



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



CARLOS KENNETON ROCHA DE ALMEIDA

**ESTUDO E IMPLANTAÇÃO DE UMA ARQUITETURA DE
HARDWARE E SOFTWARE PARA CONTROLE E SUPERVISÃO DOS
LABORATÓRIOS DO IFAM-CMDI UTILIZANDO INTERNET DAS
COISAS (IOT)**

**MANAUS
2018**

CARLOS KENENTON ROCHA DE ALMEIDA

**ESTUDO E IMPLANTAÇÃO DE UMA ARQUITETURA DE
HARDWARE E SOFTWARE PARA CONTROLE E SUPERVISÃO DOS
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.

Orientador: Hillermann Ferreira Osmídio Lima

MANAUS

2018

CARLOS KENNETON ROCHA DE ALMEIDA

**ESTUDO E IMPLANTAÇÃO DE UMA ARQUITETURA DE
HARDWARE E SOFTWARE PARA CONTROLE E SUPERVISÃO DOS
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. MSc. HILLERMAN FERREIRA OSMÍDIO LIMA
Orientador

Prof. MSc. ADRIANO BRUNO DOS SANTOS FRUTUOSO
Professor Avaliador

Prof. MSc. 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.

Agradeço também 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 que já temos alguns.

Sou muito grato também pela orientação do professor Hillermann, mas principalmente 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 projeto visa automatizar o processo de utilização dos 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-Chaves: NodeMCU, ESP8266, Raspberry pi 3, Protocolos, Sistema web, IoT.

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 project aims to automate the process of using the 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 ESP8266 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: ESP8266, Raspberry pi 3, Protocols, Web System, IoT.

SUMÁRIO

1	INTRODUÇÃO	10
2	OBJETIVOS	12
2.1	OBJETIVO GERAL.....	12
2.2	OBJETIVOS ESPECÍFICOS	12
3	METODOLOGIA.....	13
3.1	PROCEDIMENTOS METODOLÓGICOS	13
3.2	FERRAMENTAS UTILIZADAS	13
4	JUSTIFICATIVA.....	14
5	REFERÊNCIAL TEÓRICO	15
5.1	NODEMCU	16
5.2	RASPBERRY PI 3	18
5.3	SISTEMA WEB.....	19
5.3.1	NODEJS.....	20
5.3.2	EXPRESS	21
5.3.3	MYSQL.....	21
5.3.4	ANGULARJS E BOOTSTRAP	22
5.4	MÓDULO RFID-MFRC522.....	22
6	SOFTWARE	24
6.1	DESENVOLVIMENTO DOS SOFTWARES DAS ESP8266	24
6.1.1	PROGRAMA DA ESP8266 PARA O ACESSO AOS LABORATÓRIOS	25
6.1.2	PROGRAMA DA ESP8266 PARA O CADASTRO DE USUÁRIOS	26
6.2	DESENVOLVIMENTO DO SISTEMA WEB.....	27
6.2.1	USUÁRIO COMUM	29
6.2.1	USUÁRIO ADMINISTRADOR	30
7	HARDWARE.....	36
7.1	PLACA PARA O CADASTRO DE USUÁRIOS	37
7.2	PLACA PARA OS ACESSOS AOS LABORATÓRIOS	38

8. ANÁLISE DOS RESULTADOS.....	40
9. CONSIDERAÇÕES FINAIS	41
10. REFERÊNCIAS.....	42

LISTA DE ILUSTRAÇÕES

FIGURA 1 - INTERNET DAS COISAS. FONTE: FERRAZ, 2016.....	16
FIGURA 2 - NODEMCU. FONTE: FERREIRA, 2016.....	18
FIGURA 3 - RASPBERRY PI 3, MODELO B. FONTE: CRUZ, 2017.	19
FIGURA 4 - MÓDULO MFRC522. FONTE: FILIPEFLOP, 2017.....	23
FIGURA 5 - FUNÇÃO QUE MONTA O JSON PARA O ACESSO AOS LABORATÓRIOS. FONTE: AUTORIA PRÓPRIA.	25
FIGURA 6 - FUNÇÃO QUE MONTA O JSON PARA O CADASTRO DE USUÁRIOS. FONTE: AUTORIA PRÓPRIA.....	27
FIGURA 7 - PÁGINA DE LOGIN. FONTE: AUTORIA PRÓPRIA.	29
FIGURA 8 - PÁGINA DE LISTAR RESERVAS. FONTE: AUTORIA PRÓPRIA.	30
FIGURA 9 - PÁGINA DE CADASTRO DE USUÁRIOS. FONTE: AUTORIA PRÓPRIA.	31
FIGURA 10 - LISTAR USUÁRIOS. FONTE: AUTORIA PRÓPRIA.	31
FIGURA 11 - MODAL DE ATUALIZAÇÃO DO USUÁRIO. FONTE: AUTORIA PRÓPRIA.	32
FIGURA 12 - MODAL DE EXCLUSÃO DO USUÁRIO. FONTE: AUTORIA PRÓPRIA.....	32
FIGURA 13 - CADASTRAR RESERVA. FONTE: AUTORIA PRÓPRIA.	33
FIGURA 14 - EXCLUIR RESERVA. FONTE: AUTORIA PRÓPRIA.....	34
FIGURA 15 - CADASTRAR LABORATÓRIO. FONTE: AUTORIA PRÓPRIA.....	34
FIGURA 16 - LISTAR LABORATÓRIOS. FONTE: AUTORIA PRÓPRIA.	35
FIGURA 17 - LISTA DE ACESSOS. FONTE: AUTORIA PRÓPRIA.	36
FIGURA 18 - PROTÓTIPO UTILIZADO PARA OS TESTES. FONTE: AUTORIA PRÓPRIA.....	37
FIGURA 19 - DIAGRAMA DA PLACA DE CADASTRO DE USUÁRIOS. FONTE: AUTORIA PRÓPRIA.	38
FIGURA 20 - DIAGRAMA DA PLACA DE ACESSOS AOS LABORATÓRIOS. FONTE: AUTORIA PRÓPRIA.....	39

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 nosso meio físico.

Os sistemas embarcados são utilizados em uma grande diversidade de aplicações, desde as mais simples como ligar/desligar uma lâmpada até as mais complexas como um sistema de controle de uma aeronave, segundo Carro e Wagner (2003). Há projetos de sistemas computacionais extremamente complexos, por envolver conceitos até agora pouco analisados pela computação de propósitos gerais, 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.

Segundo os autores Wolf (2001) *apud* Gervini (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.

Uma das grandes áreas da atualidade onde em 100% das aplicações são utilizados sistemas embarcados é a Internet das Coisas, do inglês, *Internet of Things* (IoT), na qual todos os dias cresce com um número exorbitante 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 (em algumas versões) entre outras especificações. Pode também ser utilizado como ponto de acesso, 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.

Outro componente de grande utilidade atualmente e que será utilizado neste projeto é a Raspberri PI 3, principalmente pela sua impressionante capacidade de processamento e realização de tarefas. Essa placa se encaixa na classe de mini PCs, por dispor de atributos de alto desempenho e de tamanho reduzido. Neste projeto a mesma será utilizada como servidor e assim fará o controle de dados entre o sistema web e a NodeMCU.

Para o sistema web foram utilizadas ferramentas de alto desempenho, onde algumas delas são desenvolvidas e utilizadas pela Google Inc. para o desenvolvimento de seus sistemas. Para este projeto foi utilizada a plataforma NodeJs, o framework Express para gerenciamento de rotas e outros recursos, o banco de dados MySQL e o framework para desenvolvimento do front-end AngularJs.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Este trabalho objetiva o melhoramento do processo de reserva e utilização dos laboratórios do Instituto Federal do Amazonas – CMDI, de modo a permitir que os usuários e os laboratoristas tenham um maior controle sobre o ambiente, transformando o processo atual que é manual, em automático, agregando as facilidades ao ser humano a implementação do conceito de internet das coisas no dia a dia do instituto.

2.2 OBJETIVOS ESPECÍFICOS

- Desenvolver um sistema web executado pela Raspberry pi 3 que permita o controle de agendamento dos laboratórios por data e hora;
- Desenvolver a comunicação entre o leitor de RFID e o dispositivo NodeMCU, assim como a comunicação com o servidor da Raspberry pi 3;
- Desenvolver, fabricar, montar e testar as placas de circuito impresso para montagem dos componentes do sistema;
- Validar a operação do sistema através de testes em ambiente compatível com os laboratórios do IFAM-CMDI.

