

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
RONDÔNIA**

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

TRABALHO DE CONCLUSÃO DE CURSO

**INFORMATIZAÇÃO DAS ATIVIDADES DESEMPENHADAS POR CONTROLES
DE PORTÕES ELETRÔNICOS**

ALEX SILVA

Ji-Paraná, 2023

DESENVOLVIMENTO DE PRODUTO CIENTÍFICO E TECNOLÓGICO

INFORMATIZAÇÃO DAS ATIVIDADES DESEMPENHADAS POR CONTROLES DE PORTÕES ELETRÔNICOS

ALEX SILVA

Trabalho de Conclusão de Curso apresentado ao curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, como parte dos requisitos necessários à obtenção do título de tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Jackson Henrique da Silva Bezerra

Ji-Paraná, 2023

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO,
com dados informados pelo(a) próprio(a) autor(a).

Silva, Alex.

Informatização das atividades desempenhadas por controles de portões eletrônicos
/ Alex Silva, Ji-Paraná-RO, 2023.

127 f. : il.

Orientador(a): Prof. Me. Jackson Henrique da Silva Bezerra.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia
de Rondônia - IFRO, Ji-Paraná-RO, 2023.

1. Portão eletrônico. 2. Desenvolvimento. 3. Automação residencial. 4. IoT. 5.
Inovação. 6. Controle remoto. I. Bezerra, Jackson Henrique da Silva (orient.). II.
Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

Bibliotecário(a) Responsável: Cleuza Diogo Antunes, CRB-11/864 (Campus Ji-Paraná)

ALEX SILVA

**INFORMATIZAÇÃO DAS ATIVIDADES DESEMPENHADAS POR CONTROLES
DE PORTÕES ELETRÔNICOS**

Trabalho de Conclusão de Curso apresentado ao curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, como parte dos requisitos necessários à obtenção do título de tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Jackson Henrique da Silva Bezerra

Aprovado pela Banca Examinadora em 07 de dezembro de 2023.

BANCA EXAMINADORA

Orientador:

Prof. Me. Jackson Henrique da Silva Bezerra

Membro da Comissão de TCC:

Prof. Dr. Wanderson Roger Azevedo Dias

Membro da Comissão de TCC:

Prof. Esp. Jefferson Antonio dos Santos

DEDICATÓRIA

Este trabalho é dedicado à minha família e amigos, cujo apoio e incentivo foram essenciais durante esta jornada. Além disso, dedico este trabalho a você, caro leitor, e a todos os entusiastas da tecnologia, cuja paixão impulsiona constantemente a inovação e o progresso.

AGRADECIMENTOS

Quero expressar minha profunda gratidão a Deus, e a todos que me ajudaram durante esta jornada. À minha amada Luana M. Engelhardt, por sua ajuda na construção da maquete, e seu constante apoio em todo o processo foram vitais para a realização deste trabalho. Ao meu amigo André S. Aquilau, pela troca de conhecimento que tivemos, sua contribuição e sua ajuda no desenvolvimento do servidor. Ao meu amigo Rafael C. Rosina, por me incentivar e me apoiar em pilares que foram fundamentais para a minha persistência e sucesso neste trabalho.

À minha mãe Duvirgens L. C. Silva, por seu apoio incondicional, amor e encorajamento. Ao meu pai Moisés J. Silva, que desde cedo despertou em mim a paixão pela eletrônica e engenharia. À minha irmã Patricia Silva, por ter me inscrito neste curso de Análise e Desenvolvimento de Sistemas e ser minha inspiração. À minha irmã Eliceia Silva, pela contribuição significativa ao refinamento deste trabalho, com suas ideias e críticas que foram fundamentais para o resultado obtido. À minha irmã Regiane A. Silva, por sua ajuda e apoio. Ao meu sobrinho Arthur S. C. Santos, pela colaboração nos testes do projeto.

Aos meus professores que compartilharam conhecimento e me direcionaram. Em especial ao meu professor orientador Me. Jackson H. S. Bezerra, meu profundo reconhecimento por sua dedicação e seu direcionamento no desenvolvimento desta monografia. E por fim, ao Instituto Federal de Educação, Ciências e Tecnologia de Rondônia – *Campus Ji-Paraná*.

A todos, meus sinceros agradecimentos por terem feito parte desta minha trajetória acadêmica, enriquecendo-a com suas presenças, contribuições e apoio inestimável.

*“Não creio que haja uma emoção, mais intensa
para um inventor do que ver suas criações
funcionando. Essa emoção faz você esquecer de
comer, de dormir, de tudo.”*

Nikola Tesla

RESUMO

Os portões eletrônicos, tornaram-se elementos fundamentais na segurança e privacidade de residências e empresas. Com as atuais tecnologias, e os conceitos de Internet das Coisas (IoT) e Automação Residencial, novas possibilidades surgiram para a inovação desses sistemas. Este trabalho tem por objetivo não apenas explorar o funcionamento dos atuais portões eletrônicos, mas também as tendências que buscam aprimorar sua funcionalidade. Com isso, optou-se por uma investigação qualitativa embasada na abordagem de estudo de caso, na perspectiva descritiva. Porém, no âmbito do desenvolvimento de software, a metodologia *Scrum* foi a estratégia adotada. Os resultados dessa pesquisa foram alcançados por meio da elaboração de uma aplicação e um dispositivo, que juntos tornam os portões eletrônicos convencionais mais inteligentes. Nesta aplicação, os usuários conseguem operar remotamente seus portões, visualizar o histórico de acionamentos, controlar o acesso, entre outras funcionalidades. Isso torna os portões eletrônicos mais interativos e evidencia o potencial do projeto.

Palavras-chave: Portão eletrônico. Desenvolvimento. Automação residencial. IoT. Inovação. Controle remoto.

ABSTRACT

The electronic gates have become fundamental elements in the security and privacy of homes and businesses. With current technologies and the concepts of the Internet of Things (IoT) and Home Automation, new possibilities have emerged for innovating these systems. This work aims not only to explore the functioning of current electronic gates but also the trends seeking to enhance their functionality. Therefore, a qualitative investigation based on a case study approach in a descriptive perspective was chosen. However, in the realm of software development, the Scrum methodology was the adopted strategy. The results of this research were achieved through the development of an application and a device, which together make conventional electronic gates smarter. In this application, users can remotely operate their gates, view activation history, control access, among other functionalities. This makes electronic gates more interactive and highlights the potential of the project.

Keywords: Electronic gate. Development. Home automation. IoT. Innovation. Remote control.

LISTA DE FIGURAS

Figura 1 - Elementos de um portão eletrônico	21
Figura 2 - SONOFF Basic e interface de seu aplicativo	23
Figura 3 - Contatto Wi-Fi	24
Figura 4 - Módulo Garagem Izzy Open.....	25
Figura 5 - Arduino UNO	26
Figura 6 - Placa ESP32 DevKitC	27
Figura 7 - Imagem da Arquitetura de Comunicação via Bluetooth.....	39
Figura 8 - Arquitetura Completa da Aplicação	40
Figura 9 - Protótipo do <i>hardware</i> em fase de testes	43
Figura 10 - Diagrama esquemático do hardware.....	44
Figura 11 - Materiais e componentes selecionados para a construção do <i>hardware</i>	46
Figura 12 - <i>Hardware</i> finalizado	47
Figura 13 - Teste do botão <i>reset</i> e circuito de acionamento do portão.....	48
Figura 14 - Teste do sensor para obter o estado do portão	48
Figura 15 - <i>Hardware</i> ligado na réplica do portão eletrônico	50
Figura 16 - Instalação do <i>hardware</i> no portão eletrônico	51
Figura 17 - Solicitação de permissões de acesso a recursos do sistema.....	52
Figura 18 - Convite para finalização de registro recebido por e-mail	54
Figura 19 - Telas de finalização de registro do usuário.....	54
Figura 20 - <i>Login</i> do usuário	55
Figura 21 - Processo de conexão com um portão	57
Figura 22 - Controlar um portão.....	58
Figura 23 - Funcionamento dos botões de controle do portão	59
Figura 24 - Menu principal.....	60
Figura 25 - Perfil do usuário (Visualização, edição e confirmação de edição).....	61
Figura 26 - Exclusão do próprio perfil de usuário.....	62
Figura 27 - Tela de usuários registrados e tela para convidar novos usuários	63
Figura 28 - Controle de acesso dos usuários	64
Figura 29 - Telas com detalhes das operações realizadas pelo portão	65
Figura 30 - Tela de configurações do portão.....	66

LISTA DE TABELAS

Tabela 1 - Características do Arduino UNO e ESP32.....	27
Tabela 2 - Lista de materiais e componentes para construção do <i>hardware</i>	45

LISTA DE QUADROS

Quadro 1 - Diferentes perspectivas de estudo de caso	36
Quadro 2 - Detalhes das <i>sprints</i> do projeto	37

LISTA DE ABREVIATURAS E SIGLAS

- ADC - Analog-to-Digital Converter (Conversor Analógico-Digital)
- API - Application Programming Interface (Interface de Programação de Aplicativos)
- DIY - Do it Yourself (Faça Você Mesmo)
- GND - Ground (Terra)
- GPIO - General-Purpose Input/Output (Entrada/Saída de Propósito Geral)
- HTTP - Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto)
- IDE - Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
- IoT - Internet of Things (Internet das Coisas)
- JWT - JSON Web Token (Token Web JSON)
- JSON - JavaScript Object Notation (Notação de Objetos JavaScript)
- LED - Light-Emitting Diode (Diodo Emissor de Luz)
- MQTT - Message Queuing Telemetry Transport (Transporte de Telemetria de Mensagens em Filas)
- REST - Representational State Transfer (Transferência de Estado Representacional)
- SGBD - Sistema de Gerenciamento de Banco de Dados
- SGBDOR - Sistema Gerenciador de Banco de Dados Objeto-Relacional
- SQL - Structured Query Language (Linguagem de Consulta Estruturada)
- UML - Unified Modeling Language (Linguagem de Modelagem Unificada)
- VCC - Voltage Common Collector (Coletor Comum de Tensão)
- XP - Extreme Programming (Programação Extrema)

SUMÁRIO

1. INTRODUÇÃO	15
1.1. Justificativa	15
1.2. Hipótese	16
1.3. Objetivos	16
1.3.1. Objetivo Geral	16
1.3.2. Objetivos Específicos	17
2. REFERENCIAL TEÓRICO	18
2.1. Informatização e Automatização de Processos	18
2.2. Sistema Computacional e Suas Aplicações	18
2.3. Internet das Coisas (IoT).....	19
2.4. Automação Residencial	20
2.5. Portão Eletrônico	21
2.6. Tecnologias IoT para Automação de Portões Eletrônicos.....	22
2.6.1. SONOFF Basic	22
2.6.2. Contatto Wi-Fi.....	23
2.6.3. Módulo Garagem Izzy Open	24
2.7. Placas de Prototipagem e Desenvolvimento	25
2.8. Processo de Desenvolvimento de <i>Software</i>	28
2.8.1. Metodologia de Desenvolvimento Ágil <i>Scrum</i>	29
2.9. Tecnologias Utilizadas no Desenvolvimento do Produto.....	31
2.9.1. Jira <i>Software</i>	31
2.9.2. Figma	31
2.9.3. Astah.....	31
2.9.4. Br Modelo.....	32
2.9.5. PostgreSQL.....	32
2.10. Programação.....	32
2.10.1. Editores de Código e Ambientes de Desenvolvimento Integrado (IDEs)	33
2.10.2. Controle de Versão	33
2.10.3. Desenvolvimento do <i>firmware</i> com C/C++.....	34
2.10.4. Desenvolvimento do <i>front-end</i> com Dart e Flutter	34
2.10.5. Desenvolvimento do <i>back-end</i> com C# e .NET	35
3. METODOLOGIA.....	36
3.1. Metodologia de Pesquisa	36

3.2.	Metodologia de Desenvolvimento de Projeto.....	37
4.	RESULTADOS E DISCUSSÕES	39
4.1.	Resultados da Arquitetura e Componentes do Projeto	39
4.1.1.	Arquitetura de Comunicação	39
4.1.2.	Arquitetura da Aplicação.....	39
4.2.	Resultados do Desenvolvimento do <i>Hardware</i>	43
4.3.	Resultados do Desenvolvimento do <i>Firmware</i>	47
4.4.	Ligação do <i>Hardware</i> no Portão Eletrônico	50
4.5.	Resultados do Desenvolvimento do <i>Software</i>	52
4.5.1.	Primeiro acesso ao aplicativo	52
4.5.2.	Conexão com o servidor	53
4.5.3.	Finalização de registro.....	54
4.5.4.	Realização do <i>login</i>	55
4.5.5.	Tela principal.....	56
4.5.6.	Perfil do usuário	61
4.5.1.	Usuários registrados	63
4.5.2.	Histórico de operações.....	65
4.5.3.	Configuração do portão	66
5.	CONCLUSÃO.....	67
	REFERÊNCIAS	69
	APÊNDICE A – PROJETO DE SOFTWARE	74

1. INTRODUÇÃO

Com o avanço das tecnologias na atualidade, a sociedade testemunha a rápida evolução de dispositivos e sistemas que moldam significativamente nosso cotidiano. Nesse contexto, os portões eletrônicos deslizantes, elementos fundamentais na segurança e praticidade residencial, estão sujeitos a aprimoramentos que refletem as inovações contemporâneas. Observa-se a necessidade de integrar os benefícios proporcionados pela automação residencial e pela Internet das Coisas (IoT) a esses dispositivos.

A presente pesquisa visa explorar e desenvolver uma aplicação inovadora para o gerenciamento de portões eletrônicos deslizantes, permitindo seu controle por meio de dispositivos móveis. Ao identificar as limitações dos atuais sistemas disponíveis, a proposta busca superar alguns obstáculos. A introdução de um dispositivo eletrônico específico, interligado à central do portão, possibilitará uma comunicação eficiente e sem fio com dispositivos móveis, tornando-se uma solução mais acessível.

1.1. Justificativa

Com o mundo moderno, o surgimento de novas tecnologias é constante, isso abre um leque de ideias e oportunidades, o que possibilita importantes avanços em nosso modo de vida. Ao analisar o funcionamento dos atuais portões eletrônicos, observa-se que é possível melhorá-los com a adição dos novos recursos, e tecnologias que se tornaram bastante populares na atualidade. Sendo assim, esta pesquisa tem como foco desenvolver e apresentar uma nova aplicação e dispositivo, para o gerenciamento das atividades dos portões eletrônicos deslizantes, garantindo seu controle a partir de dispositivos móveis, por meio de conceitos de automação residencial e Internet das Coisas.

É importante destacar que, atualmente já existe alguns dispositivos de funcionalidade semelhante disponíveis ao consumidor, porém não funcionam sem conexão com a internet. Para que seja possível controlar o portão diretamente através de dispositivos móveis, será desenvolvido um dispositivo eletrônico para intermediar a comunicação entre a central do portão eletrônico e os dispositivos móveis. Sendo assim, o dispositivo será interligado fisicamente, direto na central do portão eletrônico, já os comandos do dispositivo móvel pelo aplicativo serão enviados e recebidos por conexão sem fio.

A escolha do microcontrolador ESP32 para o desenvolvimento do *hardware*, foi feita levando em consideração seu baixo custo, e principalmente o suporte a múltiplas conexões sem fio (*wi-fi* e *bluetooth*), isso permitirá que o usuário opere o portão eletrônico mesmo na falta de conexão com a internet, sendo utilizado o *bluetooth* para configuração do portão e como segunda opção de controle. Além disso, outras funcionalidades também serão implementadas, por exemplo: abertura parcial do portão por um botão de controle deslizante, controle de acesso, apresentação de status do portão, histórico de acesso, entre outras.

A distância até o portão deixará de ser uma limitação, pois dispositivos conectados funcionam de qualquer local que tenha conexão com a internet. Pode-se dizer que, como o controle e utilização do portão eletrônico se dará por meio de dispositivos móveis do próprio utilizador, o processo de configuração e inclusão de novos usuários será mais simplificado, até mesmo dispensará a necessidade de aquisição de novos controles do próprio portão. Sendo assim, além do que já foi mencionado, este trabalho se justifica por produzir uma nova tecnologia para beneficiar a sociedade mediante a automação residencial e a IoT.

1.2. Hipótese

Com o uso da tecnologia desenvolvida neste projeto, os usuários conseguirão controlar seus portões eletrônicos pelo celular. Será mais fácil a inclusão de novos usuários para o uso do portão eletrônico, sem gerar custos adicionais por usuário com a compra e a configuração de novos controles. Será registrado o histórico de acionamento do portão, indicando o portão, o responsável e a data e hora. E o portão poderá ser acionado pelo celular mesmo que não tenha conexão de internet.

1.3. Objetivos

1.3.1. Objetivo Geral

Desenvolver um sistema para a informatização das atividades desempenhadas por controles de portões eletrônicos.

1.3.2. Objetivos Específicos

- Revisar o estado da arte referente aos recursos e tecnologias IoT necessárias para o desenvolvimento do projeto;
- Desenvolver um protótipo de *hardware* para integrar na placa controladora do motor do portão eletrônico;
- Analisar, projetar, desenvolver e testar um *firmware* para realizar a comunicação do protótipo de *hardware* com o portão eletrônico.
- Analisar, projetar, desenvolver e testar aplicativo móvel para realizar a abertura do portão eletrônico;
- Analisar, projetar, desenvolver e testar o servidor da aplicação;

2. REFERENCIAL TEÓRICO

2.1. Informatização e Automatização de Processos

As revoluções sempre causaram grandes avanços e mudanças na sociedade, alterando permanentemente o modo de vida das pessoas. Um período marcante foi a revolução industrial, que teve início em 1760, com a construção de máquinas à vapor e o desenvolvimento dos primeiros sistemas mecânicos. Com o domínio da eletricidade, foram criadas as primeiras linhas de montagens, com isso foi possível construir produtos de maneira massiva a partir da automatização dos processos de fabricação. Logo depois, foram desenvolvidas as primeiras tecnologias baseadas nos semicondutores, o que possibilitou o início da era computacional com os dispositivos analógicos e digitais (SCHWAB, 2016, p. 18).

Mesmo que previamente os sistemas computacionais tenham sido desenvolvidos no passado com foco nas indústrias para ganhos de lucratividade, atualmente permite a interação entre empresas e seus consumidores globalmente (MORAES *et al.*, 2018, p. 42). A consolidação desses avanços tecnológicos permitiu o início da informatização de atividades, que antes eram realizadas de maneira repetitiva e manual, sendo assim, *softwares* puderam ser desenvolvidos através de algoritmos e a inteligência artificial, para automatizar processos e atividades. De acordo com Cavalheiro *et al.* (2018, p. 3), hoje o mundo está todo interligado, e isso proporcionou avanços jamais vistos anteriormente na história.

2.2. Sistema Computacional e Suas Aplicações

Um sistema computacional é composto por diversos elementos que, basicamente tem por finalidade a execução de cálculos e tomada de decisões mediante aos dados de entrada. O computador é o produto de todo esse sistema, pode realizar diversos cálculos instantaneamente e são ótimos na repetição de tarefas. Formado por três partes principais: dados, *software* e *hardware*. Em resumo, os dados são os elementos de entrada e saída do sistema computacional, o *software* é a parte lógica também conhecido como programa, já o *hardware* é a parte física do computador, onde se pode tocar, por exemplo: teclado, fios, placas eletrônicas, monitor, entre outros componentes (CARVALHO e LORENA, 2017, p. 61).

Aprofundando mais sobre *software*, Pressman (2011, p. 32-33) também afirma que *software* é um componente lógico, porém diz que o mesmo está muito além de ser apenas

instruções e programas de manipulação de informações como dito na maioria das vezes. Portanto, é preciso aprofundar mais, diferenciando-o de características particulares de outros itens construídos pelo ser humano. Diferente do *hardware* que é fabricado, o *software* é desenvolvido e não se degrada, pois não está sujeito às variáveis ambientais. Grande parte dos *softwares* ainda são feitos de maneira integral e customizada, o que dificulta a reutilização de suas partes, porém isso já tem mudado bastante. O autor ainda menciona o seguinte:

O *software* distribui o produto mais importante de nossa era — a informação. Ele transforma dados pessoais (por exemplo, transações financeiras de um indivíduo) de modo que possam ser mais úteis num determinado contexto; gerencia informações comerciais para aumentar a competitividade; fornece um portal para redes mundiais de informação (Internet) e os meios para obter informações sob todas as suas formas (PRESSMAN, 2011, p. 31).

Carvalho e Lorena (2017, p. 43 e 96), ainda falam sobre algumas particularidades do *hardware*, se tratando de sistemas digitais podem apenas entender e guardar em memória os estados ligado e desligado. Essa é a única linguagem padrão que os computadores são capazes de compreender em baixo nível, ou seja, linguagem binária. Sendo assim, toda a informação do sistema computacional que se pode ver facilmente em um contexto de alto nível, em uma camada mais profunda são representações de sequências de *bits* 0 (desligado) ou 1 (Ligado). Quando um *hardware* é projetado para uma atividade específica, com restrições de memória e espaço físico, são denominados de sistemas embarcados.

Contudo, existe ainda o *firmware*, um elemento do sistema computacional muito usado em sistemas embarcados e conceito de IoT. É capaz de realizar operações a partir de instruções de baixo nível diretamente no *hardware* de um equipamento eletrônico, como se fosse seu sistema operacional. As instruções são gravadas diretamente em um circuito integrado de memória durante a fabricação do dispositivo nas indústrias, por exemplo, memórias ROM, EEPROM e memória *flash* (SALLES, 2017, p. 24). A diferenciação de *software* e *firmware* de acordo com Costa *et al.* (2014, p. 40) se dá pelo *software* ser o meio para que o usuário possa interagir com o sistema, já o *firmware* é essencial para a interação do sistema com o meio físico.

2.3. Internet das Coisas (IoT)

O termo IoT (do inglês *Internet of Things*) que significa Internet das Coisas, surgiu com a finalidade de conectar na internet os dispositivos e equipamentos do dia a dia. Tornou possível a interligação e interação com elementos variados trazendo flexibilidade e dinamismo entre máquinas e a rede mundial de internet, com resultados rápidos e praticamente na palma da mão

de seus usuários a partir dos *smartphones*. Na área da educação, os impactos tem sido muito mais notáveis, já que maior parte dos alunos cresceram em contato direto com as tecnologias e costumam aprender na maioria das vezes com mais facilidade no contexto prático. Algumas ferramentas de prototipagem como o Arduino se destacaram na concepção de IoT no meio educacional (HORST e CORNENONSI, 2020, p. 471).

Ferrari *et al.* (2021, p. 25) diz que Internet das Coisas não parte de uma mesma tecnologia e sim a união de tecnologias distintas e variadas, capazes de interligar meios físicos e virtuais através de sensores e atuadores, formando um sistema coordenado de compartilhamento de informações por meio de redes, protocolos e aplicações de internet. Portanto pode-se entender que o termo está ligado diretamente a uma tecnologia inteligente, que pode interagir em ambiente e reagir a eventos recebidos do meio físico e digital. Alguns exemplos de dispositivos que podem usar essa tecnologia são: cafeteiras, relógios inteligentes, robôs aspiradores, TVs, entre outros (ALMEIDA *et al.*, 2019, p. 2).

2.4. Automação Residencial

O surgimento dos dispositivos conectados fez com que a automação residencial pudesse evoluir aceleradamente, principalmente devido a expansão da internet e o uso de *smartphones* ter se tornado bastante comum, o que trouxe mais praticidade no cotidiano das pessoas, em seus lares e até mesmo longe de casa. Através da IoT tornou-se possível por meio da automatização residencial controlar dispositivos da residência de qualquer lugar com conexão de internet, sendo uma tecnologia mais usada através dos *smartphones* por meio de aplicativos *mobile* ou até mesmo via *web browser* (NUNES, 2018, p. 14 e 17). O autor ainda informa alguns outros benefícios:

A automação residencial visa trazer maior conforto, segurança e praticidade, utilizando de dispositivos eletrônicos para facilitar a gestão dos recursos residenciais, conforme a necessidade do morador. Essa gestão, de forma autônoma ou manual, se dá por meio do monitoramento do ambiente por câmeras, por sensores de temperatura, umidade, do controle de acesso, do sistema de iluminação, dispositivos eletrônicos e outros (NUNES, 2018, p. 14).

A popularização da automatização residencial instigou o interesse de pessoas ao redor do mundo, isso motivou o desenvolvimento de diversos dispositivos para a realização das tarefas residenciais de maneira automática. Com isso, algumas tecnologias puderam ser barateadas, ficando mais acessíveis para os consumidores em geral, por consequência esse mercado se expandiu significativamente. Portanto, surgiram diversos dispositivos que

automatizam tarefas, desde as mais simples até as mais avançadas, como por exemplo: controle de eletrodomésticos, dimerização de lâmpadas, controle de cortinas e até mesmo o controle de portões eletrônicos (SANTOS e LARA JUNIOR, 2019, p. 10).

2.5. Portão Eletrônico

Seu uso é muito comum atualmente, pois possibilita mais comodidade, segurança e agilidade ao entrar e sair dos lugares. O sistema de um portão eletrônico funciona da seguinte maneira. Por meio do pressionamento de um botão no controle, é emitido um sinal via radiofrequência para a central de comando. A partir do sinal recebido, a central aciona o motor elétrico, que por meio da rotação de seu eixo, gira engrenagens da caixa de redução. Com isso, se reduz a velocidade aumentando o torque. Por fim, o portão é aberto ou fechado com facilidade (SANTOS e LARA JUNIOR, 2019, p. 29).

Figura 1 - Elementos de um portão eletrônico



Fonte: Site de venda Tudo Forte¹.

Não são todos os portões eletrônicos que são desenvolvidos já com alguma tecnologia de automação residencial integrada em seu sistema, isso significa que seu uso só é possível

¹ Disponível em: <<https://www.tudoforte.com.br/motor-de-portao/motor-de-portao-deslizante/kit-motor-completo/kit-motor-portao-rossi-dz-nano-turbo-600kg-14-deslizante-automatico-de-correr-eletronico-com-abertura-rapida-cremalheira>>. Acesso em: 27 mai. 2022.

mediante acionamento manual por meio do controle de radiofrequência do próprio portão. No entanto, esses portões convencionais apresentam algumas limitações, sendo elas: defeitos nos controles, gastos extras na aquisição de novos controles, o acionamento só funciona se o usuário estiver próximo do portão e para a configuração de novos controles é preciso que um profissional técnico se desloque até a residência do cliente, entre outras.

2.6. Tecnologias IoT para Automação de Portões Eletrônicos

Com a alta no mercado de automação residencial pelo aumento de interesse por parte dos consumidores e a expansão do IoT, diversos fabricantes ao redor do mundo desenvolveram novos produtos neste contexto (ROCHA JÚNIOR, 2014, p. 12). Com isso, emergiram várias tecnologias relacionadas à Internet das Coisas (IoT) que viabilizam a incorporação de portões eletrônicos convencionais aos sistemas de automação residencial. Esse avanço proporcionou não apenas a criação de portões eletrônicos equipados com essa tecnologia desde a sua fabricação, mas também abriu caminho para a adaptação de modelos tradicionais ao contexto da automação residencial, impulsionando a acessibilidade e a modernização desse mercado.

2.6.1. SONOFF Basic

É basicamente um interruptor inteligente, tecnologia desenvolvida pela empresa chinesa SONOFF, utiliza o conceito DIY (do inglês, *Do it Yourself*) que significa faça você mesmo, no idioma português. Por ter conectividade wi-fi, pode ser utilizado em diversos contextos da automação, aceitando comandos de voz e agendamento de ações automáticas, tudo pode ser realizado pelo próprio aplicativo chamado eWeLink (SONOFF, 2022). Souza *et al.* (2019, p. 13090) diz que o dispositivo foi desenvolvido utilizando basicamente um relé e um microcontrolador ESP8266EX produzido pela Espressif, versão anterior ao ESP32. Pode acionar cargas em corrente alternada de até 10A (Amperes) remotamente, de qualquer local com conexão de internet.

Figura 2 - SONOFF Basic e interface de seu aplicativo



Fonte: Site de vendas Amazon².

Observa-se que o dispositivo está fazendo sucesso na área de automação residencial, principalmente por ser uma das opções mais acessíveis aos consumidores em relação aos concorrentes, custando em média R\$ 50,00 reais. Porém, devido ser de propósito geral acaba sendo uma opção não muito viável para automação de portões, mesmo que atualmente seja utilizado para tal. Os principais motivos são: só é possível seu acionamento remoto se houver conexão com a internet, não é possível saber o estado do portão (aberto ou fechado) e o aplicativo não é desenvolvido implementando recursos específicos para gerenciar portões eletrônicos.

2.6.2. Contatto Wi-Fi

Desenvolvido pela PPA, uma indústria 100% brasileira que desenvolve eletrônicos para automatização de portas, portões e cancelas. O Contatto Wi-Fi permite controlar remotamente via internet cercas elétricas, portões eletrônicos e até mesmo alarmes. O controle é realizado pelo aplicativo da PPA chamado Contatto, além de possibilitar o controle do portão eletrônico e demais dispositivos, o aplicativo também oferece recursos de emitir relatórios e possibilita o monitoramento do *status* de abertura e fechamento do portão (PPA, 2022). O dispositivo

² Disponível em: <<https://www.amazon.com.br/Sonoff-Basic-Automa%C3%A7%C3%A3o-Residencial/dp/B077D3P9J2/>>. Acesso em: 29 mai. 2022.

também é fabricado utilizando um microcontrolador produzido pela Espressif, especificamente o ESP8266.

Figura 3 - Contatto Wi-Fi



Fonte: PPA (2022).

Nota-se que em relação ao SONOFF Basic, o Contatto Wi-Fi é uma opção que oferece funcionalidades e recursos bem mais específicos para a automatização de portões eletrônicos, pois foi desenvolvido pensando justamente no ramo desse tipo de produto, porém seu custo é um pouco mais alto se comparado ao anterior, sendo em média R\$ 125,00 reais. Contudo, o produto oferece algumas limitações, só funciona com acesso à internet e não permite definir privilégios, ou seja, todos os usuários do aplicativo podem mudar as configurações do portão.

2.6.3. Módulo Garagem Izzy Open

Fabricado pela AGL, uma empresa brasileira que atua no mercado de segurança eletrônica nacional e internacional. O módulo de garagem pode ser controlado por comandos de voz pelo Google Assistente, Alexa e Siri. É compatível com diversos motores de portões eletrônicos do mercado, foi desenvolvido especificamente para garagens, para controle da abertura e fechamento do portão, para isso deve ser instalado no celular o aplicativo AGL Home. Outros recursos que se destacam são: possibilidade de atribuir privilégios de acesso aos usuários, histórico de aberturas, apresentação do estado atual do portão e envio de notificações (AGLFECHADURAS, 2022).

Figura 4 - Módulo Garagem Izzy Open



Fonte: AGLFECHADURAS (2022).

Este foi o produto que se mostrou mais similar ao que é proposto neste projeto de pesquisa, o que se destaca em relação ao Contatto Wi-Fi é o recurso de definição de privilégios, interface e usabilidade do aplicativo, porém seu custo médio fica em torno de R\$ 160 reais. No entanto, essa opção também não permite o uso do portão caso falte internet no local onde o portão eletrônico está instalado ou no dispositivo do usuário. Essa é uma das limitações que podem ser resolvidas com o uso de placas eletrônicas de prototipagem, devido algumas oferecerem outros tipos de conectividade, o que abre portas para o desenvolvimento de uma nova solução para automação de portões eletrônicos.

2.7. Placas de Prototipagem e Desenvolvimento

A primeira a se popularizar foi a placa Arduino, e nada mais é que um *hardware* formado por uma placa de circuito impresso com um microcontrolador e seus componentes periféricos. A placa de prototipagem pode ser reproduzida por pessoas com conhecimento em eletrônica ou pode ser adquirida pronta para o desenvolvimento de projetos. Podem fazer leituras de sensores, estados de botões e até mesmo obter informações da internet através alguns módulos conectados a placa. Através disso, depois do processamento das entradas recebidas, pode enviar respostas e emitir sinais de saída de dados, como ligar um LED, acionar um motor e até mesmo mandar uma mensagem (SANTOS *et al.*, 2020, p. 72).

Figura 5 - Arduino UNO



Fonte: Site oficial do Arduino³.

O elemento principal do Arduino e qualquer placa de prototipagem é o microcontrolador, de acordo com Santos e Lara Junior (2019, p. 21) microcontroladores são componentes de baixo custo que são programáveis, possuem assim como todo sistema computacional um processador, memória, e portas de entrada e saída dentro de um mesmo encapsulamento. A programação do microcontrolador pode ser realizada por meio da IDE (do inglês *Integrated Development Environment*) do Arduino, uma plataforma desenvolvida especificamente para isso (SANTOS *et al.*, 2020, p. 72). Sendo assim, a programação desenvolvida é gravada no microcontrolador da placa e diz para o Arduino com base em instruções de máquina o que deve ser feito de acordo com o fluxo de execução do programa.

