Vale salientar que nesse processo de saída, o ESP não envia e nem recebe informações do banco de dados. Logo a qualquer momento o usuário com seu cartão RFID consegue realizar esse procedimento citado anteriormente, independente do que esteja cadastrado no banco de dados.

A figura 13 ilustra o fluxograma da lógica do código-fonte do ESP32.

Figura 13 – Fluxograma da lógica do código-fonte do ESP32



Fonte: Própria.

2.2.2 Desenvolvimento da lógica do código-fonte do banco de dados

O desenvolvimento da lógica do código-fonte do banco de dados consiste basicamente no armazenamento de informações necessária para o projeto, o recebimento de uma requisição enviada pelo ESP, o tratamento dessa mensagem recebida, onde as informações nela contidas serão comparadas com informações já cadastradas no banco de dados e através disso será enviada uma resposta do banco de dados para o ESP que pode ser positiva, negativa ou um erro de comunicação. Além disso, existe a comunicação entre o *site* do projeto e o banco de dados, onde o site envia uma requisição a o banco de dados contendo informações, e essas informações são recebidas, processadas e cadastradas no banco de dados, e ele enviará uma resposta ao site dizendo se foi ou não possível cadastrar a informação.

A plataforma na qual foi desenvolvida o banco de dados é a Back4App que fornece as ferramentas para desenvolver um *back-end* completo incluindo um banco de dados com API (*Application Programming Interface*), que foi exatamente o que foi desenvolvido no projeto, um banco de dados, onde foram armazenadas as informações como o nome do professor, código RFID, número da sala, dia da semana e horário início e fim das aulas.

O banco de dados possui dois códigos fontes principais, o primeiro é a API do banco de dados, onde são feitas basicamente duas lógicas. A primeira é a lógica que possibilita o recebimento e a comparação das informações enviadas pelo ESP com as já contidas no banco de dados, gerando e enviando uma resposta da comparação para o ESP. E a segunda é a lógica que possibilita o recebimento de informações do *site*, onde essas informações são recebidas, processadas e armazenadas no banco de dados, e ele envia uma resposta ao *site*, contendo uma mensagem que diz se foi ou não possível cadastrar as informações enviadas pelo *site*.

O segundo código fonte do banco de dados é o arquivo de configuração base do projeto, ou seja, nele foram colocadas todas as bibliotecas necessárias para a API do banco de dados rodar.

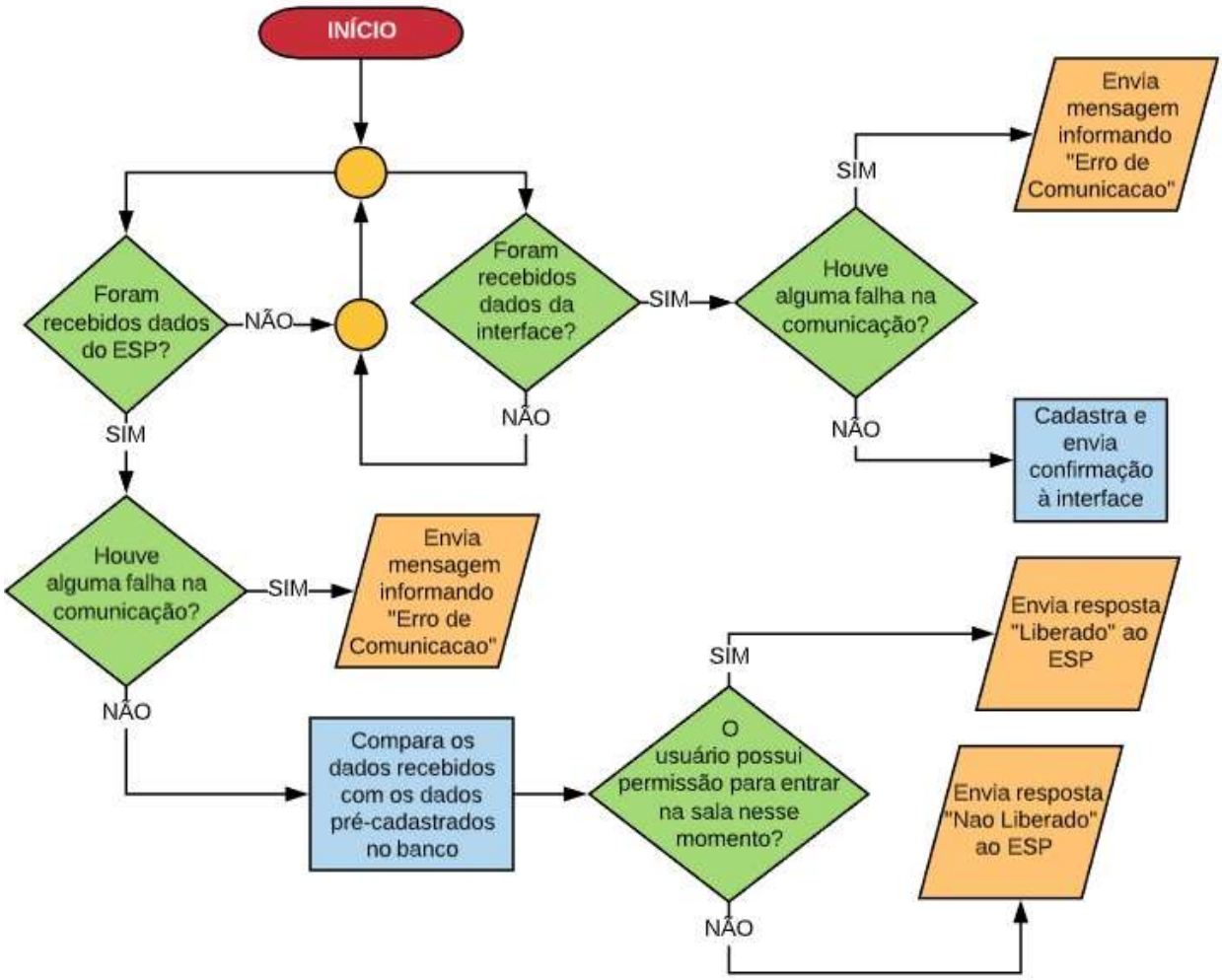
O ESP32 estabelece uma comunicação com o banco de dados na qual se foi utilizado o protocolo HTTP, onde o ESP32 é o cliente e o banco de dados o servidor. Nessa interação entre as duas plataformas o ESP envia uma requisição toda vez que um cartão RFID é lido pelo leitor RFID, que está integrado ao sistema. Nessa requisição, é enviado também ao servidor o código do cartão RFID que está sendo lido, o número da sala e o dia da semana e o horário (hora e minuto) em que o mesmo foi lido. Então, o servidor recebe essas informações e faz uma comparação com dados já cadastrados, confirmando se aquele cartão RFID que foi lido anteriormente possui cadastro para utilizar a sala naquele momento. Se tiver, o servidor enviará uma resposta positiva ao ESP dizendo liberado, e senão tiver o servidor irá mandar uma resposta negativa dizendo não liberado.

Além disso, pode ocorrer um erro na comunicação entre o ESP e o banco de dados, e caso isso aconteça, o banco de dados enviará uma resposta ao ESP dizendo que houve um erro na comunicação. E para restabelecer a comunicação será necessário refazer toda a operação, ou seja, o cartão RFID do usuário terá que ser lido novamente.

Como foi citado anteriormente, no site desenvolvido para este projeto são cadastradas as seguintes informações: nome do professor, código RFID, sala de aula, horários e dias da semana em que o professor ministra aula em determinada sala. O site faz uma requisição, onde essas informações são enviadas para o banco de dados, o banco de dados por sua vez recebe, processa e armazena essas informações, e envia uma resposta ao site com uma mensagem dizendo se foi ou não possível cadastrar essas informações.

A figura 14 ilustra o fluxograma da lógica do código-fonte do banco de dados desenvolvido para este projeto.

Figura 14 – Fluxograma da lógica do código-fonte do banco de dados



Fonte: Própria.

2.2.3 Desenvolvimento da lógica do código-fonte do site

O *site* também possui dois códigos-fonte principais, onde o primeiro guarda as referências das bibliotecas que a aplicação precisa para rodar e as funções necessárias para fazer a compilação do site. E o segundo é onde está a configuração da comunicação entre o site e o banco de dados, ou seja, nesse código-fonte foi feita toda a lógica do envio das informações cadastradas no site para o banco de dados.

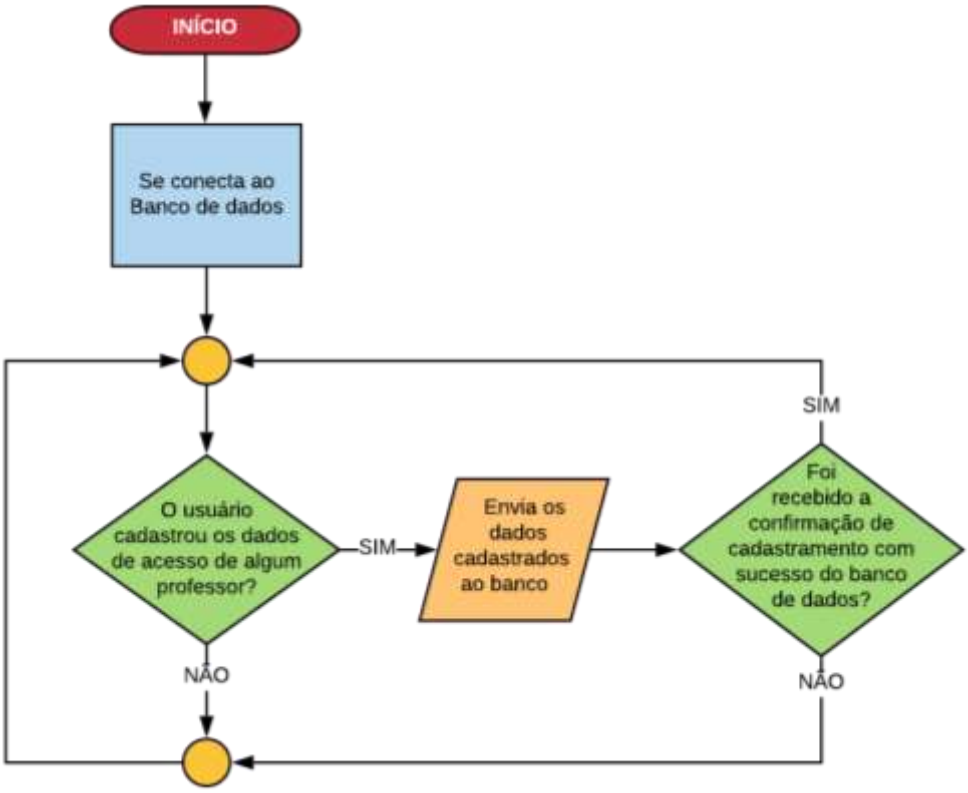
Para o desenvolvimento da lógica do código-fonte do site foi pensado inicialmente em como fazer um cadastro de informações necessárias para o projeto através de uma interface gráfica, que são essas: o nome do professor, código RFID, número da sala, dia da semana e o horário de início e fim das aulas. Para esse cadastro ocorrer foi-se feito todo um trabalho de *front-end*, de tal forma que o cliente não precise cadastrar informações diretamente no banco

de dados, ao invés disso criou-se um site que tornou mais interativo e intuitivo o cadastro de todas informações necessárias para o sistema. Portanto, toda vez em que formem cadastradas informações no *site* desenvolvido para este projeto, as mesmas serão automaticamente e instantaneamente cadastradas também no banco de dados.