3 METODOLOGIA

3.1 PROCEDIMENTOS METODOLÓGICOS

O trabalho será iniciado analisando as principais bibliografias que tratam do tema deste trabalho, buscando dar ênfase em tecnologias aperfeiçoadas no que se refere à Internet das Coisas e a viabilidade do sistema como um todo. Nesta fase serão verificadas as ferramentas de desenvolvimento que proporcionem um sistema web com páginas leves, visando sua execução em uma Raspberry pi 3. Como resultado desta fase, espera-se obter a melhor seleção das linguagens de programação a serem utilizadas para o desenvolvimento do sistema web, bem como a seleção do microcontrolador e o protocolo de comunicação entre os dispositivos instalados nos laboratórios e o servidor da Raspberry pi 3.

Obtendo a seleção dos componentes utilizados no projeto começará a parte do desenvolvimento dos códigos que serão executados tanto no microcontrolador como na Raspberry pi 3. Um dos maiores desafios deste projeto será o desenvolvimento do sistema web, para que o mesmo seja executado em uma Raspberry pi 3, renderizando muito bem as páginas e realizando a comunicação do microcontrolador, enviando sinais para que o mesmo atue sobre a fechadura eletrônica, registrando usuários, laboratórios utilizados, datas e horários.

A última fase do desenvolvimento são os testes, muito importante para a longa vida do sistema como um todo, de modo a prevenir falhas que venham a acontecer quando o mesmo estiver em funcionamento.

3.2 FERRAMENTAS UTILIZADAS

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

- Plataforma NodeMCU versão 1.0;
- Raspberry pi 3, modelo B;
- Leitor de RFID MFRC522 com tags tipo cartão e chaveiro;
- Fechadura eletrônica acionada por um solenoide.

Para o desenvolvimento do sistema para a Raspberry pi 3, assim como para a plataforma NodeMCU e as placas de circuito impresso, as ferramentas utilizadas foram:

- IDE do Arduino, para programação do NodeMCU;
- ATOM, para o desenvolvimento do sistema web;
- Proteus, para desenho do esquema elétrico e layout das placas de circuito impresso.

4 JUSTIFICATIVA

O processo atual de reserva e utilização de um laboratório consiste normalmente de uma solicitação antecipada do espaço através de um sistema web, explicitando a data e horário de utilização, nome de quem vai utilizar e nome do laboratório. Assim, na data estipulada o solicitante recebe a chave para acesso ao laboratório. Este sistema, apesar de funcional, pode apresentar falhas, principalmente no que diz respeito à devolução das chaves para a coordenação de laboratórios. Segundo relatos, acontece eventualmente de um usuário fazer o uso do laboratório e não devolver as chaves, o que pode prejudicar o uso por outro usuário e até impedir o uso do laboratório.

Com o sistema proposto não haverá a necessidade de uso de chaves, uma vez que o acionamento será através da tecnologia RFID e cada usuário terá sua TAG de identificação. Uma outra facilidade que o sistema proporcionará é o registro de todos os acessos aos laboratórios no banco de dados, informando o usuário, a data e hora do acesso.

5 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, proporcionando melhores especificações, como, alta taxa de armazenamento em memórias, velocidade de processamento, tipos de comunicação, número maior de porta I/O e conversores ADCs, etc.

Segundo Vargas (2010), 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.

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, 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 esta para qualquer dispositivo.

Para Oliveira (2017), 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. Principalmente porque o mercado exige que os produtos sejam cada vez mais capacitados a desenvolver tarefas mais difíceis que as versões anteriores, tudo isso com os produtos tendo ciclo de vida baixo, e uma dessas saídas é tornar os sistemas cada vez mais inteligentes a ponto de interagir ainda mais com os usuários.



Figura 1 - Internet das coisas. Fonte: Ferraz, 2016.

5.1 NODEMCU

O dispositivo microcontrolado NodeMCU é produzido em larga escala na china de onde se origina, ele 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 o conjunto de instruções de 32 bits, que é adicionado ao mesmo o núcleo microprocessado Tensilica L106, a frequência 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.

Um dos pontos de grande importância no mundo atualmente da tecnologia é o consumo de energia dos dispositivos, e a ESP8266 trata de modo excelente esta questão tendo em vista sua capacidade de realizar um auto nível de processamento, sua tensão nominal de funcionamento é de 3,3 volts, a corrente no dispositivo varia de acordo com o modo de utilização selecionado pelo usuário na programação, assim sendo 20 μ A no modo *sleep*, cerca de 50 mA quando conectado a um ponto

de acesso em modo de recepção e no modo transmissão com potencia máxima de 17 dBm o consumo pode chegar até 170 mA.

Outro ponto de grande destaque são as interfaces de entrada e saída, juntamente com as possibilidades de comunicação do chip, o NodeMCU possui 11 interfaces GPIO (*General Purpose Input/Output*), 4 saídas PWM (*Pulse-Width Modulation*), 1 entrada analógica de 10 bits, o mesmo disponibiliza as interfaces seriais síncronas, SPI e I2C; e assíncronas USART, assim podendo utilizar uma interface RS-232, porém o que torna o microcontrolador diferenciado é a interface WiFi, que pode ser utilizada no modo de ponto de acesso ou no modo estação.

O microcontrolador ESP8266 é comumente utilizado em internet das coisas, por ser um componente excepcional para interconexão entre equipamentos distintos, devido principalmente às funcionalidades que este apresenta no que remete a conexão Wi-Fi, sem contar a portabilidade do mesmo, sendo extremamente pequeno e barato com já abordado.

Segundo os autores, Kolban (2015) *apud* Fernandes (2017), o módulo ESP8266 envia dados de forma serial pela rede Wi-Fi, em que é 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.

De acordo Ferreira (2016), o dispositivo WiFi (designado ESP8266) é um componente com TCP/IP integrado que dá a qualquer microcontrolador acesso a uma rede Wi-fi, após ser programado para tal. O ESP8266 é capaz de hospedar ou mesmo ir buscar todas as funções da rede WiFi. Este módulo vem pré-programado com um conjunto firmware de comandos AT. Com este é possível fazer qualquer coisa como se fosse usado um WiFi Shield's. O chip pode ser programado utilizando a IDE do arduino, Linguagem Lua desenvolvida no Brasil por professores 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 baseados neste, cada um buscando ainda mais especificidades em aplicações de internet das coisas, um dos mais populares é o módulo denominado NodeMCU

especificado na Figura 2, que já agrega em si a plataforma de desenvolvimento em linguagem Lua.



Figura 2 - NodeMCU. Fonte: Ferreira, 2016.

5.2 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 ser utilizada 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 o Raspberry pi, inicialmente nos modelos A+ e B+ e chegando até a versão 3 modelo B, e ultimamente foi lançada a Raspberry Pi Zero, sendo um modelo mais simples e de maior baixo custo ainda, chegando a custar 5 dólares no site oficial da fundação.

O 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 e 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.

Segundo Cruz (2017), a Raspberry PI 3 é um computador pessoal de baixo custo. Em 2006 no Reino Unido, país de origem, esta placa foi lançada a \$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.

Uma das grandes vantagens dessa placa, segundo Oliveira (2017), é a disponibilidade de software para sistema Linux que podem ser utilizados dentro da 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.



Figura 3 - Raspberry Pi 3, modelo B. Fonte: Cruz, 2017.

5.3 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 independentemente do sistema operacional já que o navegador é o responsável pela interpretação das informações e este já é padronizado.

Segunda Aguiar (2017), 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. Isso aumenta a capacidade do usuário de interagir mais com as informações.

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, inclui-se celulares, computadores pessoais, tablets e etc.

5.3.1 NODEJS

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

De acordo com a NodeBR (2016), essa ferramenta 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.

Neste projeto o node é de extrema importância principalmente na gerência de atividades o que proporciona agilidade no processamento de informações, organizando de forma primordial as dependências do sistema, de modo a manipulá-las de modo singelo. Na utilização do node é explícito o conceito de programação orientada a evento, no qual quando um evento é detectado, por exemplo, um botão é apertado e assim o node 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 criação de um front-end, responsável por obter as entradas fornecidas pelos usuários e no decorrer do processo esses dados são entregues ao

back-end, no qual sua função é tomar as decisões já pré-estabelecidas no momento da criação do sistema.

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

5.3.3 MYSQL

O MySQL é uma plataforma de gerenciamento de banco de dados de alta performance, sendo relacional. Utiliza a linguagem SQL (*Structured Query Language*), Atualmente pertence a gigante empresa do ramo da computação, Oracle. Possui bastante material disponível em livros e internet no que desrespeito a desenvolvimento de sistemas. Em muitas situações por ser um banco relacional, traz grandes facilidades no modelamento de sistemas com alta relação entre as tabelas no banco, realizando operações de modo muito mais simples e proveitoso.

