



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO AMAZONAS
CAMPUS MANAUS DISTRITO INDUSTRIAL
ENGENHARIA DE CONTROLE E AUTOMAÇÃO**



CARLOS KENNETON ROCHA DE ALMEIDA

**ESTUDO E IMPLEMENTAÇÃO DE UMA ARQUITETURA DE
HARDWARE E *SOFTWARE* PARA CONTROLE DE ACESSO AOS
LABORATÓRIOS DO IFAM-CMDI UTILIZANDO INTERNET DAS
COISAS (IOT)**

MANAUS - AM

2018

CARLOS KENNETON ROCHA DE ALMEIDA

**ESTUDO E IMPLEMENTAÇÃO DE UMA ARQUITETURA DE
HARDWARE E SOFTWARE PARA CONTROLE DE ACESSO AOS
LABORATÓRIOS DO IFAM-CMDI UTILIZANDO INTERNET DAS
COISAS (IOT)**

Trabalho de Conclusão de Curso
apresentado como requisito parcial à
obtenção do título de Bacharel em
Engenharia de Controle e Automação, do
Instituto Federal de Educação, Ciência e
Tecnologia do Amazonas, Campus
Manaus Distrito Industrial – IFAM/CMDI.

Orientador: Prof. Me. Hillermann Ferreira
Osmídio Lima.

MANAUS - AM

2018



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO AMAZONAS
CAMPUS MANAUS DISTRITO INDUSTRIAL
ENGENHARIA DE CONTROLE E AUTOMAÇÃO**



ESTUDO E IMPLEMENTAÇÃO DE UMA ARQUITETURA DE *HARDWARE* E *SOFTWARE* PARA CONTROLE DE ACESSO AOS LABORATÓRIOS DO IFAM-CMDI UTILIZANDO INTERNET DAS COISAS (IOT)

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Controle e Automação do Campus Manaus Distrito Industrial, Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (CMDI/IFAM), como requisito parcial para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Aprovado em ____ de _____ de 2018.

BANCA EXAMINADORA

Prof. Me. HILLERMANN FERREIRA OSMÍDIO LIMA
Orientador

Prof. Me. ADRIANO BRUNO DOS SANTOS FRUTUOSO
Professor Avaliador

Prof. Me. ANDRÉ BELTRÃO DE LUCENA
Professor Avaliador

AGRADECIMENTOS

Agradeço primeiramente a Deus, força maior que rege as leis do universo, pela saúde, por me proporcionar a capacidade de raciocínio, a ponto de não haver limites para o conhecimento e todas as bênçãos sobrepostas em minha vida.

A minha família materna, dos quais tenho o maior orgulho do mundo, em especial, minha mãe Vanusa, padrasto Antonio, Avô Raimundo, Avó Lourdes e meu irmão Kalill, por sempre me incentivaram a fazer o que eu gostasse inclusive me deslocar do interior do Amazonas para a capital para realizar este curso. Agradeço também ao meu irmão de batalha Adrisson Galvão, pelas dicas sobre quais linguagens usar no sistema *web*, por alguns códigos disponibilizados, pela grande amizade e os anos de programação.

Sou muito grato também pela orientação do professor Hillermann, mas também por suas aulas, das quais foram de grande aprendizado neste curso, considerado não só por mim, mas por diversos outros colegas de classe como um dos grandes exemplos de profissionais a ser seguido, com certeza uma grande referência na área de sistemas embarcados e na vida como ser humano.

RESUMO

Atualmente, com a alta velocidade no desenvolvimento de novas tecnologias, se busca cada vez mais utilizá-las com o intuito de melhorar a vida do ser humano, seja de modo direto ou indireto. Este trabalho visa o desenvolvimento de um sistema de controle de acesso que possa ser utilizado nos laboratórios do IFAM - Campus Manaus Distrito Industrial, registrando horários, reservas, usuários e laboratórios utilizados, podendo, através de pequenos ajustes, também ser aplicado ao controle de acesso em empresas, escolas e qualquer outro local que disponha de tais ambientes tratados no âmbito deste trabalho. O controle é feito através de um sistema *web*, podendo ser acessado de qualquer lugar dependendo dos requisitos dos usuários, porém, para o caso específico do IFAM-CMDI, ficará restrito ao uso interno da instituição. Os principais pontos de inovação e tecnologia utilizados são, a IoT (*Internet of things*) utilizando o módulo NodeMCU, juntamente com o sistema embarcado Raspberry Pi 3 funcionando como servidor e o desenvolvimento e utilização dos meios de comunicação entre os dispositivos de todo sistema.

Palavras-chave: NodeMCU. Sistema *web*. Raspberry Pi 3. Protocolos. Internet das Coisas.

ABSTRACT

Nowadays, with the high speed in the development of new technologies, it is increasingly sought to use them in order to improve the life of the human, being either directly or indirectly. This work aims at the development of an access control system that can be used in laboratories of IFAM - Campus Manaus Distrito Industrial, recording schedules, reservations, users, used laboratories and etc. Through small adjustments, it can also be applied to access control in companies, schools and any other place that has environments such as treated in the scope of this work, being controlled through a web system from anywhere depending on the requirements of the users, to this particular case will be restricted to internal use on the institution's campus. The main points of innovation and technology used are IoT (Internet of Things) using the NodeMCU module, together with the embedded Raspberry Pi 3 system as a server and the development and use of the media between the devices of the whole system.

Keywords: NodeMCU. Web system. Raspberry Pi 3. Protocols. Internet of Things.

LISTA DE ILUSTRAÇÕES

Figura 1: Fluxograma de desenvolvimento do projeto.....	14
Figura 2: Internet das Coisas.	17
Figura 3: NodeMCU.	19
Figura 4: Raspberry Pi 3, modelo B.	20
Figura 5: Módulo MFRC522.	24
Figura 6: Função que monta o JSON para o acesso aos laboratórios.	26
Figura 7: Fluxograma do código da NodeMCU dos laboratórios.....	27
Figura 8: Função que monta o JSON para o cadastro de usuários.....	28
Figura 9: Fluxograma do código da NodeMCU do administrador.....	28
Figura 10: Página de login.....	30
Figura 11: Página de listar reservas.....	31
Figura 12: Página de cadastro de usuários.....	32
Figura 13: Página de listar usuários.....	32
Figura 14: <i>Modal</i> de atualização de usuário.....	33
Figura 15: <i>Modal</i> de exclusão de usuário.....	33
Figura 16: Página de cadastro de reservas.....	34
Figura 17: Página de exclusão de reservas.	35
Figura 18: Página de cadastro de laboratórios.....	35
Figura 19: Página de listar laboratórios.....	36
Figura 20: Página de listar acessos.	37
Figura 21: Protótipo utilizado para testes.	38
Figura 22: Esquema elétrico da placa de cadastro de usuários.....	39
Figura 23: Esquema elétrico da placa de acesso aos laboratórios.	40
Figura 24: PCI responsável pelo cadastro de usuários.....	42
Figura 25: PCI responsável pelo acesso aos laboratórios.	42

LISTA DE ABREVIATURAS

IoT	Internet of Things
RFID	Radio Frequency Identification
SQL	Structured Query Language
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
GPIO	General Purpose Input/Output
PWM	Pulse Width Modulation
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
ADC	Analog to Digital Converter
USART	Universal Synchronous Asynchronous Receiver Transmitter
JSON	JavaScript Object Natation
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator
XML	eXtensible Markup Language
EEPROM	Electrically-Erasable Programmable Read-Only Memory
I/O	Input / Output
PCI	Placa de Circuito Impresso
RTOS	Real Time Operating System
AJAX	Asynchronous JavaScript And XML
RTC	Real Time Clock
NTP	Network Time Protocol
PHP	Personal Home Page

SUMÁRIO

1	INTRODUÇÃO	10
1.1	JUSTIFICATIVA.....	12
1.2	OBJETIVO GERAL.....	12
1.3	OBJETIVOS ESPECÍFICOS	13
2	METODOLOGIA.....	14
2.1	FERRAMENTAS UTILIZADAS.....	15
3	REFERÊNCIAL TEÓRICO	16
3.1	NODEMCU	17
3.2	RASPBERRY PI 3	19
3.3	SISTEMA WEB.....	20
3.3.1	NodeJS.....	21
3.3.2	Express.....	22
3.3.3	MySQL.....	22
3.3.4	AngularJs e Bootstrap	22
3.3.5	Módulo MFRC522.....	23
4	DESENVOLVIMENTO DO SOFTWARE	25
4.1	PROGRAMAÇÃO DA NODEMCU.....	25
4.1.1	Programação da NodeMCU para acesso aos laboratórios.....	26
4.1.2	Programação da NodeMCU para o cadastro de usuários	27
4.2	DESENVOLVIMENTO DO SISTEMA <i>WEB</i>	28
4.2.1	Usuário comum.....	30
4.2.2	Usuário Administrador	31
5	DESENVOLVIMENTO DO <i>HARDWARE</i>.....	38
5.1	DISPOSITIVO PARA O CADASTRO DE USUÁRIOS.....	38
5.2	DISPOSITIVO PARA ACESSO AOS LABORATÓRIOS	39
6	ANÁLISE DOS RESULTADOS.....	41
7	CONSIDERAÇÕES FINAIS	43
	REFERÊNCIAS.....	44
	APÊNDICE A - Repositório do GitHub.....	46

1 INTRODUÇÃO

Com base na evolução tecnológica, podemos observar as modificações, a expansão e as facilidades encontradas na maioria dos ambientes comuns do nosso dia a dia, dentre estes, nossas casas, carros, restaurantes, universidades e etc. Porém, por trás de grande parte dessas aplicações podemos citar os sistemas embarcados, responsáveis por todo gerenciamento de dados e manipulação do meio físico.