No ano de 2016 foi apresentado um novo microcontrolador chamado ESP32, projetado pela empresa Espressif Systems, sendo um dos mais rápidos e vantajosos do mercado, outra característica que o destacou bastante é pelo fato de já vir com conectividades wi-fi e bluetooth integradas em um mesmo chip (SANTOS e LARA JUNIOR, 2019, p. 23). Na Figura 6 é possível ver o microcontrolador ESP32, componente coberto por um encapsulamento metálico, com uma antena de conectividade wi-fi e bluetooth exposta no canto superior esquerdo. O componente está soldado sobre uma placa de prototipagem e desenvolvimento chamada ESP32 DevKitC, sendo uma placa também fabricada e fornecida pela própria Espressif Systems.

³ Disponível em: <<https://store.arduino.cc/products/arduino-uno-rev3>>. Acesso em: 27 mai. 2022.

Figura 6 - Placa ESP32 DevKitC



Fonte: Site FlipeFlop⁴.

Santos e Lara Junior (2019, p. 24) ainda destacam que o ESP32 é um dos poucos microcontroladores que apresentam duas tecnologias de transmissão sem fio, talvez até mesmo seja um dos únicos, tudo dentro de um único e pequeno encapsulamento. Mesmo assim, o *chip* ainda agrega outros recursos úteis, o processador tem dois núcleos, o que permite direcionar tarefas específicas para cada um deles, um bom exemplo é programar a execução principal para rodar em um dos núcleos e deixar o outro núcleo responsável por fazer leitura de sensores e guardar em memória. Koyanagi (2017) apresenta algumas características dos microcontroladores de cada uma das placas de prototipagem apresentadas (Ver Tabela 1).

Tabela 1 - Características do Arduino UNO e ESP32

Características	Arduino UNO	ESP32
Quantidade de Núcleos do Processador	1	2
Arquitetura do Processador	8 bits	32 bits
Velocidade de <i>Clock</i> do Processador	16 MHz	160 MHz
Wi-Fi Integrado	Não	Sim
Bluetooth Integrado	Não	Sim
Tamanho da Memória RAM	2 Kb	512 Kb
Tamanho da Memória <i>Flash</i>	32 Kb	16 Mb
Quantidade de GPIO	14	36
Quantidade de ADC	6	18

Fonte: Koyanagi (2017).

⁴ Disponível em: <<https://www.filipeflop.com/produto/modulo-wifi-esp32-bluetooth/>>. Acesso em: 27 mai. 2022.

O propósito de ambas as placas é semelhante, o que difere são seus recursos e performance. O custo de um ESP32 DevKitC está saindo em média R\$40,00 reais, já o custo do Arduino UNO está na média de R\$ 50,00 reais (MERCADO LIVRE, 2023). Estes valores foram obtidos através de uma pesquisa realizada em setembro de 2023 na plataforma de comércio eletrônico Mercado Livre. Porém, vale ressaltar que os preços podem variar ao longo do tempo e diferir de acordo com o vendedor e condições de mercado.

Em decorrência do que foi levantado, não fica dúvidas sobre a viabilidade da utilização do ESP32 para o desenvolvimento da nova solução para automatizar portões eletrônicos por meio da IoT. Sua conectividade bluetooth já integrada, permitirá resolver a atual limitação das tecnologias disponíveis atualmente, a de não funcionar na falta de internet.

2.8. Processo de Desenvolvimento de *Software*

Desenvolver um *software* é uma atividade complexa, além disso, envolve muitos profissionais, que dedicam longos períodos de tempo durante todo o processo. Sendo assim, é fundamental que sua gestão seja bem definida, e realizada por todos os envolvidos desde o seu planejamento. Compreende-se que é difícil ter total segurança se o plano de projeto está sendo desenvolvido da melhor maneira, de tal forma que entregue um resultado com qualidade mais próxima de acordo com o orçamento apresentado, porém é de extrema importância que a equipe seja constantemente encorajada, no objetivo de entregar ao cliente um produto de qualidade, conforme suas necessidades (PRESSMAN, 2011, p. 566).

Mesmo com uma gestão bem definida, a aplicação de uma metodologia de desenvolvimento é algo chave para uma entrega bem sucedida do projeto. Pressman (2011, p. 82) afirma que mudanças no escopo do projeto não saem barato, principalmente quando não são bem gerenciadas, sendo assim, o uso de uma metodologia de desenvolvimento ágil na execução de um projeto reduz os custos com mudanças no projeto. Quando se trata de metodologias ágeis, logo vem em mente o XP (Programação Extrema, do inglês *Extreme Programming*) e o *Scrum*, por serem as metodologias mais conhecidas e utilizadas no desenvolvimento de *software* atualmente.

Souza e Oliveira (2021, p. 140) fizeram um comparativo entre as metodologias ágeis XP e *Scrum*, os mesmos concluíram que apesar dos prós e contras de cada uma delas, é difícil definir qual a melhor, porém cada uma funcionará melhor de acordo com o tipo de projeto e necessidades. Segundo eles, o XP é mais aplicável ao desenvolvimento de *software*, com o foco

na satisfação do cliente com o produto, levando em conta o tempo gasto no projeto e a entrega de valor. Já o *Scrum* é mais amplo, não fica limitado apenas ao desenvolvimento de *software*, utiliza processos iterativos e incrementais, sendo entregue uma parte funcional ao cliente no final de cada ciclo.

O *Scrum* consegue se adaptar desde as pequenas empresas até as multinacionais, abrangendo várias áreas, desde desenvolvimento de *hardware*, passando pelo marketing e a gestão de pessoas, uma ferramenta simples para qualquer área (MACHADO JUNIOR *et al.*, 2019, p. 9).

Portanto, a metodologia ágil que se mostra mais viável e eficiente no desenvolvimento deste projeto é o *Scrum*, principalmente por ser mais ampla e ter aplicabilidade no desenvolvimento de outras partes do projeto, como por exemplo, no planejamento e desenvolvimento do *hardware* e interfaces. Também, por possibilitar o desenvolvimento de projetos com equipes de tamanhos reduzidos, garantir que as entregas sejam de qualidade e diminuir retrabalhos posteriormente. Outro ponto que contribuiu para a escolha dessa metodologia é a existência de ferramentas que auxiliam toda a equipe na aplicação do *Scrum* de maneira eficiente e flexível.

2.8.1. Metodologia de Desenvolvimento Ágil *Scrum*

Essa seção por inteiro foi escrita baseando-se no próprio Guia do *Scrum*, de acordo com as definições dos principais criadores dessa metodologia, sendo eles Schwaber e Sutherland (2020, p. 2).

A metodologia de desenvolvimento ágil *Scrum* é definida como um *framework* simples que ajuda a solucionar problemas complexos, foi criado por Ken Schwaber e Jeff Sutherland em 1990, sua primeira apresentação formal foi em 1995 em uma conferência, desde então, com a colaboração de vários indivíduos foi possível refinar a metodologia, que logo passou a ser bem difundida e conhecida por pessoas do mundo todo, principalmente depois da publicação do primeiro guia em 2010, chamado Guia do *Scrum*. Um fator essencial para a aplicação da metodologia ágil *Scrum* é a colaboração, todos os integrantes do projeto fazem parte do que é denominado *Scrum Team*.

Seus integrantes são divididos em três papéis fundamentais, cada um dos papéis define determinadas tarefas e ações que precisam ser realizadas. O *Scrum Master* lidera a equipe, remove impedimentos e treina o time, garantindo que todos os envolvidos entendam e apliquem os padrões do *Scrum*. O *Product Owner* é o que conhece o produto, deve garantir uma entrega

de alto valor ao cliente, precisa se comunicar bem para deixar claro as metas do produto e define os itens do *Product Backlog* sendo transparente com os envolvidos. Já os *Developers* são responsáveis por desenvolver os incrementos que serão entregues ao cliente no final da *Sprint*.

A *Sprint* é o item mais importante do *Scrum*, costuma ter a duração de 2 ou 4 semanas, nela que planejamento e ideias do projeto se transformam em valor. Durante a *Sprint* não pode ocorrer alterações que afetem negativamente sua entrega, porém sempre que necessário pode ser feito o refinamento do *Product Backlog*. Segue abaixo detalhes dos quatro eventos que acontecem durante uma *Sprint*:

- ***Sprint Planning***: Momento para *Product Owner* junto com os *Developers* definirem quais itens do *Product Backlog* devem ir para a *Sprint*, pode durar no máximo 8 horas e todos do *Scrum Team* precisam participar.
- ***Daily Scrum***: Os *Developers* falam basicamente o que foi feito, o que ainda será feito e comentam sobre os problemas que estão ocorrendo, a duração não deve ultrapassar 15 minutos.
- ***Sprint Review***: É o momento onde as partes interessadas avaliam o resultado da *Sprint*, é debatido o que será feito depois e o progresso do produto, a partir disso, se discute sobre futuras mudanças, deve durar no máximo 4 horas.
- ***Sprint Retrospective***: Evento para conversar e refletir sobre o que aconteceu na *Sprint*, o deve ser resolvido primeiro, os problemas que ocorreram e quais foram as abordagens que trouxeram bons resultados, não deve exceder 3 horas.

Os componentes que agregam valor ao produto são os artefatos, para que todos tenham clareza sobre que precisa ser feito é preciso que sejam construídos de forma transparente e objetiva. O *Product Backlog* é o primeiro artefato desenvolvido, é constituído de uma lista ordenada, composta por tudo o que precisa ser feito para desenvolver o produto e realizar melhorias. O *Sprint Backlog* é um artefato que conterà apenas os itens do *Product Backlog* definidos para a *Sprint*. Já o Incremento, é a soma de recursos da *Sprint* finalizada definida como pronta com as anteriores. É importante ressaltar que para facilitar a aplicação da metodologia ágil *Scrum* durante o desenvolvimento do produto, pode-se introduzir tecnologias que flexibilize o trabalho dos envolvidos, uma delas é a ferramenta Jira Software.

2.9. Tecnologias Utilizadas no Desenvolvimento do Produto

2.9.1. Jira Software

Criado e mantido pela Atlassian, o Jira *Software* é utilizado para o gerenciamento de projetos por equipes ágeis, pois oferece o painel de *Scrum* para definição e controle do fluxo das atividades. Os projetos criados na ferramenta são divididos em quadros, assim o projeto pode ser facilmente fracionado e gerenciado por toda a equipe durante o andamento das *Sprints*. Com os roteiros, é possível ter uma visualização ampla do andamento das atuais atividades, o que facilita a tomada rápida de decisões e sincronia de todos da equipe. Além disso, pode-se gerar relatórios em tempo real, o que agrega valor ao produto, pois os relatórios auxiliam na tomada de decisões orientada por dados (ATLASSIAN, 2022).

2.9.2. Figma

Antes de desenvolver as interfaces de uma aplicação é importante desenhar os protótipos das telas primeiramente, a vantagem da realização desse processo é a obtenção de uma prévia de como será a cara da aplicação para apresentar ao cliente, até mesmo com os protótipos prontos, o processo de desenvolvimento do *front-end* da aplicação se torna bem menos moroso. Uma ferramenta que tem esse propósito é o Figma, próprio para desenvolvimento de interfaces do usuário. É gratuito e está disponível para uso *online*, o que permite a colaboração de diversas pessoas no mesmo projeto durante a prototipagem e inspeção das telas (CORDEIRO *et al.*, 2021, p. 31).

2.9.3. Astah

Para o desenvolvimento de sistemas, na fase de projeto é essencial a criação de diagramas para que o time possa ter uma visão clara dos requisitos e objetivos a serem seguidos. O Astah é um *software* desenvolvido para isso, pois utiliza conceitos de UML (Linguagem de Modelagem Unificada, do inglês *Unified Modeling Language*) e permite a criação de diversos diagramas de modelagem de *softwares*. É uma ferramenta que permite criar diagramas de Caso de Uso, Classes, Sequência, Atividades e muitos outros (ASTAH, 2022). Se destaca em relação

as outras, devido a quantidade de recursos oferecidos, a ferramenta é utilizada tanto no meio profissional quanto no meio acadêmico (MORI, 2015, p. 13).

2.9.4. Br Modelo

Uma ferramenta de código aberto, que teve seu desenvolvimento iniciado em 2005, como produto de um trabalho de conclusão de curso, com o objetivo de oferecer uma ferramenta nacional de modelagem de banco de dados, seguindo os conceitos apresentados pelo Dr. Carlos Aberto Heuser na 4ª edição de seu livro “Projeto de Banco de Dados” (CÂNDIDO, C., 2005, p. 9). O *software* utiliza abordagem relacional, permite a criação dos esquemas de banco de dados conceitual e lógico, realiza a conversão entre modelos, e também gera o modelo físico, composto por um *script* SQL (do inglês, Structured Query Language), responsável por criar as tabelas no SGBD (Sistema de Gerenciamento de Banco de Dados).

2.9.5. PostgreSQL

Se tratando de banco de dados, o PostgreSQL é um SGBDOR (Sistema Gerenciador de Banco de Dados Objeto-Relacional) que oferece diversos recursos como a integridade de transação, gatilhos e funções. Mesmo que o PostgreSQL tenha algumas funções relacionadas a orientação a objetos, sua principal característica de destaque são os conceitos de banco de dados relacionais. Atualmente é considerado ser o mais avançado banco de dados disponível para uso gratuitamente e de código aberto. Assim como maior parte dos bancos de dados relacionais, a linguagem que o PostgreSQL utiliza é o SQL, uma linguagem de consulta estruturada (BONFIOLI, 2006, p. 20 e 21).

2.10. Programação

O desenvolvimento de *software* é feito por meio da programação, um processo de definição e escrita de instruções, projetadas através de algoritmos, ou seja, passos previamente definidos que precisam ser seguidos para a realização de uma determinada tarefa. Contudo, os computadores não são capazes de entender a linguagem humana, portanto foram criadas diversas linguagens específicas para os computadores, de forma que possam compreender e então executar o que é solicitado (CARVALHO e LORENA, 2017, p. 197 e 198). Para

programar um *software* pode-se utilizar uma ferramenta de edição de texto simples, porém existem ferramentas específicas para essa finalidade.

2.10.1. Editores de Código e Ambientes de Desenvolvimento Integrado (IDEs)

Escritas de código e modificações são constantes durante a programação, para que esse processo seja mais eficiente foram criados os editores de código. Existem diversos editores de código disponíveis para uso, cada um tem uma implementação específica e os recursos oferecidos também depende da linguagem de programação alvo. Os editores de código contam com recursos bastante úteis, como por exemplo: facilita a análise do código, sugere código para auto completar nomes de variáveis, funções e classes, fornece soluções para refatoração, entre outros recursos (NIEHUES, 2013, p. 11). Alguns editores de código são: Notepad++, VS Code e Brackets.

Contudo, os editores de código nem sempre contam com todos os recursos necessários para determinados projetos, principalmente os mais grandes e complexos, sendo assim, surgiu o que é conhecido como ambiente de desenvolvimento integrado (IDE), uma ferramenta de interface gráfica que além possibilitar a edição de código, integra funções como depuração, interpretação, compilação e permite a realização de testes automatizados. É comum que as IDEs sejam projetadas para linguagens específicas, porém atualmente a maioria suporta múltiplas linguagens, com os propósitos mais diversos, até mesmo são multiplataforma (PIZETTA, 2014, p. 43 e 44). O Android Studio, Visual Studio e Arduino IDE são exemplos de IDEs.

2.10.2. Controle de Versão

De acordo com João Jr. *et al.* (2019, p. 1) o sistema de controle de versão transformou o processo de desenvolvimento de *software*, os programadores podem salvar versões do trabalho de maneira organizada e ainda conseguem colaborar entre si no mesmo projeto. Uma dessas ferramentas é o Git, o código-fonte atualizado do projeto é salvo em um histórico de versões, a partir disso pode-se rastrear e voltar em etapas que já foram antes desenvolvidas. Contudo, para tornar o repositório do projeto Git disponível para compartilhamento e colaboração entre os integrantes do projeto e a comunidade, é necessário hospedar o repositório em uma plataforma para essa finalidade.

Oliveira e Sirqueira (2021, p. 4) afirma que o GitHub é uma plataforma de hospedagem de repositórios pioneira neste quesito, basta o usuário se registrar na plataforma e já é livre para colaborar tanto em projetos privados quanto nos de código aberto. Seus repositórios apresentam um histórico de versões e também mostra quem está contribuindo, possibilitando que os desenvolvedores modifiquem códigos-fonte de outros desenvolvedores, adicionem outras funcionalidades e corrijam *bugs*, porém não se limita a apenas isso. É interessante mencionar que o WordPress e o *kernel* do GNU/Linux são exemplos de projetos que estão hospedados na plataforma GitHub.

2.10.3. Desenvolvimento do *firmware* com C/C++

De acordo com Pinotti (2021, p. 30 e 31) a linguagem C surgiu em 1970 aproximadamente, à medida que os microcomputadores foram se tornando mais populares, novos recursos foram adicionados. A linguagem possibilita trabalhar tanto em baixo como em alto nível, possibilitando manipular *bits*, ter acesso direto a componentes do *hardware* e até mesmo acessar valores da memória a partir de seus endereços. Porém, devido os *softwares* terem se tornado maiores e mais complexos conforme o tempo foi passando, a programação com a linguagem C acabou ficando mais difícil, principalmente pelo tamanho do código e dificuldade de manutenção posteriormente.

Visando vencer as limitações da linguagem C, em 1979 foi criada a linguagem C++, já sua padronização ocorreu em 1998, sendo assim, diversas revisões foram feitas, adicionou-se o paradigma de orientação a objetos e outros recursos como encapsulamento, polimorfismo e herança (PINOTTI, 2021, p. 31). A linguagem é capaz de acessar recursos do *hardware* diretamente por também ser de baixo nível, é gratuita e tem um padrão bem difundido com diversas bibliotecas disponíveis para uso. Contudo, a linguagem de programação C/C++ se mostrou mais propícia para o desenvolvimento do *firmware*, pois até mesmo o ESP32 e o Arduino são desenvolvidos com essa linguagem de maneira nativa.

2.10.4. Desenvolvimento do *front-end* com Dart e Flutter

O Dart é uma linguagem que se caracteriza por ser fortemente tipada, bastante flexível e multiparadigma, um conceito que permite a realização de uma mesma tarefa de maneiras diferentes. Como objetivo inicial, a Google criou o Dart em 2011 para substituir a linguagem

de programação JavaScript nos navegadores *web*, porém a linguagem tem se destacado desde o surgimento do Flutter. Um *framework* criado em Dart para desenvolvimento de aplicativos multiplataforma, um mesmo código pode ser compilado para executar nativamente em sistemas operacionais Android e IOS por exemplo (FRANKLIN e SAMUEL FILHO, 2020, p. 3 e 4). Por essas características o Flutter tem se mostrado uma ótima opção para desenvolver o aplicativo.

2.10.5. Desenvolvimento do *back-end* com C# e .NET

A linguagem de programação C# tem um alto nível de abstração, esse é um dos fatores que chama a atenção das empresas e desenvolvedores, devido ser uma característica que contribui para um desenvolvimento mais ágil. Criada utilizando o que há de melhor nas linguagens C++ e Java, a linguagem C# permite a implementação de múltiplas interfaces, porém a herança só pode ser realizada a partir de uma única classe. É orientada a objetos assim como o Java, porém, diferente do C++, não é possível declarar variáveis e funções de escopo global, ou seja, as declarações só podem ocorrer dentro de classes (AFONSO, STANGHERLINI e MAILLARD, 2008, p. 1 e 2).

É importante destacar que o C# é uma linguagem utilizada na plataforma .NET da Microsoft, porém a plataforma também utiliza outras linguagens, como o Visual Basic e F#. A plataforma .NET, permite desenvolver APIs REST e aplicações para diversas plataformas: IoT, aplicativos móveis e programas para *desktop* (NIQUINI, 2021, p. 36). De acordo com Figueiredo (2020, p. 27), a plataforma subdivide-se em .NET *Framework* (funciona apenas em plataformas que tenham suporte) e .NET *Core* (é de código aberto e funciona em múltiplas plataformas). Contudo, o motivo da escolha dessas tecnologias é pelo fato de juntas disporem de alta performance, flexibilidade e segurança.

3. METODOLOGIA

3.1. Metodologia de Pesquisa

Visando atingir os objetivos propostos, se faz necessário compreender os métodos a serem aplicados neste estudo, uma estratégia que provê de maneira organizada a direção e foco a ser seguido, na busca de encontrar a solução para os problemas apontados através do objeto de estudo. Os conceitos de uma metodologia são fatores determinantes no desenvolvimento da pesquisa, sendo assim, as perspectivas do pesquisador devem seguir seus fundamentos. O estudo de caso é uma metodologia difundida no meio científico, pode ser aplicado em diferentes áreas do conhecimento, porém para utilizar essa abordagem é importante conhecer bem suas definições (CASTRO FILHO, FREIRE e MAIA, 2021, p. 3 e 4).

Proetti (2017, p. 3) descreve sobre os tipos de investigação que a pesquisa deve abordar, a quantitativa se caracteriza por ser mais objetiva, fornece mais precisão por trabalhar com dados que podem ser medidos, já a realizada por meio da investigação qualitativa, o objetivo é compreender, analisar e descrever os eventos da pesquisa, porém ambas podem ser mescladas dependendo do objeto de estudo. É importante enfatizar que devido o estudo de caso ter características de estudos mais profundos, utiliza a abordagem qualitativa, compondo também fenômenos e circunstâncias de interesse do pesquisador e que chamem sua atenção.

Conforme afirmado por Castro Filho *et al.* (2021, p. 8), o estudo de caso tem três perspectivas distintas, segue as características de cada no quadro abaixo:

Quadro 1 - Diferentes perspectivas de estudo de caso

Perspectiva	Característica
Descritiva	Descrição minuciosa e completa do fenômeno.
Exploratória	Elaboração de conhecimentos ou levantamento de informação acerca do fenômeno ainda pouco conhecido.
Explanatória (Explicativa)	Explicação de causas ou efeitos relacionais ao fenômeno.

Fonte: Castro Filho *et al.* (2021, p. 8).

Com base no que foi dito neste trabalho, será realizado uma investigação qualitativa, por meio do estudo de caso, utilizando uma perspectiva descritiva. Dessa forma, serão descritos os processos para solucionar os problemas levantados, a partir de informações detalhadas, para que o leitor compreenda todo o processo para solucionar o problema.

3.2. Metodologia de Desenvolvimento de Projeto

Conforme descrito no item 2.8 do capítulo anterior, foi utilizado a metodologia ágil *Scrum* para o desenvolvimento do projeto. Após a conclusão do projeto, chegou-se num total de 15 *sprints*. O quadro apresentado abaixo, mostra a descrição sobre o que foi desenvolvido em cada *sprint*, e também o seu período:

Quadro 2 - Detalhes das *sprints* do projeto

<i>Sprint</i>	Descrição	Período
<i>Sprint 01</i>	Desenvolver documentos, diagramas e interfaces do aplicativo.	01/04/2022 à 30/04/2022
<i>Sprint 02</i>	Desenvolver o <i>hardware</i> .	01/07/2022 à 31/07/2022
<i>Sprint 03</i>	Iniciar o desenvolvimento do <i>firmware</i> com a funcionalidade de leitura do sensor, para obter estado do portão.	01/08/2022 à 31/08/2022
<i>Sprint 04</i>	Desenvolver no <i>firmware</i> a funcionalidade para acionar o portão eletrônico.	01/09/2022 à 30/09/2022
<i>Sprint 05</i>	Desenvolver no <i>firmware</i> a funcionalidade para fazer o reset do <i>hardware</i> .	01/10/2022 à 31/10/2022
<i>Sprint 06</i>	Desenvolver no <i>firmware</i> a funcionalidade para gerenciar os LEDs indicadores.	01/11/2022 à 30/11/2022
<i>Sprint 07</i>	Configurar o <i>Broker</i> MQTT e desenvolver no <i>firmware</i> a funcionalidade de comunicação com o mesmo.	01/01/2023 à 31/01/2023
<i>Sprint 08</i>	Desenvolver o core do servidor, já com autenticação JWT e conexão com o SGBD.	01/02/2023 à 28/02/2023
<i>Sprint 09</i>	Configurar serviço de e-mail no servidor e desenvolver as funcionalidades de todas as rotas de acesso.	01/03/2023 à 31/03/2023
<i>Sprint 10</i>	Desenvolver no <i>firmware</i> funcionalidade de comunicação por bluetooth, comunicação com o servidor pela internet e sincronização das operações com o servidor.	01/04/2023 à 30/04/2023
<i>Sprint 11</i>	Iniciar o desenvolvimento do <i>core</i> do aplicativo.	01/05/2023 à 31/05/2023
<i>Sprint 12</i>	Desenvolver as interfaces do aplicativo.	01/06/2023 à 30/06/2023
<i>Sprint 13</i>	Desenvolver os controladores do aplicativo.	01/07/2023 à 31/07/2023
<i>Sprint 14</i>	Desenvolver no aplicativo a funcionalidade de persistência de dados e comunicação com o <i>Broker</i> MQTT.	01/08/2023 à 31/08/2023

<i>Sprint 15</i>	Desenvolver no aplicativo a funcionalidade para comunicação por bluetooth e comunicação com o servidor na internet.	01/09/2023 à 30/09/2023
------------------	---	-------------------------

Fonte: Próprio autor.

Contudo, dentro do item 1.8.2 do Projeto de *Software* (Apêndice A) pode ser visto o quadro de *Product Backlog*, onde é apresentado todos os itens com sua estimativa de esforço e nível de prioridade. No interior do mesmo item, pode ser encontrado os quadros de cada uma das *sprints*, com os itens da *sprint* definidos, identificando em cada item, o responsável e previsão do período de duração.

4. RESULTADOS E DISCUSSÕES

4.1. Resultados da Arquitetura e Componentes do Projeto

4.1.1. Arquitetura de Comunicação

Foi utilizado a arquitetura cliente-servidor, para comunicação entre todos os componentes da aplicação. O aplicativo funciona como cliente. A camada *server-side*, é formada pelo Servidor da Aplicação e o *Broker* MQTT, e funciona como servidor. O *hardware* da aplicação, funciona tanto em modo cliente quanto em modo servidor, ou seja, no caso de comunicações via internet com a camada *server-side*, funciona como cliente, já em comunicações bluetooth diretamente com o aplicativo funciona como servidor.

4.1.2. Arquitetura da Aplicação

A arquitetura da aplicação é formada por diversos componentes, alguns deles se conversam diretamente de maneira física, outros por conexão sem fio. A comunicação entre o *hardware* da aplicação e o portão eletrônico ocorre com a utilização de fios, sensores e atuadores. Já a comunicação do aplicativo com o *hardware* é sem fio, e pode ocorrer de duas maneiras, por bluetooth (Veja a Figura 7), ou pela internet que será o modo padrão (Veja a Figura 8).

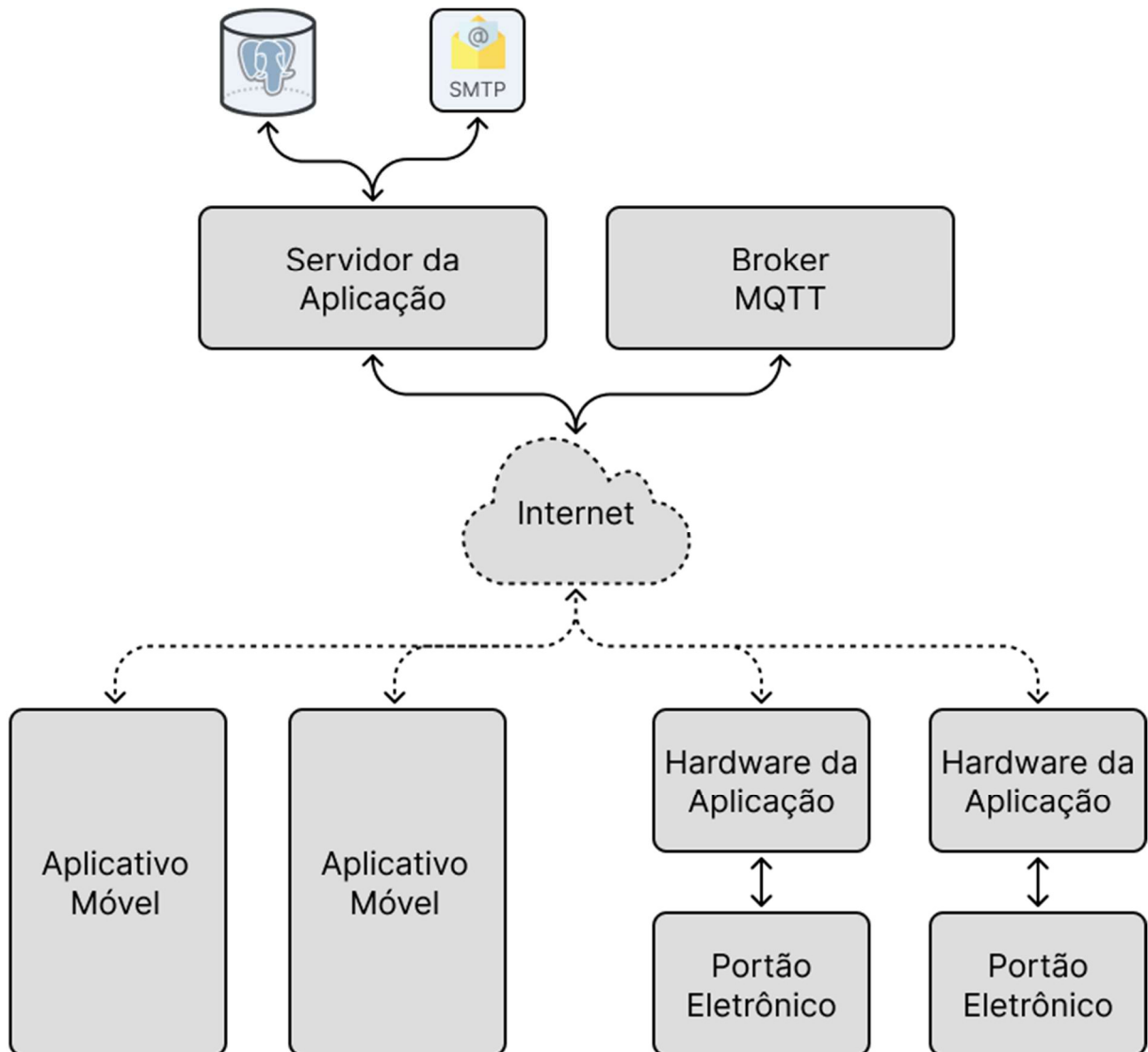
Figura 7 - Imagem da Arquitetura de Comunicação via Bluetooth



Fonte: Próprio autor.

A conexão via bluetooth, é utilizada nos momentos em que o portão é configurado pela primeira vez, ou quando não é possível estabelecer conexão com a internet. Porém, mesmo que por bluetooth, as operações realizadas são sincronizadas com o servidor posteriormente quando a internet é restabelecida. Quando a conexão for por meio da internet, o aplicativo envia os comandos de acionamento para a camada *server-side*, e esta é responsável por entregar os comandos para execução no portão eletrônico.

Figura 8 - Arquitetura Completa da Aplicação



Fonte: Próprio autor.

Depois que o portão é acionado, o *hardware* se encarrega de registrar no servidor a operação realizada. Uma operação é composta por: tipo (Abertura, Fechamento, Parada e Acionamento), portão acionado, usuário responsável e data e hora do acionamento. O histórico

das operações realizadas, pode ser visualizado posteriormente pelo aplicativo, porém neste processo, o aplicativo busca os dados diretamente do servidor.

4.1.2.1. Aplicativo Móvel

Caracterizado por ser o *front-end* da aplicação, seu intuito é dispor de interfaces e funcionalidades para que os usuários possam compreender facilmente, e utilizar o produto de maneira mais intuitiva. Deve ser instalado nos *smartphones* dos usuários, dessa forma poderão realizar configurações no portão, obter informações de funcionamento e principalmente realizar as operações de abertura e fechamento do portão.

4.1.2.2. Servidor da Aplicação

É basicamente uma API, que fica hospedada na internet, baseia-se no protocolo HTTP e utiliza o formato JSON para trocar dados. Permite aos usuários autenticarem-se de forma segura, um passo crucial para garantir a segurança no uso da aplicação. Os usuários podem fazer *login* na aplicação por meio dessa API, usando credenciais de *login*, como nome de usuário e senha. Além disso, oferece funcionalidades para registrar as operações de acionamento do portão.

