

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE
DO NORTE

ADELSON DE OLIVEIRA CÂMARA DA CRUZ

**SISTEMA INTEGRADO HARDWARE-SOFTWARE-MOBILE-WEB PARA
CONTROLE DE ACESSO COM ESTUDO DE CASO NO IFRN – CAMPUS NOVA
CRUZ**

NOVA CRUZ / RN

2020

ADELSON DE OLIVEIRA CÂMARA DA CRUZ

**SISTEMA INTEGRADO HARDWARE-SOFTWARE-MOBILE-WEB PARA
CONTROLE DE ACESSO COM ESTUDO DE CASO NO IFRN – CAMPUS NOVA
CRUZ**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Dr. Rodrigo Leone Alves.

NOVA CRUZ / RN

2020

ADELSON DE OLIVEIRA CÂMARA DA CRUZ

**SISTEMA INTEGRADO HARDWARE-SOFTWARE-MOBILE-WEB PARA
CONTROLE DE ACESSO COM ESTUDO DE CASO NO IFRN – CAMPUS NOVA
CRUZ**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Aprovado em: ___/___/_____.

BANCA EXAMINADORA

Dr. Rodrigo Leone Alves – Orientador

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Ma. Maria Jane de Queiroz – Examinadora

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Me. José de Anchieta Gomes dos Santos – Examinador

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Dedico esse projeto a minha noiva Gilmara S. Lima, que está caminhando comigo na jornada da vida. Me incentivando e me apoiando desde o início.

RESUMO

A automação vem ganhando destaque pela agilidade e facilidade no dia a dia de uma organização, substituindo atividades manuais por recursos computacionais, tornando o trabalho mais rápido e com menos chances de falhas do que executado por um humano. Esse projeto tem como objetivo desenvolver um sistema para o gerenciamento das solicitações de cada laboratório e seu controle de acesso a todos os usuários, como a elaboração de um mecanismo de abertura e fechamento das portas. Foi desenvolvido os sistemas *Back-end*, onde manipula e gerencia os dados; *Front-end*, no qual possui interfaces gráficas para o gerenciamento dos laboratórios; *Mobile*, responsável por enviar o comando de abrir ou fechar o laboratório e o sistema embarcado no hardware, que visa se comunicar com o *Wireless Fidelity* e o *Message Queuing Telemetry Transport* para controlar a abertura e fechamento do laboratório. Foi desenvolvido também o protótipo do hardware responsável pelo controle do mecanismo da porta. Após o desenvolvimento, foram realizados testes com êxito nos sistemas e protótipo. Com o projeto implantado no Campus, será dispensável o uso da chave dos laboratórios, sendo somente necessário apontar a câmera do celular para o *Quick Response Code* para utilizar o laboratório. Após diversas pesquisas, desenvolvimentos e testes para a concretização do projeto, foi observado que o objetivo de desenvolver um sistema para o gerenciamento das solicitações e o mecanismo de abertura e fechamento das portas foi alcançado.

Palavras-chave: Automação; Controle; Gerenciamento; Laboratório; Sistema.

ABSTRACT

In recent years, automation has been gaining prominence due to the ease in the daily life of a company, making computational resources gain more space, replacing manual resources. Thus, using automated resources, activities become faster and with fewer failures than performed by a human. The main objective of this project is to develop a system for managing requests to use computer labs and controlling access to all users. Back-end systems were developed, where it manipulates and manages the data; Front-end, which has graphical interfaces for laboratory management; Mobile, responsible for sending the command to open or close the laboratory and the system embedded in the hardware to communicate with Wireless Fidelity and Message Queuing Telemetry Transport to control the opening and closing of the laboratory. The hardware prototype responsible for controlling the door mechanism was also developed. After the development, tests were successfully performed on the system and prototype. With the project installed on the Campus, it will be possible to dispense with the use of the laboratory key, being only necessary to point the cell phone camera at the Quick Response Code to use the laboratory. After several research, development and testing for the realization of the project, the aim of developing a system for the management of requests and the opening mechanism and closing of the doors has been reached.

Keys words: Automation; Control; Management; Laboratory; System.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de Caso de Uso	24
Figura 2 – Diagrama Entidade Relacionamento.....	26
Figura 3 – Front-end – Autenticar	28
Figura 4 – Front-end – Perfil Aluno e Professor - Início.....	29
Figura 5 – Front-end – Perfil Aluno e Professor - Início (Laboratório selecionado)	30
Figura 6 – Front-end – Perfil Aluno e Professor - Reservar laboratório	31
Figura 7 – Front-end – Perfil Aluno e Professor - Minhas reservas (Confirmada).....	32
Figura 8 – Front-end – Perfil Aluno e Professor - Minhas reservas (Usuários).....	32
Figura 9 – Front-end – Perfil Aluno e Professor - Minhas reservas (Horários)	33
Figura 10 – Front-end – Perfil Aluno e Professor - Minhas reservas (Não confirmada)	33
Figura 11 – Front-end – Perfil Gerenciador - Início.....	34
Figura 12 – Front-end – Perfil Gerenciador - Início (Laboratório selecionado).....	35
Figura 13 – Front-end – Perfil Gerenciador - Confirmar reserva de laboratórios.....	35
Figura 14 – Front-end – Perfil Gerenciador - Reserva confirmar	36
Figura 15 – Front-end – Perfil Gerenciador - Reservas	37
Figura 16 – Front-end – Perfil Gerenciador - Reservas (Criador).....	37
Figura 17 – Front-end – Perfil Gerenciador - Detalhes da reserva.....	38
Figura 18 – Front-end – Perfil Gerenciador - Alunos	39
Figura 19 – Front-end – Perfil Gerenciador - Professores	39
Figura 20 – Front-end – Perfil Gerenciador - Gerenciadores.....	40
Figura 21 – Front-end – Perfil Gerenciador - Laboratórios.....	40
Figura 22 – Front-end – Perfil Gerenciador - Laboratórios (Cadastrar laboratório).....	41

Figura 23 – Front-end – Perfil Gerenciador - Laboratórios (Atualizar laboratório)	42
Figura 24 – Front-end – Perfil Gerenciador - Laboratórios (Confirmar remoção)	42
Figura 25 – Mobile – Autenticação	43
Figura 26 – Mobile – Início	44
Figura 27 – Mobile – Início (Laboratório selecionado I)	45
Figura 28 – Mobile – Início (Laboratório selecionado II)	46
Figura 29 – Mobile – Barra de Navegação Lateral	47
Figura 30 – Mobile – Abrir laboratório	48
Figura 31 – Mobile – Abrir laboratório (Usuários)	49
Figura 32 – Mobile – Abrir laboratório (Horários)	50
Figura 33 – Mobile – Escaneamento do QR Code	51
Figura 34 – Mobile – Reservar laboratório	52
Figura 35 – Mobile – Reservar laboratório (Laboratório)	53
Figura 36 – Mobile – Reservar laboratório (Alunos)	54
Figura 37 – Mobile – Reservar laboratório (Professores)	55
Figura 38 – Mobile – Reservar laboratório (Selecionar data)	56
Figura 39 – Mobile – Reservar laboratório (Horários)	57
Figura 40 – Mobile – Minhas reservas (Confirmada)	58
Figura 41 – Mobile – Minhas reservas (Usuários)	59
Figura 42 – Mobile – Minhas reservas (Horários)	60
Figura 43 – Mobile – Minhas reservas (Não confirmada)	61
Figura 44 – Hardware – WeMos D1	63
Figura 45 – Hardware – Servo motor	64
Figura 46 – Hardware – Bateria 9V	65

Figura 47 – Hardware – Representação da fechadura	66
Figura 48 – Comunicação MQTT	68
Figura 49 – Teste do protótipo	69

LISTA DE ABREVIATURAS E SIGLAS

AES	<i>Advanced Encryption Standard</i>
API	<i>Application Programming Interface</i>
BPM	<i>Business Process Management</i>
CLI	<i>Command-Line Interface</i>
CSS	<i>Cascading Style Sheets</i>
DC	<i>Direct Current</i>
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
IBM	<i>International Business Machines Corporation</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IFRN	Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
IOT	<i>Internet of Things</i>
JWT	<i>JSON Web Token</i>
MDF	<i>Medium Density Fiberboard</i>
MDP	<i>Medium Density Particle</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
OASIS	<i>Organization for the Advancement of Structured Information Standards</i>
PHP	<i>Hypertext Preprocessor</i>
QR Code	<i>Quick Response Code</i>
REST	<i>Representational State Transfer</i>
SDK	<i>Software Development Kit</i>

SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	<i>Structured Query Language</i>
SUAP	Sistema Unificado de Administração Pública
URL	<i>Uniform Resource Locator</i>
V	<i>Volts</i>
Wi-Fi	<i>Wireless Fidelity</i>

SUMÁRIO

1. INTRODUÇÃO	13
1.1. PROBLEMÁTICA	13
1.2. JUSTIFICATIVA	14
1.3. OBJETIVOS	14
1.3.1. Objetivo Geral	15
1.3.2. Objetivos específicos	15
1.4. TECNOLOGIAS UTILIZADAS.....	15
1.4.1. Laravel	15
1.4.2. Vue.js	16
1.4.3. Flutter	16
1.4.4. Microcontrolador ESP8266	17
1.4.5. QR Code	17
1.4.6. Protocolo MQTT	17
2. REFERENCIAL TEÓRICO	19
2.1. AUTOMAÇÃO	19
2.2. AUTOMAÇÃO DE PROCESSOS.....	19
2.3. INTERNET DAS COISAS.....	19
2.4. ESP8266 EMPREGADO EM INTERNET DAS COISA	20
2.5. <i>FRAMEWORKS</i>	20
2.6. PROTOCOLO MQTT	21
3. METODOLOGIA	22
3.1. CASO DE USO	23

3.2.	BANCO DE DADOS	24
3.3.	SISTEMAS	26
3.3.1.	<i>Back-end</i>	27
3.3.2.	<i>Front-end</i>	28
3.3.3.	<i>Mobile</i>	42
3.3.4.	Sistema Embarcado	61
3.4.	HARDWARE	62
3.5.	COMUNICAÇÃO MQTT	66
3.6.	TESTES	68
4.	RESULTADOS E DISCUSSÕES	70
5.	CONSIDERAÇÕES FINAIS	71
REFERÊNCIAS	72
APÊNDICE A – Rotas da API	75
APÊNDICE B – Código do sistema embarcado	77
APÊNDICE C – Imagem do hardware completo	82

1. INTRODUÇÃO

A automação vem ganhando destaque pela agilidade e facilidade no dia a dia de uma organização. Segundo PRNewswire (2019), “O Índice de Automação do Mercado Brasileiro chega ao terceiro ano de publicação e constata que o nível de automação nas empresas aumentou 4% de 2018 para 2019”. Dentre as categorias está a indústria de pequeno porte que aumentou seu índice de 0,273 em 2018 para 0,275 em 2019, onde o índice é mensurado em um intervalo de 0 a 1, sendo 0 a ausência de automação e 1 a representação da automação plena (PRNEWSWIRE, 2019).

Na automação empresarial, as atividades manuais são substituídas por recursos computacionais (DOCUSIGN, 2018), assim, tornando o trabalho mais rápido e com menos chances de falhas do que executado por um humano. Graças a evolução crescente da tecnologia, com o aprimoramento de inteligências artificiais e robótica cada vez mais sofisticada, o maquinário está mais autossuficiente e com softwares mais robustos para realizar as atividades.

Encontra-se também a automatização nas organizações, que diferente da automação onde o trabalho é executado inteiramente por máquinas, a automatização tem a interferência humana. A automatização nos processos é muito usada por trazer agilidade na execução de tarefas, os funcionários ficam mais concentrados com a estratégia que vai seguir para ter o melhor resultado do que como irá fazer a tarefa (SCHULTZ, 2020). De acordo com Schultz (2020), a “Automatização de processos se refere ao uso de equipamentos e sistemas ao executar tarefas repetitivas e operacionais. Assim sobra para o profissional mais tempo para pensar estrategicamente, utilizando suas competências no crescimento dos resultados da empresa”.

Há inúmeras vantagens na utilização de automatização de processos como aumento na produtividade da equipe, padronização dos produtos e serviços, redução de custos. Com a implementação da automatização de processos a organização tem um ganho de desempenho e produtividade pois os processos se tornam mais ágeis, facilitados por sistemas que abstrai sua complexidade.

1.1. PROBLEMÁTICA

No Campus Nova Cruz do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN) o acesso aos laboratórios se dá de modo convencional, utilizando chaves comuns identificadas pelo número do laboratório para abrir a porta ao entrar e fechar

obrigatoriamente ao sair. Com isso, é necessário um gerenciamento do acesso de cada laboratório no que diz respeito ao horário e quais pessoas terão a chave, visto que alunos e professores podem ter acesso aos laboratórios.

Esse gerenciamento atualmente é realizado em papel, no qual são escritos o nome, data, hora e o laboratório que se deseja utilizar. Esses dados são fornecidos na retirada e na devolução da chave. Esse método é potencialmente falho, pois pode haver extravio da folha onde contém os dados ou esquecimento da assinatura por parte do requerente da chave.

Houve ocasiões no qual a pessoa utilizou o laboratório, fechou e esqueceu de devolver a chave ao local designado, impossibilitando que outra pessoa usufrísse do laboratório. O gerenciador dos laboratórios não tem mecanismos para determinar se o utilizador está no laboratório solicitado e desfrutando do local, impossibilitando assim que essas ocorrências deixem de existir.

1.2. JUSTIFICATIVA

Por meio deste projeto, espera-se sanar os problemas de gerenciamento manual, através do uso de um mecanismo físico para a automação da abertura e fechamento de portas, ligado a um sistema com o objetivo de automatizar o gerenciamento dos laboratórios, horários e utilizadores. O controle de abertura dos laboratórios será realizado por leitura de *Quick Response Code* (QR Code), sendo dispensável o uso da chave cotidianamente, mas não descartando seu uso, tornando-a uma redundância para utilização do laboratório.

Todo o gerenciamento será realizado pelo sistema, tornando desnecessário o uso de papel, dificultando assim a perda dos dados e automatizando a assinatura para aquisição da chave, sendo somente necessária a aprovação da solicitação pelo gerenciador dos laboratórios.

Com tal gerenciamento, o próprio sistema identificará o horário de cada usuário no laboratório, impossibilitando que outro usuário fique impedido de usar, como também, evitando conflitos de utilização.

1.3. OBJETIVOS

Para um melhor direcionamento do trabalho, foi elaborado o objetivo geral afim de ter uma visão ampla do projeto e os objetivos específicos para então nortear cada etapa do projeto.

1.3.1. Objetivo Geral

Desenvolver um sistema para o gerenciamento das solicitações diárias de uso de cada laboratório e seu controle de acesso a todos os usuários, como também a elaboração de um protótipo do mecanismo de abertura e fechamento das portas.

1.3.2. Objetivos específicos

- Desenvolver aplicativo Android para utilização dos usuários;
- Desenvolver sistema web para gerenciamento dos horários;
- Desenvolver sistema para controle e manipulação dos dados;
- Desenvolver sistema embarcado para o hardware;
- Confeccionar protótipo do hardware para controle dos laboratórios.

1.4. TECNOLOGIAS UTILIZADAS

Para a realização desse projeto foram utilizados diversos recursos como softwares, serviços e hardwares, onde os mais relevantes estão listados nos tópicos a seguir, com uma breve descrição do seu conceito.

1.4.1. Laravel

Laravel é um *framework* com linguagem *Hypertext Preprocessor* (PHP) de código aberto para aplicações web, desenvolvido por Taylor B. Otwell, com sua primeira versão em 2011, e atualmente um dos *frameworks* mais utilizados.

Segundo Andrade (2019), o Laravel

[...] é um facilitador no desenvolvimento de diversas aplicações [...], pois de forma mais básica, é um conjunto de bibliotecas utilizadas para criar uma base onde as aplicações são construídas, um otimizador de recursos. Tem como principal objetivo resolver problemas recorrentes com uma abordagem mais genérica. Ele permite ao desenvolvedor focar nos “problemas” da aplicação, não na arquitetura e configurações.

Ele utiliza o Composer¹ para gerenciar as dependências do projeto e de acordo com D3T ([2019?]) o Laravel

[...] faz uso da arquitetura de desenvolvimento MVC [...] e do padrão de desenvolvimento PSR-2 que tem como foco a escrita de código limpo e com boa legibilidade, além disso o *framework* tem como objetivo aumentar a velocidade de codificação, sem esquecer características importantes como a segurança e performance da aplicação.

1.4.2. Vue.js

Vue.js é um *framework* JavaScript de código aberto, lançado em 2014 por Evan You. “Com versatilidade, modularização e facilidade de aprendizado, o *framework* vem se tornando a escolha número 1 entre profissionais da área que procuram criar *websites* profissionais e de alto desempenho” (PICOLLO, 2020).

As aplicações desenvolvidas em Vue.js são constituídas de arquivos contendo sintaxes *Hypertext Markup Language* (HTML), *Cascading Style Sheets* (CSS) e JavaScript em um mesmo local, facilitando o isolamento e a manutenção. Cada componente possui um escopo isolado dos demais, tanto em lógica quanto em estilo (PICOLLO, 2020).

O Vue.js possui uma ferramenta de linha de comando denominada *Vue Command-Line Interface* (CLI) criada pela comunidade para facilitar ainda mais a inicialização de uma nova aplicação. Ela gera uma estrutura específica na criação de um novo projeto, evitando o desenvolvimento do zero.

1.4.3. Flutter

Flutter é um *Software Development Kit* (SDK) de código aberto criado pela Google em 2017. Com ele é possível desenvolver aplicativos nativos para Android, iOS, Web e Desktop, utilizando o mesmo código.

De acordo com Andrade (2020) grandes empresas como: Google, Alibaba, Groupon e Nubank usam o Flutter. Um dos grandes motivos é usar um único código para diversas plataformas e sua execução ser nativa, ou seja, a aplicação não necessita de outros artifícios para executar os recursos do dispositivo onde está instalado.

¹ Uma ferramenta para gerenciar as dependências em PHP. Ele permite a declaração das bibliotecas necessárias para o projeto e as gerencia na instalação, atualização e remoção (COMPOSER, [201-?]).

A linguagem utilizada para codificar as aplicações é o Dart. Criada pela Google em 2011, com a missão inicial de substituir a linguagem JavaScript. Porém, essa missão não se concretizou e com o sucesso do Flutter tendo o Dart como sua linguagem principal, voltou a se popularizar em meio aos desenvolvedores (GUEDES, 2019).

1.4.4. Microcontrolador ESP8266

O ESP8266 é um microcontrolador produzido pela Espressif Systems, com seu lançamento em 2014. O seu grande diferencial é a possibilidade de comunicação *Wireless Fidelity* (Wi-Fi) e “[...] por esse motivo ele é largamente utilizado como módulo WiFi para outros microcontroladores, como o Arduíno, por exemplo, apesar de possuir um processador próprio [...]” (OLIVEIRA, 2017, p. 3).

O microcontrolador é encapsulado em diferentes plataformas como ESP-01, WeMos D1 e NodeMCU. Cada plataforma usando um determinado número de portas ou tamanhos físicos distintos. Muito usado no mundo da *Internet of Things* (IOT) por possuir um tamanho reduzido, baixo custo, possibilidade de comunicação WI-FI e independência, pois tem seu próprio processador.

1.4.5. QR Code

Criado em 1994 pela Denso-Wave, o código QR é a evolução do código de barras. Consiste em um gráfico 2D bidirecional, ao contrário do código de barra que possui um sentido horizontal (ANDRION, 2019).

O QR *Code* é amplamente usado hoje em dia pela sua facilidade, sendo somente necessário apontar a câmera do celular para o código. Atividades como pagamento de contas, passagem de metrô, detalhamento de propagandas, usam esse meio por ser de fácil acesso e simples de usar.

1.4.6. Protocolo MQTT

De acordo com EngProcess (2018) o *Message Queuing Telemetry Transport* (MQTT)

[...] é um protocolo de mensagens destinado a sensores e pequenos dispositivos móveis. Seu principal uso é fazer as máquinas trocarem informações, modalidade de

comunicação conhecida como *Machine-to-Machine* (M2M, ou em português, de máquina para máquina).

Desenvolvido pela *International Business Machines Corporation* (IBM) no final dos anos 90, com a finalidade de conectar sensores de satélites ou pipelines de petróleo (ENGPROCESS, 2018). Seu funcionamento é bastante simples, o publicador envia a mensagem com um tópico (identificador da mensagem) para o *broker* (servidor MQTT que recebe e envia as mensagens), então ele envia para todos os assinantes desse tópico (podendo ser milhares de dispositivos).

2. REFERENCIAL TEÓRICO

Nos próximos tópicos são apresentados conceitos relacionados ao desenvolvimento do projeto e uma breve descrição.