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 e diversas 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.

5.3.4 ANGULARJS E BOOTSTRAP

A ideia por trás do angular é de um framework MV (Model - View), ou seja, voltado para a interface com o usuário, assim 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 (*HyperText Markup Language*) em uma linguagem para o desenvolvimento de interfaces web dinâmicas. Segundo Waltenberg (2016), o angular foi desenvolvido pela Google, e devido ao sucesso, internamente a empresa ele é utilizado em mais de 1600 projetos. Isso se deve principalmente as possibilidades de controle e gerência do front-end, juntamente com a padronização dos frames nos mais diversos tipos de navegadores.

Neste projeto o angular controla toda a parte de animação dos botões, frames, menus e 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 angular para que a tela não ficasse sendo recarregada por completa a todo o momento, e sim fosse 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 do 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. Assim trazendo agilidade no desenvolvimento e melhorando o visual das páginas.

5.4 MÓDULO RFID-MFRC522

Uma das primeiras tecnologias ligadas ao conceito de internet das coisas surgiu em 1940, denominada RFID (*Radio Frequency Identification*), 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 e etc.

Uma das áreas de aplicação desta tecnologia é o controle de acesso, empregado em residências, estabelecimentos comerciais e 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 o mínimo consumo de energia que o mesmo proporciona. Este funciona a uma frequência de 13,56 MHz, possui tensão de alimentação de 3,3 volts, a corrente de trabalho varia entre 13-26 mA, sendo que no modo *sleep* o mesmo consome apenas 80 μ A, utiliza comunicação spi, e o mesmo normalmente vem de fábrica com duas tags, uma em forma de cartão e a outra em chaveiro, como mostra a Figura 4.

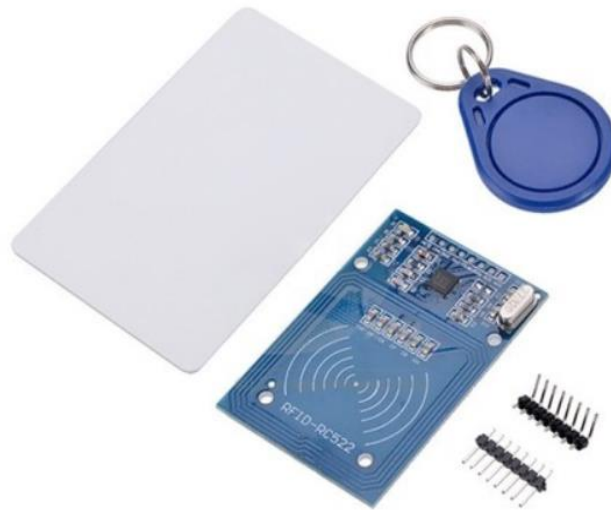


Figura 4 - Módulo MFRC522. Fonte: FilipeFlop, 2017.

6 SOFTWARE

Os softwares são sempre parte essencial na maioria dos projetos de engenharia de controle e automação, ainda mais quando se trata de internet das coisas, principalmente por decretarem 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 função.

No desenvolvimento dos softwares foram utilizadas diversas linguagens, divididas entre linguagens compiladas e interpretadas e em alguns casos as duas funções ao mesmo tempo como é o caso do JavaScript, uma das principais linguagens do trabalho, utilizada para o sistema o web. Pode-se citar também a linguagem Arduino que é totalmente baseada no C++, facilitando a forma de programar o ESP8266 já que esta é um sucesso no mundo todo por oferecer diversos recursos para os programadores de microcontroladores.

6.1. DESENVOLVIMENTO DOS SOFTWARES DAS ESP8266

Após o estudo das ferramentas que poderiam ser utilizadas no projeto, deu-se início a programação da ESP8266, assim foi acertado que necessitaria de dois códigos distintos sendo um para os laboratórios, com o objetivo de ler as tags e acionar o solenoide da fechadura, e o outro para ler as tags e cadastrar um novo usuário.

Primeiramente o objetivo foi fazer com que a ESP8266 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, como por exemplo, foi adotado que os caracteres lidos das tags seriam transformados em hexadecimais e ambos ficariam no formato maiúsculo, para isso foram utilizadas funções das bibliotecas MFRC522.h e SPI.h já que as informações entre a ESP8266 e o módulo de RFID eram trocadas através de comunicação SPI (*Serial Peripheral Interface*), e para completar a padronização os códigos de cada tag foram manipulados com funções específicas de *strings*.

6.1.1. PROGRAMA DA ESP8266 PARA O ACESSO AOS LABORATÓRIOS

O código que fica nas ESP8266 para gerenciamento dos laboratórios possui a seguinte estrutura, declaração das bibliotecas e variáveis, seleção dos pinos para o devido microcontrolador que serão utilizados na comunicação entre os dispositivos, configuração de rede (login e senha da rede, endereço ip do Host), um dos pontos de bastante significância na comunicação entre os dispositivos é a utilização de um modelo para transmissão e armazenamento de informações no formato de texto denominado JSON (*JavaScript Object Notation*), Segundo Corrêa ferramenta 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 (*eXtensible Markup Language*), neste código o modelo JSON foi utilizado para enviar para o servidor os dados “id_laboratorio” que um código único de cada laboratório e a “tag” que guarda o número do usuário que tentou adentrar no ambiente, como expressa a Figura 5.

```

/*
 * 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;
}

```

Figura 5 - Função que monta o JSON para o acesso aos laboratórios. Fonte: Autoria própria.

Dando sequência no código que fica nas ESPs dos laboratórios, em seguida são criadas as funções que iniciam e conectam os microcontroladores à rede, logo após vem a função que monta o objeto JSON que será enviado para o servidor contendo as informações necessárias para identificação do laboratório e do usuário, em seguida outra função foi criada para envio dos dados contidos no JSON, utilizando a biblioteca “ArduinoJson.h”, esta função realiza uma requisição do tipo POST para o servidor utilizando o protocolo HTTP (*Hypertext Transfer Protocol*) e a biblioteca “ESP8266HTTPClient.h”, ou seja, são enviados dados encapsulados junto ao corpo da requisição para um determinado endereço identificado por uma URL.

Uma das funções de maior utilidade no processo é a que permite a entrada de usuários mesmo após uma queda na rede, agregando ainda mais persistência no controle de acesso e tornando o mesmo ainda mais robusto. Esta função funciona da seguinte forma, após o cadastro do usuário no sistema, se faz necessário salvar a tag do mesmo na própria ESP8266 através da EEPROM, memória não volátil capaz de guardar dados mesmo após o desligamento do dispositivo. Assim quando não houver conexão o sistema reconhece e o controle de acesso funciona localmente.

Outra função bem interessante é a de configuração dos dados de rede através do terminal serial do arduino, sem a necessidade de ficar alterando o código e regravando a ESP8266, podendo ocorrer erros, como alterações indevidas no código e outros problemas. Proporcionando facilidades para os usuários inseridos no processo.

E por último vêm as duas funções padrões da linguagem arduino, a primeira é a “setup” onde estão às configurações de como vai funcionar o algoritmo (pinagem do microcontrolador, velocidade da porta serial, inicialização de funções e etc.) sendo que esta só roda uma vez durante a execução do programa. E a outra é a função “loop” na qual a mesma executa diversas vezes atendendo aos comandos pré-estabelecidos na programação.

6.1.2. PROGRAMA DA ESP8266 PARA O CADASTRO DE USUÁRIOS

O presente algoritmo possui basicamente a estrutura apresentada no tópico 6.1.1, diferenciando apenas em três partes do código, a primeira é que dentro da função que monta o JSON, só existe o atributo “tag”, como mostrado na Figura 6, já que para cadastrar o usuário não se necessita saber informações de nenhum laboratório, as ESPs gravadas com esse código só ficarão a disposição do administrador do sistema.

```

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

```

Figura 6 - Função que monta o JSON para o cadastro de usuários. Fonte: Autoria própria.

A segunda diferença é que já que esse código só tem a função de enviar a tag para o cadastro de usuários, não se necessita de mais um I/O para o acionamento da fechadura, apenas para a sinalização através dos leds vermelho e verde.

E a outra diferença é com relação á função de persistência que grava as tags dos usuários cadastrados na EEPROM, que não se faz necessário já que não tem acesso, esta ESP8266 só é utilizada para cadastrar um novo usuário.

