

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO
GRANDE DO NORTE CAMPUS CURRAIS NOVOS**

EMANUEL RANGEL DANTAS

**SLOCKED:SISTEMA PARA GERENCIAMENTO DE TRANCAS ELETRÔNICAS
VIA DESKTOP E MOBILE**

**CURRAIS NOVOS/RN
2025**

EMANUEL RANGEL DANTAS

**SLOCKED:SISTEMA PARA GERENCIAMENTO DE TRANCAS ELETRÔNICAS
VIA DESKTOP E MOBILE**

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

Orientador: M.e Ricardo Kléber Martins Galvão

Coorientador: Esp. Jorge Chrystiann Guimarães da Cunha Nunes

CURRAIS NOVOS/RN
2025

Dantas, Emanuel Rangel.

D192s Slocked: sistema para gerenciamento de trancas eletrônicas via desktop e mobile. / Emanuel Rangel Dantas. 2025.

46f : il. color.

Trabalho de Conclusão de Curso (Tecnologia em Sistema para Internet) – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte. Currais Novos, 2024.

Orientador: M.e Ricardo Kléber Martins Galvão.

1. Controle de acesso - Tranca. 2. Sistema de segurança. 3. Sistema computacional. I. Galvão, Ricardo Kléber Martins. II. Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte. III. Título.

SIBi/IFRN

CDU 004.056

Sistema Integrado de Bibliotecas
Elaborada pela Bibliotecária
Kátia Cristina Dantas da Câmara Silva – CRB-15/496

EMANUEL RANGEL DANTAS

**SLOCKED:SISTEMA PARA GERENCIAMENTO DE TRANCAS ELETRÔNICAS
VIA DESKTOP E MOBILE**

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

Trabalho de Conclusão de Curso apresentado e aprovado em ___/___/___
pela seguinte Banca Examinadora:

Ricardo Kléber Martins Galvão, M.e - Orientador
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Jorge Chrystiann Guimarães da Cunha Nunes, Esp. - Examinador
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Moroni Neres Vieira, M.e – Coordenador do Curso
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

RESUMO

Com o crescimento e avanço das tecnologias, diversas atividades e tarefas que antes exigiam trabalho manual foram substituídas e adaptadas com o auxílio de aparelhos eletrônicos e sistemas tecnológicos, os quais são capazes de entregar resultados mais rapidamente e com menor custo. Uma das áreas mais demandadas por essas tecnologias é a segurança, especialmente no que se refere aos serviços de controle de acesso. Entre os exemplos mais comuns, destaca-se o controle de acesso a salas de instituições e edifícios, realizado por meio de trancas eletrônicas implementadas nas portas. Este trabalho tem como objetivo apresentar um sistema computacional para o controle de uma tranca eletrônica, utilizando uma aplicação web, móvel e uma solução IoT, oferecendo uma alternativa mais prática, econômica e segura para otimizar o gerenciamento e controle de acesso, visando atender às necessidades dos usuários de forma eficiente e acessível.

Palavras-chave: Controle de acesso; Trancas; Sistema computacional.

ABSTRACT

With the growth and advancement of technologies, various activities and tasks that previously required manual labor have been replaced and adapted with the help of electronic devices and technological systems, which are capable of delivering results more quickly and at a lower cost. One of the areas most in demand for these technologies is security, especially regarding access control services. Among the most common examples is access control to rooms in institutions and buildings, implemented through locks on doors. This work aims to present a computational system for controlling an electronic lock, using a web application, mobile app, and an IoT solution, providing a more practical and cost-effective alternative to optimize access management and control, aiming to meet user needs efficiently and affordably.

Keywords: Access control; Locks; Computational system.

LISTA DE ILUSTRAÇÕES

Figura 1: Tabela do banco de dados.	14
Figura 2: Tabela com os testes de Usuários.	15
Figura 3: Tabela com os testes de Salas.	16
Figura 4: Tela de login do sistema.	17
Figura 5: Tela de login Mobile do sistema.	18
Figura 6: Tela home.	19
Figura 7: Tela home da versão mobile com as tranças abertas.	20
Figura 8: Tela home da versão mobile com as tranças fechadas.	20
Figura 9: Tela de usuários.	21
Figura 10: Tela de cadastro de usuário.	23
Figura 11: Tela de visualização das tranças.	24
Figura 12: Tela de cadastrar tranca.	26
Figura 13: Tela de exibir perfil do usuário.	27
Figura 14: Tela home da versão mobile.	27
Figura 15: Tela de Editar Usuário.	29
Figura 16: Informações da tranca.	30
Figura 17: Adicionar e buscar usuário.	32
Figura 18: Editar sala.	33
Figura 19: Circuito da Tranca.	34
Figura 20: Circuito da Tranca V2.	36
Figura 21: Circuito da Tranca com LED aceso.	37
Figura 22: Circuito da Tranca com LED aceso.	38

SUMÁRIO

1 .INTRODUÇÃO.....	7
2. OBJETIVOS GERAIS.....	8
2.1 OBJETIVOS ESPECÍFICOS.....	8
3. JUSTIFICATIVA.....	9
4 .METODOLOGIA.....	9
5. FUNDAMENTAÇÃO TEÓRICA.....	10
5.1.COMPOSIÇÃO GERAL DO SISTEMA.....	12
5.2.LINGUAGEM DE PROGRAMAÇÃO.....	13
5.3.FRONT-END.....	13
5.4.BACK-END.....	14
5.5.BANCO DE DADOS.....	14
5.6 TESTE DO SISTEMA.....	15
5.7 PROTOCOLO MQTT.....	16
6. DESCRIÇÃO DO SISTEMA.....	17
6.1.TELA DE LOGIN.....	17
6.2.TELA HOME.....	19
6.3.TELA DE USUÁRIOS.....	21
6.4.CADASTRAR USUÁRIO.....	23
6.5.TRANCAS.....	24
6.6.CADASTRAR TRANCAS.....	26
6.7.EXIBIR PERFIL DO USUÁRIO.....	27
6.8.EDITAR USUÁRIO.....	29
6.9.INFORMAÇÕES DA TRANCA.....	30
6.10.ADICIONA USUARIO.....	32
6.11.EDITAR SALA.....	33
7.DESCRICÃO DO HARDWARE DESENVOLVIDO.....	34
8. RESULTADOS E DISCUSSÕES.....	37
9. TRABALHOS FUTUROS.....	40
10. CONSIDERAÇÕES FINAIS.....	41
REFERÊNCIAS.....	43

1.INTRODUÇÃO

Com o constante crescimento dos aparelhos eletrônicos no Brasil, o estudo sobre o uso de TI revela que o país possui mais de dois dispositivos digitais por habitante. Esse aumento também reflete nos gastos e investimentos na área de tecnologia da informação (Uso de TI no Brasil, 2023), fazendo com que a segurança eletrônica se destaque no mercado. Embora já existam métodos simples, a evolução da TI no país traz constantemente novas opções, como dispositivos móveis e aparelhos menores e mais acessíveis.

Notando-se uma crescente utilização de aparelhos móveis em diversos países (Valente, 2019), nota-se que os smartphones estão cada vez mais presentes nas rotinas dos cidadãos, sendo utilizados por um tempo considerável ao longo do dia. Essa tendência pode trazer benefícios como mostrado por Valente (2019) com a crescente da média brasileira de downloads de aplicativos se tornar maior do que em 2016 e citando também os números do mapeamento do mercado de aplicativos móveis apresentando um aumento de 75% em relação a 2016.

Esses dados demonstram o impacto que o mercado de dispositivos móveis vem tendo nos últimos anos, aliado aos investimentos na área de TI e na área de segurança eletrônica, onde ambos podem exercer funções similares. Ao implementar uma solução de segurança da informação, um dos benefícios que podem ser obtidos é a criação de um ambiente mais seguro e redução de custos (Brasil, 2024), assim como garantir o acesso exclusivo de indivíduos de determinados grupos ou funções a locais e áreas mais reservados.

Para exercer esse trabalho, algumas opções comuns, como crachás de identificação, são amplamente utilizadas para controle de acesso, mas ainda geram gastos com trabalho manual e materiais. Aliando essas necessidades às práticas da segurança da informação, é possível chegar à solução da implementação de uma tranca eletrônica, que utiliza métodos já comuns junto com tecnologias aplicadas de programação e criação de circuitos eletrônicos para criar esse dispositivo.

Assim, tanto os setores privados quanto os públicos podem usufruir dessas soluções que podem otimizar o tempo e diminuir os gastos, auxiliando nas atividades de controle de acesso a locais automatizados e oferecendo um ambiente mais seguro. É possível usar fechaduras eletrônicas em diferentes áreas, tanto residenciais quanto comerciais, onde a sua função principal seja necessária.

Diante disso, este trabalho tem como objetivo abordar e solucionar algumas limitações encontradas em sistemas similares, desenvolvendo o SLOCKED: um sistema de autenticação de fechaduras integrado a um aplicativo, que visa não apenas autenticar, mas também gerenciar o controle de acesso às portas, oferecendo uma solução prática e economicamente viável para sua implementação sendo uma opção viável para as áreas em que possa ser requisitado.

2.OBJETIVOS GERAIS

Desenvolver uma fechadura eletrônica capaz de ser acessada tanto via desktop e mobile para atender necessidades do IFRN Campus Currais Novos.

2.1 OBJETIVOS ESPECÍFICOS

- Desenvolver uma aplicação que permita administrar o controle de acesso e atribuir permissões aos usuários para acessar salas específicas.
- Tornar possível que usuários possam adentrar as salas com o uso do aplicativo móvel sendo uma ferramenta de autenticação do usuário.
- Criar um sistema economicamente viável do que os demais sistemas similares do mercado.
- Trazer um sistema de fechaduras que auxilie o IFRN campus Currais Novos como os temas instituições
- Facilitar o trabalho dos administradores no controle de acesso.
- Auxiliar o acesso de alunos a salas e laboratórios para estudo ou prática de projetos.

3.JUSTIFICATIVA

A principal motivação para o desenvolvimento deste projeto surgiu das dificuldades enfrentadas por alunos e servidores do IFRN Campus Currais Novos/RN ao acessar salas com controle de acesso. Esses sistemas muitas vezes exigem o uso de chaves físicas ou autorização de uma pessoa designada, o que pode causar transtornos devido à falta de disponibilidade de um responsável ou técnico durante o período necessário. Essa ausência frequentemente resulta em atrasos e obstáculos.