2.1. AUTOMAÇÃO

Segundo o site Significados (2016) “A automação é um sistema que faz uso de técnicas computadorizadas ou mecânicas com o objetivo de dinamizar e otimizar todos os processos produtivos dos mais diversos setores da economia”. Assim, agilizando os processos, aumentando a produtividade da equipe, melhorando as condições de trabalho e trazendo segurança aos operadores que realizam trabalhos considerados perigosos.

Na era atual, a automação chegou a seu ápice, a partir dos avanços da tecnologia, como o Wi-Fi, robótica, nuvem, internet das coisas, dentre outros. Assim, hoje em dia ela se baseia na projeção e implantação de sistemas *cyber*-físicos. Ou seja, se baseia em sistemas que se comunicam entre si, controlando todos os processos de uma indústria (SOUZA, 2020).

A utilização da automação em uma organização traz benefícios econômicos, pois a automação proporciona o desenvolvimento de produtos mais eficientes e com alta qualidade, visto que as máquinas mantêm padrões altos de produção e falham menos (SOUZA, 2020).

2.2. AUTOMAÇÃO DE PROCESSOS

De acordo com Ribeiro (2016) a automação de processos é uma técnica de *Business Process Management* (BPM) que busca, por meio da tecnologia, a integração de sistemas para aprimorar o controle e o andamento do fluxo de trabalho, substituindo atividades manuais por automatizadas.

Uma das vantagens na utilização da automação de processos é a redução nas despesas com funcionários, pois com a automação, é possível ter uma equipe reduzida, economizando nos gastos ao eliminar a margem de erro e o retrabalho. Outra vantagem é a qualidade de produtos e serviços, visto que com a utilização do processo é possível manter o padrão, sendo fundamental para o sucesso no mercado (CLICK, 2020).

2.3. INTERNET DAS COISAS

Conforme o site PROOF ([201-?]) descreve, a internet das coisas é semelhante a “[...] um grande sistema nervoso que possibilita a troca de informações entre dois ou mais pontos”. Ela é a maneira como dispositivos se comunicam e estão conectados uns com os outros.

Nos últimos anos, a IoT se tornou uma das tecnologias mais importantes do século XXI. Agora que podemos conectar objetos do dia a dia: eletrodomésticos, carros, termostatos, babás eletrônicas, à internet por meio de dispositivos incorporados, é possível uma comunicação perfeita entre pessoas, processos e coisas (ORACLE, 2020).

Os dispositivos na IOT se comunicam entre si para proporcionar conforto, produtividade, informação e praticidade no dia a dia. Esses dispositivos podem ser relógios, geladeiras, carros, máquinas de lavar, computadores, smartphones, aparelhos para monitoramento de saúde ou até vagas de estacionamento (PROOF, [201-?]).

Para obtenção das informações, os dispositivos, juntamente com sistemas automatizados, cruzam dados de usuários e outros dispositivos para gerar informações e ações, como informar o tempo do percurso até o trabalho e desligar as lâmpadas da casa automaticamente ao sair sem que o usuário informe qualquer comando.

2.4. ESP8266 EMPREGADO EM INTERNET DAS COISA

O ESP8266 é amplamente utilizado em aplicações IOT, como domótica, redes *mesh*, controle sem fio industrial, câmeras IP, redes de sensores e dispositivos vestíveis por possuir baixíssimo custo e um tamanho muito reduzido (ASSUNÇÃO, 2016).

O diferencial do ESP8266 para outros microcontroladores é a possibilidade de comunicação Wi-Fi, que é a peça-chave para a comunicação IOT. Possuindo um processador próprio, esse dispositivo é capaz de controlar sensores, relés, atuadores, sem o uso de outro microcontrolador.

2.5. FRAMEWORKS

Segundo Revelo (2020) o *framework* é uma ferramenta que busca oferecer determinadas funcionalidades prontas aos desenvolvedores, servindo de base para novos projetos e gerando mais produtividade e lucratividade ao economizar tempo e reduzir os custos.

Na criação dos *frameworks*, as funcionalidades são desenvolvidas de forma genérica para que um *framework* seja capaz de abranger circunstâncias variadas em cada projeto. “Suas

funções são adequadas para determinadas tarefas repetitivas e mais simples, que geralmente são comuns em softwares de diferentes naturezas” (NOLETO, 2020), economizando tempo e aumentando a produtividade no projeto.

2.6. PROTOCOLO MQTT

De acordo com o site MQTT (2020), o MQTT é um protocolo de mensagens padrão criado pela *Organization for the Advancement of Structured Information Standards* (OASIS) para a IOT. Bastante usado hoje em uma ampla variedade de indústrias como automotiva, manufatura, telecomunicações e petróleo e gás.

Segundo Yuan (2017) “O MQTT é um protocolo de ligação que especifica como os bytes de dados são organizados e transmitidos pela rede TCP/IP”. Possuindo mensagens assíncronas, eficiente para redes de largura da banda limitada e de alta latência, como também hardwares com pouco desempenho. “Sua flexibilidade possibilita o suporte a diversos cenários de aplicativo para dispositivos e serviços de IoT” (YUAN, 2017).

3. METODOLOGIA

Diante dos objetivos estabelecidos para o projeto, foi desenvolvido cada componente para seu respectivo papel onde em conjunto concretiza o objetivo geral. Neste tópico será dada uma breve descrição de cada elemento utilizado.

O *framework* Laravel foi utilizado para a codificação do sistema *Back-end*, que tem a função de uma *Application Programming Interface* (API). Essa API não possui interface gráfica, sendo responsável pela manipulação dos dados dos demais sistemas, gerenciando e armazenando-os no banco de dados. Ela é a ponte para o serviço MQTT, onde faz a comunicação do dispositivo Android com o hardware que se encontra na porta do laboratório.

Para o sistema *Front-end*, foi aplicado o uso do Vue.js, que consiste na interface gráfica da aplicação web, podendo ser utilizada tanto por desktop quanto por dispositivos móveis, pois é responsiva. Ele consome e fornece dados a API para a realização das funcionalidades, podendo ter acesso por alunos, professores e gerenciadores, onde o gerenciador tem acesso a todas as funcionalidades, enquanto os alunos e professores possuem acesso limitado a algumas funcionalidades.

No sistema para dispositivos móveis utilizou-se o Flutter. Mesmo sendo possível o desenvolvimento para diversas plataformas, o sistema para dispositivos móveis foi criado somente para sistema Android. Ele é responsável por abrir os laboratórios por meio da leitura do QR Code. Possui também interface gráfica com funções contidas no sistema citado anteriormente, como: realizar a solicitação de horários para um determinado laboratório e visualizar as reservas cadastradas.

Para a elaboração do código empregado ao hardware, utilizou-se da *Integrated Development Environment* (IDE) Arduino, onde foi possível codificar para o microcontrolador ESP8266. Com a utilização deste componente, um servo motor e uma bateria de 9 Volts (V), constituiu-se o protótipo do projeto.

Utilizou-se também da leitura de QR Code pelo dispositivo Android, onde o código se encontra na porta do laboratório para informar o seu número e facilitar a identificação no momento de abri-lo.

A fim de realizar a comunicação via protocolo MQTT, operou-se com o serviço on-line e gratuito CloudMQTT disponibilizado no domínio (<https://www.cloudmqtt.com/>) no qual é

possível criar um *broker* e então efetuar a comunicação. Nele são disponibilizados todos os dados para configurar os dispositivos que irão se comunicar entre si.

O MySQL foi o Sistema de Gerenciamento de Banco de Dados (SGBD) utilizado para o armazenamento dos dados, compondo todas as tabelas² necessárias para a utilização do sistema. Há um único banco de dados em todo o projeto e este encontra-se no *Back-end*, pois é onde se realiza toda a manipulação dos dados.

3.1. CASO DE USO

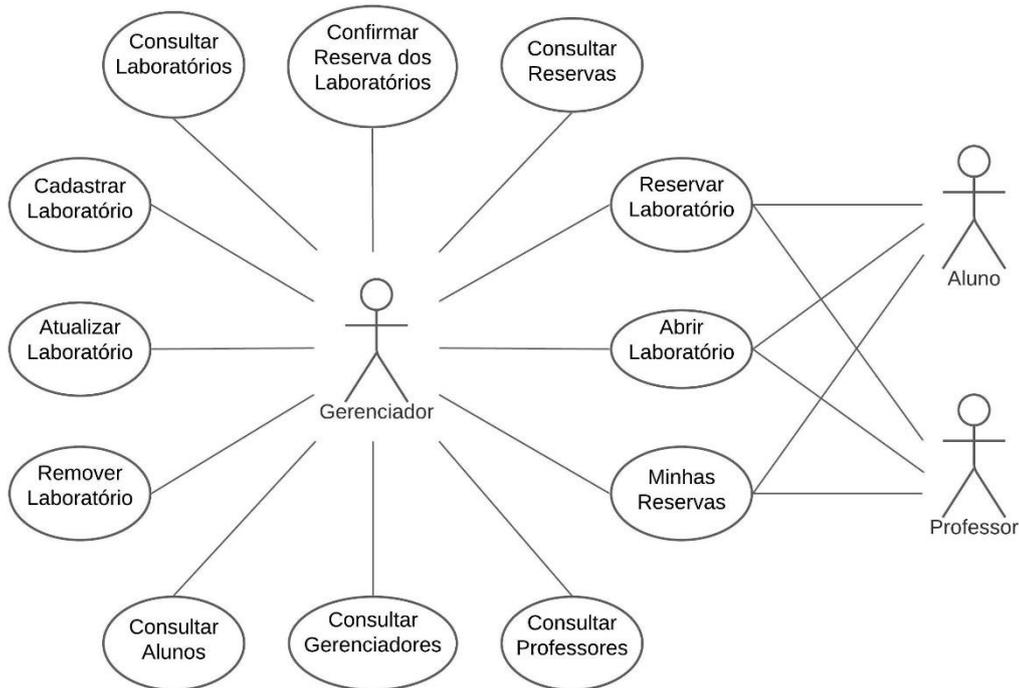
A Figura 1 ilustra o diagrama de caso de uso do sistema, com seus atores **Gerenciador**, **Aluno** e **Professor** e funcionalidades relacionadas a cada usuário. Foi desenvolvido o caso de uso para obter uma visão das funcionalidades do programa a cada usuário.

O **Gerenciador** contém as funcionalidades Consultar Reservas, Consultar Laboratórios, Consultar Gerenciadores, Consultar Professores e Consultar Alunos, onde é possível listar todos os dados cadastrados de cada consulta; Abrir Laboratório, que é capaz de abrir os laboratórios reservados para o dia; Cadastrar Laboratório, Atualizar Laboratório e Remover Laboratório, em que é possível cadastrar, atualizar e remover os laboratórios; Confirmar Reserva dos Laboratórios, em que é possível selecionar quais horários devem ser atribuídos a cada reserva; Reservar Laboratório, onde está disponível um formulário para realizar a solicitação e Minhas Reservas, que exibe a lista de reservas realizadas pelo usuário.

Os atores **Aluno** e **Professor** possuem algumas das funcionalidades atribuídas ao ator **Gerenciador**, como Reservar Laboratório, Abrir Laboratório e Minhas reservas.

² Contêm os dados de um determinado objeto em um banco de dados, organizados em forma de planilha, onde cada linha representa um registro e cada coluna representa um campo do objeto.

Figura 1 – Diagrama de Caso de Uso



Fonte: Elaborado pelo autor, 2020.

3.2. BANCO DE DADOS

Para a elaboração do banco de dados, foi utilizado o MySQL, um dos SGBD mais utilizados, e possui acesso nativo no Laravel. Ele é relacional³ e de código aberto, atualmente pertencente a Oracle e utiliza a linguagem de programação *Structured Query Language* (SQL) (PISA, 2012). Toda a estrutura do banco de dados está ilustrada na Figura 2.

O banco de dados constitui-se de 10 tabelas: *migrations*, *usuario*, *aluno*, *professor*, *gerenciador*, *reserva*, *laboratorio*, *horario*, *reserva_usuario* e *reserva_horario*. Abaixo será detalhada cada tabela e seus relacionamentos.

Os campos *created_at* e *updated_at* inclusos nas tabelas *usuario*, *aluno*, *professor*, *gerenciador*, *reserva*, *laboratorio* e *horario* são criados pelo próprio *framework*, onde *created_at* armazena a data e hora da criação do dado e *updated_at* armazena a data e hora em que o dado é atualizado. O campo *id*, presente em todas as tabelas, é um identificador incremental de cada dado cadastrado nas tabelas.

³ Se trata de um banco de dados que organiza os dados em tabelas, podendo se relacionarem umas com as outras.

O Laravel cria uma tabela denominada *migrations*, onde realiza a gerência de atualizações de todas as tabelas presentes, fazendo com que o banco tenha um controle de versão das tabelas. No cadastro de uma tabela, é atribuído ao campo *migration* o nome que referencia a tabela e ao campo *batch* o número 1. A cada atualização dessa tabela o número é acrescido de mais 1, criando uma sequência de atualizações e com isso podendo regredir para outra atualização caso necessário.

A tabela **usuario** contém os campos matricula, *password* e imagem. Onde matricula e *password* são fornecidos no momento da autenticação nos sistemas *Front-end* e *Mobile*, caso a autenticação seja concedida e os dados sejam armazenados junto com o campo imagem onde consta o endereço da imagem do usuário vinda do SUAP. Essa tabela possui o relacionamento (um para muitos) com as tabelas aluno, professor, gerenciador e reserva.

As tabelas **aluno**, **professor** e **gerenciador** possuem os campos nome, nome_usual e usuario_id. Os dados dos campos nome e nome_usual são disponibilizados pelo retorno da autenticação do SUAP e o usuario_id armazena o identificador do usuário ao qual esses dados pertencem. Todas as tabelas possuem relacionamento (muitos para um) com a tabela usuario.

A tabela **reserva** contém os campos data, confirmada, criador_id e laboratorio_id. O campo data armazena a data para utilização do laboratório; o campo confirmada indica se a reserva já foi confirmada pelo gerenciador; o campo criador_id serve para referenciar a qual usuário a reserva pertence e o campo laboratorio_id indica o laboratório que foi selecionado. Essa tabela contém os relacionamentos (muitos para um) com as tabelas usuario e laboratorio e (muitos para muitos) com usuario e horario.

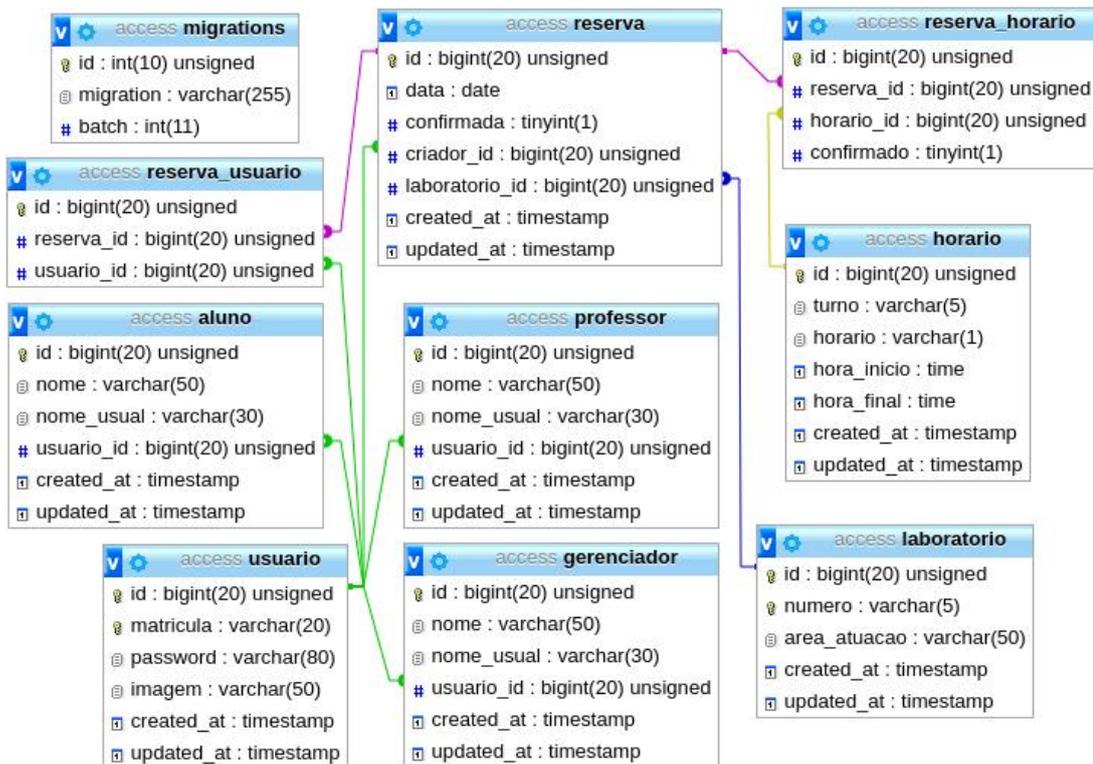
A tabela **laboratorio** possui os campos numero e area_atuacao. O numero representa a configuração presente no IFRN para identificação dos laboratórios e a area_atuacao informa qual a área que o laboratório pertence como por exemplo: Informática, Química, Matemática. Possui o relacionamento (um para muitos) com a tabela reserva.

A tabela **horario** contém os campos turno, horario, hora_inicio, hora_fim. O campo turno representa os turnos manhã, tarde ou noite, o campo horario corresponde a numeração sequencial começando de 1 até o final dos horários de cada turno, o campo hora_inicio indica qual a hora que se inicia a reserva e o hora_fim indica a hora do seu término. Essa tabela é a única que possui os dados previamente cadastrados. Ela possui o relacionamento (muitos para muitos) com a tabela reserva.

A tabela **reserva_usuario** contém os campos `reserva_id` e `usuario_id`. O `reserva_id` armazena a identificação da tabela `reserva` e o `usuario_id`, a da tabela `usuario`. Essa tabela é criada devido ao relacionamento (muitos para muitos) das tabelas `reserva` e `usuario`, armazenando seus identificadores para fazer a relação.

A tabela **reserva_horario** possui os campos `reserva_id`, `horario_id` e `confirmado`. O campo `reserva_id` representa a referência da tabela `reserva` e `horario_id` da tabela `horario`, o campo `confirmado` indica se o horário de uma determinada reserva foi confirmado para que o usuário possa usar o laboratório. A tabela é criada por causa do relacionamento (muitos para muitos) entre as tabelas `reserva` e `horario`.

Figura 2 – Diagrama Entidade Relacionamento



Fonte: Elaborado pelo autor, 2020.

3.3. SISTEMAS

Os próximos tópicos descrevem mais detalhes de cada sistema utilizado para a realização do projeto.

3.3.1. *Back-end*

Esse sistema é responsável por efetuar a autenticação do projeto, o qual se comunica com o Sistema Unificado de Administração Pública (SUAP) para realizar a autenticação, verificando se a matrícula e a senha fornecidas pelos outros sistemas coincidem com as credenciais no SUAP. Caso as credenciais sejam válidas, os dados retornados são armazenados no banco de dados para a utilização no sistema.

A senha de cada usuário é criptografada para obter a segurança dos dados. Essa criptografia utiliza o OpenSSL⁴ para fornecer criptografia *Advanced Encryption Standard* (AES) com AES-256 e AES-128 (LARAVEL, [201-?]). Além da criptografia foi implementado o JSON Web *Token* (JWT), no qual é gerado pelo sistema um *token* único para cada usuário se sua autenticação for concedida, e somente com ele é possível acessar as funcionalidades da aplicação.

Ele também realiza a comunicação, utilizando o protocolo MQTT, entre o sistema Android e o hardware. O sistema para dispositivos móveis envia os dados para a função de abrir e fechar o laboratório. Após o recebimento, os dados são processados para identificar se o laboratório cujo QR *Code* foi escaneado corresponde ao cadastrado na reserva em questão e se está no horário correto. Se todos os dados coincidirem, é enviada uma mensagem ao *broker* para que então envie para o sistema embarcado no hardware correspondente uma mensagem contendo o comando de abrir ou fechar o laboratório em questão.

As solicitações ou fornecimento de dados passados pelos sistemas são processados para então serem consultados ou registrados no banco de dados. Os demais sistemas não têm acesso direto ao banco de dados, assim centralizando toda a informação em um único lugar, melhorando a manutenção e a segurança dos dados.

Toda a comunicação utiliza o padrão *Representational State Transfer* (REST) para transacionar informações entre os sistemas. Desse modo, a comunicação pode ser realizada pela *internet* e cada sistema será capaz de estar distante geograficamente, ou seja, cada sistema poderá ser hospedado em servidores distintos uns dos outros, não sendo necessário localizar-se todos em um mesmo servidor, e mesmo assim seu funcionamento não será prejudicado.

⁴ Implementação dos protocolos SSL e TLS, onde implementa e disponibiliza funções utilitárias de criptografia.