6.2. DESENVOLVIMENTO DO SISTEMA WEB

O sistema web é uma das partes mais importantes e trabalhosas do projeto com um todo, pois tem de haver todo um cuidado para que o usuário possa interagir de modo simples, objetivo e animado, porém tudo isso tem que ser agregado ao que é chamado de back-end, ou seja, a parte do sistema que executar acima do conhecimento do usuário, as funções por trás de cada botão pressionado, de cada tela exibida. Logo essa atividade não se torna tarefa fácil.

Um dos grandes cuidados tomados ao iniciar o sistema web foi a escolha das ferramentas de trabalho, principalmente porque se deseja um sistema robusto, com um certo nível de complexidade e que execute facilmente em equipamentos com

processamento limitados, como é o caso da raspberry pi 3, comparada a um servidor comumente utilizado para esse tipo de sistema.

Após testes as ferramentas adotadas para o desenvolvimento foram o NodeJS, a linguagem JavaScript, o banco de dados sendo o MySQL, para gerencia de atividades por parte do servidor foi utilizado o Express e para o front-end foi utilizado ferramenta AngularJS 1, o *template* base para o front-end foi o admin-lte, devido as suas facilidades, com ferramentas já prontas e bastante úteis para o desenvolvimento da parte visual de sistemas.

No desenvolvimento do sistema foi previsto dois tipos de usuários, sendo que o primeiro são os administradores, com acesso liberado para operações em todo o sistema, como exemplo, criar, selecionar, atualizar e excluir, usuários, laboratórios reservas e acessos. Sendo que no menu de acessos não se pode cria, excluir e nem deletar, apenas visualizar se o usuário com a reserva adentrou o recinto reservado.

O outro tipo de usuário é o comum que só é disponível a aba de listar reservas, assim este pode visualizar se há reservas disponíveis para o laboratório, data e hora que o mesmo deseja e assim consultar o administrador para realizar sua reserva caso necessário.

Inicialmente o sistema conta com uma página de *login* conforme a Figura 7, ao conseguir entrar no sistema, o usuário já será identificado com relação as suas permissões, no qual o algoritmo já disponibiliza as páginas certas para cada modelo de usuário.

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 verificado 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*.

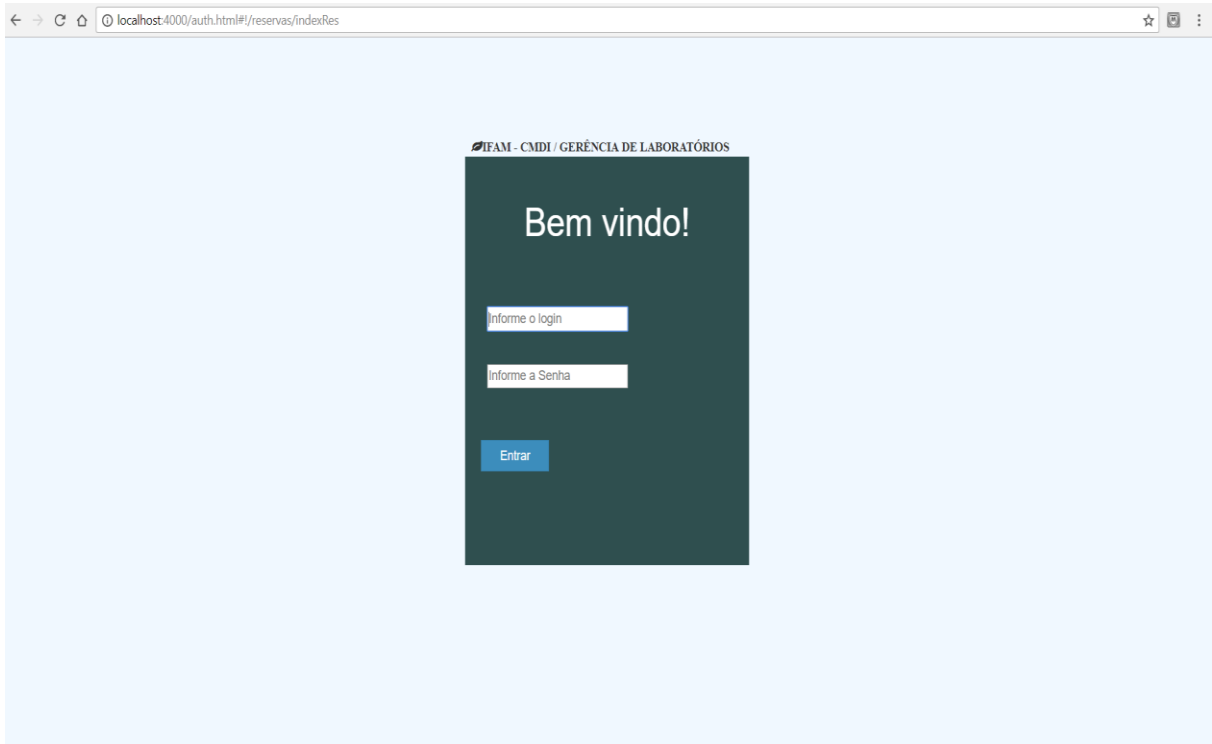


Figura 7 - Página de login. Fonte: Autoria Própria.

6.2.1. USUÁRIO COMUM

No momento do cadastro do usuário, o mesmo é inserido no sistema de modo a possuir um determinado nível de permissão como já mostrado anteriormente, no caso do usuário comum, o mesmo só tem acesso a uma página após ter entrado no sistema. Que é a página de “Listar Reservas”, assim o usuário comum pode realizar uma busca por laboratório e data que deseja visualizar, proporcionando ao mesmo saber os horários livres e assim solicitar ao administrador uma reserva. A Figura 8 mostra a página para listar as reservas.

Vale ressaltar que o usuário comum não pode fazer uma reserva, isso foi definido como um dos requisitos de projeto, adotando que cada reserva devesse ser cadastrada por um administrador.

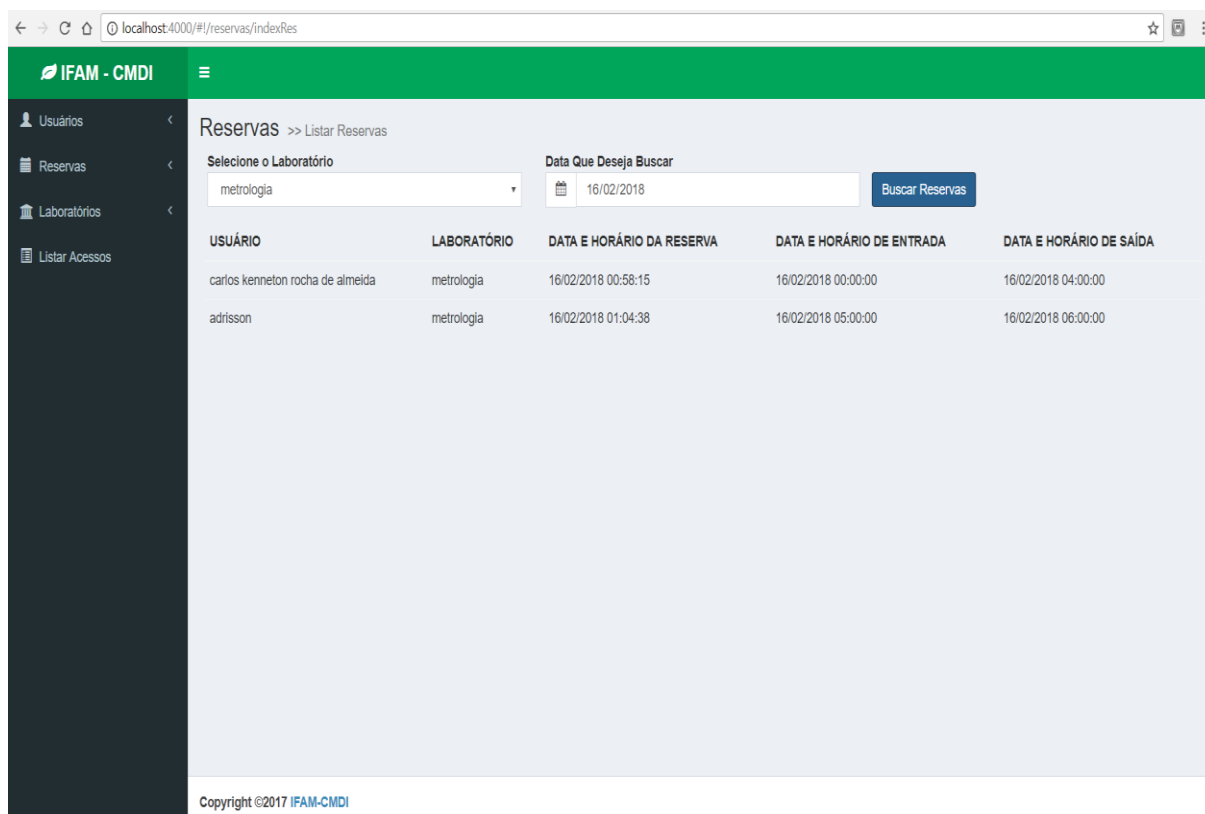


Figura 8 - Página de Listar Reservas. Fonte: Autoria Própria.