No desenvolvimento da lógica do código-fonte também se preocupou com a aplicabilidade do projeto, tendo em vista que poucas pessoas tem a habilidade de manusear um banco de dados mesmo no que diz respeito a cadastro de informações diretamente no banco de dados. Entretanto com a enorme acessibilidade e desenvolvimento da internet ultimamente, a maior parte da população possui a habilidade de manusear *sites*, ainda mais quando estes são de fácil manuseio e bem interativos, como é o caso do desenvolvido para esse projeto. A preocupação em fazer um *site* interativo para o usuário foi tamanha que na hora de cadastrar informações o usuário não precisa digitar nada, todas elas já estão pré-cadastradas no próprio código-fonte do *site*, e basta o usuário escolher e clicar na informação que deseja cadastrar. Esse procedimento além de tornar o *site* mais intuitivo, também diminui significativamente a possibilidade de informações erradas serem cadastradas pelo usuário e implicar no funcionamento correto do sistema desenvolvido para este projeto.

A figura 15 ilustra o fluxograma da lógica do código-fonte do *site* desenvolvido para este projeto.

Figura 15 – Fluxograma da lógica do código-fonte do site



Fonte: Própria.

3 IMPLEMENTAÇÃO DO PROJETO

Neste tópico será apresentado como foi realizada a implementação do projeto, abordando sobre a construção do circuito protótipo desenvolvido, e sobre como foram desenvolvidos o banco de dados e o *site* utilizados nesse sistema. Onde serão citados sobre os materiais utilizados, montagem e testes dos periféricos juntamente com o microcontrolador ESP32.

3.1 MATERIAIS UTILIZADOS

Para construir o circuito protótipo foi necessário a utilização de alguns materiais, a seguir será mostrado e bordado sobre algumas das características dos componentes que foram utilizados para compor o circuito protótipo deste projeto.

O cartão RFID utilizado nesse projeto, ilustrado na figura 16, faz parte do Kit do leitor RFID MFRC522, ele opera com uma frequência de 13,56 MHz, alcance de 2 a 10 cm e tempo de leitura de 1 a 2 ms.

Figura 16 – Cartão RFID

Fonte: TecEletron (2019).

O leitor RFID utilizado neste projeto é o MFRC522, ilustrado na figura 17, ele é utilizado em comunicação sem contato a uma frequência de 13,56 MHz, possui uma tensão de operação de 3V3, taxa de transferência de 10Mbit/s e alcance de 0 a 3 cm.

Figura 17 – Leitor RFID MFRC522

Fonte: Cruzes (2015).

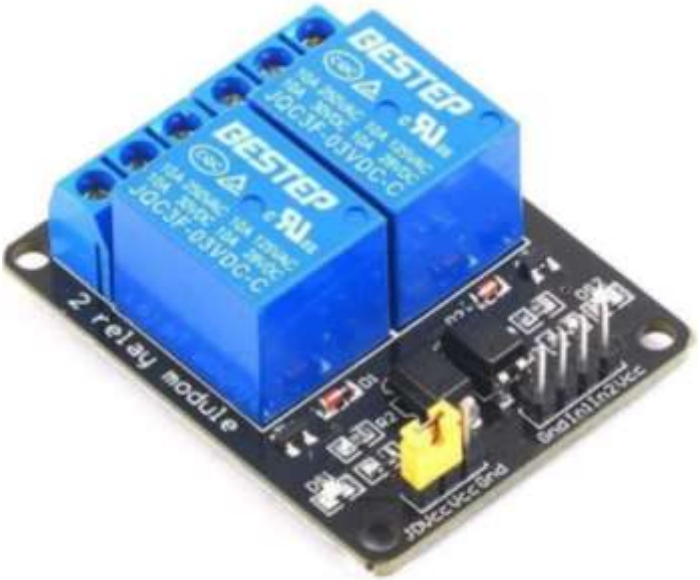
E o sistema embarcado utilizado nesse projeto é o DOIT ESP32 Dev Kit V1, ilustrado na figura 18, sua tensão de alimentação pode ser de 4,5 à 12,0 VDC, tensão de nível lógico é de 3V3, corrente de consumo típica é de 80 mA e máxima de 500 mA, conexão com rede WIFI 802.1 1b/g/n: 2,4 à 2,5 GHz e é compatível com a IDE do Arduino.

Figura 18 – Plataforma DOIT ESP32 Dev Kit V1

Fonte: Própria.

O módulo relé utilizado no projeto, ilustrado na figura 19, possui 2 canais, ou seja, o módulo possui dois relés que podem ser utilizados individualmente, sua tensão de alimentação é de 3V e através desse módulo é possível escolher se as configurações dos contatos do relé serão normalmente abertos ou fechados.

Figura 19 – Módulo relé 3V com dois canais



Fonte: Mirella (2015).

O diodo emissor de luz infravermelha utilizado no projeto foi o LED Infravermelho TIL32 5mm, ilustrado na figura 20, possui 5mm de diâmetro, emissor IR 940 nm, tensão de operação de 1,2 a 1,4 V e comprimento de onda de 940 nm.

Figura 20 – LED Infravermelho TIL32 5mm

Fonte: FilipeFlop (2019)

A fechadura eletrônica utilizada no projeto foi a fechadura eletroímã Intelbras Eletrônica Fe-20150, ilustrada na figura 21, é alimentada com 12 VDC e possui tração de 150 Kgf.

Figura 21 – Fechadura eletroímã Intelbras Eletrônica Fe-20150

Fonte: Intelbras (2017).

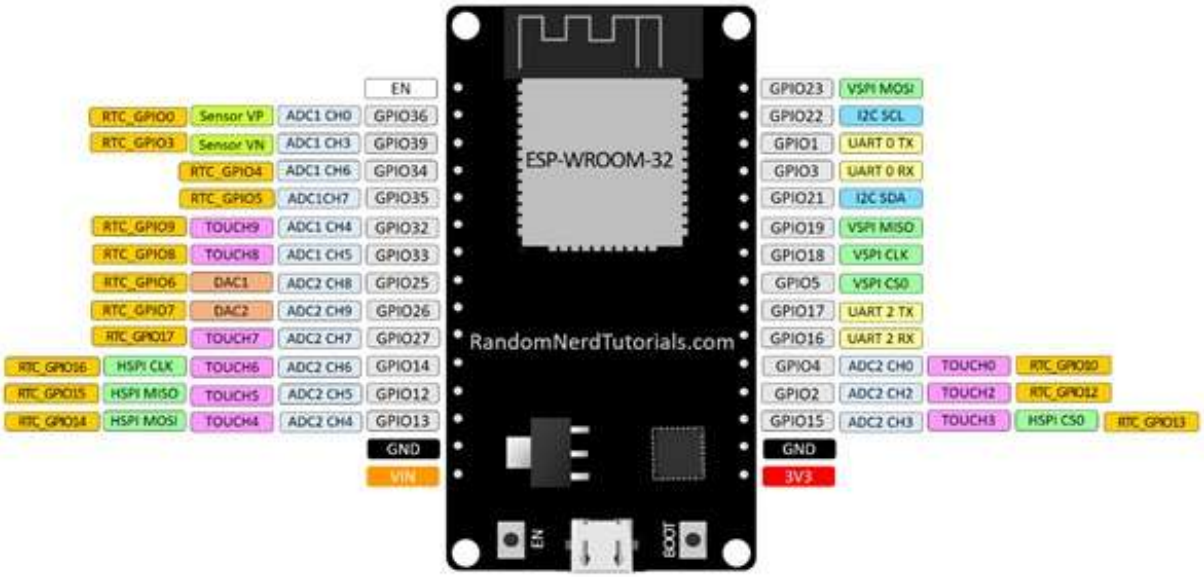
E esses foram os principais componentes utilizados para montar o circuito protótipo do projeto desenvolvido neste trabalho.

3.2 MONTAGEM DOS PERIFÉRICOS

O ESP32 é responsável por fazer o controle da ativação e desativação da refrigeração, iluminação e fechadura eletrônica, logo os periféricos que estarão responsáveis por acionar/desativar esses itens estarão ligados ao ESP32. Além deles o módulo leitor RFID MFRC522 também estará conectado a ele.

Para o funcionamento do projeto é de fundamental importância saber em quais pinos do DOIT ESP32 Dev Kit V1 estarão conectados a esses periféricos, citados anteriormente. Portanto, na figura 22 está ilustrada uma imagem que mostra a pinagem do ESP32.

Figura 22 – Pinagem do ESP32



Fonte: Cruzes (2015).

3.2.1 Montagem do módulo leitor RFID MFRC522

No circuito protótipo deste projeto o módulo leitor RFID MFRC522 é conectado ao ESP32, onde as pinagens de cada módulo estão sendo especificadas nas figuras 22 e 23, respectivamente, e as conexões entre os pinos deles foram ligadas conforme a tabela 1.

Figura 23 – Pinagem do módulo leitor RFID MFRC522

Fonte: Koyanagi (2018).

Tabela 1 – conexão entre o módulo leitor RFID MFRC522 e o ESP32

Módulo Leitor RFID MFRC522	Módulo ESP32
VCC	3V3
RST	D15
GND	GND
IRQ	NÃO CONECTADO
MISO	D19
MOSI	D23
SCK	D18
SDA	D21

Fonte: Própria.