Para resolver essas dificuldades, foi elaborada uma nova estratégia baseada nas trancas biométricas já utilizadas na instituição. A proposta consiste na criação de uma fechadura eletrônica que, inserida nas portas, possa ser destrancada por meio de um dispositivo móvel. Essa solução visa reduzir o trabalho manual e melhorar a gestão do acesso. Além disso, a fechadura poderá ser gerenciada por um sistema web, permitindo o controle remoto dos usuários e possibilitando a liberação ou restrição de acesso a determinadas áreas de maneira mais eficiente.

4.METODOLOGIA

Este trabalho teve como objetivo o desenvolvimento de uma tranca eletrônica acessível tanto via dispositivos *desktop* quanto móveis, com o propósito de atender tanto instituições públicas, como o Instituto Federal do Rio Grande do Norte (IFRN) Campus Currais Novos, quanto a rede privada. Para alcançar esse objetivo, uma abordagem metodológica foi adotada ao longo do projeto e as técnicas utilizadas.

As técnicas aplicadas neste trabalho basearam-se nos conhecimentos e práticas adquiridos nas disciplinas do Curso de Tecnologia em Sistemas para Internet (TSI). Buscou-se utilizar as principais tecnologias apresentadas nas matérias de Desenvolvimento *Front-end*, Banco de Dados e Desenvolvimento *Back-end* para compor a parte inicial do sistema, garantindo que aspectos cruciais para a versão *desktop* fossem aproveitados também na versão *mobile*.

Além disso, foram incorporadas práticas das disciplinas de Desenvolvimento para Dispositivos Móveis, Teste de *Software* e Programação para Internet das Coisas (IoT). Essas tecnologias permitiram a criação de uma interface *mobile* acessível em diversos dispositivos, assim como a implementação de funcionalidades de *hardware* através de técnicas de IoT. Realizamos testes sistemáticos para garantir a qualidade do sistema.

O desenvolvimento da aplicação iniciou-se com a construção do sistema como um todo. Utilizou-se a metodologia ágil para garantir que cada área do projeto recebesse a devida atenção. O primeiro passo foi a criação do sistema web, que posteriormente foi integrado com a versão para dispositivos móveis, mantendo a funcionalidade e os recursos já estabelecidos.

Em seguida, foi realizado o planejamento e a execução do protocolo MQTT (*Message Queuing Telemetry Transport*). Este protocolo foi escolhido para facilitar a comunicação entre as diferentes partes do sistema e para executar as funções da tranca, com o auxílio das informações recebidas da aplicação e do circuito construído. O ESP-32 desempenha um papel fundamental na comunicação via MQTT, integrando-se ao circuito criado.

Logo em seguida, foi realizada uma revisão bibliográfica sobre as tecnologias utilizadas no sistema, além de artigos e documentos que auxiliaram no desenvolvimento. Durante essa revisão, foram identificadas informações que

poderiam ser úteis para proporcionar um melhor desempenho ao sistema, assim como um funcionamento mais eficiente de suas funções principais.

5.FUNDAMENTAÇÃO TEÓRICA

As fechaduras eletrônicas foram desenvolvidas para atender a uma necessidade social recorrente: a segurança de determinados locais. Utilizando diversas tecnologias, esses dispositivos buscam oferecer proteção adequada, permitindo um maior controle do fluxo de pessoas, sem a necessidade de meios tradicionais, como fechaduras comuns ou chaves físicas. Assim, a implantação de uma solução de controle de acesso proporciona maior segurança aos ambientes em que estão instaladas, restringindo o acesso a pessoas mal-intencionadas (Pacheco; Araújo, 2015).

Embora já existam algumas opções no mercado, ainda não é comum encontrar sistemas que possuam dispositivos de instalação simples, com preços acessíveis e gerenciamento remoto (Guedes; Santos, 2015). Isso representa uma oportunidade para o desenvolvimento de sistemas que atendam a essa demanda. Utilizando os conceitos de IoT, a Internet das Coisas é um conceito no qual dispositivos do nosso dia a dia são equipados com sensores capazes de captar aspectos do mundo real (Nascimento, 2015).

A conexão móvel, ou a utilização de dispositivos móveis, tornou-se um meio de acesso cada vez mais comum na população. De acordo com dados apresentados pelo (Brasil, 2022), em dezembro de 2020 o Brasil registrou mais de 234 milhões de acessos móveis à internet. Esses números mostram que o setor de aplicativos móveis tende a continuar crescendo, juntamente com o setor de TI, que desenvolve essas aplicações. Conforme dados apresentados pela (FGV, 2024), na última década, os gastos e investimentos em TI nos bancos dobraram, e a expectativa é que esse crescimento continue alcançando aproximadamente R\$56 bilhões até 2026.

Uma alternativa para interação com as trancas eletrônicas envolve o uso de dispositivos diferentes. No projeto desenvolvido por (Gusmão; Castro; Zampolo, 2016), o destaque era a possibilidade de acesso através do reconhecimento facial. Já em (Guedes; Santos,2015), o sistema desenvolvido é mais robusto, porém, em ambos os casos, não foi utilizada a tecnologia do microcontrolador ESP-32, que oferece vantagens significativas. O ESP-32 já possui, em sua estrutura, conexões Wi-Fi e *Bluetooth*, facilitando a integração com dispositivos móveis e permitindo a conexão remota (Costa; Costa Neto, 2022).

O ESP-32 se apresenta como uma alternativa viável para sistemas que buscam reduzir o custo de seus componentes, além de ser compatível com uma variedade de linguagens de programação. Além disso, a ESP-32 possui conectividade Wi-Fi e *Bluetooth*, permitindo comunicação com outros dispositivos e a Internet. (Makiyama, 2023), tornando-o uma excelente opção de *hardware* para sistemas que necessitam de conexão Wi-Fi.

O projeto foi desenvolvido com base nos conceitos fundamentais de um sistema web, aplicados tanto à versão *desktop* quanto à versão *mobile*, garantindo acessibilidade em diferentes dispositivos. O *front-end* foi projetado tendo uma interface de forma simples e direta, atendendo às necessidades dos usuários comuns que são as necessidades para o acesso a salas de maneira mais fácil e proporcionando uma experiência mais intuitiva, além de facilitar o trabalho de gerenciamento dos administradores conseguindo melhorar o controle de acesso às portas de maneira mais segura que através do trabalho manual.

Esse *front-end* foi integrado ao *back-end*, o que possibilitou o funcionamento do sistema para a execução das funcionalidades requisitadas, além de garantir as regras de segurança necessárias. O *back-end*, em conjunto com o banco de dados, é responsável pelo armazenamento, gerenciamento e cadastramento das informações que serão geradas e inseridas no sistema. Essa integração assegurou a funcionalidade em ambas as versões e permitiu uma comunicação eficaz com o microcontrolador ESP-32, presente no circuito, que desempenha o papel da tranca eletrônica. A comunicação é realizada por meio do protocolo MQTT, que garante a troca de informações com o dispositivo.

5.1.COMPOSIÇÃO GERAL DO SISTEMA

A composição geral do sistema SLOCKED trata-se tanto de um sistema web *desktop* quanto de um aplicativo *mobile*, no qual o usuário efetua o *login*, tem acesso à informação de qual locais lhe é permitido o acesso, o aplicativo é também a forma na qual o usuário pede autorização ao sistema para acessar um local, o aplicativo envia uma mensagem ao microcontrolador (Um modelo ESP no caso deste sistema) que utiliza do protocolo MQTT para enviar uma mensagem um *broker* MQTT, que atua junto do back end do sistema.

Esse por sua vez aplica as regras de negócio, ou seja, verifica se é permitido acesso ao usuário aquela porta, caso o usuário tenha direito a adentrar aquele local, o acesso é liberado, e o aplicativo exibe uma mensagem confirmando. A aplicação web que foi desenvolvida é utilizada para o administrador do sistema, que pode cadastrar, excluir ou editar, os usuários do sistema, as portas/trancas, ou seja, a aplicação web é voltada mais para o gerenciamento do sistema enquanto o aplicativo móvel é destinado para os usuários comuns.

5.2.LINGUAGEM DE PROGRAMAÇÃO

A linguagem utilizada no desenvolvimento foi o *TypeScript*, devido à sua tipagem estática, que facilita a identificação de erros no código. O C++ foi empregado na criação do código do ESP-32.

O *TypeScript* é compatível com os frameworks utilizados no projeto, sendo aplicado tanto no front-end quanto no *back-end*. Essa linguagem desempenha um papel importante nas funções do sistema, incluindo a criação de rotas, requisições e regras de segurança, o que contribui para uma aplicação e manutenção mais eficientes.

Por outro lado, o C++, por ser uma linguagem mais tradicional, desempenha um papel crucial na configuração do EPS, utilizando as bibliotecas disponibilizadas para sua execução.

5.3.FRONT-END

Na etapa de desenvolvimento do front-end, foi escolhido o *framework* React para a criação da versão *desktop* do projeto, tanto para entregar uma interface simples aos usuários quanto para ser mais direta para conseguir ter uma usabilidade mais fácil, estilizada e a fácil adaptação a linguagem de programação escolhida.

Já na versão *mobile*, foi optado pela utilização de *frameworks* React Native que tem alguns benefícios sendo uma grande economia de tempo e recurso e é implementado em alguns aplicativos do mercado como *discord* e *instagram* (Cunha, 2022) possibilitando o desenvolvimento tanto com *javascript* como também *typescript*

juntamente com o uso dos conceitos de componentização, estado, estilos das páginas entre outros.

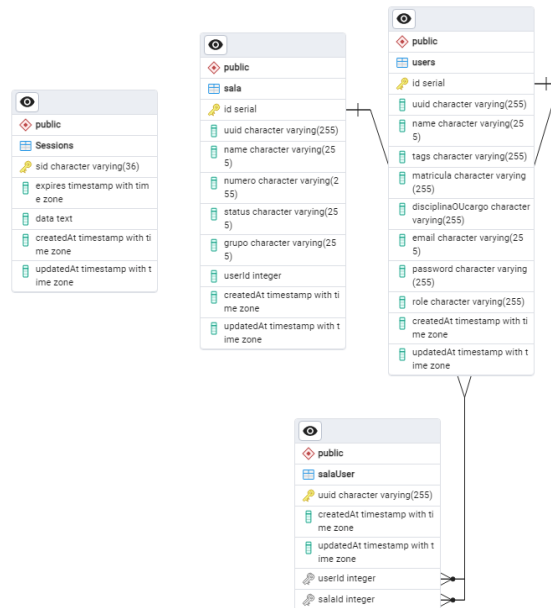
5.4.BACK-END

No *back-end* deste projeto, foram utilizados Node.js, Express.js, Sequelize e PostgreSQL. Esses componentes proporcionam um ambiente coeso para o desenvolvimento, permitindo a separação de preocupações e a implementação eficiente das funcionalidades requeridas. Além disso, possibilitaram a criação das rotas e a realização das requisições feitas tanto pelos usuários comuns quanto pelos administradores juntamente com as regras de segurança criadas para segurança dos usuários.