6.2.2. 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 o usuário, a página carregada aparece com os campos dos formulários todos vazios, porém para cadastrar um novo usuário é necessário aproximar a tag do leitor, após esta ação e o led verde da placa acender é só atualizar a página e o sistema já busca automaticamente a tag lida, o restante é só preencher os campos e clicar em cadastrar usuário. A Figura 9 ilustra a página de cadastro após a leitura da tag.

O outro subitem é o listar, que tem a função de buscar os usuários cadastrados ordenados de forma ascendente pelo nome, explicitando algumas informações sobre cada usuário e para cada usuário essa página disponibiliza a função de atualizar informações ou deletar usuário. A Figura 10 ilustra a página responsável por listar os usuários.

IFAM - CMDI

Usuários >> Cadastrar

Cadastrar um novo Usuário

Tag
3AE697E5

Nome
Informe o nome do usuário

Função Institucional
▼

Login
Informe o login do usuário

Senha
Informe a senha do usuário

☐ Administrador

[Cadastrar Usuário](#)

Copyright ©2017 IFAM-CMDI

Figura 9 - Página de cadastro de usuários. Fonte: Autoria Própria.

IFAM - CMDI

Usuários >> Listar

TAG	NOME	FUNÇÃO INSTITUCIONAL	TIPO DE USUÁRIO	AÇÕES
BD2798E5	adriisson	ALUNO	ADMINISTRADOR	✎ ✖
3AE697E5	carlos kenneton rocha de almeida	PROFESSOR	ADMINISTRADOR	✎ ✖

Copyright ©2017 IFAM-CMDI

Figura 10 - Listar Usuários. Fonte: Autoria Própria.

Como já citado na página de listar, são mostradas as funcionalidades de atualizar e deletar usuários, para melhorar esses dois ambientes foram criados *modals* que exibem as informações sobre o usuário que será atualizado ou deletado. As Figuras 11 e 12 respectivamente, ilustram os *modals* de atualização e exclusão.

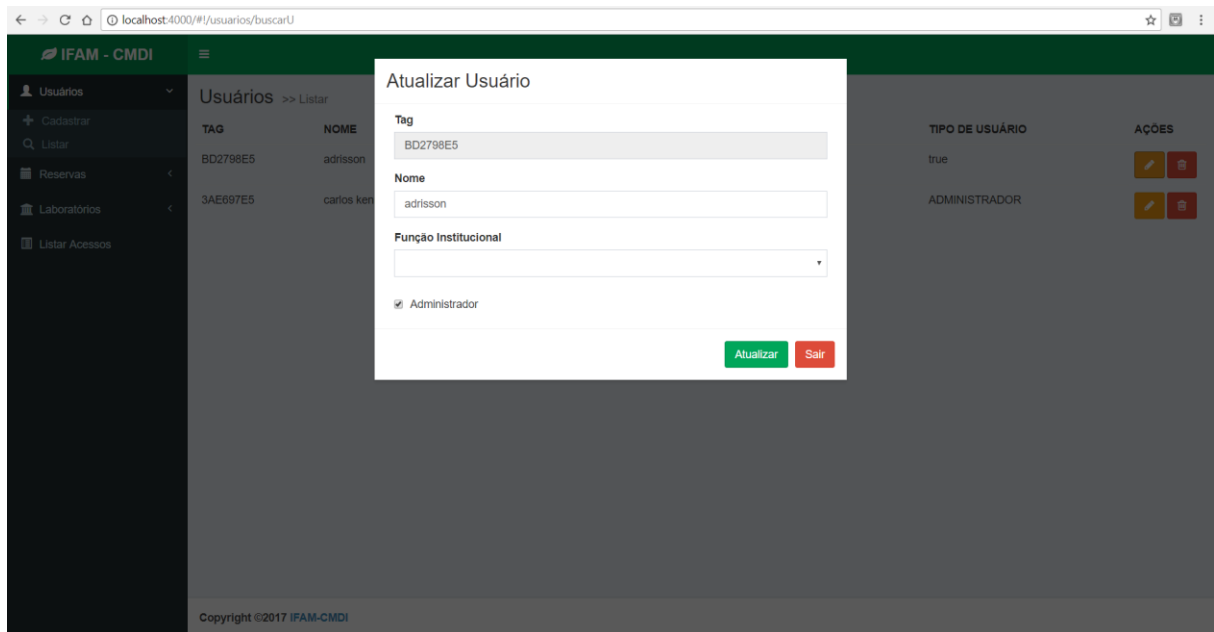


Figura 11 - Modal de atualização do usuário. Fonte: Autoria Própria.

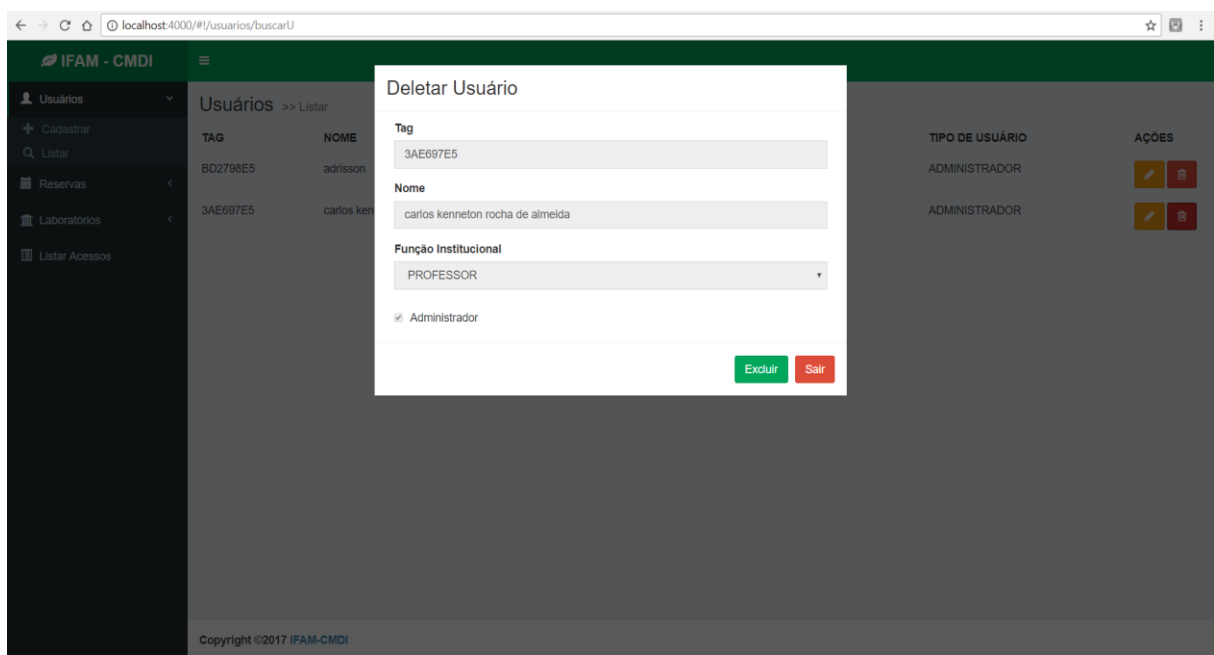


Figura 12 - Modal de exclusão do usuário. Fonte: Autoria Própria.

Na aba de reservas, há três subitens, o primeiro é o “listar reservas” já demonstrado no item 6.2.1, o segundo é o “cadastrar reservas”, esta funcionalidade é bem simples, principalmente pelo ambiente ser bem dinâmico, por exemplo, nos campos “nome do usuário” e “nome do laboratório” é exibida uma lista de todos usuários e laboratórios já cadastrados em ordem ascendente, para que não haja divergências nos nomes dos componentes e assim tornar a vida dos usuários mais fácil ainda. A Figura 13 demonstra o ambiente de cadastro das reservas.