Quando o portão é acionado, a API é capaz de receber requisições POST para registrar essas operações no banco de dados. Isso permite que os usuários mantenham um registro das ações realizadas em seus portões, o que pode ser útil para fins de segurança ou monitoramento. Também é possível registrar, obter, alterar e remover informações sobre os usuários, portões e localidades cadastradas. Os usuários podem acessar esses dados pelo aplicativo, sendo que este realiza as requisições, o que facilita o acesso e gerenciamento das informações importantes relacionadas aos dispositivos e usuários registrados na aplicação.

Outra funcionalidade interessante, é o envio de e-mail, realizado sempre que um novo usuário é convidado para usar a aplicação. Isso simplifica o processo de integração de novos membros, pois no e-mail consta as informações necessárias, para acessar o aplicativo, e então controlar o portão eletrônico. Todos os dados relacionados a usuários, portões e operações do portão são armazenados com segurança em um banco de dados, garantindo a integridade e a disponibilidade das informações.

4.1.2.3. *Broker* MQTT

O *Broker* MQTT também fica hospedado na internet, é um intermediário que permite a comunicação entre o aplicativo de *smartphone* e o *hardware* ligado ao portão eletrônico. O *Broker* MQTT recebe as mensagens publicadas pelos dispositivos, filtra as mensagens por tópico e distribui as mensagens para os dispositivos que se inscreveram no tópico correspondente. O *Broker* também oferece recursos como segurança, qualidade de serviço e retenção de mensagens.

Por exemplo, se um aplicativo publica uma mensagem com o tópico “portao1/abrir” e o conteúdo “true”, o *Broker* MQTT irá enviar essa mensagem para o portão eletrônico que está inscrito no tópico “portao1/abrir”. Assim, o portão eletrônico recebe o comando para abrir. Isso facilita a implementação do modelo de comunicação *publish-subscribe*, que é o mais adequado para as aplicações de Internet das Coisas, um conceito base para o desenvolvimento deste projeto.

4.1.2.4. Portão Eletrônico

Traz mais praticidade no dia a dia das pessoas, pois possibilita a abertura do portão em curtas distâncias sem necessidade de deslocamento do usuário até o mesmo, e descarta a necessidade de uso de força física por parte do utilizador. É composto principalmente por um motor elétrico que, ao ser alimentado por energia elétrica gira seu eixo, o eixo movimenta engrenagens que, mecanicamente é capaz de abrir ou fechar facilmente o portão. O componente responsável por gerenciar o funcionamento do motor elétrico, é uma placa de circuito eletrônica, denominada central eletrônica.

A central recebe os comandos do controle remoto por radiofrequência, realiza o processamento, faz a leitura dos sensores, e através de atuadores envia eletricidade para que o motor gire, assim o portão pode ser aberto ou fechado. A central tem conectores, assim é possível adicionar outros componentes externos, para configuração ou inclusão de novos recursos. Com base nisso, nos conectores de botoeira da central, foi interligado os fios para realizar o acionamento do portão eletrônico.

4.1.2.5. *Hardware* da aplicação

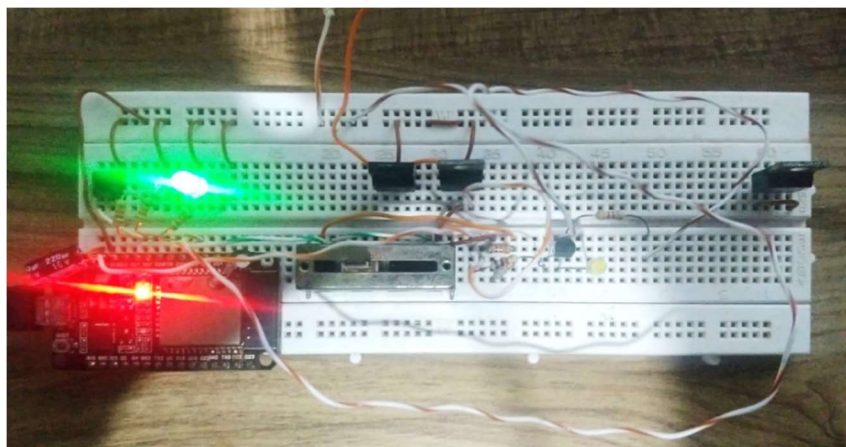
É a parte física da aplicação, consiste em uma placa eletrônica cheia de componentes, que são responsáveis por controlar o funcionamento do portão eletrônico pelo celular. A placa eletrônica tem um microcontrolador, que é um circuito integrado que contém processador, memória e interface de entrada e saída. O microcontrolador é o cérebro da placa, que executa o programa que define a lógica e o comportamento da aplicação. A placa eletrônica também tem um sensor magnético e alguns fios ligados a ela.

O sensor magnético é um componente eletrônico que detecta a presença ou a ausência de um campo magnético, e é usado para saber se o portão está aberto, fechado ou em movimento. Os fios são usados para conectar a placa à botoeira da central do portão eletrônico e à fonte de alimentação. A conexão wi-fi permite que a placa se comunique com o servidor e com o aplicativo de *smartphone* por meio do *Broker* MQTT. A conexão Bluetooth permite que a placa se comunique direto com o *smartphone* do usuário.

4.2. Resultados do Desenvolvimento do *Hardware*

De início, foi montado um protótipo para realizar alguns testes, na tentativa de obter o melhor arranjo de componentes, e identificar os valores dos componentes eletrônicos que seriam mais adequados para cada funcionalidade do *hardware*. Contudo, este ainda não é o protótipo definitivo (Veja a Figura 9).

Figura 9 - Protótipo do *hardware* em fase de testes



Fonte: Próprio autor.

O principal componente é o microcontrolador ESP32, nele que o *firmware* desenvolvido é gravado. Para a alimentação elétrica, foi adicionado no circuito um regulador de tensão (U2), responsável por regular a tensão recebida da central do portão eletrônico ou de alguma outra fonte externa, para 5 volts, tensão utilizada pelo ESP32, sendo filtrada pelos capacitores (C3 e C4). Também foi incluso um botão (B1) para permitir o *reset* das configurações. Um sensor de efeito *hall* linear (U1), juntamente com o capacitor (C1), para fazer a leitura do estado atual do portão.

Para a apresentação de *status* de funcionamento do *hardware*, foi adicionado quatro LEDs indicadores (D1, D2, D3 e D4), junto com seus resistores de limitação de corrente (R3, R4, R5 e R6). Por fim, para acionamento da botoeira da central foi utilizado um transistor (Q1) ligado aos resistores (R1 e R2). Com base nisso, foi montado uma lista com os materiais e componentes necessários para a construção da placa eletrônica, a lista tornou mais ágil a seleção dos componentes para construir o *hardware*, veja a Tabela 2 a seguir:

Tabela 2 - Lista de materiais e componentes para construção do *hardware*

Quantidade	Item
1	Placa de fenolite ilhada 5x10 cm.
1	ESP32 DevKitC.
4	LED colorido (azul, amarelo, verde e vermelho).
4	Resistor 220 Ω .
1	Resistor 4,7 K Ω .
1	Resistor 330 Ω .
1	Capacitor 100 nF de cerâmica.
1	Capacitor 100 nF de poliéster.
1	Capacitor 1000 μ F eletrolítico.
1	Capacitor 2,2 μ F eletrolítico.
1	Transistor NPN de uso genérico.
1	Regulador de tensão LM7805.
1	Sensor magnético de efeito <i>hall</i> linear 49E.
1	Botão (<i>push-button</i>).
1	Fios de cores variadas.

Fonte: Próprio autor.

A montagem do *hardware* foi uma etapa crucial para concretização deste projeto. Cada passo foi executado, iniciando com a seleção dos materiais e componentes, conforme previamente detalhado na lista de materiais e componentes. A escolha criteriosa desses

elementos, alinhada com as necessidades e requisitos do projeto, estabeleceu a base para a eficácia e desempenho desejados. Durante a montagem, cada componente foi integrado de acordo com o arranjo previamente concebido, garantindo a funcionalidade esperada.

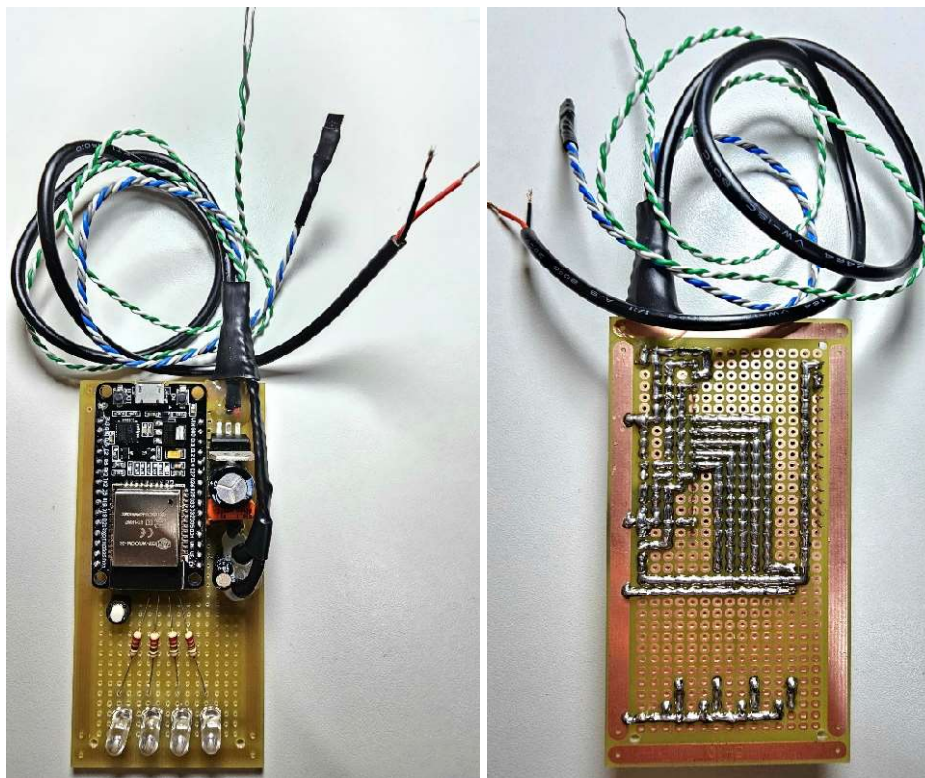
Figura 11 - Materiais e componentes selecionados para a construção do *hardware*



Fonte: Próprio autor.

A minuciosidade durante o processo de montagem não apenas assegurou a integridade do *hardware*, mas também estabeleceu as condições ideais para a eficiência e confiabilidade do sistema. Os componentes eletrônicos foram fixados na placa por meio de soldagem eletrônica, com o uso de um ferro de solda aquecido a uns 300 °C, e o fio de solda, composto basicamente por uma liga de estanho e chumbo. Cada passo da soldagem foi conduzido com atenção aos detalhes, garantindo a correta conexão física entre os componentes e a placa.

Primeiro os terminais dos componentes eletrônicos foram inseridos nos orifícios da placa, então foi encostado o ferro de solda e ao mesmo tempo o fio de solda, entre o componente e o cobre do orifício da placa, para que fossem unidos. Depois que todos os componentes ficarem fixos na placa, foi feito as trilhas, ou seja, a ligação entre os componentes, seguindo o projeto do modelo esquemático. Por último, foram soldados os fios e o sensor magnético, e a placa eletrônica ficou pronta (Ver a Figura 12).

Figura 12 - *Hardware* finalizado

Fonte: Próprio autor.

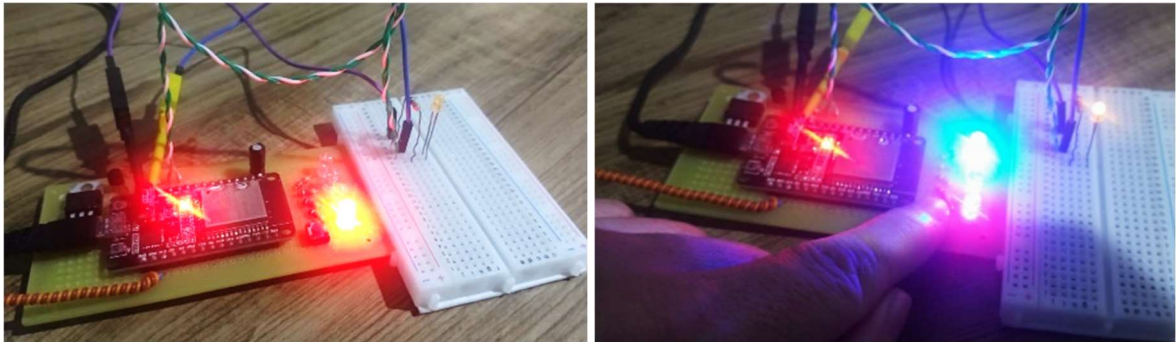
Com a conclusão deste processo, a placa eletrônica agora se apresenta de maneira sólida, capaz de suportar as demandas do projeto de maneira consistente e confiável. A eficiência, qualidade e precisão empregadas em cada etapa confirmam a viabilidade e sucesso deste projeto eletrônico. Contudo, ainda é preciso gravar no microcontrolador da placa o *firmware* desenvolvido, a inteligência por trás do funcionamento da placa eletrônica, para que possa desempenhar suas tarefas conforme o que foi previsto.

4.3. Resultados do Desenvolvimento do *Firmware*

O *firmware* foi desenvolvido utilizando o *framework* ESP-IDF, produzido pela própria fabricante do microcontrolador ESP32. Inicialmente, foi necessário configurar o ambiente de desenvolvimento, instalando o ESP-IDF e integrando com o VS Code. Com o ambiente pronto, iniciou-se a codificação em linguagem C/C++, aproveitando as bibliotecas e APIs fornecidas pelo ESP-IDF. Essa etapa foi crucial para configurar os pinos de entrada e saída do *chip* e integrar os componentes essenciais do sistema.

A lógica de controle foi implementada para interpretar as interações com o botão de pressionar, estabelecendo a funcionalidade de *reset* da placa. Porém, inicialmente, o botão foi testado com um método peculiar. Um LED foi conectado aos fios de acionamento do portão, isso permitiu verificar tanto o funcionamento do botão, quanto o envio do sinal de acionamento para a central do portão eletrônico (Ver a Figura 13).

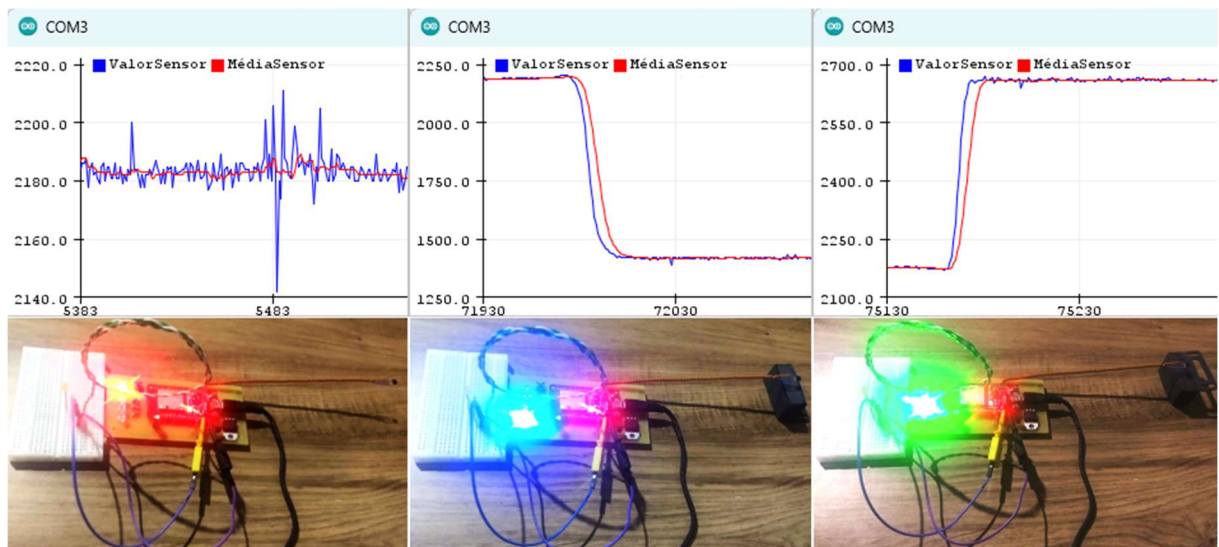
Figura 13 - Teste do botão *reset* e circuito de acionamento do portão



Fonte: Próprio autor.

Em seguida, foi desenvolvido o algoritmo para interpretar os dados do sensor magnético, permitindo a detecção de estado do portão. Foram criadas situações simuladas para representar as diversas posições do portão. Isso foi feito utilizando um ímã de fim de curso, para criar três cenários distintos: ímã distante do sensor, um dos polos do ímã próximo ao sensor e o outro polo do ímã próximo ao sensor (Observe a Figura 14).

Figura 14 - Teste do sensor para obter o estado do portão



Fonte: Próprio autor.

Conforme pode ser observado, para cada um dos três cenários de teste do sensor magnético, existe uma representação gráfica da leitura obtida deste mesmo sensor, e um LED de cor distinta, aceso na placa para sinalizar cada estado do portão. Os gráficos apresentam em sua legenda, duas cores diferentes de linha, a azul demonstra o valor obtido no momento da leitura do sensor, e a vermelha demonstra a média de uma amostragem de quantidade fixa, dos últimos valores obtidos do sensor.

Com isso, pode-se observar o quanto a leitura direta do sensor sem algum filtro é suscetível a ruídos, esses ruídos são muito comuns em dispositivos eletrônicos, porém geram funcionamentos inadequados no *firmware*. Portanto, com a utilização da média das leituras, este problema foi resolvido de maneira satisfatória. O microcontrolador foi preparado para obter do sensor, uma entrada que varia de 0 a 4096, esse é um valor padrão do ESP32 para leitura de tensões analógicas.

O primeiro cenário com o ímã ausente, representa o estado de um portão parcialmente aberto, o LED aceso é o vermelho, e o valor do sensor fica praticamente a metade de 4096, ou seja, um pouco mais ou um pouco menos de 2048. O segundo cenário, representa o estado de um portão aberto ou fechado, dependendo do que for definido, o LED aceso é o azul, e o valor do sensor cai ficando abaixo da metade. Já o terceiro cenário, representa o estado inverso do portão, o LED aceso é o verde, e o valor do sensor sobe ficando acima da metade.

Essas foram as funcionalidades fundamentais, para identificação do estado do portão, acionamento do portão, e *reset* do *firmware*. No entanto, para trocar informações com outros dispositivos, foi preciso desenvolver alguns módulos de comunicação, sendo eles: o Bluetooth, o Wi-Fi, o HTTP e o MQTT. Foi preciso também, desenvolver módulos com outras funcionalidades, para gerir a segurança, a sincronização de dados, o armazenamento de dados de maneira persistente, entre outros.

Durante o processo, surgiu a necessidade de desenvolver uma ferramenta destinada a facilitar os testes. Isso se deu pela constante ativação do portão, indispensável para ajustes e refinamentos, até que a lógica do *firmware* alcançasse o comportamento desejado. A repetida operação em um portão físico real poderia acarretar riscos e acelerar seu desgaste. Assim, para contornar essa questão, concebeu-se uma maquete com um portão que replica fielmente o funcionamento de um portão eletrônico deslizante real (Veja a Figura 15).

Figura 15 - *Hardware* ligado na réplica do portão eletrônico



Fonte: Próprio autor.

A maquete desempenhou um papel crucial ao possibilitar a realização de testes de forma segura e precisa. Além disso, ao replicar o funcionamento de um portão eletrônico real, permitiu a simulação de diferentes cenários, para os refinamentos necessários na lógica do *firmware* e correção de bugs. Porém, embora tenha sido valiosa em etapas intermediárias, à medida que certas funcionalidades do *firmware* eram concluídas, a validação final ocorria em um portão eletrônico real.

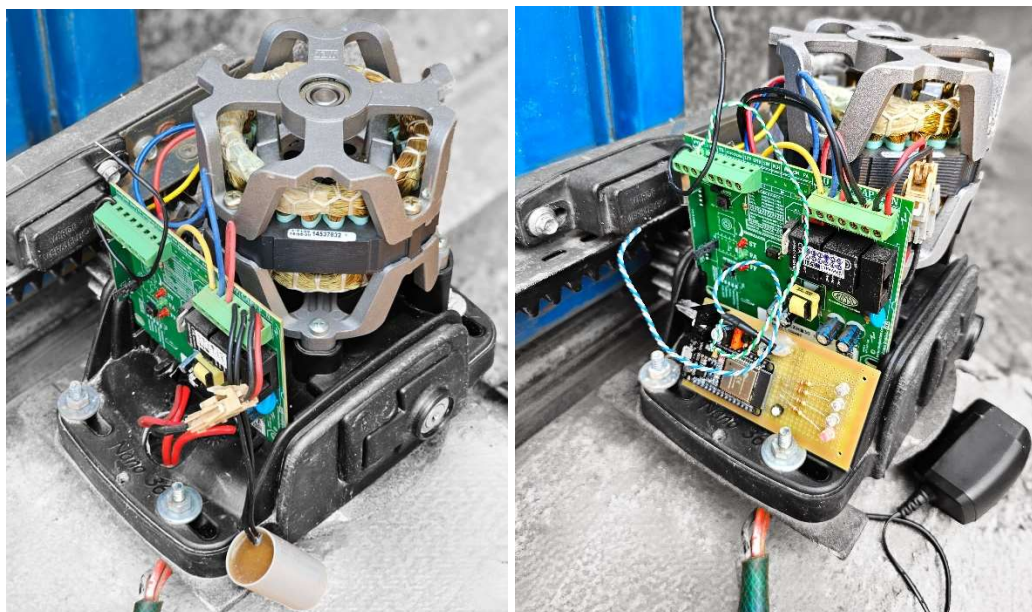
Depois de concluído o desenvolvimento do *firmware*, chegou o momento de gravar a versão definitiva no microcontrolador do *hardware*. Esse procedimento, marcou o término de uma fase intensiva de trabalho, cuidado para assegurar a estabilidade e o correto funcionamento do sistema. Com a versão finalizada e testada, o processo de gravação representou um ponto de transição para a implementação prática do *firmware*, incorporando as inovações e ajustes realizados ao longo do desenvolvimento.

4.4. Ligação do *Hardware* no Portão Eletrônico

Após a conclusão do desenvolvimento do *hardware* e do *firmware*, foi instalado o *hardware*, no portão eletrônico. É possível observar na Figura 16 uma clara distinção entre o antes e o depois no sistema do portão eletrônico. No lado esquerdo, está o motor do portão, retratado em sua configuração padrão, sem qualquer adição, conectado por meio de fios à sua

placa de controle, conhecida como central eletrônica. Já no lado direito da figura, destaca-se o resultado do trabalho realizado, com o *hardware* desenvolvido neste projeto integrado à central eletrônica do portão.

Figura 16 - Instalação do *hardware* no portão eletrônico



Fonte: Próprio autor.

O sensor magnético foi colocado junto com o sensor magnético do próprio portão eletrônico, dessa maneira, o *hardware* desenvolvido e a central eletrônica conseguem compartilhar do mesmo *status* operacional do portão. Os fios de acionamento da botoeira da central, foram dispostos no borne da central eletrônica, interligando o terminal negativo e o terminal de acionamento de ambas as placas. Normalmente, a central eletrônica também já disponibiliza em seus bornes uma tensão de alimentação, que permite a ligação de componentes externos, porém não são todas que conseguem fornecer corrente suficiente para estes novos componentes, principalmente no caso de demandarem mais energia do que é fornecido.

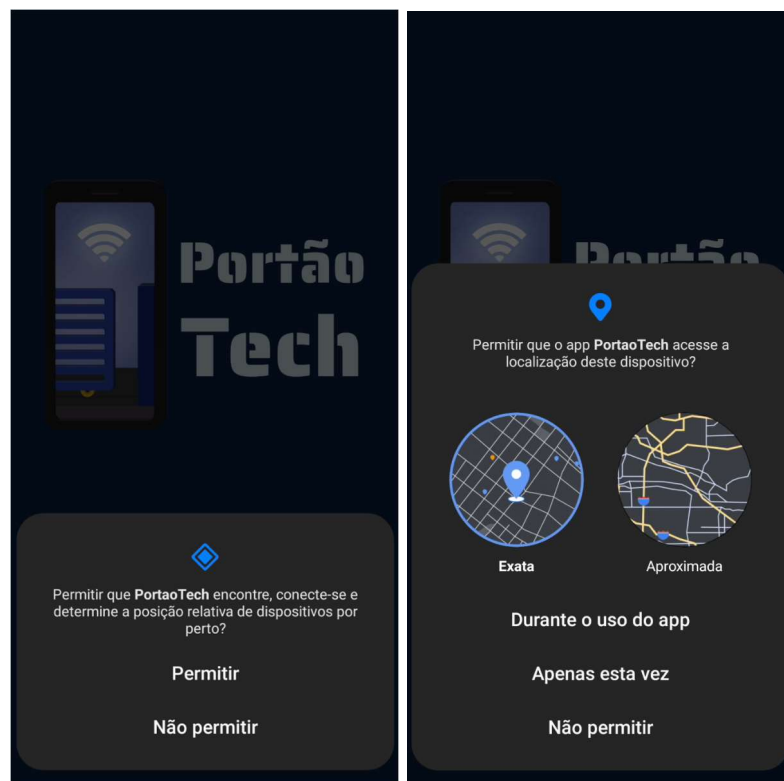
No contexto do portão eletrônico utilizado neste projeto, identificou-se que sua central eletrônica poderia não entregar a corrente necessária para operar o *hardware* desenvolvido. Considerando estes aspectos, optou-se por estabelecer a conexão de alimentação por meio de uma fonte externa, essa adaptação foi importante para garantir o funcionamento adequado e segurança do sistema. Em vista do exposto, foi criada uma sinergia entre o novo *hardware* e a central eletrônica já existente, de modo a agregar as novas funcionalidades idealizadas nesta iniciativa.

4.5. Resultados do Desenvolvimento do *Software*

4.5.1. Primeiro acesso ao aplicativo

Ao abrir o aplicativo pela primeira vez, é solicitado algumas permissões que são essenciais para acesso ao bluetooth e a localização do dispositivo. Essas permissões são necessárias, para que o aplicativo funcione com todas as suas funcionalidades.

Figura 17 - Solicitação de permissões de acesso a recursos do sistema



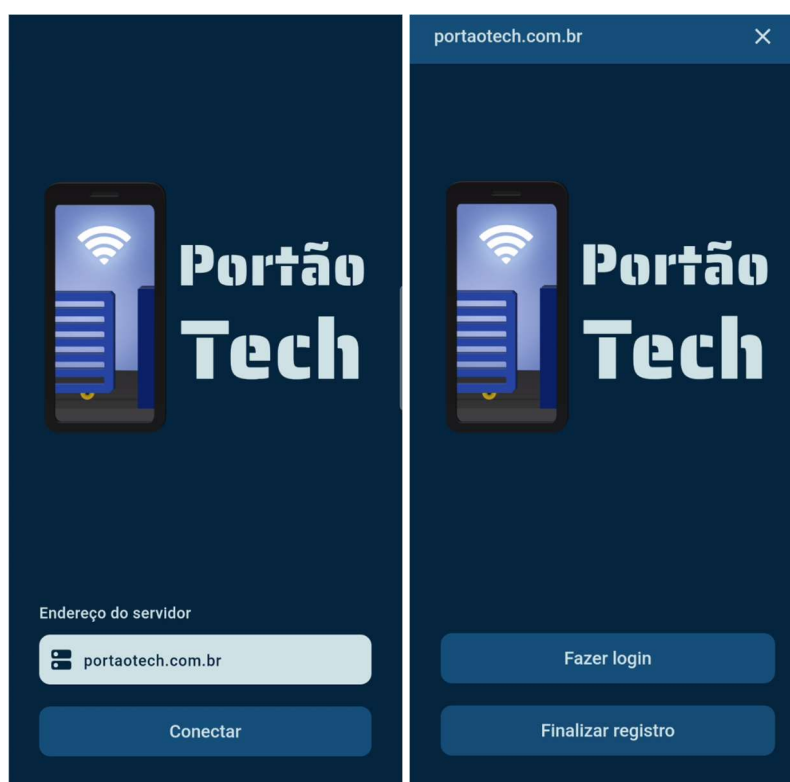
Fonte: Próprio autor.

É importante ressaltar que, o bluetooth é necessário para que o usuário possa configurar um portão pela primeira vez. Além disso, também é usado quando não houver conexão de internet, seja no *smartphone* do usuário ou no portão, ou seja, quando a placa conectada ao portão eletrônico perde conexão com a internet, não poderá receber comando e nem se comunicar com o servidor. Depois dessa etapa, é apresentado a tela para que o usuário informe o endereço do servidor.

4.5.2. Conexão com o servidor

Para que o aplicativo funcione é necessário conectá-lo em uma máquina que hospede o servidor *back-end* do PortãoTech. No servidor que é realizado a autenticação dos usuários, e os dados são inseridos, buscados, alterados e deletados por meio de requisições HTTP. Por padrão já vem preenchido o endereço “portaotech.com.br”, porém nada impede que seja utilizado outro endereço.

Depois que o endereço do servidor é informado, o usuário deve tocar o botão “Conectar” para que o aplicativo se conecte com o servidor.



Fonte: Próprio autor.

Caso não seja a primeira vez que o aplicativo é aberto, no momento da abertura é identificado se o usuário já fez o *login* antes. Caso tenha uma sessão de *login* ativa, é apresentado a tela principal ao usuário. Caso não tenha uma sessão ativa, é verificado se o endereço do servidor já foi informado anteriormente.

Se existir um endereço de servidor informado, é apresentado ao usuário a tela com as opções “Fazer login” e “Finalizar registro”, assim o usuário pode fazer o *login* se já tiver cadastro, ou finalizar o registro se estiver sido convidado e está acessando pela primeira vez.

4.5.3. Finalização de registro

A finalização de registro é parte do procedimento de registro de um novo usuário. Para que um novo usuário possa controlar os portões, antes ele precisa ser convidado. O convite é recebido via e-mail, com instruções de acesso e um código de verificação. Como pode ser visto na imagem abaixo:

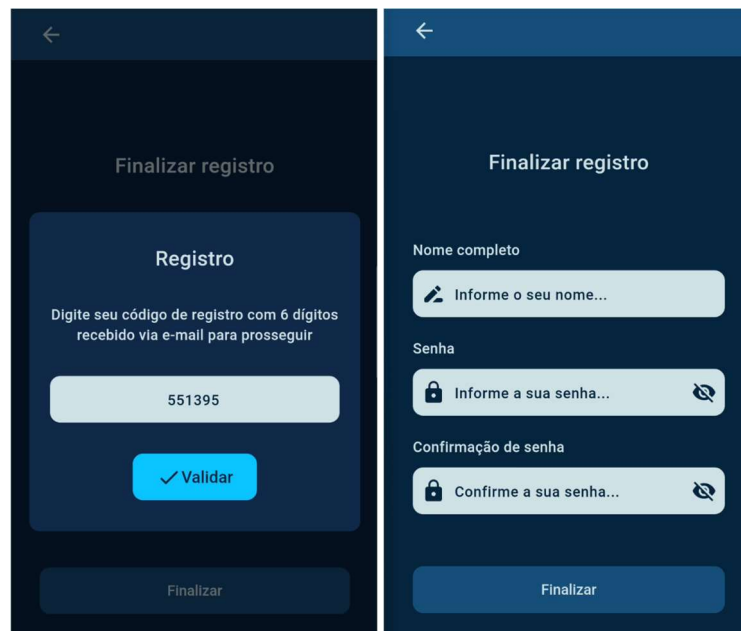
Figura 18 - Convite para finalização de registro recebido por e-mail



Fonte: Próprio autor.

Após aberto a tela de finalização de registro, é apresentado um modal onde é solicitado que seja informado o código de registro para prosseguir. O usuário precisa informar no campo o código de seis dígitos, recebido no convite via e-mail, depois deve tocar no botão validar.