3.2.2 Montagem do módulo relé eletromecânico de 2 canais

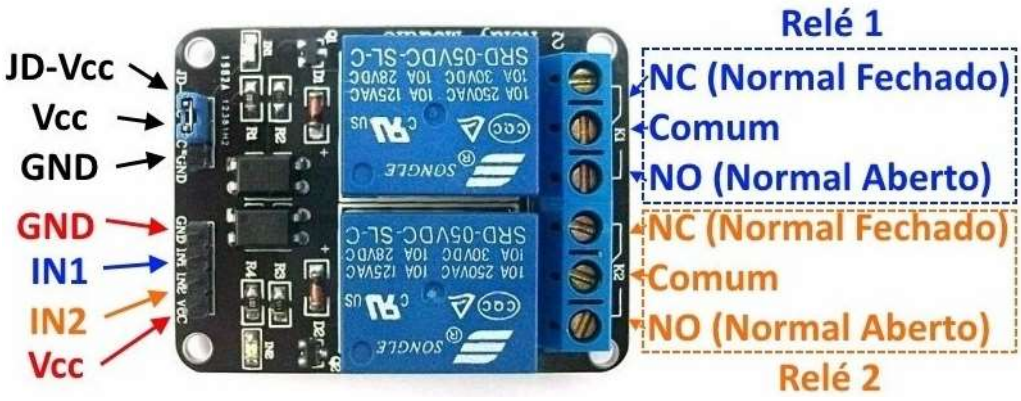
Para o acionamento e desativação da iluminação e para trancar e destrancar a fechadura eletrônica foi conectado ao ESP32 um módulo relé eletromecânico de 2 canais de 3V. A figura 24 ilustra a pinagem do módulo relé e as conexões entre o ESP32 e o módulo relé estão especificadas na tabela 2.

Tabela 2 – conexão entre o módulo relé 2 canais e o ESP32

Módulo RELÉ 2 CANAIS	Módulo ESP32
JD-VCC	NÃO CONECTADO
VCC	NÃO CONECTADO
GND	NÃO CONECTADO
GND	GND
IN1	D13
IN2	D14
VCC	3V3

Fonte: Própria.

Figura 24 – Pinagem do módulo relé de 2 canais



Fonte: Mirella (2015).

Na tabela 3 é mostrado a conexão entre o módulo relé com a lâmpada e a fechadura eletrônica. Onde a fechadura está conectada no relé 1 e a lâmpada no 2. A fechadura é conectada para que os seus contatos fiquem normalmente fechados, portanto normalmente ela fica trancada, e a lâmpada para os seus contatos fiquem normalmente abertos, logo normalmente ela fica desligada. Como foi citado anteriormente esse módulo relé é acionado com nível lógico baixo, ou seja, para destrancar a fechadura é necessário enviar um nível lógico baixo do pino digital D13 do ESP para o pino IN1 do módulo relé. E para ligar a lâmpada é necessário enviar um nível lógico baixo do pino digital D14 do ESP para o pino IN2 do módulo relé.

Na prática deve-se ser instalada uma bateria de 12 V para alimentar a fechadura eletrônica, no caso de falta de energia elétrica. Dessa forma a fechadura eletrônica permanecerá trancada mesmo na falta de energia.

Tabela 3 – conexão entre o módulo relé de 2 canais com a lâmpada e a fechadura eletrônica

Módulo Relé 2 Canais			
Canal 1		Canal 2	
NC	NÃO CONECTADO	NC	Terminal GND do imã
COMUM	LÂMPADA	COMUM	Fonte 12V
NO	NEUTRO	NO	NÃO CONECTADO

Fonte: Própria.

Vale salientar, que o relé na verdade deve ser ligado na saída do disjuntor da iluminação da sala de aula, de tal forma que quando o relé for acionado, todas as lâmpadas do local serão acionadas, da mesma forma que quando relé for desativado, todas as lâmpadas se apagarão. Mas como o circuito montado foi um protótipo utilizou-se apenas uma lâmpada.

3.2.3 Montagem do diodo emissor de luz infravermelha

Para realizar o acionamento e desativação da refrigeração é utilizado o diodo emissor de luz infravermelha. Onde este estará conectado ao ESP32 conforme mostrado na tabela 4.

Tabela 4 – conexão entre o diodo emissor de luz infravermelha e o ESP32

Diodo emissor de luz infravermelha	Módulo ESP32
SINAL	D4
GND	GND

Fonte: Própria.

3.3 DESENVOLVIMENTO DO BANCO DE DADOS

O primeiro passo para o desenvolvimento do banco de dados foi criar uma conta no *site* Back4App. Este é um banco de dados que fornece toda uma configuração de API pronta, oferecendo uma rotina de backup, controle de dados, controle de usuário, uma política de segurança, entre outros.

Após criar a conta no banco de dados, criou-se um projeto associado à uma instância exclusiva para a aplicação desenvolvida. Dentro dessa instância foi criada uma tabela, onde ficam armazenadas as informações referentes ao nome do professor, código RFID, número da sala de aula, horário de início e fim de cada aula, além dos dias da semana em que estas ocorrem.

O Back4App também fornece a chave de acesso ao banco, onde essa libera o acesso ao banco para leitura ou escrita. Seguindo o mesmo princípio, ele também fornece a chave da linguagem, tendo em vista que para programar no banco de dados criado por esse sistema, podem ser utilizadas diversas linguagens, onde cada uma dessas possuirá uma chave de linguagem própria que permitirá o acesso ao banco de dados. No caso deste projeto utilizou-se a javaScriptKEY, pois a programação utilizada no banco de dados desse projeto foi JavaScript.

O Back4App também fornece o link do banco de dados utilizado, que é o endereço de onde este está localizado na *Web*. Essas duas chaves e o *link* são necessários para que a API possa acessar o banco de dados.

Para evitar que o banco de dados seja exposto, criou-se uma API para encapsular esses dados de acesso ao banco. Em especial, a API foi desenvolvida utilizando Node.js+express que é o *framework* da API.

A API tem a configuração do banco de dados baseado nas chaves e no link que o Back4App forneceu. A sintaxe da API é baseada na biblioteca parse que é desenvolvida justamente para se trabalhar com a plataforma do Back4App.

No desenvolvimento da API foram criadas duas rotas, onde uma diz respeito ao cadastro de informações no banco de dados, essa rota em especifico é utilizada pelo *front-end* do *site*. E a segunda rota é baseada numa consulta no banco de dados que verifica o acesso do professor, onde essa rota é utilizada especificamente pelo ESP32.

3.4 DESENVOLVIMENTO DO SITE

A IDE utilizada para o desenvolvimento do *site* foi o Visual Studio Code, onde foi codificado o *front-end*.

O segundo passo foi instalar as ferramentas de linguagem, onde utilizou-se o Node.js. Estas ferramentas foram responsáveis por auxiliar e facilitar o projeto desenvolvido. Para o Node funcionar adequadamente foi necessário instalar outra ferramenta, o Git, nele foi necessário realizar as configurações de usuário, que consiste em cadastrar o e-mail e o nome de usuário.

Depois de finalizada essas instalações, foi instalado o Vue.js que é o *framework* do JavaScript, que foi a linguagem utilizada para desenvolver a interface.

Em seguida foi feito o desenho do *layout* da interface e a criação da mesma. Depois foi realizada a conexão desta com a API do banco de dados.

Por fim, foi enviado todos os arquivos que foram criados para o desenvolvimento do *site* para uma pasta, onde posteriormente foi feito o reconhecimento desta pelo Heroku e este por sua vez disponibilizou o *site online*.

4 RESULTADOS

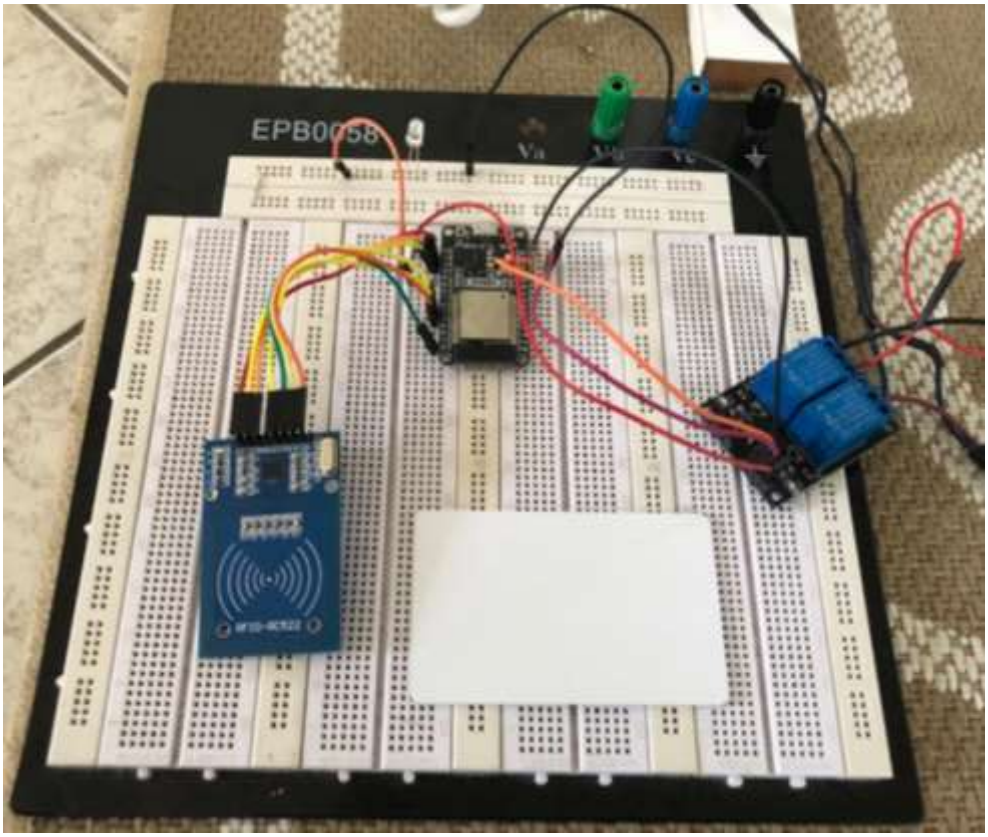
O primeiro resultado obtido no projeto foi a montagem do circuito protótipo, ilustrado na figura 25, onde monotou-se todo um cenário buscando deixar o projeto o mais próximo possível do que seria este sistema na prática.

Figura 25 – Circuito protótipo desenvolvido para este projeto

Fonte: Própria.

Na figura 26, pode ser observado mais detalhadamente os periféricos conectados ao ESP32.