Os sistemas embarcados são utilizados em uma grande diversidade de aplicações, desde as mais simples como ligar e desligar uma lâmpada, até as mais complexas, como um sistema de controle de um ônibus espacial. Há projetos de sistemas computacionais extremamente complexos, por envolver conceitos até agora pouco analisados pela computação de propósitos gerais, como por exemplo, as questões da portabilidade e do limite de consumo de potência sem perda de desempenho, a baixa disponibilidade de memória, a necessidade de segurança e confiabilidade. (Carro e Wagner, 2003).

Os sistemas embarcados apresentam características em comum com sistemas computacionais de propósitos gerais, mas não possuem a uniformidade desses. Cada aplicação pode apresentar requisitos diferentes de desempenho, consumo de potência e área ocupada, o que vai acarretar em uma combinação distinta de módulos de hardware e software para atender estes requisitos. (Gervini, 2003).

Os sistemas de controle de acesso são cada vez mais disseminados em ambientes nos quais se faz presente o ser humano. Podemos citar: empresas, casas, hotéis e etc., e em conjunto a essa aplicação estão os sistemas embarcados, autenticando os acessos de acordo com os requisitos do sistema, acionando atuadores e processando outras informações.

Com a procura por segurança, comodidade e a facilidade em tarefas de registro de dados, os sistemas de controle de acesso ficam cada vez mais robustos, podendo utilizar tecnologias confiáveis e inteligentes a ponto de interagir cada vez mais com os usuários e tornar a vida dos mesmos mais simples nos mais diversos âmbitos do dia a dia. Em conjunto a essas funcionalidades surge a Internet das

Coisas, do inglês, *Internet of Things* (IoT), trazendo novos dispositivos, modelos de comunicação (protocolos), sistemas inteligentes e outros.

Uma das grandes áreas da atualidade onde em 100% das aplicações são utilizados sistemas embarcados é a Internet das Coisas, na qual todos os dias cresce com um número significativo de novas aplicações e dispositivos. Um destes será utilizado neste projeto, sendo o mesmo denominado NodeMCU.

O NodeMCU é composto basicamente por um módulo ESP8266, versão 12E, que possui interface Wi-Fi embutida, pinos de I/O, PWM, SPI, I2C, ADC, entre outras especificações. Pode também ser utilizado como ponto de acesso, como estação ou ainda os dois modos ao mesmo tempo, porém as duas grandes características desse chip é o tamanho e o preço, onde dependendo do modelo pode ser encontrado por aproximadamente 3 dólares a unidade, o que facilita bastante a não utilização de fios e conexão com outros dispositivos. (Oliveira, 2017).

Outro componente de grande utilidade atualmente e que será utilizado neste projeto é a placa Raspberry Pi 3, principalmente pela sua capacidade de processamento e realização de tarefas. Essa placa se encaixa na classe de mini PCs, por dispor de atributos como, alto poder de processamento se for comparada com plataformas como Arduino, baixo custo quando comparada a um PC e tamanho reduzido, praticamente com as mesmas dimensões de um cartão de crédito. Neste projeto a mesma será utilizada como servidor e assim fará o controle de dados entre o sistema *web* e a NodeMCU.

O sistema *web* é uma das partes onde requer atenção especial no projeto, pois realiza a maioria das atividades nas quais há interação direta ou indireta com o usuário, sendo responsável por mostrar todas as informações referentes aos laboratórios, reservas, acessos e usuários. Este sistema também é responsável por fazer o gerenciamento e validações das reservas e acessos, como por exemplo, verificar no momento do acesso, se existe uma reserva já cadastrada para o devido laboratório, usuário, data e horário e após obter a resposta enviar um comando para a NodeMCU, para que a mesma libere ou não o acesso ao ambiente requisitado.

Para o sistema *web* foram utilizadas ferramentas atuais, onde algumas delas são desenvolvidas e utilizadas pela Google em seus sistemas. Para o desenvolvimento do *back-end* foi utilizada a plataforma NodeJs, o *framework*

Express para gerenciamento de rotas e outros recursos e o banco de dados MySQL e para o *front-end* foi selecionado o AngularJs.

1.1 JUSTIFICATIVA

O processo atual de reserva e utilização de um laboratório no IFAM-CMDI consiste normalmente de uma solicitação antecipada do espaço para a coordenação de laboratórios através de um sistema *web*, explicitando a data e horário de utilização, nome de quem vai utilizar e o nome do laboratório. Assim, na data da reserva o solicitante recebe a chave para acesso ao laboratório. Este sistema, apesar de funcional, pode apresentar possíveis falhas, principalmente no que diz respeito à devolução das chaves para a coordenação dos laboratórios. Segundo relatos, acontece eventualmente de usuários, após o uso do ambiente, por algum motivo não devolverem as chaves, o que pode prejudicar outros usuários que desejam fazer uso do mesmo ambiente, gerando discursões, que não beneficiam a instituição.

O sistema proposto neste trabalho elimina a necessidade de uso de chaves, uma vez que o acesso aos ambientes será através da tecnologia RFID, onde cada usuário receberá uma *tag* de identificação que será utilizada para abertura das portas. Outra facilidade que o sistema proporcionará é o registro de todos os acessos aos laboratórios em um banco de dados, informando o usuário, a data e hora do acesso.

Em conjunto com as melhorias mencionadas, será possível agregar novas funcionalidades em trabalhos futuros, aplicando cada vez mais o conceito de Internet das Coisas na instituição, transformando-a em um ambiente cada vez mais interativo e inteligente.

1.2 OBJETIVO GERAL

Desenvolver um sistema composto por *hardware* e *software* para facilitar o controle de acesso aos laboratórios do IFAM-CMDI, eliminando a necessidade da retirada e devolução de chaves na coordenação de laboratórios, além de manter um histórico dos acessos realizados por data e por usuário, através do uso da tecnologia de Internet das Coisas, RFID e programação *web*.

1.3 OBJETIVOS ESPECÍFICOS

- Desenvolver um sistema *web* que se comunique com um banco de dados que permita o cadastro e consulta de laboratórios, usuários e reservas, assim como a validação do acesso para um determinado usuário quando o mesmo tentar adentrar um laboratório através de sua *tag* RFID;
- Criar um banco de dados para armazenar as informações do sistema, como laboratórios, usuários, data e hora de reservas;
- Projetar um *hardware* capaz de fazer leitura de *tags* RFID e que possa conectar-se à rede lógica do IFAM-CMDI para que os dados sejam transmitidos e recebidos por um servidor conectado à mesma rede;
- Desenvolver, fabricar, montar e testar as placas de circuito impresso que se fizerem necessárias para alcançar as funcionalidades do sistema proposto;
- Validar a operação do sistema através de testes em ambiente compatível com os laboratórios do IFAM-CMDI.

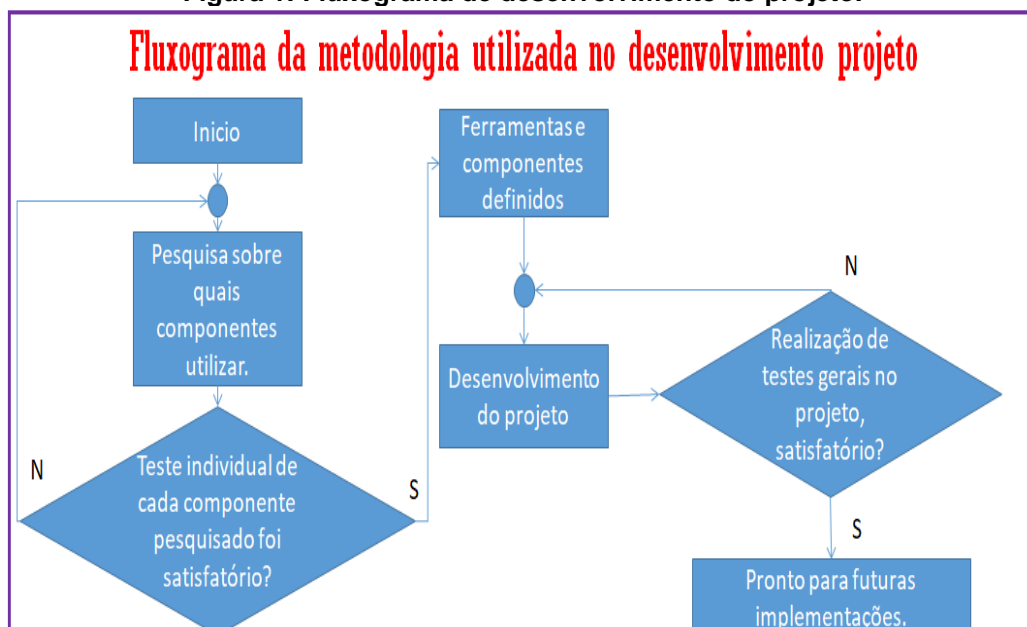
2 METODOLOGIA

O trabalho foi iniciado analisando as principais bibliografias que tratam do tema abordado, buscando dar ênfase a tecnologias no que se refere à Internet das Coisas e a viabilidade do sistema como um todo. Nesta fase foram verificadas as ferramentas de desenvolvimento para o sistema *web*, bem como a seleção do microcontrolador, o servidor e o protocolo de comunicação entre os dispositivos.

Após a seleção dos componentes utilizados no projeto foi iniciado o desenvolvimento dos códigos que serão executados tanto no microcontrolador (NodeMCU), como no servidor (Raspberry Pi 3). Um dos maiores desafios deste projeto foi o desenvolvimento do sistema *web*, para que o mesmo conseguisse ser executado em uma Raspberry Pi 3 renderizando bem as páginas, realizando a comunicação com o microcontrolador, enviando sinais para que o mesmo atue sobre a fechadura eletrônica, registrando usuários, laboratórios utilizados, datas e horários.