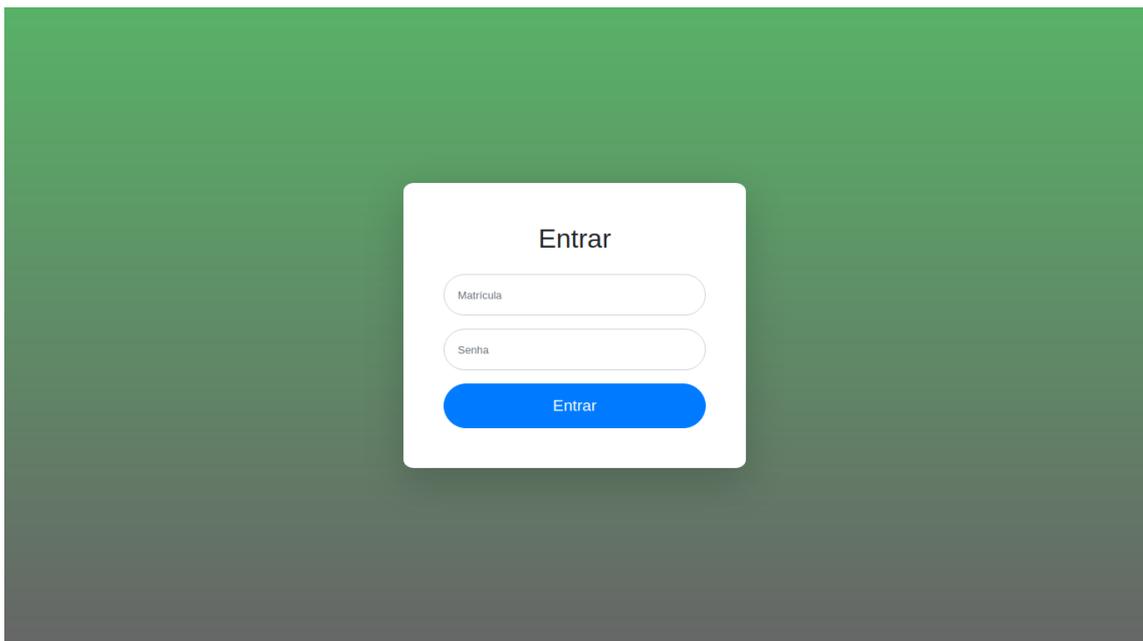
Para sua utilização, são criadas rotas⁵ a fim de realizar a comunicação. Cada funcionalidade tem sua rota única e os sistemas que necessitarem se comunicar, usarão a rota correspondente a funcionalidade desejada. Cada rota está descrita no APÊNDICE A – Rotas da API.

3.3.2. *Front-end*

Nesse tópico, serão detalhadas as interfaces do sistema, explanando suas funcionalidades e usuários que terão acesso. Foi utilizado o Vue CLI para o desenvolvimento do sistema.

A Figura 3 ilustra o formulário de autenticação, onde o usuário fornece a matrícula e a senha de seu SUAP para autenticar. Os dados são dirigidos a API onde é feita a autenticação.

Figura 3 – *Front-end* – Autenticar

A imagem mostra um formulário de login centralizado em uma tela com fundo verde escuro. O formulário é branco e possui o título "Entrar" no topo. Abaixo do título, há dois campos de entrada: "Matricula" e "Senha". Abaixo dos campos, há um botão azul com o texto "Entrar".

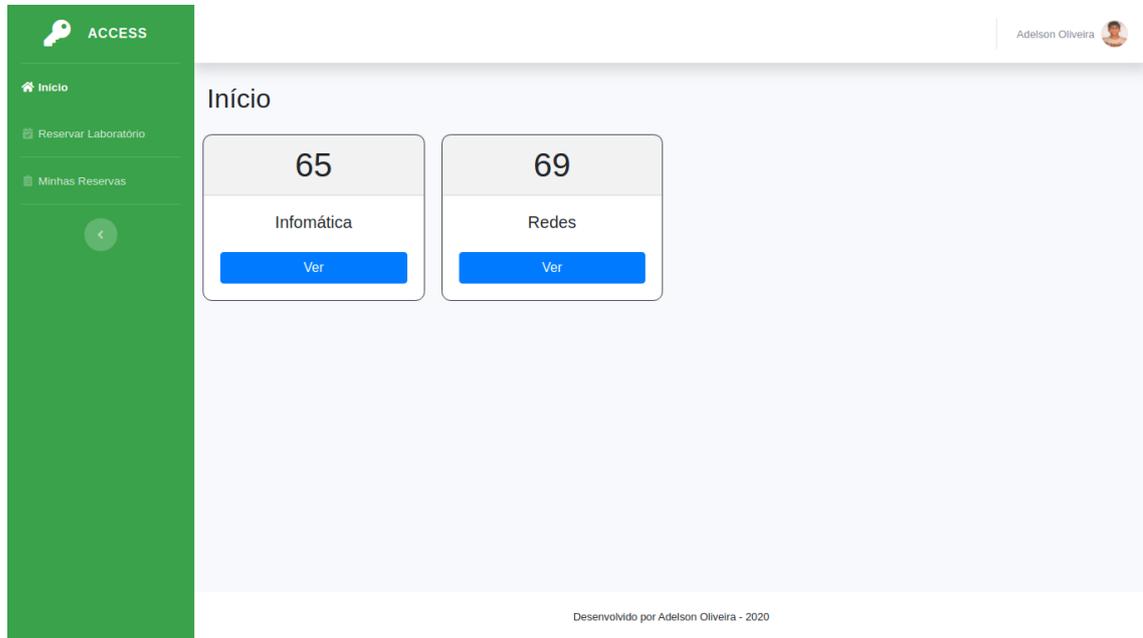
Fonte: Elaborado pelo autor, 2020.

Caso o usuário autenticado seja aluno ou professor, será direcionado para a tela ilustrada na Figura 4. A partir da barra lateral esquerda, ao clicar em **Início**, é capaz visualizar os laboratórios cadastrados, clicando em **Reservar Laboratório**, é possível realizar solicitações

⁵ São *Uniform Resource Locator* (URL) utilizadas pela API para que outros sistemas consultem ou forneçam dados.

de reserva de laboratórios e ao clicar em **Minhas Reservas**, é possível visualizar a lista de solicitações realizadas pelo usuário em questão. Ao clicar na foto do usuário, no canto superior direito, é possível sair da aplicação.

Figura 4 – *Front-end* – Perfil Aluno e Professor - Início



Fonte: Elaborado pelo autor, 2020.

Ao clicar em **Ver** em um dos laboratórios exibidos, abrirá uma janela em que o título exibirá o laboratório selecionado e abaixo, seus horários disponíveis e indisponíveis para o dia, como ilustrado na Figura 5. Os horários com tonalidade mais escura ainda estão disponíveis e os com tonalidade mais clara estão ocupados, como mostra a legenda no rodapé da janela.

Figura 5 – *Front-end* – Perfil Aluno e Professor - Início (Laboratório selecionado)

The screenshot shows a web application interface. On the left is a dark green sidebar with the text 'ACCESS' and a key icon. Below it are navigation links: 'Início', 'Reservar Laboratório', and 'Minhas Reservas'. The main content area has a header 'Início' and a card for 'Laboratório 65' with the subject 'Infomática' and a 'Ver' button. A modal window titled 'Laboratório 65' is open, displaying a table of time slots:

Manhã	Tarde	Noite
1º - 07:00 - 07:45	1º - 13:00 - 13:45	1º - 19:00 - 19:45
2º - 07:45 - 08:30	2º - 13:45 - 14:30	2º - 19:45 - 20:30
3º - 08:50 - 09:35	3º - 14:50 - 15:35	3º - 20:40 - 21:25
4º - 09:35 - 10:20	4º - 15:35 - 16:20	4º - 21:25 - 22:10
5º - 10:30 - 11:15	5º - 16:30 - 17:15	
6º - 11:15 - 12:00	6º - 17:15 - 18:00	

Below the table, the status is shown as 'Livre' (Free) and 'Ocupado' (Occupied). The footer of the application reads 'Desenvolvido por Adelson Oliveira - 2020'.

Fonte: Elaborado pelo autor, 2020.

Selecionando **Reservar Laboratório** na barra lateral esquerda, será carregado o formulário para solicitação, como visto na Figura 6. Na primeira sessão, está disponível uma lista de todos os laboratórios cadastrados; na segunda, a lista de alunos e professores que deseja adicionar à solicitação; a terceira, selecionará a data que deseja usufruir do laboratório; e na quarta e última, os horários que deseja realizar a solicitação.

O responsável pela solicitação será quem a criou, podendo abrir e fechar o laboratório selecionado. Os demais alunos ou professores adicionados à reserva são somente para registro de quem estará no laboratório no momento designado na reserva.

Caso o usuário tenha um perfil gerenciador, o formulário será o mesmo ao clicar em **Reservar Laboratório**.

Figura 6 – *Front-end* – Perfil Aluno e Professor - Reservar laboratório

Reservar laboratório

Laboratório
65 - Informática

Alunos
20171134040008 - Outro Aluno

Professores
1234567 - Rodrigo Leone Alves
7654321 - Outro Professor

Data da Reserva
27/10/2020

Horários

Manhã	Tarde	Noite
1º - 07:00 - 07:45	1º - 13:00 - 13:45	1º - 19:00 - 19:45
2º - 07:45 - 08:30	2º - 13:45 - 14:30	2º - 19:45 - 20:30
3º - 08:50 - 09:35	3º - 14:50 - 15:35	3º - 20:40 - 21:25
4º - 09:35 - 10:20	4º - 15:35 - 16:20	4º - 21:25 - 22:10
5º - 10:30 - 11:15	5º - 16:30 - 17:15	
6º - 11:15 - 12:00	6º - 17:15 - 18:00	

Enviar

Desenvolvido por Adelson Oliveira - 2020

Fonte: Elaborado pelo autor, 2020.

A Figura 7 ilustra a lista de solicitações de reservas confirmadas do usuário ao clicar em **Minhas Reservas**, no qual está disponível a data da reserva, laboratório desejado, usuários, horários e data e hora da criação da solicitação. Ao clicar na lupa na coluna **Usuários**, será exibida uma janela com todos os usuários da solicitação (Figura 8). Da mesma forma, ao clicar na lupa disponível na coluna **Horários**, será aberta uma janela com a listagem dos horários (Figura 9).

Figura 7 – *Front-end* – Perfil Aluno e Professor - Minhas reservas (Confirmada)

The screenshot shows a web interface for 'Minhas Reservas' (My Reservations). On the left is a green sidebar with navigation options: 'Início', 'Reservar Laboratório', and 'Minhas Reservas'. The main content area has a title 'Minhas Reservas' and two tabs: 'Confirmada' (active) and 'Não Confirmada'. Below the tabs is a table with the following data:

Data	Laboratório	Usuários	Horários	Data - Hora de criação
26/10/2020	65 - Infomática	Q	Q	26/10/2020 - 17:16
Data	Laboratório	Usuário	Horários	Data - Hora de criação

At the bottom of the page, it says 'Desenvolvido por Adelson Oliveira - 2020'.

Fonte: Elaborado pelo autor, 2020.

Figura 8 – *Front-end* – Perfil Aluno e Professor - Minhas reservas (Usuários)

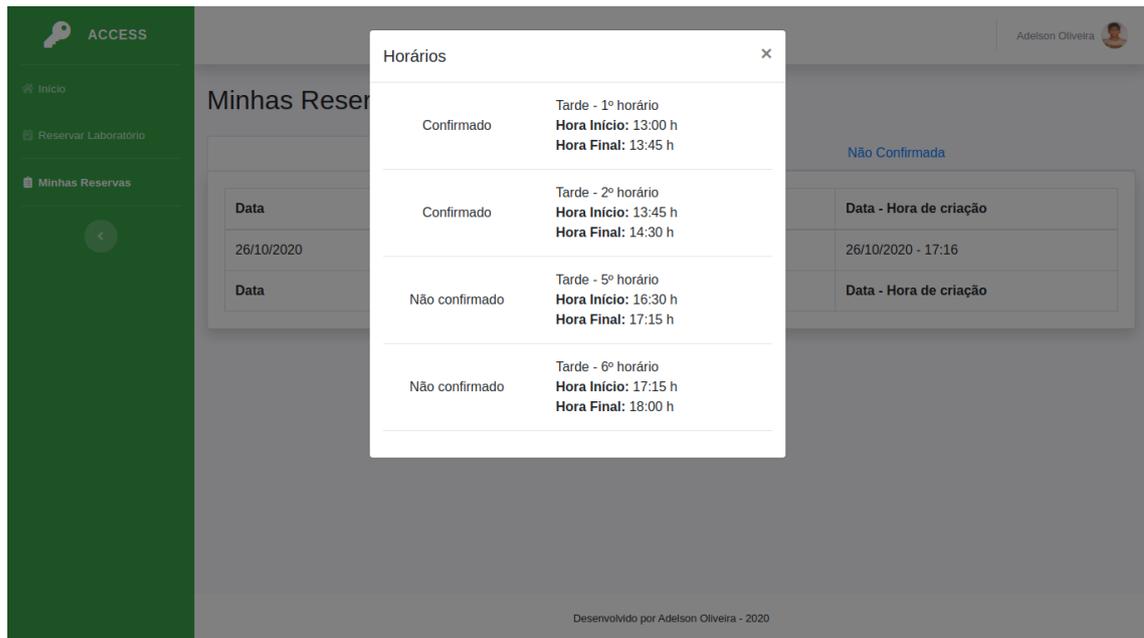
This screenshot is similar to Figure 7, but with a modal window open over the 'Usuários' column of the table. The modal is titled 'Usuários' and contains the following information:

- Nome: Rodrigo Leone Alves
- Matrícula: 1234567
- Perfil: Professor

The background table and interface elements are dimmed. At the bottom, it says 'Desenvolvido por Adelson Oliveira - 2020'.

Fonte: Elaborado pelo autor, 2020.

Figura 9 – *Front-end* – Perfil Aluno e Professor - Minhas reservas (Horários)

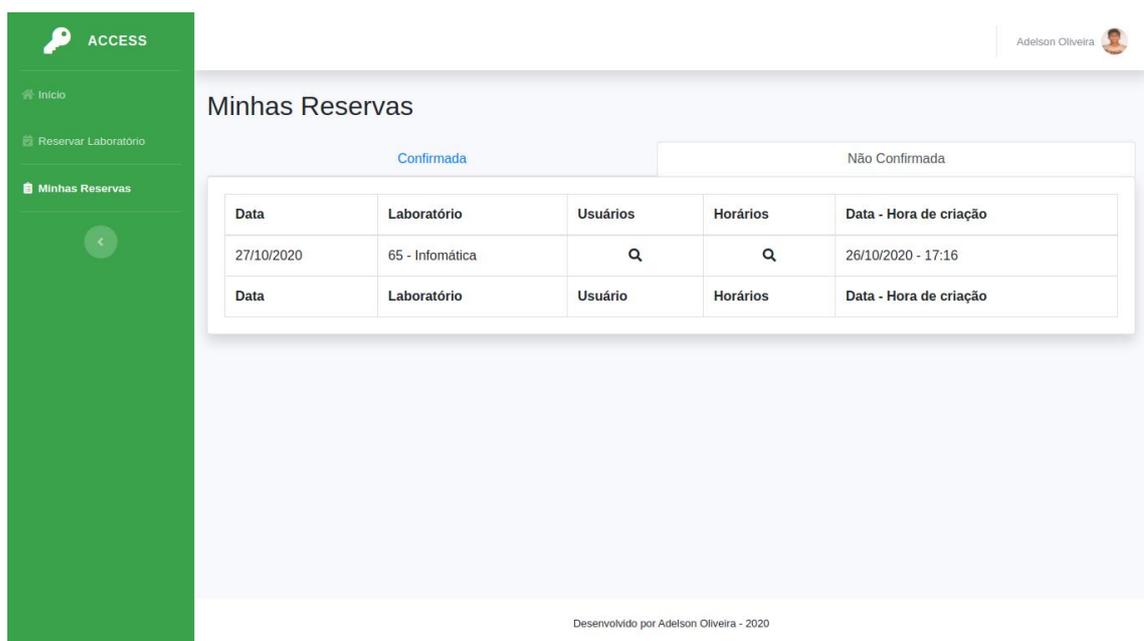


Fonte: Elaborado pelo autor, 2020.

Na Figura 10 é exibida a listagem das solicitações não confirmadas ao clicar em **Não Confirmada**, tendo os mesmos campos e ações da Figura 7, Figura 8 e Figura 9.

Caso o usuário tenha um perfil gerenciador, terá acesso idêntico a listagem de suas reservas clicando em **Minhas Reservas**.

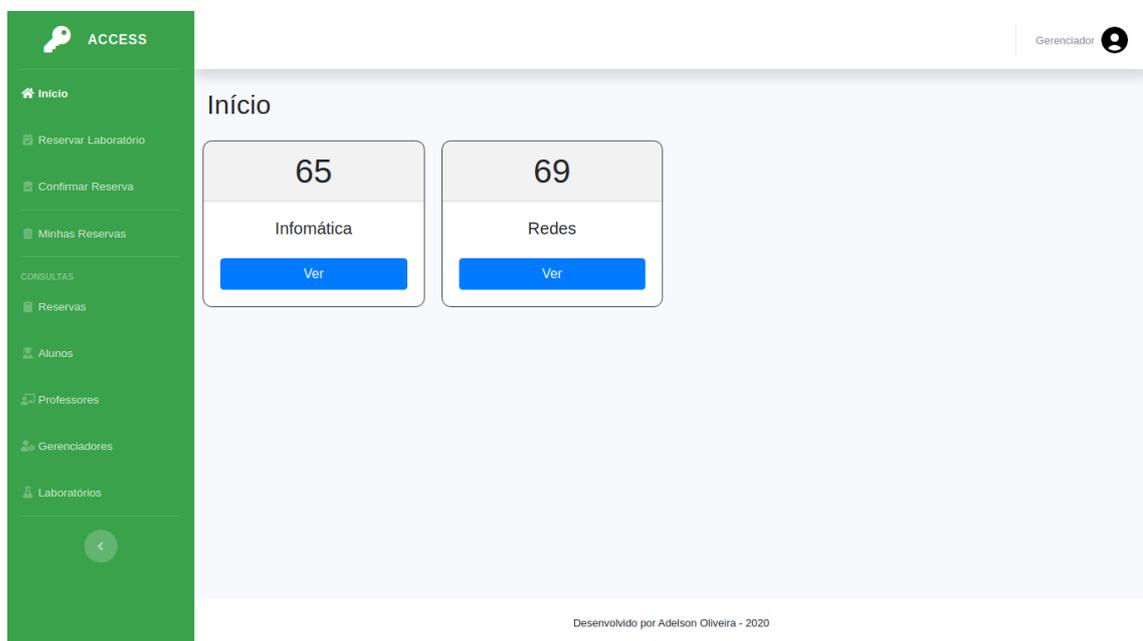
Figura 10 – *Front-end* – Perfil Aluno e Professor - Minhas reservas (Não confirmada)



Fonte: Elaborado pelo autor, 2020.

Caso o usuário autenticado na Figura 3 seja um gerenciador, será direcionado para uma tela conforme apresentada na Figura 11. Tendo acesso pela barra lateral às opções **Início**, no qual são exibidos os laboratórios cadastrados; **Reservar Laboratório**, onde um usuário poderá realizar as solicitações; **Confirmar Reserva**, no qual exibe uma lista das solicitações a serem confirmadas; **Minhas Reservas**, que exibe a lista das solicitações realizadas pelo usuário e as próximas opções são consultas aos dados cadastrados, possuindo a listagem das **Reservas**, **Alunos**, **Professores**, **Gerenciadores** e **Laboratórios**, respectivamente.

Figura 11 – *Front-end* – Perfil Gerenciador - Início



Fonte: Elaborado pelo autor, 2020.

Do mesmo modo como ilustrado na Figura 5, clicando em **Ver** em um dos laboratórios exibidos, será aberta uma janela com os horários do dia. A diferença para o perfil de gerenciador é a exibição do estado do laboratório no título da janela, se está **ABERTO** ou **FECHADO** no momento, como mostrado na Figura 12. Esse estado é obtido a partir do serviço MQTT, no qual a aplicação assina o tópico correspondente ao laboratório para obter a informação.

Figura 12 – *Front-end* – Perfil Gerenciador - Início (Laboratório selecionado)

The screenshot shows the 'Início' page of the Gerenciador profile. A modal window titled 'Laboratório 65' is open, displaying a schedule for 'FECHADO' (Closed). The schedule is organized into three columns: 'Manhã', 'Tarde', and 'Noite'. Each column lists six time slots. Below the schedule, it indicates 'Livre' (Free) and 'Ocupado' (Occupied) status.