5.5.BANCO DE DADOS

No armazenamento de dados da aplicação, foi utilizado o banco de dados PostgreSQL devido à sua escalabilidade em relação ao volume de dados, bem como ao seu suporte aos padrões SQL, que auxiliam na criação e no gerenciamento de contas de usuários e das portas cadastradas, auxiliando juntamente com o *back-end* para a realização das funções do sistema como sendo uma delas o recebimento de dados, cadastro de usuários e de salas, gerenciamento de salas e usuários e entre outras atividades que o banco de dados deverá auxiliar.

Figura 1 - Tabela do banco de dados



Fonte: Banco de dados PostgreSQL do sistema SLOKED Acesso em: 11 dez.2024.

5.6 TESTE DO SISTEMA

Testes automatizados são utilizados para verificar o funcionamento de outros programas, permitindo a identificação de eventuais erros e possibilitando a realização de verificações constantes e de baixo custo por meio de um software específico para testes (Spirlandeli; Roland, 2019).

Para realizar testes nas diversas funcionalidades do sistema, incluindo aquelas relacionadas às salas e aos usuários, foi escolhido o *framework* Jest, que utiliza a linguagem JavaScript e é compatível com projetos em Node.js, React, *TypeScript*, entre outros. Além disso, o Jest é amplamente utilizado por grandes empresas em aplicativos como Instagram, Facebook e Spotify (JESTJS, 2025).

No total, foram desenvolvidos 27 testes, distribuídos em dois arquivos com extensão .test.js: um dedicado aos testes relacionados aos usuários e o outro às salas. Esses testes foram projetados para validar tarefas específicas do sistema e prevenir erros técnicos na aplicação. Para o aplicativo, foram aplicados testes padrões do React Native e *TypeScript*, além de testes unitários e end-to-end (de ponta a ponta) com o uso de bibliotecas apropriadas.

Figura 2 - Tabela com os testes de Usuários

```

PASS controllers/Users.test.js
  Creating Users
    ✓ Shouldn't create a User without passing a name (135 ms)
    ✓ Shouldn't create a User without passing a 'matricula' (62 ms)
    ✓ Shouldn't create a User without passing a 'disciplinaOUcargo' (59 ms)
    ✓ Shouldn't create a User without passing a 'email' (61 ms)
    ✓ Shouldn't create a User without passing a 'password' (2 ms)
    ✓ Shouldn't create a User without passing a 'confPassword' (1 ms)
    ✓ Shouldn't create a User passing a 'confPassword' diferent from 'password' (1 ms)
    ✓ Shouldn't create a User without passing neither 'password' neither 'confPassword' (1 ms)
    ✓ Shouldn't create a User without passing a 'role' (63 ms)
    ✓ Shold create a admin User (203 ms)
  Find Users
    ✓ get all users (9 ms)
    ✓ get user by id (15 ms)
    ✓ Should update the 1st user created in earlyer tests (69 ms)
    ✓ delete the 1st user created in earlyer tests (17 ms)

```

Fonte: Arquivo Users.test.js do sistema SLOKED Acesso em: 07 jan.2025.

Figura 3 - Tabela com os testes de Salas

```

PASS controllers/Salas.test.js
  creating Salas
    ✓ shouldn't create a sala without passing a name (9 ms)
    ✓ shouldn't create a sala without passing a numero (2 ms)
    ✓ shouldn't create a sala without passing a status (2 ms)
    ✓ shouldn't create a sala without passing a grupo (2 ms)
    ✓ shouldn't create a sala without passing a userId (2 ms)
    ✓ should create a sala called labredes2 (196 ms)
  get salas
    ✓ get all salas (11 ms)
    ✓ get sala by it's UUID (10 ms)
    ✓ shold not get any sala when pass invalid UUID (4 ms)
    ✓ get sala by user (10 ms)
    ✓ don't get sala by passing invalid userId (4 ms)
  update Sala
    ✓ shold update grupo on the early created sala (13 ms)
  delete salas
    ✓ delete early created sala by passing it's ID (14 ms)

Test Suites: 2 passed, 2 total
Tests:       27 passed, 27 total
Snapshots:  0 total
Time:        2.695 s, estimated 3 s

```

Fonte: Arquivo Salas.test.js do sistema SLOKED Acesso em: 07 jan.2025.

5.7 PROTOCOLO MQTT

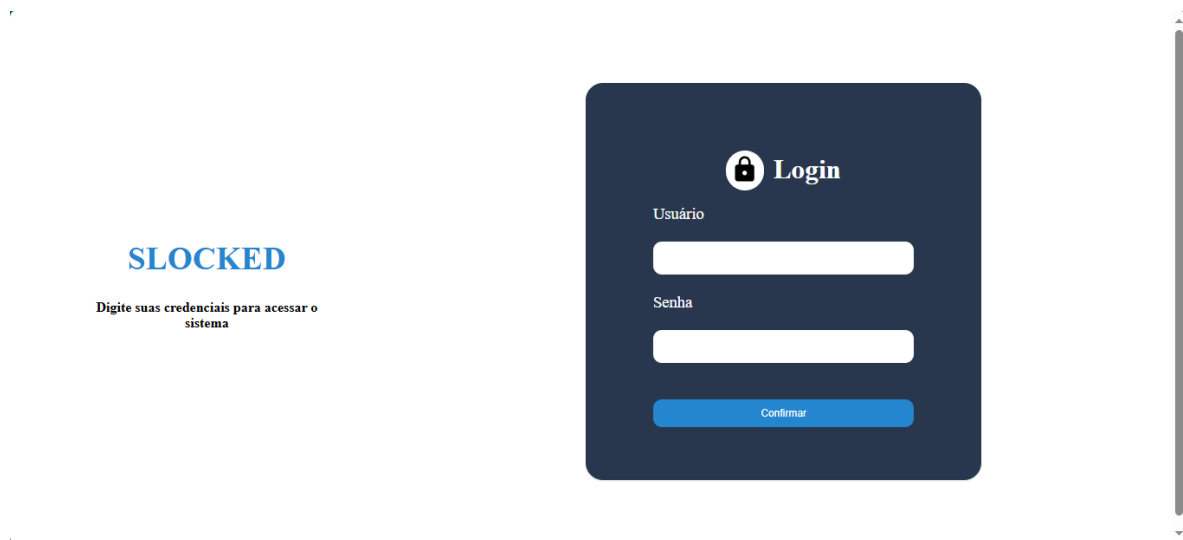
O MQTT (*Message Queuing Telemetry Transport*, em português, Transporte de Telemetria de Enfileiramento de Mensagens) é um protocolo que permite a comunicação entre dispositivos na IoT (*Internet of Things*, em português, Internet das Coisas). foi implementado a o uso do corretor de mensagem com código aberto Eclipse Mosquitto para desempenhar o papel de comunicação e troca de mensagem e sendo adequado para os dispositivos IoT. Para o sistema SLOCKED, a utilização do MQTT é crucial para o funcionamento integrado do sistema para a realização da comunicação entre o sistema web e mobile e das fechaduras eletrônicas.

6. DESCRIÇÃO DO SISTEMA

Nesta etapa, serão descritas as funcionalidades do sistema, utilizando imagens para melhor compreensão. Serão abordadas tanto a versão *desktop* quanto a *mobile*, disponíveis para usuários comuns e administradores. Além disso, será apresentado o circuito criado para a tranca eletrônica, com destaque para as peças de *hardware* utilizadas.

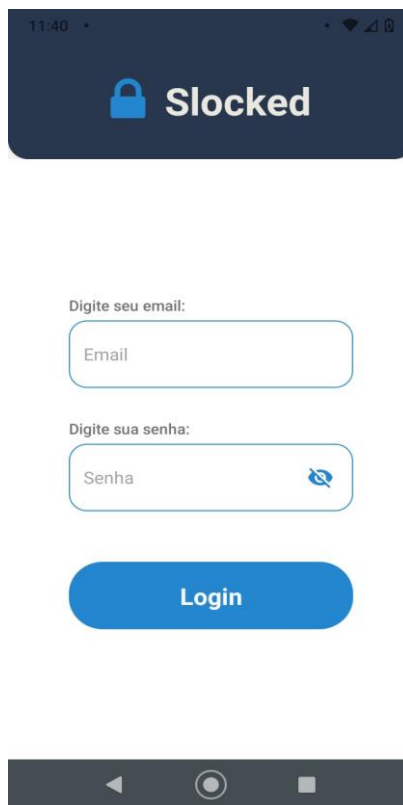
6.1.TELA DE LOGIN

Figura 4 - Tela de login do sistema



Fonte: Sistema SLOKED Acesso em: 19 dez.2024.

Figura 5 - Tela de login Mobile do sistema



Fonte: Sistema SLOKED Acesso em: 09.02.2025.

A tela inicial do sistema consiste em uma tela de login, na qual o usuário deve informar suas credenciais para que o acesso seja liberado. Após preencher os dados e clicar no botão de "continuar", o sistema verifica, primeiramente, o e-mail no banco de dados por meio do método **User.findOne**, buscando um usuário que possua o e-mail fornecido na requisição. Caso o e-mail não seja encontrado, o sistema retorna um erro com *status* 404 (usuário não encontrado). Após a validação do e-mail, o próximo passo é verificar se a senha informada corresponde àquela armazenada no banco de dados.

Para a validação segura da senha, foi utilizada a biblioteca *argon2*, que recebe dois parâmetros: o *hash* da senha armazenada e a senha fornecida pelo usuário na requisição. O método *verify* é utilizado para comparar as senhas, retornando *true* ou *false*, dependendo de sua correspondência. Se as senhas não coincidirem, o sistema retorna um erro com *status* 400 (senha incorreta). Caso a validação seja bem-

sucedida e o usuário seja encontrado, uma sessão é criada para manter o estado de autenticação entre as requisições.