Figura 19 - Telas de finalização de registro do usuário



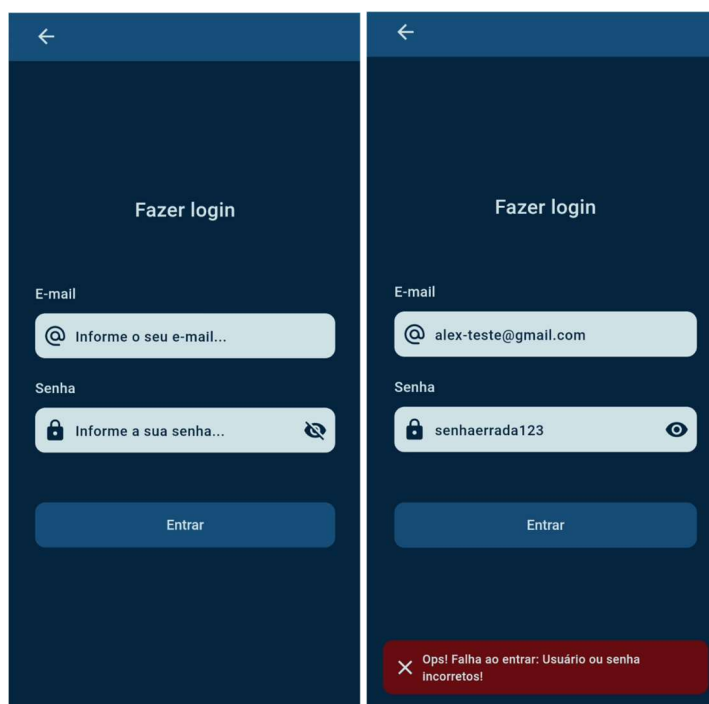
Fonte: Próprio autor.

Se for um código inválido, uma mensagem de erro é apresentada ao usuário. Caso seja um código válido, a janela *modal* é fechada, e o usuário já pode preencher suas informações. As informações solicitadas são “Nome completo”, “Senha” e “Confirmação de senha”. Após preencher as informações é preciso tocar no botão “Finalizar”, se tudo ocorrer bem, será apresentado a tela de *login*.

4.5.4. Realização do *login*

O acesso é realizado por meio da tela de *login*, nela o usuário precisa informar seu e-mail cadastrado e sua senha informada no momento da finalização do registro. Para facilitar o preenchimento da senha, existe um botão no lado direito do campo para exibir ou ocultar a senha digitada.

Figura 20 - *Login* do usuário



Fonte: Próprio autor.

Se ocorrer erro ao realizar o *login* por algum motivo, por exemplo: falta de conexão com a internet, servidor indisponível, usuário e/ou senha incorretos, entre outros, é apresentado uma mensagem de erro. Se o *login* for realizado com sucesso, o usuário será direcionado para a tela principal do aplicativo.

4.5.5. Tela principal

Como o próprio nome já diz, esta é a tela principal do aplicativo, se o usuário estiver feito seu *login* com sucesso, chegará diretamente nela. Quando esta tela é aberta, é verificado se já estava sendo controlado algum portão, da última vez que o aplicativo foi fechado. Se existir, o aplicativo já tenta se conectar automaticamente no mesmo portão. De início, é utilizado o mesmo tipo de conexão da última vez, seja pela internet ou bluetooth, porém se não for possível estabelecer a conexão, é utilizado o outro tipo de conexão como medida alternativa.

Se a conexão automática estiver sido estabelecida com sucesso, é mostrado ao usuário os botões de controle do portão, conforme a primeira tela da Figura 22. Agora, se ocorrer falha na conexão automática ou caso não exista um portão que estava sendo controlado anteriormente, é apresentado uma mensagem, e um botão logo abaixo chamado “Conectar”, como mostra a primeira tela da Figura 21. Sendo assim, basta o usuário tocar no botão, que aparecerá um diálogo *modal*, e será possível escolher o portão a ser controlado.

Contudo, se for preciso fazer alguma outra operação no aplicativo, basta abrir o menu principal, que fica acessível no canto superior esquerdo da tela, representado por um ícone, formado por três linhas horizontais.

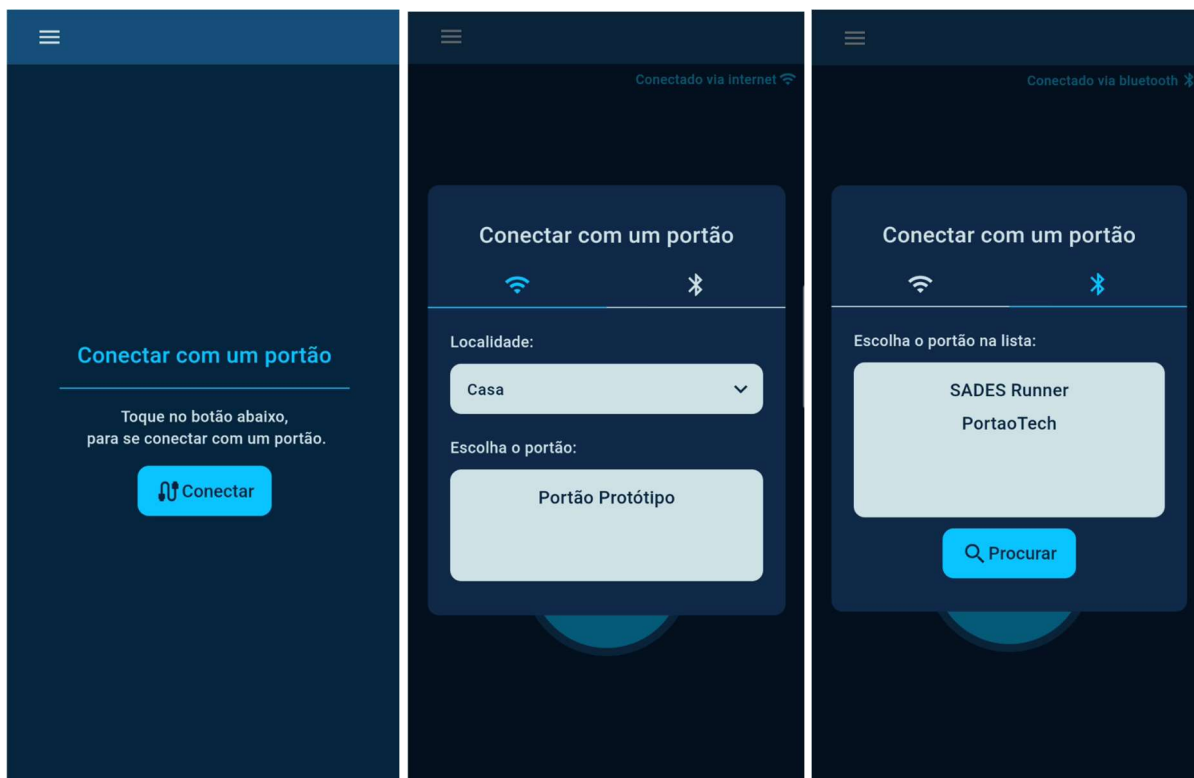
4.5.5.1. Escolha de um portão para se conectar e controlar

A escolha de um portão para controle, é realizada por meio de um diálogo *modal*, este *modal* é aberto e apresentado ao usuário por três caminhos. O primeiro, é por meio do botão “Conectar” que aparece na tela principal, quando nenhum portão está sendo controlado. O segundo caminho, é por meio de uma opção do menu principal. Já o terceiro, é tocando no nome e localidade do portão que está sendo controlado. Isso permite maior agilidade na escolha do portão, porque pode existir diversos portões cadastrados na aplicação.

O aplicativo pode se conectar com os portões através de dois tipos de conexão, a internet e o bluetooth. O usuário pode escolher qual o tipo de conexão deseja, para isso, é só tocar na aba com o ícone correspondente antes de escolher o portão. A conexão por meio da internet dever ser preferível, e por padrão é a primeira opção apresentada ao usuário. Já o bluetooth, dever ser utilizado como alternativa, nos casos que não houver conexão com a internet, seja no dispositivo do usuário ou no portão.

Na segunda e terceira tela da Figura 21, pode ser visualizado como se comporta a tela para cada tipo de conexão:

Figura 21 - Processo de conexão com um portão



Fonte: Próprio autor.

Para se conectar em um portão por internet, é necessário primeiro escolher a localidade que o portão pertence, depois disso, é apresentado uma lista com os portões que têm ligação com a localidade selecionada. Por fim, é só o usuário tocar no nome do portão desejado que está contido na lista, e então a conexão será estabelecida. Para se conectar por bluetooth, é preciso estar próximo do portão desejado, quando a tela é aberta já é apresentado os dispositivos pareados anteriormente.

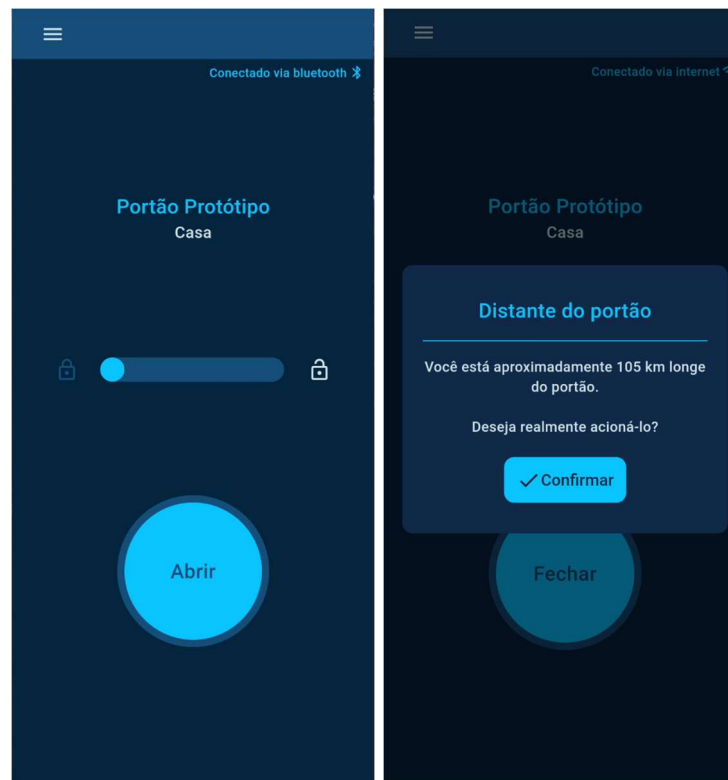
Se o portão já estiver sido pareado antes, já estará na lista, caso contrário é só tocar no botão “Procurar” que a descoberta de dispositivos será iniciada e o portão aparecerá na lista de dispositivos bluetooth. Encontrado o portão na lista, a próxima etapa é tocar no nome do portão, e então a conexão será estabelecida. Enquanto uma conexão é estabelecida é apresentado uma mensagem informativa e um progresso circular. Se ocorrer erro, uma mensagem é mostrada, caso contrário, o *modal* de conexão é fechado e os botões de controle do portão aparecerão.

4.5.5.2. Controlar um portão

A tela de controle dos portões eletrônicos é projetada para proporcionar aos usuários um meio intuitivo e eficaz de gerenciar o acesso aos seus portões. A tela é dividida em diferentes seções, cada uma com funcionalidades específicas. No topo da tela, é apresentado o tipo e *status* da conexão com o portão atual, sendo representado de maneira textual e por um ícone. O *status*, informa se a conexão está ativa ou se foi perdida. Isso fornece ao usuário informações cruciais sobre o estado da comunicação com o portão.

Logo abaixo do *status* de conexão, são exibidos o "Nome do Portão" e o "Local" ao qual o portão pertence, como "Casa" ou "Trabalho." Essas informações ajudam o usuário a identificar e selecionar o portão correto que deseja controlar.

Figura 22 - Controlar um portão



Fonte: Próprio autor.

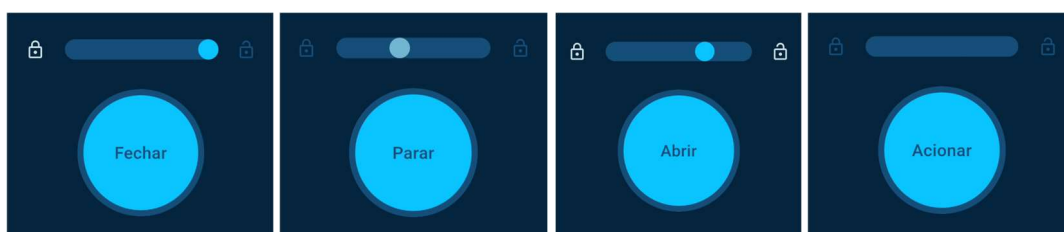
Depois que um portão é conectado, no primeiro acionamento é checado a distância entre o portão e usuário, por meio de coordenadas geográficas. Se a distância exceder 100 metros, é apresentado uma mensagem informativa, indicando a distância entre ambos, conforme mostra a segunda tela da Figura 22. Isso é por questão de segurança, para que o portão não seja acionado acidentalmente, como por exemplo: quando o celular é colocado no bolso desbloqueado, ou no

caso de o usuário se esquecer de escolher o portão correto na hora de acioná-lo. Para prosseguir com o acionamento do portão, o usuário precisa tocar no botão “Confirmar”.

Um dos botões de controle do portão, é o botão deslizante, nele os usuários podem definir onde desejam que o portão pare, permitindo um controle preciso sobre o movimento do portão. Para isso, o usuário só precisa segurar o botão do controle deslizante, arrastar onde deseja que o portão pare, e soltar. Quando o botão é solto, o portão inicia seu movimento e para exatamente onde a botão foi solto, proporcionando um controle proporcional com base em um percentual de abertura do portão.

Movendo o botão deslizante para a direita, o portão é aberto, para a esquerda, o portão é fechado, e em posições intermediárias, o portão para proporcionalmente. Nas extremidades do botão deslizante, existem botões adicionais com ícones de cadeado fechado e aberto. Ao tocar no ícone do cadeado fechado, o portão é fechado, e ao tocar no ícone do cadeado aberto, o portão é aberto.

Figura 23 - Funcionamento dos botões de controle do portão



Fonte: Próprio autor.

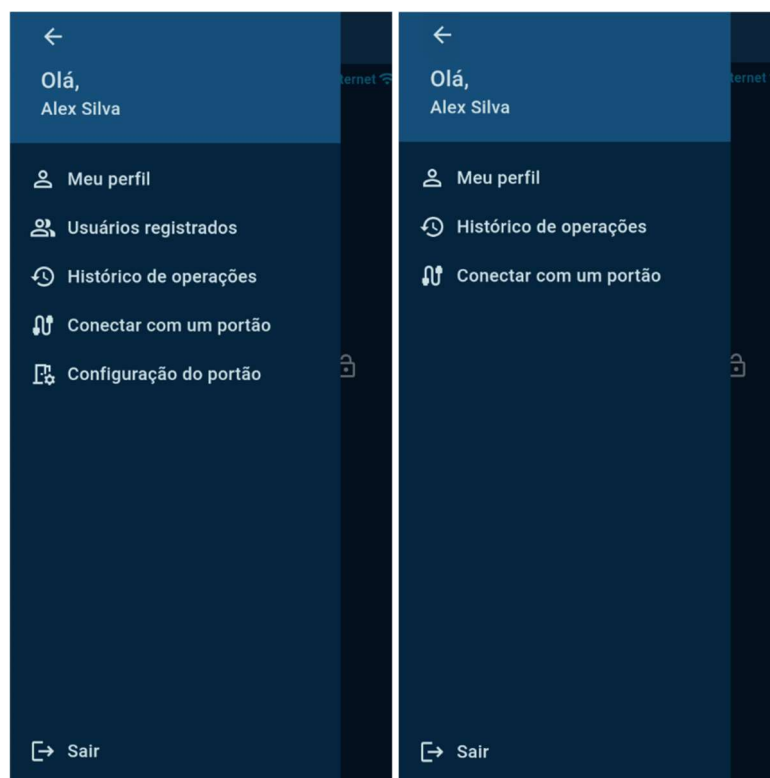
Abaixo do botão deslizante, encontra-se o botão principal, que é um botão redondo bem maior do que os outros. Ele permite ao usuário realizar ações imediatas no portão. No centro desse botão, há uma descrição clara da próxima ação que ocorrerá quando o botão for pressionado. As ações disponíveis incluem "Abrir", "Fechar", "Parar" e "Acionar". Esse botão principal fornece uma alternativa mais direta para controlar o portão, sem a necessidade de arrastar o botão deslizante.

No geral, essa tela de controle de portões proporciona uma interface amigável e eficiente para os usuários gerenciarem seus portões eletrônicos. Eles têm a capacidade de controlar a abertura, fechamento e parada do portão de maneira precisa, além de receber informações importantes sobre o estado da conexão e o portão que estão controlando.

4.5.5.3. Menu principal

O menu principal é acessado ao tocar no ícone das três linhas horizontais localizado na parte superior esquerda da tela principal do aplicativo. No topo do menu, é apresentado um botão "Voltar" que permite ao usuário fechar o menu e retornar à tela principal. Isso garante que a navegação seja intuitiva e que o usuário possa alternar entre o menu e a tela principal de forma conveniente. Abaixo do botão "Voltar", o nome do usuário logado é exibido, identificando qual conta está atualmente logada.

Figura 24 - Menu principal



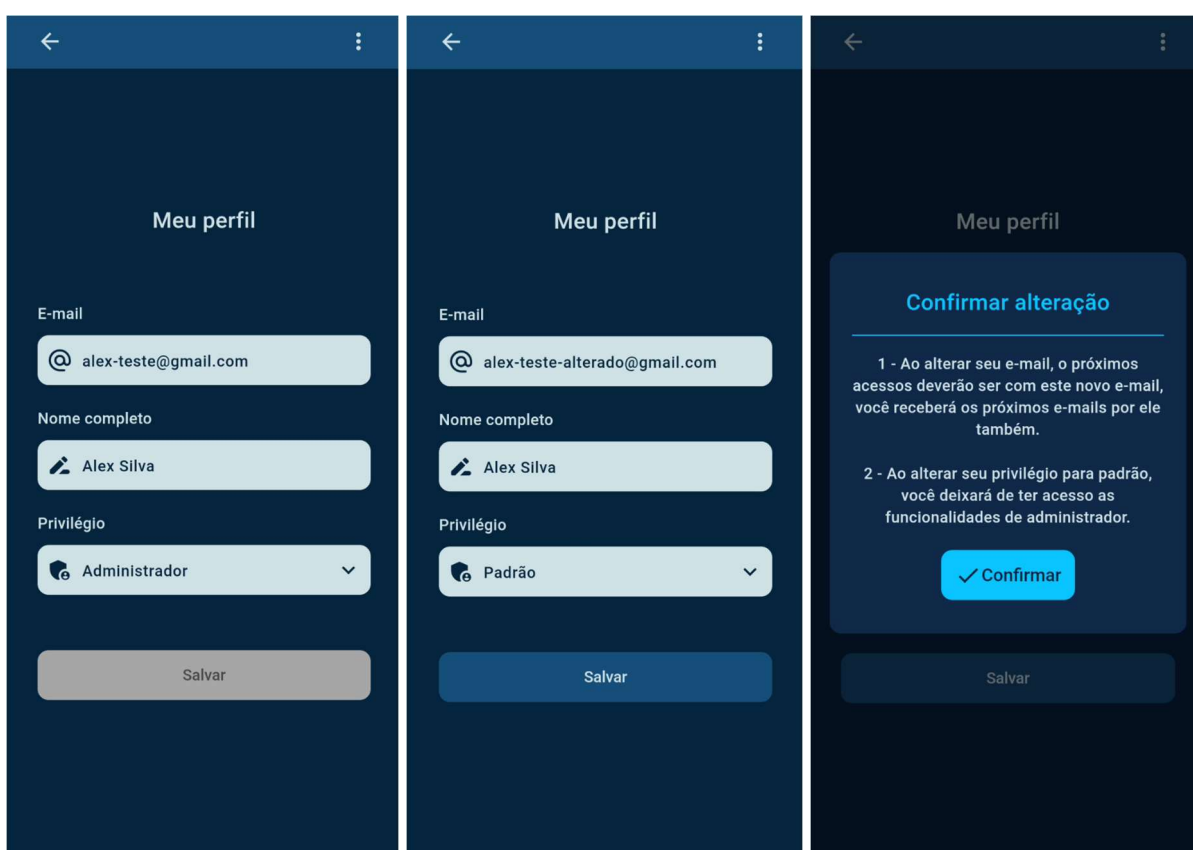
Fonte: Próprio autor.

A parte central do menu é dedicada à lista de funcionalidades. Para um usuário comum, denominado "Padrão", as funcionalidades apresentadas incluem "Meu perfil", "Histórico de operações" e "Conectar com um portão". Por outro lado, para os usuários com privilégios de administrador, além das funcionalidades disponíveis para o usuário "Padrão," eles têm acesso a funcionalidades específicas, como "Usuários registrados" e "Configuração do portão". No final do menu, um botão "Sair" permite que o usuário encerre sua sessão.

4.5.6. Perfil do usuário

A tela de perfil do usuário é uma das opções do menu principal do aplicativo. Nessa tela, o usuário pode visualizar e editar suas informações pessoais, como nome, e-mail e privilégio. Também pode excluir seu perfil, se assim preferir. Logo que a tela é aberta, o usuário pode ver os dados de seu perfil, e o botão “Salvar” fica desabilitado porque ainda não tem alteração nos campos.

Figura 25 - Perfil do usuário (Visualização, edição e confirmação de edição)



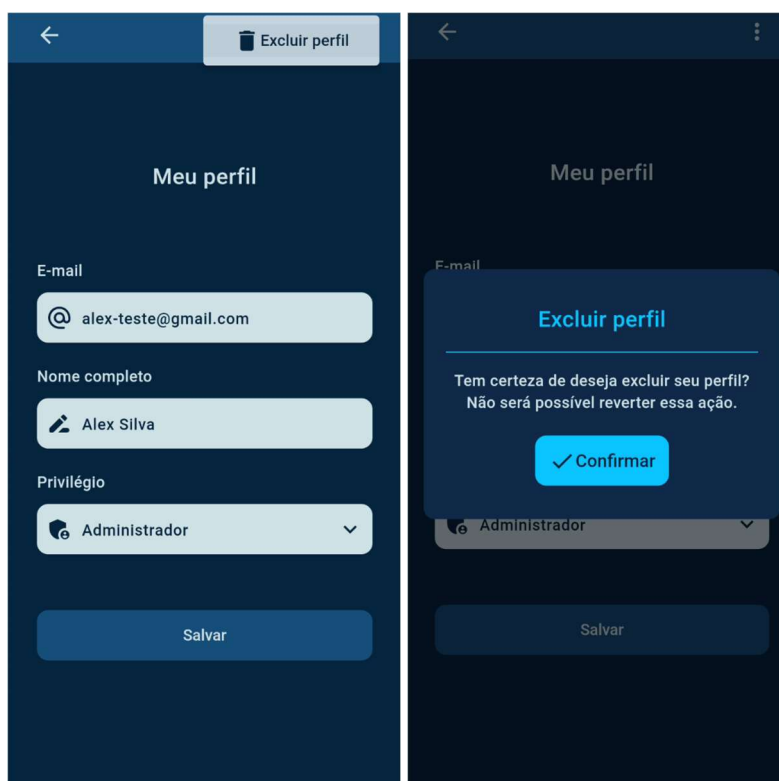
Fonte: Próprio autor.

Para editar qualquer um desses dados, basta tocar no campo desejado e digitar as novas informações. O primeiro campo da tela é o “E-mail”, este é o mesmo que o usuário usou para se registrar no aplicativo e fazer o login, mesmo assim, o e-mail pode ser alterado. O segundo campo é o “Nome completo”, serve para identificar o usuário na aplicação. Já o terceiro e último campo é o “Privilégio”, permite que o usuário altere seu privilégio, as opções são “Padrão” e “Administrador”.

Logo que o valor de algum campo é alterado, o botão “Salvar” é habilitado permitindo que o usuário salve as alterações realizadas em seu perfil. Contudo, no caso de alterações no e-mail e/ou privilégio, é apresentado um diálogo com mensagens importantes com orientações. Quando um usuário altera seu e-mail, os novos e-mails da aplicação serão enviados para o novo, os próximos logins também deixarão de funcionar com o e-mail antigo, portanto deve ser utilizado o novo e-mail depois de confirmado as alterações.

Caso um usuário não deseje mais utilizar o aplicativo PortãoTech para controlar e acompanhar o uso dos portões eletrônicos, ele pode excluir seu perfil. O processo pode ser visto nas telas da figura abaixo:

Figura 26 - Exclusão do próprio perfil de usuário



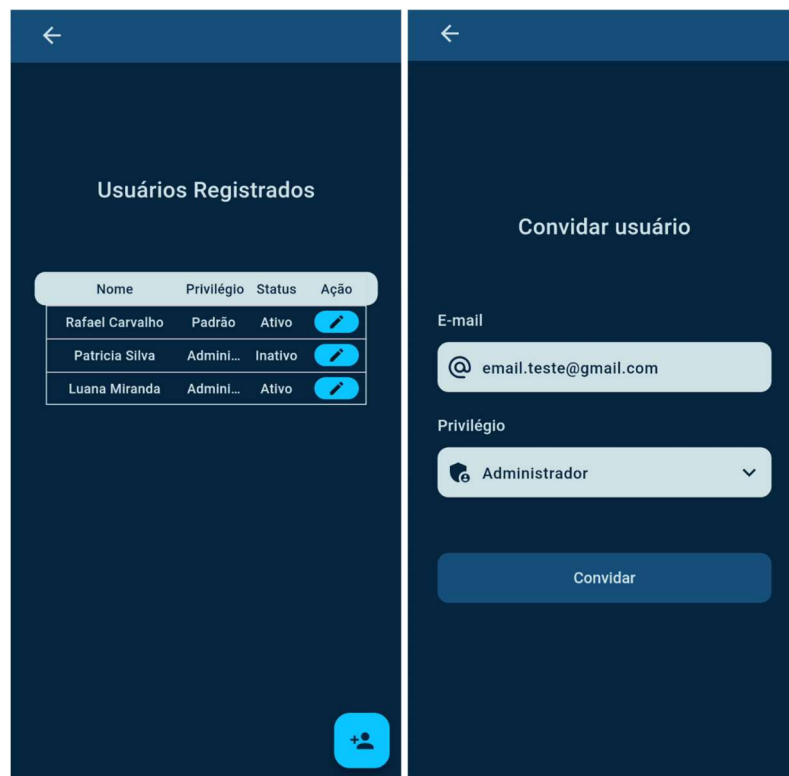
Fonte: Próprio autor.

Estando no perfil, existe um ícone de opções no canto superior direito, localizado na barra do aplicativo. O usuário deve tocar nele, sendo assim, aparecerá a opção “Excluir perfil”, ao tocar nesta opção, aparecerá um diálogo alertando sobre as consequências dessa ação, como a perda de acesso, por exemplo. Se o usuário tocar em “Confirmar”, o aplicativo prossegue com a exclusão do perfil do usuário e mostra uma mensagem de sucesso na tela. Se o usuário desistir, basta tocar fora do diálogo, ou tocar no botão voltar de seu próprio dispositivo.

4.5.1. Usuários registrados

Na tela de usuários registrados, o usuário pode visualizar, convidar outros usuários para acessar o aplicativo e controlar o acesso. A lista mostra os usuários que já estão cadastrados no aplicativo, cada usuário é representado por um item na lista, que contém as seguintes informações: nome, privilégio e *status*. O nome é o identificador do usuário no aplicativo. O privilégio pode ser padrão ou administrador, sendo que o administrador tem mais permissões para gerenciar os portões e os usuários. O *status* pode ser ativo ou inativo, indicando se o usuário está habilitado ou não para usar o aplicativo.

Figura 27 - Tela de usuários registrados e tela para convidar novos usuários

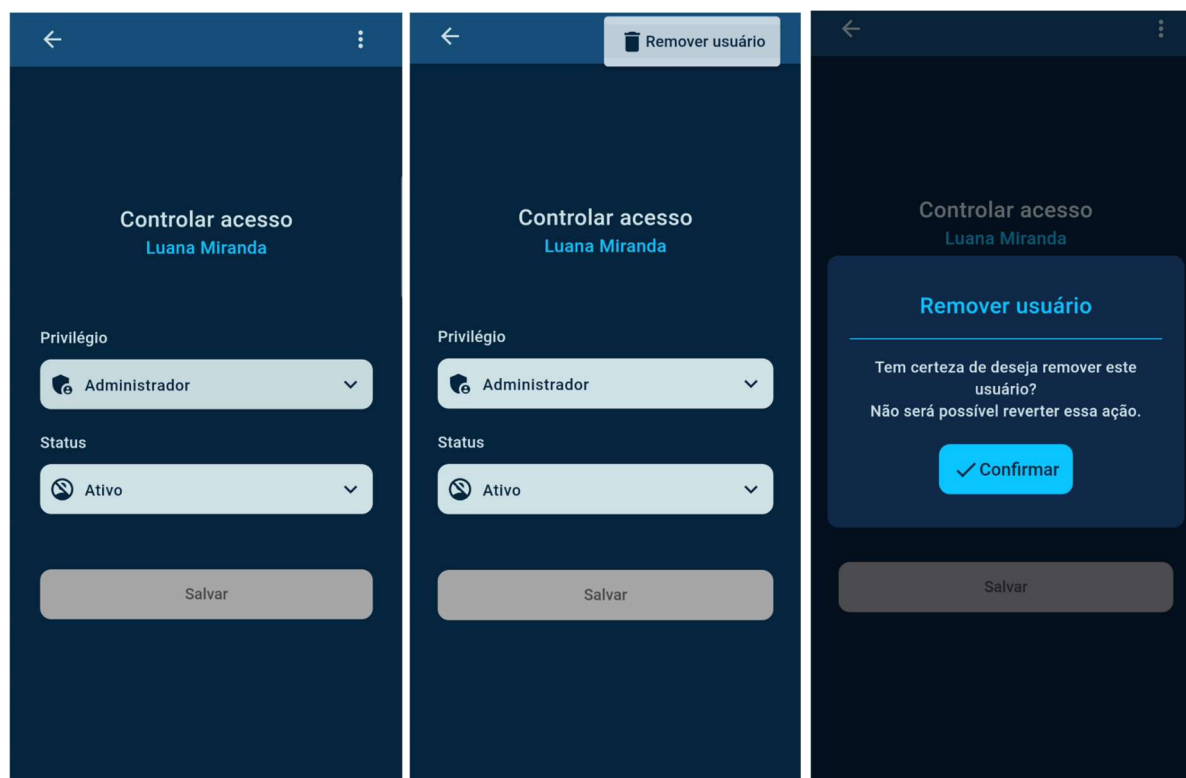


Fonte: Próprio autor.

No canto inferior direito da tela, tem um botão que tem a finalidade de abrir a tela para convidar novos usuários para a aplicação. Ao tocar nesse botão, a tela será aberta, e o usuário deve informar o e-mail do usuário convidado e o privilégio que ele terá ao finalizar seu registro. O novo usuário receberá um e-mail com um código de registro e instruções de como finalizar seu registro, conforme a Figura 18. Depois que o novo usuário finalizar seu próprio registro, já estará apto para controlar e gerenciar os portões cadastrados na aplicação.

Para cada item da lista de usuários registrados, há um botão de ação no lado direito, que tem um ícone de um lápis. Esse botão abre a tela de controle de acesso, onde os dados daquele usuário podem ser alterados. Na tela de controle de acesso, pode-se alterar o privilégio do usuário selecionado, sendo que as possibilidades são “Administrador” ou “Padrão”, e também é possível mudar o *status* para “Ativo” ou “Inativo”. Qualquer dos valores que forem alterados, habilitará o botão “Salvar”, ao tocar no mesmo, aquele usuário terá seu acesso modificado.

Figura 28 - Controle de acesso dos usuários



Fonte: Próprio autor.

Para excluir o usuário selecionado, é só tocar no ícone de opções que fica localizado no canto superior direito da tela. Com isso, aparecerá a opção “Remover usuário”, ao tocar nesta opção, aparecerá um diálogo alertando que a ação não poderá ser revertida. Se a ação for confirmada por tocar no botão “Confirmar”, o perfil é excluído e uma mensagem de sucesso é apresentada na tela. Depois da exclusão do perfil do usuário, a tela de controle de acesso do mesmo é fechada, e o aplicativo volta para a tela de usuários registrados.

4.5.2. Histórico de operações

Com essa funcionalidade, o usuário pode visualizar as operações realizadas por ele ou por outros usuários, como: abrir, fechar, abrir parcialmente, fechar parcialmente ou acionar. A tela é composta por um filtro e uma lista com detalhes das operações realizadas. A seção de filtro permite ao usuário escolher que o aplicativo filtre apenas as operações do próprio usuário, quando é selecionado a opção “Apenas o meu”, ou as operações realizadas por todos os usuários, quando é selecionado a opção “De todos”.

Figura 29 - Telas com detalhes das operações realizadas pelo portão

The image shows two side-by-side screenshots of a mobile application interface. Both screens are titled 'Histórico de operações' and feature a filter at the top with two radio buttons: 'Apenas o meu' (selected in the left screenshot) and 'De todos' (selected in the right screenshot). Below the filter is a table with columns: Nome, Operação, Data, Hora, Local, and Portão. The left screenshot shows a list of operations for Alex Silva, including actions like 'Abriu', 'Parou', 'Fechou Pa...', 'Abriu Parc...', and 'Fechou Pa...'. The right screenshot shows a list of operations for various users, including Luana Mir..., Controle RF, and Alex Silva, with similar operation types.