Laboratório 65			FECHADO		
Manhã		Tarde		Noite	
1º - 07:00 - 07:45	1º - 13:00 - 13:45	1º - 19:00 - 19:45			
2º - 07:45 - 08:30	2º - 13:45 - 14:30	2º - 19:45 - 20:30			
3º - 08:50 - 09:35	3º - 14:50 - 15:35	3º - 20:40 - 21:25			
4º - 09:35 - 10:20	4º - 15:35 - 16:20	4º - 21:25 - 22:10			
5º - 10:30 - 11:15	5º - 16:30 - 17:15				
6º - 11:15 - 12:00	6º - 17:15 - 18:00				

Below the schedule, it indicates 'Livre' (Free) and 'Ocupado' (Occupied) status.

Fonte: Elaborado pelo autor, 2020.

A Figura 13 ilustra a tela de **Confirmar Reserva** em que são listadas as reservas a serem confirmadas, constando a data da reserva, o laboratório desejado, quem a solicitou, qual a data e hora de solicitação e o botão para mais detalhes.

Figura 13 – *Front-end* – Perfil Gerenciador - Confirmar reserva de laboratórios

The screenshot shows the 'Confirmar Reserva de Laboratórios' page. A table lists reservations to be confirmed. The table has five columns: 'Data', 'Laboratório', 'Criador', 'Data - Hora de criação', and 'Ações'. A single reservation is listed for 27/10/2020 at 65 - Infomática, created by Adelson de Oliveira Camara da Cruz on 26/10/2020 at 17:16 h. An 'Ações' button with a right arrow is visible next to the reservation.

Data	Laboratório	Criador	Data - Hora de criação	Ações
27/10/2020	65 - Infomática	Adelson de Oliveira Camara da Cruz	26/10/2020 - 17:16 h	→

Fonte: Elaborado pelo autor, 2020.

Ao clicar no botão para mais detalhes, ilustrado na Figura 13 o usuário será redirecionado para o detalhamento da reserva selecionada, exibida na Figura 14, de modo a poder observar todos os dados e confirmar os horários apropriados para essa reserva.

Caso outro usuário já tenha esses horários confirmados previamente, mas não se encontre no Campus no horário da reserva, o gerenciador poderá selecionar a caixa de seleção denominada **Sobrepôr**, localizada acima dos horários para sobrepôr a reserva anterior, liberando assim o laboratório para acesso por outro usuário naquela data e horário.

Figura 14 – *Front-end* – Perfil Gerenciador - Reserva confirmar

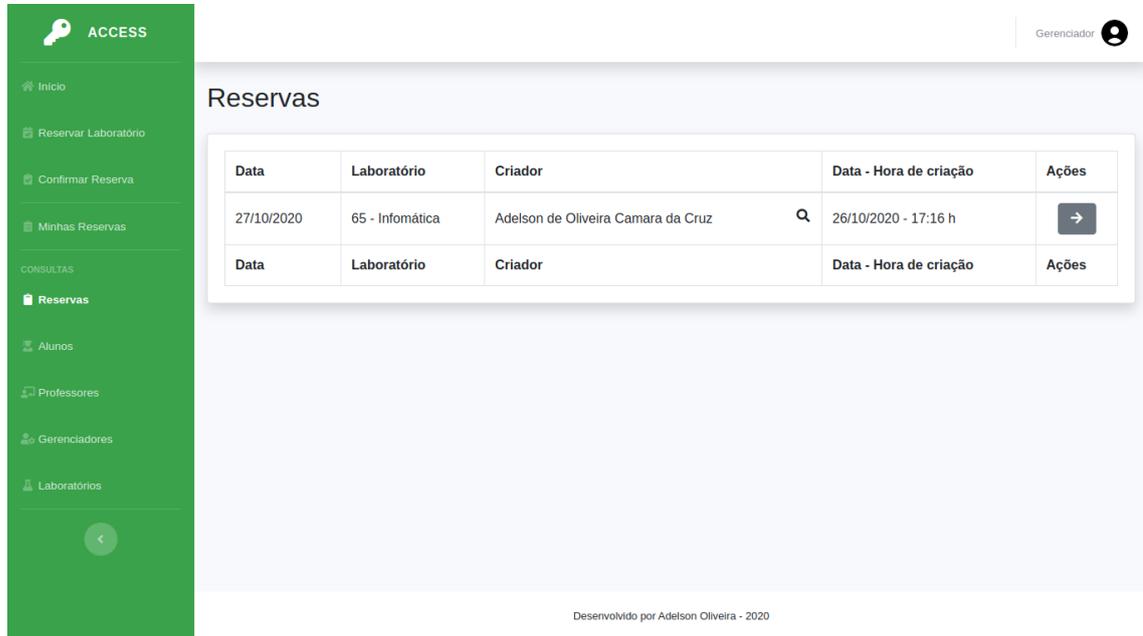
The screenshot displays the 'Reserva Confirmar #1' interface. On the left is a green sidebar with navigation options: Início, Reservar Laboratório, Confirmar Reserva, Minhas Reservas, CONSULTAS, Reservas, Alunos, Professores, Gerenciadores, and Laboratórios. The main content area includes a 'Voltar' button, a 'Laboratório' section (65 - Informática), a 'Data' section (27/10/2020), a 'Criador' section (Adelson de Oliveira Camara da Cruz, Matrícula: 20171134040009, Perfil: Aluno), and a 'Usuários' section (Rodrigo Leone Alves, Matrícula: 1234567, Perfil: Professor). Below is the 'Horários' section with a 'Sobrepôr' checkbox and four time slots, each with a 'Confirmar' checkbox and start/end times. A 'Concluir' button is at the bottom right. The footer reads 'Desenvolvido por Adelson Oliveira - 2020'.

Fonte: Elaborado pelo autor, 2020.

Na Figura 15 estão listadas todas as reservas cadastradas e confirmadas. Possuindo a data da reserva, o laboratório desejado, quem solicitou a reserva, qual a data e hora foi criada e o botão para mais detalhes em cada reserva.

Ao clicar na lupa ao lado do nome do criador, abrirá uma janela com seus dados, como visto na Figura 16.

Figura 15 – *Front-end* – Perfil Gerenciador - Reservas



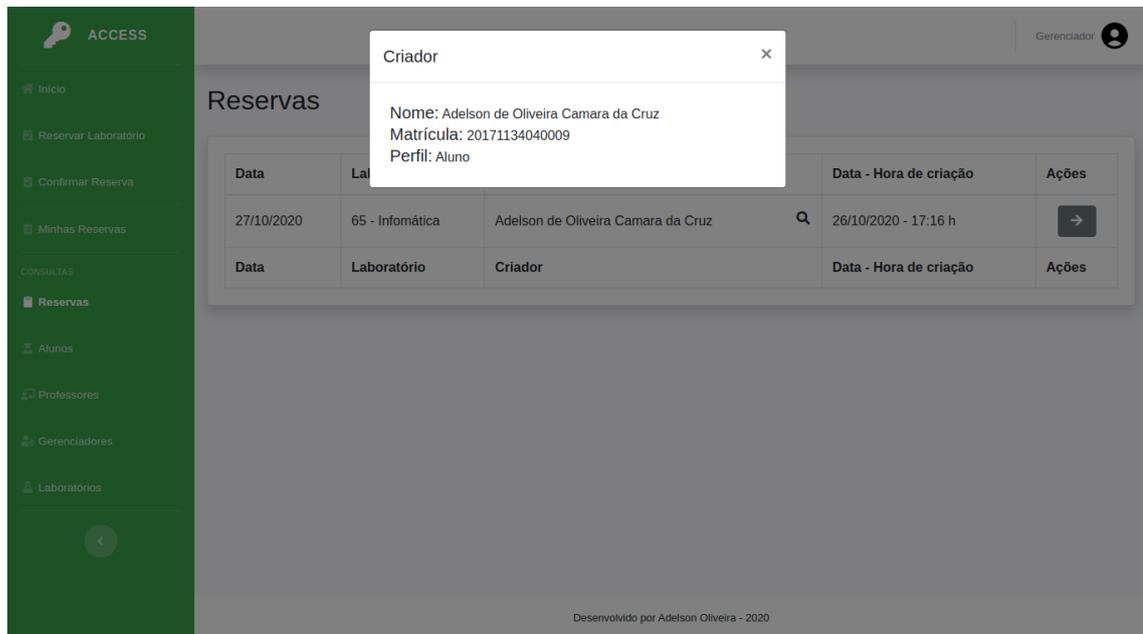
The screenshot shows the 'Reservas' page in the 'Perfil Gerenciador' system. The page has a green sidebar with navigation options: 'Início', 'Reservar Laboratório', 'Confirmar Reserva', 'Minhas Reservas', and 'CONSULTAS' (Reservas, Alunos, Professores, Gerenciadores, Laboratórios). The main content area displays a table with the following data:

Data	Laboratório	Criador	Data - Hora de criação	Ações
27/10/2020	65 - Infomática	Adelson de Oliveira Camara da Cruz	26/10/2020 - 17:16 h	[Lupa]
Data	Laboratório	Criador	Data - Hora de criação	Ações

The user is logged in as 'Gerenciador'. The footer indicates 'Desenvolvido por Adelson Oliveira - 2020'.

Fonte: Elaborado pelo autor, 2020.

Figura 16 – *Front-end* – Perfil Gerenciador - Reservas (Criador)



The screenshot shows the 'Reservas' page with a modal window open over the first row of the table. The modal window displays the following information:

Criador

Nome: Adelson de Oliveira Camara da Cruz
 Matrícula: 20171134040009
 Perfil: Aluno

The background table is dimmed, showing the same data as in Figure 15. The user is logged in as 'Gerenciador'. The footer indicates 'Desenvolvido por Adelson Oliveira - 2020'.

Fonte: Elaborado pelo autor, 2020.

Ao clicar no botão para mais detalhes ilustrado na Figura 15, o usuário será redirecionado para a tela conforme a da Figura 17, assim, podendo observar todos os dados da reserva.

Figura 17 – *Front-end* – Perfil Gerenciador - Detalhes da reserva

The screenshot displays the 'Reserva #1' details page. On the left is a green sidebar with navigation options: Início, Reservar Laboratório, Confirmar Reserva, Minhas Reservas, CONSULTAS, Reservas, Alunos, Professores, Gerenciadores, and Laboratórios. The main content area shows the reservation details:

- Reserva #1** (with a 'Voltar' button)
- Laboratório:** 65 - Infomática
- Data:** 27/10/2020
- Criador:**
 - Nome: Adelson de Oliveira Camara da Cruz
 - Matricula: 20171134040009
 - Perfil: Aluno
- Usuários:**
 - Nome: Rodrigo Leone Alves
 - Matricula: 1234567
 - Perfil: Professor
- Horários:**

Confirmado	Confirmado	Não confirmado	Não confirmado
Tarde - 1º horário	Tarde - 2º horário	Tarde - 5º horário	Tarde - 6º horário
Hora Início: 13:00 h Hora Final: 13:45 h	Hora Início: 13:45 h Hora Final: 14:30 h	Hora Início: 16:30 h Hora Final: 17:15 h	Hora Início: 17:15 h Hora Final: 18:00 h

At the bottom, it says 'Desenvolvido por Adelson Oliveira - 2020'.

Fonte: Elaborado pelo autor, 2020.

Na Figura 18, Figura 19 e Figura 20 se encontram as telas que listam os **Alunos**, **Professores** e **Gerenciadores** respectivamente, contendo a matrícula, nome e data e hora do seu cadastro.

Figura 18 – *Front-end* – Perfil Gerenciador - Alunos

ACCESS

Gerenciador

Alunos

Matrícula	Nome	Data - Hora de criação
20171134040008	Outro Aluno	23/10/2020 - 17:07 h
20171134040009	Adelson de Oliveira Camara da Cruz	23/10/2020 - 17:07 h
Matrícula	Nome	Data - Hora de criação

Desenvolvido por Adelson Oliveira - 2020

Fonte: Elaborado pelo autor, 2020.

Figura 19 – *Front-end* – Perfil Gerenciador - Professores

ACCESS

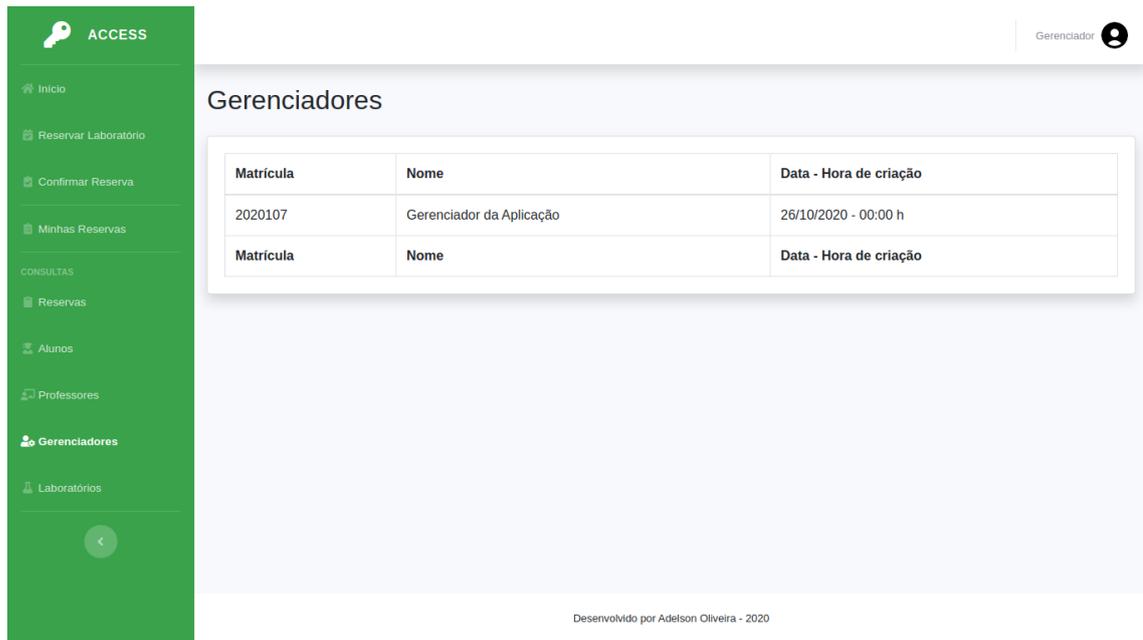
Gerenciador

Professores

Matrícula	Nome	Data - Hora de criação
1234567	Rodrigo Leone Alves	26/10/2020 - 00:00 h
7654321	Outro Professor	26/10/2020 - 00:00 h
Matrícula	Nome	Data - Hora de criação

Desenvolvido por Adelson Oliveira - 2020

Fonte: Elaborado pelo autor, 2020.

Figura 20 – *Front-end* – Perfil Gerenciador - Gerenciadores


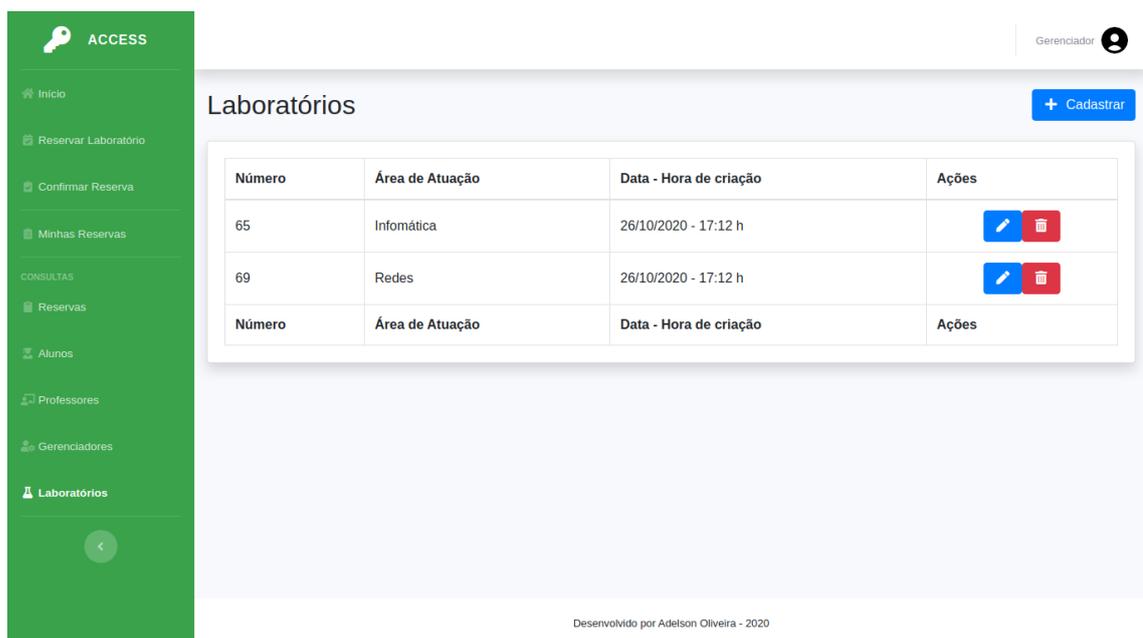
Gerenciadores

Matrícula	Nome	Data - Hora de criação
2020107	Gerenciador da Aplicação	26/10/2020 - 00:00 h
Matrícula	Nome	Data - Hora de criação

Desenvolvido por Adelson Oliveira - 2020

Fonte: Elaborado pelo autor, 2020.

A Figura 21 ilustra a tela com todos os laboratórios cadastrados quando o item **Laboratórios** é selecionado na barra lateral esquerda. Ao selecionar essa opção, são apresentados o número do laboratório, qual a área de atuação, data e hora que foi cadastrado e as ações de atualizar ou excluir o laboratório. Há ainda a opção de cadastrar outro laboratório no canto superior direito.

Figura 21 – *Front-end* – Perfil Gerenciador - Laboratórios


Laboratórios

+ Cadastrar

Número	Área de Atuação	Data - Hora de criação	Ações
65	Infomática	26/10/2020 - 17:12 h	 
69	Redes	26/10/2020 - 17:12 h	 
Número	Área de Atuação	Data - Hora de criação	Ações

Desenvolvido por Adelson Oliveira - 2020

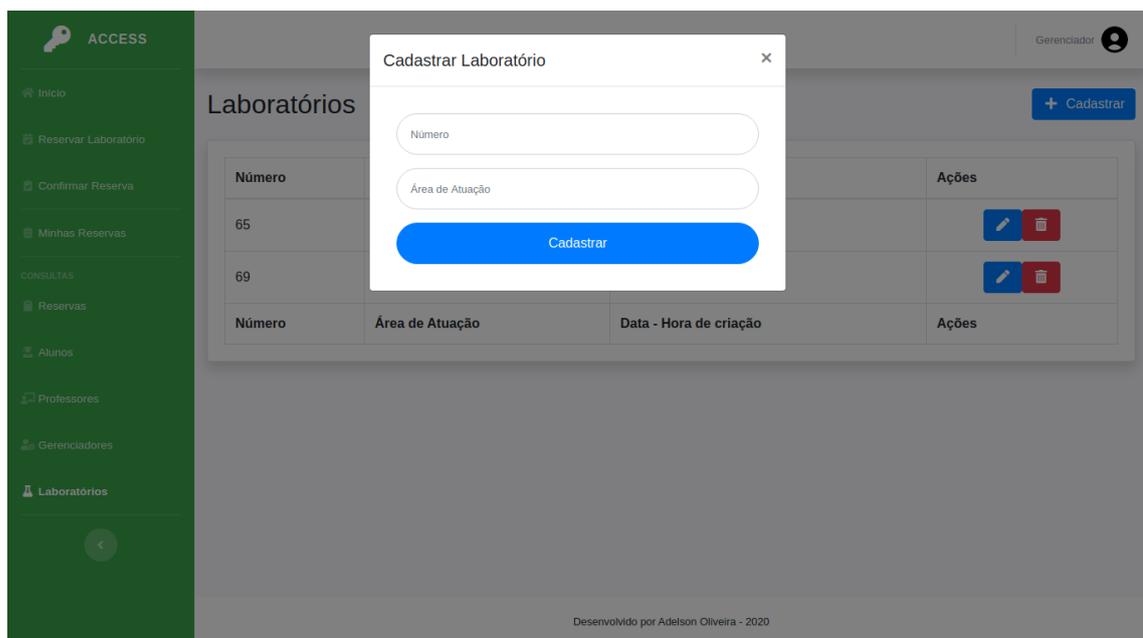
Fonte: Elaborado pelo autor, 2020.

Ao clicar no botão **Cadastrar**, como mostrado na Figura 21, é exibida a janela com o formulário de cadastro constando o número e a área de atuação (Figura 22).