Além disso, será gerado e armazenado o UUID (identificador único) do usuário, que é utilizado para identificá-lo nas requisições subsequentes enquanto ele permanecer autenticado. Após a autenticação, o usuário é redirecionado para a tela inicial, completando assim o processo de login.

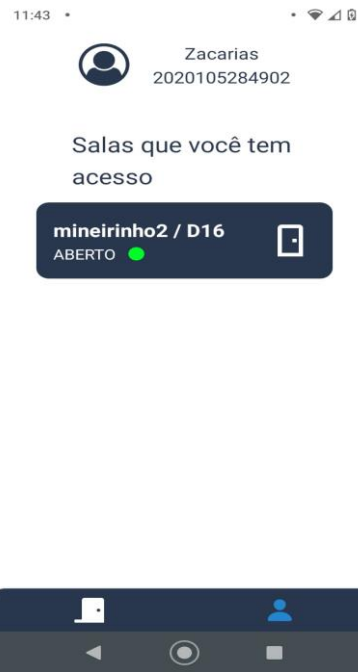
6.2.TELA HOME

Figura 6 - Tela home



Fonte: Sistema SLOKED Acesso em: 19 dez.2024.

Figura 7 - Tela home da versão mobile com as tranças abertas



Fonte: Sistema SLOKED Acesso em: 09.02.2025.

Figura 8 - Tela home da versão mobile com as tranças fechadas



Fonte: Sistema SLOKED Acesso em: 09.02.2025.

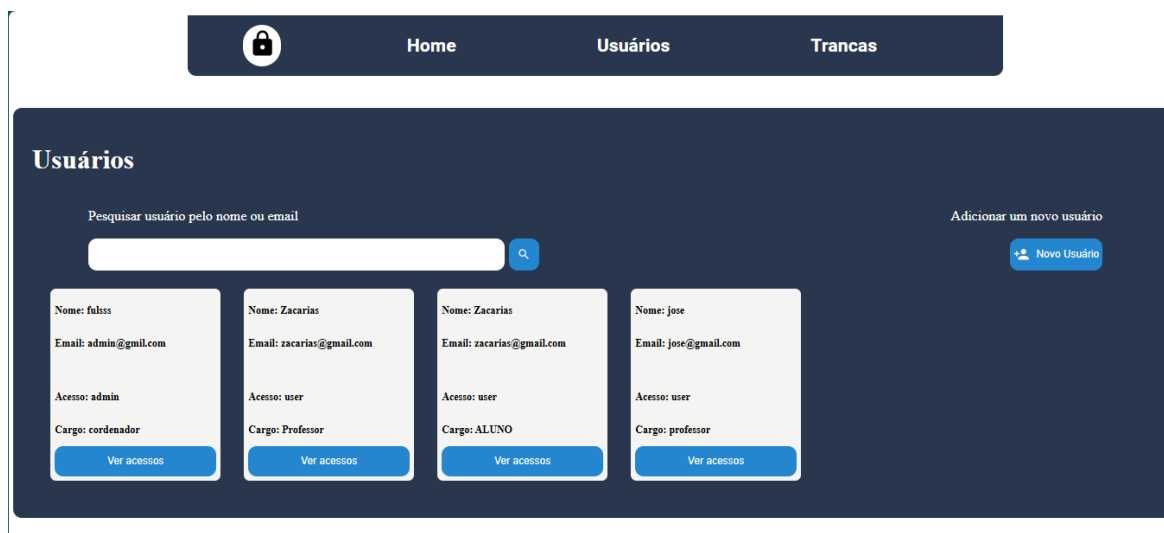
Assim que o processo de login for concluído, o usuário será redirecionado para a tela inicial. Nela, são exibidas informações básicas, como nome, cargo, data de ingresso no sistema e as salas às quais o usuário possui acesso, juntamente com o nome e o ID dessas salas. Além disso, há um botão que permite ao usuário sair da conta, executando a função de *logout* e retornando à tela de *login*.

As informações exibidas na tela são provenientes de um *hook* chamado **userProfile**, que é acionado assim que o componente *Home* é montado. Quando a função **useEffect** é executada, ela realiza uma requisição à API para buscar os dados do usuário, como nome, cargo, data de ingresso e as salas associadas a ele. Assim que os dados do *hook* são retornados, o componente *Home* renderiza e exibe as informações do perfil do usuário e as salas com seus respectivos nomes e IDs.

Adicionalmente, é exibido um *navbar* que permite ao usuário navegar entre as diferentes telas do sistema. As opções de navegação incluem: "*Home*", que leva à tela principal; "*Usuários*", onde é possível visualizar, pesquisar e cadastrar novos usuários (caso o usuário possua o cargo de administrador); e, por fim, a tela das trancas, onde são exibidas as trancas já criadas e é possível cadastrar uma nova tranca no sistema.

6.3.TELA DE USUÁRIOS

Figura 9 - Tela de usuários



Fonte: Sistema SLOKED Acesso em: 19 dez.2024.

Na tela de Usuários, o sistema exibirá os usuários cadastrados juntamente com suas principais informações, como nome, e-mail e cargo. O *back-end*, nesta página,

terá as seguintes funções para auxiliar na execução das operações disponíveis. A função **getUsers** tem como objetivo retornar a lista de todos os usuários cadastrados no banco de dados, com seus respectivos atributos e informações. Esses dados serão entregues ao *front-end*, que os renderizará na tela por meio do componente *Card*, exibindo as informações do usuário junto com um botão "Ver Acesso", que, ao ser clicado, redireciona para a página de detalhes do usuário.

Além de exibir a lista de usuários, a página contará com um filtro de texto para realizar buscas mais específicas. O parâmetro *filter*, caso não esteja vazio, será utilizado para filtrar os usuários com base nos dados inseridos, verificando se algum usuário contém o texto fornecido. A função **setFilter** atualizará o valor do estado de *filter* sempre que o usuário digitar algo na barra de busca. Caso nenhum texto seja inserido, a lista será carregada com todos os usuários cadastrados.

Por fim, a tela contará com um botão para cadastrar um novo usuário, que direciona o administrador para uma nova página, onde será possível realizar o cadastro de um usuário.

6.4.CADASTRAR USUÁRIO

Figura 10 - Tela de cadastro de usuário



A imagem mostra a tela de cadastro de usuário com o seguinte layout:

- Botão de voltar (←) no canto superior esquerdo.
- Título "Cadastrar Usuário" no topo central.
- Campos de entrada para: Nome, Email, Cargo, ID TAG RFID (acompanhado de um botão de toggle), Matricula (Opcional), Senha e Confirme sua senha.
- Seção "Nível de acesso:" com duas opções de radio buttons: "Administrador" e "Usuário".
- Botão "Confirmar" no rodapé.

Fonte: Sistema SLOKED Acesso em: 19 dez.2024.

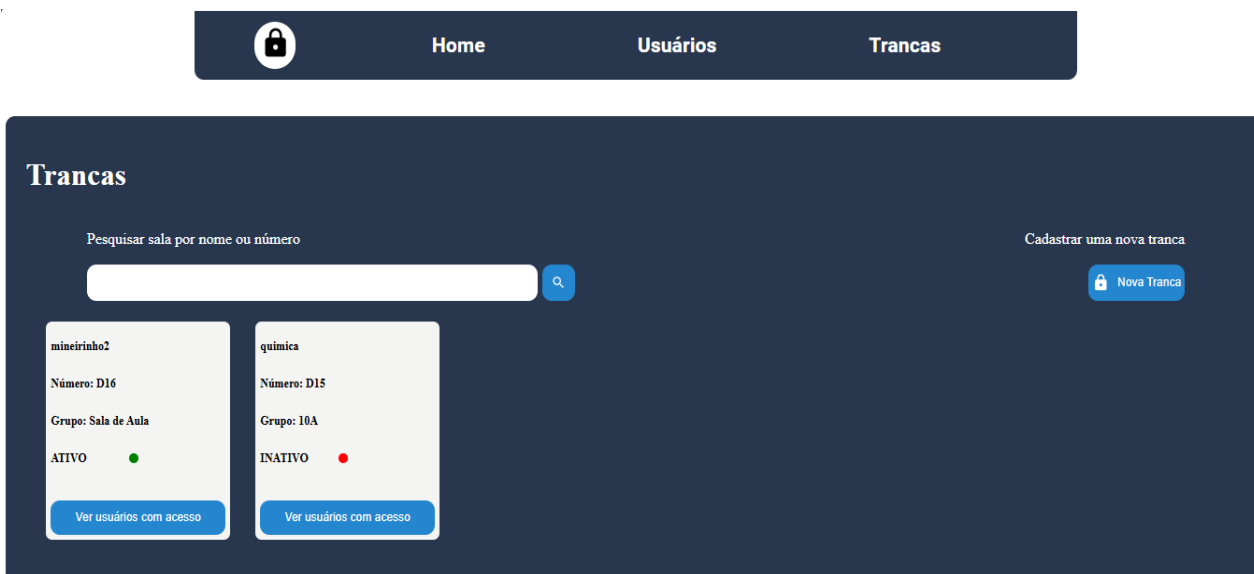
Caso o administrador deseje cadastrar um novo usuário e clique no botão "Novo Usuário" na página de exibição de usuários, ele será redirecionado para uma nova tela através da rota *users*, que utiliza o método *POST*, permitindo ao administrador criar novos usuários. Antes disso, são realizadas duas verificações: a primeira confere se o usuário está autenticado por meio do parâmetro *verifyUser*; a segunda verifica se o usuário possui permissões de administrador, utilizando o parâmetro *adminOnly*. Caso ambas as verificações sejam bem-sucedidas, a função **createUser** será chamada para realizar o cadastro do novo usuário.

Após a validação, será exibido um formulário com as informações necessárias para o cadastro. Esses dados serão usados tanto para permitir o acesso à plataforma na tela de login quanto para indicar as salas às quais o usuário está vinculado, o cargo que ocupa na instituição e seu nível de acesso. Este último determinará as regras de segurança e as funções que o usuário poderá executar no sistema. O formulário pode ser utilizado para cadastrar tanto um usuário comum quanto um administrador.

