UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ
INSTITUTO DE GEOCIÊNCIAS E ENGENHARIAS
FACULDADE DE ENGENHARIA DA COMPUTAÇÃO
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

Projeto Final de Curso II

SISTEMA INTEGRADO DE GERENCIAMENTO DE ACESSO E ATIVOS DE LABORATÓRIOS OPERACIONAIS (SIGA-LO)

THIAGO ELEUTERIO DA SILVA

Marabá - PA

2024

UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ
INSTITUTO DE GEOCIÊNCIAS E ENGENHARIAS
FACULDADE DE ENGENHARIA DA COMPUTAÇÃO

THIAGO ELEUTERIO DA SILVA

SISTEMA INTEGRADO DE GERENCIAMENTO DE ACESSO E ATIVOS DE LABORATÓRIOS OPERACIONAIS (SIGA-LO)

Projeto Final de Curso II, apresentado à Universidade Federal do Sul e Sudeste do Pará, como critério de avaliação e requisitos necessários para obtenção do Título de Bacharel em Engenharia da Computação.

Orientador:
Prof. Dr. Diego Kasuo Nakata da Silva

Marabá - PA

2024

UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ
INSTITUTO DE GEOCIÊNCIAS E ENGENHARIAS
FACULDADE DE ENGENHARIA DA COMPUTAÇÃO

THIAGO ELEUTERIO DA SILVA

SISTEMA INTEGRADO DE GERENCIAMENTO DE ACESSO E ATIVOS DE LABORATÓRIOS OPERACIONAIS (SIGA-LO)

Projeto Final de Curso II, apresentado à Universidade Federal do Sul e Sudeste do Pará, como critério de avaliação e requisitos necessários para obtenção do Título de Bacharel em Engenharia da Computação.

Marabá, 18 de setembro de 2024.

BANCA QUALIFICADORA:

Diego Kasuo N. da Silva
Prof. Dr Diego Kasuo Nakata da Silva
(Orientador - UNIFESSPA - IGE - FEC)

Leslye Estefania Castro Eras
Prof.ª Dra Leslye Estefania Castro Eras
(Membro da Banca - UNIFESSPA)

Cindy Stella Fernandes
Prof.ª Dra Cindy Stella Fernandes
(Membro da Banca - UNIFESSPA)

Marabá - PA

2024

Dados Internacionais de Catalogação na Publicação
(CIP) Universidade Federal do Sul e Sudeste do Pará
Biblioteca Setorial II da UNIFESSPA

S729s Silva, Thiago Eleuterio da
Sistema integrado de gerenciamento de acesso e ativos de laboratórios operacionais (SIGA-LO) / Thiago Eleuterio da Silva. — 2024.
53 f.: il., color.

Orientador (a): Diego Kasuo Nakata da Silva.
Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Sul e Sudeste do Pará, Instituto de Geociências e Engenharias, Faculdade de Computação e Engenharia Elétrica, Curso de Engenharia da Computação, Marabá, 2024.

1. Engenharia de computação. 2. Laboratórios – Análise – Equipamento e acessórios. 3. Automação. 4. Sistemas de controle inteligente. I. Silva, Diego Kasuo Nakata da, orient. II. Título.

CDD: 22. ed.: 006.8

Elaborado por Nádia Lopes Serrão – CRB-2/575

AGRADECIMENTOS

Dedico esse trabalho a minha mãe, Roberta, e a meu pai, Francelino, ambos que hoje já não estão juntos de mim na jornada. Mas como já disse uma vez a figura fictícia Anthony Stark, “Parte da jornada é o fim” e de certeza ambos concluíram com maestria a sua.

A minha mãe, uma grande mãe, excelente companheira e defensora das causas justas, gostaria de agradecê-la por ter me introduzido ao primeiro computador que utilizei, a criação de empatia e ajuda ao próximo, ao senso de aventura e de criação de soluções, além de me mostrar sempre a infinidade de possibilidades do que posso criar, realizar e conquistar com o pouco que tenho, sempre almejando maior crescimento pessoal. Agradeço por sempre estar presente enquanto viva, nas mais felizes e tristes histórias de minha vida.

