



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

Desenvolvimento de fechadura inteligente utilizando tecnologia RFID

Olimpio Pimenta

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:

Nome Completo do Orientador

COORIENTAÇÃO:

Nome Completo do Coorientador

Março, 2019

João Monlevade–MG

Olimpio Pimenta

**Desenvolvimento de fechadura inteligente
utilizando tecnologia RFID**

Orientador: Nome Completo do Orientador

Coorientador: Nome Completo do Coorientador

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Março de 2019

A Ficha Catalográfica é elaborada exclusivamente pela Biblioteca. Substitua esta página pelo documento gerado na versão final da sua monografia.

FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

Desenvolvimento de fechadura inteligente utilizando tecnologia RFID

Olimpio Pimenta

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI499 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Sistemas de Informação e aprovada pela Banca Examinadora abaixo assinada:

Nome Completo do Orientador
Titulação
DECSI – UFOP

Nome Completo do Coorientador
Titulação
DECSI – UFOP

Nome Completo do Convidado1
Titulação
Examinador
DECSI – UFOP

Nome Completo do Convidado2
Titulação
Examinador
DECSI – UFOP

Nome Completo do Convidado3
Titulação
Examinador
DECSI – UFOP

João Monlevade, 11 de março de 2019

Este trabalho é dedicado aos meus pais, meus irmãos, cujo amor incondicional sempre existiu e muitas vezes foi além de qualquer coisa que exista, tornando possível a concretização deste sonho.

Agradecimentos

Agradeço a Deus por todos momentos de aprendizado e convivência durante todo curso. Aos amigos que encontrei na universidade foram somando a medida que a caminhada se iniciava durante os períodos até findar-se, em especial Paulo Sérvulo, Gabriel Cordeiro, Glaucus Riveira. Aos meus amigos que sempre estão comigo Alisson Mamacendo, André Sales, Victor Augusto e aos demais que não citei, levo em meu coração sempre. Minha família, minha Avó, meus primos, tios e aos que estavam comigo durante esta caminhada.

“Science is more than a body of knowledge; it is a way of thinking.”

— Carl Sagan (1934 – 1996),
in: The Demon-Haunted World: Science as a Candle in the Dark.

Resumo

A Internet das coisas está cada vez mais presente na vida da população. Sendo ela por meio de dispositivos vestíveis, como pulseiras e relógios inteligentes. Nesse sentido, tecnologia por rádio frequência pode ser utilizada cartão de passagem em ônibus coletivo, cobrança automática de pedágio e entre outras aplicações. Sendo assim, este trabalho descreve os elementos necessários, para protótipo de sistema IoT, com sistema web para agendamento de laboratório. Sendo aplicação IoT desenvolvida com sensor de rádio frequência, utilizando chaveiro ou cartão para autenticação no sistema e ter seu acesso ao laboratório, de acordo com suas regras de acesso. O sistema será aplicado Universidade Federal de Ouro Preto - campus João Monlevade/MG. Ao longo do trabalho serão abordados assuntos pertinentes sobre o processo de composição dos componentes que integra o sistema IoT, com funcionamento uníssono com aplicação web responsiva, seja por computador, dispositivo móvel ou *tablet*. Sua concepção foi possível pela integração de microcontrolador, sensor de rádio frequência, servo motor e LCD responsável pela interação com usuário, previamente cadastrado no sistema com seu código de acesso proveniente do cartão/chaveiro.

Palavras-chaves: trava eletrônica por RFID. IoT. agendamento.

Abstract

IoT is increasingly present in the lives of a portion of the population. Being present for wearable devices such as bracelet or smart watch. In this sense, radio frequency technology can be used as a collective bus ticket, automatic toll collection and other applications. Thus, this work describes the necessary elements, for prototype development of IoT system, with web system for laboratory scheduling. Being IoT application developed with radio frequency sensor, using keychain or card for authentication in the system and have their access to the laboratory, according to their access rules. Will be applied Federal University of Ouro Preto - campus João Monlevade / MG. Relevant content is addressed about the process of component composition that integrates the IoT system, working in unison with responsive web application, whether by computer, mobile device or tablet. Its conception was made possible by the integration of microcontroller, radio frequency sensor, servo motor and LCD responsible for user interaction, previously registered in the system with its access code from the card / key ring.

Key-words: latex. abntex. text editoration.

Lista de ilustrações

Figura 1 – Exemplo de circuito da tag RFID.	18
Figura 2 – Diagrama do sistema.	21
Figura 3 – Arduíno MKR1000.	23
Figura 4 – Pinagem Arduíno MKR100.	24
Figura 5 – Figura lcd 16x2 com seus respectivos pinos de comunicação	25
Figura 6 – Sensor RFID Mfrc522 de 12.56MHZ.	26
Figura 7 – cartão e chaveiro utilizada no projeto, para cadastro e autenticação de acesso ao laboratório.	26
Figura 8 – Figura lcd 16x2 com seus respectivos pinos de comunicação	27
Figura 9 – Processo de montagem com adição de sensor Mfrc522 e microcontrolador.	28
Figura 10 – Curcuito com lcd e microcontrolador.	29
Figura 11 – Adicionando servo motor.	30
Figura 12 – Circuito completo contendo todos componentes.	31
Figura 13 – Decomposição de layout em componente, pelo jeito de pensar em React.	32
Figura 14 – Fluxo de cadastro de usuário no sistema web.	35
Figura 16 – Escolha de conta Google e tela tela principal da aplicação.	36
Figura 15 – Tela de login utilizando Google Chrome.	36
Figura 17 – Fluxo de agendamento no sistema web.	37
Figura 18 – Laboratórios agendados do usuário.	38
Figura 19 – Laboratórios disponíveis ao usuário.	39
Figura 20 – Tela do perfil do usuário.	39
Figura 21 – Protótipo finalizado do sistema IoT.	40
Figura 22 – Log de inicialização do dispositivo com informações de acesso a rede wifi configurada.	40
Figura 23 – Teste de leitura da tag, com envio de código por rede sem fio para verificação de agendamento.	41

Lista de tabelas

Sumário

1	INTRODUÇÃO	13
1.1	O problema de pesquisa	14
1.2	Objetivos	14
1.3	Justificativa	15
1.4	Organização do trabalho	15
2	REVISÃO BIBLIOGRÁFICA	16
2.1	Trabalhos relacionados	16
2.2	Arduino	16
2.3	RFID	17
2.4	React	18
2.5	Redux	18
2.6	Firebase	19
3	DESENVOLVIMENTO	20
3.1	Visão unificada do sistema	20
3.2	Ferramentas utilizadas	21
3.3	Componentes utilizados e construção do esquema elétrico	23
3.3.1	Arduino Mkr1000	23
3.3.2	LCD 16X2	24
3.3.3	Sensor RFID	25
3.3.4	Mirco servo motor	26
3.4	Montagem dos componentes	27
3.5	Aplicação web	31
3.6	Aplicação IoT	32
3.7	Servidor	33
4	RESULTADOS	34
4.1	Funcionalidades de sistema	34
4.1.1	Cadastro de usuário	34
4.1.2	Login	35
4.1.3	Cadastro de agendamentos	37
4.1.4	Laboratórios agendados	37
4.1.5	Laboratórios livres	38
4.1.6	Meu perfil	39
4.2	Funcionalidades de sistema IoT	40

5	CONCLUSÃO	42
	REFERÊNCIAS	43

1 Introdução

A tecnologia vem mudando a forma como se interage com mundo, seja por automação de processos, que antes eram feitos manualmente ou pelo desenvolvimento de novos produtos. Empresas vem aumentando investimento tecnológico para suprir a demanda do mercado, seja por inteligência artificial, robótica, ciência de dados entre outras. Dentro das tecnologias que estão deixando de serem promessa, Internet das Coisas (Internet of Things (IoT)) vem ganhando força, deixando as pessoas mais conectadas, seja por utilizar pulseiras inteligentes que captam dados por sensores, assim transformando os dados em informação apresentada no celular ou visor LCD. Relógio que apresenta notificações em forma mensagem ao usuário, sensores que coletam dados e transformam em informações sobre a quantidade de calorias perdidas, batimentos cardíacos, estimativa de passo dados e etc. Internet das coisas é muito mais que apenas dispositivos vestíveis utilizados no corpo humano, é possibilidade automatização de vários setores da economia e da vida social, e assim apresentando uma nova ótica da interação homem máquina (MARGRANI, 2018).

Internet das coisas, é uma extensão da internet atual, onde dispositivos com melhores capacidades computacionais se comunicam pela internet, de maneira remota através de provedores de serviços, que acessados pelo meio (serviço), se controla o dispositivo (SANTOS et al., 2016) . Em seu rigor semântico, internet das coisas é um termo apresentando pelo aumento de comunicação entre máquinas, passando em volume a comunicação interpessoal pela internet; desenvolvimento de novos produtos, microdispositivos, sensores e demais componentes. Todos esses itens combinados enviando/recebendo dados tornam-se parte integrante da internet (MARGRANI, 2018) . Como conceito, internet das coisas não foi oficialmente apresentando até mil novecentos e noventa e nove, seu primeiro exemplo inicia na década de oitenta, com a máquina da empresa Coca Cola, localizada na Universidade Carnegie Melon. Foi desenvolvido um software para ser utilizado com o refrigerador, sendo como intuito apenas verificar disponibilidade de bebida em estoque e o estado (quente/frio) dessa bebida (FOOTE, 2013) . Os autores acima apresentam suas formalizações acerca do IoT, em que suas opiniões são necessárias existir comunicação entre dois dispositivos eletrônicos por meio da internet, caracterizando-os como IoT.

Entre as tecnologias encontradas no mercado, pode-se destacar à identificação por rádio frequência (RFID, *Radio-Frequency IDentification*) presente em pedágio, empresas de cobrança de passagem de ônibus circular. Hotéis, onde o hóspede pode utilizar o cartão para destravar a porta e assim conseguir entrar. Na prevenção de roubo de veículos e mercadoria, gerenciamento de tráfego; rastreamento de livros de biblioteca, além de outras aplicações e inovações por ainda surgir (LANDT, 2005) . Assim esta tecnologia em sua base histórica, remonta à Segunda Guerra Mundial, alemães, japoneses, americanos e

britânicos. Utilizavam radar, descoberto pelo físico escocês Robert Alexander Watson, para alertar aproximação de aviões inimigos, mesmo estando a quilômetros de distância, e seu problema (radar) se dava por não conseguir identificar aeronaves amigas ou inimigas e qual era o piloto do país de origem retornando de sua missão. Por estudo ou apenas teste, os alemães descobriram que se os pilotos realizassem uma determinada manobra área enquanto retornasse a base, isso mudaria o sinal de rádio refletido de volta. Este método serviu de alerta a equipe no solo responsável pelo radar, de que se tratava de aeronaves alemães e não aliadas (sendo em sua essência, o primeiro sistema passivo de RFID) (ROBERTI, 2005) .

Com esta breve explicação sobre IoT e RFID, pode-se dizer resumidamente que é uma unidade de hardware com alguns componentes eletroeletrônicos que se comunicam externamente por meio da internet, utilizando padrões de protocolo preestabelecido por agência reguladora de comunicação. Essa comunicação feita de maneira heterogênea entre componentes, se dá pelos serviços oferecidos por camadas de abstrações fornecidas pelo fabricante, onde atua em conjunto com órgãos reguladores. Promovendo assim uma quantidade variável de dados a serem inferidos. Este trabalho irá apresentar protótipo de dois sistemas. O primeiro para agendamento de laboratórios, por serviço web dividido em duas partes, uma parte layout e outra como servidor. O segundo sistema é da trava eletrônica utilizando RFID, que se comunica com servidor, para controle de acesso.