A última fase do desenvolvimento refere-se aos testes do sistema, fase esta muito importante para a longa vida do conjunto como um todo, de modo a prevenir falhas que venham acontecer quando o mesmo estiver em funcionamento.

Figura 1: Fluxograma de desenvolvimento do projeto.



Fonte: Autor, 2018.

2.1 FERRAMENTAS UTILIZADAS

Para o desenvolvimento do *hardware* foram utilizadas as seguintes ferramentas e plataformas:

- Placa NodeMCU versão 1.0;
- Raspberry Pi 3, modelo B;
- Leitor de RFID MFRC522 com *tags* do tipo cartão e chaveiro;
- Fechadura eletrônica acionada por um solenoide;
- Prototipadora LPKF para fabricação das placas.

Os *softwares* utilizados para o desenvolvimento do sistema *web*, do código para a plataforma NodeMCU e das placas de circuito impresso, foram:

- IDE do Arduino versão 1,8.3, para programação do NodeMCU;
- ATOM e o Visual Studio Code, para o desenvolvimento do sistema *web* e assim manipular as linguagens e *frameworks* abaixo;
 - NodeJs 8.9.4
 - Express 4.15.2
 - AngularJs 1.6.6
 - MySQL 2.13.0
- Proteus versão 8.2, para desenho do esquema elétrico e layout das placas de circuito impresso.

3 REFERÊNCIAL TEÓRICO

Devido ao atual momento da tecnologia mundial, diversas aplicações surgem a cada dia, ou até mesmo processos já estabelecidos são atualizados por métodos mais aprimorados, descobertos recentemente. Devido a tais situações os sistemas embarcados evoluem da mesma forma em suas especificações, como, memórias com mais capacidade de armazenamento, maiores velocidades de processamento, diversidade de tipos de comunicação, maior quantidade de portas de I/O, canais para conversão AD (analógico-Digital) e DA (digital-analógico), etc.

O aumento desta flexibilidade e complexidade impacta diretamente no esforço de desenvolvimento destes sistemas. Uma das tendências tecnológicas é o desenvolvimento de todo sistema em um único chip. Os SoCs (do inglês, System-on-a-Chip), como são chamados, consistem no hardware e também no software que o compõe. (Vargas, 2010).

Junto a tal necessidade de melhorar os processos, surge uma nova área dentro da automação que é a Internet das Coisas, que busca conectar diversos dispositivos entre si através da *Internet*, da melhor forma possível trabalhando em altos níveis de padrões e protocolos. Para Diniz (2006), “A ideia por trás da Internet das Coisas nasce de uma nova dimensão de conexão propiciada pela *Internet*, além de possibilitar a comunicação a qualquer tempo e em qualquer lugar, agora também considera para qualquer dispositivo”.

A internet das coisas é muito mais que apenas ligar lâmpadas pelo smartphone. Não é somente ligar “coisas” pela internet, mas também torná-las inteligentes, capazes de coletar e processar informações do ambiente ou das redes às quais estão conectadas. A internet das coisas está alterando radicalmente os ambientes e as formas de relacionamento entre as pessoas e os dispositivos no quais as mesmas interagem, por exemplo, meio ambiente, mobilidade, logística, energia, segurança e etc. (Oliveira, 2017).

Um dos pontos principais da Internet das Coisas em meio à vida do ser humano é a responsabilidade que o mercado impõe, exigindo que os produtos sejam capazes de desenvolver tarefas cada vez mais complexas que em as versões anteriores e tudo isso tendo ciclo de vida cada vez mais baixo. Sendo assim, a tendência é tornar os sistemas cada vez mais inteligentes, interagindo ainda mais com os usuários. A Figura 2 ilustra uma residência automatizada.

Figura 2: Internet das Coisas.



Fonte: Ferraz, 2016.

3.1 NODEMCU

A plataforma de *hardware* NodeMCU é comumente utilizada em aplicações para Internet das Coisas, por ser um componente que pode ser dedicado para interconexão entre equipamentos distintos, devido principalmente às funcionalidades que esta apresenta no que refere-se à conexão WiFi.

O dispositivo microcontrolado NodeMCU é produzido em larga escala na china de onde se origina e pode ser encontrado custando algo em torno de 3 dólares, sendo o baixo custo um dos grandes potenciais deste. O mesmo é constituído basicamente por um microcontrolador com um conjunto de instruções de 32 bits, que é adicionado ao mesmo o núcleo microprocessado Tensilica L106. A frequência de operação padrão é de 80 MHz, podendo ser expandido até 160 MHz. O processamento de suas funções básicas, no caso de protocolos WiFi e outras instruções de inicialização do chip, consomem 20% do processador, possibilitando que a capacidade disponibilizada para aplicações dos usuários utilizando o mesmo seja de 80%. O chip conta também com uma memória de programa de 4 MB. (Oliveira, 2017).

Um dos pontos de grande importância no mundo da tecnologia atualmente é o consumo de energia dos dispositivos e o chip ESP8266, modelo 12E, trata esta questão. Com alimentação nominal de 3,3 volts, a corrente no dispositivo varia de

acordo com o modo de utilização selecionado pelo usuário na programação. O chip consome apenas 20µA no modo *sleep*, cerca de 50mA quando conectado a um ponto de acesso em modo de recepção e, no modo transmissão, com potencia máxima de 17dBm, o consumo pode chegar até 170 mA. (Oliveira, 2017).

Outro ponto de destaque são as interfaces de entrada e saída, juntamente com as possibilidades de comunicação. O ESP8266, modelo 12E possui 17 GPIO, 4 saídas PWM, 1 entrada analógica com resolução de 10 bits, interfaces seriais síncronas, SPI e I2C e interface serial assíncrona, USART. Porém, o que torna o microcontrolador mais versátil é a sua capacidade de conectar-se a outros dispositivos através de sua interface WiFi, que pode ser utilizada no modo de ponto de acesso ou no modo estação. (Oliveira, 2017).

Segundo Kolban (2015), “o módulo ESP8266 envia dados de forma serial pela rede WiFi na qual está conectada. Uma das principais vantagens desse módulo é que não há necessidade da conexão física para o recebimento dos dados que serão transmitidos para a *Internet*”.

O dispositivo WiFi (designado ESP8266) é um componente com TCP/IP integrado que dá a qualquer microcontrolador acesso a uma rede WiFi, após ser programado para tal. O ESP8266 é capaz de hospedar ou mesmo buscar todas as funções da rede WiFi. Este módulo vem pré-programado com um conjunto *firmware* de comandos AT. (Ferreira 2016).

O NodeMCU pode ser programado utilizando a IDE do Arduino, Linguagem Lua desenvolvida no Brasil por professores e alunos da PUC-Rio e também no sistema operacional de tempo real (RTOS), específico para aplicações profissionais, explorando ainda mais a capacidade do microcontrolador e suas diversas funcionalidades.

A partir do microcontrolador ESP8266, surgiram diversos outros módulos, cada um buscando ainda mais aplicações para Internet das Coisas. Um dos mais populares é o módulo denominado NodeMCU, mostrado na Figura 3, que já agrega em si a plataforma de desenvolvimento em linguagem Lua e foi o módulo utilizado no desenvolvimento deste trabalho.

Figura 3: NodeMCU.



Fonte: Ferreira, 2016.

3.2 RASPBERRY PI 3

Com a disseminação do sistema operacional Linux e as facilidades que o mesmo oferece, sendo gratuito e muito estável na sua arquitetura, uma grande quantidade de aplicações foram desenvolvidas para serem utilizadas nesse sistema. Logo diversas arquiteturas, como é o caso do Intel x86 receberam versões do Linux, dentre essas mais conhecidas, a arquitetura ARM também recebeu sua versão, devido principalmente a sua escalabilidade de processos e o baixo consumo de energia. Aproveitando tal possibilidade e agregando o baixo custo, surgiu a *The Raspberry Pi Foundation*, responsável por lançar a plataforma de *hardware* Raspberry Pi, inicialmente nas versões A+ e B+ e chegando até a versão 3, modelo B. Recentemente foi lançada a plataforma Raspberry Pi Zero, sendo um modelo mais simples e de custo mais baixo ainda, chegando a 5 dólares no site oficial da fundação.

A Raspberry Pi se enquadra na classe dos mini PCs, devido a sua grande capacidade de processamento, portas para periféricos, como dispositivos USB, HDMI, saída de áudio, etc. Atualmente é utilizada em uma gama de aplicações, principalmente as que exijam muito desempenho, como processamento digital de imagens através da visão computacional, algoritmos de tempo real, entre outras.

A Raspberry Pi 3 é um computador pessoal de baixo custo. Em 2006 no Reino Unido, país de origem, esta placa foi lançada a um custo de \$35 dólares. A principal ideia dos criadores foi desenvolver um produto com preço acessível, tamanho reduzido e com diversas funcionalidades capazes de integrar facilmente no desenvolvimento de projetos eletrônicos com *software*. (Cruz, 2017).

Uma das grandes vantagens dessa placa é a disponibilidade de software para sistema Linux que podem ser utilizados na mesma. Ambientes de desenvolvimento em uma grande quantidade de linguagens, juntamente com interfaces gráficas e APIs para integração de uma infinidade de dispositivos, sem contar com os servidores de banco de dados, protocolos de rede para as demais aplicações e vários outros recursos para desenvolvimento envolvendo sistemas embarcados são outras vantagens da plataforma. (Oliveira, 2017). A Figura 4 ilustra a Raspberry Pi 3, modelo que foi selecionado para o desenvolvimento deste projeto.

Figura 4: Raspberry Pi 3, modelo B.



Fonte: Cruz, 2017.