Figura 26 – Circuito do ESP32 conectados aos seus periféricos

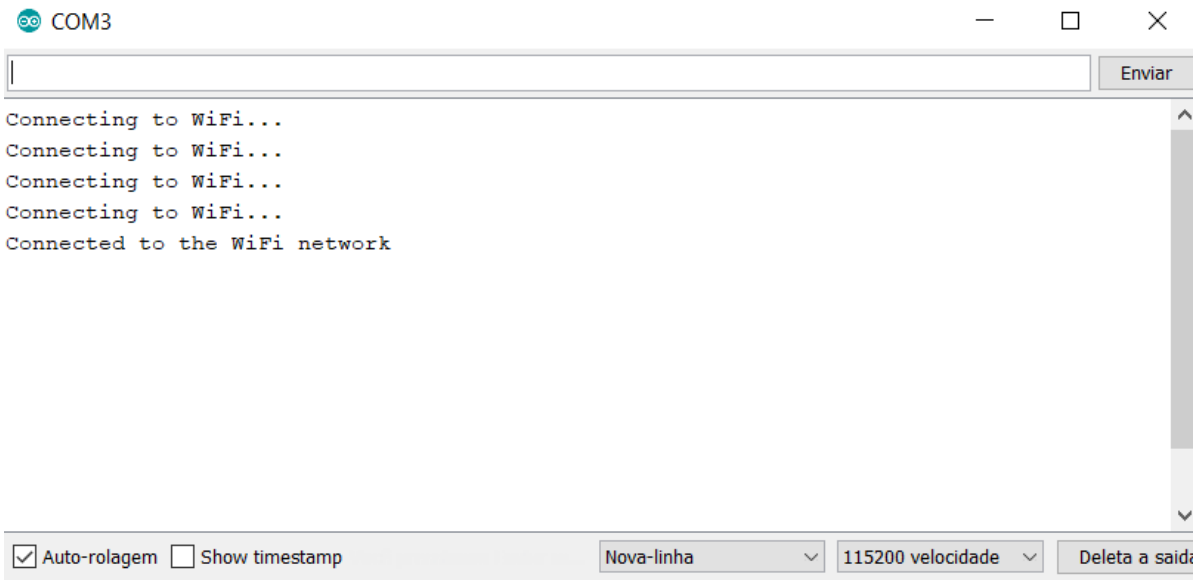


Fonte: Própria.

O segundo resultado alcançado no projeto foi conseguir conectar o ESP32 a uma rede *Wi-Fi* local, o que foi de suma importância para o funcionamento do projeto, pois através deste foi possível fazer com que o ESP enviasse e recebesse informações.

A seguir será ilustrado na figura 27 o monitor serial da IDE do Arduino quando o ESP32 se conecta ao Wi-Fi.

Figura 27 – Monitor serial do arduino quando o ESP32 se conecta ao Wi-Fi

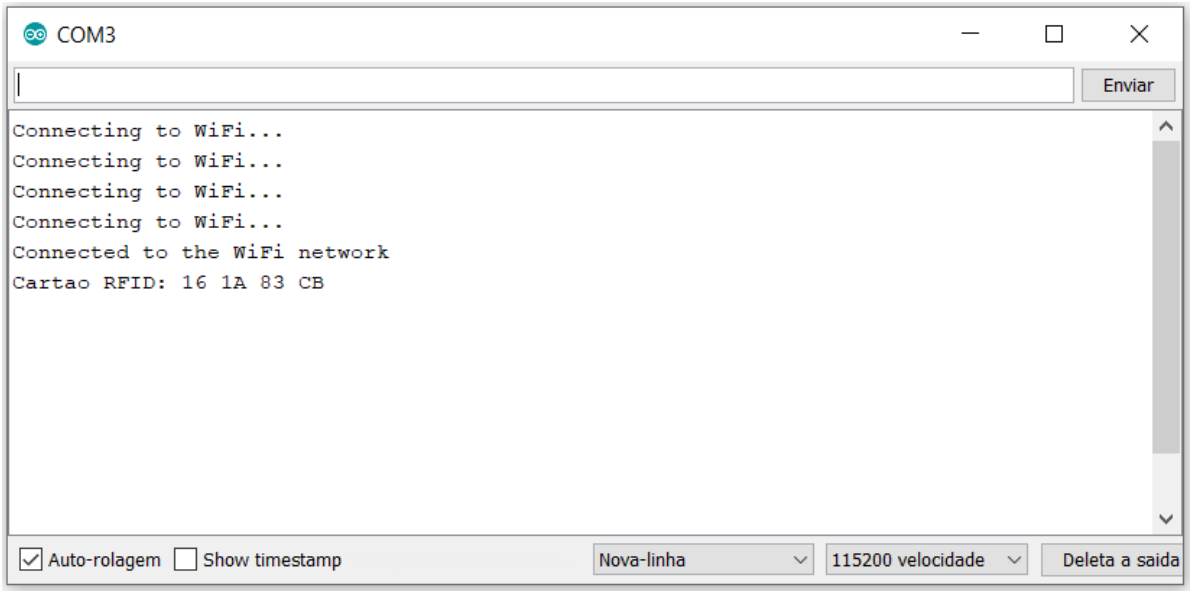


Fonte: Própria.

O terceiro resultado obtido foi fazer com que o ESP32 reconhecesse o código RFID do cartão do usuário que era lido pelo leitor RFID MFRC522. Assim gerando a possibilidade que o sistema reconhecesse quem era o usuário, pois cada usuário possuía um código RFID único, apesar do ESP32 não fazer este reconhecimento no projeto.

Na figura 28 está sendo ilustrado o código RFID que é recebido e processado pelo ESP, quando o cartão RFID for lido pelo leitor.

Figura 28 – Monitor serial quando o ESP32 recebe e processa o código do cartão RFID



Fonte: Própria.

O quarto resultado atingido foi fazer com que o relógio interno do ESP sincronize com a hora da rede *Wi-Fi* local a qual o ESP32 está conectado. Dessa forma o ESP32 consegue saber a data e a hora em que a leitura do cartão RFID está ocorrendo.

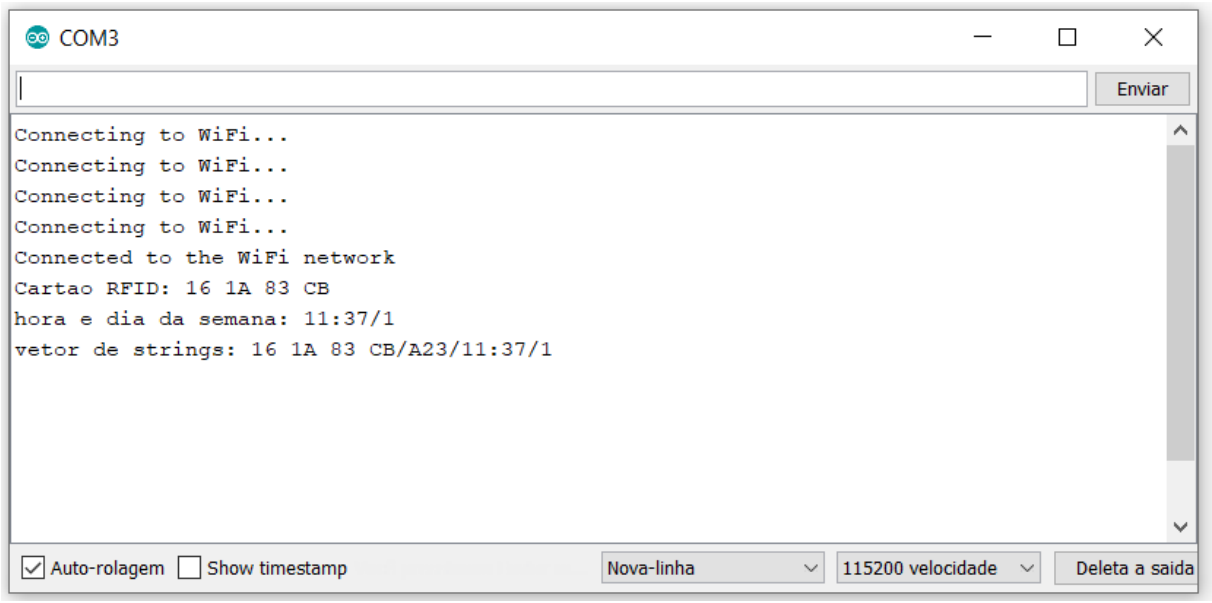
Na figura 29 está sendo ilustrado data e a hora no monitor serial em que uma leitura do cartão RFID está acontecendo. Os dias da semana estão sendo representados de 0 a 6, onde o 0 é o domingo, o 1 a segunda-feira e assim respetivamente.

Figura 29 – Monitor serial exibindo o horário e dia da semana em que uma leitura do cartão RFID ocorreu

Fonte: Própria.

O quinto resultado foi a criação de um vetor de *strings*, contendo o código RFID, número da sala de aula, horário (hora e minuto) e o dia da semana, respectivamente. Na figura 30 está sendo ilustrado esse vetor exibido no monitor serial.

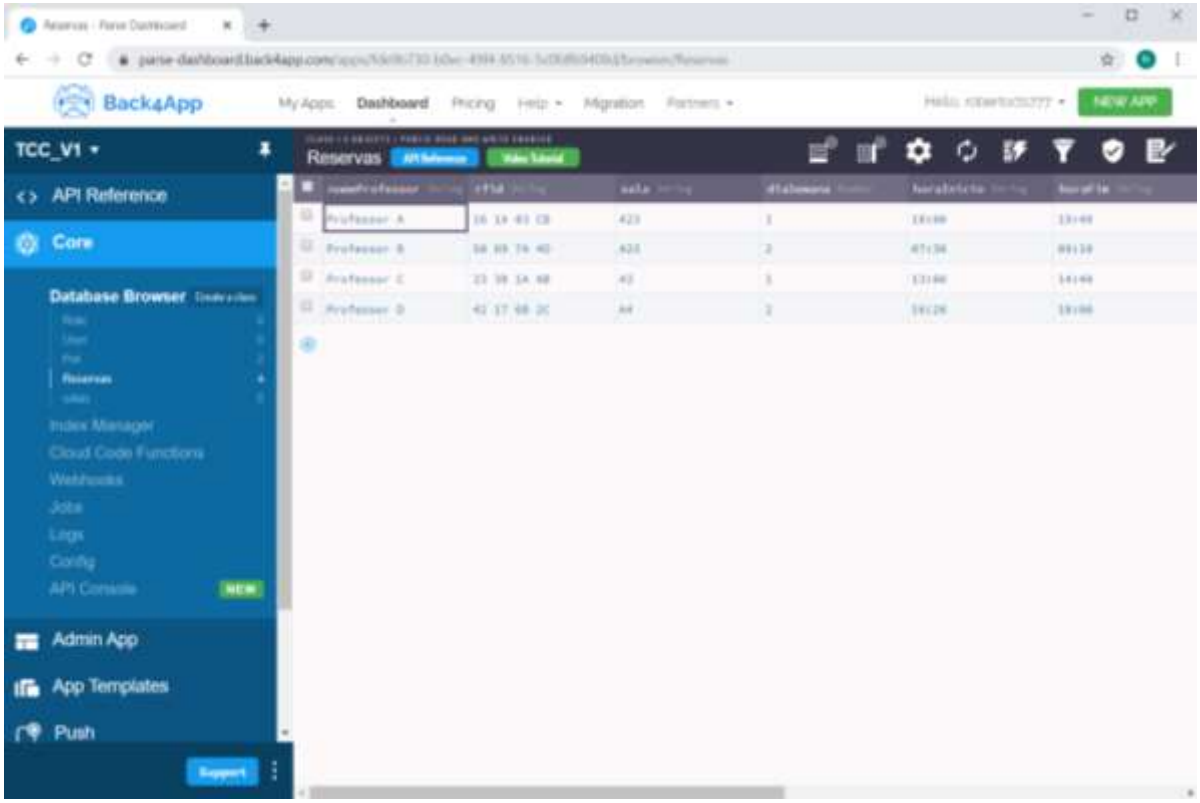
Figura 30 – Vetor de strings exibido no monitor serial do arduino



Fonte: Própria.