Ao inserir as informações, a função **createUser** receberá os dados preenchidos — como nome, e-mail, cargo, ID da tag, matrícula (opcional), senha, confirmação de senha e nível de acesso. A senha deverá ser digitada duas vezes para confirmação e será criptografada utilizando a biblioteca *argon2*, a fim de ser armazenada de forma segura no banco de dados. Ao final, o usuário será criado no banco de dados do sistema, e uma mensagem será retornada informando se a operação foi bem-sucedida ou se ocorreu algum erro durante o processo.

6.5. TRANCAS

Figura 11 - Tela de visualização das trancas



Fonte: Sistema SLOKED Acesso em: 19 dez.2024.

Na terceira tela, acessível pela barra de navegação, segue uma lógica semelhante à dos usuários, mas voltada para as trancas e salas. O front-end do sistema realiza uma requisição *GET* para o *back-end*, a fim de obter as informações de todas as salas salvas no banco de dados. O *hook useEffect* garante que a requisição seja feita assim que a página for carregada. Após a recepção dos dados, eles são entregues ao *front-end* e armazenados na variável *salas*, que contém a lista obtida pela API. Essa lista será utilizada para realizar a filtragem e renderização das salas na página.

Para realizar a filtragem, assim que o usuário digitar algo no campo de entrada, a função **setFilter** é chamada para atualizar o estado de *filter*, que armazena o texto digitado. Isso provoca a renderização dos componentes. Em seguida, o *hook useMemo*, que otimiza o desempenho, filtra as salas com base no valor de *filter* (que pode ser o nome ou o número da sala), atualizando a lista de salas filtradas. Caso o campo de busca esteja vazio, todas as salas serão exibidas.

As salas filtradas são renderizadas como cartões na página, utilizando o componente *Card*, que exibe as principais informações de cada sala, como nome, número e *status*. O *status* será indicado por uma cor verde (para salas ativas) e

vermelha (para salas inativas). Cada cartão possui um botão que, ao ser clicado, direciona o administrador para a página de permissões da sala.

Na página de permissões, o administrador poderá visualizar todas as informações e executar funções adicionais, como editar e alterar os dados cadastrados, incluindo o status da sala. O administrador também terá a opção de excluir a sala do sistema. Por fim, na tela das trancas, há um botão "Nova Tranca", que, ao ser clicado, redireciona para um formulário onde o administrador poderá cadastrar uma nova tranca no sistema.

6.6.CADASTRAR TRANCAS

Figura 12 -Tela de cadastrar tranca

A imagem mostra uma tela de formulário para cadastrar uma tranca. O formulário tem um fundo escuro azul com elementos em branco e um botão de confirmação em azul. No topo, há uma seta para voltar e o título "Cadastrar Tranca". Abaixo, há três campos de entrada rotulados "Nome da sala", "Identificação da sala" e "Grupo da sala". No rodapé, há um botão "Confirmar".

←

Cadastrar Tranca

Nome da sala

Identificação da sala

Grupo da sala

Confirmar

Fonte: Sistema SLOKED Acesso em: 19 dez.2024.

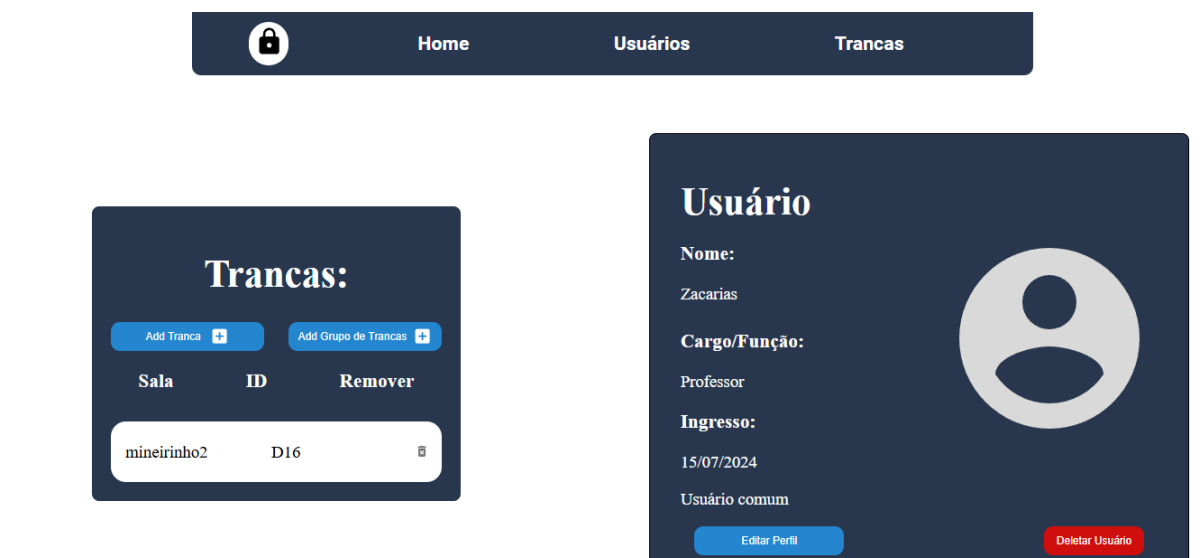
Caso o administrador deseje cadastrar uma nova tranca, ao clicar no botão "Nova Tranca", será redirecionado para uma tela contendo um formulário simples a ser preenchido com as informações necessárias, como o nome da sala onde a tranca será instalada, a identificação da sala (número de identificação) e o grupo da sala (ao qual a sala será associada). Após preencher as informações e clicar no botão "Confirmar", será realizada uma requisição **POST** para criar a nova sala.

A requisição é recebida no endpoint onde está definida a função **createSala**. Dentro dessa função, os dados necessários para o corpo da requisição são extraídos. Em seguida, o processo cria a sala utilizando o método `create()` do Sequelize. Esse método cria uma nova entrada na tabela de salas com os valores fornecidos pela requisição, além de associar a sala ao usuário que está realizando a requisição, tarefa que só pode ser executada pelo administrador.

Ao final do processo, caso a criação seja bem-sucedida, o sistema enviará uma mensagem de sucesso com *status* 201, informando que a tranca foi criada com sucesso. Caso ocorra algum erro, o sistema retornará uma mensagem de erro com *status* 500, indicando que houve um problema no processo de criação da sala.

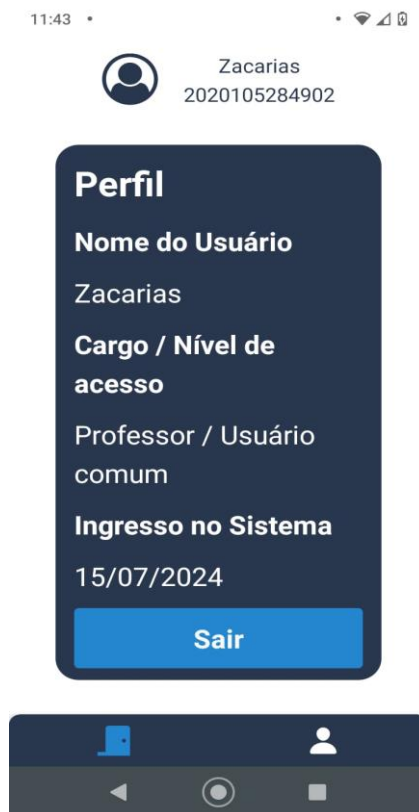
6.7.EXIBIR PERFIL DO USUÁRIO

Figura 13 - Tela de exibir perfil do usuário



Fonte: Sistema SLOKED Acesso em: 19 dez.2024.

Figura 14 - Tela home da versão mobile



Fonte: Sistema SLOKED Acesso em: 09.02.2025.

Ao clicar para visualizar o perfil de um determinado usuário, o administrador será redirecionado para outra tela. Nela, é utilizada uma função do *hook useEffect*, que buscará os dados do usuário por meio de seu UUID. O *back-end* faz uma requisição *GET* para o *endpoint* /usuarios/:id, retornando os dados do usuário, que serão então renderizados na tela. Essas informações incluem: nome, e-mail, cargo ou função desempenhada, data de cadastro, tipo de usuário (comum ou administrador) e as trancas das salas às quais o usuário tem acesso.

Entre as funções disponíveis para o administrador nesta tela, destacam-se: a adição de uma tranca ou de um grupo de trancas. Ao clicar no botão "Add Trancas", será aberto um modal que exibirá todas as salas cadastradas, permitindo ao administrador filtrá-las por nome ou número. Ao selecionar a sala desejada, a função **handleAddUserToSala(salaid)** será chamada para adicionar o usuário à sala selecionada. Simultaneamente, o *back-end* realizará uma requisição *POST* por meio

da função **addUserSala(salaid, user?.id)**, associando o usuário à sala no banco de dados. Em seguida, a lista será atualizada na tela.

Esse processo segue a mesma lógica para o botão "Adicionar grupo de trancas", com a diferença de que algumas funções têm nomenclaturas distintas, mas realizam processos semelhantes. Para a exclusão de uma tranca do perfil do usuário, um ícone de lixeira será exibido ao lado do nome e ID da tranca na lista de trancas do usuário. Caso o administrador clique neste ícone, será solicitada uma confirmação para a exclusão. Se confirmada, será chamada a função **handleDeleteUserSala()**, que enviará uma requisição *DELETE* para o endpoint responsável por remover a associação do usuário à tranca. A lista será atualizada, refletindo a desvinculação.

As duas últimas funções presentes na tela são: editar perfil e deletar usuário. Ao clicar no botão "Editar Perfil", o administrador será redirecionado para outra página onde poderá alterar as informações do usuário. Já ao clicar em "Deletar Usuário", será exibido um pedido de confirmação da exclusão. Se confirmado, o sistema chamará a função **handleDeleteUser()**, que enviará uma requisição *DELETE* para o *endpoint* /usuarios/:id, excluindo o usuário do banco de dados. Após a exclusão, a interface redireciona o administrador de volta à tela de usuários, sinalizando que o usuário foi excluído. Não será mais possível visualizar o perfil desse usuário nem encontrá-lo na lista geral de usuários cadastrados.

6.8. EDITAR USUÁRIO

Figura 15 - Tela de Editar Usuário



A tela de "Editar Usuário" apresenta um formulário com os seguintes campos e opções:

- Nome: Zacarias
- Email: zacarias@gmail.com
- Cargo: Professor
- ID TAG RFID: 882147 (com um ícone de tag)
- Matricula (Opcional): 2020105284902
- Senha: (campo oculto)
- Confirme sua senha: (campo oculto)
- Nível de acesso: Administrador Usuário

Um botão "Confirmar" está localizado na base da tela.