3.3 SISTEMA WEB

Os sistemas *web* são aplicações que utilizam um domínio, onde consta a estrutura funcional do sistema. Quando um usuário utiliza um navegador, o mesmo interpreta as informações e as exibe na tela, de acordo com o conjunto de instruções expresso no momento do desenvolvimento.

Uma das grandes vantagens de se utilizar aplicações *web* é a possibilidade de padronização nas mais diversas plataformas e independentes do sistema operacional, já que o navegador é o responsável pela interpretação das informações e este é padronizado.

Quando se utiliza sistemas *web*, todo o conteúdo de sua ferramenta ou site pode ser personalizado para ser utilizado em qualquer dispositivo. Com o conceito de *Design Responsivo*, uma técnica de estruturação utilizada para permitir

que o sistema funcione nos mais diversos dispositivos, como *smartphones* e *tablets*, é possível aumentar a capacidade de interação com o usuário. (Aguiar, 2017).

Neste trabalho foram utilizadas ferramentas específicas para o desenvolvimento de sistemas *web* de tempo real, buscando o máximo desempenho na renderização das páginas, reduzindo ao mínimo o trabalho do servidor em ações que podem ser executadas no próprio dispositivo que está acessando o sistema, incluindo celulares, computadores pessoais, *tablets*, etc.

3.3.1 NodeJs

É uma plataforma desenvolvida em JavaScript para construir aplicações de rede, rápidas e escaláveis. Segundo a NodeBR (2016), “o NodeJs usa um modelo de I/O direcionado a evento não bloqueante que torna os sistemas leves e eficientes, ideal para aplicações em tempo real com troca intensa de dados através de dispositivos distribuídos”.

O NodeJs surgiu na JSConf 2009 Européia, no qual um programador que atendia por Ryan Dahl, apresentou um projeto em que estava trabalhando. Este projeto era uma plataforma que combinava a máquina virtual JavaScript V8 da Google e um laço de eventos. O projeto apontava para uma direção diferente das outras plataformas em JavaScript que rodam no servidor: todos I/O primitivos são orientado a evento. (NodeBR, 2016).

Neste projeto o NodeJs é de extrema importância principalmente na gerência de atividades, proporcionando agilidade no processamento de informações, organizando de forma primordial as dependências do sistema, de modo a manipular as mesmas de modo singelo. Na utilização do NodeJs é explícito o conceito de programação orientada a evento, no qual quando um evento é detectado, por exemplo, um botão é pressionado, o NodeJs verifica a ocorrência do mesmo e atua com uma determinada ação.

O desenvolvimento de sistemas *web* em NodeJs é dividido basicamente em duas partes, a primeira é o *front-end*, responsável por obter as entradas fornecidas pelos usuários e no decorrer do processo esses dados são entregues à segunda, o *back-end*, cuja função é tomar as decisões já pré-estabelecidas no momento da criação do sistema.

3.3.2 Express

O Express é um *framework* desenvolvido para o gerenciamento de rotas, manipulação dos *middlewares*, que é um conceito essencial em NodeJs, manipulação de erros, integração com o banco de dados e outras funcionalidades.

Neste sistema o Express facilita bastante principalmente no que se refere à comunicação com o banco de dados, juntamente com o roteamento do sistema, no qual este *framework* faz a chamada das páginas de acordo com os links solicitados pelos usuários e ao mesmo tempo gerencia os *middlewares*, que são basicamente as requisições, as respostas do servidor e o próximo processo que será executado.

3.3.3 MySQL

O MySQL é uma plataforma de gerenciamento de banco de dados de alta performance, sendo relacional. Utiliza a linguagem SQL e atualmente pertence a gigante empresa do ramo da computação, Oracle. Possui bastante material disponível em livros e *Internet* no que diz respeito a desenvolvimento de sistemas.

O MySQL é bastante disseminado principalmente pela fácil utilização, logo também é muito utilizado por grandes empresas que dispõe de altos volumes de dados, sendo estas, NASA, Bradesco, Texas Instruments, Sony, entre outras. Outro ponto de suma importância é sua interface extremamente simples e compatível com a maioria dos sistemas operacionais. Assim pode-se dizer que essas são as características que fazem do MySQL o sistema de gerenciamento de banco de dados mais utilizado atualmente.

3.3.4 AngularJs e Bootstrap

A ideia por trás do AngularJs é de um *framework* MV (*Model - View*), ou seja, voltado para a interface com o usuário, podendo manipular as formas de visualização do sistema. Ele permite uma gama de possibilidades quando se fala de *layout*. Com tudo isso o intuito é transformar o HTML em uma linguagem para o desenvolvimento de interfaces *web* dinâmicas. Segundo Waltenberg (2016), “o AngularJs foi desenvolvido pela Google, e devido ao sucesso, ele é utilizado em mais de 1600 projetos.” Isso se deve principalmente às possibilidades de controle e

gerência do *front-end*, juntamente com a padronização dos *frames* nos mais diversos tipos de navegadores.

Neste projeto, o AngularJs controla toda a parte de animação dos botões, *frames*, menus, etc. É interessante ressaltar a facilidade da ferramenta no desenvolvimento do sistema, no qual basicamente foi feita uma página central e após isto foram feitas requisições AJAX controladas pelo AngularJs para que a página *web* não seja recarregada por completa a todo o momento, e sim que seja alterado apenas o *frame* que está sendo modificado no momento.

O *bootstrap* é um conjunto de ferramentas de *front-end* de código aberto para manipulação de HTML, CSS e JavaScript. Idealizado para auxiliar o programador principalmente com componentes pré-construídos, *plugins* criados em JQuery, juntamente com um sistema de *grid* responsivo, trazendo agilidade no desenvolvimento e melhorando o visual das páginas.

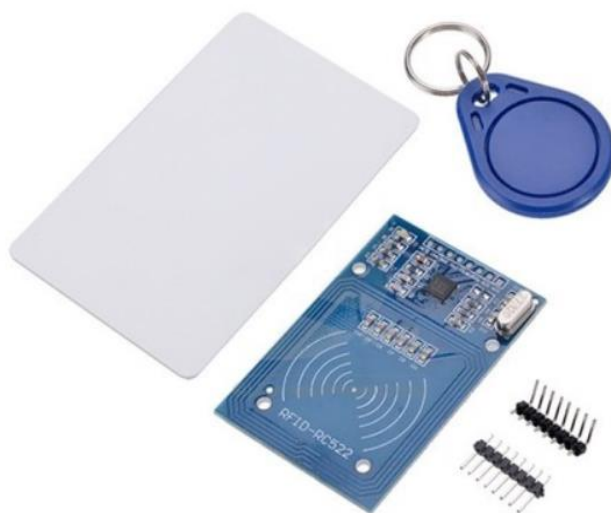
3.3.5 Módulo MFRC522

Uma das primeiras tecnologias ligadas ao conceito de Internet das Coisas surgiu em 1940, denominada RFID e vem sendo utilizada desde a segunda guerra mundial em aviões, no qual um dispositivo enviava um código único através de radiofrequência, e assim eles conseguiam identificar os aviões que estavam a uma determinada distância. Atualmente essa tecnologia é muito empregada em código de barras, veículos, controles remotos, etc.

Uma das áreas de aplicação desta tecnologia é no controle de acesso, empregado em residências, estabelecimentos comerciais, etc., proporcionando tanto segurança, quanto um controle maior sobre todo o processo de entrada e saída, registrando horários, pessoas e outras funções pertinentes para cada sistema.

O módulo MFRC522 de radiofrequência é extremamente utilizado para controle de acesso, sistemas de segurança e outras aplicações, principalmente devido ao seu baixo custo e consumo de energia. Este funciona a uma frequência de 13,56 MHz, possui tensão de alimentação de 3,3 volts, a corrente de trabalho varia entre 13-26mA, chegando a apenas 80μA no modo *sleep*. Utiliza comunicação SPI, e o mesmo normalmente vem de fábrica com duas *tags* de identificação, sendo uma em forma de cartão e a outra em chaveiro, como mostra a Figura 5.

Figura 5: Módulo MFRC522.



Fonte: FelipeFlop, 2017.

4 DESENVOLVIMENTO DO SOFTWARE

O *software* é parte essencial na maioria dos projetos de engenharia de controle e automação, principalmente quando se trata de Internet das Coisas, por determinar como será o funcionamento do sistema, ou seja, na medida em que o mesmo recebe entradas, os programas manipulam essas variáveis, realizando operações pré-estabelecidas no momento da criação dos códigos e assim constituem uma saída para que seja realizada uma determinada ação.

No desenvolvimento dos diversos *softwares* neste trabalho foram utilizadas diferentes linguagens de programação, divididas entre linguagens compiladas e interpretadas, e em alguns casos, as duas ao mesmo tempo, como é o caso do JavaScript, utilizada para o sistema *web*. Pode-se citar também a linguagem *Processing*, utilizada na plataforma Arduino, baseada na linguagem C++, facilitando a forma de programar a NodeMCU, já que esta é um sucesso no mundo todo por oferecer diversos recursos para os programadores de microcontroladores.

4.1 PROGRAMAÇÃO DA NODEMCU

Após o estudo das ferramentas que poderiam ser utilizadas no projeto, foi iniciada a programação do NodeMCU. Foi determinado que seriam necessários dois códigos distintos, sendo um para os laboratórios, com o objetivo de ler as *tags* e acionar o solenoide da fechadura eletrônica, e o outro para ler as *tags* e cadastrar um novo usuário, ficando este último na coordenação de laboratórios.

Primeiramente o objetivo foi fazer com que a NodeMCU pudesse, em conjunto com o módulo MFRC522 ler as *tags* RFID, e assim a determinar o código referente a cada uma delas. Após este passo, houveram ajustes para deixar o código de cada *tag* padronizado. Foi adotado que os caracteres lidos seriam transformados em hexadecimais no formato maiúsculo. Para isso foram utilizadas funções das bibliotecas MFRC522.h e SPI.h, já que a comunicação entre a NodeMCU e o módulo RFID é feita através de uma interface serial síncrona, SPI.