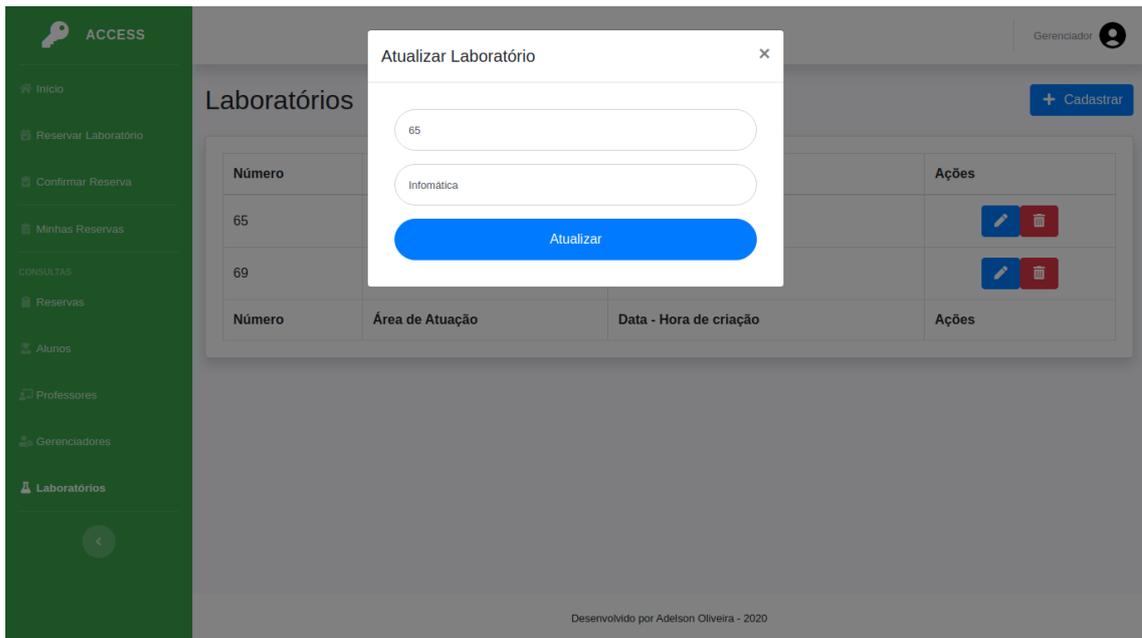
Selecionando o botão azul com o lápis para atualizar, uma janela com o formulário preenchido para a atualização aparece. Sendo possível a modificação dos campos para a realização da atualização (Figura 23).

Clicando no botão com uma lixeira para remover o laboratório, será perguntado se deseja excluir o laboratório. Caso o usuário opte por **Cancelar**, nada será feito, mas se ele selecionar **OK**, o laboratório será excluído e a página será recarregada (Figura 24).

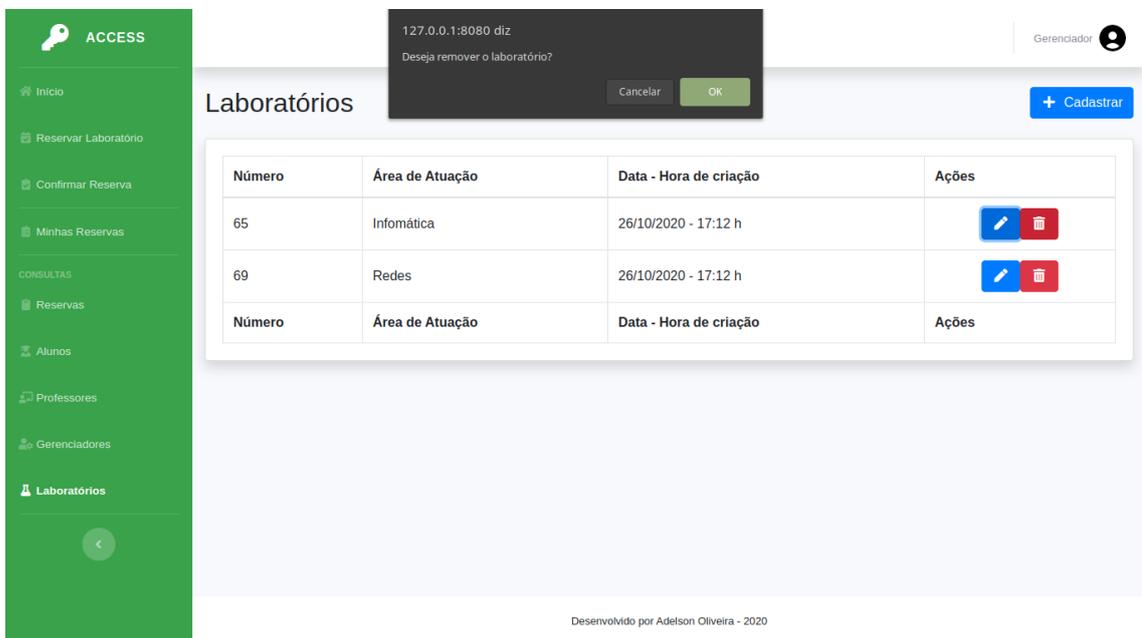
Figura 22 – Front-end – Perfil Gerenciador - Laboratórios (Cadastrar laboratório)



Fonte: Elaborado pelo autor, 2020.

Figura 23 – *Front-end* – Perfil Gerenciador - Laboratórios (Atualizar laboratório)

Fonte: Elaborado pelo autor, 2020.

Figura 24 – *Front-end* – Perfil Gerenciador - Laboratórios (Confirmar remoção)

Fonte: Elaborado pelo autor, 2020.

3.3.3. *Mobile*

A seguir, será detalhada as interfaces do sistema Android, esclarecendo suas funcionalidades em cada figura.

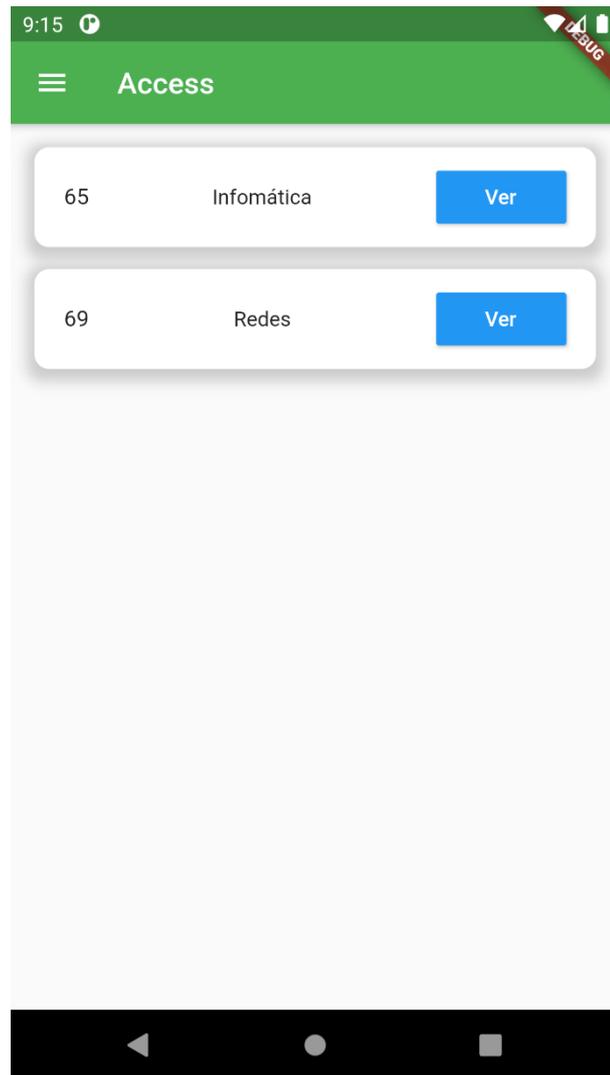
Na Figura 25 é exibida a tela de autenticação do aplicativo onde o usuário fornece a matrícula e a senha do SUAP para realizar a autenticação e então esses dados são enviados a API para a autenticação, como na Figura 3.

Figura 25 – Mobile – Autenticação



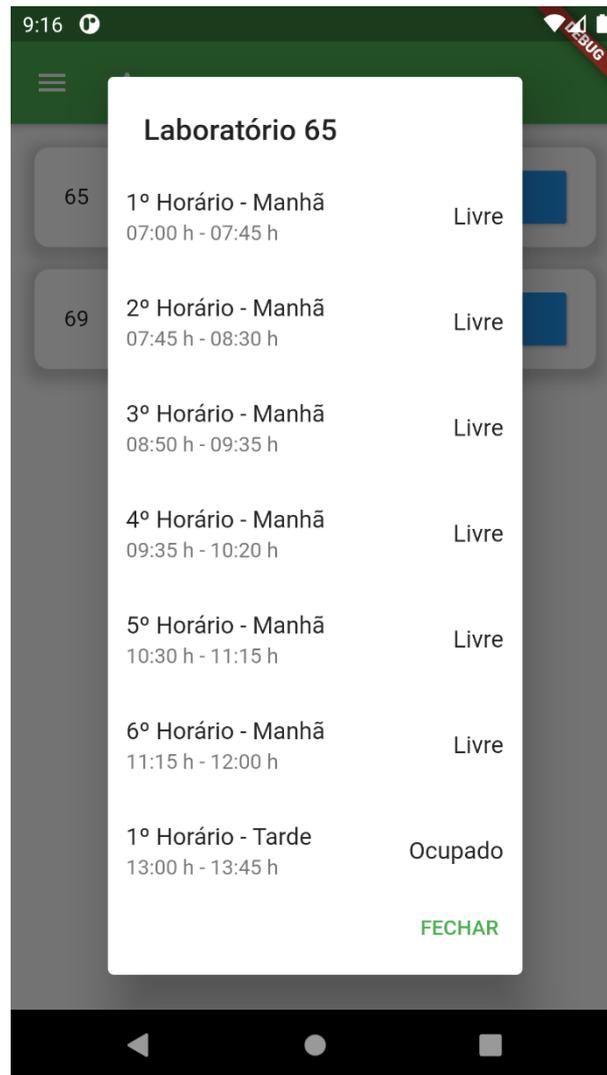
Fonte: Elaborado pelo autor, 2020.

Após a autenticação, o usuário será redirecionado para a tela de início da aplicação, a qual contém a lista dos laboratórios (Figura 26). Clicando no botão **Ver**, abrirá uma janela com os horários e a informação se o laboratório está livre para que o usuário possa realizar uma solicitação ou ocupado no horário desejado, como ilustrado na Figura 27 e na Figura 28.

Figura 26 – *Mobile* – Início

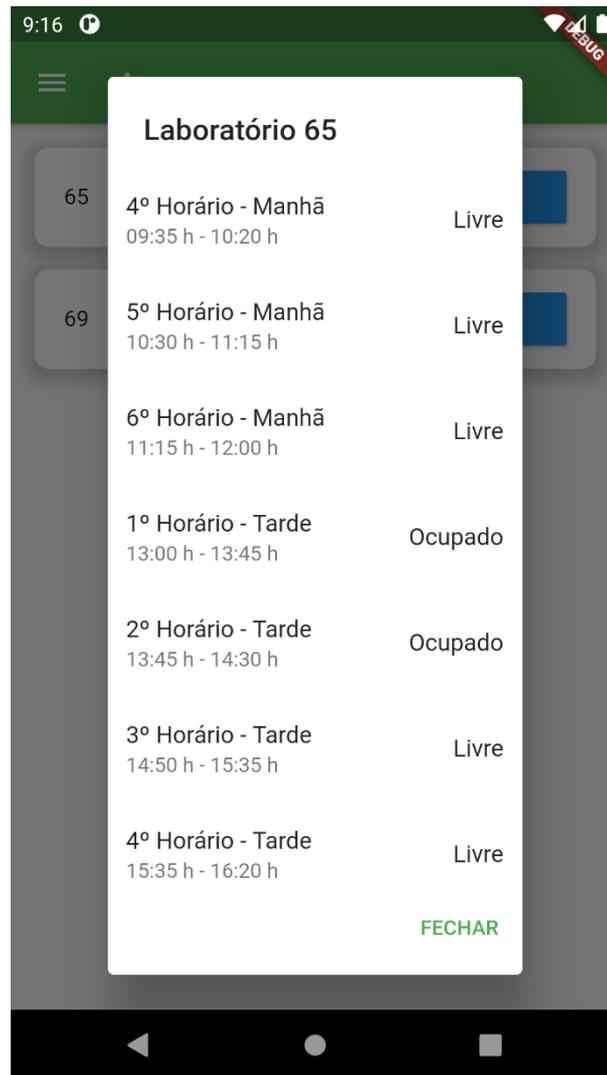
Fonte: Elaborado pelo autor, 2020.

Figura 27 – Mobile – Início (Laboratório selecionado I)



Fonte: Elaborado pelo autor, 2020.

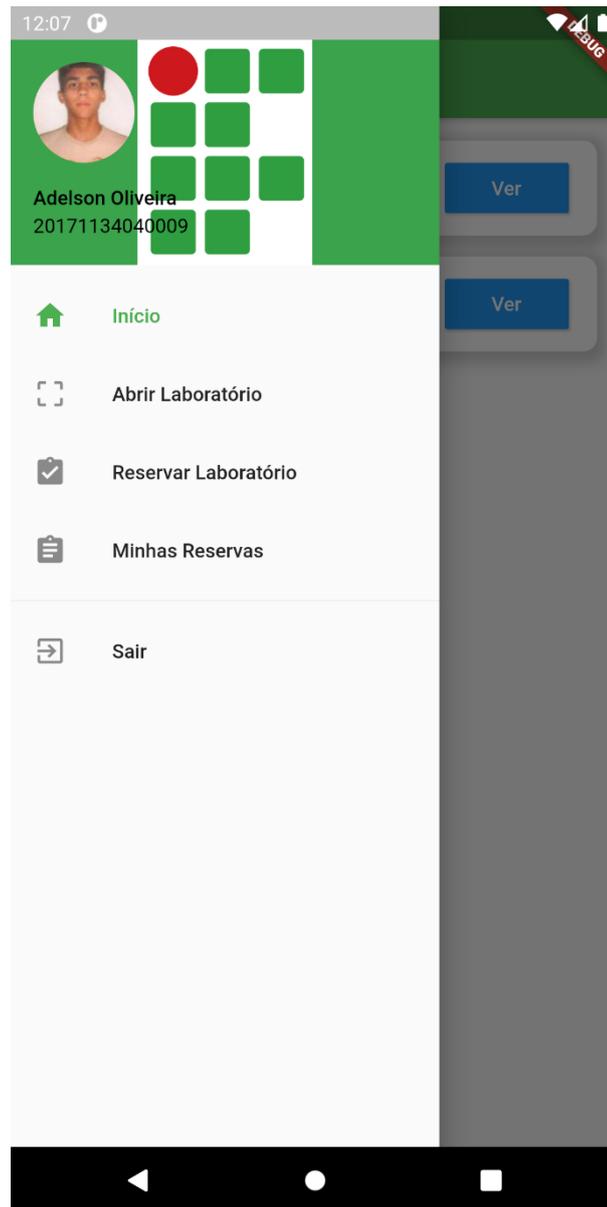
Figura 28 – Mobile – Início (Laboratório selecionado II)



Fonte: Elaborado pelo autor, 2020.

Ao clicar nos três traços horizontais no canto superior esquerdo da Figura 26, é exibida a barra de navegação lateral onde encontram-se as opções **Início**, que lista os laboratórios; **Abrir Laboratório**, onde é exibida a lista de laboratórios reservados no dia; **Reservar Laboratório**, no qual o usuário poderá realizar suas solicitações; **Minhas reservas**, onde terá a lista das solicitações do usuário e **Sair**, para sair da aplicação (Figura 29).

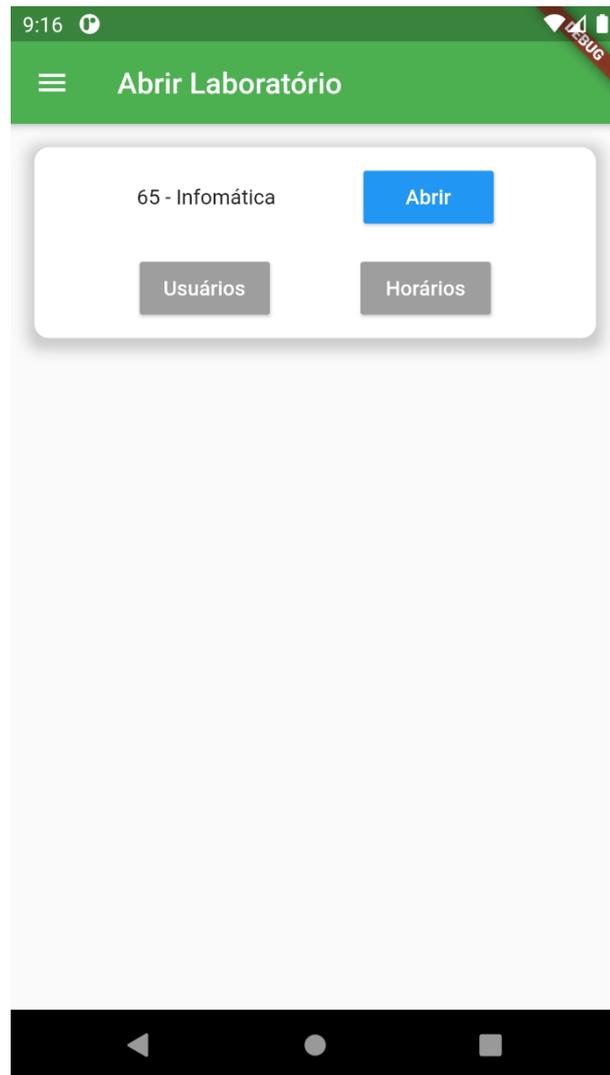
Figura 29 – Mobile – Barra de Navegação Lateral



Fonte: Elaborado pelo autor, 2020.

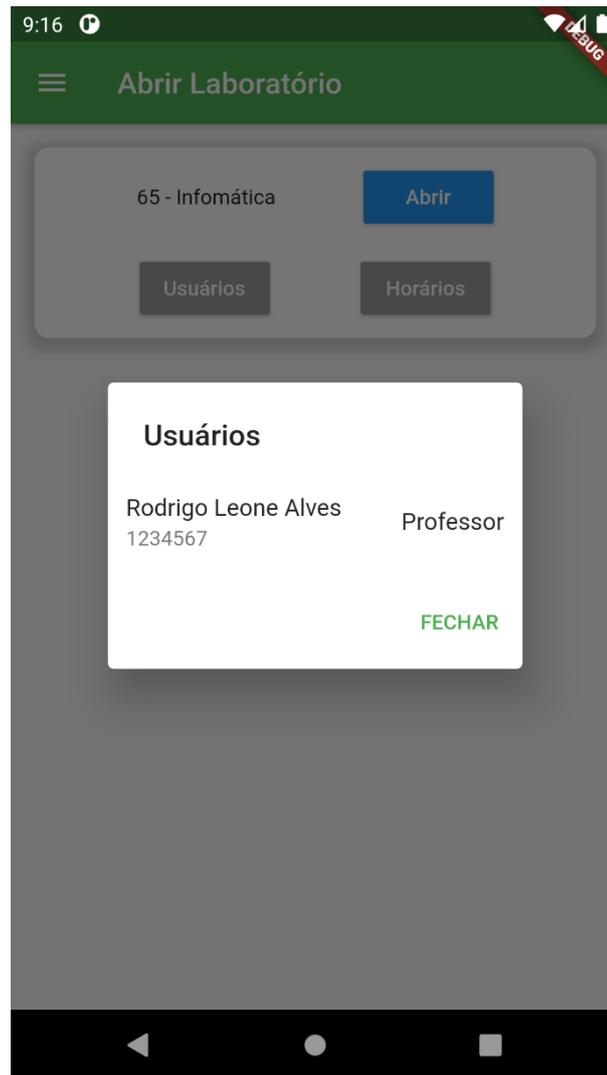
Na Figura 30 é exibida a lista de laboratórios reservados pelo usuário para o dia clicando em **Abrir Laboratório**. Nesta tela, é apresentado o número e a área de atuação do laboratório; o botão **Abrir**, para realizar a abertura do laboratório; o botão **Usuários**, que abre uma tela com os usuários da solicitação, ilustrado na Figura 31 e o botão **Horários**, que exibe uma janela com os horários confirmados para a reserva, ilustrado na Figura 32.