Fonte: Sistema SLOKED Acesso em: 19 dez.2024.

Ao clicar no botão “Editar Perfil”, o administrador será redirecionado para uma tela onde será chamada a função **getSingleUser(id)** do *hook useProfile*. Por meio de uma requisição *GET* ao backend, serão obtidos os dados do usuário, que serão armazenados e utilizados para preencher os campos do formulário. Esse formulário exibirá todas as informações cadastradas do usuário, permitindo ao administrador visualizar e editar os seguintes dados: nome, e-mail, cargo, tags, matrícula, senha, confirmação de senha e nível de acesso. O administrador poderá, então, promover o usuário para o nível de administrador ou rebaixá-lo para o nível de usuário comum.

Os campos do formulário já estarão preenchidos com as informações salvas anteriormente, e o administrador poderá editá-los. Após realizar as alterações, o administrador deverá clicar no botão “Confirmar” para salvar as modificações. Ao clicar, será chamada a função **handleUpdateUser**, que validará as senhas inseridas, comparando os dois campos e verificando se a senha possui pelo menos quatro caracteres. Caso haja algum erro, uma mensagem será exibida na tela. Se a validação

for bem-sucedida, a função `updateUser` será chamada, enviando uma requisição *PUT* ao *backend* com os dados atualizados.

Ao receber os dados enviados pelo *frontend*, o *backend* utilizará a função `updateUser` novamente, localizando o usuário no banco de dados através do `uuid` passado na URL. Se uma nova senha tiver sido enviada, ela será criptografada usando a biblioteca *argon2*. As demais informações serão atualizadas no banco de dados utilizando o método `User.update()`. Após a atualização, será exibida na tela a mensagem “*User Updated*”, confirmando que a edição foi bem-sucedida.

6.9. INFORMAÇÕES DA TRANCA

Figura 16 - Informações da tranca



Fonte: Sistema SLOKED Acesso em: 19 dez.2024.

Ao clicar no botão "Ver usuários com acesso", presente nas telas de visualização das tranças, o usuário será redirecionado para outra página. Nesta página, a função `loadSala`, que está presente dentro do `useEffect`, será chamada assim que os componentes forem montados. Essa função obtém os dados da trança, como nome, ID, grupo, status, usuários com acesso e a data de cadastro no sistema, que são carregados na interface. Além disso, será exibida uma lista de usuários que

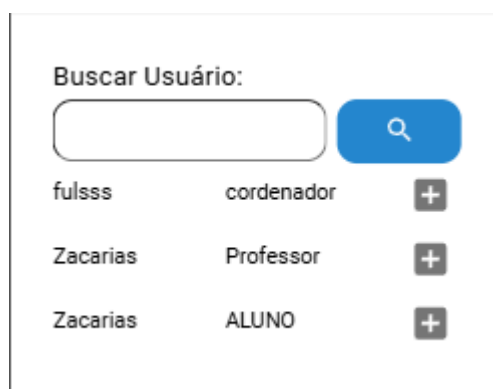
pertencem à sala, sendo cada um mapeado e mostrado juntamente com a opção de remoção.

A função de remoção, representada por um ícone de lixeira na lista logo após o nome do usuário, desvincula o usuário da trança. Ao clicar no ícone, um modal de confirmação será exibido. Se o administrador confirmar a remoção, a função **handleDeleteSalaUser** será chamada, utilizando o método **deleteSalaUser** para remover o vínculo entre o usuário e a sala. Após a remoção, a função **loadSala** recarregará a página com a lista de usuários atualizada.

Além disso, a página contém três botões localizados na lista de informações da trança. O botão "Adicionar um usuário" abrirá um modal para selecionar um usuário a ser adicionado à sala. O botão 'Editar' permitirá alterar as informações da trança. Por fim, o botão "Deletar Sala" solicitará uma confirmação antes de excluir a sala. Caso o administrador confirme a operação, o método **deleteSala**, do *hook useSalas*, será utilizado para remover a sala do banco de dados do sistema. Após a exclusão, o administrador será redirecionado de volta para a página da lista de tranças, confirmando a exclusão da sala no sistema.

6.10.ADICIONA USUARIO

Figura 17 - Adicionar e buscar usuário



Buscar Usuário:		
<input type="text"/>	<input type="button" value="🔍"/>	
fulsss	cordenador	<input data-bbox="986 1525 1023 1563" type="button" value="+"/>
Zacarias	Professor	<input data-bbox="986 1592 1023 1630" type="button" value="+"/>
Zacarias	ALUNO	<input data-bbox="986 1659 1023 1697" type="button" value="+"/>

Fonte: Sistema SLOKED Acesso em: 19 dez.2024.

Ao clicar para adicionar um usuário, será exibida uma lista com os usuários cadastrados no banco de dados do sistema, mostrando o nome e o cargo de cada um. Dentro dessa lista, há um campo de busca onde é possível digitar o nome do usuário,

cada vez que um termo é inserido, o estado *filter*, que utiliza o **useMemo** para otimizar a operação de busca, é atualizado, acionando a filtragem dos usuários disponíveis no sistema. Após a busca, será exibida uma lista com os usuários que atendem aos critérios definidos, a filtragem garante que apenas os usuários correspondentes aos critérios de busca sejam mostrados.

Além disso, a lista é filtrada para não exibir usuários que já fazem parte da sala, ao selecionar o usuário desejado e clicar no ícone de adicionar, localizado ao lado do cargo, a função **handleAddSalaToUser** será chamada. Essa função é responsável por adicionar o usuário à sala, recebendo o `userId` e acionando a função **addUserSala**, que envia a requisição ao *back-end* para realizar a inserção no banco de dados, assim que a operação é concluída, o modal é fechado e a página é atualizada, refletindo a inclusão do novo usuário na sala.

6.11.EDITAR SALA

Figura 18 - Editar sala



A imagem mostra um formulário de edição de sala em um aplicativo móvel. O formulário tem um fundo escuro azul-marrom com elementos em branco e um botão de confirmação em azul. No topo, há um ícone de seta para trás e o título "Editar Sala". Abaixo, há três campos de entrada: "Nome da sala" com o valor "química", "Identificação da sala" com o valor "D15" e "Grupo da sala" com o valor "10A". No rodapé, há um botão azul com o texto "Confirmar".

Fonte: Sistema SLOKED Acesso em: 19 dez.2024.

Caso o administrador deseje editar alguma informação da tranca, ao clicar no botão "Editar" no perfil da tranca, será redirecionado para uma página com um formulário contendo o nome da sala, a identificação da sala e o seu grupo. Através do

uuid da sala, serão retornados os dados já salvos no sistema, e o formulário será preenchido automaticamente com essas informações, o administrador poderá alterar qualquer um dos campos do formulário, sendo necessário clicar no botão "Confirmar" para validar as alterações.

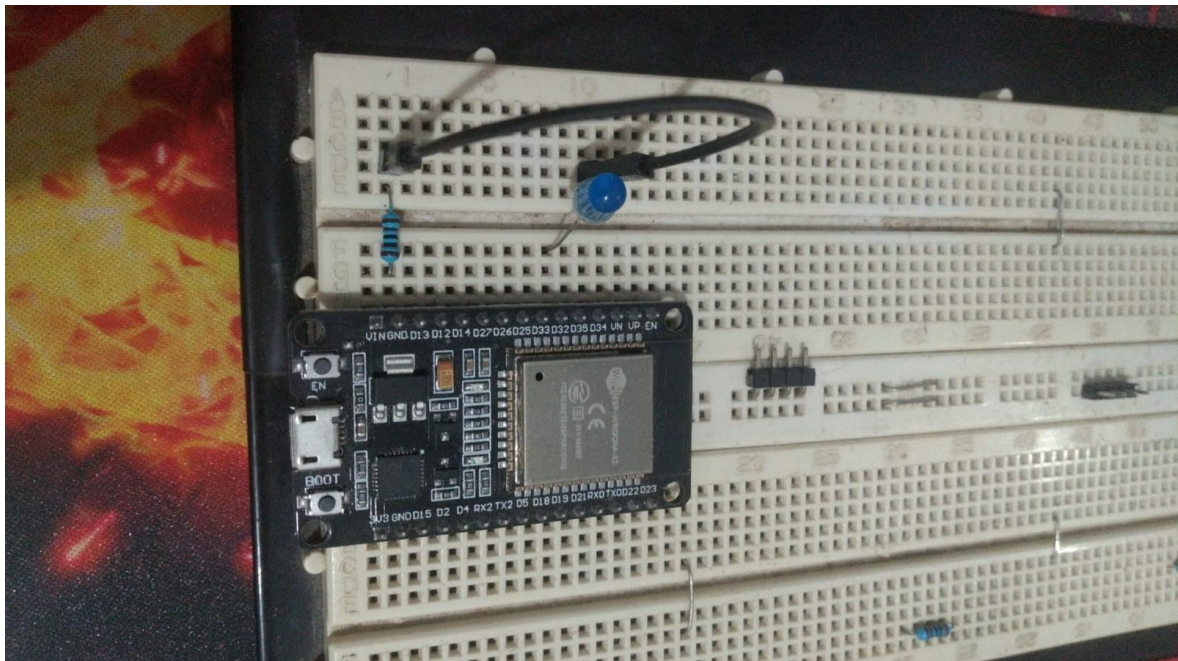
Ao clicar no botão "Confirmar", a função **handleEditar** será chamada, ela obterá os dados atualizados da página e fará uma requisição ao *backend*, que receberá as novas informações da sala. O *backend* verificará se o usuário tem permissão de administrador para realizar a alteração, se autorizado, os dados serão atualizados no banco de dados e uma mensagem de sucesso será retornada, indicando que a operação foi concluída, caso ocorra algum erro, uma mensagem de erro será exibida.

7.DESCRICÃO DO HARDWARE DESENVOLVIDO

O circuito do sistema SLOCKED foi projetado de forma simplificada, com foco na economia de recursos. Todo o circuito está centralizado em uma placa protoboard, onde os componentes são interligados. O principal dispositivo é o ESP-WROOM-32, um microcontrolador com conexão Wi-Fi e *Bluetooth*, responsável por receber as informações provenientes da interface gráfica e executar o acionamento da tranca, conectando-se com os demais componentes do circuito.