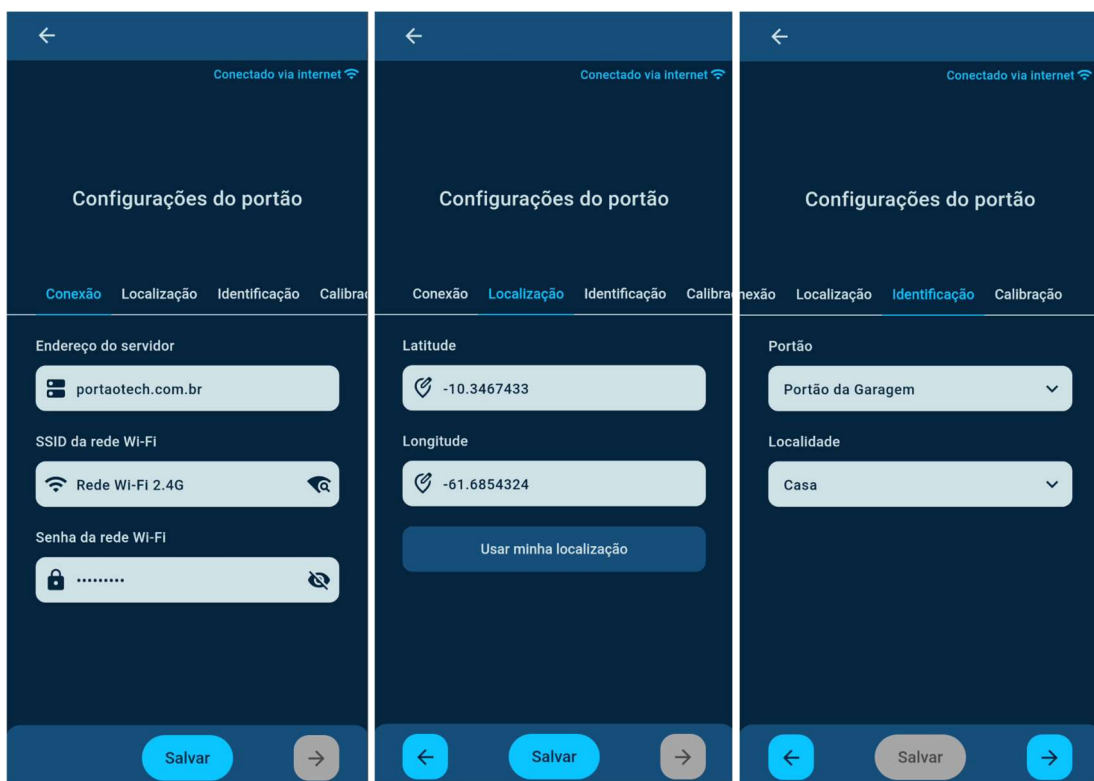
Fonte: Próprio autor.

A lista mostra as operações realizadas no aplicativo em ordem decrescente, ou seja, da mais recente para a mais antiga. Cada operação é representada por um item na lista, que contém o nome do usuário que realizou a operação, o tipo de operação realizada, data, hora, local e nome do portão. A tela também pode ajudar a identificar possíveis problemas ou irregularidades, como portões que não foram fechados adequadamente, permitindo também ter mais controle e segurança.

4.5.3. Configuração do portão

A tela de configurações do aplicativo, é projetada para oferecer uma organização intuitiva das configurações do portão, dividindo-as em abas específicas. A aba "Conexão" permite a personalização dos parâmetros de conexão, incluindo a definição do endereço do servidor, e o SSID e senha da rede local. Essas configurações são essenciais para que o portão se conecte à internet e ao servidor (Observe a Figura 30).

Figura 30 - Tela de configurações do portão



Fonte: Próprio autor.

A aba "Localização" permite o usuário visualizar e definir a "Latitude" e "Longitude" do portão. Além disso, há um botão chamado "Usar minha localização", que ao ser acionado, preenche automaticamente os campos com base na localização atual do dispositivo do usuário. Na aba "Identificação", o usuário pode personalizar o nome e a localidade do portão, essa configuração proporciona uma identificação clara e específica do mesmo. Por fim, a aba "Calibração" é dedicada à calibração interna dos parâmetros de funcionamento do portão, sua principal importância é para a obtenção do tempo que o portão levar para abrir/fechar, e relacionar o sensor com o estado do portão.

5. CONCLUSÃO

Em síntese, a pesquisa e desenvolvimento desta nova aplicação para o gerenciamento de portões eletrônicos deslizantes, representam um passo significativo em direção à modernização do acesso a tecnologias residenciais.

Os objetivos deste projeto, foram alcançados por meio de um processo minucioso e contínuo. Inicialmente, a revisão das tecnologias IoT estabeleceu as bases essenciais para o desenvolvimento do projeto. A etapa de desenvolvimento do protótipo do *hardware* focou na integração com a placa controladora do portão eletrônico, visando melhorias funcionais. Em paralelo, o *firmware* foi crucial para a comunicação entre o *hardware* e o sistema do portão. Outro aspecto significativo foi a criação e avaliação do aplicativo móvel, com detalhado planejamento, desenvolvimento e extensos testes para garantir sua funcionalidade. Por fim, o servidor da aplicação foi concebido para garantir a comunicação adequada entre dispositivos móveis e o sistema do portão.

Esse processo abrangente, desde a pesquisa até a implementação, trouxe melhorias notáveis na acessibilidade e controle dos portões eletrônicos. A escolha do microcontrolador ESP32, pautada em critérios de baixo custo e suporte a múltiplas conexões sem fio, evidencia a preocupação em oferecer uma solução acessível e resiliente, capaz de operar mesmo em condições adversas de conectividade. Ao superar as limitações dos sistemas existentes, a proposta não apenas aprimora a funcionalidade dos portões eletrônicos, mas também introduz recursos adicionais, como abertura parcial dinâmica, controle de acesso, status do portão em tempo real e histórico de acesso.

Ao alinhar eficiência operacional, acessibilidade e simplicidade no uso, a aplicação proposta não apenas otimiza a gestão de portões eletrônicos, mas também redefine a experiência do usuário. O controle intuitivo por dispositivos móveis não apenas simplifica a configuração e inclusão de usuários, mas também elimina a necessidade de aquisição de controles físicos adicionais. Nesse sentido, este trabalho apresenta um paradigma inovador, na interação cotidiana com a tecnologia residencial.

Novas funcionalidades são previstas para implementação no futuro, com o intuito de aprimorar a experiência dos usuários e ampliar a versatilidade do sistema. Uma extensão crucial seria integrar um sistema de alertas no aplicativo, informando aos usuários sobre o funcionamento regular do portão, além de notificar imediatamente sobre quaisquer defeitos detectados. Essa funcionalidade não apenas ofereceria tranquilidade aos usuários, mas também, permitiria uma intervenção rápida em caso de problemas.

Outra funcionalidade interessante, seria uma interface que apresente algumas estatísticas do portão, com gráficos que favoreçam análises, como: horários de pico, frequência de uso e até mesmo métricas de desempenho do portão. Da mesma forma, a implementação de uma tela dedicada para a configuração de abertura e fechamento automático do portão, de acordo com os dias da semana e horários, permitindo acesso sem intervenção manual. Estas ampliações agregariam valor substancial ao projeto, elevando sua funcionalidade e usabilidade para um nível ainda mais prático e completo.

REFERÊNCIAS

AFONSO, F.; STANGHERLINI, I.; MAILLARD, N. Estudo, Testes e Expansão do Projeto MPI.NET. In: WSPPD, 6., 2008, Porto Alegre. **Anais [...]**. Porto Alegre: Universidade Federal do Rio Grande do Sul, 2008. Disponível em: <<https://www.inf.ufrgs.br/gppd/wsppd/2008/papers/Afonso.pdf>>. Acesso em: 03 jul. 2022.

AGLFECHADURAS. **AGL Fechaduras**, 2022. Soluções em Segurança Eletrônica. Disponível em: <<https://www.aglfechaduras.com.br/>>. Acesso em: 29 mai. 2022.

ALMEIDA *et al.* Revisão Sistemática sobre Segurança Adaptativa Ciente de Contexto para a Internet das Coisas. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 37., 2019, Gramado. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 735-748. DOI: 10.5753/sbrc.2019.7399. Disponível em: <<https://sol.sbc.org.br/index.php/sbrc/article/view/7399>>. Acesso em: 23 mai. 2022.

ASTAH, **Astah**, 2022. Ferramenta de Modelagem UML. Disponível em: <<https://astah.net/>>. Acesso em: 02 jun. 2015.

ATLASSIAN. **Atlassian**, 2022. Jira Software. Disponível em: <<https://www.atlassian.com/>>. Acesso em: 02 jun. 2015.

BONFIOLI, G. **Banco de Dados Relacional e Objeto-Relacional: Uma Comparação Usando PostgreSQL**. 2006. 50 f. Monografia (Graduação em Ciência da Computação) – Universidade Federal de Lavras, Lavras, 2006. Disponível em: <<http://repositorio.ufla.br/handle/1/8354>>. Acesso em: 06 jun. 2022.

CÂNDIDO, C. **Aprendizagem em Banco de Dados: Implementação de Ferramenta de Modelagem E.R.** 2005. 44 f. Monografia (Pós-Graduação em Banco de Dados) – Universidade Federal de Santa Catarina em convênio com Universidade de Várzea Grande, Várzea Grande, 2005. Disponível em: <<http://www.inf.ufsc.br/~r.mello/bdnc/Especializacao-CarlosCandido-FerramentaModelagemER-2005.pdf>>. Acesso em: 22 mai. 2022.

CARVALHO, A.; LORENA, A. **Introdução à Computação: Hardware, Software e Dados**. 1. ed. Rio de Janeiro: LTC, 2017.

CASTRO FILHO, J.; FREIRE, R.; MAIA, D. Estudo de Caso como método de pesquisa em Informática na Educação. In: PIMENTEL, Mariano; SANTOS, Edméa. (Org.), Metodologia de pesquisa científica em Informática na Educação: abordagem qualitativa. Porto Alegre: SBC, 2021. **Série Metodologia de Pesquisa em Informática na Educação**, v. 3. Disponível em: <<https://metodologia.ceie-br.org/livro-3/>> Acesso em: 15 jul. 2022.

CAVALHEIRO, D. *et al.* **A Tecnologia da Informação no Agronegócio: Uma Revisão Bibliográfica**. 2018. 16 f. XVIII Mostra de Iniciação Científica, (Pós-Graduação em Administração) - UCS, Caxias do Sul, 2018. Disponível em: <<http://www.ucs.br/etc/conferencias/index.php/mostraucspgga/xviiimostrappga/paper/viewFile/5937/1968>>. Acesso em: 22 mai. 2022.

CORDEIRO, R. *et al.* IPET: aplicativo mobile que conecta ongs protetoras de animais domésticos com apoiadores. In: Encontro Anual de Computação (Enacomp), 16., 2021, [s.l.], **Anais [...]**. [s.l.]: IF Goiano, 2021. p. 27-36. Disponível em: <<https://repositorio.ifgoiano.edu.br/handle/prefix/2277>>. Acesso em: 09 jun. 2022.

COSTA, A. *et al.* **Placa Microcontrolada para a Aquisição e Condicionamento de Sinais elétricos**. 2014. 78 f. Monografia (Graduação em Engenharia de Elétrica) – Universidade São Francisco, Itatiba, 2014. Disponível em: <<http://lyceumonline.usf.edu.br/salavirtual/documentos/2608.pdf>>. Acesso em: 26 mai. 2022.

FERRARI, A. *et al.* Indústria 4.0 e Sustentabilidade: Uma Aplicação da Internet das Coisas (IoT) na Proteção Ambiental. In: ENSUS, 9., 2021, Florianópolis. **Conferência [...]**, Florianópolis: UFSC, 2021. p. 21 – 36. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/228929>>. Acesso em: 23 mai. 2022.

FIGUEIREDO, R. **Desenvolvimento de aplicação de gestão de histórico de colaboradores**. 2020, 91 f. Relatório de estágio (Licenciatura em Engenharia Informática) – Instituto Politécnico da Guarda, [s.l.], 2020. Disponível em: <<http://bdigital.ipg.pt/dspace/handle/10314/5273>>. Acesso em: 03 jul. 2022.

FRANKLIN, M. SAMUEL FILHO, R. Desenvolvimento de um Sistema de Gestão Escolar com o Uso da Linguagem Dart com Framework Flutter. **Revista Eletrônica e-F@tec**, Garça, v.10, n.1, 2020. Disponível em: <<http://ric.cps.sp.gov.br/handle/123456789/7070>>. Acesso em: 23 jun. 2022.

HORST, E.; CORDENONSI, A. O Uso das Plataformas de Prototipagem Arduino e Raspberry Pi na Educação Brasileira: Uma Revisão Sistemática de Literatura. **RENOTE**, Porto Alegre, v. 18, n. 2, p. 470–480, 2021. DOI: 10.22456/1679-1916.110287. Disponível em: <<https://seer.ufrgs.br/index.php/renote/article/view/110287>>. Acesso em: 23 jun. 2022.

JOÃO JR. *et al.* Compartilhamento de Recursos em Sistemas Distribuídos para Sistemas de Controle de Versão. In: Escola Regional de Alto Desempenho do Rio de Janeiro (ERAD-RJ), 5., 2019, Rio de Janeiro. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 11-15. Disponível em: <<https://sol.sbc.org.br/index.php/eradrj/article/view/9534>>. Acesso em: 12 jun. 2022.

KOYANAGI, F. **Introdução ao ESP32**. Fernando K Tecnologia, 2017. Disponível em: <<https://www.fernandok.com/2017/11/introducao-ao-esp32.html>>. Acesso em: 02 jun. 2022.

MACHADO JUNIOR, W. *et al.* Controle de estoque: gestão de processos utilizando a ferramenta Kanban com suporte da metodologia ágil Scrum. **Research, Society and Development: Universidade Federal de Itajubá**, Brasil, v. 8, n. 1, p. 1-21, 2019. DOI: 10.33448/rsd-v8i1.531. Disponível em: <<https://rsdjournal.org/index.php/rsd/article/view/531>> Acesso em: 29 mai. 2022.

MERCADO LIVRE. **Mercado Livre Brasil**, 2023. Disponível em: <<https://www.mercadolivre.com.br>>. Acesso em: 10 set. 2023.

MORAES, J. Tecnologia da Informação, Sistemas de Informações Gerenciais e Gestão do Conhecimento com Vistas à Criação de Vantagens Competitivas: Revisão de Literatura. **Revista Visão: Gestão Organizacional**, Caçador (SC), Brasil, v. 7, n. 1, p. 39-51, 2018. DOI: 10.33362/visao.v7i1.1227. Disponível em: <<https://periodicos.uniarp.edu.br/index.php/visao/article/view/1227>>. Acesso em: 22 mai. 2022.

MORI, A. **Desenvolvimento de um Plug-In para a Contagem de Pontos por Função no Software Astah**. 2015. 70 f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) - Universidade Tecnológica Federal do Paraná, Cornélio Procópio, 2015. Disponível em: <<https://repositorio.utfpr.edu.br/jspui/handle/1/7113>>. Acesso em: 02 jun. 2015.

NIEHUES, L. **Estudo e Criação de um Editor de Código Estruturado**. 2013, 78 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade Federal de Santa Catarina, Florianópolis, 2013. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/184206>>. Acesso em: 12 jun. 2022.

NIQUINI, C. **Anota-e: uma ferramenta multiplataforma para pesquisa de preços e controle de gastos domésticos**. 2021. 86 f. Monografia (Graduação em Engenharia de Computação) - Instituto de Ciências Exatas e Aplicadas, Universidade Federal de Ouro Preto, João Monlevade, 2021. Disponível em: <<https://www.monografias.ufop.br/handle/35400000/3495>>. Acesso em: 03 jul. 2022.

NUNES, A. **ARVIM: Automação Residencial Via Mobile**. 2018. 80 f. Monografia (Graduação em Engenharia Elétrica) – Universidade de Brasília, [s.l.], 2018. Disponível em: <https://bdm.unb.br/bitstream/10483/27960/1/2018_AlvaroHenriqueFigueiredoNunes_tcc.pdf>. Acesso em: 27 mai. 2022.

OLIVEIRA, N.; SIRQUIERA, T. Uma Plataforma para Coleta e Análise de Dados do GitHub. In: Seminário de Extensão e Pesquisa, 7., 2021, Juiz de Fora, **Anais [...]**, Juiz de Fora: Centro Universitário Academia, 2021. Disponível em: <<https://seer.uniacademia.edu.br/index.php/ANL/article/view/3103>>. Acesso em: 13 jun. 2022.

PINOTTI, M. **Sistema Mobile para Utilização de Vagas de Estacionamento Preferenciais com Auxílio de Sensores**. 2022. 72 f. Trabalho de Conclusão de Curso (Bacharelado em Sistema de Informação) - Instituto Federal do Espírito Santo, Colatina, 2022. Disponível em: <<https://repositorio.ifes.edu.br/handle/123456789/1676>>. Acesso em: 20 jun. 2022.

PIZETTA, D. **Biblioteca, API e IDE para o Desenvolvimento de Projetos de Metodologias de Ressonância Magnética**. 2014. 93 f. Dissertação (Pós-Graduação em Física) – Instituto de Física de São Carlos da Universidade de São Paulo, São Carlos, 2014. DOI: 10.11606/D.76.2014.tde-28042014-160738. Disponível em: <<https://teses.usp.br/teses/disponiveis/76/76132/tde-28042014-160738/pt-br.php>>. Acesso em: 12 jun. 2022.

PPA. **PPA**, 2022. Maior indústria de automatização de portas, portões e cancelas da América Latina. Disponível em: <<https://ppa.com.br/>>. Acesso em: 29 mai. 2022.

PRESSMAN, R. **Engenharia de Software: Uma Abordagem Profissional**. 7. Edição. Porto Alegre: AMGH, 2011.

PROETTI, S. As pesquisas qualitativa e quantitativa como métodos de investigação científica: Um estudo comparativo e objetivo. **Revista Lumen**, [s.l.], v. 2, n. 4, 2018. Disponível em: <<http://www.periodicos.unifai.edu.br/index.php/lumen/article/view/60>>. Acesso em 14 jul. 2022.

ROCHA JÚNIOR, B. **Projeto de Automação Residencial com a Utilização do Arduino**. 2014. 64 f. Trabalho de Conclusão de Curso (Graduação de Tecnologia em Automação Industrial) - Universidade Tecnológica Federal do Paraná, Cornélio Procópio, 2014. Disponível em: <https://repositorio.utfpr.edu.br/jspui/bitstream/1/27314/1/CP_COAUT_2014_1_05.pdf>. Acesso em: 12 nov. 2023.

SALLES, L. **Desenvolvimento de Um Dispositivo de Medição de Ruído com Base no Sistema OpenWrt**. 2017. 83 f. Dissertação (Mestrado em Engenharia Mecânica) - Universidade Estadual Paulista, Ilha Solteira, 2017. Disponível em: <<http://hdl.handle.net/11449/154132>>. Acesso em: 25 mai. 2022.

SANTOS, B. *et al.* Automação de Casas e Estabelecimentos Comerciais Através de Microcontroladores: Uma Revisão da Aplicabilidade do Arduino. **Revista Tecnológica da Fatec Americana**, [s.l.], v. 8 n. 2, p. 70-80, 2020. DOI: 10.47283/244670492020080270. Disponível em: <<https://fatec.edu.br/revista/index.php/RTecFatecAM/article/view/245>>. Acesso em: 23 mai. 2022.

SANTOS, J.; LARA JUNIOR, R. **Sistema de Automação Residencial de Baixo Custo Controlado Pelo Microcontrolador ESP32 e Monitorado Via Smartphone**. 2019, 46 f. TCC (Tecnólogo em Automação Industrial) - Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2019. Disponível em: <<http://repositorio.utfpr.edu.br/jspui/handle/1/16960>>. Acesso em: 23 mai. 2022.

SCHWAB, K. **A Quarta Revolução Industrial**. 1. ed. São Paulo: Edipro, 2016.

SCHWABER, K.; SUTHERLAND, J. **Guia do Scrum: O Guia Definitivo para o Scrum: As Regras do Jogo**. 2020. Disponível em: <<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-2.0.pdf>>. Acesso em: 23 abr. 2022.

SONOFF. **SONOFF**, 2022. Site oficial da indústria chinesa que desenvolve dispositivos de automação residencial. Disponível em: <<https://sonoff.tech/>>. Acesso em: 29 mai. 2022.

SOUZA, A. *et al.* Automação residencial e eficiência energética: um estudo de caso. **Brazilian Journal of Development**, Curitiba (PR), v. 5, n. 8, p. 13086-13101, 2019. DOI: 10.34117/bjdv5n8-124. Disponível em: <<https://brazilianjournals.com/index.php/BRJD/article/view/2899>>. Acesso em: 29 mai. 2022.

SOUZA, J.; OLIVEIRA, M. METODOLOGIAS ÁGEIS: Um Comparativo Entre Extreme Programming (XP) e Scrum. **Ciência & Tecnologia: FATEC-JB**, Jaboticabal (SP), v. 13, n. 1, p. 133-141, 2021. DOI: 10.52138/citec.v13i1.205. Disponível em: <<https://citec.fatecjaboticabal.edu.br/index.php/citec/article/view/205>>. Acesso em: 29 mai. 2022.

APÊNDICE A – PROJETO DE *SOFTWARE*

Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – IFRO
Campus Ji-Paraná

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

PROJETO DO SISTEMA PORTÃO TECH

Cliente: Inovação Portões Eletrônicos

Equipe: Tech Developers

Ji-Paraná, 2023

ALEX SILVA

PROJETO DO SISTEMA PORTÃO TECH

Projeto de Software elaborado e apresentado como requisito de nota na disciplina de Desenvolvimento de Projeto Científico e Tecnológico do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do IFRO Campus Ji-Paraná, sob a orientação do professor Me. Jackson Henrique da Silva Bezerra.

Ji-Paraná, 2023

SUMÁRIO

1. DOCUMENTO DE ESPECIFICAÇÃO DE SOFTWARE.....	5
1.1. Dados da Empresa Cliente.....	5
1.2. Produto	5
1.3. Aspectos Técnicos do Produto.....	5
1.3.1. Linguagens e Ferramentas.....	5
1.3.2. Arquitetura da aplicação.....	6
1.4. Missão do produto	8
1.5. Requisitos de Qualidade.....	8
1.6. Objetivos Gerenciais do Projeto.....	9
1.7. Equipe de Desenvolvimento.....	10
1.8. Metodologia de Desenvolvimento.....	10
1.8.1. Sobre a Metodologia	10
1.8.2. Ciclo de Vida da Metodologia no Projeto do Sistema	13
2. DOCUMENTO DE REQUISITOS	22
2.1. Conectar com servidor	22
2.2. Finalizar registro	22
2.3. Fazer login	23
2.4. Controlar portão.....	23
2.5. Visualizar perfil	24
2.6. Visualizar usuários	24
2.7. Convidar usuário	25
2.8. Controlar acesso	25
2.9. Visualizar histórico.....	26
2.10. Configurar portão.....	26
3. PROJETO DE INTERFACE	28
3.1. Conectar com servidor	28
3.2. Finalizar registro	28
3.2.1. Validar registro de usuário	28
3.3. Fazer login	28
3.4. Controlar portão.....	29
3.4.1. Controlar outro portão.....	29
3.4.2. Controlar portão por bluetooth.....	29
3.4.3. Selecionar portão bluetooth.....	29

3.4.4.	Menu principal	30
3.5.	Visualizar perfil	30
3.5.1.	Editar perfil	30
3.5.2.	Confirmar edição de perfil	30
3.6.	Visualizar usuários	31
3.7.	Convidar usuário	31
3.8.	Controlar acesso	31
3.9.	Visualizar histórico.....	31
3.10.	Configurar portão: Conexão.....	32
3.10.1.	Selecionar rede Wi-Fi.....	32
3.10.2.	Configurar portão: Localização	32
3.10.3.	Configurar portão: Identificação	32
3.10.4.	Configurar portão: Calibração	33
3.10.5.	Confirmação de estado	33
4.	CASO DE USO	34
4.1.	Diagrama de Caso de Uso	34
4.2.	Caso de Uso Expandido.....	35
4.2.1.	Conectar com servidor	35
4.2.2.	Finalizar registro	35
4.2.3.	Fazer login.....	36
4.2.4.	Controlar portão	37
4.2.5.	Visualizar perfil.....	38
4.2.6.	Visualizar usuários	39
4.2.7.	Convidar usuário	40
4.2.8.	Controlar acesso	40
4.2.9.	Visualizar histórico	41
4.2.10.	Configurar portão	42
5.	DIAGRAMA DE CLASSES.....	44
6.	DIAGRAMAS DE SEQUÊNCIA	45
6.1.	Conectar com servidor	45
6.2.	Finalizar registro	45
6.3.	Fazer login	46
6.4.	Controlar portão.....	46
6.4.1.	Controlar portão pela internet	46
6.4.2.	Controlar portão por bluetooth.....	47
6.5.	Visualizar perfil	47
6.6.	Visualizar usuários	48

6.7. Convidar Usuário	48
6.8. Controlar acesso	48
6.9. Visualizar histórico.....	49
6.10. Configurar portão.....	49
6.10.1. Configurar conexão do portão	49
6.10.2. Configurar localização do portão	50
6.10.3. Configurar identificação do portão.....	50
6.10.4. Configurar calibração do portão.....	51
7. DIAGRAMA DE BANCO DE DADOS	52

1. DOCUMENTO DE ESPECIFICAÇÃO DE SOFTWARE

1.1. Dados da Empresa Cliente

Nome Fantasia: Inovação Portões Eletrônicos

Razão Social: Inova Portões e Cia – ME

CNPJ: 33.223.322/0001-32

Endereço: Av. Marechal Rondon, 321 – Centro – Ji-Paraná-RO – 76900-036

Telefones: 3423-2233 / 99322 2233

Cliente/Master: Carlos de Souza

Cliente/Usuários: Carlos de Souza

Descrição da Empresa/Comércio: A Inovação Portões Eletrônicos é uma empresa bem jovem, com apenas 4 anos de existência, sua atuação desde o princípio é na instalação de portões eletrônicos deslizantes. Os serviços de instalação oferecidos são de ótima qualidade, devido a isso sempre surgem clientes com novas demandas e até mesmo solicitam novas funcionalidades. Contudo, os atuais portões oferecidos pela empresa não são informatizados, ou seja, são de controles simples, apenas com funcionalidade de abertura e fechamento. Como a empresa sempre visa o bem estar, flexibilidade, segurança e tranquilidade de seus clientes, está sempre em busca de agregar novos recursos e tecnologias em seus serviços prestados.

1.2. Produto

Será desenvolvido uma aplicação para informatizar o funcionamento dos atuais portões eletrônicos, com isso a Inovação Portões Eletrônicos terá a opção de oferecer um produto mais sofisticado para seus clientes. A aplicação será chamada de Portão Tech.

1.3. Aspectos Técnicos do Produto

1.3.1. Linguagens e Ferramentas

Durante o processo de análise, será desenhado os diagramas com a ferramenta de modelagem Astah, sendo assim será possível desenvolver os principais diagramas UML, como por exemplo: o diagrama de caso de uso, diagrama de atividades,

diagrama de classes, entre outros julgados necessários. Já para os diagramas de banco de dados, será utilizado a ferramenta Br Modelo e para as interfaces do aplicativo a ferramenta de design Figma com o apoio do Photoshop.

Durante o processo de desenvolvimento, será desenvolvido o aplicativo front-end na linguagem de programação Dart utilizando o framework Flutter, e a IDE a ser utilizada neste processo é o Android Studio. O Firmware será feito na linguagem de programação C/C++ para microcontroladores, utilizando a ferramenta Visual Studio Code e/ou Arduino IDE. O back-end será desenvolvido na linguagem C# com o framework .NET, utilizando a IDE Visual Studio.

Os dados do back-end serão armazenados no sistema de gerenciamento de banco de dados PostgreSQL. Os e-mails serão enviados por meio de um serviço SMTP. A comunicação do aplicativo com o hardware pela internet, ocorrerá por meio do broker MQTT, conhecido pelo nome Mosquitto. Com isso, aplicativo publica a mensagem no broker, e o broker por sua vez entrega a mensagem para o hardware, assim ocorre também no processo inverso.

Para a composição do hardware, o microcontrolador escolhido foi o ESP32, pois já tem os principais recursos necessários integrados, como o Bluetooth e Wi-Fi. Será utilizado a ferramenta Jira para que a equipe possa gerenciar os projetos, e com isso aplicar a metodologia de desenvolvimento ágil Scrum com mais fluidez e facilidade. O controle de versão será realizado pelo Git durante o processo de desenvolvimento, sendo todo o código fonte sincronizado em um repositório remoto no GitHub.

1.3.2. Arquitetura da aplicação

Na comunicação, será adotada a estrutura cliente-servidor para conectar todos os elementos do projeto. O aplicativo funcionará como cliente, enquanto a camada server-side será composta pelo Servidor da Aplicação e o Broker MQTT. O hardware terá um papel de cliente e servidor, funcionando como cliente ao se conectar à camada server-side pela internet e como servidor ao se comunicar via bluetooth diretamente com o aplicativo.

A arquitetura da aplicação será composta por vários componentes que se comunicam de diferentes formas: alguns fisicamente e outros sem fio. A comunicação entre o hardware da aplicação e o portão eletrônico ocorrerá através de fios, sensores

e atuadores, enquanto a comunicação do aplicativo com o hardware se dará sem fio, podendo ocorrer por bluetooth ou pela internet, que será o modo padrão.

A conexão via bluetooth será utilizada na configuração inicial do portão ou quando a conexão com a internet não estiver disponível. Mesmo utilizando bluetooth, as operações serão sincronizadas com o servidor assim que a conexão com a internet for restabelecida. Quando conectado pela internet, o aplicativo enviará os comandos para a camada server-side, que os repassará para execução no portão eletrônico.

Após o acionamento do portão, o hardware registrará a operação realizada no servidor. Cada operação incluirá informações como tipo (Abertura, Fechamento, Parada e Acionamento), portão acionado, usuário responsável, e a data e hora do acionamento. O histórico dessas operações poderá ser visualizado posteriormente pelo aplicativo, que acessará esses dados diretamente do servidor.

O Aplicativo Móvel será o front-end da aplicação, proporcionando interfaces e funcionalidades intuitivas para os usuários, permitindo configurações no portão e operações de abertura e fechamento.

O Servidor da Aplicação será uma API hospedada na internet, utilizando HTTP e JSON para trocar dados. Oferecerá autenticação segura para os usuários e registrará as operações do portão, permitindo o acesso e gerenciamento dos dados importantes relacionados aos dispositivos e usuários registrados na aplicação.

O Broker MQTT, também hospedado na internet, facilitará a comunicação entre o aplicativo de smartphone e o hardware do portão eletrônico, empregando o modelo de comunicação publish-subscribe, ideal para aplicações de Internet das Coisas.

O Portão Eletrônico, traz praticidade ao possibilitar a abertura sem a necessidade de deslocamento físico, sendo composto por um motor elétrico, engrenagens e uma central eletrônica responsável por gerenciar o funcionamento do motor e seus componentes.

O Hardware da Aplicação será a parte física do sistema, uma placa eletrônica repleta de componentes que controlarão o funcionamento do portão eletrônico pelo celular. A placa incluirá um microcontrolador, sensor magnético, e conexões Wi-Fi e Bluetooth para comunicação com o servidor e o aplicativo.

Todos estes elementos irão compor a arquitetura da aplicação, destacando-se o funcionamento do aplicativo móvel, do servidor, do broker MQTT, do portão eletrônico e do hardware da aplicação.

1.4. Missão do produto

A aplicação Portão Tech tem como missão sofisticar e informatizar os portões eletrônicos já existentes, trazendo a eles novos recursos, que geralmente não são obtidos durante sua compra. Além disso traz mais segurança e flexibilidade para os usuários e clientes da Inovação Portões Eletrônicos.