Figura 30 – Mobile – Abrir laboratório



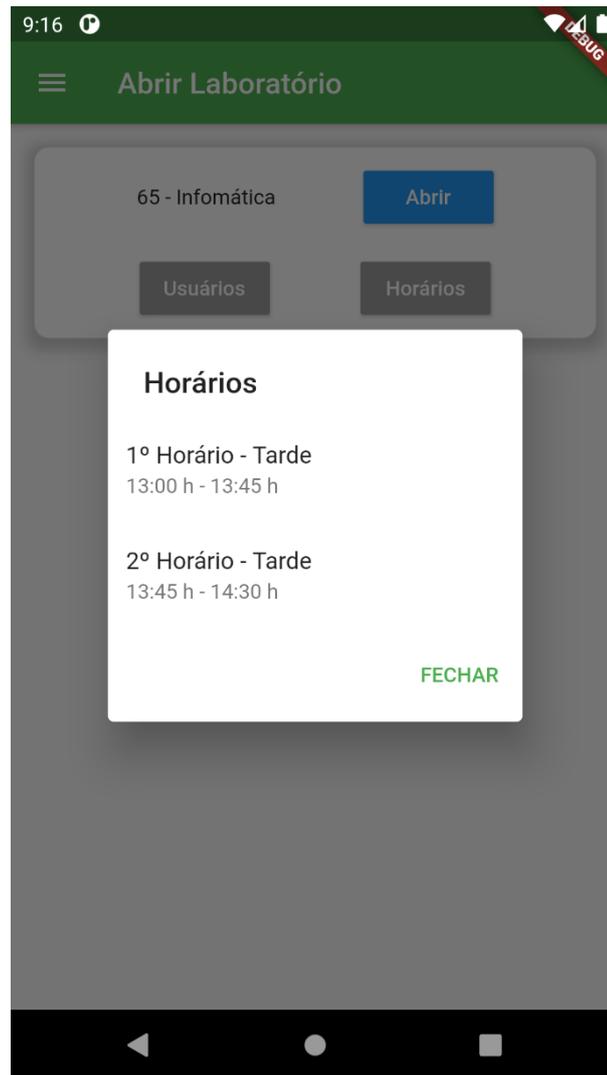
Fonte: Elaborado pelo autor, 2020.

Figura 31 – Mobile – Abrir laboratório (Usuários)



Fonte: Elaborado pelo autor, 2020.

Figura 32 – Mobile – Abrir laboratório (Horários)

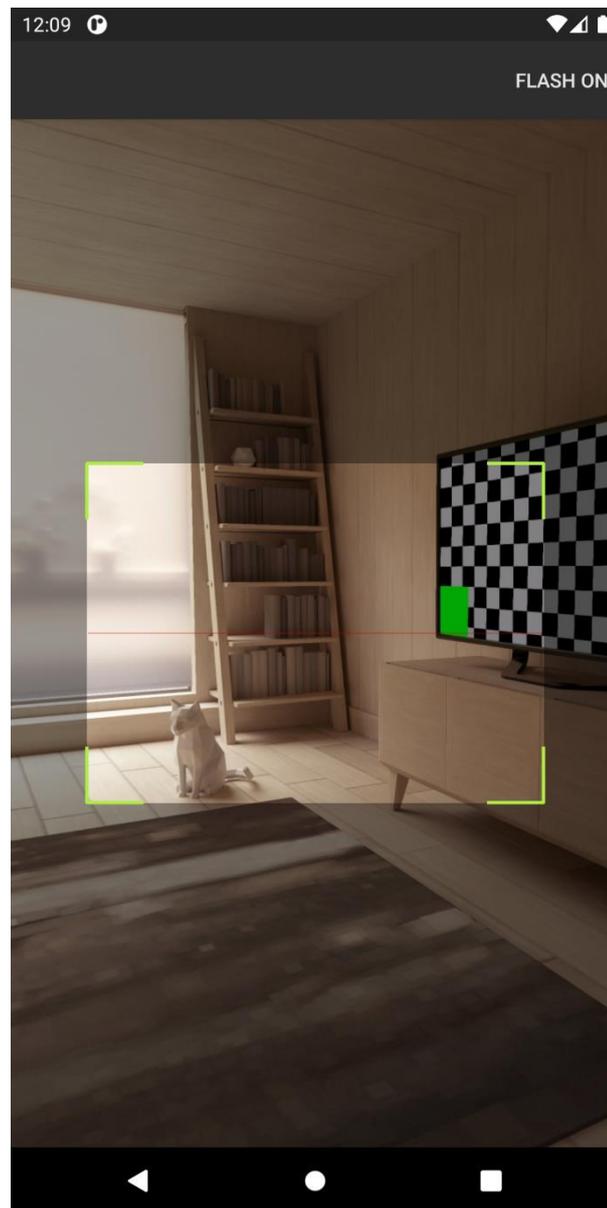


Fonte: Elaborado pelo autor, 2020.

Ao clicar no botão **Abrir** na Figura 30, abrirá a tela para o escaneamento do código QR contido na porta do laboratório, ilustrada na Figura 33. Após a leitura, os dados referentes à reserva e o número do laboratório escaneado são enviados para a API onde será conferido se o laboratório pertence à reserva e se o horário do escaneamento corresponde ao solicitado. Caso haja inconsistência nos dados, o usuário receberá uma mensagem correspondente ao erro ocorrido. Caso os dados coincidam, a API enviará ao *broker* uma mensagem contendo o comando para abrir o laboratório e o botão **Abrir** mudará o estado para **Fechar**. Esse estado é obtido do serviço MQTT, no qual a aplicação assina o tópico correspondente ao laboratório para obter a informação vinda do *broker*.

Com o botão no estado **Fechar**, ao clicar, é enviado à API o número da reserva e do laboratório para, do mesmo modo, conferir os dados. Caso os dados coincidam, a API enviará ao *broker* uma mensagem contendo o comando para fechar o laboratório e o botão voltará ao estado **Abrir**.

Figura 33 – *Mobile* – Escaneamento do QR Code



Fonte: Elaborado pelo autor, 2020.

A Figura 34 exibe a tela para reservar o laboratório selecionando **Reservar Laboratório** ilustrado na Figura 29. Na primeira sessão é exibida uma listagem dos laboratórios cadastrados, sendo possível selecionar somente um para a reserva (Figura 35); na segunda, há uma lista de alunos e professores que deseja adicionar à solicitação, podendo selecionar mais de um,

conforme ilustrado na Figura 36 e na Figura 37, respectivamente; na terceira, o usuário escolherá a data que deseja realizar a reserva (Figura 38); e na quarta, os horários em que deseja reservar o laboratório (Figura 39).

Figura 34 – *Mobile* – Reservar laboratório

9:15

Reservar Laboratório

Laboratório
65 - Infomática

Alunos
Selecionar 0 selecionados

Professores
Selecionar 1 selecionados

Data
Selecionar 26/10/2020

Horários

Manhã
Selecionar 2 selecionados

Tarde
Selecionar 0 selecionados

Noite
Selecionar 0 selecionados

Enviar

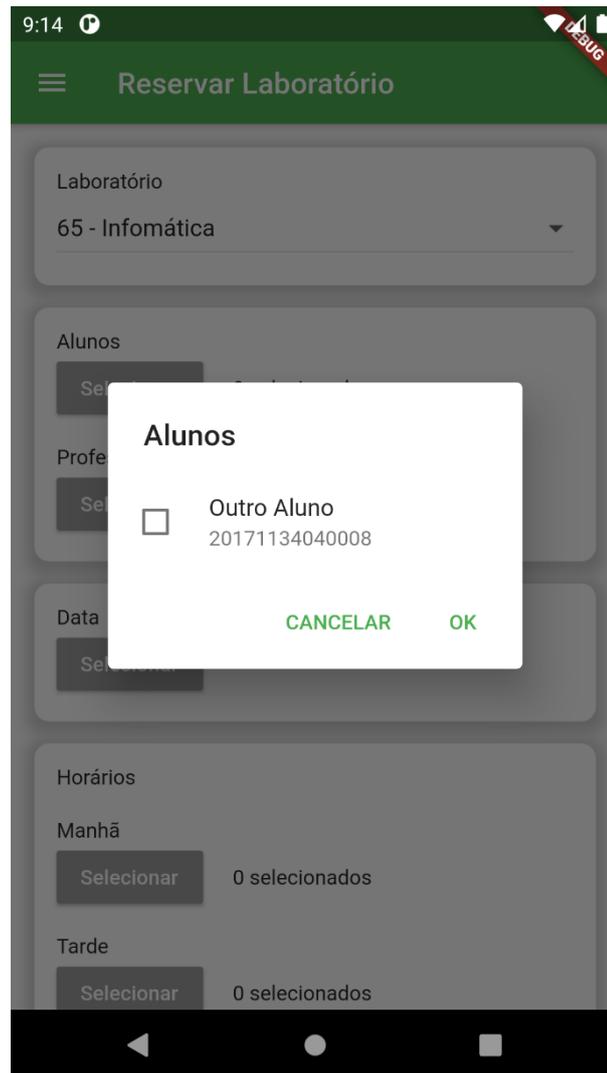
Fonte: Elaborado pelo autor, 2020.

Figura 35 – Mobile – Reservar laboratório (Laboratório)



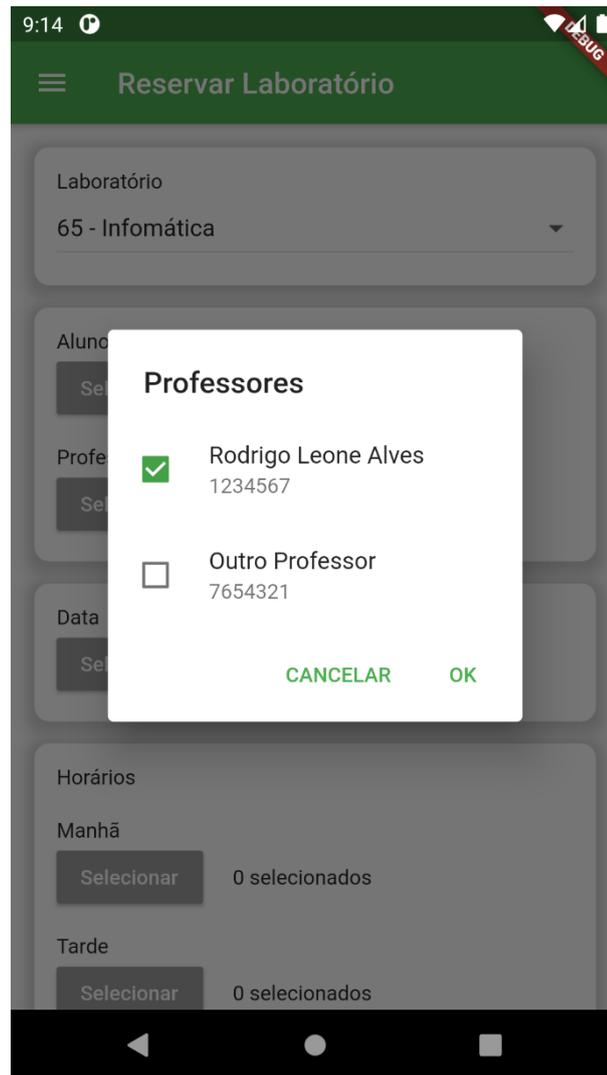
Fonte: Elaborado pelo autor, 2020.

Figura 36 – Mobile – Reservar laboratório (Alunos)

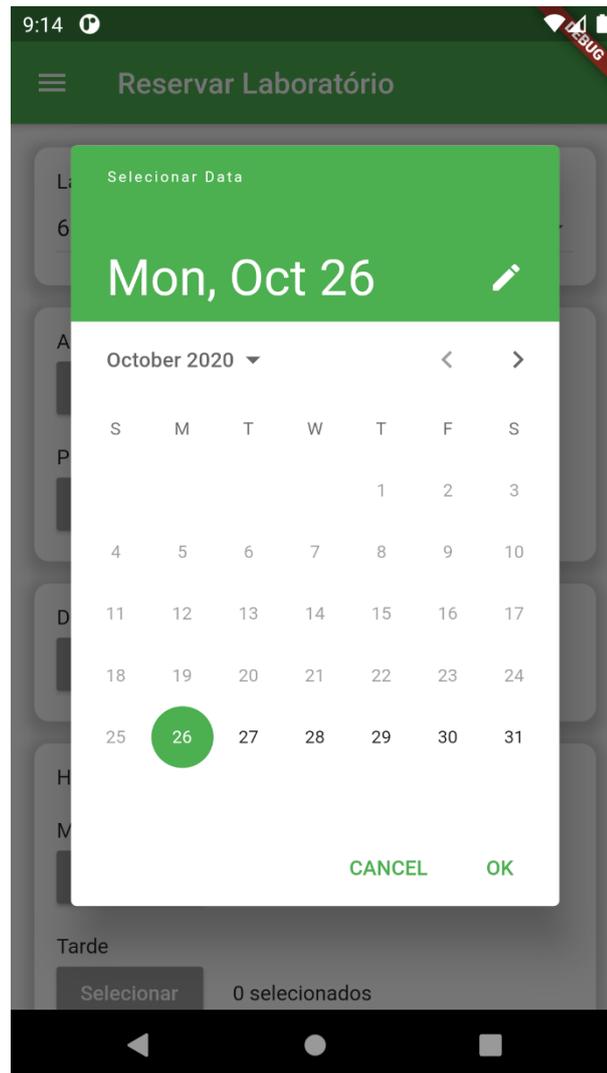


Fonte: Elaborado pelo autor, 2020.

Figura 37 – Mobile – Reservar laboratório (Professores)

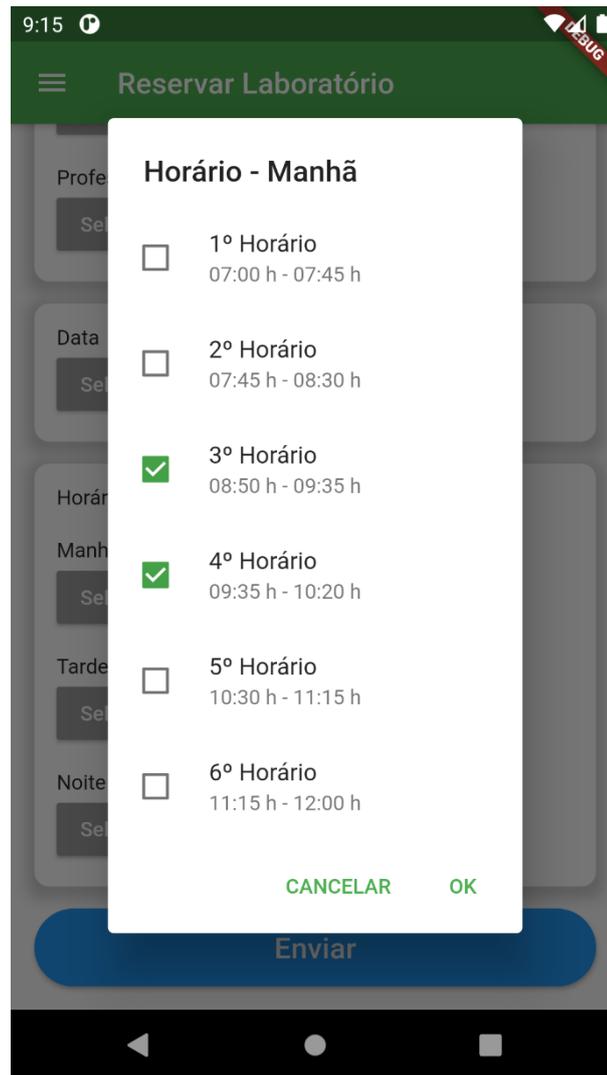


Fonte: Elaborado pelo autor, 2020.

Figura 38 – *Mobile* – Reservar laboratório (Selecionar data)

Fonte: Elaborado pelo autor, 2020.

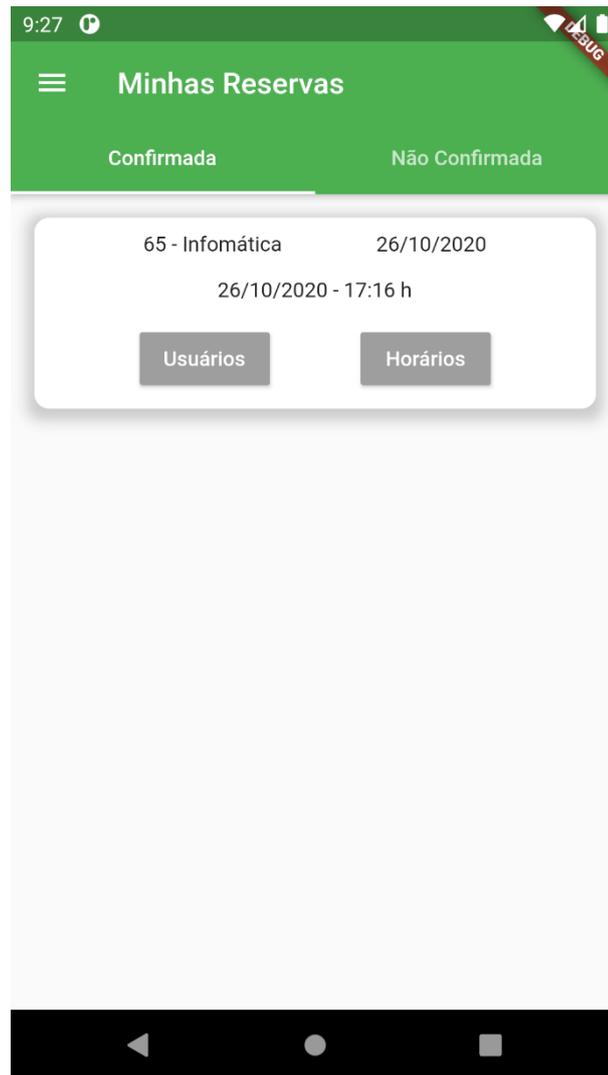
Figura 39 – Mobile – Reservar laboratório (Horários)



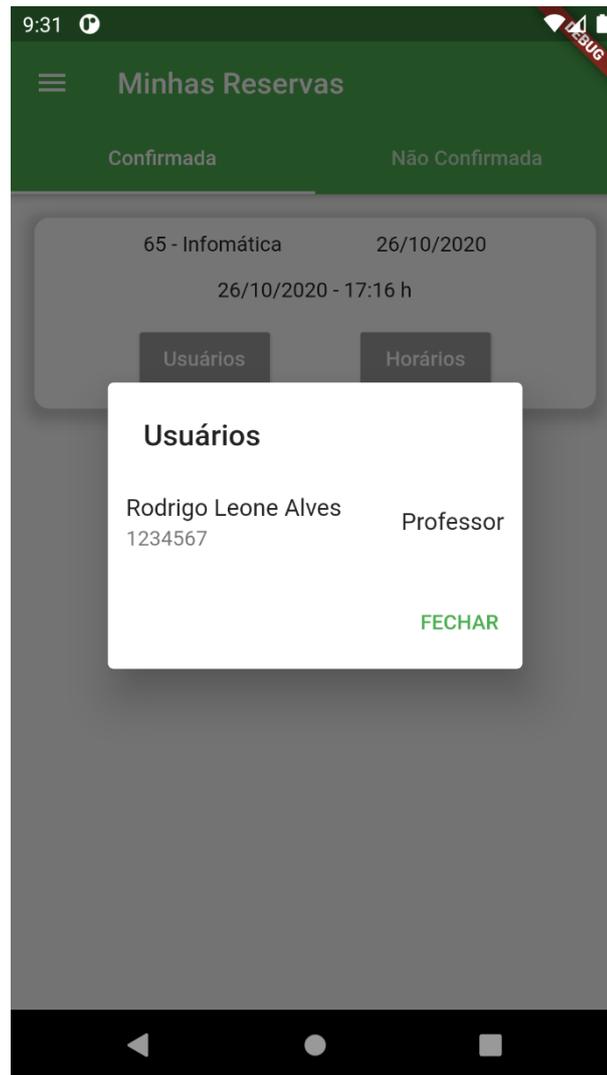
Fonte: Elaborado pelo autor, 2020.

Ao clicar em **Minhas Reservas**, é exibida a tela ilustrada pela Figura 40 com a lista de reservas confirmadas do usuário no qual é exibido o laboratório em questão e a data da reserva, abaixo a data e hora de sua criação e por fim o botão **Usuários** que exhibe todos os usuários incluídos na reserva ilustrada na Figura 41 e o botão **Horários** que exhibe todos os horários da solicitação com o status indicando se foi confirmado ou não (Figura 42).

Figura 40 – Mobile – Minhas reservas (Confirmada)



Fonte: Elaborado pelo autor, 2020.