Entre os itens que compõem o circuito, destacam-se os resistores, que regulam o fluxo de energia elétrica e estão conectados ao pino GND do ESP-WROOM-32. Um cabo conecta a energia vinda do GND, passando pelo resistor, até o *LED*, que sinaliza o estado atual da tranca por meio da luz emitida. O LED possui uma extremidade ligada ao resistor e a outra conectada ao pino D32 do ESP-WROOM-32.

Figura 19 - Circuito da Tranca



Fonte: Autoria própria.

Periféricos:	Pinos:
ESP-WROOM-32	-
LED	D32

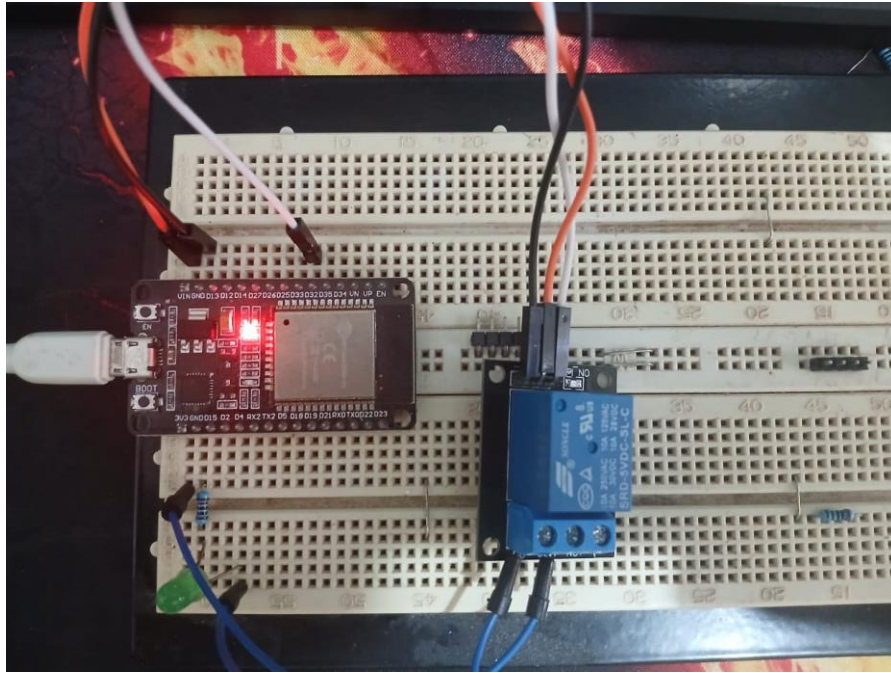
Resistor	GND
Cabo	-

Foi desenvolvida uma segunda versão do circuito, baseada na mesma lógica da versão anterior, diferenciando-se apenas pela adição de um relé e de alguns cabos responsáveis pelas conexões entre os dispositivos, possibilitando a comunicação entre eles. Essa modificação visa demonstrar o funcionamento do sistema, mantendo a proposta de um circuito mais simples e economicamente viável.

O principal destaque dessa nova versão é a utilização do relé 5V Single SRD-05VDC-SL-C, esse componente permite uma melhor percepção do funcionamento do sistema por meio do som emitido durante sua ativação, simulando a abertura de uma tranca. As conexões dos pinos realizadas na primeira versão foram mantidas sendo apenas redirecionadas para o relé, a principal diferença está na conexão do pino VIN, responsável por fornecer energia ao relé e do pino 3V3 que ao ser conectado à porta COM do relé, também energiza um LED.

Este LED atua como indicador visual da abertura da tranca, enquanto o som emitido pelo relé oferece uma indicação sonora, proporcionando assim um feedback tanto visual quanto auditivo do acionamento do sistema, o que evidencia a sua funcionalidade similar com a primeira versão.

Figura 20 - Circuito da Tranca V2



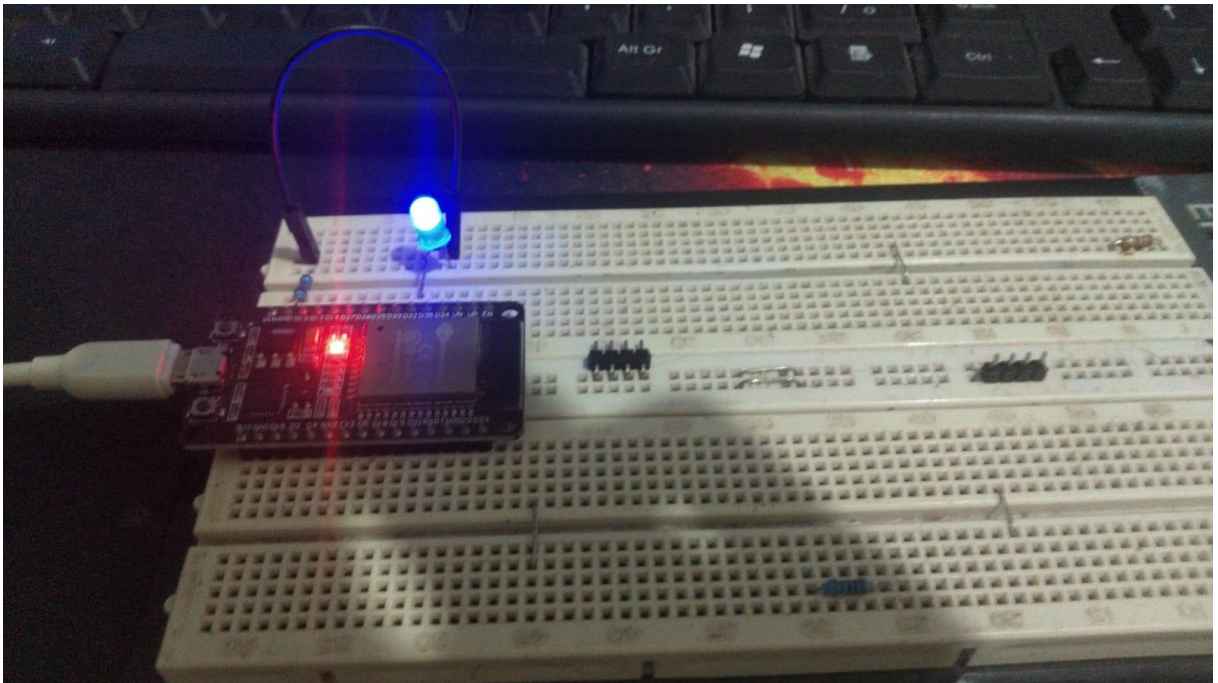
Fonte: Autoria própria.

Periféricos:	Pinos:
ESP-WROOM-32	-
LED	D32
Resistor	GND
Cabo	VIN,D32,GND,+,-,S,CO,COM
Relé 5V Souble SRD-05VDC-SL-C	-

8. RESULTADOS E DISCUSSÕES

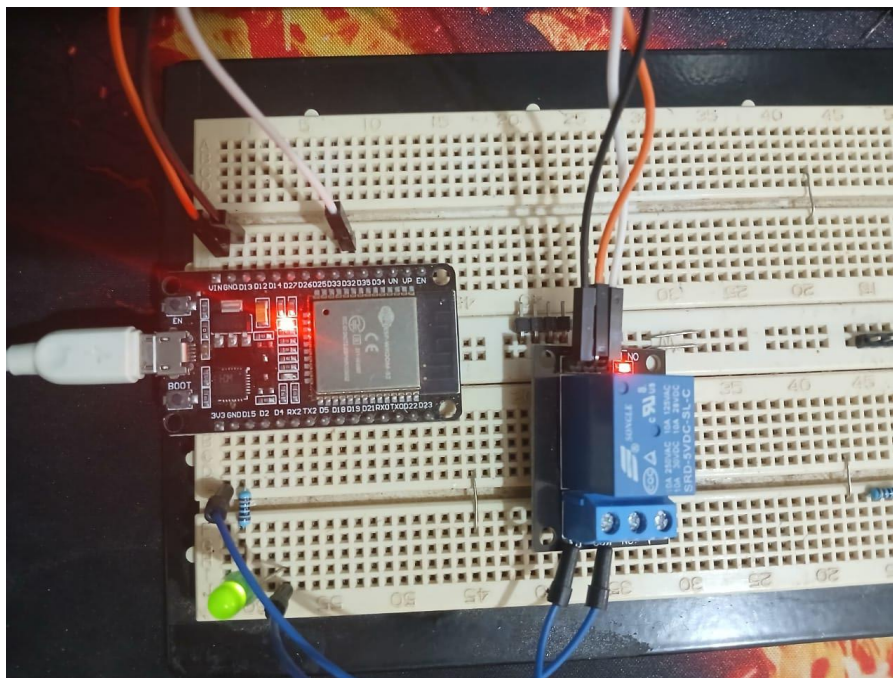
No decorrer deste trabalho, foi planejado e implementado um sistema com o objetivo de atender aos requisitos estabelecidos. Um desses objetivos foi a criação de um sistema de tranca eletrônica acessível via desktop e mobile, o qual foi alcançado por meio do desenvolvimento de um código funcional utilizando as tecnologias escolhidas. Outro resultado importante foi a criação da primeira versão de um circuito eletrônico capaz de receber o sinal do sistema, através de uma solução IoT que através da rede wi-fi irá mandar o sinal para o sistema web realizando uma requisição para atualizar o status da tranca e sinalizar a liberação de acesso à sala por meio de um *LED* que emite luz quando o acesso é autorizado.

Figura 21 - Circuito da Tranca com LED aceso



Fonte: Autoria própria.

Figura 22 - Circuito da Tranca V2 com LED aceso



Fonte: Autoria própria.

Uma segunda versão do sistema foi desenvolvida com o objetivo de demonstrar de forma mais clara sua funcionalidade por meio de outro módulo eletrônico, que também exerce a mesma função seguindo a lógica do primeiro modelo. Dessa forma, é possível apresentar uma demonstração tanto visual quanto sonora, evidenciando a compatibilidade do código com diferentes módulos eletrônicos, os quais podem ser implementados sem comprometer a ideia central da primeira versão nem do sistema como um todo.

Ao analisar os resultados obtidos, observou-se que foi possível atingir um dos objetivos do sistema, que é a comunicação entre os dispositivos, permitindo que tanto o *hardware* desenvolvido quanto o código do sistema interajam de forma eficaz. Esse resultado trouxe o sistema para mais perto da proposta teórica do trabalho, que visa melhorar a segurança de portas por meio do controle eletrônico de tranças.