1.1 O problema de pesquisa

Foi observado no ICEA uma dificuldade para gerenciar o acesso aos laboratórios de ensino. Por vezes, professores esquecem de devolver a chave, se atrasam por ter que pegar a chave, ou ainda tem que se planejar por não conseguir acesso ao laboratório.



1.2 Objetivos

O trabalho consistirá de um sistema embarcado utilizando áreas do conhecimento de engenharia elétrica e computação, aplicado ao Instituto de Ciências Exatas e Aplicadas (ICEA). O sistema via web possuirá credenciais de acesso dos usuários cadastrados, sendo possível realizar agendamento de laboratórios do Instituto de Ciências Exatas e Aplicadas (ICEA), campus João Monlevade.

Os objetivos específicos do trabalho são:

- Permitir cadastro de docentes na plataforma;
- Realizar cadastro de agendamentos;
- Visualizar laboratórios agendados;



- Visualizar laboratórios disponíveis para agendamento por bloco;
- Desenvolvimento de circuito elétrico do sistema embarcado;
- Desenvolvimento de aplicação para controle de sensores, aplicado ao microcontrolador do sistema embarcado;

1.3 Justificativa

Com intuito de melhorar a qualidade de gerenciamento de agendamento e acesso aos laboratórios, o presente trabalho com as tecnologias empregadas, irá conceder melhor flexibilidade no acesso à plataforma de agendamentos, sendo possível realizar por *tablet*, computador e *smartphones*. O acesso aos laboratórios por RFID, facilitará no processo de acesso ao laboratório e que será feita de forma automatizada, por ser a identificação do usuário.

1.4 Organização do trabalho

Este trabalho está organizado da seguinte maneira. O capítulo dois, apresenta revisão bibliográfica, apresentando os conceitos necessários das tecnologias empregadas no trabalho. Capítulo três, apresenta metodologia empregada na pesquisa e na prática. Capítulo quatro, apresenta o resultado final prático do trabalho proposto e seus testes. Capítulo cinco, considerações finais e propostas de melhoria.



2 Revisão bibliográfica

Este capítulo apresenta uma revisão da literatura, descrevendo os conceitos necessários de cada item do projeto separadamente e trabalhos relacionados.

2.1 Trabalhos relacionados

O trabalho apresentado por ([TECHNOLOGY, 2016](#)), é implementação de sistema de gerenciamento automático de porta. É implementado pelo esquema elétrico e seu respectivo código fonte, sendo um modelo bem próximo, exceto por algumas características. No esquema do circuito, utiliza-se resistores para limitar corrente ao LED RGB, de acordo com a passagem do cartão de identificação ao sensor, sendo também adicionado em seu projeto uma buzina, seguindo mesma ideia do LED ao passar o cartão no sensor. O cadastramento do código do cartão é feito diretamente no microcontrolador Arduino Uno, sendo comparado ao aproximar do sensor e assim aciona a trancar (servo motor) para destravar ou não a porta.

O trabalho apresentado por ([MUKHERJEE, 2016](#)), é acesso de segurança usando leitor RFID. É adicionado em um Arduino Uno um sensor de leitura de cartões RFID. No início do artigo demonstra como utilizar um código de exemplo da biblioteca do sensor usado, indicando como realizar os testes de leitura do cartão, sendo seu código de identificação visto por uma janela. Ao final demonstra com código fonte disponibilizado a leitura do cartão, e a autorizando ou não acesso pelo código de acesso gravado na memória do Arduino.

Apresentado por ([SMERKOUS, 2016](#)), leitor de cartão RFID Ethernet Arduino. Seu sistema é composto pelos seguintes componentes, Arduino Nano, sensor RFID e módulo de conexão com internet. Em seu diagrama é apresentada uma visão unificada do sistema. Sua comunicação feita pela internet e seu processo de verificação do código do cartão após leitura, sendo enviado seu código pela internet, e após recebido sua resposta é feita verificação para liberação da porta.

2.2 Arduino

Projeto de pesquisa iniciado nos anos dois mil por Banzi, Guartilles, Igoe, Martion e Melis no instituto de Design de Interação de Ivrea, baseava-se no projeto Processing que era uma linguagem de aprendizado para codificação dentro de contextos. Este sistema de código aberto, aliado a um hardware de fácil manuseio, é capaz de transformar praticamente

qualquer ideia em solução embarcada. Estima-se que até 2021, o mercado de IoT movimentará seis trilhões de dólares, sendo Arduino a principal escolha para fabricantes de eletrônicos, por sua adoção em massa em diversos projetos eletrônicos e sendo assim um dos responsáveis pelo Movimento Maker. (ARDUINO, 2019a).

Sendo no Arduino, pela facilidade em desenvolver, familiaridade com a linguagem C/C++, por fácil acesso às bibliotecas com sua plataforma de desenvolvimento e documentação. Comparado ao Raspberry PI, o Arduino tem um menor custo em sua aquisição, e pela necessidade apenas de realizar controle de sensores e sem a necessidade de microprocessador, tão menos sistema operacional; realizando comunicação com servidor, enviando requisições de tentativas de acesso ao laboratório. O microcontrolador escolhido, será apresentado no desenvolvimento.

2.3 RFID

Na concepção dos primeiros modelos de identificação por rádio frequência, Watt, foi o responsável pelo desenvolvimento do sistema ativo amigo de identificação (IFF). Adicionando um transmissor em cada avião britânico, ao receber sinais de estações de radar via solo, o sistema ativo amigo de identificação enviava um sinal de volta, e este assim era identificado pela estação de radar em solo, como aeronave amigável. Esta breve caracterização básica do funcionamento em sua criação, descreve o conceito do funcionamento do RFID, assim o sinal é recebido por transponder, que após alimentado é iniciado a reflexão do sinal, podendo ser ativo ou passivo (ROBERTI, 2005) .

Como observa-se na [Figura 1](#), os componentes internos de uma tag RFID, são formados por antena de recepção e envio de sinal gerado por ondas de rádio, e um microchip contendo código de identificação da respectiva tag. A leitura da tag é feita ao se aproximar do leitor, sendo assim gerado uma onda eletromagnética na tag que por sua vez gera corrente elétrica no circuito, por fim enviando do microchip para antena o dado armazenado e este sendo recebido pelo leitor. Neste sentido foi escolhido o sensor Mfrc522, acoplado ao microcontrolador, responsável pela leitura do cartão de acesso, sendo posteriormente lido, decodificado e enviado pelo microcontrolador ao servidor, para verificação de acesso ao local.



Figura 1 – Exemplo de circuito da tag RFID.

Fonte: ([AFIXGRAF, 2018](#)).

2.4 React

Em 29 de maio de 2013 surge uma biblioteca para auxiliar no desenvolvimento de interfaces de usuário baseada em JavaScript ([REACT, 2019b](#)). Seu conceito de desenvolvimento é feito por virtual DOM (*Document Object Model*), sendo alocado em memória uma cópia do estado da aplicação. A sincronização com DOM original é feita pela biblioteca *ReactDOM*, neste processo de sincronização é verificada partes que foram modificadas e são atualizadas no layout do usuário. Neste modelo de atualização de aplicação, realiza abstração do processo manual de manipulação e atualização de atributos dentro da aplicação web ([REACT, 2019c](#)).

Pela facilidade no desenvolvimento de aplicações web empregando com esta biblioteca, foi utilizada na criação dos layouts dinâmicos pela sua estratégia de divisão do layout por componentes. Na divisão do layout da aplicação, utiliza-se técnica de princípio de responsabilidade única, um componente é concebido em sua essência apenas com uma responsabilidade. Caso ele cresça por necessidade, ou seja, diagnosticado outras responsabilidades que não foram pensadas, é aconselhável a refazê-lo em componentes menores ([REACT, 2019a](#)). Assim todo layout da aplicação para agendamentos de laboratórios foi pensado e desenvolvido, por este princípio (responsabilidade única) a divisão dos componentes que integram o contexto da aplicação é melhor aproveitada, com facilidade em depuração de código atrás erros que possam ocorrer e melhor diagnosticado.

2.5 Redux

Como apresenta o site mantenedor, Redux é um container para manipulação de estado para aplicações JavaScript ([REDUX, 2019](#)). Criada por Dan Abramov da arquitetura Flux, desenvolvida pelo Facebook. Durante o desenvolvimento de aplicações grandes, utilizando

ou não React, é necessário em alguns casos, realizar compartilhamento de informações entre componentes de sua aplicação web. Isso se torna um problema, por causa dos diferentes níveis de hierarquia que pode acontecer. Esta informação precisa ser disponibilizada de maneira global, sendo consumida por outros elementos em diferentes estados da aplicação (MARUTA, 2018).

O Redux resolve este problema por ser acessível globalmente dentro de qualquer componente da aplicação. Sua composição de atualização de estado é feita por: actions disparadas pelo comportamento da aplicação com usuário; reducers que irão atualizar o estado pelas modificações realizadas; stores pra gerenciar o armazenamento do estado a cada atualização da aplicação (CAPRETZ, 2018).

Para controlar o estado do usuário estar logado na aplicação, foi utilizado o Redux para gerenciar estado global, em que os componentes apenas consumiam este estado. Nesse sentindo, o fluxo após realizar login era armazenar os dados do usuário e posteriormente ser consumido a cada vez que acessava funcionalidades dentro da aplicação, caso contrário era redirecionado para página de login.

2.6 Firebase

Dentre as ferramentas que auxiliam os desenvolvedores na caminhada em seus projetos, o *Firebase* apresenta uma boa proposta, pela sua curva de aprendizado inicial, com alta disponibilidade e planos, indo do gratuito ao pago, que podem engajar seu produto. Seu ponto de partida foi uma empresa chamava Envolv, oferecendo ferramentas para bate-papo em tempo real, e anos depois se tornou *Firebase* (FRANKEL, 2017).

Ao longo dos anos, é visto uma crescente adoção de serviços para computação em nuvem, moldando novamente como é feito o modelo até então, aplicações computacionais. Serviços de computação em nuvem como, *Microsoft Azure*, *Google Apps Engine* e *Heroku*, permite criação de software com maior facilidade de escalonamento, deixando de se preocupar com infraestrutura na sua execução. Na visão dos criadores, *Firebase* é um serviço como *Dropbox*, serviço de compartilhamento de arquivos por repositório online, ou seja, seu *Dropbox* para aplicações (METZ, 2012).

Assim foi escolhido os serviços do *Firebase* autenticação e armazenamento. O primeiro para realizar o processo de validação de contas, na hora do cadastramento do usuário na aplicação e *Realtime Database* para armazenamento, sendo seu funcionamento por banco de dados documental ou não Sql.

3 Desenvolvimento

Este capítulo irá descrever a concepção do sistema proposto, bem como os métodos, tecnologias, ferramentas e demais acessórios pertinentes durante as fases de desenvolvimento da aplicação. São também descritos procedimentos que visam facilitar o desenvolvimento e os testes da aplicação. A seguinte metodologia foi utilizada.