Figura 13 - Cadastrar Reserva. Fonte: Autoria Própria.

O terceiro subitem é o “excluir reserva”, essa página é bem parecida com o “listar reservas”, porém possui uma funcionalidade extra que é a opção de excluir uma determinada reserva, um dos questionamentos que podem surgir é porque não existe o atualizar reserva, porém a resposta é que diferentemente das abas de usuários e laboratórios que possuem ligação direta com os códigos das ESP8266 e precisam que seus “id” sejam preservados para que não se torne necessário ficar regravando o microcontrolador, sendo que em alguns casos só precise de uma atualização e o “id” continue mantido, como no caso de um laboratório que se mudou para outro prédio e só precisa de uma atualização, já as reservas são facilmente criadas e deletadas, e existirão diversas reservas diferentemente de

laboratórios e usuários, vale ressaltar que ao clicar no botão de excluir surge um modal de confirmação de exclusão. A Figura 14 ilustra a função de deletar reserva.

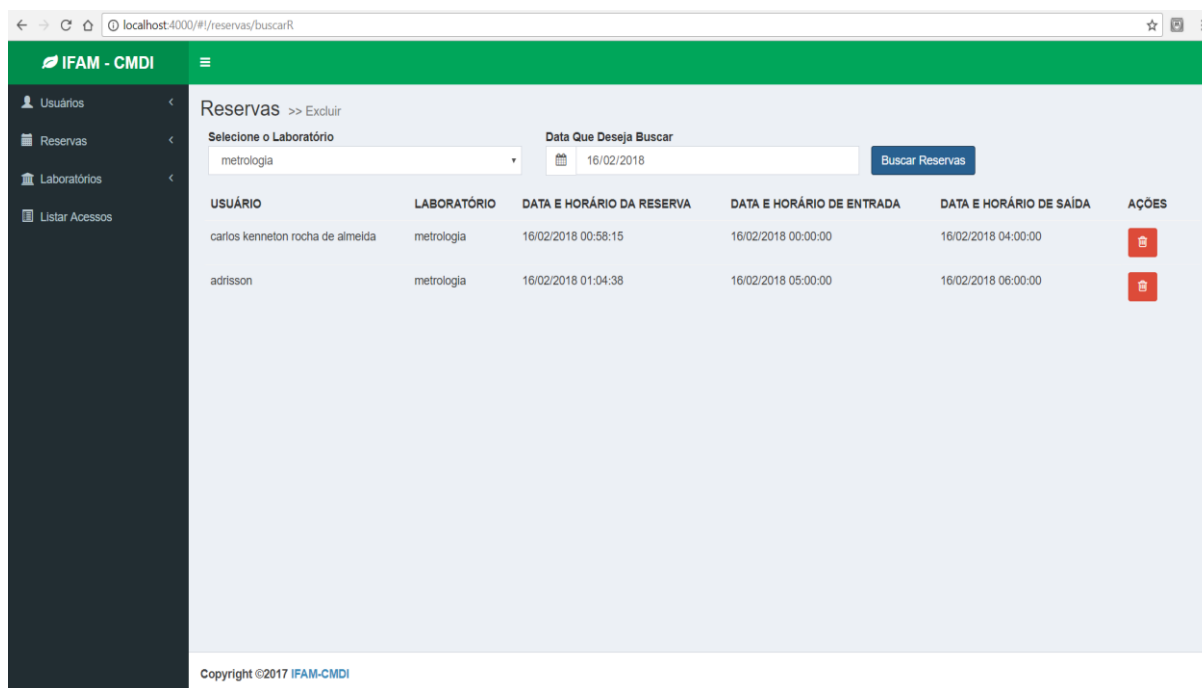


Figura 14 - Excluir Reserva. Fonte: Autoria Própria

A próxima aba que é a de “laboratórios” que possui dois subitens, o primeiro é o “cadastrar”, que 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 15 ilustra a página de cadastro.

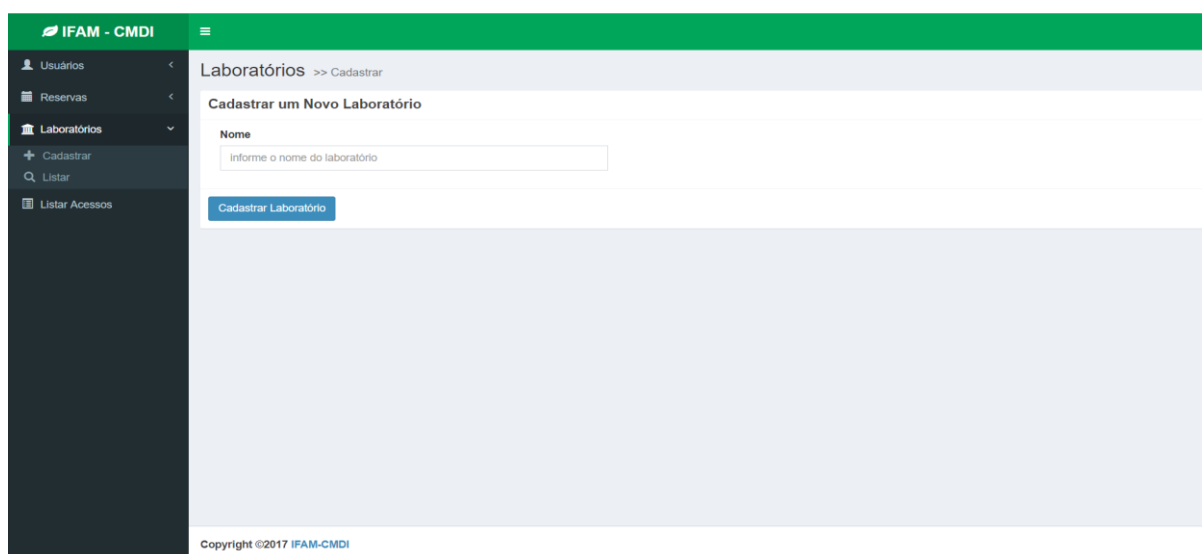
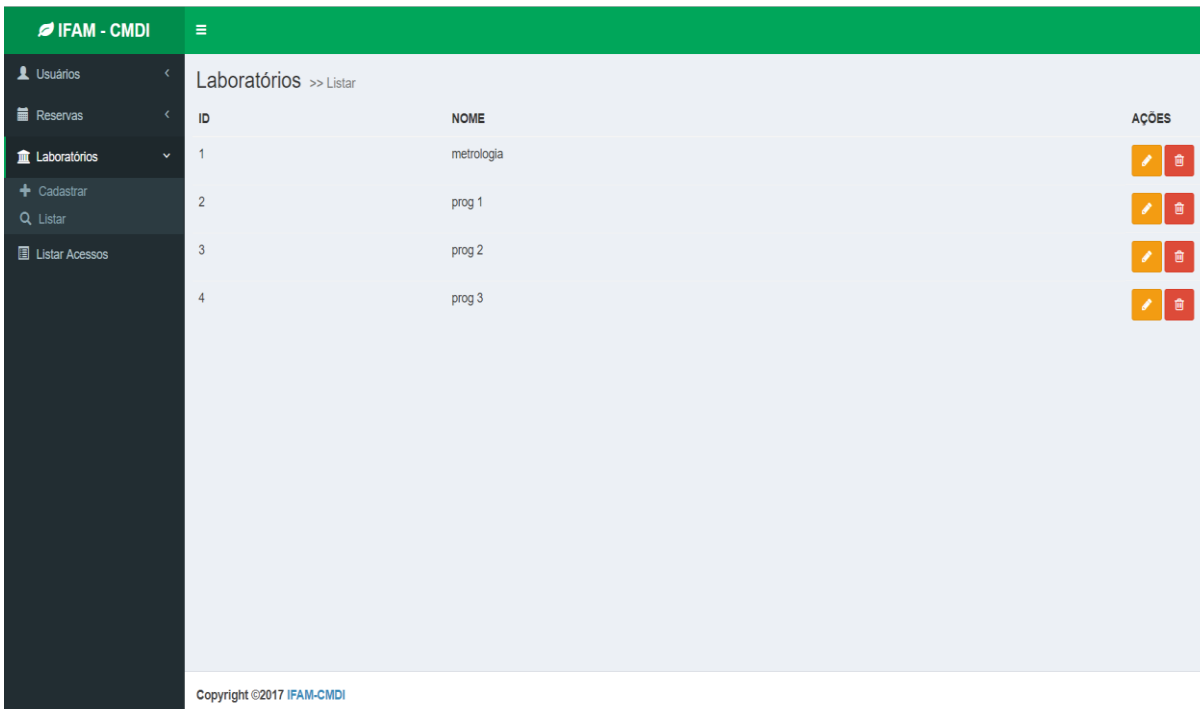










Figura 15 - Cadastrar Laboratório. Fonte: Autoria Própria.