4.1.1 Programação da NodeMCU para acesso aos laboratórios

O *software* executado pelo módulo NodeMCU para gerenciamento dos laboratórios realiza diversas atividades, configuração e conexão com a rede WiFi, comunicação com o módulo MFRC522 para leitura dos *tags* RFID, comunicação com o servidor, indicação visual das operações através de LEDs e operação local do sistema em caso de falha na conexão com a rede WiFi.

A comunicação entre a NodeMCU e o servidor é feita através da utilização de um modelo para transmissão e armazenamento de informações no formato de texto, denominado JSON. Segundo Corrêa (2016) “esta ferramenta é adotada pela Google e Yahoo, no qual este modelo compacta significativamente a estrutura de informações de modo até melhor que o modelo já estabelecido no mercado, o XML”. Neste código, o modelo JSON foi utilizado para enviar para o servidor os dados “id_laboratorio” que é um código único de cada laboratório e “tag” que contém a identificação do usuário que tentou adentrar no ambiente, como ilustra a Figura 6.

Figura 6: Função que monta o JSON para o acesso aos laboratórios.

```
/*
 * função que armazena cada dado do sensor em um objeto JSON
 * utiliza a biblioteca ArduinoJson
 */
void montaJSON() {

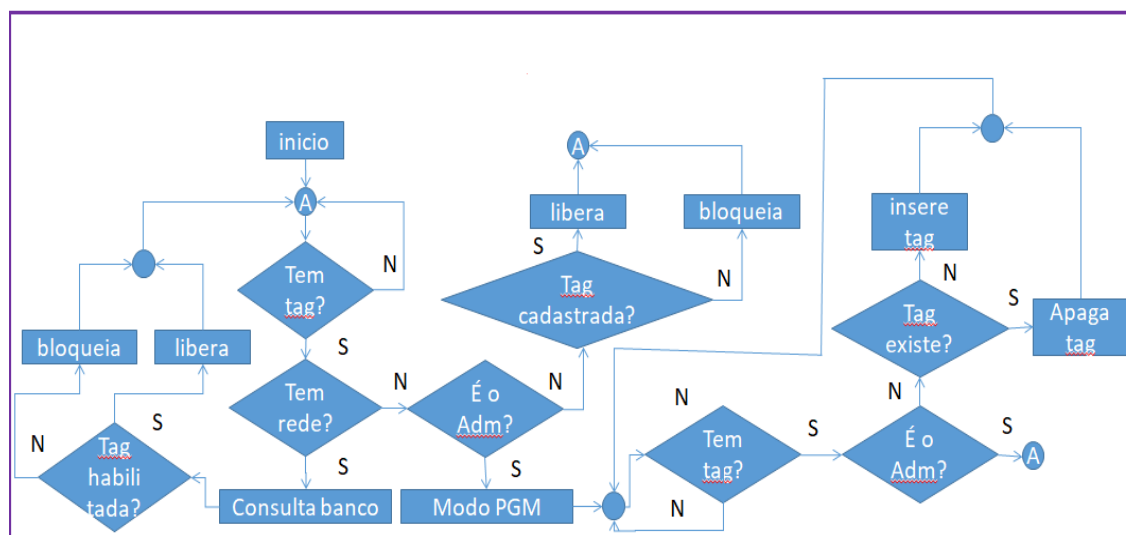
    dado["id_laboratorio"] = id_lab;
    dado["tag"] = tag;
}
```

Fonte: Autor, 2018.

Uma característica importante do dispositivo instalado em um laboratório é que o mesmo permite a entrada de usuários no ambiente, ainda que a NodeMCU não esteja conectada a uma rede WiFi, agregando ainda mais persistência no controle de acesso e tornando o mesmo ainda mais robusto. Para isso, faz-se necessário cadastrar a *tag* do usuário localmente na NodeMCU, que ficará armazenada na sua memória não volátil, do tipo EEPROM, capaz de reter os dados mesmo após o desligamento do dispositivo. Assim, quando não houver conexão com a rede WiFi, o dispositivo funciona de forma local. Na Figura 7 é mostrado o fluxograma do *software* da NodeMCU, na presença ou ausência de conexão com a rede WiFi. Vale ressaltar o significado de algumas siglas que aparecem neste

fluxograma, como por exemplo, “modo PGM”, que significa modo de Programação, “S” e “N”, quer dizer Sim e Não, respectivamente, “Adm”, que significa administrador e a letra “A”, dentro da circunferência, corresponde a um ponto de referência no fluxograma.

Figura 7: Fluxograma do código da NodeMCU dos laboratórios.



Fonte: Autor, 2018.

A primeira função a ser executada na inicialização da NodeMCU é a função `setup()`, onde estão as configurações iniciais do dispositivo, sendo esta executada uma única vez durante o funcionamento do dispositivo, a menos que o mesmo seja reiniciado ou a alimentação seja retirada e restabelecida. Após a execução da função `setup()`, entra em execução a função `loop()`, que possui o código responsável por todas as funcionalidades do dispositivo enquanto o mesmo estiver energizado.

4.1.2 Programação da NodeMCU para o cadastro de usuários

Sempre que houver um novo usuário para acessar um laboratório, o mesmo deverá receber uma *tag* RFID que por sua vez deverá ser cadastrada na coordenação de laboratórios, procedimento este, feito uma única vez.

Para o cadastro das *tags*, foi desenvolvido um dispositivo semelhante ao que é instalado nos laboratórios, porém com funcionalidades reduzidas, uma vez que não há necessidade de controle de fechadura eletrônica e nem gerenciar modos de operação local e remoto.

Neste caso, dentro da função que monta o JSON, só existe o atributo “tag”, como mostrado na Figura 8, já que para cadastrar o usuário não é necessário saber informações de laboratório.

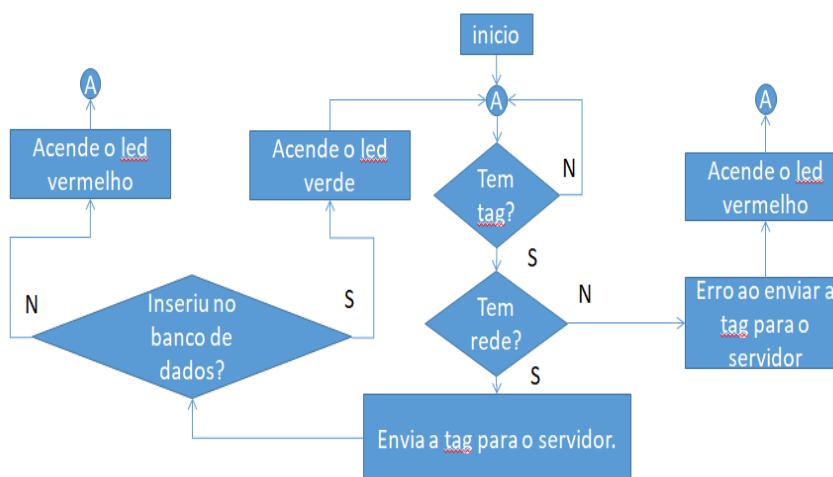
Figura 8: Função que monta o JSON para o cadastro de usuários.

```
/*
 * função que armazena cada dado do sensor em um objeto JSON
 * utiliza a biblioteca ArduinoJson
 */
void montaJSON() {
    dado["tag"] = tag;
}
```

Fonte: Autor, 2018.

Também não é necessário aguardar a resposta do servidor para liberar ou não o usuário após a leitura de uma *tag* RFID, uma vez que não existe a necessidade de ter fechadura eletrônica. Na Figura 9 pode ser visto o fluxograma do *software* para o dispositivo instalado na coordenação de laboratórios.

Figura 9: Fluxograma do código da NodeMCU do administrador.



Fonte: Autor, 2018.

4.2 DESENVOLVIMENTO DO SISTEMA WEB

O sistema *web* possui um papel muito importante no projeto e deve ser desenvolvido de forma que o usuário possa interagir de modo simples e objetivo, porém isso deve ser agregado ao que é chamado de *back-end*, ou seja, a parte do

sistema que executa acima do conhecimento do usuário, onde estão as funções por trás de cada operação, como um botão pressionado, a exibição de uma tela, etc.

Um dos grandes cuidados que deve se tomar ao iniciar o desenvolvimento de um sistema *web* é a escolha das ferramentas de trabalho, principalmente porque se deseja robustez no sistema, e no caso deste projeto, para que o mesmo seja executado com sucesso na placa Raspberry Pi 3, que apesar de ter um bom poder de processamento, não possui a mesma capacidade que um servidor comumente utilizado para esse tipo de aplicação.

Após o estudo e teste de algumas ferramentas para desenvolvimento de sistemas *web*, foram adotadas as ferramentas NodeJs, a linguagem JavaScript, o banco de dados MySQL, o Express para gerenciar atividades por parte do servidor e o AngularJs para o *front-end*, tendo o *template* admin-lte como base, devido às suas facilidades com ferramentas já prontas e bastante úteis para o desenvolvimento da parte visual.

No desenvolvimento do sistema foi previsto dois tipos de usuários. O primeiro é o administrador, com acesso liberado para operações em todo o sistema, como cadastrar, editar e excluir usuários, laboratórios e reservas.

O segundo tipo de usuário é o comum, que pode somente consultar a disponibilidade dos laboratórios em uma determinada data e hora, ficando sob a responsabilidade do administrador a efetivação da reserva.

A tela inicial do sistema é uma página de *login*, conforme a Figura 10. Ao acessar o sistema, o usuário é identificado com relação às suas permissões e o sistema se encarrega de disponibilizar somente as funções para as quais o usuário pode ter acesso.