- Desenvolvimento do esquema elétrico do protótipo, indo ao estudo de utilização por *datasheets* e documentação oficial, bibliotecas de comunicação por protocolos entre componentes e sua ferramenta de desenvolvimento utilizada na sua concepção, além do processo de adição de partes menores até sua finalização.
- Desenvolvimento do sistema de comunicação que atende requisições por protocolo HTTP, recebendo requisições tanto do sistema do Arduino e da aplicação web.
- Desenvolvimento de sistema web baseado em componentes, indo de sua concepção de usabilidade em diferentes dispositivos. Implementação da camada de serviço comunicando com servidor.
- Estruturação e desenvolvimento do banco de dados, contendo solicitações diretas do servidor.
- Processo de validação e realização de testes, possibilitando melhores chances de correções, prevenção de erros, e otimização de código, possibilitando um menor custo de processamento.

3.1 Visão unificada do sistema

A aplicação apresenta três vertentes, primeiro um sistema gravado no microcontrolador, sendo que este gerencia os componentes acoplados no circuito. A primeira dessas vertentes audita acesso feito pelo usuário ao passar o cartão no sensor, enviado ao servidor uma solicitação de acesso ao local e posterior resposta do servidor ao microcontrolador. Segundo é aplicação web de acesso pelo usuário para realizar agendamentos. Servidor, para processamento de requisições do sistema web e do microcontrolador.

Na aplicação web, com propósito de responsividade por orientação de dispositivo eletrônico inteligente é realizado o cadastramento de usuário, autenticação por conta Google vinculada à universidade, agendamento de laboratório, exclusão, edição e visualização por bloco os laboratórios livres de acordo com a data do dia. É realizado o cadastramento de usuário, utilizando conta Google vinculada à universidade e fornecida aos docentes. Após

realizado cadastramento é possível autenticar no sistema, assim possível criar, editar e excluir agendamentos pela data vigente. A [Figura 2](#) apresenta um esquema ilustrando a arquitetura utilizada na aplicação.

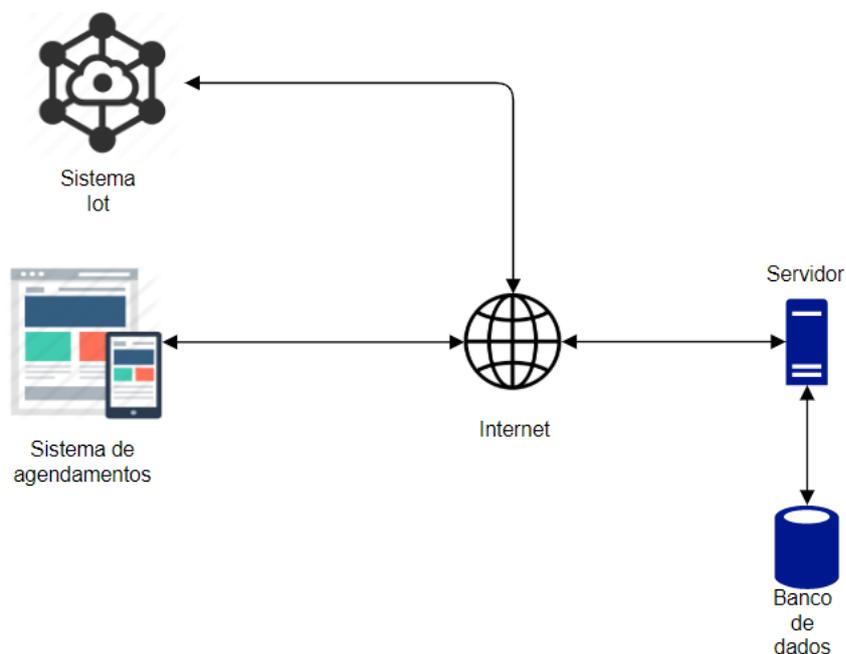


Figura 2 – Diagrama do sistema.

Fonte: ([VISUAL PARADIGMA, 2019](#)).

O servidor, tem função de aceitar requisições, processar e enviar resposta ao destinatário solicitante do serviço. Foram adicionadas bibliotecas que acessem os serviços necessários a cada solicitação do cliente, como acesso ao banco de dados, realizando tarefas de leitura, escrita e exclusão. Autenticação de usuário, pela conta cadastrada e demais regras necessárias no projeto.

3.2 Ferramentas utilizadas

Durante a fase de pesquisa e revisão bibliográfica, foram escolhidas ferramentas de desenvolvimento, linguagens, banco de dados e entre outros. Foi utilizado Microsoft Visual Studio Code, para desenvolvimento do projeto, sendo utilizando com React para desenvolvimento, sendo baseado em JavaScript. Foi adicionado ao projeto um conjunto de ferramentas, apresentado pela Microsoft como um super set, sendo ferramenta para adicionar tipagem ao JavaScript. Por questões de padronização na linguagem e tornando ambiente mais produtivo pela inferência de tipos, verificação estática e facilidade na refatoração de código sem perda robustez na criação de componentes ([TYPESCRIPT, 2019](#)).

Durante o desenvolvimento do sistema para o Arduino em linguagem C, foi utilizando um software oferecido pela própria Arduino, tendo uma biblioteca rica e exemplos de projetos para testes de componente eletrônico.

Também foi utilizado *Firebase*, com plano Spark gratuito que fornece vários serviços, dentre estes, Realtime Database e serviço de autenticação. Realtime Database é um serviço de armazenamento em banco de dados na nuvem, sendo seu foco em banco documental. Os dados são enviados e atualizados na base de dados em milissegundos, toda sua aplicação pode ser em torno do cliente, deixando o desenvolvimento, escalonamento, e demais atividades de tratamento de dados aos encargos do servidor, possibilitando agilidade em seu desenvolvimento seja com layout ou servidor. A praticidade por esta aplicação torna a sinergia do programador toda em sua aplicação, e não nos detalhes de implementação de serviços para atendimento a requisições ao banco ([DATABASE, 2019](#)). O serviço de autenticação, possibilita simplificações no processo de login de usuários, sua fácil integração com outros serviços como Facebook, GitHub, Twitter, Google, e contas de e-mail com processo de confirmação de conta. Componente FirebaseUI Auth implementa boas práticas de autenticação de usuários para dispositivos móveis e páginas web. Fácil implementação do serviço sem perda de segurança no processo de autenticação fornecida internamente pela Google ([AUTHENTICATION, 2019](#)). Neste trabalho serão utilizadas as contas de serviço de e-mail disponibilizadas pela Google aos docentes, em que facilita o processo de validação do usuário na aplicação, ou seja, não será preciso realizar o processo de validação de conta fornecida pelo usuário, além de utilizar uma conta fornecida pela universidade em vez da conta pessoal. Logo, todo fluxo da aplicação será por validação de informações sobre a inserção de dados armazenamento no banco de dados.

Foi utilizado no desenvolvimento software Fritzing, desenvolvida por Universidade Aplicada de Postdam, na Alemanha ([NEGROMONTE, 2019](#)). Este projeto é uma iniciativa de código aberto que apoia profissionais na área da pesquisa, designers, artistas e demais entusiastas no desenvolvimento de protótipos eletrônicos, sendo modelo documentável de esquemas elétricos, layout de placas de circuito impresso (PCI), sendo assim possível compartilhar projetos. Menos dispendiosa em sua curva de aprendizado comparando outros softwares para desenvolvimento de placas do mercado, como Proteus.

Durante o planejamento da solução web, foi pensado na responsividade sendo um dos pilares do desenvolvimento do layout da aplicação, sendo possível utilizar para quase qualquer dispositivo com tela de interação seja por toque ou periférico. Durante a realização de pesquisa, foi encontrado uma biblioteca de interface para usuário, com foco em designer limpo, responsivo, utilizando a mesma arquitetura de componentes e framework em React. Foi encontrado dentre as ofertadas na atualidade, o Material UI ([MATERIAL-UI, 2019](#)). Seu foco é implementar os conceitos de designer apresentados pela Material designer ([DESIGN, 2019](#)) apresentada pelo Google, sendo o mesmo conceito

adotado pelo Google em seus dispositivos Android.

Durante do desenvolvimento foi pensado na responsividade sendo um dos pilares do desenvolvimento do layout da aplicação, sendo possível utilizar para quase qualquer dispositivo com tela de interação seja por toque ou periférico. Durante algumas pesquisas foi encontrado uma biblioteca de interface para usuário, com foco em designer limpo, responsivo, utilizando a mesma arquitetura de componentes e framework em React. Foi encontrado dentre as ofertadas na atualidade, o Material UI ([MATERIAL-UI, 2019](#)). Seu foco é implementar os conceitos de designer apresentados pela Material designer ([DESIGN, 2019](#)) apresentada pelo Google, sendo o mesmo conceito adotado pelo Google em seus dispositivos Android.

3.3 Componentes utilizados e construção do esquema elétrico

3.3.1 Arduino Mkr1000

Durante escolha do sistema microcontrolador a ser usado no projeto, o que satisfizesse por seu consumo baixo de energia e sendo a linha definida pela empresa (Arduino) para IoT, foi MKR 1000 ([Figura 3](#)). Sendo projetado para eficiência em soluções com utilização de conexão a redes sem fio, de maneira econômica ([ARDUINO, 2019b](#)). Baseado em Atmel ATSAMW25 SoC (*System on Chip*), que se tem incorporado sua família SmartConnect da Atmel Wireless, específicas para IoT.



Figura 3 – Arduino MKR1000.

Fonte: ([ARDUINO, 2019b](#))

Esta placa permite utilizar mais de um meio para alimentação, na sua pinagem, é possível utilizar uma bateria de lítio com tensão nominal de 3.7V e mínimo 700mAh de corrente, com Li-po. Pelo pino Vin, pode se acoplar uma fonte com mínimo de tensão de 5V e máximo de 6V. Alimentação usb de 5v e podendo alimentar outro circuito e por fim Vcc de geração 3.3V regulado por regulador interno (??).

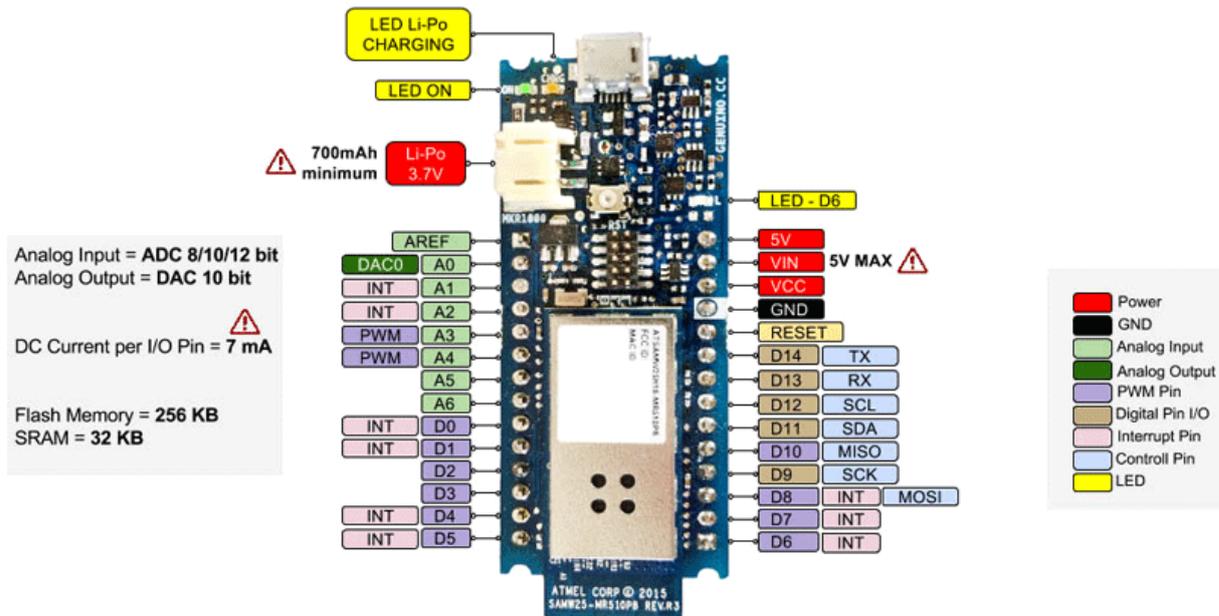


Figura 4 – Pinagem Arduino MKR100.