Figura 41 – *Mobile* – Minhas reservas (Usuários)

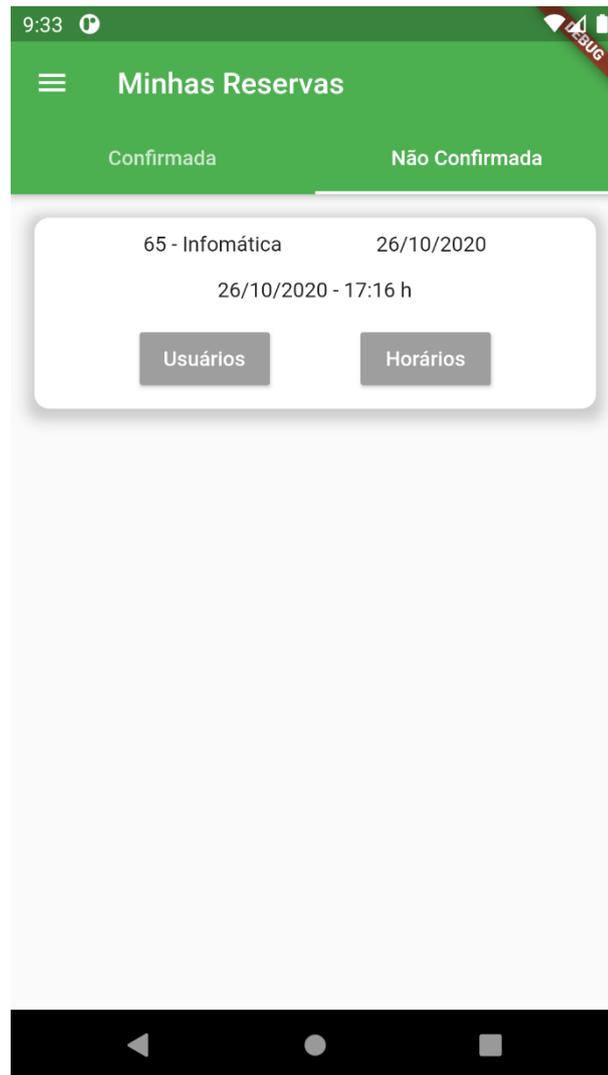
Fonte: Elaborado pelo autor, 2020.

Figura 42 – *Mobile* – Minhas reservas (Horários)

Fonte: Elaborado pelo autor, 2020.

Na Figura 43 é exibida a listagem das solicitações não confirmadas ao clicar em **Não Confirmada**, tendo os mesmos campos e ações da Figura 40.

Figura 43 – Mobile – Minhas reservas (Não confirmada)



Fonte: Elaborado pelo autor, 2020.

3.3.4. Sistema Embarcado

Utilizou-se a IDE Arduino para a codificação do sistema embarcado. Ela possui uma linguagem baseada em C++.

No algoritmo criado, está contido o código para conexão com a rede Wi-Fi e o serviço MQTT, o controle do servo motor, como também o recebimento e envio de dados para o *broker* pelo Wi-Fi. Todo o código desenvolvido está presente no APÊNDICE B – Código do sistema embarcado com comentários explicando cada trecho do código.

A comunicação com a rede Wi-Fi se dá pela biblioteca **ESP8266WiFi.h**, onde usa o nome da rede e senha para estabelecer a conexão. A comunicação com o serviço MQTT utiliza

a biblioteca **PubSubClient.h**, que utiliza o nome e porta do servidor para a conexão, e usuário e senha para autenticação. O controle do servo motor é gerenciado pela biblioteca **Servo.h**, onde coordena os comandos para mudar a posição do servo.

A estrutura do código contém dois métodos nativos da linguagem: **setup** e **loop**. O **setup** é executado uma única vez quando o ESP8266 é ligado, podendo conter todas as inicializações das funcionalidades e o método **loop** é executado em um loop infinito para a execução do código.

No método **setup**, foi iniciada a comunicação serial do ESP8266 para exibir mensagens contidas no código, como também a comunicação com o servo motor e a inicialização da conexão Wi-Fi e do serviço MQTT. O método **loop** contém o método do MQTT para manter a conexão e receber os dados.

Ao iniciar a comunicação com o MQTT, também é assinado o tópico **67** que referencia o laboratório com número **67** ao qual será aplicado o controle de abertura e fechamento. As mensagens dos tópicos assinados são recebidas pelo método **callback** que é utilizado pela biblioteca MQTT. Este método verifica a mensagem recebida do tópico (se **1** abrirá a fechadura, se **0** fechará a fechadura) e encaminha para o método que controla o servo.

Com o recebimento da mensagem, o método **acionarServo** verifica se o comando corresponde a abrir ou fechar o laboratório. Caso o comando seja verdadeiro, irá abrir a fechadura da porta posicionando o servo a 5 graus e enviando para o método **enviarMQTT** o comando **1**. Caso o comando seja falso, fechará a fechadura posicionando o servo a 175 graus e enviará o comando **0** para o método.

O método **enviarMQTT** é responsável por enviar uma confirmação se o ESP8266 recebeu o comando e efetuou a movimentação do servo. Ele envia ao *broker* a mensagem **1**, caso execute o comando e **0**, caso contrário, para o tópico **67c**. Esse tópico possui o *retain* ativado, ou seja, mantém no *broker* a última mensagem enviada, pois, caso outro dispositivo inicie e esteja assinando o tópico, receberá mensagem.

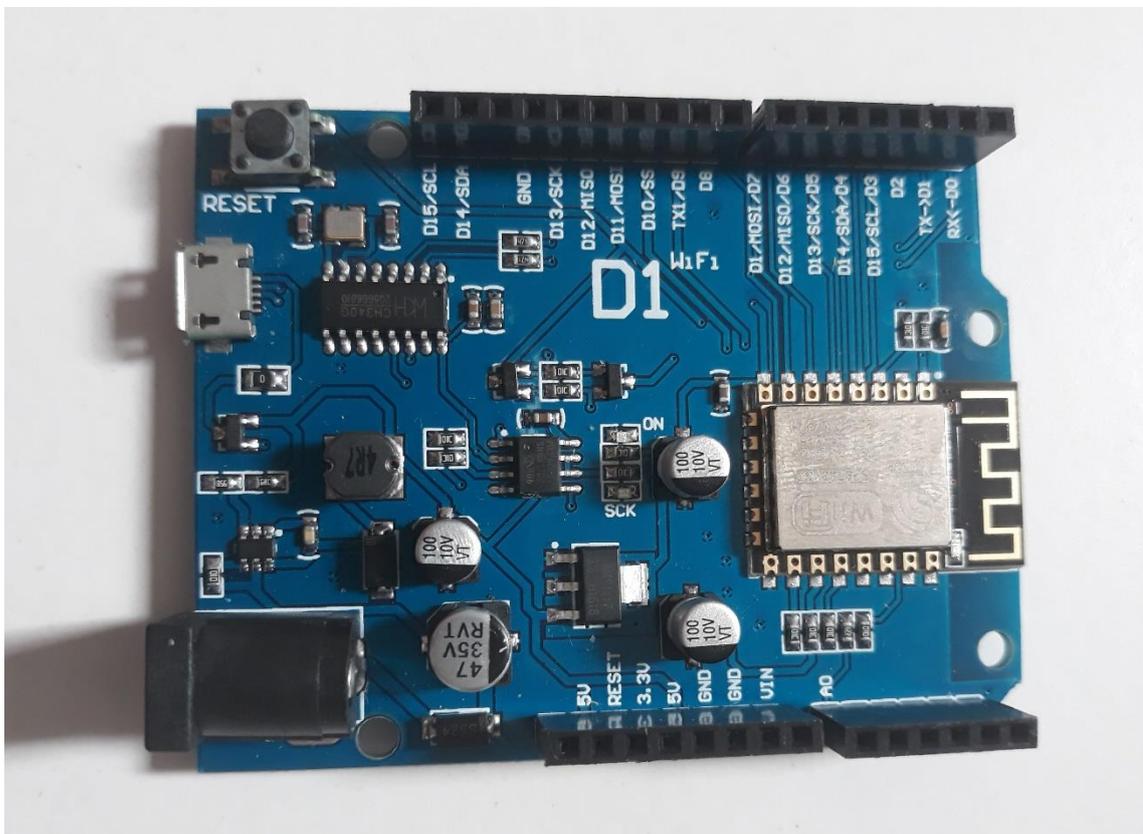
3.4. HARDWARE

Para a confecção do hardware foram utilizados o WeMos D1, uma placa de prototipagem que contém o ESP8266, o servo motor, a bateria de 9V e elaborada uma representação da fechadura da porta.

Na Figura 44 é ilustrado o WeMos D1 que possui algumas características relevantes para o projeto, como o padrão de transmissão *Institute of Electrical and Electronic Engineers* (IEEE) 802.11 b/g/n que possibilita a conexão Wi-Fi, ser alimentado com 9V a 12V *Direct Current* (DC) e possuir 11 pinos de entradas e saídas digitais e 1 analógico.

O WeMos D1 é utilizado para comportar o código que realiza a comunicação Wi-Fi e MQTT e o controle do servo motor. É alimentado por uma bateria de 9V onde também alimenta o servo motor pelos pinos **3.3V** e **GND** e para a comunicação, o sinal de controle do servo motor é conectado ao pino de entrada e saída digital **TX1/D9**.

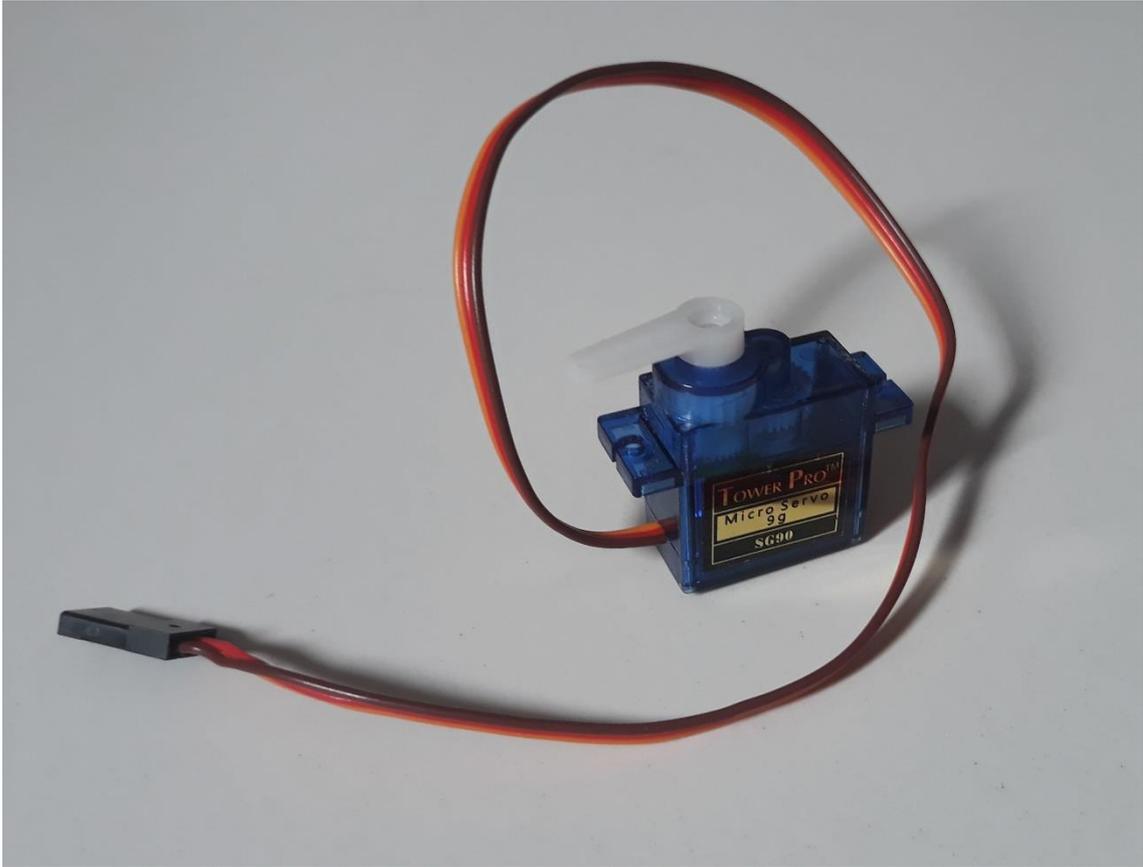
Figura 44 – Hardware – WeMos D1



Fonte: Elaborado pelo autor, 2020.

A Figura 45 exibe o servo motor que é responsável por realizar a abertura e o fechamento da fechadura. Ao receber o comando de abrir ou fechar do WeMos D1, o servo motor movimenta seu eixo para posicionar a fechadura de acordo com o comando recebido.

Figura 45 – Hardware – Servo motor



Fonte: Elaborado pelo autor, 2020.

Na Figura 46 é ilustrada a bateria de 9V que é utilizada para alimentar o hardware e o manter em funcionamento sem a necessidade de conexão com a rede elétrica, transformando-o em um dispositivo móvel para a utilização na porta.

Figura 46 – Hardware – Bateria 9V

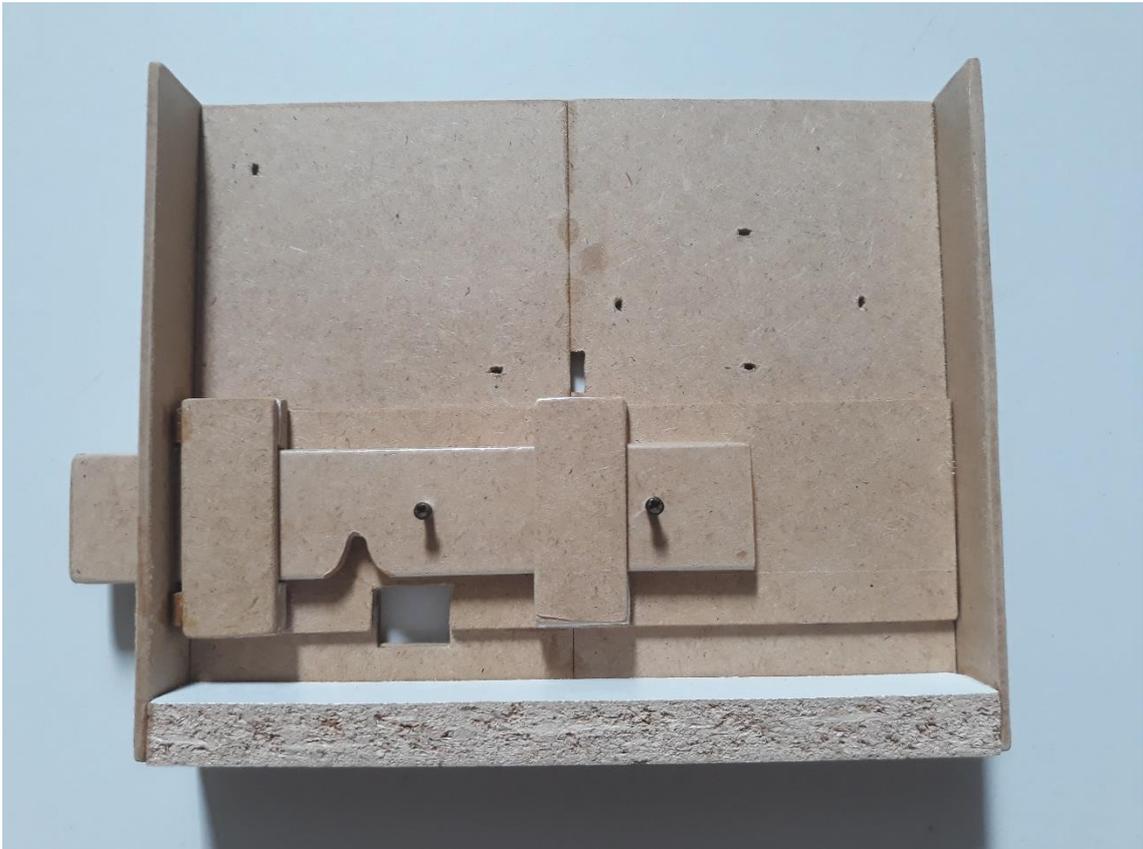


Fonte: Elaborado pelo autor, 2020.

A Figura 47 exhibe a representação da fechadura contida na porta de cada laboratório, confeccionada com *Medium Density Fiberboard* (MDF) e *Medium Density Particle* (MDP) ilustra a utilização do mecanismo de abertura e fechamento.

O projeto não descarta a possibilidade de utilização da chave, pois, caso o hardware venha a falhar ou seja removido para manutenção, o laboratório deve ser capaz de ser utilizado. Assim, permanecerá funcionando simultaneamente pelo uso da chave e do hardware, caso ocorra algum empecilho com o hardware, poderá ser utilizada a chave. No lado de dentro do laboratório se encontra o hardware para controlar a fechadura e no lado de fora, permanecerá com o mecanismo da chave.

Figura 47 – Hardware – Representação da fechadura



Fonte: Elaborado pelo autor, 2020.

3.5. COMUNICAÇÃO MQTT

A Figura 48 descreve a comunicação MQTT do laboratório 67, entre todos os componentes do projeto.

O fluxo da comunicação para abrir o laboratório se inicia com o dispositivo *Mobile* escaneando o número do laboratório através do QR Code presente na porta. Após o escaneamento, são enviados ao *Back-end*, através da rota do padrão REST, o identificador da reserva em questão (**1**), o número do laboratório escaneado (**67**), e o comando para abrir (**1**).

Com a chegada desses dados no *Back-end*, é verificado se a reserva possui o laboratório escaneado e se o horário do escaneamento corresponde aos horários confirmados para a reserva. Após a verificação, é publicado no tópico **67** a mensagem **1** para indicar a abertura do laboratório. Essa publicação não possui o *retain* ativo, pois não necessita que os dispositivos assinantes desse tópico recebam a mensagem ao se conectar com o broker.

O próximo componente no fluxo da comunicação é o **Hardware** presente no laboratório 67. Ele é configurado para assinar o tópico **67**⁶, e após o envio do **Back-end** é recebida a mensagem **1** do tópico **67** vinda do *broker*. O método presente no hardware para o recebimento das mensagens verifica qual foi a mensagem recebida e executa o método para acionar o servo e abrir o laboratório. Após a abertura, é enviada ao *broker* o tópico **67c** com a mensagem **1** para confirmar aos assinantes desse tópico a abertura do laboratório. Essa publicação possui o *retain* ativo, pois os dispositivos assinantes desse tópico necessitam receber a confirmação.

O dispositivo **Mobile** assina todos os tópicos de confirmação⁷ dos laboratórios presentes nas reservas do dia. Assim, recebe a mensagem **1** do tópico **67c** e altera, na reserva, o estado do botão para **Fechar**. No sistema **Front-end** é assinado o tópico de confirmação de todos os laboratórios cadastrados para que sejam capazes de modificar o estado do laboratório na interface gráfica. Com isso, ao receber a mensagem **1** do tópico **67c**, é alterado o estado para **ABERTO** na janela do laboratório 67.

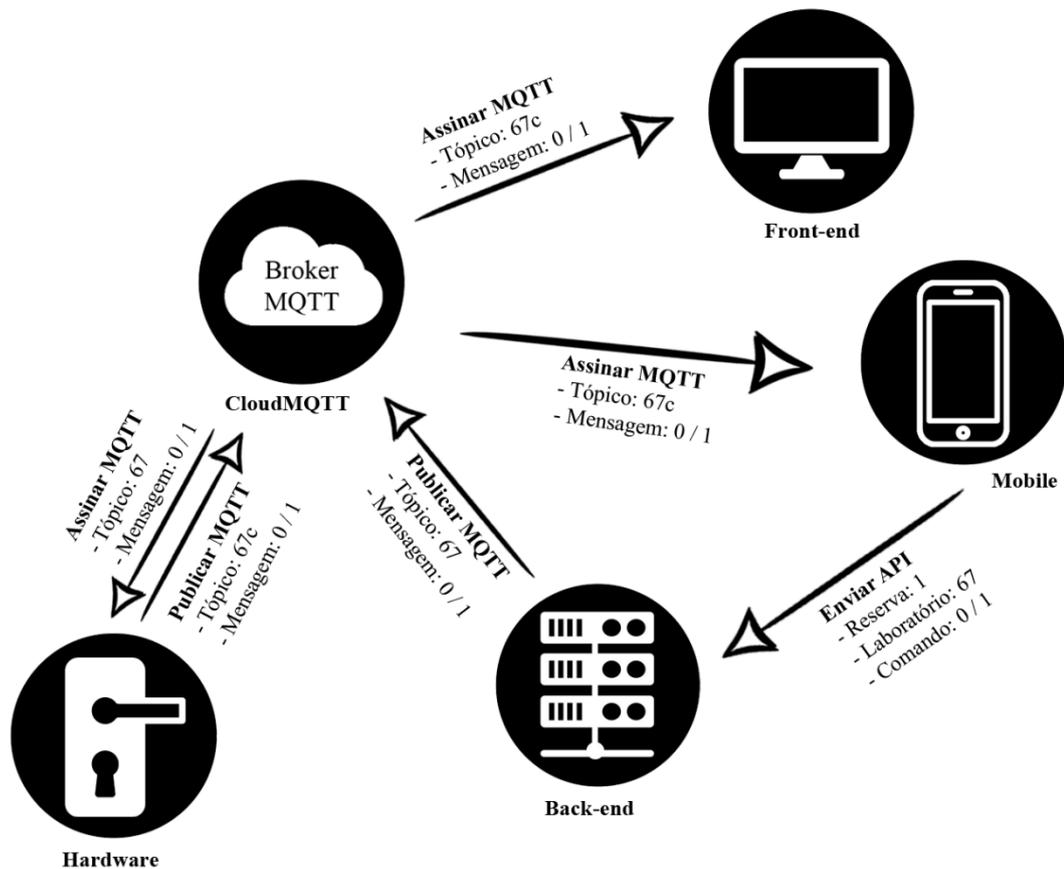
No fluxo da comunicação para fechar o laboratório, é utilizado o caractere **0** na mensagem. O usuário no sistema **Mobile** clica no botão **Fechar** referente ao laboratório desejado e envia ao **Back-end**, através da mesma rota REST, o identificador da reserva em questão (**1**), o número do laboratório (**67**) e o comando para fechar (**0**), para realizar a mesma verificação. Caso os dados estejam corretos, é publicado no tópico **67** a mensagem **0** para indicar o fechamento do laboratório.