1.5. Requisitos de Qualidade

O produto a ser desenvolvido deverá atender os seguintes requisitos de qualidade de software:

- A interface gráfica será desenvolvida para que seja intuitiva, simples e de fácil utilização.
- Será utilizado ícones nos menus para que seja mais fácil a identificação das funcionalidades pelos usuários.
- O funcionamento será estável com boa interoperabilidade para que os componentes do sistema possam se sincronizar para obter trocas de informações de maneira eficaz.
- As funcionalidades serão implementadas utilizando recursos e técnicas para garantir a segurança dos usuários e seus dados pessoais.
- Se manterá operacional mesmo durante a falta de conexão com a internet por meio de conexão bluetooth.
- A Configuração será simplificada para que os usuários não tenham dificuldades para utilizar o produto.
- O hardware da aplicação a ser desenvolvido será apenas incorporado no sistema do portão eletrônico sem exigir modificações ou adulterações em sua originalidade.
- A aplicação será responsiva para que possa se adaptar em dispositivos com diferentes configurações de hardware.
- Os campos serão validados para impedir envios de dados incorretos e garantir o fluxo completo das operações.

- Exceções serão tratadas para que a aplicação continue funcionando mesmo na ocorrência de erros ou demora de resposta entre os componentes do sistema.

1.6. Objetivos Gerenciais do Projeto

Entregas: Ao final de cada Sprint será entregue ao cliente uma versão da aplicação para que o mesmo possa avaliar o andamento do desenvolvimento da aplicação.

Prazo: O desenvolvimento do projeto terá a duração de 18 meses.

Custos: Segue detalhes dos custos do projeto nas tabelas a seguir:

Tabela 01 - Custo mensal do projeto

Item	Descrição	Quantidade	Valor Unitário	Valor Total
01	Remuneração de membros	02	R\$ 3.000,00	R\$ 6.000,00
02	Energia	01	R\$ 100,00	R\$ 100,00
03	Outros Gastos	01	R\$ 200,00	R\$ 200,00
			Total geral	R\$ 6.300,00

Tabela 2 - Custo total do projeto

Item	Descrição	Tempo Previsto	Valor Unitário	Valor Total
01	Custo mensal	18 meses	R\$ 6.300,00	R\$ 113.400,00
			Total geral	R\$ 113.400,00

Manutenção: Após a implantação da aplicação, a mesma terá um custo mensal de R\$ 750,00 para realização de manutenções.

Contrato: Este projeto ficará valendo como contrato de prestação de serviço entre as partes envolvidas, na necessidade de alteração do mesmo por qualquer das partes, será necessário a comunicação com no mínimo 24 horas de antecedência;

Comunicação: Para tornar a comunicação mais prática, a equipe utilizará a ferramenta de reuniões Google Meet.

1.7. Equipe de Desenvolvimento

Ficou definido para este projeto o uso da metodologia de ágil Scrum, sendo assim a equipe foi organizada pelas seguintes funções e atribuições.

Nome	Função	Atribuições
Alex Silva	Scrum Master	Liderar a equipe.
Prof. Jackson	Product Owner	Ajudar no entendimento do negócio.
Alex Silva	Time Scrum	Desenvolver o projeto de software.

1.8. Metodologia de Desenvolvimento

Para o gerenciamento do projeto da aplicação Portão Tech, foi definido pela equipe que a metodologia ágil a ser utilizada seria o Scrum. A escolha foi feita levando em consideração que a equipe já possui conhecimento, e experiência com o uso dessa metodologia. Até mesmo vale ressaltar que é uma metodologia de desenvolvimento muito usada atualmente por ser bem difundida e entregar ótimos resultados em tempo hábil.

1.8.1. Sobre a Metodologia

A metodologia de desenvolvimento Scrum começou a ser idealizada no início dos anos 1990, por Ken Schwaber e Jeff Sutherland. A partir de uma conferência realizada em 1995, seus principais criadores fizeram a primeira apresentação formal de seu trabalho, desde então a metodologia Scrum começou a ser conhecida e difundida pelas pessoas. Contando com a colaboração de outros indivíduos puderam refinar bastante sua definição, no objetivo de ajudar as pessoas a compreender e aplicar a metodologia por eles criada, publicaram o primeiro guia em 2010 chamado Guia do Scrum.

Por definição, o Scrum é um Framework simples que tem por finalidade, ajudar os indivíduos, organizações e grupos de pessoas a resolverem problemas mais complexos, garantindo valor no produto final. A base do Scrum é o empirismo e lean thinking, isso significa que, a partir da experiência que se obtém o conhecimento e habilidade para desenvolver as tarefas propostas, e por focar nas tarefas mais importantes reduz perdas. A colaboração entre pessoas e grupos é essencial para

aplicação do Scrum, pois é através disso que a equipe consegue obter mais clareza e conhecimento para a realização das tarefas.

Para obter mais controle, em relação aos riscos durante o desenvolvimento do projeto, ganho de previsibilidade e clareza, o Scrum aplica os modelos iterativo e incremental. As engrenagens do Scrum se mantêm funcionando através da relação e colaboração mútua de indivíduos, todos esses indivíduos formam o que é denominado Scrum Team, que é dividido em três papéis fundamentais. Sendo eles um Scrum Master, um Product Owner e os Developers.

O Scrum Master é responsável por liderar a equipe, garantindo que os padrões e definições do Scrum sejam bem aplicados por todos os envolvidos, até mesmo por toda a organização. Outras de suas responsabilidades são, o gerenciamento e treinamento de todo o time, contribuir na elaboração de incrementos de alto valor, remover os impedimentos ocorridos durante o desenvolvimento do projeto e garantir que os eventos sigam o tempo como é estabelecido.

Já o Product Owner é o conhecedor do produto, ficando responsável por gerenciar e garantir a entrega de um produto de alto valor na etapa final. Deve ter boa comunicação e deixar bem claro quais as metas do produto, definir bem os itens do Product Backlog assim como também transmitir para os Developers, ordenar o Product Backlog de acordo com o nível de prioridade de cada funcionalidade e garantir transparência entre todos os envolvidos.

Os Developers são pessoas responsáveis por desenvolver os incrementos que são entregues para o cliente ao final de cada Sprint, cada incremento é uma parte funcional do projeto, somada aos incrementos anteriores. Os conhecimentos e habilidades que cada Developer precisa ter é bem variado e é relacionado ao domínio necessário para a realização das tarefas definidas em cada Sprint. São responsáveis por criar os planos do Sprint Backlog, adaptar os planos diariamente com o objetivo de atingir a meta da Sprint e desenvolver visando qualidade até a Definição de Pronto.

O componente mais importante do Scrum é a Sprint, é nela que o time começa agregar valor ao projeto, transformando planos e ideias no produto. Sprints são eventos que duram geralmente 2 ou 4 semanas, ao finalizar uma Sprint uma nova deve ser iniciada. No decorrer da Sprint, nenhuma mudança que afete negativamente o seu andamento pode acontecer, se mantêm a qualidade e sempre que necessário o Product Backlog pode ser refinado. É possível prever, inspecionar e adaptar o progresso rumo ao produto pelo menos uma vez dentro de seu período de duração.

Dentro do escopo da sprint ocorrem 4 eventos, denominados Sprint Planning, Daily Scrums, Sprint Review e Sprint Retrospective. Segue abaixo detalhes sobre cada um dos eventos:

- **Sprint Planning:** Evento onde é definido o que será realizado na Sprint, todo o Scrum Team trabalha neste plano em colaboração e tem duração máxima de 8 horas. Outras pessoas também podem ser convidadas para participar, sendo o Product Owner responsável por garantir que todos discutam de maneira produtiva, para que seja definido as funcionalidades de mais importância do Product Backlog. A partir da discussão, os Developers definem quais os itens do Product Backlog irão para a Sprint atual, de modo que a soma do tempo dos itens não ultrapasse a duração da Sprint.
- **Daily Scrum:** Tem o intuito de observar o andamento da Sprint e adaptar o Sprint Backlog para os próximos planejamentos. O tempo máximo é de 15 minutos e deve sempre ser realizado o evento no mesmo local e horário nos dias úteis. Os Developers apresentam o que fizeram antes da Daily, o que farão depois e quais foram os problemas encontrados que dificultaram o andamento das tarefas. É um momento essencial para identificar os impedimentos e tomar novas decisões.
- **Sprint Review:** Tem o propósito de refletir sobre mudanças futuras e analisar o resultado da Sprint, com duração máxima de 4 horas. As partes interessadas, denominadas Stakeholders, visualizam o resultado apresentado pelo Scrum Team, sendo então debatido questões relacionadas ao progresso do produto e o que deve ser feito a seguir.
- **Sprint Retrospective:** Momento para refletir sobre como foi o andamento da Sprint e discutir sobre o que deu certo e o que deu errado. Quais problemas foram ou não resolvidos, o que for mais significativo deverá ser resolvido de maneira mais prioritária. O tempo máximo de duração, denominado Timebox, é de 3 horas.

Os artefatos são componentes que constituem o valor do produto, precisam ser definidos de maneira clara e transparente para que todos compreendam e se adaptem de forma rápida ao domínio. Cada artefato tem por finalidade um propósito específico, o Product Backlog é composto por itens do produto, o Sprint Backlog por itens da

Sprint e o incremento é a Sprint definida como pronto. O primeiro artefato a ser desenvolvido é o Product Backlog, que nada mais é do que uma lista produzida de maneira ordenada com tudo o que é necessário para desenvolver e melhorar o produto.

A partir do Product Backlog, é selecionado as funcionalidades que irão para a Sprint no evento Sprint Planning. O Scrum Team sempre realiza refinamentos e quebra os itens em pedaços menores para torná-los mais transparentes e claros, e os Developers são os principais responsáveis por dimensionar cada item do Product Backlog. O Sprint Backlog é composto pelos itens do Product Backlog que foram selecionados para a Sprint, sendo atualizados constantemente pelos Developers durante o andamento da Sprint. Já o incremento é cada Sprint finalizada, somada com os incrementos anteriores, definido como pronto e apresentado ao cliente.

1.8.2. Ciclo de Vida da Metodologia no Projeto do Sistema

1.8.2.1. Product Backlog

Item	Descrição	Estimativa de Esforço	Nível de Prioridade
01	Criar Documento de Especificação de Software	40h	100
02	Criar Documento de Requisitos de Software	60h	95
03	Construir Projeto de Interface	20h	90
04	Criar Diagrama de Caso de Uso	1h	90
05	Criar Caso de Uso Expandido	10h	90
06	Construir Modelos do Banco de Dados	2h	90
07	Criar Esquema Eletrônico do Hardware	4h	90
08	Desenvolver Circuito de Alimentação do Hardware	40h	98
09	Adicionar no Hardware Sensor para Obter o Estado do Portão	30h	95
10	Adicionar no Hardware Vias para Obter e Enviar Sinais Entre o Hardware da Aplicação e a Central do Motor do Portão	15h	95
11	Adicionar no Hardware Botão para o Reset do Firmware	10h	80
12	Adicionar no Hardware Leds indicadores	5h	85
13	Desenvolver no Firmware Funcionalidade para Obter Leitura do Sensor de Identificação do Estado do Portão	40h	95

14	Desenvolver no Firmware Funcionalidade para Obter e Enviar Sinais Entre o Hardware da Aplicação e a Central do Motor do Portão	40h	95
15	Desenvolver no Firmware Funcionalidade para Fazer o Reset	40h	85
16	Desenvolver no Firmware Funcionalidade para Gerenciar os LEDs Indicadores do Hardware	40h	80
17	Configurar o broker MQTT	10h	90
18	Desenvolver no Firmware Funcionalidade para conversar com o broker MQTT	30h	95
19	Criar o Core do Servidor da Aplicação	25h	98
20	Desenvolver a Autenticação por Token JWT no Servidor da Aplicação	15h	95
21	Realizar Conexão com o Banco de Dados no Servidor da Aplicação	10h	90
22	Configurar serviço de E-Mail no Servidor da Aplicação	20h	85
23	Adicionar Rotas com Todos os Endpoints necessários para o funcionamento da aplicação e desenvolver suas funcionalidades	150h	98
24	Desenvolver no Firmware Funcionalidades da Comunicação Via Bluetooth com o Aplicativo	20h	90
25	Desenvolver no Firmware Funcionalidades para Comunicação com o Servidor da Aplicação na Internet pelo Wi-Fi	40h	95
26	Desenvolver no Firmware Funcionalidade de Sincronização de Dados com o Servidor da Aplicação	20h	95
27	Adicionar Dependências do Aplicativo	10h	95
28	Desenvolver Core do Aplicativo	20h	98
29	Desenvolver as interfaces do aplicativo	120h	95
30	Desenvolver os controladores do aplicativo	150h	95
31	Desenvolver no Aplicativo Recurso para Persistir os Dados	10h	90
32	Desenvolver no Aplicativo Recurso para Comunicação com o broker MQTT	20h	98
33	Desenvolver no Aplicativo Recurso para Comunicação Via Bluetooth	30h	95
34	Desenvolver no Aplicativo Recurso para Comunicação com o Servidor da Aplicação	40h	98

1.8.2.2. Sprints Backlog

Item: Sprint 1.

Descrição: Desenvolver documentos, diagramas e interfaces do aplicativo.

Período: 01/04/2022 à 30/04/2022.

Item do Product Backlog	Responsável	Período
01 - Criar Documento de Especificação de Software	Alex Silva	Dias 01 a 05
02 - Criar Documento de Requisitos de Software	Alex Silva	Dias 06 a 13
03 - Construir Projeto de Interface	Alex Silva	Dias 14 a 16
04 - Criar Diagrama de Caso de Uso	Alex Silva	Dias 21 a 22
05 - Criar Caso de Uso Expandido	Alex Silva	Dias 23 a 24
06 - Construir Modelos do Banco de Dados	Alex Silva	Dias 25 a 26
07 - Criar Esquema Eletrônico do Hardware	Alex Silva	Dias 27 a 30

Item: Sprint 2.

Descrição: Desenvolver o hardware.

Período: 01/07/2022 à 31/07/2022.

Item do Product Backlog	Responsável	Período
08 - Desenvolver Circuito de Alimentação do Hardware	Alex Silva	Dias 01 a 07
09 - Adicionar no Hardware Sensor para Obter o Estado do Portão	Alex Silva	Dias 08 a 12
10 - Adicionar no Hardware Vias para Obter e Enviar Sinais Entre o Hardware da Aplicação e a Central do Motor do Portão	Alex Silva	Dias 13 a 18
11 - Adicionar no Hardware Botão para o Reset do Firmware	Alex Silva	Dias 19 a 25
12 - Adicionar no Hardware Leds indicadores	Alex Silva	Dias 25 a 31

Item: Sprint 3.

Descrição: Iniciar o desenvolvimento do firmware com a funcionalidade de leitura do sensor, para obter estado do portão.

Período: 01/08/2022 à 31/08/2022.

Item do Product Backlog	Responsável	Período
13 - Desenvolver no Firmware Funcionalidade para Obter Leitura do Sensor de Identificação do Estado do Portão	Alex Silva	Dias 01 a 31

Item: Sprint 4.

Descrição: Desenvolver no firmware a funcionalidade para acionar o portão eletrônico.

Período: 01/09/2022 à 30/09/2022.

Item do Product Backlog	Responsável	Período
14 - Desenvolver no Firmware Funcionalidade para Obter e Enviar Sinais Entre o Hardware da Aplicação e a Central do Motor do Portão	Alex Silva	Dias 01 a 30

Item: Sprint 5.

Descrição: Desenvolver no firmware a funcionalidade para fazer o reset do hardware.

Período: 01/10/2022 à 31/10/2022.

Item do Product Backlog	Responsável	Período
15 - Desenvolver no Firmware Funcionalidade para Fazer o Reset	Alex Silva	Dias 01 a 31

Item: Sprint 6.

Descrição: Desenvolver no firmware a funcionalidade para gerenciar os LEDs indicadores.

Período: 01/11/2022 à 30/11/2022.

Item do Product Backlog	Responsável	Período
16 - Desenvolver no Firmware Funcionalidade para Gerenciar os LEDs Indicadores do Hardware	Alex Silva	Dias 01 a 30

Item: Sprint 7.

Descrição: Configurar o broker MQTT e desenvolver no firmware a funcionalidade de comunicação com o mesmo.

Período: 01/01/2023 à 31/01/2023.

Item do Product Backlog	Responsável	Período
17 - Configurar o broker MQTT	Alex Silva	Dias 01 a 05
18 - Desenvolver no Firmware Funcionalidade para conversar com o broker MQTT	Alex Silva	Dias 06 a 31

Item: Sprint 8.

Descrição: Desenvolver o core do servidor, já com autenticação JWT e conexão com o SGBD.

Período: 01/02/2023 à 28/02/2023.

Item do Product Backlog	Responsável	Período
19 - Criar o Core do Servidor da Aplicação	Alex Silva	Dias 01 a 05
20 - Desenvolver a Autenticação por Token JWT no Servidor da Aplicação	Alex Silva	Dias 06 a 15
21 - Realizar Conexão com o Banco de Dados no Servidor da Aplicação	Alex Silva	Dias 16 a 28

Item: Sprint 9.

Descrição: Configurar serviço de e-mail no servidor e desenvolver as funcionalidades de todas as rotas de acesso.

Período: 01/03/2023 à 31/03/2023.

Item do Product Backlog	Responsável	Período
22 - Configurar serviço de E-Mail no Servidor da Aplicação	Alex Silva	Dias 01 a 05
23 - Adicionar Rotas com Todos os Endpoints necessários para o funcionamento da aplicação e desenvolver suas funcionalidades	Alex Silva	Dias 06 a 31

Item: Sprint 10.

Descrição: Desenvolver no firmware funcionalidade de comunicação por bluetooth, comunicação com o servidor pela internet e sincronização das operações com o servidor.

Período: 01/04/2023 à 30/04/2023.

Item do Product Backlog	Responsável	Período
24 - Desenvolver no Firmware Funcionalidades da Comunicação Via Bluetooth	Alex Silva	Dias 01 a 15
25 - Desenvolver no Firmware Funcionalidades para Comunicação com o Servidor da Aplicação na Internet pelo Wi-Fi	Alex Silva	Dias 16 a 22
26 - Desenvolver no Firmware Funcionalidade de Sincronização de Dados com o Servidor da Aplicação	Alex Silva	Dias 23 a 30

Item: Sprint 11.

Descrição: Iniciar o desenvolvimento do core do aplicativo.

Período: 01/05/2023 à 31/05/2023.

Item do Product Backlog	Responsável	Período
27 - Adicionar Dependências do Aplicativo	Alex Silva	Dias 01 a 03
28 - Desenvolver Core do Aplicativo	Alex Silva	Dias 04 a 31

Item: Sprint 12.

Descrição: Desenvolver as interfaces do aplicativo.

Período: 01/06/2023 à 30/06/2023.

Item do Product Backlog	Responsável	Período
29 - Desenvolver as interfaces do aplicativo	Alex Silva	Dias 01 a 30

Item: Sprint 13.

Descrição: Desenvolver os controladores do aplicativo.

Período: 01/07/2023 à 31/07/2023.

Item do Product Backlog	Responsável	Período
30 - Desenvolver os controladores do aplicativo	Alex Silva	Dias 01 a 31

Item: Sprint 14.

Descrição: Desenvolver no aplicativo a funcionalidade de persistência de dados e comunicação com o broker MQTT.

Período: 01/08/2023 à 31/08/2023.

Item do Product Backlog	Responsável	Período
31 - Desenvolver no Aplicativo Recurso para Persistir os Dados	Alex Silva	Dias 01 a 12
32 - Desenvolver no Aplicativo Recurso para Comunicação com o broker MQTT	Alex Silva	Dias 13 a 31

Item: Sprint 15.

Descrição: Desenvolver no aplicativo a funcionalidade para comunicação por bluetooth e comunicação com o servidor na internet.

Período: 01/09/2023 à 30/09/2023.

Item do Product Backlog	Responsável	Período
33 - Desenvolver no Aplicativo Recurso para Comunicação Via Bluetooth	Alex Silva	Dias 01 a 18
34 - Desenvolver no Aplicativo Recurso para Comunicação com o Servidor da Aplicação	Alex Silva	Dias 19 a 30

1.8.2.3. Informações sobre as Sprints

Sprint	Reunião	Resumo da Reunião	Data da Reunião
1	Planejamento da Sprint	Foi tratado e definido sobre os documentos, diagramas e interfaces do aplicativo que precisam ser criados.	01/04/2022
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
2	Planejamento da Sprint	Foi tratado sobre detalhes da construção da placa de hardware.	01/07/2022
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
3	Planejamento da Sprint	Foi tratado sobre o desenvolvimento da funcionalidade para obter a leitura do sensor, para identificar o estado do portão.	01/08/2022
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
4	Planejamento da Sprint	Foi tratado sobre a funcionalidade de comunicação com a central do portão eletrônico, para realizar seu acionamento.	01/09/2022
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
5	Planejamento da Sprint	Foi tratado sobre a inclusão de um botão de reset na placa, para resetar o firmware para as configurações de fábrica.	01/10/2022
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
6	Planejamento da Sprint	Foi tratado sobre desenvolver a funcionalidade dos LEDs indicadores, para	01/11/2022

		sinalizar o funcionamento do hardware, para que possibilite acompanhar o que está ocorrendo.	
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
7	Planejamento da Sprint	Foi definido que deveria ser configurado o broker MQTT, para possibilitar o acionamento do portão remotamente. Também foi definido que a funcionalidade de comunicação com o broker MQTT deveria ser desenvolvida no firmware.	01/01/2023
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
8	Planejamento da Sprint	Foi tratado detalhes sobre o desenvolvimento do servidor da aplicação, e discutido sobre onde ficaria hospedado. Também foi definido os principais itens do core do servidor, que teriam prioridade para o desenvolvimento. Além disso, foi levantado pontos sobre a segurança, como utilização de tokens JWT, e a configuração que precisaria ser feita, para comunicação do servidor com o banco de dados.	01/02/2023
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
9	Planejamento da Sprint	Foi tratado sobre o convite de novos usuários que deveria ser enviado por e-mail. Também foi conversado sobre as rotas, funcionalidades que a API do servidor deveria ter.	01/03/2023
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
10	Planejamento da Sprint	Foi tratado sobre as novas funcionalidades do firmware, que precisariam ficar prontas. Precisaria de desenvolver a funcionalidade de comunicação por bluetooth, e também a comunicação com o servidor pela internet.	01/04/2023
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
11	Planejamento da Sprint	Foi tratado sobre o desenvolvimento do aplicativo que precisaria ser iniciado. Então, foi definido as dependências que seriam necessárias. Também foi falado sobre quais seriam as principais funcionalidade do aplicativo, que são essenciais e fazem parte de seu core.	01/05/2023

	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
12	Planejamento da Sprint	Foi tratado sobre as interfaces do aplicativo que precisariam ser desenvolvidas.	01/06/2023
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
13	Planejamento da Sprint	Foi tratado sobre as funcionalidades do aplicativo e sobre os controladores das interfaces.	01/07/2023
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
14	Planejamento da Sprint	Foi tratado sobre a sessão de quando um usuário faz login, e definido que a sessão deveria permanecer ativa, mesmo após o aplicativo ser fechado. Também foi discutido sobre com o portão seria acionado utilizando o broker MQTT.	01/08/2023
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias
15	Planejamento da Sprint	Foi tratado sobre a comunicação do aplicativo por bluetooth, e definido alguns comportamentos. Outro ponto discutido, foi sobre a comunicação do aplicativo com o servidor.	01/09/2023
	Daily Scrum	O que foi realizado, o que será feito e quais foram os impedimentos.	Todos os dias

2. DOCUMENTO DE REQUISITOS

Após a fase de levantamento de requisitos, foram identificados e coletados os seguintes requisitos para o software:

2.1. Conectar com servidor

Requisito Funcional		
Nome: Conectar com servidor		Código: F01
Descrição: O aplicativo deve ter uma interface para conectar com o servidor.		
Estimativa de Esforço: 8h		Prioridade: 100 pontos
Requisitos Não funcionais		
ID NF	Descrição	Categoria
1.1.	Será solicitado o endereço do servidor para que o aplicativo móvel possa se conectar.	Especificação
1.2.	A interface terá um campo para que o usuário digite o endereço do servidor e um botão para realizar a conexão.	Interface
1.3.	O endereço do servidor será salvo no dispositivo para que não seja necessário digitar toda vez que o aplicativo for aberto.	Usabilidade

2.2. Finalizar registro

Requisito Funcional		
Nome: Finalizar registro		Código: F02
Descrição: Deverá ter uma opção para que o usuário possa definir sua senha e informar seus dados.		
Estimativa de Esforço: 8h		Prioridade: 100 pontos
Requisitos Não funcionais		
ID NF	Descrição	Categoria
2.1.	O usuário deve finalizar seu registro utilizando o código de registro recebido via e-mail. Dessa forma, deverá copiar o código de registro do e-mail, abrir a tela de finalização de registro de usuário, ao abrir será apresentado um modal solicitando o código de registro com um botão para realizar a validação.	Especificação
2.2.	O usuário deverá informar o nome completo, digitar uma senha, confirmar a senha digitada, depois finalizar salvando os dados.	Especificação
2.3.	A tela terá um campo para o nome, um campo para a senha e um campo para a confirmação de senha. Logo abaixo terá um botão para finalizar e salvar os dados.	Interface
2.4.	Depois que o usuário clicar em finalizar, deve ser direcionado para a tela de login.	Usabilidade

2.3. Fazer login

Requisito Funcional		
Nome: Fazer login		Código: F03
Descrição: O aplicativo deve ter uma funcionalidade para permitir o login do usuário.		
Estimativa de Esforço: 8h		Prioridade: 100 pontos
Requisitos Não funcionais		
ID NF	Descrição	Categoria
3.1.	O usuário deve informar seu e-mail e senha definida para acessar o aplicativo.	Especificação
3.2.	A tela terá campos para que o usuário digite seu e-mail e senha. Logo abaixo dos campos terá um botão para que o usuário possa realizar o login.	Interface
3.3.	Caso o usuário informe a senha errada deve ser apresentado uma mensagem informando que o usuário e senha estão incorretos. Caso contrário, deve ser direcionado para a tela inicial do aplicativo.	Usabilidade
3.4.	Os dados de login serão persistidos para que não seja necessário o login toda vez que o aplicativo for aberto.	Usabilidade

2.4. Controlar portão

Requisito Funcional		
Nome: Controlar portão		Código: F04
Descrição: A aplicação deve permitir que os usuários controlem o portão.		
Estimativa de Esforço: 8h		Prioridade: 100 pontos
Requisitos Não funcionais		
ID NF	Descrição	Categoria
4.1.	Deve ser um recurso da tela principal do aplicativo e disponibilizará o acesso às outras telas a partir de botões e menus.	Especificação
4.2.	Quando a tela for aberta, deverá se conectar automaticamente no último portão controlado. Caso não for possível, deve apresentar no centro da tela um botão que permita que o usuário se conecte em um portão.	Especificação
4.3.	A aplicação deve permitir a realização das operações de abertura e fechamento do portão eletrônico deslizante.	Especificação
4.4.	Deverá ter um controle dinâmico deslizante, para que o portão possa ser aberto parcialmente, de acordo com o percentual indicado. Permitindo o portão variar gradualmente de completamente fechado à completamente aberto.	Especificação
4.5.	Se o usuário estiver distante do portão a mais de 100 metros, deve apresentar uma mensagem e pedir confirmação no primeiro acionamento, por questão e segurança.	Especificação
4.6.	A tela terá um botão deslizante para abrir o portão parcialmente, e um botão principal bem destacado para a ação comum de abrir e fechar o portão. A tela também	Interface

	mostrará qual o local do portão que está sendo controlado e também o seu nome na parte superior. Além disso, apresentará um menu na lateral esquerda para acesso às outras telas. No menu terá uma opção para escolher outro portão a ser controlado, chamada "Conectar com um portão".	
4.7.	Deve oferecer controles intuitivos e dinâmicos para as operações do portão.	Usabilidade
4.8.	Deve mostrar o tipo de conexão (Bluetooth ou Wi-Fi) no canto superior direito da tela.	Usabilidade

2.5. Visualizar perfil

Requisito Funcional		
Nome: Visualizar perfil		Código: F05
Descrição: O aplicativo deve ter uma funcionalidade para possibilitar que o usuário veja, edite e exclua seu perfil.		
Estimativa de Esforço: 8h		Prioridade: 100 pontos
Requisitos Não funcionais		
ID NF	Descrição	Categoria
5.1.	O usuário deverá ver seus dados de perfil e ter a possibilidade de editá-los e também excluir o perfil.	Especificação
5.2.	Caso o usuário altere seu e-mail e/ou privilégio deverá apresentar orientações e pedir confirmação em um modal.	Especificação
5.3.	A tela apresentará o e-mail, nome completo e privilégio do usuário. Abaixo terá um botão que permite salvar as modificações. No canto superior direito terá um botão de opções, com a opção excluir perfil.	Interface
5.4.	Quando não houver alterações, o botão Salvar ficará desabilitado. Porém, quando alguma alteração for realizada, o botão será habilitado.	Usabilidade
5.5.	Quando o usuário tocar em remover seu perfil, deverá pedir confirmação em um pop-up.	Usabilidade

2.6. Visualizar usuários

Requisito Funcional		
Nome: Visualizar usuários		Código: F06
Descrição: O aplicativo deve ter uma funcionalidade para permitir visualizar os usuários cadastrados.		
Estimativa de Esforço: 8h		Prioridade: 100 pontos
Requisitos Não funcionais		
ID NF	Descrição	Categoria
6.1.	Será um recurso disponível apenas para os usuários administradores.	Especificação
6.2.	Deverá apresentar uma lista dos usuários cadastrados, identificando seu nome, privilégio, status e ter um botão para realizar o controle de acesso daquele usuário.	Especificação

6.3.	A tela terá uma lista com as colunas nome, privilégio, status e ação. Na coluna ação deverá ter botões para controlar o acesso de determinado usuário.	Interface
6.4.	Ao clicar no botão de editar o acesso, deverá direcionar para a tela de controle de acesso daquele usuário específico.	Usabilidade

2.7. Convidar usuário

Requisito Funcional		
Nome: Convidar usuário		Código: F07
Descrição: O aplicativo deve ter uma funcionalidade para permitir convidar um novo usuário.		
Estimativa de Esforço: 8h		Prioridade: 100 pontos
Requisitos Não funcionais		
ID NF	Descrição	Categoria
7.1.	O convite só pode ser enviado por um usuário administrador.	Especificação
7.2.	As únicas informações necessárias para convidar um usuário, são: e-mail e privilégio. Assim que os dados do novo usuário forem preenchidos, e o botão Convidar for tocado, será enviado um e-mail para o novo usuário. No e-mail terá instruções para finalizar o registro, e um código de acesso.	Especificação
7.3.	A tela terá um campo para o e-mail do usuário, uma opção para escolher o privilégio (administrador ou padrão) e um botão para convidar o novo usuário.	Interface
7.4.	Depois que o convite é enviado, o usuário deve ser direcionado para a interface Visualizar Usuários.	Usabilidade

2.8. Controlar acesso

Requisito Funcional		
Nome: Controlar acesso		Código: F08
Descrição: O aplicativo deve ter uma funcionalidade para permitir o controle do acesso de um usuário.		
Estimativa de Esforço: 8h		Prioridade: 100 pontos
Requisitos Não funcionais		
ID NF	Descrição	Categoria
8.1.	Será um recurso disponível apenas para os usuários administradores.	Especificação
8.2.	Deverá permitir controlar o acesso de um usuário, ou seja, mudar privilégios, desativar acesso ou removê-lo.	Especificação
8.4.	Deve ter campos para mudar os privilégios e desativar o usuário. Terá botões para salvar e para remover o usuário.	Interface
8.5.	Para remover um usuário deverá pedir confirmação em um pop-up.	Usabilidade

2.9. Visualizar histórico

Requisito Funcional		
Nome: Visualizar histórico		Código: F09
Descrição: O aplicativo deve ter uma funcionalidade para visualizar o histórico de operações do portão.		
Estimativa de Esforço: 8h		Prioridade: 100 pontos
Requisitos Não funcionais		
ID NF	Descrição	Categoria
9.1.	Apresentará comportamento diferente dependendo do privilégio do usuário. Caso seja administrador, deverá permitir ver o histórico de todos os usuários. Caso seja comum, mostrará apenas histórico do próprio usuário.	Especificação
9.2.	Deve apresentar um histórico do uso do portão, identificando o usuário, operação (abertura, fechamento), data, hora, local e o nome do portão.	Especificação
9.3.	A tela terá uma lista com as colunas nome, operação, data e hora, local e nome do portão.	Interface
9.4.	Deverá ter um scroll para permitir rolar para baixo e ver todos os registros.	Usabilidade
9.5.	No caso de usuários administradores, deve ter opção para alternar entre o histórico do usuário administrador apenas ou o histórico de todos os usuários.	Usabilidade