Fonte: (COMPONENTS101, 2017)

Percebe-se na Figura 4 a pinagem para comunicação SPI, que facilita na utilização do sensor RFID escolhido no trabalho, com 7 pinos de comunicação analógica com resolução de 8/10/12 bits. Os pinos D0 - D14, podem ser utilizados como entrada ou saída, doze pinos que fornecem PWM de 8 bits, um pino DAC0 que é realmente analógico e demais configurações (COMPONENTS101, 2017).

3.3.2 LCD 16X2

Pela proximidade de viabilização em decorrer da proposta oferecida pelo sistema de integração, utiliza-se um LCD 16X2 (Figura 5), em que serão dispostas algumas mensagens que forneça ao usuário quando o mesmo for tentar acessar o sistema pelo cartão de identificação, uma mensagem condizente mostrando seus acesso autorizado ou negado ao sistema. No folha de especificações do produto tem informações relevantes de como realizar a comunicação com microcontrolador (Arduino MKR1000), sua tensão máxima e mínima de operação (LCD), tabela contendo descrição de cada pino respeitado sua configuração, podendo ela ser entrada ou saída (THOMSEN, 2011).

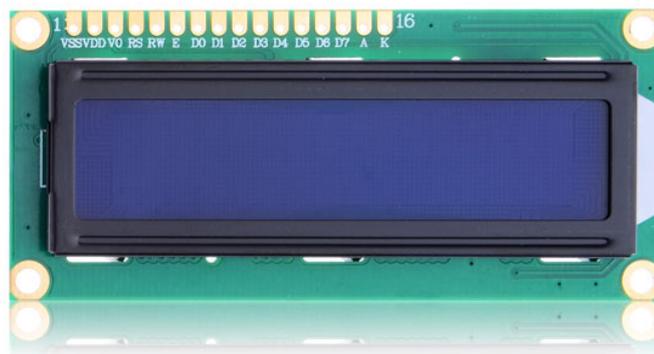


Figura 5 – Figura lcd 16x2 com seus respectivos pinos de comunicação

Fonte: (ELETRÔNICOS, 2019)

3.3.3 Sensor RFID

Dentre os leitores mostrados em alguns sites eletrônicos de venda, foi encontrado um em particular com preço consideravelmente mais em conta que as soluções oferecidas por outras empresas. O sensor escolhido foi RFID Mfrc522 da fabricante NXP, desenvolvido pela empresa NXP, com uma frequência de operação de 13.56MHZ. Componente funciona com faixa de operação de tensão de 3.3 volts. Pelo seu tamanho pequeno apresenta menor consumo energético se comparado a outros modelos oferecidos por outras empresas que apresenta soluções integradas em suas vendas, ou seja, não oferecem apenas o componente e sim uma solução integrada com sensor, que não se adéqua a proposta (Figura 6) do projeto (GBUR, 2017).



Figura 6 – Sensor RFID Mfrc522 de 12.56MHZ.

Fonte: (GBUR, 2017)

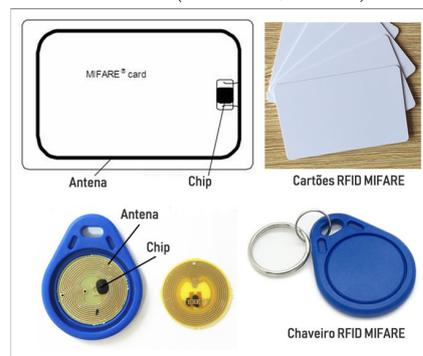


Figura 7 – cartão e chaveiro utilizada no projeto, para cadastro e autenticação de acesso ao laboratório.

Fonte: (ELETRÔNICA, 2019)

Foi utilizado no projeto uma cartão e chaveiro azul (Figura 7). Este sensor foi posteriormente acoplado ao microcontrolador em sua fase de construção.

3.3.4 Mirco servo motor

Os servos possuem engrenagens integradas com eixo que pode ser controlado com precisão. Servos posicionam em vários ângulos distintos, geralmente indo de zero até cento e oitenta graus. Servos de rotação contínua permitem rotação em seu eixo seja variada em diferentes velocidades (SERVO, 2019).

Foi escolhido para o projeto micro servo 9g SG90 TowerPro, para simulação de tensão aplicada na fechadura, acionando a mudança de posição em cento e oitenta graus a partir do seu estado inicial, simbolizando abertura e fechamento da tranca da porta (Figura 8). Contendo em sua extremidade um pino com três fios, fio vermelho, para energia. Fio amarelo, laranja ou branco para potência. Fio marrom ou preto para terra. A alimentação é de 5V, sendo um valor considerável para projetos com mais de um servo, e assim necessitando de uma fonte externa de energia, como vem a título de exemplo na

viabilidade do projeto e sendo utilizado apenas um servo, não tem nenhum problema neste caso (THOMSEN, 2019).

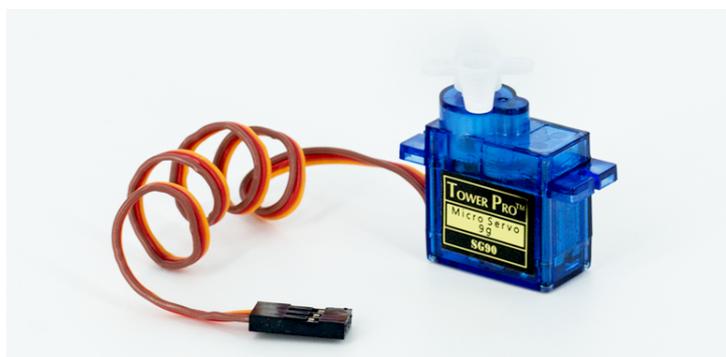


Figura 8 – Figura lcd 16x2 com seus respectivos pinos de comunicação
(THOMSEN, 2019)

3.4 Montagem dos componentes

Foram feitas pesquisas dentro da documentação oficial do Arduino, bibliotecas de interface de comunicação do componente pelo GitHub, até encontrar uma biblioteca compatível. Após a pesquisa por trabalhos correlacionados, foi encontrado um modelo para teste do componente dentro das bibliotecas do Arduino.

Após realizado testes, foi configurado o componente com microcontrolador (MKR1000) (Figura 4), estas foram utilizadas na comunicação entre dispositivo, sendo MISO, MOSI e VCC de 3.3 volts do sensor (Figura 9).

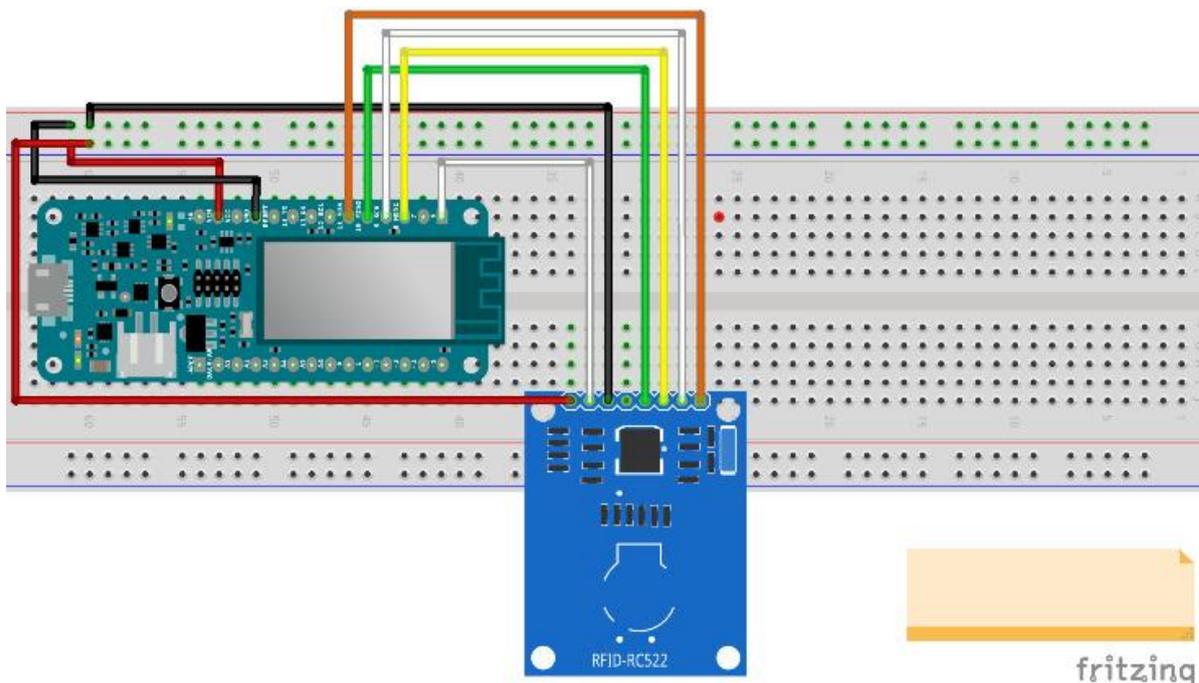


Figura 9 – Processo de montagem com adição de sensor Mfrc522 e microcontrolador.

Com o drive de manipulação Hitachi HD44780, geralmente utilizado com interface LCD de 16 pinos e aplicado ao hardware usado, foram feitos testes pelos exemplos fornecidos na biblioteca do LCD. Por ter interface paralela, é necessário fazer a manipulação de vários pinos ao mesmo tempo, para controlar a exibição do conteúdo. Encontrado exemplo utilizando conexões com portas digitais, foram feitas alterações nos pinos controladores, para usarem portas analógicas em vez das digitais apresentadas no esquema de montagem. Isso diminui a quantidade de portas digitais usadas no projeto, assim liberando mais portas para adicionar novas possíveis melhorias no sistema (Figura 10). Em sua arquitetura é concebida um pino de seleção de registro (RS), controlando onde é feita a gravação de dados no LCD; pino de leitura/gravação (R/W), que seleciona o modo, seja de escrita ou leitura, 8 pinos (D0-D7) de alteração de estado em nível lógico baixo ou alto adicionado aos registradores para leitura ou escrita. No dispositivo ainda tem um pino para contraste de tela (V0), pinos para fonte de alimentação (5V), pino para terra (GND) e pino LED para controle de intensidade de luz de fundo no LCD (ARDUINO, 2019c).