O outro subitem da aba de laboratórios é o “listar”, que é bem parecido com o modelo utilizado para listar os usuários, apenas com o parâmetro nome dos laboratórios e também mostrando o “id” já que este se faz necessário saber para gravar na ESP8266 de cada laboratório e análogo ao modelo de usuários oferecendo as possibilidades de atualizar ou deletar através dos *modals*. A Figura 16 mostra a lista de laboratórios.

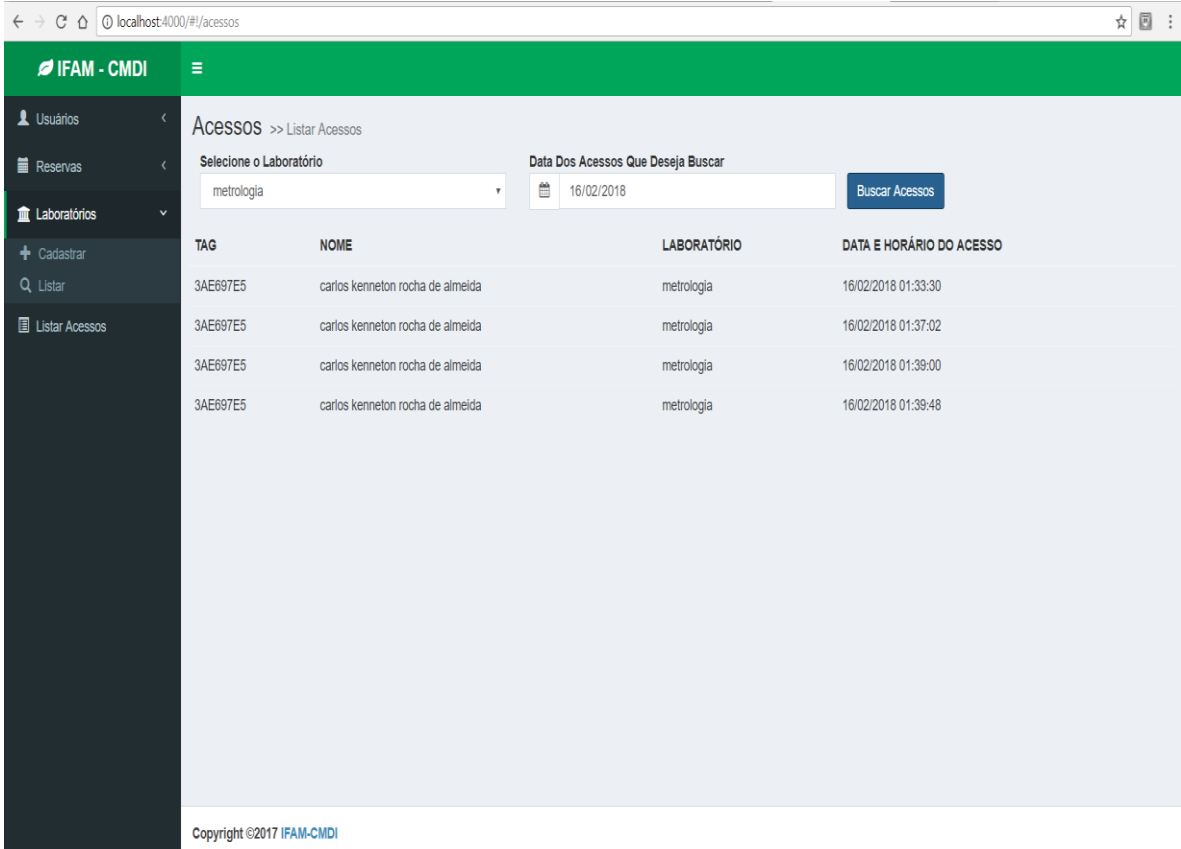


ID	NOME	AÇÕES
1	metrologia	 
2	prog 1	 
3	prog 2	 
4	prog 3	 

Copyright ©2017 IFAM-CMDI

Figura 16 - Listar Laboratórios. Fonte: Autoria Própria.

E a última aba, que possui apenas uma funcionalidade que é mostrar os acessos de um laboratório em um determinado dia, é a “listar acessos”, os acessos são registrados automaticamente pelo sistema, desde que o usuário que tentou entrar no laboratório esteja cumprindo os requisitos básicos para registrar um acesso no banco de dados, sendo esses, a tag que foi aproximada no leitor já esteja cadastrada em nome de um usuário, o laboratório esteja cadastrado no sistema e exista uma reserva para a data e horário que o usuário está tentando adentrar o recinto, neste caso a led verde da placa irá sinalizar que o acesso foi concluído e a fechadura será aberta, caso contrário o led vermelho acenderá indicando que um dos requisitos apresentados anteriormente não foi validado. A Figura 17 mostra a lista de acessos, para um determinado laboratório e uma data.



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

Figura 17 - Lista de Acessos. Fonte: Autoria Própria.

7. HARDWARE

O hardware é comumente uma parte fundamental em um projeto de engenharia, tendo em vista que o mesmo estará no meio físico, interagindo diretamente com os usuários, basicamente recebendo entradas e gerando saídas, sendo gerenciado pelos softwares que já dispõe de funções preestabelecidas no ato da programação.

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 às placas fazem leituras das tags RFID, foi desenvolvido um ambiente feito de madeira, acoplado a uma fechadura, acionada por solenoide para a apresentação do protótipo. A Figura 18 ilustra o protótipo utilizado para testes.



Figura 18 - Protótipo utilizado para os testes. Fonte: Autoria Própria.

7.1 PLACA PARA O CADASTRO DE USUÁRIOS

Esta placa tem por objetivo fazer as leituras da tags, e realizar o envio para o servidor do código referente à tag lida, e logo após retornar nos leds verde ou vermelho, se a operação ocorreu com sucesso ou não. Os componentes dessa placa e seus respectivos *labels* estão apresentados no diagrama eletrônico da Figura 19, estes são: ESP8226 NodeMCU identificada 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 diretamente no modelo NodeMCU, que possui uma entrada microUSB assim, o mesmo redistribui a energia para o leitor RFID e para os leds, facilitando a distribuição de energia na placa por inteira.

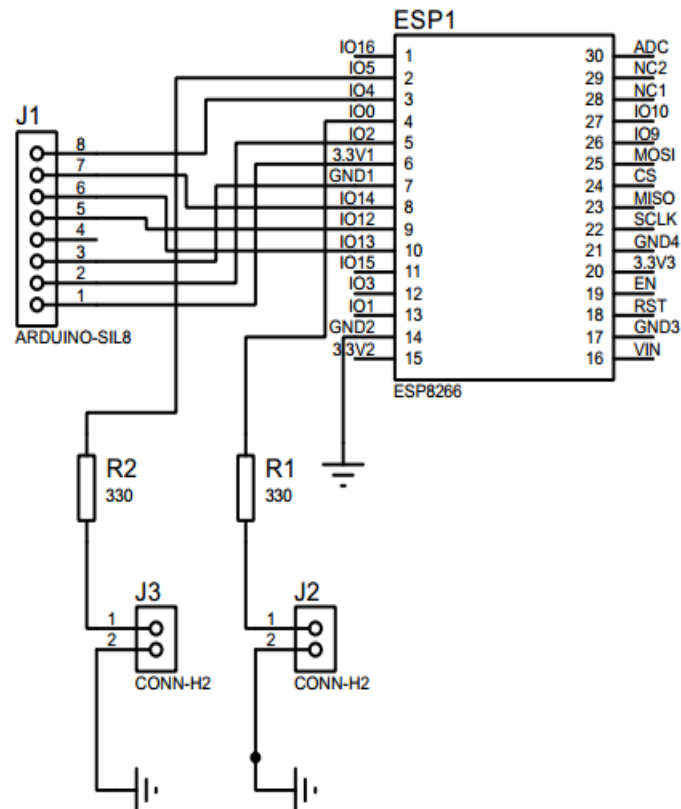


Figura 19 - Diagrama da placa de cadastro de usuários. Fonte: Autoria Própria.