2.10. Configurar portão

Requisito Funcional		
Nome: Configurar portão		Código: F10
Descrição: O aplicativo deve oferecer funcionalidade para visualizar configuração e configurar o portão.		
Estimativa de Esforço: 8h		Prioridade: 100 pontos
Requisitos Não funcionais		
ID NF	Descrição	Categoria
10.1	Será uma funcionalidade disponível apenas para os usuários administradores.	Especificação
10.2.	Deve permitir ver e editar os dados de configuração do portão, mesmo que já esteja configurado e funcional.	Especificação
10.3.	As configurações a serem feitas serão: conexão, localização, identificação e calibração do portão.	Especificação
10.4.	Deve permitir a configuração do portão no servidor para que funcione remotamente.	Especificação
10.5.	A primeira configuração será por bluetooth, depois disso poderá ser realizada remotamente também, por meio da internet.	Especificação
10.6.	A tela de configuração do portão será dividida por abas, cada aba será responsável por apresentar e salvar cada tipo de configuração.	Interface

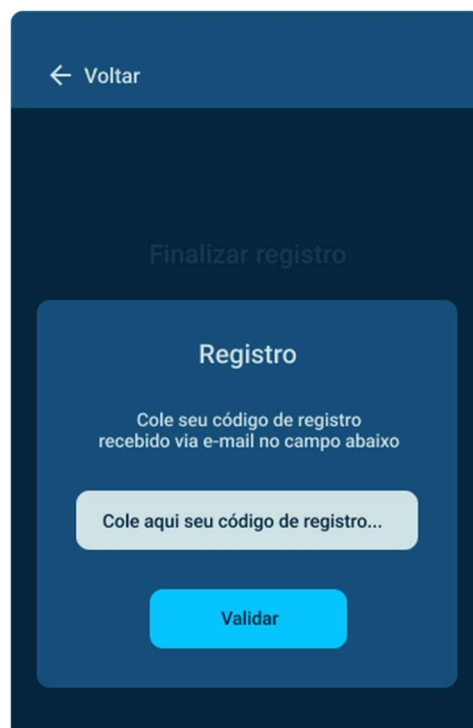
10.7.	A aba de Conexão, será para configurar o portão na internet e no servidor. Terá um campo para que o usuário informe o Endereço do Servidor, um campo para informar o SSID da rede Wi-Fi e um campo para que seja informado a Senha da rede.	Interface
10.8.	A aba de Localização, será para configurar a localização do portão, usado por questão de segurança, para identificar a distância do usuário até o portão. Terá dois campos, um para a Latitude e outro para a Longitude do portão. Também terá um botão, para que o usuário possa tocar e a localização ser preenchida automaticamente, de usando a localização do dispositivo do usuário.	Interface
10.9.	A aba de Identificação, será para identificar o portão na aplicação. Terá um campo para informar a Localidade e outro para informar o nome do portão.	Interface
10.10.	A aba calibração, será para identificar o tempo que o portão leva para abrir e fechar, e também para mapear o estado o sensor quando o portão está aberto, parcialmente aberto ou fechado.	Interface
10.11.	Abaixo de cada aba de configuração, deve ter um botão que permita salvar os dados informados.	Interface
10.12.	Deverá ter um botão para scanear redes locais disponíveis para ficar mais prático a escolha do SSID.	Usabilidade

3. PROJETO DE INTERFACE


3.1. Conectar com servidor



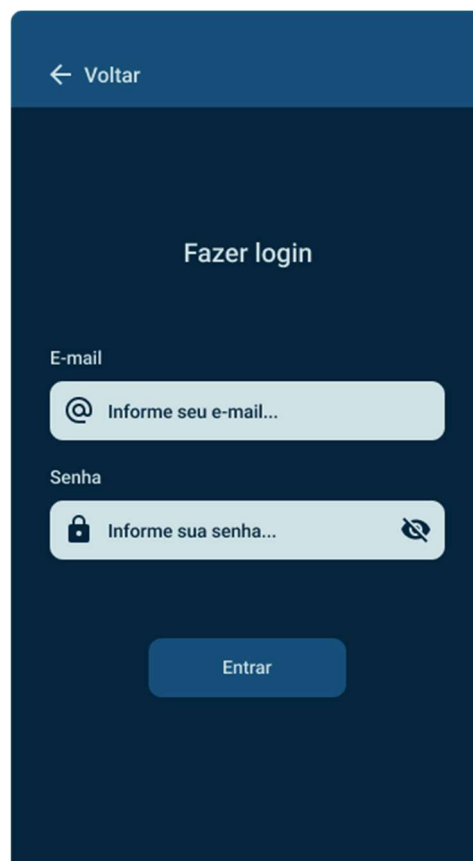
3.2.1. Validar registro de usuário



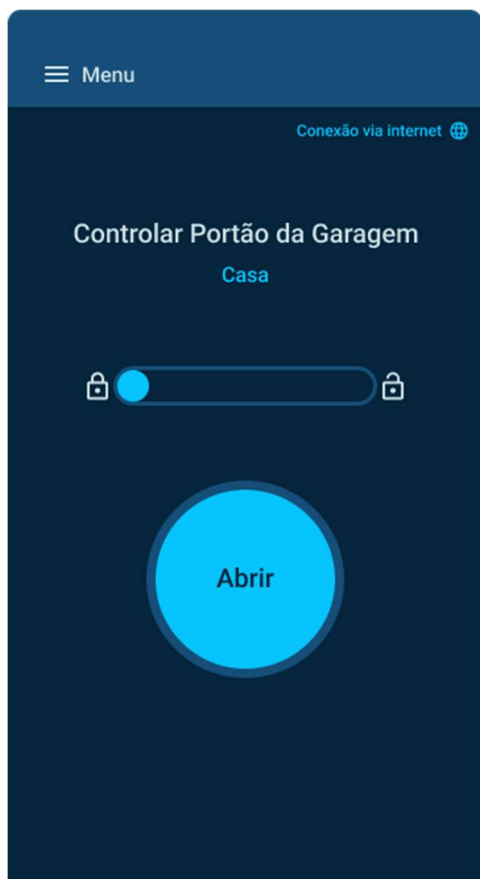
3.2. Finalizar registro



3.3. Fazer login



3.4. Controlar portão



3.4.2. Controlar portão por bluetooth



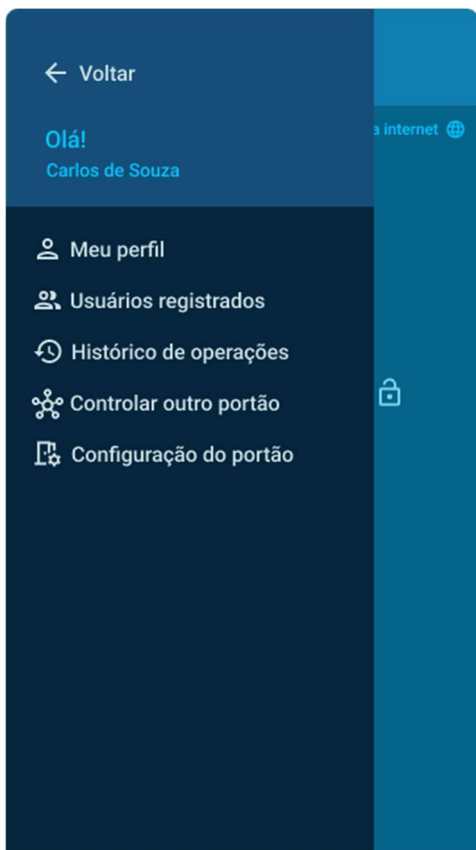
3.4.1. Controlar outro portão



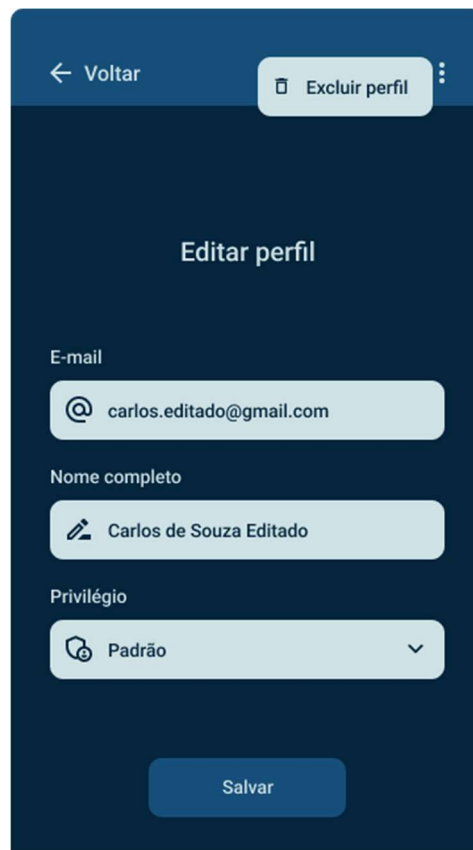
3.4.3. Selecionar portão bluetooth



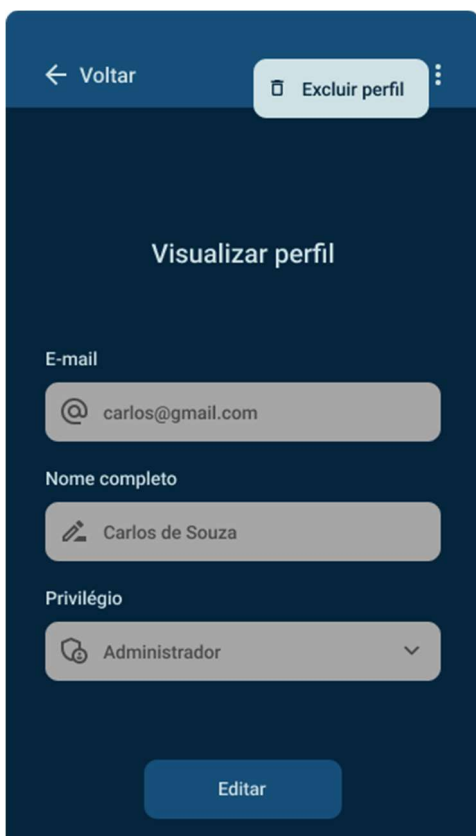
3.4.4. Menu principal



3.5.1. Editar perfil



3.5. Visualizar perfil



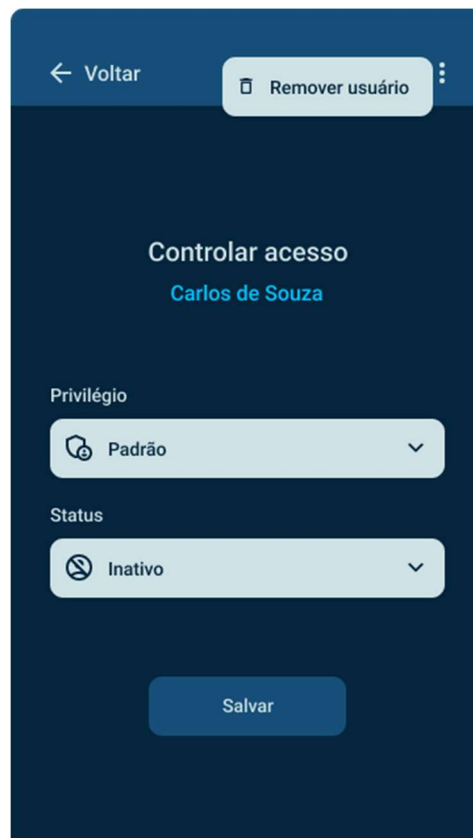
3.5.2. Confirmar edição de perfil



3.6. Visualizar usuários



3.8. Controlar acesso



3.7. Convidar usuário



3.9. Visualizar histórico



3.10. Configurar portão: Conexão

← Voltar

Configurações do portão

Conexão Localização Identificação Calibração

Endereço do servidor

https://endereco-servidor.com.br

SSID da rede Wi-Fi

Wi-Fi da minha casa 2.4G

Senha da rede Wi-Fi

●●●●●●●●

Salvar

3.10.2. Configurar portão: Localização

← Voltar

Configurações do portão

Localização Identificação Calibração

Latitude

-10.46544862

Longitude

-60.56854521

Usar minha localização

Salvar

3.10.1. Selecionar rede Wi-Fi

← Voltar

Configurações do portão

Selecionar rede Wi-Fi

Selecione a rede Wi-Fi na lista

Wi-Fi da minha casa 2.4G

Wi-Fi do vizinho

Wi-Fi ABC

●●●●●●●●

3.10.3. Configurar portão: Identificação

← Voltar

Configurações do portão

Identificação Calibração

Nome do portão

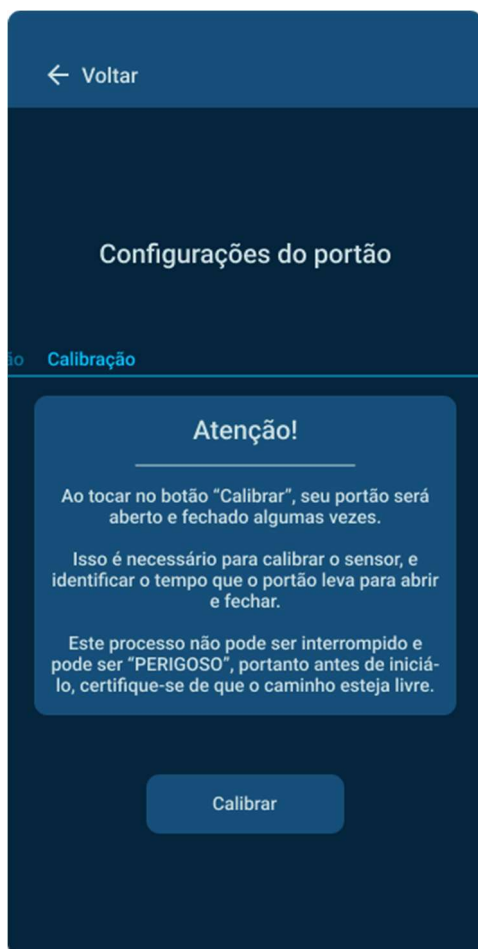
Portão da garagem

Local do portão

Casa

Salvar

3.10.4. Configurar portão: Calibração

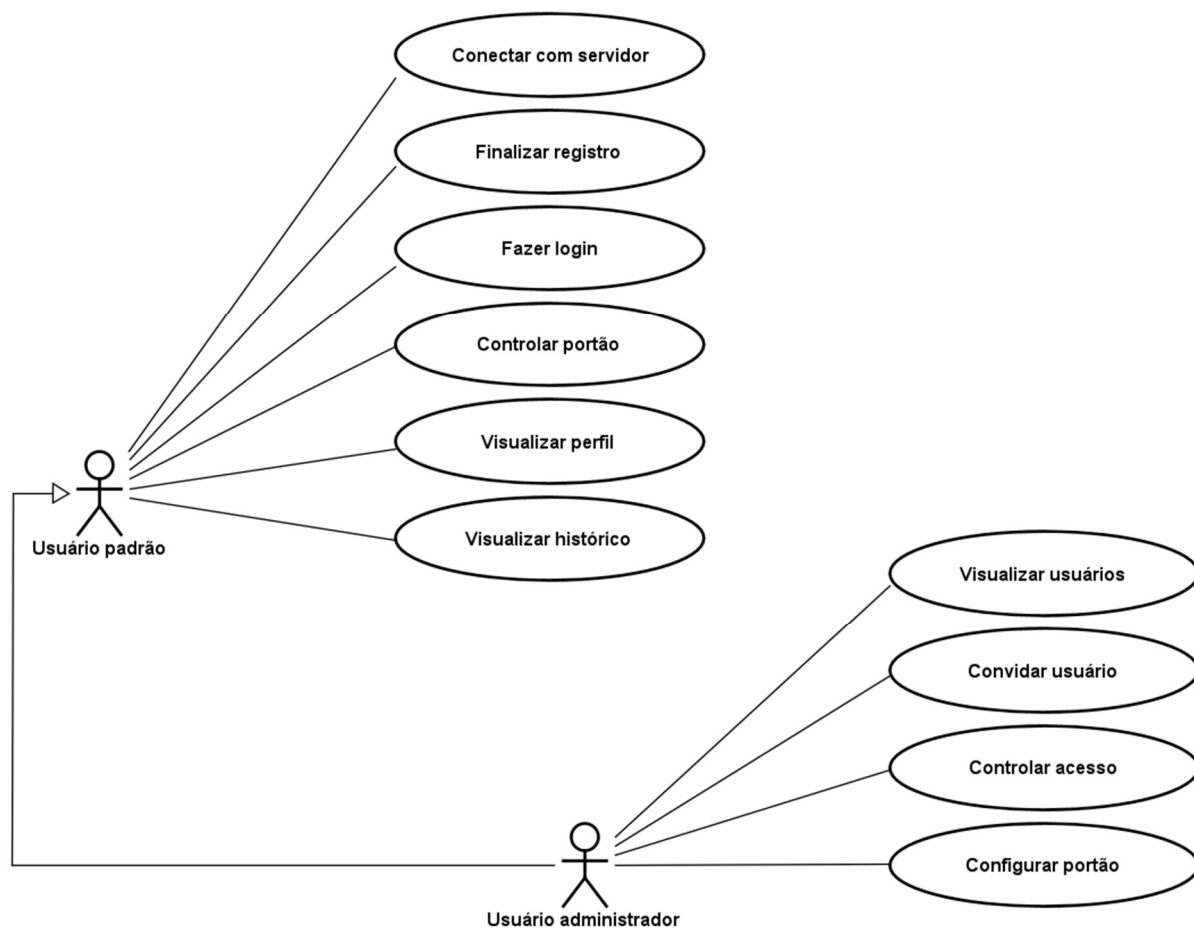


3.10.5. Confirmação de estado



4. CASO DE USO

4.1. Diagrama de Caso de Uso



4.2. Caso de Uso Expandido

4.2.1. Conectar com servidor

Caso de Uso: Conectar com servidor. (CSU01)

Descrição: Realiza a conexão do aplicativo com o servidor da aplicação hospedado na internet.

Ator Primário: Usuário padrão.

Ator Secundário: Usuário administrador.

Pré-condições: O usuário precisa ter acesso à internet e o endereço de acesso ao servidor.

Fluxo Principal:

1. A aplicação apresenta a interface Conectar com Servidor.
2. O usuário informa o endereço de acesso ao servidor no campo Endereço do Servidor.
3. O usuário toca no botão Conectar para se conectar ao servidor.
4. A aplicação valida se o campo Endereço do Servidor está preenchido e é válido.
5. A aplicação tenta se conectar ao servidor no endereço informado.
6. Caso ocorra algum erro para se conectar no endereço informado, uma mensagem de erro será apresentada ao usuário e a execução é retornada para o passo 2 do Fluxo principal.
7. Caso a conexão ocorra com sucesso, o endereço de acesso ao servidor será salvo no dispositivo para que não seja necessário ficar informando novamente nas próximas vezes que o aplicativo for aberto.

4.2.2. Finalizar registro

Caso de Uso: Finalizar registro. (CSU02)

Descrição: Finaliza o registro de um usuário na aplicação.

Ator Primário: Usuário padrão.

Ator Secundário: Usuário administrador.

Pré-condições: É necessário que o usuário tenha o código de registro que é enviado pelo e-mail.

Fluxo Principal:

1. O usuário acessa o convite recebido via e-mail, copia o código de registro, e segue as instruções.
2. O usuário precisa acessar a interface Finalizar Registro de Usuário manualmente, com isso, a interface Validar Registro de Usuário será apresentada para que seja colado o código de registro.
3. O usuário cola o código recebido no campo Código de Registro, e toca no botão Validar.
4. Caso ocorra falha na validação, uma mensagem de erro é apresentada ao usuário, e a execução é retornada ao início do Fluxo principal.
5. Caso a validação ocorra com sucesso, dados de e-mail e privilégio são pré-carregados implicitamente a partir da validação do código de registro.
6. O usuário preenche os campos Nome Completo, Senha e Confirmação de Senha.
7. O usuário toca no botão Finalizar.
8. A aplicação tenta finalizar o registro.
9. Caso ocorra falha na finalização do registro do usuário, uma mensagem de erro é apresenta ao usuário e a execução volta para o passo 7 do Fluxo Principal.
10. Caso o registro seja finalizado com sucesso, o usuário deve ser direcionado para a interface Fazer Login.

4.2.3. Fazer login

Caso de Uso: Fazer login. (CSU03)

Descrição: Realiza o login do usuário.

Ator Primário: Usuário padrão.

Ator Secundário: Usuário administrador.

Pré-condições: O usuário já precisa ter finalizado o seu registro.

Fluxo Principal:

1. A aplicação apresenta a interface Fazer Login.
2. O usuário preenche os campos E-mail e Senha com seus dados de autenticação.
3. O usuário toca no botão Entrar.

4. A aplicação tenta autenticar o usuário.
5. Caso ocorra falha na autenticação, uma mensagem de erro é apresentada ao usuário e a execução volta para o passo 2 do Fluxo Principal.
6. Caso a autenticação seja bem sucedida, o usuário é direcionado para a interface principal Controlar Portão.

4.2.4. Controlar portão

Caso de Uso: Controlar Portão. (CSU04)

Descrição: Permite que o usuário faça o controle do portão.

Ator Primário: Usuário padrão.

Ator Secundário: Usuário administrador.

Pré-condições: O usuário precisa estar autenticado.

Fluxo Principal:

1. No momento de abertura da tela, é checado se na última vez que o aplicativo foi fechado algum portão estava conectado, se sim, o aplicativo tenta se conectar automaticamente no mesmo portão.
2. Caso o aplicativo não se conecte automaticamente no portão, a execução continua no Fluxo Alternativo 1, caso contrário no Fluxo Alternativo 2.

Fluxo Alternativo 1: Conectar com um portão.

1. É apresentado um modal, para que o usuário conecte o aplicativo com um portão.
2. Se a conexão for por internet, o usuário precisa escolher a localidade do portão e depois o portão. Porém se a conexão for por bluetooth, basta o usuário escolher o portão na lista de dispositivos bluetooth.
3. Se a conexão ocorrer com sucesso, a execução continua no Fluxo Alternativo 2, caso contrário, uma mensagem de erro é apresentada ao usuário.

Fluxo Alternativo 2: Controlar o portão.

1. A aplicação apresenta a interface Controlar Portão, com o status atual do portão.
2. É mostrado o tipo de conexão que está sendo usada, se é Wi-Fi ou bluetooth.

3. A aplicação mostra o nome do portão que está sendo controlado e também o seu local.
4. O usuário pode alternar entre Abrir ou Fechar o portão, tocando no botão destacado no centro da tela.
5. Pode-se também, utilizar o slider de controle dinâmico, para abrir o portão parcialmente, e assim ter um controle mais preciso sobre até onde o portão será aberto.
6. Se o usuário estiver a mais de 100 metros de distância do portão, é apresentado uma mensagem, informando e pedindo confirmação do usuário para prosseguir com o acionamento.
7. Para controlar outro portão, o usuário abre o menu lateral e toca na opção Controlar Outro Portão, assim a execução continua no Fluxo Alternativo 1.

4.2.5. Visualizar perfil

Caso de Uso: Visualizar perfil. (CSU05)

Descrição: Para que o usuário possa ver seu perfil, editar e também excluir.

Ator Primário: Usuário padrão.

Ator Secundário: Usuário administrador.

Pré-condições: O usuário precisa estar autenticado e o privilégio só pode ser alterado de administrador para padrão.

Fluxo Principal:

1. A aplicação apresenta a interface Visualizar Perfil.
2. O usuário pode visualizar os dados do seu perfil, como E-mail, Nome Completo e Privilégio.
3. Caso o usuário queira editar dados de seu perfil, basta o usuário tocar no campo do dado que deseja modificar, e a execução continua no Fluxo Alternativo 1.
4. Caso o usuário queira excluir seu perfil, o usuário toca no ícone de opções que fica na parte superior direita da interface, depois toca na opção Excluir Perfil, e a execução continua no Fluxo Alternativo 2.

Fluxo Alternativo 1: Editar perfil.

1. O usuário edita seus dados de perfil.

2. O usuário toca no botão Salvar.
3. Caso as informações alteradas seja o e-mail e/ou privilégio, será apresentado a interface Confirmação de Edição de Perfil, passando orientações sobre o impacto das alterações e pedindo confirmação.
4. A aplicação tenta salvar as alterações no servidor.
5. Caso ocorra algum erro ao salvar, é apresentado uma mensagem de erro e a execução volta para o passo 1 do Fluxo Alternativo 1.
6. Caso a edição de perfil seja salva com sucesso, a execução retorna para o início do Fluxo Principal.

Fluxo Alternativo 2: Excluir perfil.

1. Uma mensagem solicitando confirmação da exclusão do perfil é apresentada ao usuário.
2. Caso o usuário não confirme, a execução volta ao início do Fluxo Principal.
3. Caso o usuário tenha confirmado, aplicação tenta excluir o perfil do usuário.
4. Caso ocorra algum erro ao excluir o perfil, é apresentado uma mensagem de erro e a execução volta para o início do Fluxo Principal.
5. Caso a exclusão ocorra com sucesso a aplicação fará o logout.

4.2.6. Visualizar usuários

Caso de Uso: Visualizar usuários. (CSU06)

Descrição: Visualizar os usuários registrados na aplicação.

Ator Primário: Usuário administrador.

Pré-condições: É preciso que o usuário tenha privilégio de administrador.

Fluxo Principal:

1. A aplicação apresenta a interface Visualizar Usuários.
2. O usuário administrador pode ver dados como Nome, Privilégio e Status dos usuários registrados na aplicação.
3. Caso o usuário administrador queira controlar o acesso de algum usuário, é preciso tocar no botão com o ícone de um lápis na linha correspondente ao usuário e com isso será apresentado a interface Controlar Acesso.
4. Caso seja preciso adicionar um novo usuário basta tocar no botão flutuante no canto inferior direito que será apresentado a interface Convidar Usuário.

4.2.7. Convidar usuário

Caso de Uso: Convidar usuário. (CSU07)

Descrição: Convida os usuários da aplicação.

Ator Primário: Usuário administrador.

Pré-condições: É preciso que o usuário tenha privilégio de administrador para que seja possível registrar novos usuários.

Fluxo Principal:

1. A aplicação apresenta a interface Convidar Usuário.
2. O usuário deve informar o e-mail e o privilégio do usuário a ser convidado.
3. O usuário toca no botão Convidar para enviar um convite ao novo usuário.
4. A aplicação tenta convidar o novo usuário.
5. Caso ocorra algum erro, uma mensagem de erro será apresentada e a execução é retornada ao início do Fluxo Principal.
6. Caso o convite seja realizado com sucesso, um e-mail é enviado para o novo usuário com informações para finalizar o registro.
7. A aplicação apresenta uma mensagem de sucesso, e volta para a interface Visualizar Usuários.

4.2.8. Controlar acesso

Caso de Uso: Controlar acesso. (CSU08)

Descrição: Controla o acesso de um usuário.

Ator Primário: Usuário administrador.

Pré-condições: É preciso que o usuário tenha privilégio de administrador.

Fluxo Principal:

1. A aplicação apresenta a interface Controlar Acesso.
2. O usuário administrador muda o privilégio de acesso e status de um usuário.
3. O usuário toca no botão Salvar.
4. A aplicação tenta salvar as alterações.
5. Caso ocorra falha e não seja possível salvar, uma mensagem de erro é apresentada e a execução volta ao passo 2 do Fluxo Principal.
6. Caso salve com sucesso, o usuário é informado que tudo ocorreu bem.

Fluxo Alternativo 1: Remover usuário.

1. O usuário toca no ícone de opções no canto superior direito da interface.
2. Nas opções que aparece o usuário toca na opção Remover Usuário.
3. A aplicação deve pedir confirmação da operação ao usuário administrador.
4. Caso o usuário administrador confirme, o usuário é removido.

4.2.9. Visualizar histórico

Caso de Uso: Visualizar histórico. (CSU09)

Descrição: Apresenta o histórico das operações dos usuários.

Ator Primário: Usuário padrão.

Ator Secundário: Usuário administrador.

Pré-condições: O usuário precisa estar autenticado.

Fluxo Principal:

1. A aplicação verifica o privilégio do usuário.
2. Caso seja um usuário administrador, segue o Fluxo Alternativo 1.
3. Caso seja um usuário padrão, segue o Fluxo Alternativo 2.

Fluxo Alternativo 1: Apresentar o histórico de todos os usuários.

1. A aplicação apresenta a interface Visualizar Histórico com o histórico de todos os usuários.
2. Caso o usuário queira filtrar para aparecer apenas seu próprio histórico, basta tocar na opção Apenas o Meu do filtro e vice-versa.

Fluxo Alternativo 2: Apresentar somente o histórico do próprio usuário.

1. A aplicação apresenta a interface Visualizar Histórico com o histórico do próprio usuário.

4.2.10. Configurar portão

Caso de Uso: Visualizar e editar configuração do portão. (CSU10)

Descrição: Apresenta as configurações atuais do portão e permite que o usuário as edite.

Ator Primário: Usuário administrador.

Pré-condições: O aplicativo precisa estar conectado com um portão, e o usuário precisa ter privilégio de administrador.

Fluxo Principal:

1. A tela de configuração do portão já é aberta na aba de configuração de conexão, se a conexão já estava configurada, os dados configurados são trazidos do hardware do portão e apresentados ao usuário.
2. O usuário pode alternar entre o tipo de configuração que deseja visualizar e/ou editar, basta tocar na aba de configuração correspondente ao seu desejo, então a execução continua nos Fluxos Alternativos entre 1 e 4, de acordo com o tipo de configuração selecionada.

Fluxo Alternativo 1: Configuração de conexão do portão.

1. O aplicativo apresenta a interface de configuração da conexão do portão.
2. O usuário informa o endereço do servidor.
3. O usuário informa o SSID da rede Wi-Fi, ou então toca no botão localizado no lado direito do campo, para scanear as redes existentes e apresentar para que seja escolhido a rede na Interface Selecionar rede Wi-Fi.
4. O usuário informa a senha da rede Wi-Fi.
5. O usuário toca no botão Salvar.
6. O aplicativo valida os dados informados e envia ao hardware do portão.
7. O hardware do portão tenta se conectar na rede Wi-Fi, e depois tenta se conectar ao servidor da aplicação na internet.
8. Caso ocorra algum erro, é apresentado uma mensagem informando o erro, e a execução volta ao início do Fluxo Alternativo 1.
9. Caso a conexão seja bem estabelecida, as novas configurações são salvas.

Fluxo Alternativo 2: Configuração de localização do portão.

1. O aplicativo apresenta a interface de configuração da localização do portão.
2. O usuário informa a Latitude e Longitude da localização do portão, ou então toca no botão abaixo dos campos, chamado Usar minha Localização.
3. O usuário toca no botão Salvar.
4. O aplicativo valida os dados informados pelo usuário e envia ao hardware do portão.
5. O hardware grava os dados em memória.
6. Caso ocorra algum erro, é apresentado uma mensagem informando o erro, e a execução volta ao início do Fluxo Alternativo 2.
7. Caso a configuração ocorra com sucesso, as novas configurações são salvas.

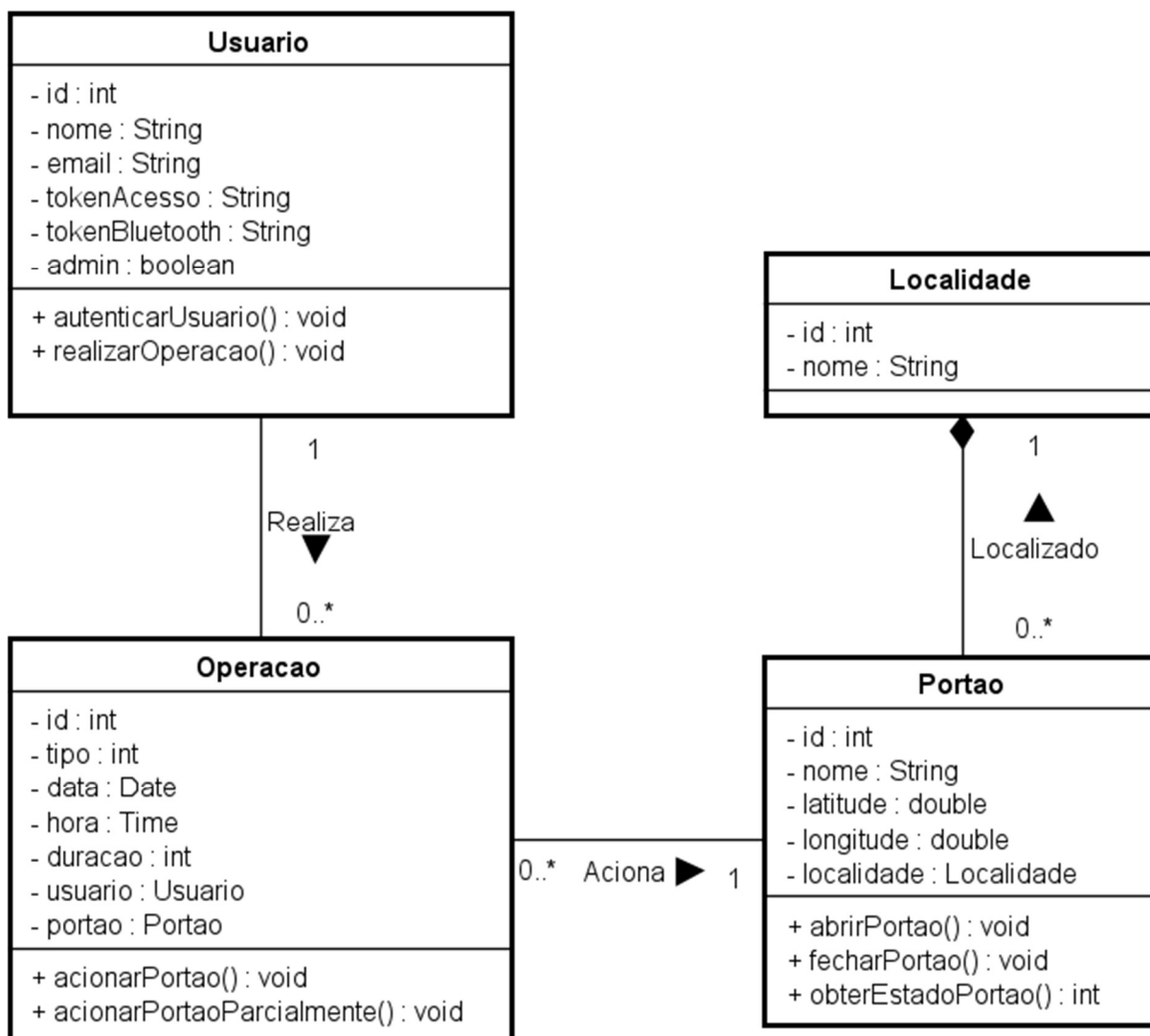
Fluxo Alternativo 3: Configuração de identificação do portão.