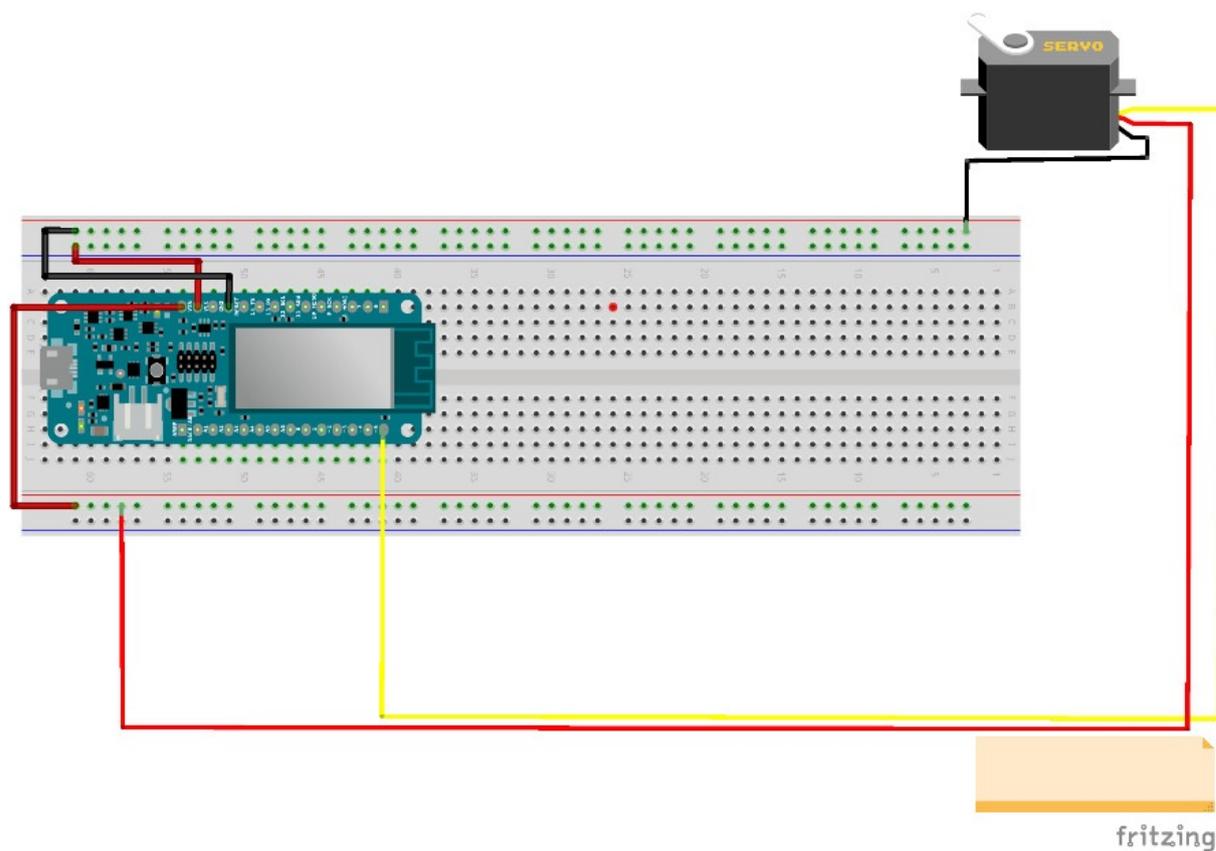


Figura 11 – Adicionando servo motor.

Abaixo (Figura 12) se encontra o resultado de cada teste realizado individual e depois coletivo com cada componente usado no projeto, sempre sendo adicionando cada parte em função da modularidade do sistema, ou seja, o seu processo de montagem por partes.

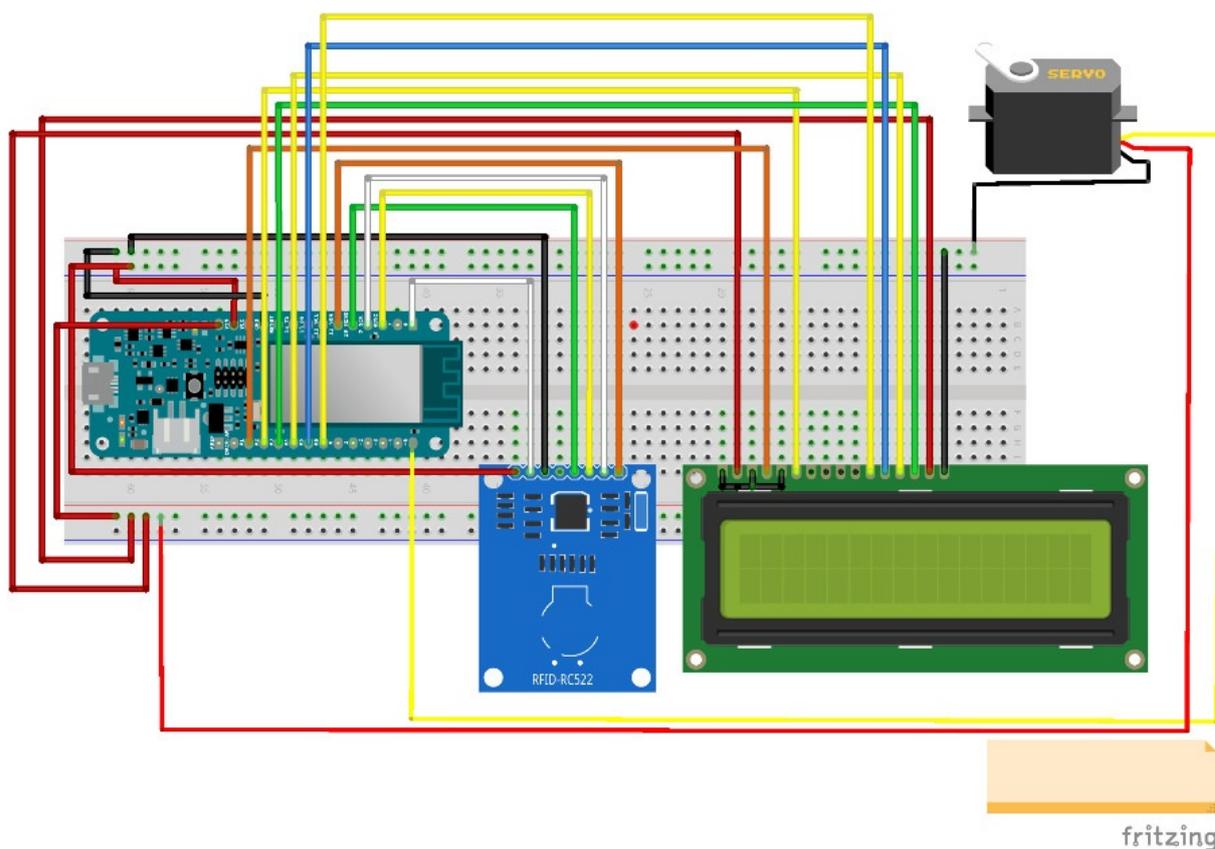


Figura 12 – Circuito completo contendo todos componentes.

3.5 Aplicação web

Seguindo a proposta de desenvolvimento do React, foi dividido o layout da aplicação em componentes, sendo eles menu, meu perfil, tela de login, agendamentos livres e novos, rodapé com menu e barra superior contendo submenu vinculado a foto do usuário. Durante o aprendizado de se pensar no jeito React de divisão de layout em web componentes, foi adotado filosofia de orientação a objetos em linguagens de alto nível, utilizar-se o princípio de responsabilidade única, ou seja, o componente deve idealmente fazer apenas o que lhe cabe em sua concepção. Caso ele aumente por características de sua solução, é feita sua decomposição em subcomponentes menores (REACT, 2019a).

A primeira divisão de componente aparece na cor cinza (Figura 13), sendo barra superior, possui logo da universidade, título e um outro subcomponente anexado. Sendo submenu caracterizado por exibir acesso ao componente de perfil e opção de sair da aplicação. Na figura que corresponde a foto do usuário (Figura 13), é visualizado após realizar login na aplicação, suas informações básicas, que por sua vez é um estado global gerenciado pelo Redux e acessível por toda aplicação. Iniciando pela cor roxa (Figura 13), é possível ver o acesso aos componentes de novo agendamento, laboratórios agendados; laboratórios livres e meu perfil.

A aplicação comunica com requisições simples utilizando padrões de comunicação por protocolos web (HTTP), sendo realizadas solicitações ao servidor pelos serviços oferecidos na aplicação web, sendo este processado pelo servidor e enviando sua resposta a aplicação solicitante. O formato de comunicação por JSON, atende ao modelo de dados trafegados entre ambos.



Figura 13 – Decomposição de layout em componente, pelo jeito de pensar em React.

3.6 Aplicação IoT

Nesta aplicação o microcontrolador gerencia os componentes responsáveis pela interação com usuário e sua posterior comunicação com servidor. Após aproximar o cartão ou chaveiro no sensor, é feita leitura do código gravado. Após realizada leitura, o dispositivo LCD apresenta informações no formato de mensagens pequenas ao usuário sobre realização de comunicação com servidor e sua resposta, sendo seu acesso liberado ou negado com base na resposta do servidor.

3.7 Servidor

Desenvolvido pela necessidade de processar requisições tanto da aplicação web e do embarcado. O desenvolvimento do servidor foi concebido utilizando JavaScript com TypeScript em ambiente node, dividida em três camadas. A primeira camada representa inicialização com as respectivas configurações de acesso aos servidores. Segunda camada com as rotas, para acessar a terceira camada que é onde realmente feito o processamento das requisições. Utilizando conceito *Model-View-Controller* (MEDEIROS, 2013), foram elaboradas as três camadas e utilizando as restrições de cada funcionalidade oferecida para aplicação web e microcontrolador. Nas restrições para o microcontrolador, ao receber solicitações de verificação de acesso é feita uma solicitação ao serviço do Realtime Database, buscando os agendamentos realizados na data vigente e pesquisado agendamento. Utilizando o código de identificação recebido, é pesquisado se existe agendamento na data e horário, caso exista, é enviada uma resposta para o microcontrolador. Para aplicação web de agendamentos, é oferecido, cadastro do usuário, cadastrados de agendamentos, filtro por laboratórios livres de acordo com a data vigente e bloco escolhido, exclusão e edição de agendamentos realizados e autenticação.

4 Resultados

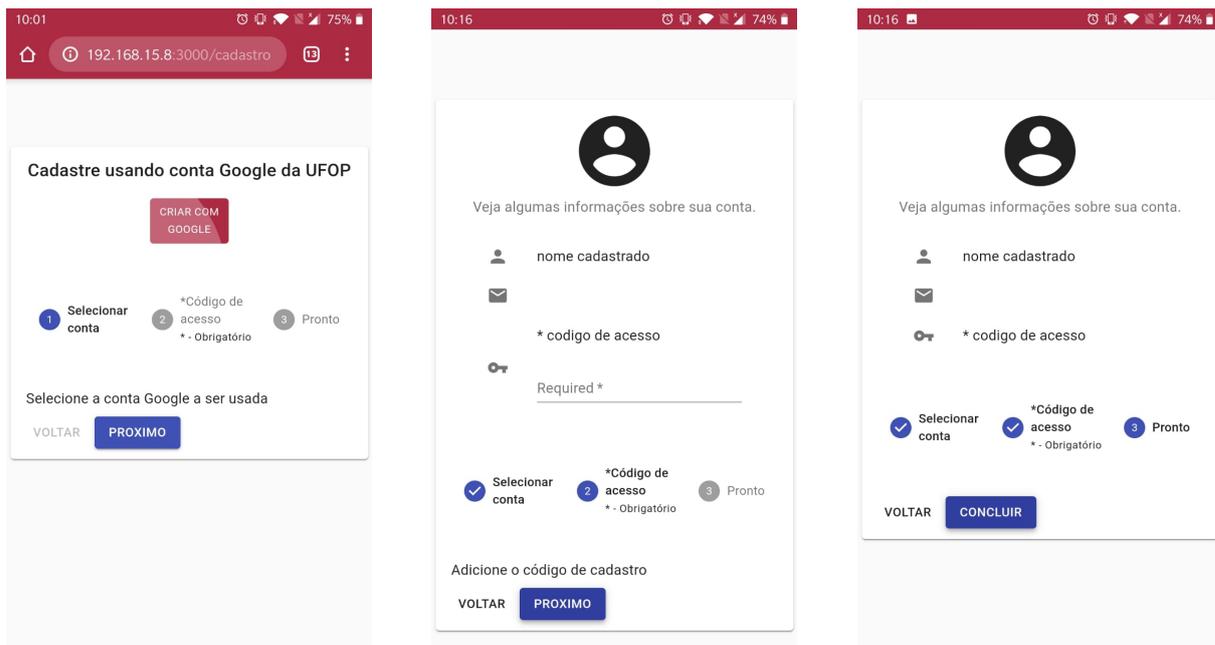
Neste capítulo serão apresentados os resultados obtidos e testes realizados. Foi realizado teste de criar, ler, atualizar e deletar agendamentos, cadastramento de novos usuários ao sistema, teste de acesso e privilégio integrado ao sistema IoT.