O sexto resultado alcançado foi a criação de um banco de dados e da sua API na plataforma Back4app. Que permitiu o armazenamento das seguintes informações: nome do usuário, código RFID, número da sala, intervalos de horários e dias da semana que o mesmo possuía permissão de utilizar esse local. No banco de dados o código RFID único já é vinculado ao nome do usuário, de posse disso é possível saber qual é o nome do usuário cujo o qual o cartão RFID foi lido. Na figura31 é mostrado o banco de dados desenvolvido para este projeto.

Figura 31 – Banco de dados desenvolvido na plataforma Back4App referente ao projeto desenvolvido para este trabalho



Fonte: Própria.

O sétimo resultado alcançado foi fazer com que o ESP conseguisse enviar o vetor de *strings* abordado anteriormente, com as seguintes informações: código RFID lido, o número da sala, horário (hora e minuto) e o dia da semana em que havia ocorrido a leitura do cartão RFID para o banco dados. Possibilitando que essas informações fossem processadas pelo banco de dados e gera-se uma resposta do mesmo. Lembrando que o protocolo utilizado para reger essa comunicação entre o ESP32 e o banco de dados foi o HTTP, onde o ESP32 é o cliente e o banco de dados o servidor.

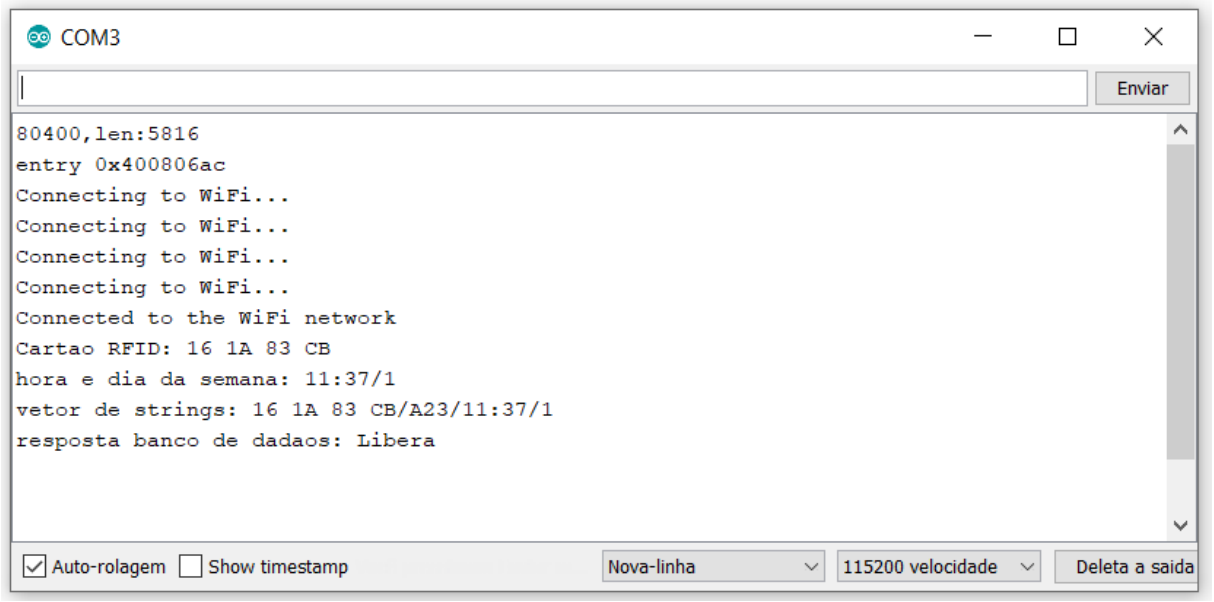
O oitavo resultado obtido foi fazer o banco de dados receber a requisição enviada pelo ESP32, processasse a informação nela contida e gerasse uma resposta positiva, negativa ou até mesmo um de erro de comunicação, tendo em vista a comparação dos dados enviados com os já cadastrados no banco de dados, onde essa resposta será enviada para o ESP32.

O nono resultado foi fazer o ESP32 receber a resposta vinda do banco de dados, processar ela, e caso a informação contida na resposta fosse positiva aciona-se a refrigeração e a iluminação e fechadura eletrônica é destrancada, assim liberando o acesso. Caso fosse negativa manteasse a iluminação e a refrigeração desativadas e a fechadura eletrônica trancada.

E caso fosse um erro de comunicação o banco de dados envia na resposta uma informação que houve erro na comunicação e pede para que a operação ser feita novamente.

As figuras 32 e 33 estão ilustrando a resposta positiva e negativa, respetivamente, que o ESP32 recebe do banco dedados, que é escrita no monitor serial toda vez que o ESP recebe uma resposta do banco de dados.

Figura 32 – Monitor serial quando o ESP32 recebe uma resposta positiva do banco de dados



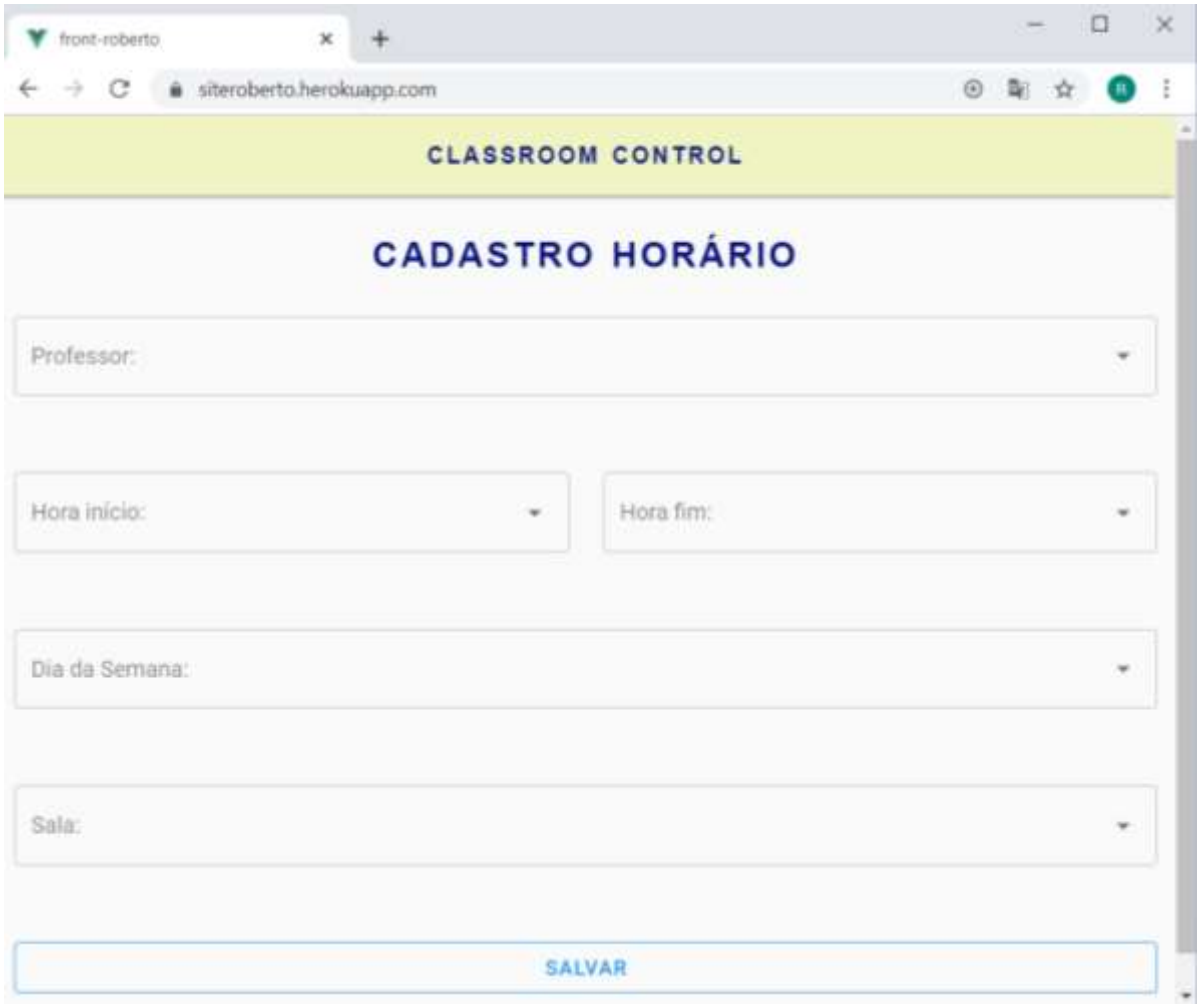
Fonte: Própria.

Figura 33 – Monitor serial quando o ESP32 recebe uma resposta negativa do banco de dados

Fonte: Própria.

E o décimo resultado foi a criação do site, uma interface gráfica online que possibilitou que toda informação que fosse cadastrada no banco de dados fosse cadastrada através desse site de forma bem intuitiva. Ou seja, toda informação que for cadastrada nesse site será automaticamente salva no banco de dados desse projeto. O site pode ser acessado pelo seguinte endereço eletrônico: <*https://siteroberto.herokuapp.com/*>. A figura 34 mostra a interface gráfica do site desenvolvido para este projeto.