As configurações de permissão das páginas foram aplicadas tanto no *back-end* quanto no *front-end* com criptografia nos dados sigilosos, para assim gerar um maior nível de segurança e confiabilidade no sistema. Outro conceito importante implementado é o de *token*, no qual após haver o *login* do usuário, é criada uma chave e para cada ação por parte do cliente, é verificada a autenticidade dessa chave, garantindo que não haja acesso interno ao sistema por um usuário, sem que este esteja cadastrado e tenha se autenticado na página de *login*.

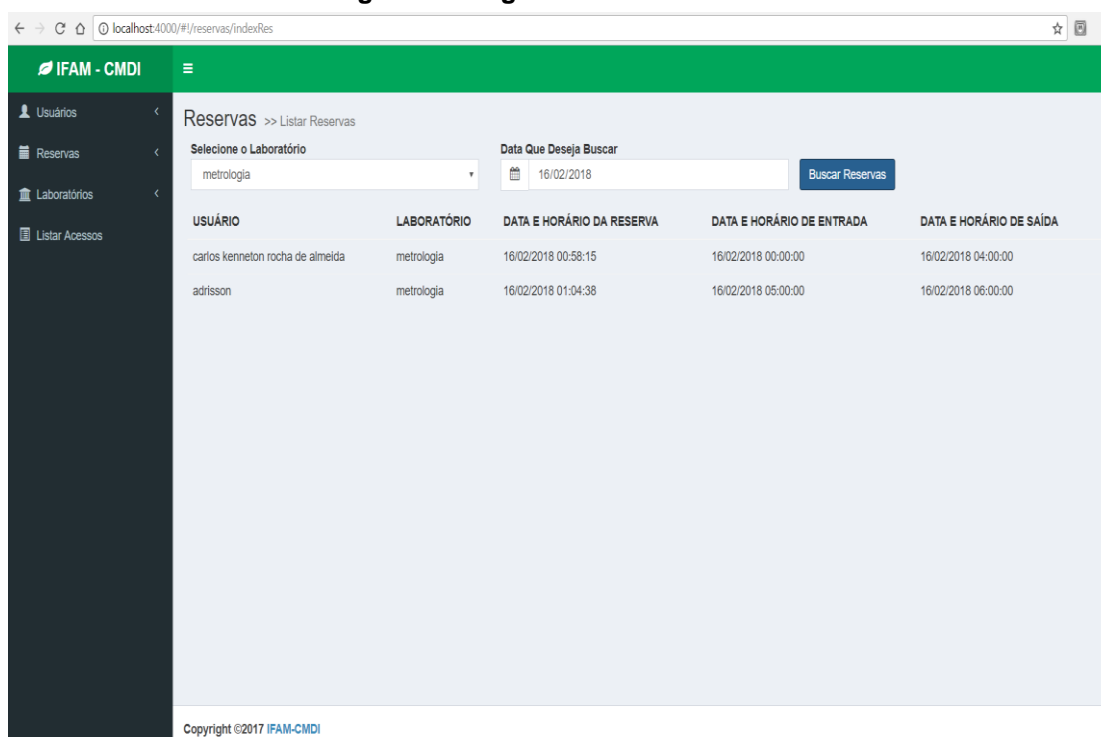
Figura 10: Página de login.

Fonte: Autor, 2018.

4.2.1 Usuário comum

No momento do cadastro do usuário, é escolhido o nível de permissão como mostrado anteriormente e no caso do usuário comum, o mesmo só tem acesso a uma página após ter realizado o *login*, que é a página “Listar Reservas”. Assim, o usuário comum pode realizar uma busca por um laboratório na data que deseja, proporcionando ao mesmo saber os horários disponíveis e assim solicitar uma reserva ao administrador. A Figura 11 mostra a página para listar as reservas. Vale ressaltar que o usuário comum não pode fazer uma reserva, podendo a mesma ser feita somente por um usuário com permissão de administrador.

Figura 11: Página de listar reservas.



Fonte: Autor, 2018.

4.2.2 Usuário Administrador

O usuário administrador, diferentemente do comum, tem acesso a todas as funcionalidades do sistema. Seguindo a sequência do menu, a aba de usuários, oferece os subitens “cadastrar” e “listar”. Ao clicar em “cadastrar”, a página carregada aparece com os campos dos formulários todos vazios, porém, para cadastrar um novo usuário é necessário aproximar o *tag* do leitor. Após esta ação, o LED verde da placa liga, quando então a página de cadastro deve ser atualizada e o sistema busca automaticamente a *tag* lida e insere no campo “Tag”. A Figura 12 ilustra a página de cadastro após a leitura da *tag*. Os outros campos devem ser preenchidos manualmente e a operação de cadastro é finalizada ao clicar no botão “Cadastrar Usuário”.

A outra opção do menu “Usuários” é “Listar”, que tem a função de mostrar os usuários já cadastrados, apresentado os mesmos de forma ascendente pelo primeiro nome. Para cada usuário listado pode-se atualizar informações ou excluir o mesmo. A Figura 13 ilustra a página responsável por listar os usuários.

Figura 12: Página de cadastro de usuários.

IFAM - CMDI

Usuários >> Cadastrar

Cadastrar um novo Usuário

Tag: 3AE697E5

Nome: informe o nome do usuário

Função Institucional: [dropdown]

Login: informe o login do usuário

Senha: informe a senha do usuário

☐ Administrador

Cadastrar Usuário

Copyright ©2017 IFAM-CMDI

Fonte: Autor, 2018.

Figura 13: Página de listar usuários.

IFAM - CMDI

Usuários >> Listar

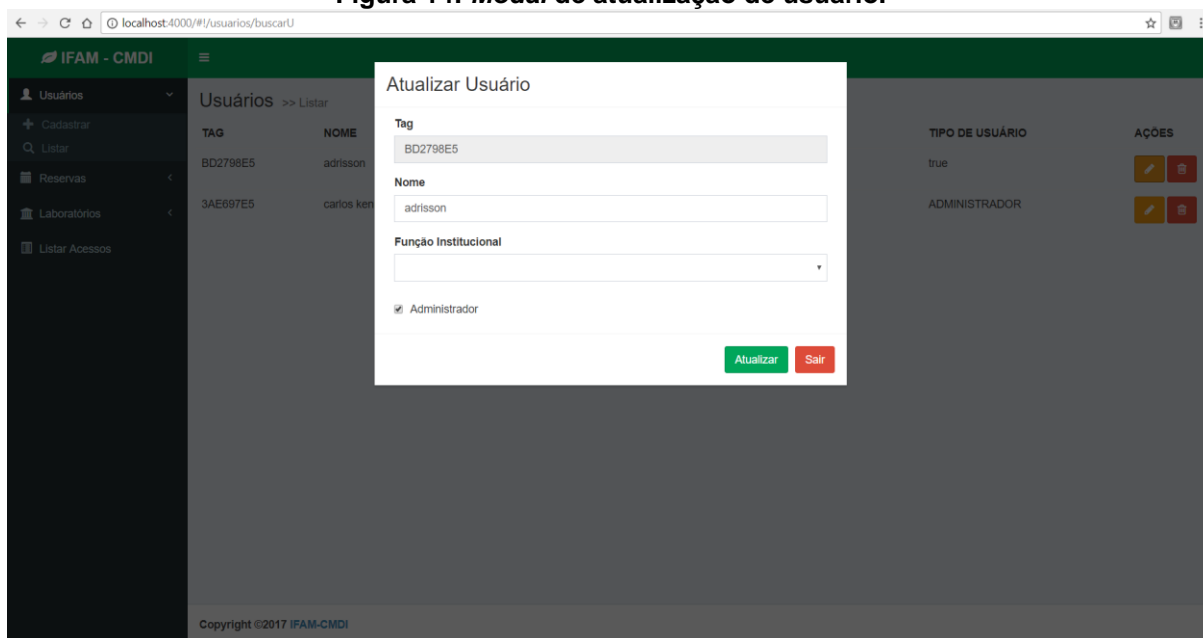
TAG	NOME	FUNÇÃO INSTITUCIONAL	TIPO DE USUÁRIO	AÇÕES
BD2798E5	adrisson	ALUNO	ADMINISTRADOR	[edit] [delete]
3AE697E5	carlos kenneton rocha de almeida	PROFESSOR	ADMINISTRADOR	[edit] [delete]

Copyright ©2017 IFAM-CMDI

Fonte: Autor, 2018.