4.1 Funcionalidades de sistema

Nesta seção apresentará funcionalidades da aplicação web, mostrando os fluxos de acesso ao sistema e comportamento em computador e móvel. Será apresentada abaixo as funcionalidades da aplicação web e sistema IoT interagindo com servidor. Após selecionada qualquer opção na tela principal, ao se redirecionar a outra tela é possível ver outro menu inferior, para acesso rápido as outras funcionalidades. Aplicação web foi pensada na utilização vertical em dispositivo móvel.

4.1.1 Cadastro de usuário

Foi criado um fluxo simples para cadastro ao sistema, podendo cadastrar qualquer usuário de conta vinculada ao Google ([Figura 14a](#)). Quanto ao cadastro, não é feita regras de restrição (o que pode ser implementando posteriormente), podendo qualquer pessoa com acesso ao link efetuar seu registro. Após escolhido a conta vinculada ([Figura 14b](#)), os respectivos campos de usuário como, nome, e-mail e foto de perfil são automaticamente preenchidos, ficando a cargo do usuário apenas digitar código do dispositivo RFID utilizado. Nesta etapa é de suma importância adicionar o código de acesso, sendo ele utilizado na verificação do acesso ao laboratório. Toda tentativa de acesso a qualquer laboratório, é utilizado o código de acesso em seu processo, assim qualquer agendamento realizado pelo usuário é adicionado também seu código de acesso no agendamento.



(a) Tela inicial do cadastro do usuário por conta Google.

(b) Tela preenchida após escolha de conta Google.

(c) tela de revisão de cadastro.

Figura 14 – Fluxo de cadastro de usuário no sistema web.

4.1.2 Login

Na tela de login (Figura 15) temos opção de somente logar na conta cadastrada anteriormente pelo provedor Google, como assim definido no contexto do trabalho. Ao clicar aparecerá uma janela apresentando qual conta você vai utilizar para prosseguir, e assim escolhido a conta cadastrada (Figura 16a). Após logado, o usuário é redirecionado para tela principal da aplicação (Figura 16b), apresentando as opções do sistema.

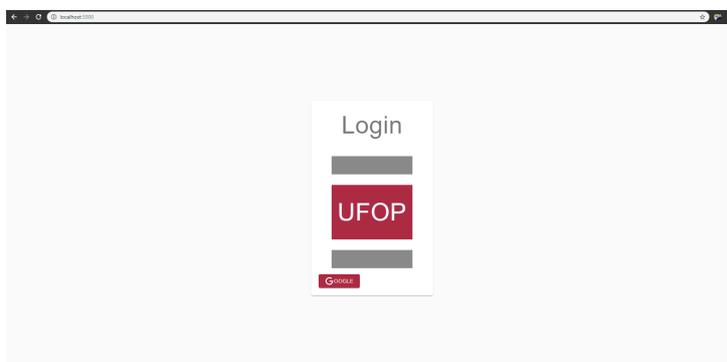


(a) Tela de escolha de conta Google do usuário.

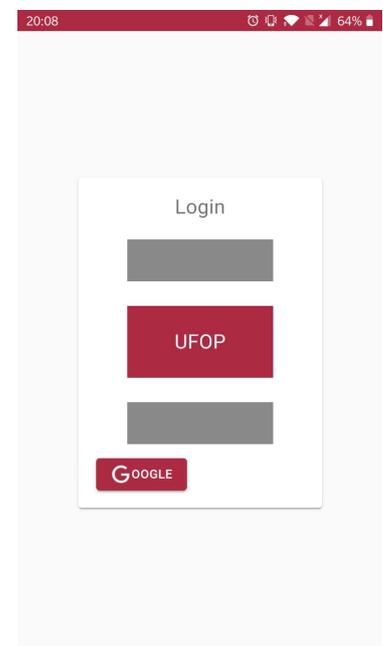


(b) Tela principal de acesso as funcionalidades do sistema.

Figura 16 – Escolha de conta Google e tela tela principal da aplicação.



(a) Visualização no computador.

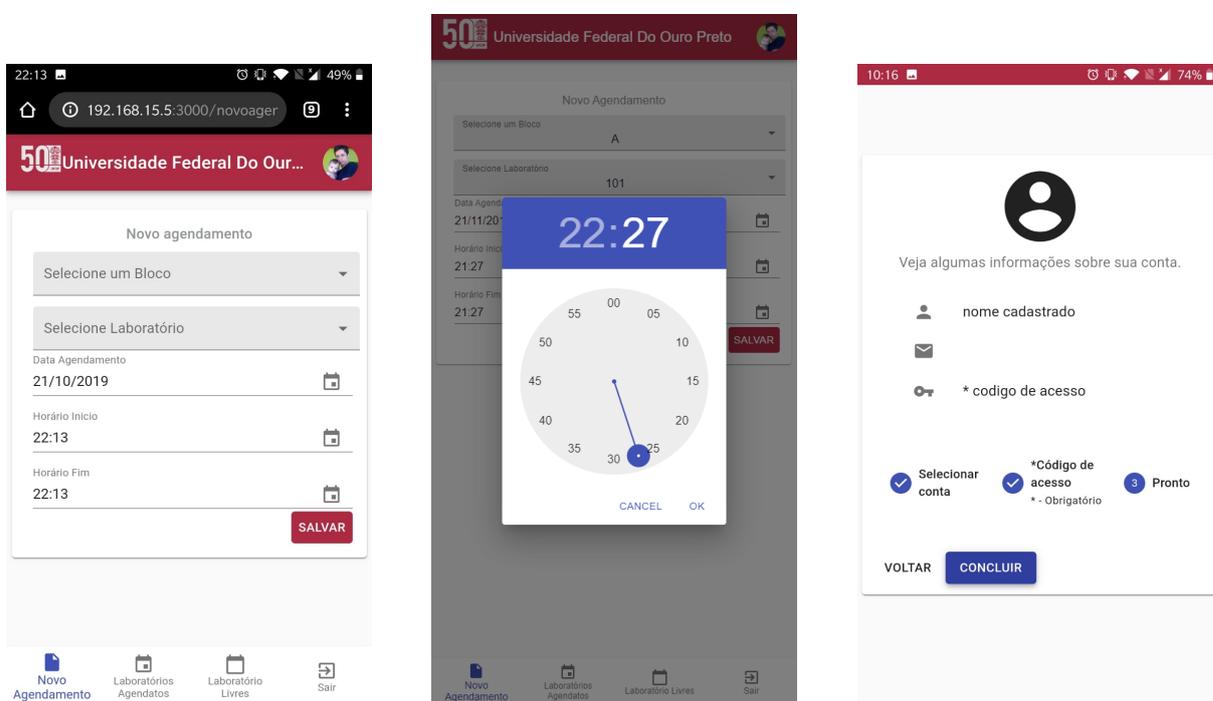


(b) Visualização no dispositivo móvel.

Figura 15 – Tela de login utilizando Google Chrome.

4.1.3 Cadastro de agendamentos

Ao escolher novo agendamento dentre as opções fornecidas na tela principal, é feito redirecionamento para tela de agendamentos (Figura 17a). Na tela é apresentado um menu com os campos definidos para agendamento. Ao clicar no campo de seleção de bloco, é aberta uma lista contendo os blocos cadastrados, em seguida após seleção é feito carregamento dos laboratórios daquele bloco. Ao clicar na região "data de agendamento" é aberto um menu com foco de seleção de data do agendamento, sendo possível realizar outros agendamentos em datas e horários diferentes para o mesmo bloco e laboratório. Por fim tem dois campos de horário início e fim (Figura 17b) para duração do agendamento seguido do botão salvar.



(a) Tela para criar novo agendamento.

(b) Preenchimento com horário do agendamento.

(c) Confirmação de cadastro de agendamento.

Figura 17 – Fluxo de agendamento no sistema web.

4.1.4 Laboratórios agendados

Nesta tela (Figura 18) são apresentados agendamentos do usuário. Agendamentos do usuário, onde a mesma apresenta o laboratório, data e duração do agendamento. Existe a possibilidade realizar o clique em um agendamento "ver" suas opções (Figura 18b). É possível editar um agendamento (Figura 18c) ou excluir.

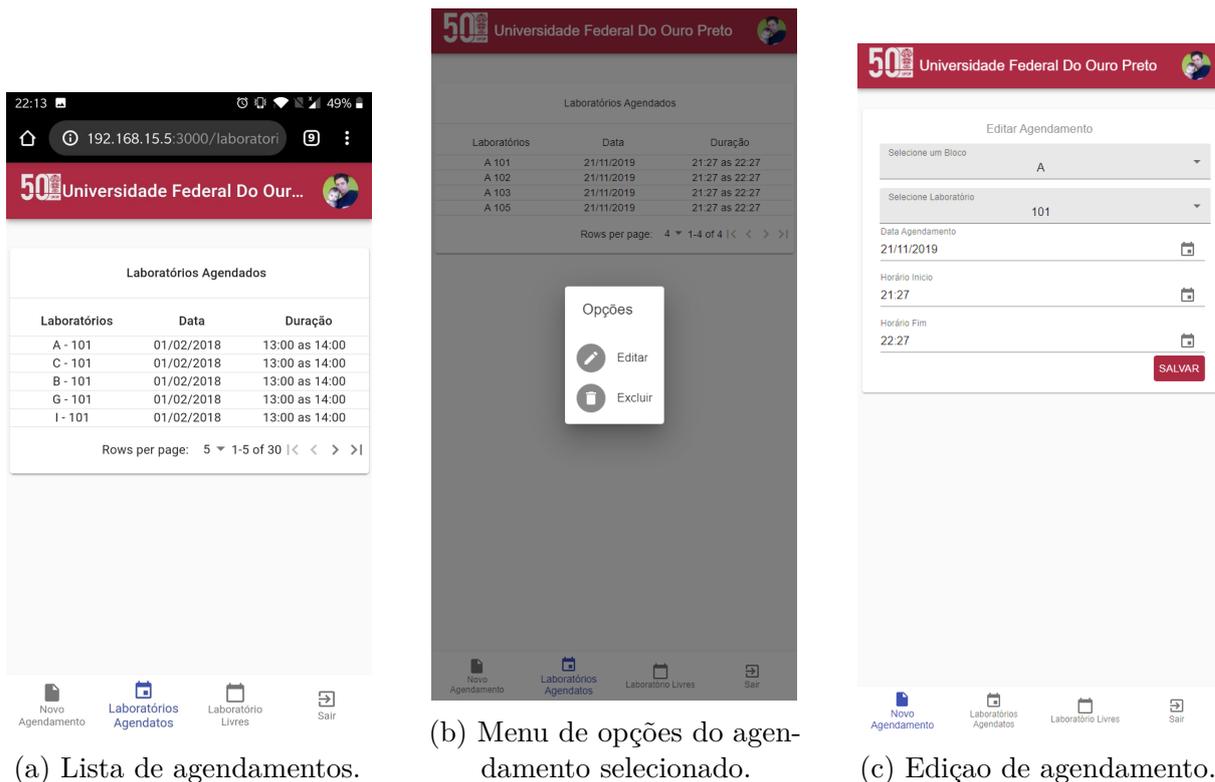
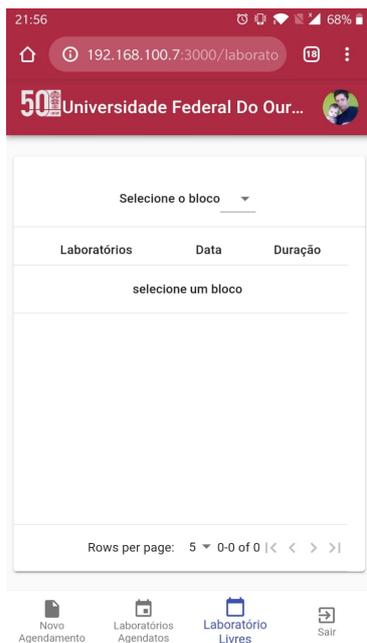