Figura 34 – Site desenvolvido para o projeto



Fonte: Própria.

E esses foram os principais resultados alcançados com o desenvolvimento deste trabalho. Não foi possível aplicar o projeto nas salas de aula na prática, logo não conseguiu-se resultados no que diz respeito a eficiência energética, que no caso seria a diminuição significativa de desperdício de energia elétrica no que diz respeito ao uso desnecessário da iluminação e refrigeração quando ninguém está utilizando a sala de aula, o que impactaria em

uma diminuição significativa na conta de energia elétrica. E também não foi possível observar o resultado no que diz respeito a questão de logística, onde haveria a diminuição de transtornos referentes a perdas de chaves de salas de aula.

Outra questão é que através desse projeto só foi possível fazer o controle da refrigeração dos ar-condicionados da marca Midea. Tendo em vista que cada marca possui o seu código próprio para ativação, logo para cada marca deveria ser feito um comando diferente e até mesmo, em alguns casos, serem importadas bibliotecas diferentes para serem acionados os refrigeradores de outras marcas, o que se tornou-se inviável de ser feito, tanto na questão de processamento pelo número de bibliotecas que deveriam ser importadas quanto pela questão de logística de ter que se procurar o código de cada marca e realizar algoritmos diferentes para cada uma delas.

Por fim, em relação aos resultados obtidos, de uma forma geral, alcançou-se os resultados esperados e propostos pelo tema inicial escolhido.

CONCLUSÃO

O sistema de controle da iluminação, refrigeração e da fechadura eletrônica utilizando Internet das Coisas foi concluído com êxito, possuindo um banco de dados que foi desenvolvido para armazenar informações referentes ao usuário, sendo estes dados cadastrados através de um site desenvolvido para este projeto, sendo este de fácil manuseio e interativo para que o usuário consiga dominá-lo facilmente e evitar erros de cadastro que podem comprometer o funcionamento do sistema.

A escolha do módulo ESP32 foi essencial para o projeto, tendo em vista que ele conseguiu atender todas as necessidades no que diz respeito a processamento de dados recebidos, controle de acionamento e desativação de objetos, conectar-se a uma rede *Wi-Fi* local, assim estabelecendo uma comunicação entre ele e o banco de dados, que permitiu que o mesmo enviasse e recebesse mensagens do banco de dados e processasse esses dados recebidos. Além disso, o relógio interno que o ESP32 possui foi de fundamental importância para saber o momento da leitura do cartão RFID do usuário. Todos esses processos foram fundamentais para o funcionamento do projeto apresentado nessa pesquisa.

A escolha da utilização de um banco de dados online mostrou-se ser durante a pesquisa e o desenvolvimento do projeto uma ótima escolha, tendo em vista que ela possibilitou usar o um servidor *online*, ao invés de ter que haver um servidor físico dedicado ao projeto. Assim, tornando o projeto mais acessível financeiramente e aumentando a sua aplicabilidade. Além disso, o banco de dados desenvolvido na plataforma Back4App teve uma excelente funcionalidade no que diz respeito a informações recebidas do ESP32, ao processamento dessas informações, comparando-as com informações já cadastradas no banco de dados e enviando uma resposta ao ESP32 com o resultado dessas comparações ou com uma mensagem de erro de comunicação.

E o desenvolvimento de uma interface gráfica *online*, o *site*, tornou o projeto mais interativo e também aumentou a sua aplicabilidade, onde nessa interface são cadastrados todos os dados do usuário necessário para o funcionamento do projeto. Caso não houvesse essa interface as informações teriam que ser cadastradas diretamente no banco de dados, o que não seria intuitivo. Logo, quando essas informações são cadastradas no *site*, elas automaticamente são cadastradas no banco de dados.

Logo, foi-se feito todo um trabalho de *front-end* e *back-end* no projeto para que o mesmo opere de forma complexa, porém intuitiva para o usuário que for utilizá-lo, assim fazendo com que o sistema tenha uma boa aplicabilidade. Aliado a isso, também foi utilizado o conceito de

internet das coisas no projeto, onde objetos foram conectados à internet e acionados e desativados remotamente, de acordo com a interação da comunicação entre o ESP32 e um banco de dados, onde foi utilizado protocolo HTTP para reger as regras dessa comunicação. Durante o desenvolvimento do sistema o ESP32 se mostrou um ótimo módulo para se trabalhar com projetos voltados ao conceito de IoT, devido a facilidade de conectá-lo a uma rede *Wi-Fi* local, a facilidade de fazer o controle remoto da ativação e da desativação de objetos através da utilização desse módulo e a facilidade de enviar e receber informações e processar esses dados em uma comunicação *wireless*.

Para trabalhos futuros, a necessidade do desenvolvimento de uma interface mais completa contendo a informação de quais salas de aulas deveriam estar fechadas, mas ainda estão abertas. Assim um funcionário pode fazer o monitoramento de todas as salas de aula da instituição, e dessa forma caso um professor esqueça de botar o seu cartão RFID no leitor na hora da saída, algum funcionário poderá ir até o local e trancar a sala, desativando a iluminação e refrigeração.

Além disso, desenvolver também na interface uma página de agendamento para caso o professor deseje utilizar alguma sala em algum horário que ele não tenha permissão, seja possível fazer o agendamento para utilizar a mesma.

E outras duas ideias para projetos futuros são fazer com que o sistema desenvolvido seja capaz de acionar e desativar qualquer marca de ar-condicionado. E a outra ideia é a seguinte: ao invés de se utilizar um servidor que seja um banco de dados *online*, seja utilizado o próprio servidor da universidade dessa forma caso a internet caia o sistema não ficará inutilizado, e além disso poderá ser utilizado os dados já cadastrados neste servidor local.

Outra sugestão para trabalhos futuros é criar uma demanda de acionamento de aparelhos de ar-condicionado, de tal forma que exista um controle de acionamento máximo desses aparelhos em um determinado momento. Assim, evitando que o acionamento de vários aparelhos em um determinado instante venha a ocasionar uma corrente de pico tão elevada, que possa danificar o sistema elétrico da faculdade.

REFERÊNCIAS

ABESCO. **O que é Eficiência Energética? (EE)**. [*S. l.*], 2017. Disponível em: <http://www.abesco.com.br/pt/o-que-e-eficiencia-energetica-ee/>.

ADAFRUIT. **Super-bright 5mm IR LED** – 940nm. 2018. Disponível em: <http://www.adafruit.com/product/387>.

AGUIAR, Matheus Fontinele de. **Desenvolvimento de um sistema de controle de periféricos via Web para ambiente residencial, utilizando Internet das Coisas (IoT) e o protocolo Message Queuing Telemetry Transport (MQTT)**. 2018. 63 p. Trabalho de conclusão de curso (Bacharelado em Engenharia Elétrica) - Universidade Estadual do Amazonas, Manaus, 2018.

ALMEIDA, H. **Internet das coisas nós, as cidades, os robôs, os carros: Tudo conectado!** Revista da Sociedade Brasileira de Computação, n.29, 2015.

ANDRADE, Erick *et al*. **O que é um servidor web (web server)**. [*S. l.*], 2019. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/Common_questions/o_que_e_um_web_server>.

BALAGUER, Adriano. **A internet das coisas: das origens ao futuro**. [*S. l.*], 2014. Disponível em: <https://canaltech.com.br/internet/A-Internet-das-Coisas-das-origens-ao-futuro/>.

CHASE, Otavio. **Sistemas embarcados**. 1 ed. [*S. l.*], 2007. 7 p.

CRUZES, Mogi. **Módulo Leitor RFID Mifare RC522**. São Paulo, 2015. *E-book*. Disponível em: <http://www.arduinomogi.com.br/pd-32844d-modulo-leitor-rfid-mifare-rc522.html>.

ELETROBRAS. **Procel**. 2018. Disponível em: <http://www.eletrobras.com/pt/Paginas/Procel.aspx>.

FILIPEFLOP. 2019. **LED Emissor Infravermelho IR 5mm**. Disponível em: <https://www.filipeflop.com/produto/led-emissor-infravermelho-ir-5mm/>.

GARCIA, Fernando. **Introdução aos sistemas embarcados e microcontroladores**. [*S. l.*], 2018. Disponível em: <https://www.embarcados.com.br/sistemas-embarcados-e-microcontroladores/>.

INSTITUTO NACIONAL DE EFICIÊNCIA ENERGÉTICA – INEE. **O que é eficiência energética?** 2014. Disponível em: <www.inee.org.br/eficiencia_o_que_eh.asp?Cat=eficiencia>.

INTELBRAS. 2017. **Fechadura eletroímã 150kgf**. Disponível em: < https://www.intelbras.com/pt-br/fechadura-eletroima-150kgf-fe-20150/>.

KOYANAGI, Fernando. **ESP32 com RFID: Controle de Acesso**. 2018. Disponível em: <https://brasilescola.uol.com.br/o-que-e/fisica/o-que-e-infravermelho.htm>.

MARTINS, Roberta. **Internet das coisas na logística: veja como ela impulsiona o setor**. [*S.* l.], 2018. Disponível em: <https://cargox.com.br/blog/internet-das-coisas-na-logistica-veja-como-ela-impulsiona-o-setor>.

MIRELLA, Andressa. **Controlando lâmpadas com módulo relé Arduino**. 2015. Disponível: <https://arduino.wordpress.com/2015/12/12/controlando-lampadas-com-modulo-rele-arduino/>.

MONK, S. **Movimento, Luz e Som com Arduino e Raspberry Pi**. 1. ed. [S.l.]: Novatec, 2016.

OLIVEIRA, S. Internet das Coisas com ESP8266, Arduino E Raspberry Pi. 1 ed. [S.L.]: Novatec, 2017.

PAIS, Júlia; COUTO, Marcos. **RFID Radio - Frequency Identification**. [*S. l.*], 2009. Disponível em: <https://www.gta.ufrj.br/grad/09_1/versao-final/rfid/index.html>.

ROBERTS, Michael. **Arduino básico**. 1 ed. São Paulo: Novatec, 2011. 456 p.

ROSE, K.; ELDRIDGE, S.; CHAPIN, L. **The Internet of Things: An Overview – Understanding the Issues and Challenges of a More Connected World**. [S.I.]: The Internet Society, 2015.