7.2 PLACA PARA OS ACESSOS AOS LABORATÓRIOS

Esta placa possui algumas funcionalidades a mais que a anterior, no qual é necessário um circuito extra para gerenciamento da fechadura, um *push button* para acionar a funcionalidade de gravar as tags localmente para o caso da rede ficar indisponível, e ainda será acrescentado externamente a placa um outro botão que ficará dentro dos laboratórios para o caso em que o usuário fique trancado por dentro, assim o mesmo pressiona o botão e a fechadura se abre.

Nessa placa são realizadas as seguintes funções, a mesma ler as tags, envia os dados para o servidor e dependendo da resposta ela abre ou não a fechadura, indicando também através de dois leds, um vermelho e um verde. Se caso houver sucesso na resposta do servidor a placa abre a fechadura ou se a rede estiver indisponível e a tag já estiver sido cadastrada localmente, o usuário também terá acesso ao laboratório.

Os principais componentes dessa placa são mostrados na Figura 20, juntamente com seus respectivos *labels*, estes são: ESP8266 NodeMCU identificada como ESP 1, três resistores R1, R2 e R3, os capacitores C1 e C2, dois leds J5 e J3, o leitor RFID representado pela barra de pinos J1, uma *chave push button*, dois bornes KRE identificados como J7 e J2 ambos para as saídas e alimentações da fechadura, relé de acionamento e botão que fica no interior dos laboratórios, regulador de tensão 7805 identificado como U1, um diodo D1 para proteção contra inversão de polaridade da fonte e um conector jack P2 para a fonte de alimentação de 12 V, vale ressaltar que neste caso serão utilizadas duas fontes, uma somente para a ESP8266 e outra no conector jack P2.

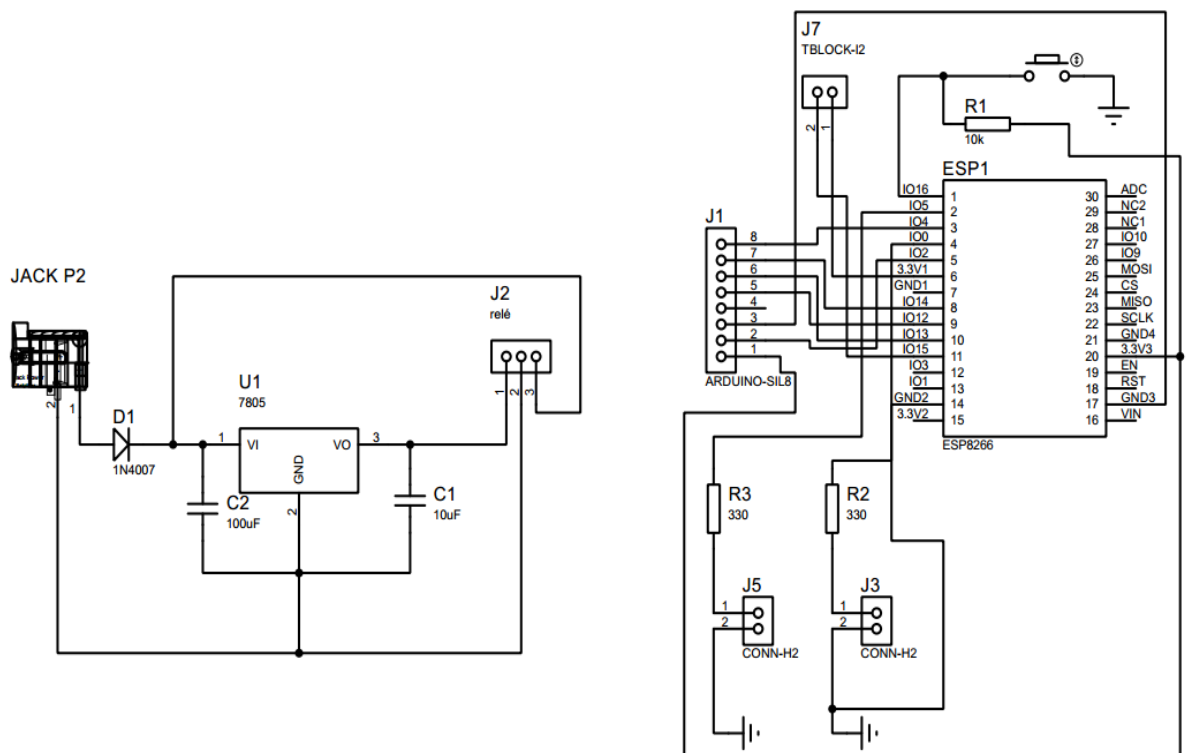


Figura 20 - Diagrama da placa de acessos aos laboratórios. Fonte: Autoria Própria.

8. ANÁLISE DOS RESULTADOS

Os resultados se mostraram bastante satisfatórios, principalmente no que desrespeita a parte funcional do sistema, que são as reservas e os acessos aos laboratórios através da leitura de tags RFID, mas vale ressaltar que as diversas funcionalidades secundárias, com o intuito de melhorar as facilidades aos usuários se mostraram bastantes proveitosas no decorrer dos testes, principalmente na parte visual com telas bem autoexplicativas e inteligentes.

Um dos pontos de suma importância nesse projeto foi testar diversas ferramentas e constatar quais eram mais ideais para serem utilizadas. Um dos primeiros testes e mais cruciais para o projeto foi logo no início quando houve o desenvolvimento de um sistema básico utilizando a linguagem PHP, banco de dados MySQL e o HTML em conjunto com JQuery, com isso foi possível observar, que o sistema apresentava demora no tempo de resposta, inclusive foram feitos testes com o mesmo código da ESP8266 e com um ping da rede igual, para esse sistema e para uma aplicação básica feita em NodeJS, Express e o MySQL, ambas com as mesmas funcionalidades, o teste tinha a seguinte metodologia, inicialmente era enviado uma requisição POST da ESP8266 para o HOST, e o mesmo realizava as funções de validação e respondia novamente para esta, com isso foi concluído que para essa aplicação, o modelo em NodeJS seria mais viável, com um tempo aproximado de quase 2 segundos menor na resposta, se aproximando de 6 segundos para realizar todas as funções necessárias, sem contar que o mesmo não apresentou nenhum travamento durante os testes, e assim foi definido que o sistema seria gerenciado e desenvolvido pelas ferramentas apresentadas anteriormente.

Outro ponto de discussão foi qual banco de dados utilizar, o mongoDB ou MySQL, devido que ambos apresentaram resultado bastante parecidos para esta aplicação, adotou-se o MySQL pelo conhecimento prévio do mesmo e a facilidade na quantidade de materiais disponíveis.

Não se pode deixar de falar também da persistência implantada no sistema no caso da rede ficar indisponível, que funcionou perfeitamente, mesmo interagindo com o usuário através de leds, se mostrou uma ferramenta de alta importância para o projeto.

9. CONSIDERAÇÕES FINAIS

Tendo em vista a realização do projeto e toda arquitetura fundamentada nesse sistema de gerenciamento de laboratórios do IFAM, pode-se considerar que o resultado obtido trará bastante benefícios aos usuários que utilizam os ambientes tratados no âmbito deste trabalho, proporcionando facilidades nas mais diversas partes do sistema.

O modelo dessa arquitetura pode ser utilizado para desenvolver ainda 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 da instituição por mais pesquisas do tipo, que busquem incentivar ainda mais o curso de engenharia de controle e automação, proporcionando que outros alunos e professores desenvolvam ainda mais sua capacidade de trabalho.

Com relação a melhoramentos neste sistema, uma das possibilidades seria a realização de um backup local das reservas diariamente dentro da ESP8266, podendo ser utilizada até uma memória externa em conjunto com um circuito RTC, para salvar esses dados e implantar um relógio local, a partir disso quando o sistema detectasse que não há conexão com a rede, a NodeMCU passasse a fazer as consultas localmente verificando o horário de cada reserva e depois quando voltasse a conexão reenviar os acessos para serem salvos no banco de dados, pois atualmente, quando ocorre esse tipo de situação, a mesma verifica apenas se a tag aproximada no leitor está cadastrada localmente, caso isso seja verdadeiro a fechadura se abre, caso contrário, continua da mesma forma, fechada.

10. 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.

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.

GERVINI, Alexandre I. et al. **Avaliação de Desempenho, Área e Potência de Mecanismos de Comunicação em Sistemas Embarcados**. SEMISH'03—XXX Seminário Integrado de Software e Hardware, 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.

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

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 I

Repositório do github contendo todos os códigos das ESP8266 e o do sistema web:

- https://github.com/carloskalil10/laboratorios_mysql