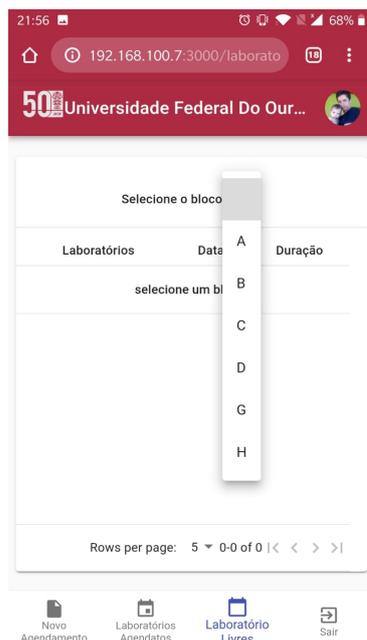
Figura 18 – Laboratórios agendados do usuário.

4.1.5 Laboratórios livres

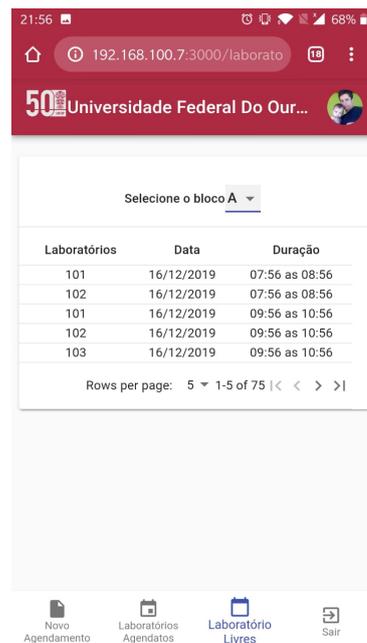
Nesta tela (Figura 19a) é possível observar os laboratórios disponíveis por bloco (Figura 19b), que ao selecionar o bloco é feita uma requisição ao servidor e assim carregado a lista de laboratórios disponíveis (Figura 19c).



(a) Tela de laboratórios disponíveis por seleção de bloco.



(b) Menu de opções de blocos cadastrados.



(c) Lista de laboratórios disponíveis de acordo com bloco selecionado.

Figura 19 – Laboratórios disponíveis ao usuário.

4.1.6 Meu perfil

Nesta tela apresenta informações do usuário (Figura 20).

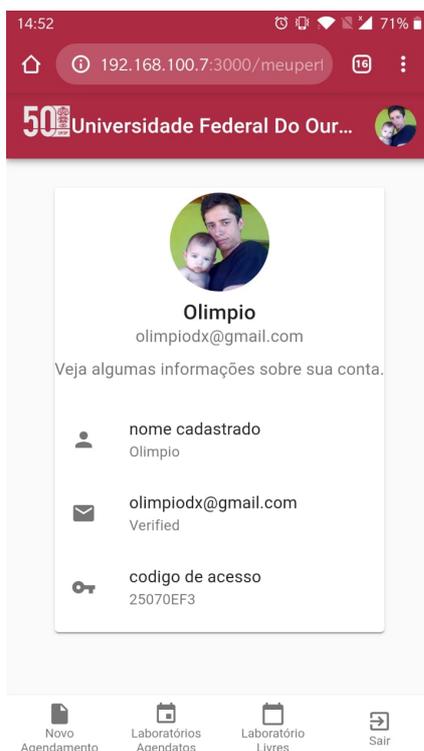


Figura 20 – Tela do perfil do usuário.

4.2 Funcionalidades de sistema IoT

Durante o processo de solução da aplicação, foram realizados testes individuais de cada componente elétrico utilizado, encapsulando as partes até sua conclusão (Figura 21). No processo de teste dos componentes, foram utilizadas bibliotecas de acordo com a interface de comunicação por componente.

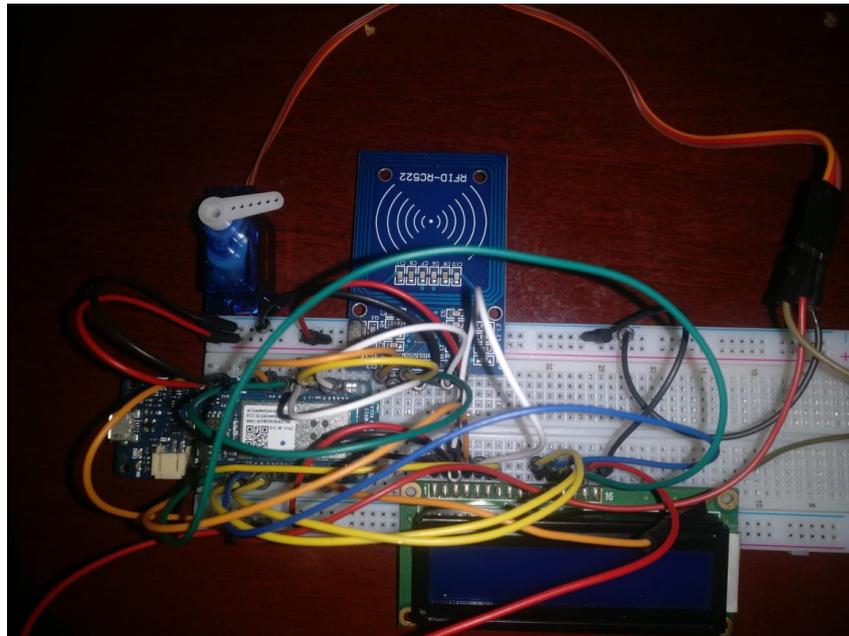


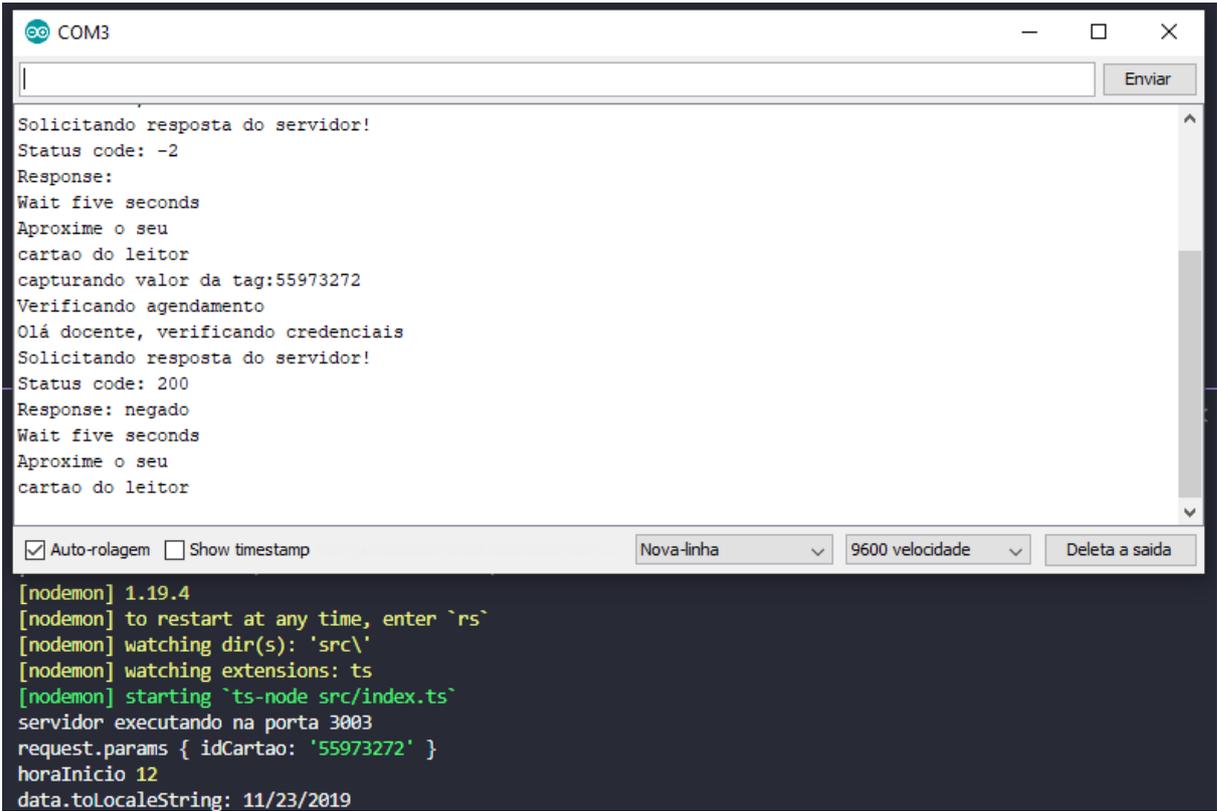
Figura 21 – Protótipo finalizado do sistema IoT.



Figura 22 – Log de inicialização do dispositivo com informações de acesso a rede wifi configurada.

No processo de teste da aplicação, foi realizado o primeiro teste cartão e segundo com chaveiro. Ao ligar o dispositivo, acoplado por uma interface usb no computador com software de desenvolvimento do Arduino, é possível observar log de mensagens usadas desde início (Figura 22) que aplicação é ligada e até cada solicitação de verificação do agendamento. Ao ligar o dispositivo, é visto seu log de tentativa de conexão com rede sem fio, configurada previamente via código no programa (Figura 23). Em sua configuração é adicionado o endereço do servidor e sua rota para envio do código do cartão. Ao passar o

cartão ou chaveiro no sensor é feito processo de leitura de código, e após leitura é informado por meio do LCD, mensagens sobre fluxo de andamento do envio da mensagem ao servidor. Durante tentativa de envio de requisição ao servidor, é esperado uma resposta, caso passe o limite padrão estabelecido pela biblioteca de comunicação do HTTP, é enviada uma mensagem ao LCD, informando ao usuário sobre a falha na conexão. Após recebida resposta do servidor é liberado acesso ao local ou negado, com base nas regras de acesso.



```
COM3
Solicitando resposta do servidor!
Status code: -2
Response:
Wait five seconds
Aproxime o seu
cartao do leitor
capturando valor da tag:55973272
Verificando agendamento
Olá docente, verificando credenciais
Solicitando resposta do servidor!
Status code: 200
Response: negado
Wait five seconds
Aproxime o seu
cartao do leitor

[nodemon] 1.19.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): 'src'
[nodemon] watching extensions: ts
[nodemon] starting `ts-node src/index.ts`
servidor executando na porta 3003
request.params { idCartao: '55973272' }
horaInicio 12
data.toLocaleString: 11/23/2019
```

Figura 23 – Teste de leitura da tag, com envio de código por rede sem fio para verificação de agendamento.