SANTOS JÚNIOR, A. S.; LIMA, A. M. C.; SALVATERRA, G. F.; MORAES, R. M. **Eficiência energética residencial**. 67f. Trabalho de Conclusão de Curso – Engenharia Elétrica. Belo Horizante: Faculdade Pitágoras, 2015.

SCHWARTZ, M. **ESP8266 Internet of Things of Cookbook**. 1 ed. [S.l.]: Packt Publishing, 2017.

TECELETRON. 2019. **Cartão Rfid Programável Bi-Frequência TK4100 125Khz + S50 13.56Mhz**. Disponível em: <http://www.teceletron.com.br/loja/toys/rfid/cartao-rfid-programavel-bi-frequencia-tk4100-125khz-s50-13-56mhz/>.

TECTRONICS. 2019. **Módulo Relé 1 canal 5v para Arduíno ARM AVR DSP PIC**. Disponível em: <https://www.tecnotronics.com.br/modulo-rele-arduino.html/>

VIEIRA, Fernando. **Entendendo um pouco mais sobre o protocolo HTTP**. [S. l.], 2007. Disponível em: < https://nandovieira.com.br/entendendo-um-pouco-mais-sobre-o-protocolo-http>.

ZAMBARDA, Pedro. **'Internet das Coisas': entenda o conceito e o que muda com a tecnologia**. [*S. l.*], 2014. Disponível em: <https://www.techtudo.com.br/noticias/noticia/2014/08/internet-das-coisas-entenda-o-conceito-e-o-que-muda-com-tecnologia.html>.

APÊNDICE A – CÓDIGO FONTE DO ESP32

```
#include <BufferedPrint.h>
#include <Constellation.h>
#include <HTTPClient.h>
#include <WiFi.h>
#include <SPI.h>
#include <MFRC522.h>
#include <TimeLib.h>
#include <IRremote.h>
#include <MideaIR.h>

    unsigned long getTime();
    const char* ntpServer = "pool.ntp.org";

    const char* ssid = "Nome da Rede";
    const char* password =  "**********";

    #define SDA_PIN 21
    #define RST_PIN 15
    MFRC522 mfrc522(SDA_PIN, RST_PIN);   // Create MFRC522 instance.
    //String response;
    int cont = 0;

    IRsend irsend(4);
    MideaIR remote_control(&irsend);

    const int ledPin = 13; // número do pino LED

    int porta_rele = 12;

    void setup()
    {
      Serial.begin(115200);
      SPI.begin();      // Inicia  SPI bus
      mfrc522.PCD_Init();   // Inicia MFRC522
      Serial.println("Aproxime o seu cartao do leitor...");
      Serial.println();
      pinMode(2, OUTPUT);
      pinMode(ledPin, OUTPUT);
      pinMode(porta_rele, OUTPUT);
      digitalWrite(porta_rele, HIGH);
      delay(4000);   //Delay needed before calling the WiFi.begin
      WiFi.begin(ssid, password);

```

```
44.       while (WiFi.status() != WL_CONNECTED)  //Check for the c
   onnection
45.       {
46.         delay(1000);
47.         Serial.println("Connecting to WiFi..");
48.       }
49.       configTime(0, 0, ntpServer);
50.
51.       Serial.println("Connected to the WiFi network");
52.
53.     }
54.     void loop()
55.     {
56.       // Procura por cartao RFID
57.       if ( ! mfrc522.PICC_IsNewCardPresent())
58.       {
59.         return;
60.       }
61.       // Seleciona o cartao RFID
62.       if ( ! mfrc522.PICC_ReadCardSerial())
63.       {
64.         return;
65.       }
66.       //Mostra UID na serial
67.       Serial.print("UID da tag :");
68.       String conteudo= "";
69.       byte letra;
70.       for (byte i = 0; i < mfrc522.uid.size; i++)
71.       {
72.          Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : "
    ");
73.          Serial.print(mfrc522.uid.uidByte[i], HEX);
74.          conteudo.concat(String(mfrc522.uid.uidByte[i] < 0x10
   ? " 0" : " "));
75.          conteudo.concat(String(mfrc522.uid.uidByte[i], HEX));

76.       }
77.       Serial.println();
78.       Serial.print("Mensagem : ");
79.       conteudo.toUpperCase();
80.       if (conteudo.substring(1) == "50 69 7A 4D" || conteudo.s
   ubstring(1) == "16 1A 83 CB" ) //UID 1 - Cartao
81.       {
82.         Serial.println("Ola PROFESSOR !");
83.         Serial.println();
84.         digitalWrite(2, HIGH); // ativa rele, abre a trava sol
   enoide
85.         delay(2000);           // espera 3 segundos
```

```
86.           digitalWrite(2, LOW);  // desativa rele, fecha a trava
    solenoide
87.         }
88.           if(WiFi.status()== WL_CONNECTED)   //Check WiFi connec
   tion status
89.           {
90.             HTTPClient http;
91.             //http.begin("https://apiroberto.herokuapp.com/");
   //Specify destination for HTTP request
92.             http.begin("https://apiroberto.herokuapp.com/taLiber
   ado");
93.             http.addHeader("Content-
   Type", "application/json");            //Specify content-
   type header
94.             unsigned long timestamps = getTime();
95.             StaticJsonBuffer<200> jsonBuffer;
96.             String jsonv=
97.             String("{")+
98.               "\"rfid\":\""+String(conteudo.substring(1))+"\","+

99.               "\"sala\":\"A23\","+
100.              "\"timestamps\":\""+String(timestamps)+"\""+
101.            "}";
102.            char json[200];
103.
104.            jsonv.toCharArray(json, sizeof(json));
105.
106.            Serial.println(json);
107.            JsonObject& root = jsonBuffer.parseObject(json);
108.
109.            char json_str[100];
110.            root.prettyPrintTo(json_str, sizeof(json_str));
111.            Serial.println(json);
112.            int httpResponseCode = http.POST(json_str);
113.            if( httpResponseCode>0)
114.            {
115.              String response = http.getString();
             //Get the response to the request
116.              Serial.println(httpResponseCode);   //Print return
    code
117.              Serial.println(response);           //Print reques
   t answer
118.              if (response == "liberado")
119.              {
120.                if(cont % 2 == 0)
121.                {
122.                  digitalWrite(ledPin, HIGH);
123.                  remote_control.turnON();
124.                  Serial.println("on");
```

```
125.                digitalWrite(porta_rele, LOW);
126.              }
127.              else
128.              {
129.                digitalWrite(ledPin, LOW);  // desativa rele,
   fecha a trava solenoide
130.                remote_control.turnOFF();
131.                Serial.println("off");
132.                digitalWrite(porta_rele, HIGH);
133.              }
134.              cont = cont + 1;
135.            }
136.          }
137.          else
138.          {
139.          Serial.print("Error on sending POST: ");
140.          Serial.println(httpResponseCode);
141.          }
142.          http.end();  //Free resources
143.        }
144.        else
145.        {
146.          Serial.println("Error in WiFi connection");
147.        }
148.        delay(500);  //Send a request every 10 seconds/
149.      //}
150.    }
151.    unsigned long getTime() {
152.      time_t now;
153.      struct tm timeinfo;
154.      if (!getLocalTime(&timeinfo)) {
155.      Serial.println("Failed to obtain time");
156.      return(0);
157.      }
158.      Serial.println(timeinfo.tm_hour-4);
159.      Serial.println(timeinfo.tm_min);
160.      time(&now);
161.      return now;
162.    }
```

APÊNDICE B – ARQUIVO DE CONFIGURAÇÃO BASE DO BANCO DE DADOS

```
{
  "name": "apitcc_roberto",
  "version": "0.0.1",
  "engines": {
    "node": "10.15.3"
  },
  "description": "API de conexao com ESP32",
  "git": "git@github.com:RobConhago/fistApi.git",
  "main": "server.js",
      "dependencies": {
        "body-parser": "^1.15.2",
        "express": "^4.14.0",
        "moment": "^2.24.0",
        "parse": "^2.8.0"
      },
      "scripts": {
        "start": "node server.js"
      },
      "author": {
        "name": "Roberto",
        "email": "****@gmail.com"
      }
    }
```

APÊNDICE C – CÓDIGO DA API DO BANCO DE DADOS

```
1. const Parse = require('./configs/database').Parse;
2. var moment = require('moment');
3.
4. const create = function () {
5.   return async function (req, res) {
6.       let prof_nome = req.body.prof_nome;
7.       let prof_rfid = req.body.prof_rfid;
8.       let horaInicio = req.body.horaInicio;
9.       let horaFim = req.body.horaFim;
10.           let sala  = req.body.sala;
11.           let diaSemana = req.body.diaSemana;
12.           //console.log('veja o professor que está vindo = ',
  prof_nome, prof_rfid);
13.           let Reserva = Parse.Object.extend("Reservas");
14.           let reserva = new Reserva();
15.           reserva.set("nomeProfessor", prof_nome);
16.           reserva.set("rfid", prof_rfid);
17.           reserva.set("horaInicio", horaInicio);
18.           reserva.set("horaFim", horaFim);
19.           reserva.set("sala", sala);
20.           reserva.set("diaSemana", diaSemana);
21.           let result = await reserva.save()
22.               .then( saved =>{
23.                 if(saved) return true;
24.                 return false;
25.               })
26.               .catch( erro => {
27.                   if(erro){
28.                       console.log('encontramos erro', erro);
29.                       return false;
30.                   }
31.               });
32.       };
33.     };
34.
35.     const hasAccess = function ()
36.     {
37.         return function( req, res) {
38.             let body = req.body;
39.             var Reserva = Parse.Object.extend("Reservas");
40.             let query = new Parse.Query(Reserva);
41.
42.             let timestemp =parseInt(body.timestamps,10);
43.             let time = new moment.unix(timestemp);
44.             time.add(-4, 'hours');
45.
```

```
46.             let weekday = time.weekday();
47.             let hour = time.hour();
48.             let min = time.minutes();
49.             let horas = moment(hour+''+min,"hmm").format("HH:m
   m");
50.         console.log("veja o que vem no body",body);
51.
52.             query.equalTo("rfid", body.rfid );
53.             query.equalTo("sala", body.sala );
54.             query.equalTo("diaSemana", weekday );
55.             query.lessThanOrEqualTo("horaInicio",horas);
56.             query.greaterThanOrEqualTo("horaFim",horas);
57.
58.             query.find()
59.                 .then( function(results){
60.                     console.log(results);
61.
62.                     if(results.length > 0){
63.                         res.send('liberado')
64.                     } else {
65.                         res.send('nao liberado')
66.                     }
67.                 }).catch(function(error){
68.                 console.log("Error: " + error.code + " " + err
   or.message);
69.                 res.send(error);
70.             });
71.         };
72.     };
73.
74.     const findSalas = function(){
75.         return function (req, res) {
76.             var Sala = Parse.Object.extend("Salas");
77.             let qry = new Parse.Query( Sala );
78.             console.log('analise isto res =',res);
79.             qry.find()
80.                 .then( results =>{
81.                     if(results.length > 0){
82.                         res.send(results)
83.                     } else {
84.                         res.send([]);
85.                     }
86.                 })
87.                 .catch(erro =>{
88.                     res.send(erro);
89.                 })
90.         }
91.     };
92.
```

```
93.     module.exports = {
94.         create,
95.         hasAccess,
96.         findSalas
97.     };
```