O hardware assinante desse tópico recebe a mensagem **0** e aciona o servo para fechar o laboratório. Após a realização da tarefa, é enviado ao *broker* o tópico **67c** com a mensagem **0** para confirmar o fechamento do laboratório. Os sistemas **Mobile** e **Front-end** são assinantes desse tópico e ao receber a mensagem **0**, mudam o estado do laboratório.

⁶ Cada hardware é configurado para assinar somente o tópico referente ao laboratório onde se encontra instalado.

⁷ Todos os tópicos de confirmação utilizam como padrão o número do laboratório acrescido do caractere **c**.

Figura 48 – Comunicação MQTT



Fonte: Elaborado pelo autor, 2020.

3.6. TESTES

Foram realizados testes com os sistemas, o hardware e as comunicações Wi-Fi e MQTT, buscando corrigir falhas e obter melhoramentos em cada componente. A seguir, uma breve descrição da realização dos testes em cada componente do projeto.

No sistema *Back-end*, foram realizados os testes em suas rotas com o programa Postman, buscando encontrar falhas em suas funcionalidades e corrigi-las, como também na comunicação MQTT, onde são verificados os dados vindos do sistema *Mobile* e enviados ao *broker*.

No sistema *Front-end*, executou-se testes na responsividade das interfaces gráficas, em que é possível ter acesso pelo aparelho celular, na comunicação com a API, para obter o pleno funcionamento das funcionalidades e na assinatura dos tópicos referente a cada laboratório para sua mudança de estado.

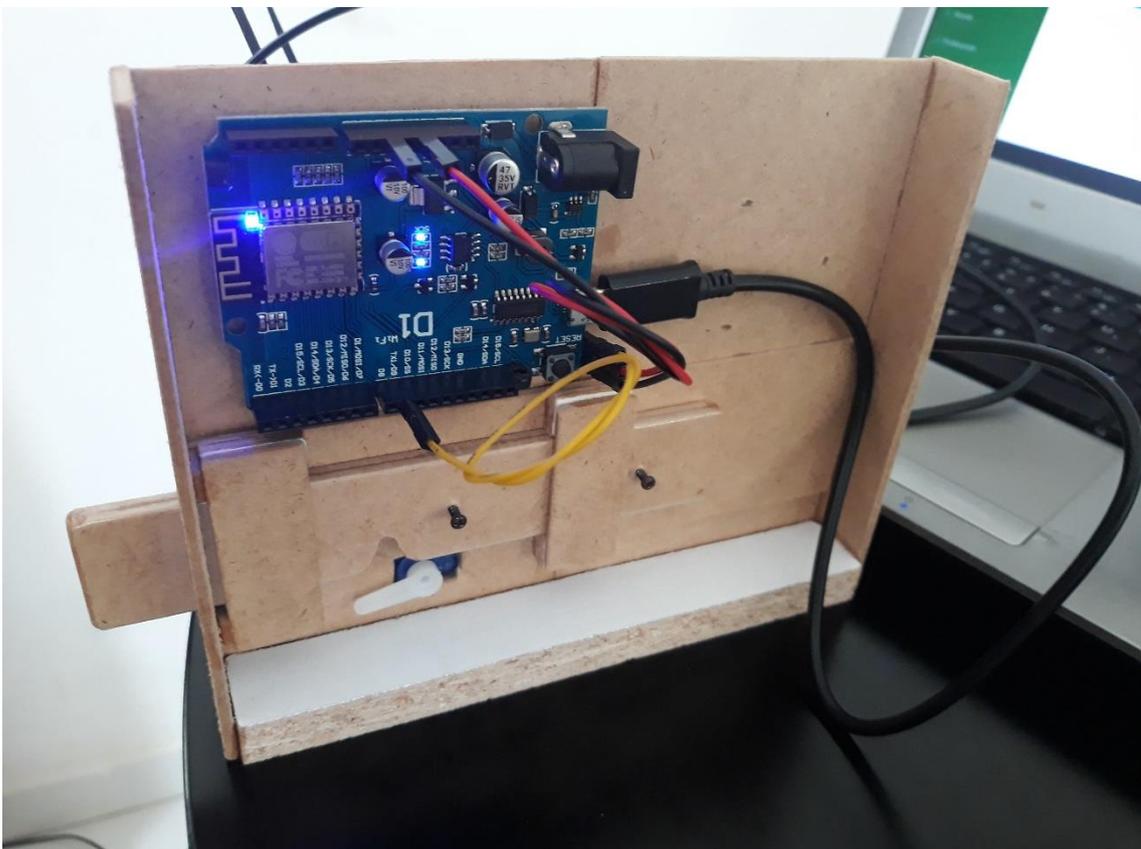
No sistema *Mobile*, foram realizados testes na comunicação com as funcionalidades presentes na API, no escaneamento do código QR empregado em cada laboratório e na comunicação MQTT para assinar os tópicos dos laboratórios.

No ESP8266 com seu algoritmo embarcado foram feitos testes com a comunicação Wi-Fi e MQTT, controle do servo motor onde verificou qual ângulo deve se movimentar para abrir e fechar a fechadura e na assinatura do tópico referente ao laboratório, como também na publicação da confirmação.

Na representação da fechadura, realizou-se testes para verificar se o servo motor é capaz de movimentar a fechadura. A Figura 49 ilustra o WeMos D1 recebendo o algoritmo e controlando o servo motor para movimentar a fechadura.

Após a finalização de todos os componentes, foram realizados testes simulando uma utilização real dos sistemas, visando encontrar falhas na utilização ou pontos de melhoria no sistema.

Figura 49 – Teste do protótipo



Fonte: Elaborado pelo autor, 2020.

4. RESULTADOS E DISCUSSÕES

Com os sistemas e o hardware implantados no Campus, será dispensável o uso da chave dos laboratórios e sua abertura e fechamento se dará pelo uso do hardware. Os sistemas gerenciarão os horários e reservas de cada usuário, se comunicando com o hardware através do serviço MQTT para enviar e receber dados.

O sistema *Front-end* possui todas as funcionalidades necessárias em interfaces intuitivas e de fácil utilização para realizar o gerenciamento dos laboratórios, tanto no computador quanto no celular.

O sistema *Mobile* destinado a alunos e professores detém um método facilitado para identificação do laboratório no momento de sua utilização por meio do escaneamento do código QR, possuindo um ambiente para realizar as solicitações e consultá-las, não sendo necessário se dirigir ao sistema *Front-end* para realizar essas ações.

Os sistemas possuem conexão com o hardware através do protocolo MQTT, sendo capaz de funcionar em uma rede com alta latência e dificuldades de conexão e possibilitando que ocorram menos falhas na comunicação em redes com baixa conectividade.

Os testes realizados foram finalizados com êxito, sendo corrigidas todas as falhas presentes e atualizados os pontos de melhoramento.

5. CONSIDERAÇÕES FINAIS

Após diversas pesquisas, escolha de hardware e software e seu desenvolvimento, como também os testes para a concretização do projeto, observou-se que o objetivo de desenvolver um sistema para o gerenciamento das solicitações e o mecanismo de abertura e fechamento das portas foi alcançado.

O gerenciamento das reservas e horários podem ser gerenciados pelo sistema *Front-end*, o sistema *Mobile* realiza a operação de abrir e fechar o laboratório no dia e horário designado, o *Back-end* manipula e gerencia os dados para submeter ao banco de dados e o hardware manipula a fechadura da porta para realizar a operação designada.

REFERÊNCIAS

ANDRADE, A. P. D. O que é Laravel? **TreinaWeb**, 2019. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-laravel/>>. Acesso em: 19 Novembro 2020.

ANDRADE, A. P. D. O que é Flutter? **TreinaWeb**, 2020. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-flutter/>>. Acesso em: 19 Novembro 2020.

ANDRION, R. Você sabe o que é o QR Code? A gente explica. **Olhar Digital**, 2019. Disponível em: <https://olhardigital.com.br/fique_seguro/noticia/voce-sabe-o-que-e-o-qr-code-a-gente-explica/90319>. Acesso em: 19 Novembro 2020.

ASSUNÇÃO, A. ESP8266 – Introdução E Primeiros Passos. **Laboratório iMobilis**, 2016. Disponível em: <<http://www2.decom.ufop.br/imobilis/esp8266-introducao-e-primeiros-passos/>>. Acesso em: 14 Dezembro 2020.

CLICK, G. O que é automação de processos, quando e como iniciar? **Gestão Click**, 2020. Disponível em: <<https://gestaoclick.com.br/blog/automacao-de-processos>>. Acesso em: 13 Dezembro 2020.

COMPOSER. Getting Started. **Composer**, [201-?]. Disponível em: <<https://getcomposer.org/doc/00-intro.md>>. Acesso em: 03 Dezembro 2020.

D3T. PORQUE UTILIZAMOS O FRAMEWORK PHP LARAVEL. **D3T**, [2019?]. Disponível em: <<https://d3t.com.br/blog/porque-utilizamos-o-framework-php-laravel>>. Acesso em: 19 Novembro 2020.

DOCUSIGN. Entenda o que é automação empresarial e faça do jeito certo. **DocuSing**, 2018. Disponível em: <<https://www.docusign.com.br/blog/entenda-o-que-e-automacao-empresarial>>. Acesso em: 03 Novembro 2020.

ENGPROCESS. Você sabe o que é MQTT? Descubra aqui! **EngProcess**, 2018. Disponível em: <<https://engprocess.com.br/mqtt/>>. Acesso em: 19 Novembro 2020.

GUEDES, M. O que é Dart? **TreinaWeb**, 2019. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-dart/>>. Acesso em: 19 Novembro 2020.

LARAVEL. Encryption. **Laravel**, [201-?]. Disponível em: <<https://laravel.com/docs/8.x/encryption>>. Acesso em: 20 Novembro 2020.

MQTT. MQTT: The Standard for IoT Messaging. **MQTT**, 2020. Disponível em: <<https://mqtt.org/>>. Acesso em: 13 Dezembro 2020.

NOLETO, C. O que é framework, como ele funciona e quais as suas vantagens? **Trybe**, 2020. Disponível em: <<https://blog.betrybe.com/framework-de-programacao/o-que-e-framework/>>. Acesso em: 13 Dezembro 2020.

OLIVEIRA, R. R. **USO DO MICROCONTROLADOR ESP8266 PARA AUTOMAÇÃO RESIDENCIAL**. Projeto de Graduação - Curso de Engenharia de Controle e Automação - Universidade Federal do Rio de Janeiro, Escola Politécnica. Rio de Janeiro, p. 42. 2017.

ORACLE. IoT. **Oracle Brasil**, 2020. Disponível em: <<https://www.oracle.com/br/internet-of-things/what-is-iot/>>. Acesso em: 13 Dezembro 2020.

PICOLLO, L. Vue JS: o que é, como funciona e vantagens. **geek hunter**, 2020. Disponível em: <<https://blog.geekhunter.com.br/vue-js-so-vejo-vantagens-e-voce/>>. Acesso em: 19 Novembro 2020.

PISA, P. O que é e como usar o MySQL? **TechTudo**, 2012. Disponível em: <<https://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>>. Acesso em: 23 Novembro 2020.

PRNEWSWIRE. Nível de automação no Brasil cresce 4%. **exame.**, 2019. Disponível em: <<https://exame.com/negocios/releases/nivel-de-automacao-no-brasil-cresce-4/>>. Acesso em: 05 Novembro 2020.

PROOF. Internet das Coisas: Um Desenho do Futuro. **PROOF**, [201-?]. Disponível em: <<https://www.proof.com.br/blog/internet-das-coisas/>>. Acesso em: 13 Dezembro 2020.

REVELO. Framework: saiba como usar e quais são os mais populares. **Revelo**, 2020. Disponível em: <<https://blog.revelo.com.br/o-que-e-framework-exemplos-e-aplicacoes/>>. Acesso em: 13 Dezembro 2010.

RIBEIRO, M. Automação de processos sem complicação. **Pluga**, 2016. Disponível em: <<https://pluga.co/blog/gestao-empresarial/automacao-de-processos/>>. Acesso em: 13 Dezembro 2020.

SCHULTZ, F. Conheça as 7 vantagens da automatização de processos. **bomcontrole**, 2020. Disponível em: <<https://bomcontrole.com.br/vantagens-automatizacao-de-processos/>>. Acesso em: 05 Novembro 2020.

SIGNIFICADOS. Significado de Automação. **Significados**, 2016. Disponível em: <<https://www.significados.com.br/automacao/>>. Acesso em: 13 Dezembro 2020.

SOUZA, G. O que é automação industrial? Qual o seu impacto nas indústrias? **Acoplast Brasil**, 2020. Disponível em: <<https://blog.acoplastbrasil.com.br/automacao-industrial/>>. Acesso em: 13 Dezembro 2020.

YUAN, M. Conhecendo o MQTT. **IBM**, 2017. Disponível em: <<https://developer.ibm.com/br/technologies/iot/articles/iot-mqtt-why-good-for-iot/>>. Acesso em: 13 Dezembro 2020.

APÊNDICE A – Rotas da API

Na primeira coluna da tabela está descrito cada verbo *HyperText Transfer Protocol* (HTTP), no qual é responsável para informar a API qual ação deve tomar, cadastrar (POST), atualizar (PUT), remover (DELETE) ou consultar (GET).

Na segunda coluna está descrito as rotas referentes a cada funcionalidade onde é precedido de (http://127.0.0.1:8080/api) para criação da URL. O símbolo ({id}) presente em algumas rotas, refere-se a uma variável onde recebe o identificador do objeto ao qual a rota pertence.

E na terceira e última coluna, uma breve descrição de cada rota.

VERBO HTTP	ROTA	DESCRIÇÃO
POST	/autenticar	Autenticar o usuário
POST	/sair	Sair da aplicação
GET	/aluno/consultar	Consultar todos os alunos
GET	/professor/consultar	Consultar todos os professores
GET	/gerenciador/consultar	Consultar todos os gerenciadores
POST	/laboratorio/cadastrar	Cadastrar um laboratório
PUT	/laboratorio/atualizar/{id}	Atualizar um laboratório específico
DELETE	/laboratorio/remover/{id}	Remover um laboratório específico
GET	/laboratorio/consultar	Consultar todos os laboratórios
GET	/laboratorio/consultar/horario	Consultar todos os horários do dia de cada laboratório
POST	/reserva/cadastrar	Cadastrar uma reserva
POST	/reserva/confirmar/{id}	Confirmar uma reserva específica

GET	/reserva/preencher	Consultar todos os dados para preencher o formulário de reservar laboratório
POST	/reserva/abrir-fechar	Recebe dados do sistema <i>Mobile</i> para verificação e envio para o MQTT
GET	/reserva/consultar	Consultar todas as reservas
GET	/reserva/consultar/minha	Consultar todas as reservas de um determinado usuário
GET	/reserva/consultar/confirmar	Consultar todas as reservas para realizar a confirmação
GET	/reserva/consultar/abrir	Consultar todas as reservas confirmadas no dia para abrir os laboratórios
GET	/reserva/selecionar/{id}	Consultar uma reserva específica
GET	/reserva/selecionar/confirmar/{id}	Consultar uma reserva específica para realizar a confirmação dos horários

APÊNDICE B – Código do sistema embarcado

```
#include <Servo.h> // Biblioteca para controle do serve motor

#include <ESP8266WiFi.h> // Biblioteca para conexão WI-Fi

#include <PubSubClient.h> // Biblioteca para conexão MQTT

Servo servo; // Objeto da biblioteca do servo

WiFiClient espClient; // Objeto da biblioteca do Wi-Fi

PubSubClient client(espClient); // Objeto da biblioteca do MQTT

const char* ssid = "rede_wifi"; // Nome da rede Wi-Fi que irá se conectar

const char* password = "senha_wifi"; // Senha da rede Wi-Fi

void setup() { // Método da linguagem onde inicializa os componentes e a comunicação

  Serial.begin(115200); // Velocidade da comunicação serial com o ESP

  servo.attach(2); // Porta do ESP para sinal do servo motor

  iniciarWifi(); // Chamada do método de inicialização da comunicação Wi-Fi

  client.setServer("servidor_mqtt", "porta_servidor_mqtt"); // Inicialização da conexão com o
servidor MQTT

  client.setCallback(callback); // Inicialização do método de callback (resposta do servidor)

  iniciarMqtt(); // Chamada do método para autenticação e comunicação com o servidor MQTT

}

void loop() { // Método da linguagem onde realiza a execução das funcionalidades

  if (!client.connected()) { // Verifica se o cliente MQTT está conectado
```

```
    reconectarMQTT(); // Se não estiver conectado, chama o método para reconectar
}

client.loop(); // Mantém a conexão com o servidor MQTT
}

void iniciarWifi() { // Método de inicialização da comunicação Wi-Fi

    delay(10); // Espera de 10 milissegundos para prosseguir

    // Imprime mensagem

    Serial.println();

    Serial.print("Conectando...");

    WiFi.begin(ssid, password); // Fornecer nome da rede e senha Wi-Fi para iniciar conexão

    while (WiFi.status() != WL_CONNECTED) { // Verifica se Wi-Fi está conectado

        delay(500); // Espera de 500 milissegundos para prosseguir

        // Imprime mensagem

        Serial.print(".");

    }

    randomSeed(micros()); // Inicializa o gerador de números aleatórios com o número de
    microssegundos passados desde que a placa ESP começou a executar

    // Imprimir mensagem

    Serial.println("");

    Serial.println("WiFi Conectado.");

    Serial.println("IP: ");

    Serial.println(WiFi.localIP()); // Imprimir IP que recebeu na conexão

}
```

```

void iniciarMqtt() { // Método para autenticação e comunicação com o servidor MQTT

  if (!client.connected()) { // Verifica se MQTT está conectado

    Serial.println("Conectando ao MQTT..."); // Imprime mensagem

    if (client.connect("identificacao_dispositivo", "usuario_mqtt", "senha_usuario_mqtt")) { //
Verifica e autentica com o servidor MQTT

      Serial.println("MQTT conectado."); // Imprime mensagem

      client.subscribe("67"); // Método para se inscrever (subscribe) em um tópico "67" para
receber as mensagens

    } else {

      // Se não conectar exibe a mensagem

      Serial.println("Falha ao conectar");

      Serial.print("Falha, rc=");

      Serial.print(client.state());

      Serial.println(" tente novamente em 5 segundos");

      delay(1000); // Espera 1 segundo

    }

  }

}

```

```

void acionarServo(boolean abrir) { // Método para acionamento do servo

  if(abrir) { // Verifica se o servo vai abrir a fechadura

    delay(10); // Espera 10 milissegundos

    servo.write(5); // Posiciona o servo a 5 graus

    enviarMQTT("1"); // Chamada do método para enviar mensagem "1" pelo MQTT

    Serial.println("Aberto"); // Imprime mensagem

```

```
} else { // Verifica se o servo vai fechar a fechadura

    delay(10); // Espera 10 milissegundos

    servo.write(175); // Posiciona o servo a 175 graus

    enviarMQTT("0"); // Chamada do método para enviar mensagem "0" pelo MQTT

    Serial.println("Fechado"); // Imprime mensagem

}

}

void enviarMQTT(char* comando) { // Método para enviar mensagem pelo MQTT

    client.publish("67c",comando, true); // Método para publicar (publish) em um tópico "67c"
    com a mensagem na variável "comando" e retain (manter mensagem) verdadeiro

}

void callback(char* topic, byte* payload, unsigned int length) { // Método de callback (resposta
do servidor)

    // Imprime mensagem

    Serial.print("Mensagem do tópico [");

    Serial.print(topic);

    Serial.print("] ");

    // Concatena caracteres da resposta do tópico para imprimir

    for (int i = 0; i < length; i++) {

        Serial.print((char)payload[i]);

    }

    Serial.println();
```

```
if ((char)payload[0] == '1') { // Verifica se resposta foi "1"  
    acionarServo(true); // Chama método para acionar servo com parâmetro "true"  
} else if ((char)payload[0] == '0') { // Verifica se resposta foi "0"  
    acionarServo(false); // Chama método para acionar servo com parâmetro "false"  
}  
}
```

```
void reconectarMQTT() { // Método para reconectar MQTT  
    while (!client.connected()) { // Verifica se MQTT está conectado  
        iniciarMqtt(); // Chama método para iniciar MQTT  
    }  
}
```

APÊNDICE C – Imagem do hardware completo