5 Conclusão

Com trabalho desenvolvido pode se concluir, que ao se utilizar diversas tecnologias combinadas na solução do problema proposto, gera-se uma solução que pode atender a Universidade Federal de Ouro Preto, podendo ser usada em dispositivo móvel, computador e *tablet*.

Em relação aos trabalhos futuros pode-se citar:

- Melhoria com um menu para administrador, apresentando informações mais relevantes ao uso da aplicação IoT, com servidor e aplicação web.
- Melhoria na funcionalidade para aplicação. Na utilização de laboratórios agendados, apresenta apenas laboratórios com data vigente em sua lista, podendo adicionar um filtro, apresentando histórico anterior.
- Melhoria no método de acessar o laboratório. Pode-se utilizar outro dispositivo na autenticação do usuário quando for realizada uma tentativa de acesso. Utilizar teclado numérico, sensor de impressão digital ou qualquer outro modelo que possa atender este trabalho.

Em suma, foi possível concluir o objetivo proposto neste trabalho, e as sugestões de trabalhos futuros podem agregar valor ao projeto final. Provendo outras formas de acesso aos laboratórios da Universidade Federal de Ouro Preto.

Referências

- AFIXGRAF. *RFID: Soluções para Identificação Automática de Dados*. [S.l.], 2018. Disponível em: <<http://www.afixgraf.com.br/rfid/>>. Acesso em: 25 dez. 2019. Citado na página 18.
- ARDUINO. *About Us*. [S.l.], 2019. Disponível em: <<https://www.arduino.cc/en/Main/AboutUs>>. Acesso em: 25 dez. 2019. Citado na página 17.
- ARDUINO. *Arduino MKR1000 WIFI*. [S.l.], 2019. Disponível em: <<https://store.arduino.cc/usa/arduino-mkr1000>>. Acesso em: 19 dez. 2019. Citado na página 23.
- ARDUINO. *"Hello World!"*. [S.l.], 2019. Disponível em: <<https://www.arduino.cc/en/Tutorial/HelloWorld>>. Acesso em: 1 jan. 2019. Citado na página 28.
- AUTHENTICATION, F. *Login simples e gratuito em várias plataformas*. [S.l.], 2019. Disponível em: <<https://firebase.google.com/products/auth>>. Acesso em: 25 dez. 2019. Citado na página 22.
- CAPRETZ, V. *Tira Redux, coloca Redux*. [S.l.], 2018. Disponível em: <<https://medium.com/trainingcenter/tira-redux-coloca-redux-eb3c4f1d7db8>>. Acesso em: 25 dez. 2019. Citado na página 19.
- COMPONENTS101. *Arduino MKR1000 Wi-Fi Board*. [S.l.], 2017. Disponível em: <<https://components101.com/microcontrollers/arduino-mkr1000-wi-fi-board>>. Acesso em: 19 dez. 2019. Citado na página 24.
- DATABASE, F. R. *Firestore Realtime Database*. [S.l.], 2019. Disponível em: <<https://firebase.google.com/docs/database>>. Acesso em: 25 dez. 2019. Citado na página 22.
- DESIGN, M. *Material is a design system – backed by open-source code – that helps teams build high-quality digital experiences*. [S.l.], 2019. Disponível em: <<https://material.io/>>. Acesso em: 19 dez. 2019. Citado 2 vezes nas páginas 22 e 23.
- ELETRÔNICA, M. *CARTÕES RFID E O MÓDULO MFRC522*. [S.l.], 2019. Disponível em: <<https://www.moduloeletronica.com.br/blog?single=Cartoes-RFID-e-o-Modulo-MFRC522>>. Acesso em: 19 dez. 2019. Citado na página 26.
- ELETRÔNICOS, B. da Eletrônica componentes. *Display LCD 16x2 (Azul)*. [S.l.], 2019. Disponível em: <<https://www.baudaeletronica.com.br/display-lcd-16x2-azul.html>>. Acesso em: 19 dez. 2019. Citado na página 25.
- FOOTE, K. D. *A INTERNET DAS COSIAS*. Data Topics 2016, 2013. Disponível em: <<https://www.dataversity.net/brief-history-internet-things>>. Acesso em: 19 dez. 2012. Citado na página 13.
- FRANKEL, D. *How to Build a Product Loved by Millions and Get Acquired by Google: The Firebase Story*. [S.l.], 2017. Disponível em: <<https://hackernoon.com/how-to-build-a-product-loved-by-millions-and-get-acquired-by-google-the-firebase-story-82dab4e3e800>>. Acesso em: 25 dez. 2019. Citado na página 19.

- GBUR, F. *Módulo RFID RC522 Mifare com Arduino*. [S.l.], 2017. Disponível em: <<https://portal.vidadesilicio.com.br/modulo-rfid-rc522-mifare/>>. Acesso em: 19 dez. 2019. Citado 2 vezes nas páginas 25 e 26.
- LANDT, J. *The history of RFID*. IEEE Xplore Digital Library, 2005. Disponível em: <<https://ieeexplore.ieee.org/document/1549751>>. Acesso em: 19 dez. 2012. Citado na página 13.
- MARGRANI, E. *A INTERNET DAS COSIAS*. Biblioteca digital FGV, 2018. Disponível em: <<https://bibliotecadigital.fgv.br/dspace/bitstream/handle/10438/23898/A%20internet%20das%20coisas.pdf>>. Acesso em: 19 dez. 2012. Citado na página 13.
- MARUTA, R. *Iniciando com Redux em 9 passos*. [S.l.], 2018. Disponível em: <<https://medium.com/reactbrasil/iniciando-com-redux-c14ca7b7dcf>>. Acesso em: 25 dez. 2019. Citado na página 19.
- MATERIAL-UI. *Componentes React para um desenvolvimento mais rápido e fácil. Construa seu próprio design, ou comece com Material Design*. [S.l.], 2019. Disponível em: <<https://material-ui.com/pt/>>. Acesso em: 19 dez. 2019. Citado 2 vezes nas páginas 22 e 23.
- MEDEIROS, H. *"Hello World!"*. [S.l.], 2013. Disponível em: <<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>>. Acesso em: 1 jan. 2019. Citado na página 33.
- METZ, C. *'Firebase' Does for Apps What Dropbox Did for Docs*. [S.l.], 2012. Disponível em: <<https://www.wired.com/2012/04/firebase>>. Acesso em: 25 dez. 2019. Citado na página 19.
- MUKHERJEE, A. *Security Access Using RFID Reader*. Hackster.io, 2016. Disponível em: <<https://www.hackster.io/Aritro/security-access-using-rfid-reader-f7c746>>. Acesso em: 25 dez. 2019. Citado na página 16.
- NEGROMONTE, E. *O que é Fritzing?* [S.l.], 2019. Disponível em: <<https://sempreupdate.com.br/o-que-e-fritzing>>. Acesso em: 25 dez. 2019. Citado na página 22.
- REACT. *Pensando do jeito React*. [S.l.], 2019. Disponível em: <https://pt-br.reactjs.org/docs/thinking-in-react.html#___gatsby>. Acesso em: 25 dez. 2019. Citado 2 vezes nas páginas 18 e 31.
- REACT. *react/CHANGELOG.md at master*. [S.l.], 2019. Disponível em: <<https://github.com/facebook/react/blob/master/CHANGELOG.md>>. Acesso em: 25 dez. 2019. Citado na página 18.
- REACT. *Virtual DOM e Objetos Internos*. [S.l.], 2019. Disponível em: <<https://pt-br.reactjs.org/docs/faq-internals.html>>. Acesso em: 25 dez. 2019. Citado na página 18.
- REDUX. *Getting Started with Redux*. [S.l.], 2019. Disponível em: <<https://redux.js.org/introduction/getting-started>>. Acesso em: 25 dez. 2019. Citado na página 18.
- ROBERTI, M. *The History of RFID Technology*. RFID Journal, 2005. Disponível em: <<https://www.rfidjournal.com/articles/view?1338>>. Acesso em: 19 dez. 2012. Citado 2 vezes nas páginas 14 e 17.

- SANTOS, B. P. et al. *Internet das Coisas: da Teoria à Prática*. [S.l.], 2016. Disponível em: <<https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>>. Acesso em: 19 dez. 2012. Citado na página 13.
- SERVO, A. *Servo library*. [S.l.], 2019. Disponível em: <<https://www.arduino.cc/en/reference/servo>>. Acesso em: 5 setem. 2019. Citado na página 26.
- SMERKOUS, D. *Arduino Ethernet Rfid card reader*. Hackster.io, 2016. Disponível em: <<https://www.hackster.io/smerkousdavid/arduino-ethernet-rfid-card-reader-1ffdee>>. Acesso em: 25 dez. 2019. Citado na página 16.
- TECHNOLOGY, R. *RFID Based Automatic Door System*. [S.l.], 2016. Disponível em: <<https://www.hackster.io/user8523373/rfid-based-automatic-door-system-7b2065>>. Acesso em: 25 dez. 2019. Citado na página 16.
- THOMSEN, A. *Controlando um LCD 16X2 com Arduino*. [S.l.], 2011. Disponível em: <<https://www.flipeflop.com/blog/controlando-um-lcd-16x2-com-arduino/>>. Acesso em: 19 dez. 2019. Citado na página 24.
- THOMSEN, A. *Micro Servo Motor 9g SG90 com Arduino Uno*. [S.l.], 2019. Disponível em: <<https://www.flipeflop.com/blog/micro-servo-motor-9g-sg90-com-arduino-uno/>>. Acesso em: 4 out. 2019. Citado na página 27.
- TYPESCRIPT. *TypeScript*. [S.l.], 2019. Disponível em: <<https://www.typescriptlang.org/>>. Acesso em: 25 dez. 2019. Citado na página 21.
- VISUAL PARADIGMA. *Scalable Marketing Website*. [S.l.], 2019. Disponível em: <<https://online.visual-paradigm.com/app/diagrams/#diagram:proj=0&type=AzureDiagram&gallery=/repository/6dee5ca3-06d7-4b75-90be-da27b6e415c2.xml&name=Scalable%%20Marketing%%20Website>>. Acesso em: 25 dez. 2019. Citado na página 21.

TERMO DE RESPONSABILIDADE

Eu, **Olimpio Pimenta** declaro que o texto do trabalho de conclusão de curso intitulado “*Desenvolvimento de fechadura inteligente utilizando tecnologia RFID*” é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 11 de março de 2019

Olimpio Pimenta

DECLARAÇÃO DE CONFORMIDADE

Certifico que o(a) aluno(a) **Olimpio Pimenta**, autor do trabalho de conclusão de curso intitulado “*Desenvolvimento de fechadura inteligente utilizando tecnologia RFID*” efetuou as correções sugeridas pela banca examinadora e que estou de acordo com a versão final do trabalho.

João Monlevade, _____ de _____ de _____.

Nome Completo do Orientador