1. O aplicativo apresenta a interface de configuração da identificação do portão.
2. O usuário informa o nome do portão e também define sua localidade.
3. O usuário toca no botão Salvar.
4. O aplicativo valida os dados informados pelo usuário e envia ao hardware do portão.
5. O hardware realiza as configurações internas necessárias e grava em memória.
6. Caso ocorra algum erro, é apresentado uma mensagem informando o erro, e a execução volta ao início do Fluxo Alternativo 3.
7. Caso a configuração ocorra com sucesso, as novas configurações são salvas.

Fluxo Alternativo 4: Configuração de calibração do portão.

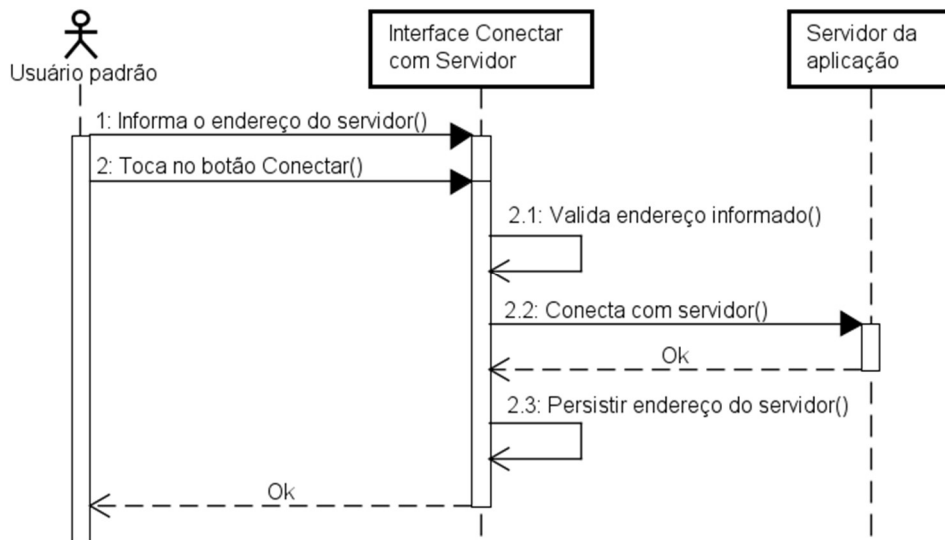
1. O aplicativo apresenta a interface de configuração da calibração do portão.
2. O usuário toca no botão Calibrar.
3. O portão realiza o processo de calibragem do sensor e outros parâmetros.
4. O usuário informa se o portão parou aberto ou fechado.
5. Se a calibração falhar, uma mensagem de erro é apresentada, e a execução volta ao início do Fluxo Alternativo 4.
6. Caso a calibração ocorra com sucesso, as novas configurações são salvas e uma mensagem de sucesso é apresentada ao usuário.

5. DIAGRAMA DE CLASSES

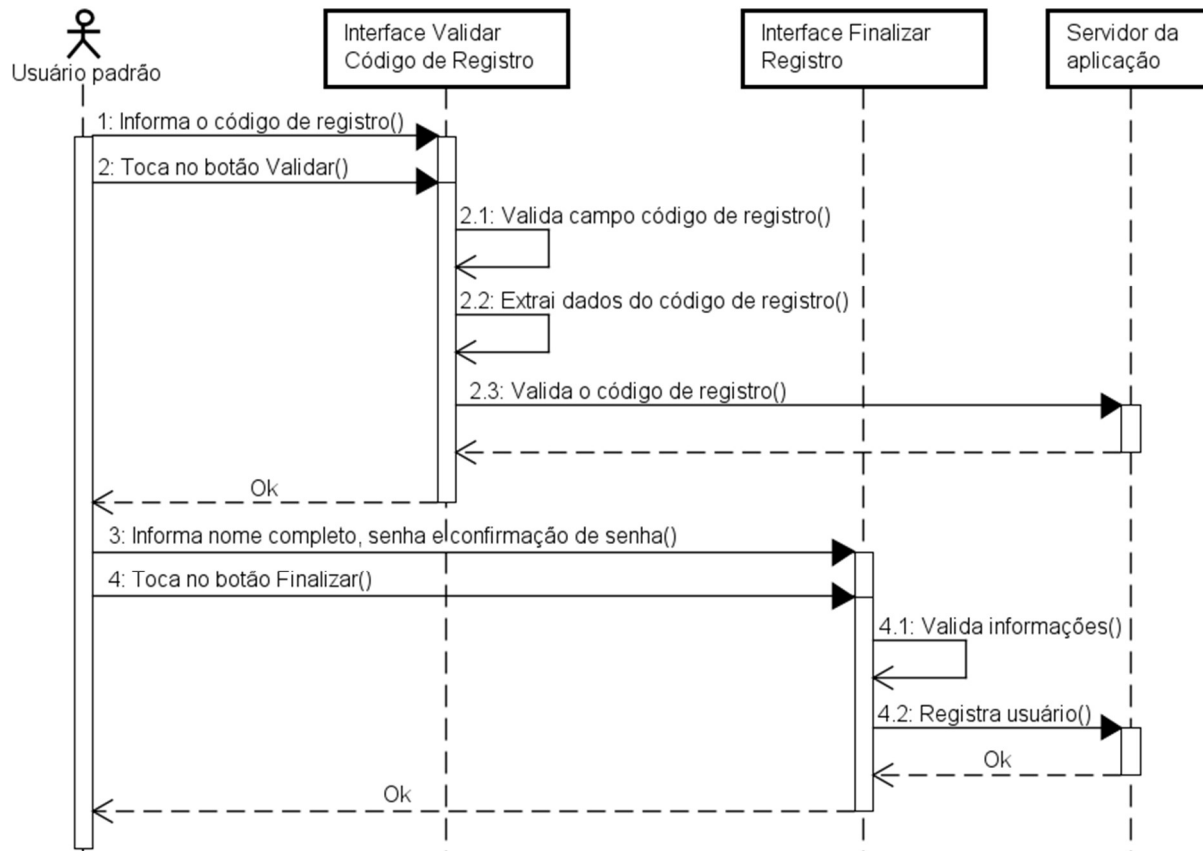


6. DIAGRAMAS DE SEQUÊNCIA

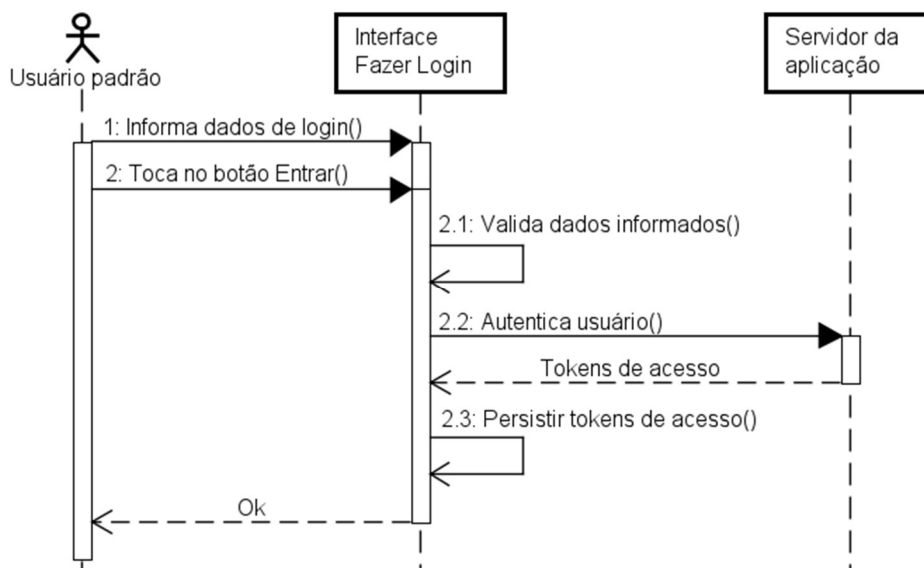
6.1. Conectar com servidor



6.2. Finalizar registro

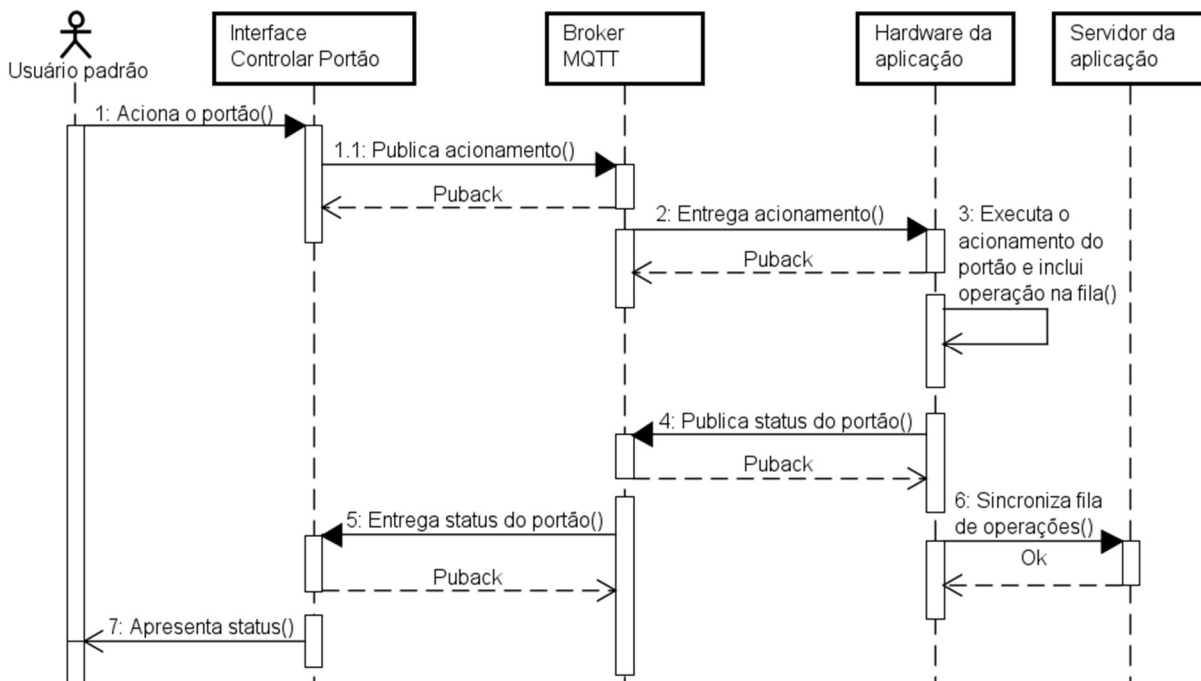


6.3. Fazer login

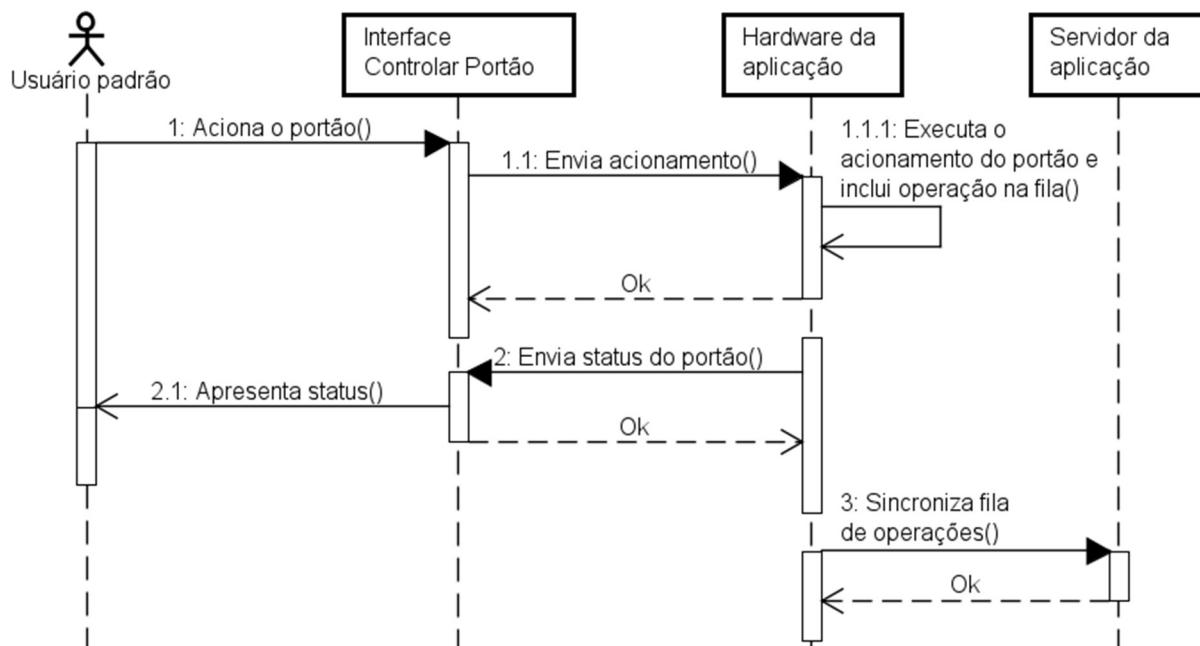


6.4. Controlar portão

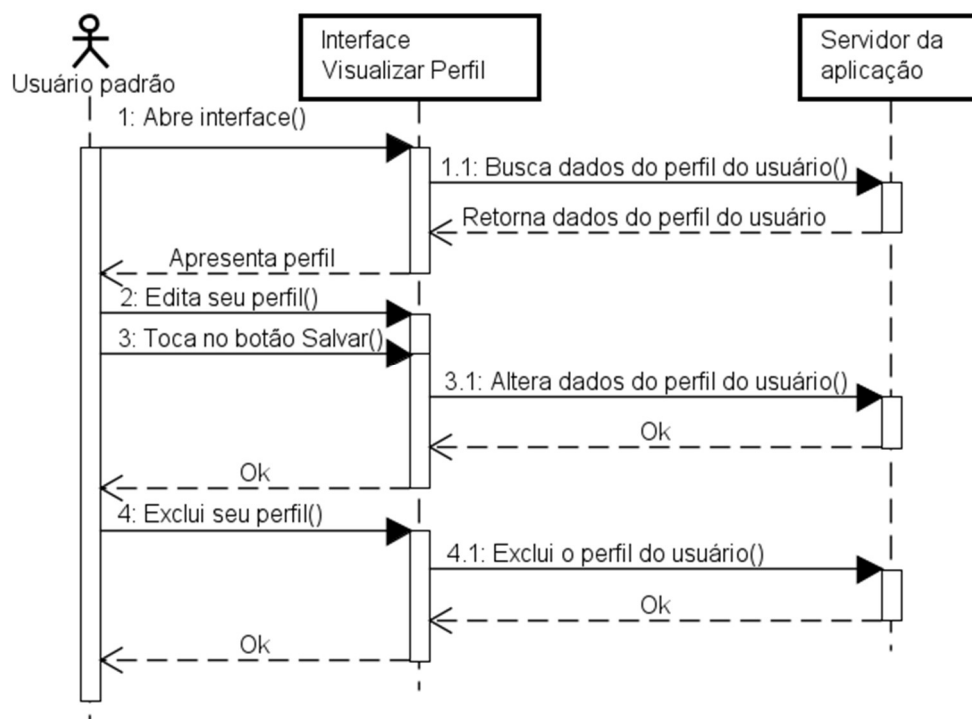
6.4.1. Controlar portão pela internet



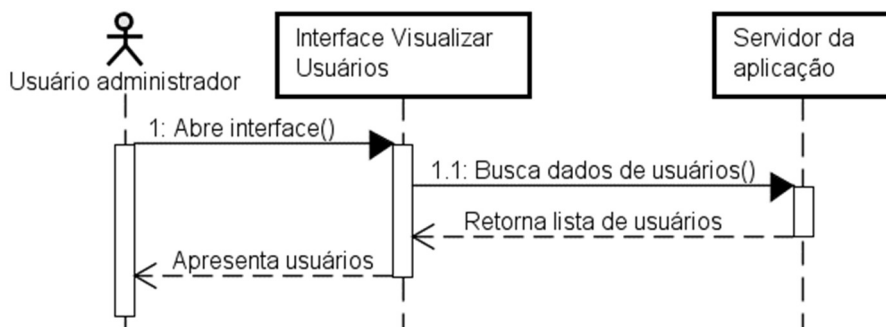
6.4.2. Controlar portão por bluetooth



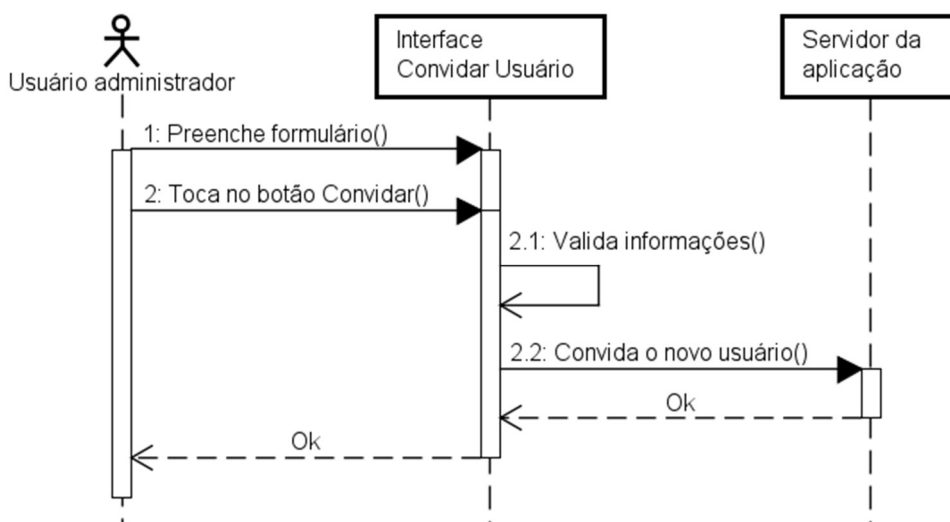
6.5. Visualizar perfil



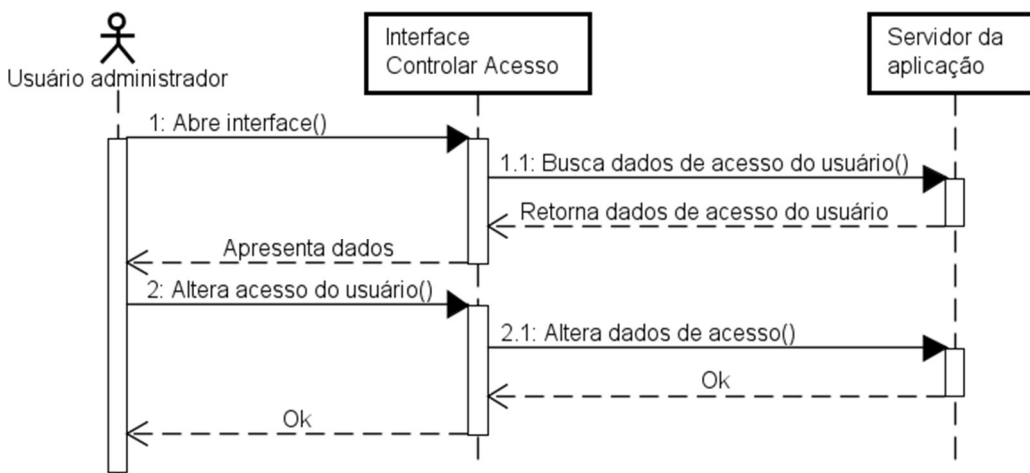
6.6. Visualizar usuários



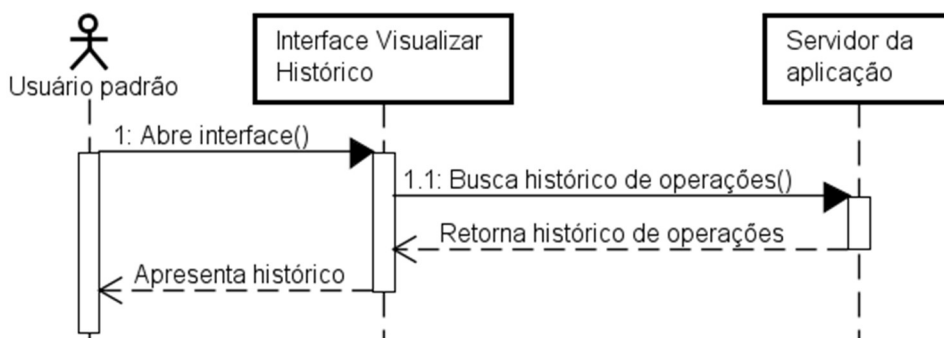
6.7. Convidar Usuário



6.8. Controlar acesso

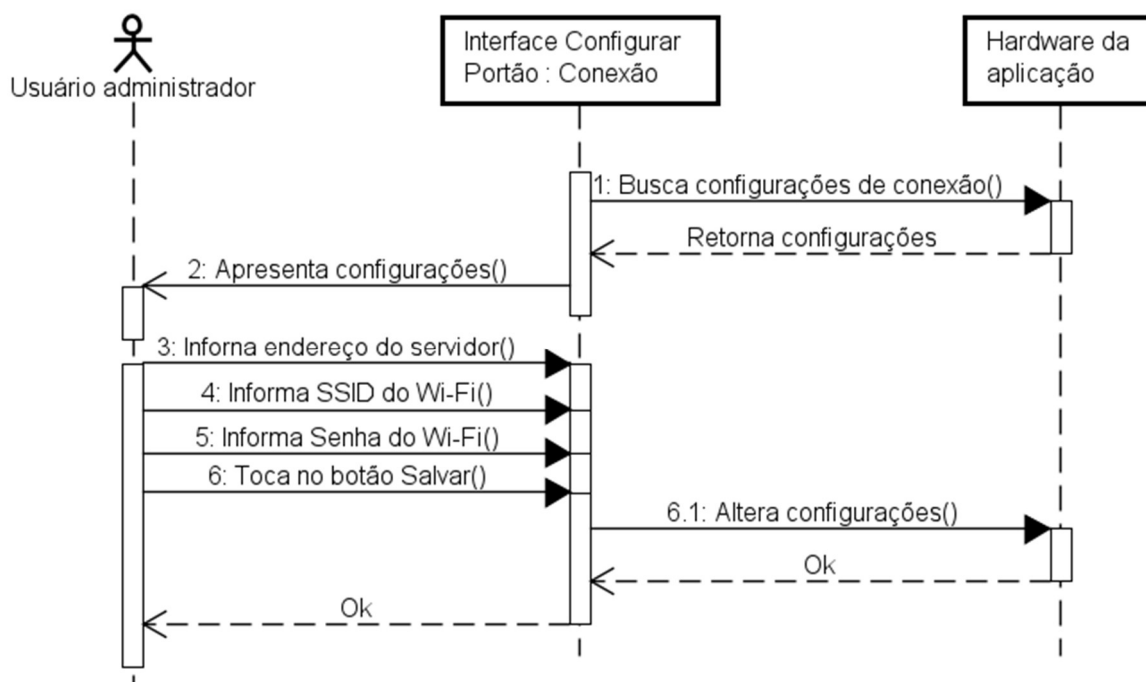


6.9. Visualizar histórico

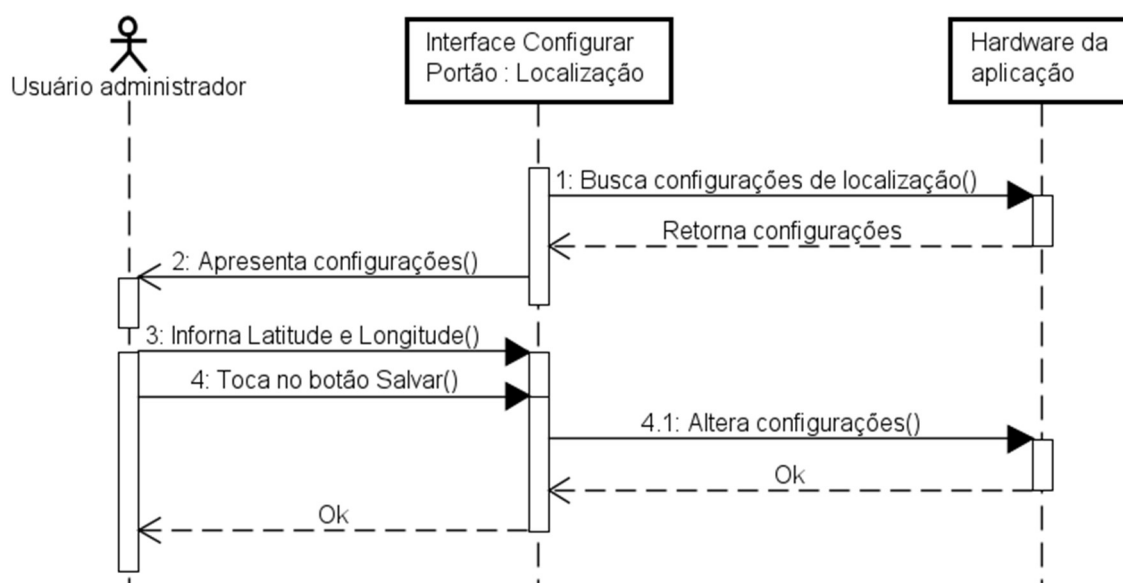


6.10. Configurar portão

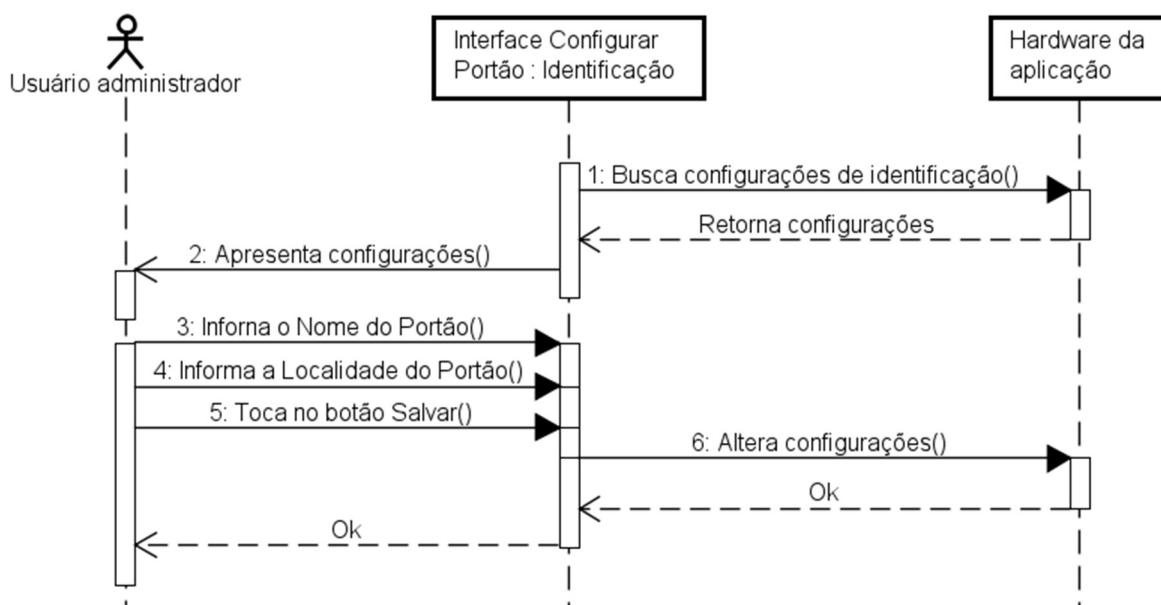
6.10.1. Configurar conexão do portão



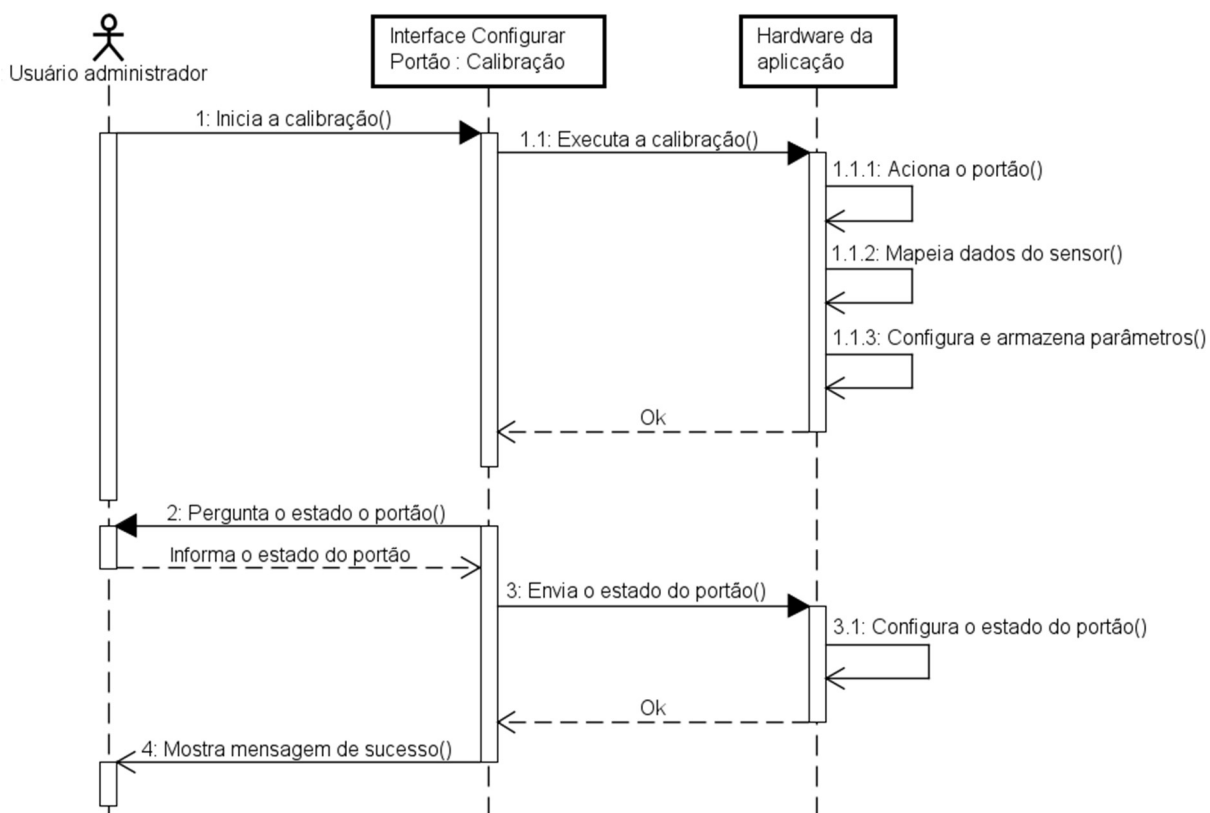
6.10.2. Configurar localização do portão



6.10.3. Configurar identificação do portão



6.10.4. Configurar calibração do portão



7. DIAGRAMA DE BANCO DE DADOS