Outros trabalhos que abordam temas semelhantes incluem a "Fechadura Microcontrolada com Gerência Centralizada" (Guedes; Santos, 2015). Embora haja semelhança, uma vez que também utiliza técnicas de IoT e comunicação Wi-Fi, esse trabalho se diferencia por integrar um sistema web e mobile para o gerenciamento e

o controle de acesso dos usuários às salas. Além disso, os componentes do circuito são mais simples, o que permite ao ESP-WROOM-32 cumprir a maior parte das tarefas necessárias. Esse aspecto torna o projeto mais econômico, sem comprometer sua eficácia.

Portanto, este trabalho contribui para a segurança eletrônica de salas, utilizando tecnologias que facilitam a implementação, sendo úteis tanto para instituições públicas quanto privadas, por ser uma solução simples e com um circuito de poucos componentes. Algumas limitações encontradas durante o estudo incluem a dificuldade de integrar o sistema com componentes de uma fechadura comum, como maçanetas e trinco de porta, devido à falta de material necessário e recursos econômicos. Como resultado, o foco deste trabalho permaneceu nos dispositivos eletrônicos criados, o que limitou o desenvolvimento do protótipo.

9. TRABALHOS FUTUROS

Apesar dos resultados alcançados ao longo deste trabalho, identificaram-se possíveis trabalhos futuros que não foram realizados devido aos custos envolvidos e ao prazo limitado para a entrega. Algumas tarefas que podem ser desenvolvidas futuramente, assim como em estudos anteriores, como o de Guedes e Santos (2015), incluem a construção de um protótipo de tranca eletrônica, utilizando o circuito apresentado neste trabalho como base, com a adição de novos componentes. Além disso, poderia ser realizada a implementação da tranca no IFRN – Campus Currais Novos ou em outras instituições e organizações públicas ou privadas.

10. CONSIDERAÇÕES FINAIS

Considerando a motivação inicial deste trabalho, que propôs um sistema de tranca eletrônica para automatização do controle de acesso a salas, o objetivo foi desenvolver um sistema aplicável em instituições e organizações. Foram estabelecidos objetivos que visavam tornar o sistema capaz de atender a maioria das demandas propostas, como o uso de diferentes dispositivos e a automatização do gerenciamento de acessos.

Dentre os principais objetivos deste trabalho, destaca-se a entrega de um sistema prático, que facilite o trabalho dos administradores no controle de permissões de acesso às salas, além de oferecer uma ferramenta simples de usar para os usuários comuns. O sistema também teve como foco a viabilidade econômica.

Ao longo do desenvolvimento, foram observadas algumas limitações, principalmente devido a dificuldades financeiras para adquirir componentes necessários para a criação de um protótipo completo. Além disso, o prazo de entrega estabelecido poderia interferir nos resultados finais, o que levou à conclusão de que seria possível desenvolver apenas o *hardware* do sistema, contudo, esse protótipo foi capaz de atingir a maioria das metas estabelecidas.

Em resumo, os resultados deste trabalho demonstraram a praticidade do sistema, que conseguiu cumprir sua função ao permitir a comunicação entre o sistema computacional e o circuito desenvolvido. Essa interação foi possível por meio do envio e recebimento de dados, viabilizando a ação de abrir e fechar a tranca, o circuito, utilizando poucos componentes eletrônicos, fez com que o custo fosse economicamente viável, além de oferecer uma solução simples para o gerenciamento de usuários com acesso a determinadas salas.

No entanto, é importante destacar que, ao contrário de outros estudos e trabalhos apresentados ao longo deste relatório, o sistema desenvolvido neste trabalho se diferencia pela interação com dispositivos móveis e *desktop*, além de contar com um circuito mais econômico e com menos componentes, o que constitui seu principal diferencial.

Com os resultados obtidos, percebe-se que este trabalho pode contribuir positivamente para a área em questão. Sua implementação pode trazer benefícios para as tarefas de segurança e contribuir para estudos futuros, possibilitando melhorias na funcionalidade do sistema.

REFERÊNCIAS

BRASIL. Ministério da Ciência, Tecnologia e Inovações. Laboratório Nacional de Computação Científica – LNCC. **Os quatro pilares da segurança da informação: confidencialidade, disponibilidade, integridade e autenticidade.** 08 ago. 2024. Disponível em: <https://www.gov.br/lncc/pt-br/centrais-de-conteudo/campanhas-de-conscientizacao/gestao-de-seguranca-da-informacao/os-quatro-pilares-da-seguranca-da-informacao-2013-confidencialidade-disponibilidade-integridade-e-autenticidade>. Acesso em: 16 out. 2024.

BRASIL. Serviços e Informações do Brasil. **Brasil registrou mais de 234 milhões de acessos móveis em 2020.** 31 out. 2022. Disponível em: < [https://www.gov.br/pt-br/noticias/transito-e-transportes/2021/04/brasil-tem-mais-de-234-milhoes-de-acessos-moveis-em-2020#:~:text=O%20Brasil%20registrou%20234%2C07,Minist%C3%A9rio%20das%20Comunica%C3%A7%C3%B5es%20\(MCom\)](https://www.gov.br/pt-br/noticias/transito-e-transportes/2021/04/brasil-tem-mais-de-234-milhoes-de-acessos-moveis-em-2020#:~:text=O%20Brasil%20registrou%20234%2C07,Minist%C3%A9rio%20das%20Comunica%C3%A7%C3%B5es%20(MCom).). >. Acesso em: 24 out. 2024.

COSTA, Pedro Henrique Santiago; COSTA NETO, João Bosco. **Sistema embarcado de baixo custo utilizando o ESP-32 na telemedicina.** 2022. Disponível em: <<https://repositorio.animaeducacao.com.br/items/1a99b886-e87d-4cc0-bc2a-32659048481f>>. Acesso em: 24 out. 2024

CUNHA, Andreo. **React Native: o que é e tudo sobre o Framework.** Alura. 01 set. 2022. Disponível em: <https://www.alura.com.br/artigos/react-native#:~:text=O%20React%20Native%20foi%20lan%C3%A7ado,Instagram%2C%20Facebook%20e%20entre%20outros>. Acesso em: 30 set. 2024.

FUNDAÇÃO GETULIO VARGA-FGV. **Pesquisa revela que Brasil tem 480 milhões de dispositivos digitais em uso, sendo 2,2 por habitante.** 27 jun. 2024. Disponível em: <<https://portal.fgv.br/noticias/pesquisa-revela-brasil-tem-480-milhoes-dispositivos-digitais-uso-sendo-22-habitante>>. Acesso em: 24 set. 2024.

GUEDES, Yuri M.; SANTOS, Marcelo C. P. Fechadura microcontrolada com gerência centralizada. **Seminários de Trabalhos de Conclusão de Curso do Bacharelado em Sistemas de Informação**, v.1, n.1, 2015. Disponível em: <https://sistemas.jf.ifsudestemg.edu.br/seer/revistabsi/article/view/35/0>. Acesso em: 11 dez. 2024.

GUSMÃO, Arthur Trindade Abreu de; CASTRO, Agostinho Luiz da Silva; ZAMPOLO, Ronaldo de Freitas. Fechadura baseada em reconhecimento facial via dispositivos móveis Android. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES – SBrT2016, 34., 2016. Santarém, PA. **Anais...** Santarém, PA, 30 de agosto a 02 de setembro, 2016. Disponível em: <https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwirj9jstaCKAxXCqZUCHQOTotUQFnoECBgQAQ&url=https%3A%2F%2Fbiblioteca.sbrt.org.br%2Farticlefile%2F1187.pdf&usg=AOvVaw37USwDaeJpr66YyWhU6Trk&opi=89978449>. Acesso em: 11 dez. 2024.

JESTJS. **Jest é um poderoso Framework de Testes em JavaScript com um foco na simplicidade**.2025. Disponível em: <https://jestjs.io/pt-BR/> Acesso em: 07 jan. 2025.

MAKIYAMA, Marcio. Placa ESP32: Descubra o que é, para que serve e muito mais! **Victor Vision**, 6 nov. 2023. Disponível em:< <https://victorvision.com.br/blog/placa-esp32/#:~:text=A%20Placa%20ESP32%20%C3%A9%20um,e%20outras%20tecnologias%20de%20conex%C3%A3o.>>. Acesso em: 24 set. 2024.

NASCIMENTO, Rodrigo. O que, de fato, é a internet das coisas e que revolução ela pode trazer? **Computerworld**, 12 mar. 2015. Disponível em: <https://computerworld.com.br/negocios/2015/03/12/o-que-de-fato-e-internet-das-coisas-e-que-revolucao-ela-pode-trazer/> . Acesso em: 11 dez. 2024.

PACHECO, Tatiane Martins; ARAÚJO, Marco Antônio Pereira. Sistema de gerência centralizada de fechaduras microcontroladas. **Seminários de Trabalhos de Conclusão de Curso do Bacharelado em Sistemas de Informação**, v.1, n.1, 2015. 2015. Disponível em: <https://sistemas.jf.ifsudestemg.edu.br/seer/revistabsi/article/view/34>. Acesso em: 11 dez. 2024.

SPIRLANDELI, Cleber; ROLAND, Carlos Eduardo de França. A utilização de testes automatizados no desenvolvimento de software. **Revista EduFatec: educação, tecnologia e gestão**, Franca, v.2, n.2, p.1-24, ago./dez. 2019. Disponível em: <https://revistaedufatec.fatecfranca.edu.br/wp-content/uploads/2020/03/edufatec-n02v2a01.pdf>. Acesso em: 08 jan. 2025.

USO de TI no Brasil: País tem mais de dois dispositivos digitais por habitante, revela pesquisa. **Portal FGV**, 2023. Disponível em: <https://portal.fgv.br/noticias/uso-ti-brasil-pais-tem-mais-dois-dispositivos-digitais-habitante-revela-pesquisa#:~:text=O%20Brasil%20tem%20464%20milh%C3%B5es,do%20FGVcia%20sobre%20o%20Mercado>. Acesso em: 16 out. 2024.

VALENTE, Jonas. Brasil é 5º país em ranking de uso diário de celulares no mundo. **Agência Brasil**, Brasília, 18 jan. 2019. Disponível em: <https://agenciabrasil.ebc.com.br/geral/noticia/2019-01/brasil-foi-5o-pais-em-ranking-de-uso-diario-de-celulares-no-mundo#:~:text=Os%20brasileiros%20passaram%20mais%20de,tempo%20dispendido%20com%20esse%20aparelho>. Acesso em: 16 out.2024.