APÊNDICE D – ARQUIVO DE CONFIGURAÇÃO BASE DO SITE

```
{
  "name": "front-roberto",
  "version": "0.1.0",
  "engines": {
    "node": "10.15.3"
  },
  "private": true,
  "scripts": {
    "serve": "vue-cli-service serve",
        "build": "vue-cli-service build",
        "postinstall": "npm run build",
        "start": "node server.js",
        "lint": "vue-cli-service lint"
      },
      "dependencies": {
        "@mdi/font": "^3.6.95",
        "axios": "^0.19.0",
        "connect-history-api-fallback": "^1.6.0",
        "core-js": "^3.3.2",
        "express": "^4.17.1",
        "path": "^0.12.7",
        "register-service-worker": "^1.6.2",
        "roboto-fontface": "*",
        "serve": "^11.2.0",
        "serve-static": "^1.14.1",
        "vue": "^2.6.10",
        "vue-router": "^3.1.3",
        "vuetify": "^2.1.0",
        "vuex": "^3.0.1"
      },
      "devDependencies": {
        "@vue/cli-plugin-babel": "^4.0.0",
        "@vue/cli-plugin-pwa": "^4.0.0",
        "@vue/cli-plugin-router": "^4.0.0",
        "@vue/cli-plugin-vuex": "^4.0.0",
        "@vue/cli-service": "^4.0.0",
        "node-sass": "^4.12.0",
        "sass": "^1.19.0",
        "sass-loader": "^8.0.0",
        "vue-cli-plugin-vuetify": "^2.0.2",
        "vue-template-compiler": "^2.6.10",
        "vuetify-loader": "^1.3.0"
      }
    }
```

APÊNDICE E – ARQUIVO DE CONFIGURAÇÃO DE COMUNICAÇÃO DO SITE COM A API DO BANCO DE DADOS

```
1. import Vue from 'vue'
2. import Vuex from 'vuex'
3. import StatusSync from './modules/status_sync'
4. Vue.use(Vuex);
5. const axios = require('axios');
6. const iphost = "https://apiroberto.herokuapp.com";
7.
8. export default new Vuex.Store({
9.   modules: {
10.         StatusSync
11.       },
12.       state: {
13.         list_professores:[],
14.         horarios:[]
15.       },
16.       mutations: {
17.         setHorarios(state, horarios){
18.           this.horarios = horarios;
19.         }
20.       },
21.       actions: {
22.         async saveHorario({commit, state}, horario){
23.           let configPost = {
24.             method: 'post',
25.             url: iphost+ '/add_horario',
26.             data: horario
27.           };
28.           return  await axios(configPost)
29.               .then((request) => {
30.                 // eslint-disable-next-line no-console
31.                 console.log('veja isso:',request.data);
32.                 if (request.data) {
33.                   return true;
34.                 } else{
35.                   return false;
36.                 }
37.               })
38.               .catch((error) => {
39.                 // eslint-disable-next-line no-console
40.                 console.log(error);
41.                 return false;
42.               });
43.           },
44.         async downloadHorarios({ commit, state }){
```

```
45.             let configPost = {
46.               method: 'get',
47.               url: iphost+ '/find_salas',
48.               data: {}
49.             };
50.             return  await axios(configPost)
51.                 .then((request) => {
52.                   if (request.data && request.data.length > 0) {

53.                     commit('setHorarios', request.data);
54.                     console.log('isso que veio do banco',request
   .data);
55.                     return true;
56.                   } else{
57.                     return false;
58.                   }
59.                 })
60.                 .catch((error) => {
61.                   return false;
62.                 });
63.           }
64.         },
65.         getters:{
66.           listProfessores(state){
67.             if ( state.list_professores.length=== 0 ){
68.               state.list_professores.push( {
69.                 nome_prof:'Professor A',
70.                 rfid:'16 1A 83 CB'
71.               } );
72.               state.list_professores.push( {
73.                 nome_prof:'Professor B',
74.                 rfid:'50 69 7A 40'
75.               } );
76.             }
77.             return state.list_professores;
78.           }
79.         }
80.       })
```

APÊNDICE F – ARQUIVO DE CADASTRO DE HORÁRIOS NO SITE

```
<template>
  <v-container fluid>
    <v-row
            class="my-3"
            justify="center"
            align="center">
      <div class="TitleGeral">
        CADASTRO HORÁRIO
      </div>
        </v-row>
        <v-row
                class="my-3"
                justify="center"
                align="center">
          <v-col class="d-flex" cols="12">
            <v-select
                    :items="professores"
                    v-model="prof_select"
                    label="Professor:"
                    outlined
            ></v-select>
          </v-col>
          <v-col class="d-flex" cols="12" sm="6">
            <v-select
                    v-model="horainicio_select"
                    :items="horaInicio"
                    label="Hora início:"
                    outlined
            ></v-select>
          </v-col>
          <v-col class="d-flex" cols="12" sm="6">
            <v-select
                    v-model="horafim_select"
                    :items="horaFim"
                    label="Hora fim:"
                    outlined
            ></v-select>
          </v-col>
          <v-col class="d-flex" cols="12">
            <v-select
                    v-model="diasSelect"
                    :items="diasSemana"
                    label="Dia da Semana:"
                    multiple
                    outlined
            ></v-select>
```

```
47.           </v-col>
48.           <v-col class="d-flex" cols="12">
49.             <v-select
50.                    v-model="salaSelect"
51.                    :items="salas"
52.                    label="Sala:"
53.                    outlined
54.             ></v-select>
55.           </v-col>
56.           <v-col cols="12">
57.             <v-row align="center"
58.                    justify="center" class="px-3">
59.               <v-btn
60.                       block
61.                       color="blue"
62.                       v-on:click="salvarhorario"
63.                       outlined>
64.                 SALVAR
65.               </v-btn>
66.             </v-row>
67.           </v-col>
68.         </v-row>
69.       </v-container>
70.     </template>
71.     <script>
72.       // @ is an alias to /src
73.       import store from '../../store'
74.       export default {
75.         name: 'CadastroHorario',
76.         components: {
77.         },
78.         data:()=>({
79.             horarios:[
80.
81.               "07:30",
82.               "08:20",
83.               "09:10",
84.               "10:00",
85.               "10:50",
86.               "11:40",
87.               "12:30",
88.               "13:00",
89.               "13:50",
90.               "14:40",
91.               "15:30",
92.               "16:20",
93.               "17:10",
94.               "18:00",
95.               "18:50",
```

```
96.                 "19:40",
97.                 "20:30",
98.                 "21:20",
99.                 "22:10",
100.              ],
101.            horainicio_select:'',
102.            horafim_select:'',
103.            prof_select:'',
104.            diasSelect:[],
105.            salaSelect:''
106.          }),
107.          methods:{
108.            salvarhorario(){
109.              for(let i=0; i< this.diasSelect.length; i++){
110.                let horario = {
111.                  prof_nome: this.prof_select.nome_prof,
112.                  prof_rfid: this.prof_select.rfid,
113.                  horaInicio: this.horainicio_select,
114.                  horaFim: this.horafim_select,
115.                  diaSemana: this.diasSelect[i],
116.                  sala: this.salaSelect
117.                };
118.                console.log(horario);
119.                store.dispatch('saveHorario', horario);
120.              }
121.            }
122.          },
123.          computed:{
124.            horaInicio(){
125.              return this.horarios;
126.            },
127.            horaFim(){
128.              return this.horarios.filter( item => item > this.h
    orainicio_select );
129.            },
130.            professores(){
131.              return store.getters.listProfessores.map( (item) =
    >  {
132.                return {
133.                  text: item.nome_prof,
134.                  value: item,
135.                }
136.              });
137.            },
138.            diasSemana(){
139.              let dias = [
140.                {
141.                  text:'Segunda',
142.                  value:1
```

```
143.                 },
144.                 {
145.                   text:'Terça',
146.                   value:2
147.                 },
148.                 {
149.                   text:'Quarta',
150.                   value:3
151.                 },
152.                 {
153.                   text:'Quinta',
154.                   value:4
155.                 },
156.                 {
157.                   text:'Sexta',
158.                   value:5
159.                 },
160.                 {
161.                   text:'Sábado',
162.                   value:6
163.                 },
164.               ];
165.               return dias;
166.             },
167.             salas(){
168.               let dias = [
169.               ];
170.               for (let i=0; i< 26; i++){
171.                 dias.push("A"+(i+1))
172.               }
173.               for (let i=0; i< 26; i++){
174.                 dias.push("B"+(i+1))
175.               }
176.               for (let i=0; i< 26; i++){
177.                 dias.push("C"+(i+1))
178.               }
179.               return dias;
180.             }
181.           }
182.         }
183.     </script>
```

APÊNDICE G – ARQUIVO DE CADASTRO DE PROFESSORES NO SITE

```
<template>
  <v-container fluid>
    <v-row
            class="my-3"
            justify="center"
            align="center">
      <div class="TitleGeral">
        CADASTRO PROFESSOR9
      </div>
        </v-row>
        <v-row
                class="my-3"
                justify="center"
                align="center">
            <v-col class="d-flex" cols="12" sm="6">
                <v-select
                        :items="professores"
                        label="Professor:"
                        outlined
                ></v-select>
            </v-col>
          <v-col cols="12">
            <v-row align="center"
                   justify="center">
              <v-btn
                      color="blue"
                      outlined>
                SALVAR
              </v-btn>
            </v-row>
          </v-col>
        </v-row>
      </v-container>
    </template>
    <script>
      // @ is an alias to /src
        import store from '../../store'
        export default {
            name: 'CadastroProfessor',
            components: {
            },
            data:()=>({

            }),
            computed:{
                professores(){
```

```
47.                     return store.getters.listProfessores;
48.                 }
49.             }
50.         }
51.     </script>
```