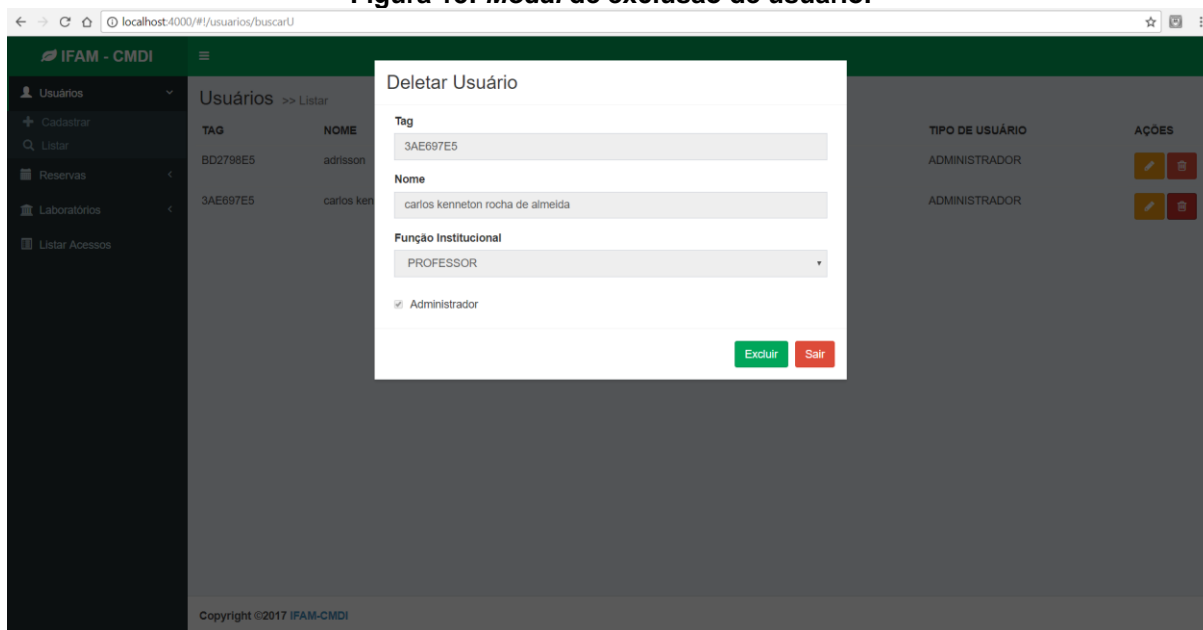
Como já mencionado, na página de “Listar” é possível atualizar ou excluir usuários e para isso foram criados *modals* que exibem as informações sobre o usuário que será atualizado ou excluído. Figura 14 e Figura 15 ilustram os *modals* de atualização e exclusão, respectivamente.

Figura 14: Modal de atualização de usuário.



Fonte: Autor, 2018.

Figura 15: Modal de exclusão de usuário.



Fonte: Autor, 2018.

Na aba “Reservas”, há três opções, “Listar Reservas”, “Cadastrar Reservas” e “Excluir Reservas”. A funcionalidade de “Cadastro de Reservas” é bem simples, principalmente pelo ambiente ser bem dinâmico. Nos campos “Nome do Usuário” e “Nome do Laboratório” é exibida uma lista de todos os usuários e laboratórios já cadastrados em ordem ascendente. A Figura 16 demonstra o ambiente de cadastro das reservas.

Figura 16: Página de cadastro de reservas.

IFAM - CMDI

Reservas >> Cadastrar

Cadastrar uma Reserva

Nome do usuário: adriisson

Nome do Laboratório: prog 2

Data de utilização: dd/mm/aaaa

Horário de entrada: --:--

Horário de Saída: --:--

Cadastrar Usuário

Copyright ©2017 IFAM-CMDI

Fonte: Autor, 2018.

A página de “Excluir Reserva” é semelhante à de “Listar Reservas”, porém possui uma funcionalidade extra que é a opção de excluir uma determinada reserva, ação esta que é efetivada ao clicar no ícone da lixeira, na coluna “Ações”, ao lado da reserva. Vale ressaltar que ao clicar na lixeira, surge um *modal* de confirmação de exclusão para a reserva. A Figura 17 ilustra a função de excluir reserva.

Figura 17: Página de exclusão de reservas.

IFAM - CMDI

Reservas >> Excluir

Selecione o Laboratório: metrologia

Data Que Deseja Buscar: 16/02/2018

Buscar Reservas

USUÁRIO	LABORATÓRIO	DATA E HORÁRIO DA RESERVA	DATA E HORÁRIO DE ENTRADA	DATA E HORÁRIO DE SAÍDA	AÇÕES
carlos kenneton rocha de almeida	metrologia	16/02/2018 00:58:15	16/02/2018 00:00:00	16/02/2018 04:00:00	
adriison	metrologia	16/02/2018 01:04:38	16/02/2018 05:00:00	16/02/2018 06:00:00	

Copyright ©2017 IFAM-CMDI

Fonte: Autor, 2018.

Na aba “Laboratórios” há duas opções, “Cadastrar” e “Listar”. A primeira possui apenas o campo “Nome”, já que o “id” do laboratório é gerado automaticamente, com o intuito de que o mesmo não se repita para dois laboratórios diferentes. A Figura 18 ilustra a página de cadastro.

Figura 18: Página de cadastro de laboratórios.

IFAM - CMDI

Laboratórios >> Cadastrar

Cadastrar um Novo Laboratório

Nome

Informe o nome do laboratório

Cadastrar Laboratório

Copyright ©2017 IFAM-CMDI

Fonte: Autor, 2018.

A segunda opção da aba “Laboratórios” é “Listar”, que é semelhante ao modelo utilizado para listar os usuários. Além de listar o nome dos laboratórios, também é listado o “ID” do mesmo, uma vez que este é necessário ser gravado na NodeMCU de cada laboratório. De forma análoga ao modelo de usuários, é possível atualizar ou excluir um laboratório através de *modals*. A Figura 19 mostra a página de listagem de laboratórios.

Figura 19: Página de listar laboratórios.

ID	NOME	AÇÕES
1	metrologia	[Edit] [Delete]
2	prog 1	[Edit] [Delete]
3	prog 2	[Edit] [Delete]
4	prog 3	[Edit] [Delete]

Fonte: Autor, 2018.

A última aba do sistema, “Listar Acessos”, possui apenas uma funcionalidade, que é listar os acessos feitos em um laboratório, em um determinado dia. Os acessos são registrados automaticamente pelo sistema, desde que a *tag* do usuário e o laboratório estejam cadastrados e exista uma reserva para a data e horário que o usuário está tentando adentrar no recinto. Neste caso, o LED verde da placa irá sinalizar que o acesso foi concluído e a fechadura será aberta, caso contrário o LED vermelho será ligado, indicando que pelo menos um dos requisitos apresentados anteriormente não foi validado. A Figura 20 mostra a lista de acessos, para um determinado laboratório em uma data.

Figura 20: Página de listar acessos.

IFAM - CMDI

Usuários <

Reservas <

Laboratórios >

+ Cadastrar

Q Listar

Listar Acessos

Acessos >> Listar Acessos

Selecione o Laboratório

metrologia

Data Dos Acessos Que Deseja Buscar

16/02/2018

Buscar Acessos

TAG	NOME	LABORATÓRIO	DATA E HORÁRIO DO ACESSO
3AE697E5	carlos kenneton rocha de almeida	metrologia	16/02/2018 01:33:30
3AE697E5	carlos kenneton rocha de almeida	metrologia	16/02/2018 01:37:02
3AE697E5	carlos kenneton rocha de almeida	metrologia	16/02/2018 01:39:00
3AE697E5	carlos kenneton rocha de almeida	metrologia	16/02/2018 01:39:48

Copyright ©2017 IFAM-CMDI

Fonte: Autor, 2018.

5 DESENVOLVIMENTO DO *HARDWARE*

O *hardware* é uma parte fundamental em um projeto de engenharia, tendo em vista que o mesmo estará no meio físico, interagindo diretamente com os usuários, basicamente analisando as entradas e gerando saídas, sendo gerenciado por um *software* que dispõe de funções preestabelecidas de acordo com a lógica da aplicação.

A parte física deste trabalho é composta basicamente por dois modelos de placas de circuito impresso, sendo uma destas para o cadastro do usuário (com utilização restrita ao administrador) e a outra que serve para o acesso aos laboratórios. Ambas as placas fazem leitura das *tags* RFID e para simular o acesso a um ambiente, foi desenvolvida uma miniatura de porta de madeira com uma fechadura eletrônica acionada por solenoide acoplada à mesma, como mostra a Figura 21.

Figura 21: Protótipo utilizado para testes.



Fonte: Autor, 2018.

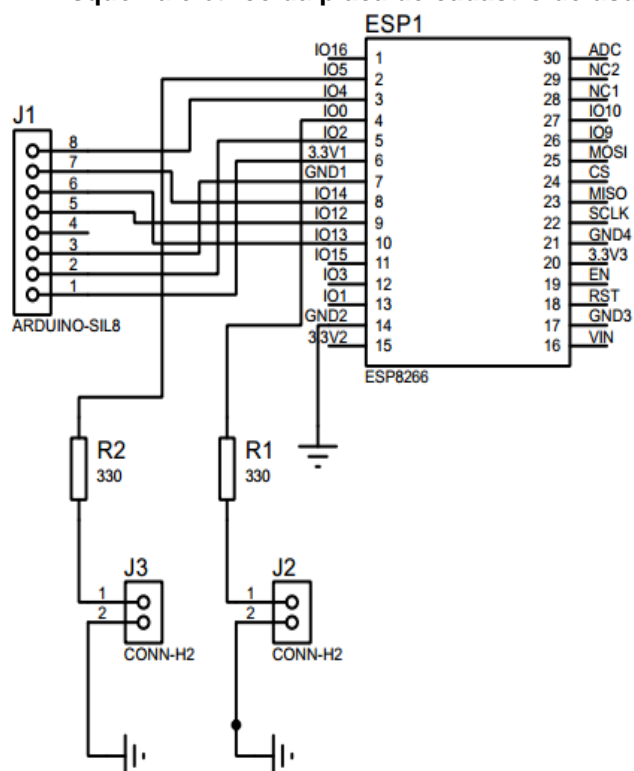
5.1 DISPOSITIVO PARA O CADASTRO DE USUÁRIOS

Tem por objetivo fazer a leitura das *tags* e enviar do código referente à *tag* lida para o servidor. O sucesso da operação é sinalizado por um LED verde e a falha por um LED vermelho. Os componentes dessa placa e seus respectivos *labels* são

apresentados no diagrama eletrônico da Figura 22. Estes são: NodeMCU identificado como ESP1, Leitor RFID representado pela barra de pinos J1, os resistores R1 e R2 e os dois LEDs, neste caso J2 e J3.

O sistema é alimentado através da NodeMCU, que possui uma entrada microUSB, onde é conectado o cabo de alimentação de +5V.

Figura 22: Esquema elétrico da placa de cadastro de usuários.



Fonte: Autor, 2018.

5.2 DISPOSITIVO PARA ACESSO AOS LABORATÓRIOS

Esta placa possui algumas funcionalidades a mais que a anterior, pois é necessário um circuito extra para acionamento da fechadura eletrônica, um botão para cadastrar uma *tag* mestre para o caso da rede ficar indisponível e ainda será acrescentado externamente à placa outro botão que ficará dentro dos laboratórios para abrir a porta para os usuários que desejam sair do laboratório ou para usuários que queiram entrar, mas não possuem *tag*.

Em resumo, esta placa faz a leitura das *tags*, envia os dados para o servidor e dependendo da resposta ela abre ou não a fechadura, indicando o resultado através de dois LEDs, um vermelho e um verde. Se houver sucesso na resposta do servidor, a fechadura eletrônica é acionada para abrir a porta, ou se a rede estiver

6 ANÁLISE DOS RESULTADOS

Após a realização dos testes previstos, foram obtidos resultados satisfatórios, inicialmente com relação ao sistema *web*, que se mostrou necessariamente estável, sendo possível explorar as funcionalidades de cadastro, consulta, atualização e exclusão dos laboratórios, usuários e reservas, além da validação dos acessos enviados pela NodeMCU, verificando, após a aproximação da *tag* no leitor RFDI, se existe reserva para o usuário, naquele laboratório e naquela data e horário.

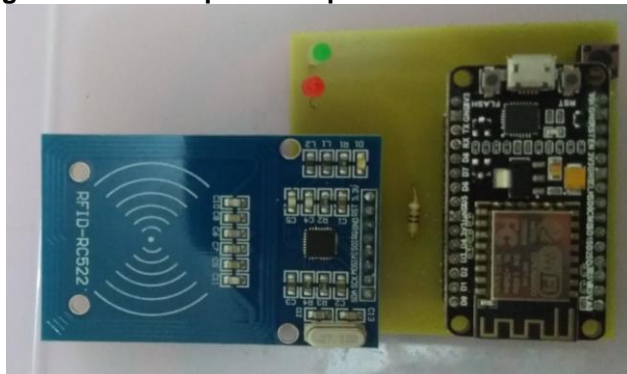
Um dos pontos de grande notoriedade são as facilidades apresentadas no sistema *web*, entre estas, o modelo visual auto-explicativo que deixa os *frames* bem distribuídos na tela, independente do tamanho da mesma. Neste âmbito, um ponto interessante são as requisições AJAX, feitas automaticamente pela ferramenta AngularJs, permitindo que as páginas *web* não sejam recarregadas por completo todas as vezes que houver uma pequena modificação, sendo atualizado apenas o *frame* modificado, tornando o carregamento das páginas muito mais rápido.

O banco de dados é uma das principais funcionalidades implementadas no sistema de controle de acesso, principalmente no que se refere ao armazenamento de informações e manipulação dos dados. Foram desenvolvidos sistemas básicos utilizando dois bancos de dados, o MongoDB e o MySQL, para que assim fosse selecionado o melhor para a presente aplicação. Como ambos apresentaram resultados semelhantes e satisfatórios, foi selecionado o MySQL por ser um banco relacional e o sistema possuir bastante relação entre as informações manipuladas. Outro fator é o conhecimento prévio do mesmo, comumente utilizado em diversas aplicações de sistemas embarcados que envolvem desenvolvimento *web* e armazenamento de informações.

No projeto do *hardware*, as funcionalidades definidas para as duas placas do sistema foram validadas através de testes e as mesmas responderam como esperado, na leitura das *tags* através do módulo MFRC522, na conexão com a rede WiFi, na comunicação com o sistema *web*, no acionamento da fechadura eletrônica e na sinalização visual através de LEDs.

As duas placas de circuito impresso do sistema foram fabricadas utilizando uma prototipadora e então foram montadas e testadas com sucesso, como pode ser visto na e .

Figura 24: PCI responsável pelo cadastro de usuários.



Fonte: Autor, 2018.

Figura 25: PCI responsável pelo acesso aos laboratórios.



Fonte: Autor, 2018.

As operações de leitura da *tag*, envio dos dados pela NodeMCU para o servidor e o recebimento da resposta, totalizaram um tempo médio de 2 segundos. Esse tempo depende de alguns fatores, como tráfego na rede, tamanho do pacote de dados e velocidade do servidor, porém, esse valor não possui grande margem de variação e no geral, o tempo de resposta ficou aceitável.

A operação da placa de controle de acesso aos laboratórios no caso da rede WiFi ficar indisponível funcionou satisfatoriamente. Mesmo interagindo com o usuário através de apenas dois LEDs, se mostrou uma ferramenta de alta importância para o projeto, permitindo o cadastro e exclusão de *tags*, assim como a validação local dos acessos aos laboratórios.

CONSIDERAÇÕES FINAIS

Tendo em vista que todos os objetivos previstos foram alcançados, pode-se considerar que os resultados obtidos trarão bastante benefícios aos usuários que utilizarem os ambientes tratados no âmbito deste trabalho, proporcionando facilidades nas mais diversas partes do sistema.

O modelo dessa arquitetura pode ser utilizado para desenvolver mais projetos dentro e fora da instituição, que poderão melhorar a qualidade de vida das pessoas, já que esse é o propósito da Internet das Coisas. É notória a necessidade de mais pesquisas nesta área, que busquem incentivar ainda mais o curso de Engenharia de Controle e Automação, proporcionando que outros alunos e professores desenvolvam ainda mais o sistema proposto nesse trabalho, agregando novas funcionalidades ao mesmo.

Com relação a melhorias neste sistema, uma das possibilidades seria a realização de um *backup* local das reservas, diariamente dentro da NodeMCU, podendo ser utilizada até uma memória externa em conjunto com Relógio de Tempo Real (RTC) ou um servidor NTP para agregar um relógio ao sistema, assim obtendo-se esses dados localmente. A partir do momento em que o sistema detectasse que não há conexão com a rede, a NodeMCU passaria a fazer consultas localmente, verificando o horário, data, laboratório e usuário de cada reserva e após restabelecer a conexão com a rede, os dados processados *offline* seriam enviados para o servidor e salvos no banco de dados. Atualmente, quando ocorre esta situação, a NodeMCU verifica apenas se a *tag* aproximada no leitor está cadastrada localmente na EEPROM da NodeMCU e, caso isso seja verdadeiro, a fechadura eletrônica é acionada para permitir o acesso ao ambiente.

REFERÊNCIAS

AGUIAR, Anderson. **Sistemas Web: conheça os benefícios das aplicações web**. 18 de agosto de 2017. Disponível em: <<https://secaoweb.com.br/blog/sistemas-web-conheca-os-beneficios/>>. Acesso em: 20 de setembro de 2017.

CARRO, Luigi; WAGNER, Flávio Rech. **Sistemas computacionais embarcados. Jornadas de atualização em informática**. Campinas: UNICAMP, 2003.

CORRÊA, Eduardo. **Introdução ao Formato JSON**. Disponível em: <<https://www.devmedia.com.br/introducao-ao-formato-json/25275>>. Acesso em 1 de dezembro de 2017.

CRUZ, Ariadne Arrais; LISBOA, Emerson Fausto. **Webhome—automação residencial utilizando Raspberry Pi**. *Revista Ciência e Tecnologia*. v. 17, n. 31, 2014.

DINIZ, Eduardo H. **Internet das coisas**. GV-executivo, 2006.

FERNANDES, Anamaria Coutinho et al. **Sistema de aquisição de sinais ECG processado pelo LabVIEW com comunicação wi-fi por meio do módulo ESP8266**. 2017.

FERRAZ, Reinaldo. **Internet das Coisas**. 11ª Mostra de Iniciação Científica Júnior, 2016.

FERREIRA, Ricardo. **Gestão de uma estufa**. 2016.

FILIPEFLOP, **Controle de Acesso usando Leitor RFID com Arduino**. 22 de abril de 2014. Disponível em: <<https://www.filipeflop.com/blog/controle-acesso-leitor-rfid-arduino/>>. Acesso em: 10 de junho de 2017.

GERVINI, Alexandre I. et al. **Avaliação de Desempenho, Área e Potência de Mecanismos de Comunicação em Sistemas Embarcados**. Seminário Integrado de Software e Hardware, 2003.

GUERRA, Filipe Henrique Moreira. **Automação residencial de baixo custo com protocolo X10 e ESP8266**. 2016.

KOLBAN, Neil. **Kolban's Book on ESP8266**. Texas, USA, 2015.

NodeBR. **O que é Node.js?**, 14 de novembro de 2016. Disponível em: <<http://nodebr.com/o-que-e-node-js/>>. Acesso em: 10 de setembro de 2017.

OLIVEIRA, Sergio. **Internet das Coisas com ESP8266**, Arduino e Raspberry Pi. São Paulo: Novatec, 2017.

SCHROEDER, Ricardo; DOS SANTOS, Fernando. **Arquitetura e testes de serviços web de alto desempenho com node.js e mongodb**.

TILKOV, Stefan; VINOSKI, Steve. Node.js: **Using JavaScript to build high-performance network programs**. *IEEE Internet Computing*. v. 14, n. 6, p. 80-83, 2010.

VARGAS, Rafael et al. **Sistemas embarcados: acoplamento do soft-core plasma ao barramento opb de um powerpc 405**. 2010.

WALTENBERG, Rodrigo. **AngularJs: o que é e porquê utilizar?**. 28 de junho de 2016. Disponível em: <<http://blog.algaworks.com/o-que-e-angularjs/>>. Acesso em: 1 de dezembro de 2017.

APÊNDICE A - Repositório do GitHub

Repositório do GitHub contendo todos os códigos do NodeMCU e o do sistema web:

- https://github.com/carloskalill10/laboratorios_mysql