Ao meu pai, por sempre me fornecer conhecimento e colaborar na criação do meu senso crítico, sempre duvidando sobre verdades absolutas de que algo deva ser como é dito e feito. Agradeço por sempre estar presente, junto a minha mãe, nos mais terríveis desafios de minha vida e sempre me apoiar na decisão de minha carreira, me dando dicas e conselhos, ao auxílio financeiro que pode me proporcionar conforto e acesso à informação e novas tecnologias que foram introduzidas juntos a este projeto.

Agradeço a minha parceira, Andressa, por me apoiar nas tomadas de decisões que vieram a aparecer no caminho, sempre acreditar no potencial que possuo, além de ser um pilar emocional em um momento tão conturbado e difícil que está sendo neste instante. Continue sendo essa pessoa incrível, dedicada, exagerada e doce que é por isso e vários outros detalhes são o que formam a incrível personalidade que possui a qual me entreguei e que me intriga todos os dias me ensinando como a vida tem outros lados a serem visualizados. Nunca esquecerei.

Agradeço também ao casal de professores Dra. Leslye Castro e Dr. Diego Kasuo que me acolheram desde o início de minha jornada acadêmica, realizando produções de artigos e pesquisas, bem como a professora Dra Cindy Stella por sempre trazer a razão com seu lado mais crítico e avaliador que sempre me trazia novas ideias de inovação. Obrigado pela chance oferecida e pela paciência com seu pupilo ansioso e agoniado. Vou sempre melhorar e manter em mente a frase que me ensinaram, “Você não precisa criar a roda, mas adaptar o que existe às suas necessidades”.

Aos meus incríveis amigos de vida, Henrique (um irmão de outra mãe), Lucas Antônio (“*Lucarion*”), Lucas Leite (“*Prefab*”), Arthur Oliveira (“*Shubunaga*”), Felipe, Messias (“*Pincher Palmeirense*”) e Naiara (“*HannaMontana*”), agradeço por estarem ao meu lado.

“A menos que modifiquemos a nossa maneira de pensar, não seremos capazes de resolver os problemas causados pela forma como nos acostumamos a ver o mundo”.

(Albert Einstein)

“A persistência é muito importante. Você não deve desistir, a menos que seja forçado a desistir”.

(Elon Musk)

RESUMO

O Galpão de Laboratórios da Universidade Federal do Sul e Sudeste do Pará (UNIFESSPA), inaugurado em 6 de abril de 2018, é um espaço destinado ao desenvolvimento das atividades práticas dos cursos de Engenharia, Sistemas de Informação e Geologia. O local abriga 17 laboratórios de pesquisa operacional e um amplo espaço aberto para o desenvolvimento de projetos. O atual modelo de acesso a esses laboratórios se dá por meio de fechaduras convencionais. Consequentemente, esse mecanismo gera empecilhos, tais como: perda de chaves, dificuldade de acesso, exposição de ativos de valor e falhas na gerência de entrada e saída de pessoas nos laboratórios. Sob esse ponto de vista, o presente trabalho objetiva a criação de um protótipo de tranca eletrônica inteligente como solução para o modelo atual, utilizando a tecnologia de comunicação LoRa para comunicação do protótipo com o banco de dados. Nesse âmbito, também é apresentado um sistema de gerenciamento web para controle de acesso e acompanhamento de entrada e saída dos laboratórios.

Palavras-chave: IoT, Trancas Inteligentes, LoRa, Laboratórios Operacionais, Long Range, Gerenciamento de Acesso.

ABSTRACT

The Laboratory Warehouse of the Federal University of Southern and Southeastern Pará (UNIFESSPA), inaugurated on April 6, 2018, is a space designed for the development of practical activities for the Engineering, Information Systems, and Geology courses. The facility houses 17 operational research laboratories and a large open space for project development. The current access model to these laboratories is through conventional locks. Consequently, this mechanism generates obstacles such as key loss, difficulty of access, exposure of valuable assets, and failure in managing the entry and exit of people within the laboratories. From this standpoint, the present work aims to create a prototype of an intelligent electronic lock as a solution for the current model, using LoRa communication technology to connect the prototype with the database. In this scope, a web-based access control and monitoring system is also presented for tracking the entry and exit of individuals in the laboratories.

Keywords: IoT, Smart Locks, LoRa, Operational Laboratories, Long Range, Access Management.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

MQTT	*Message Queuing Telemetry Transport*
SIGA-LO	*Sistema Integrado de Gerência de Acesso à Laboratórios Operacionais*
LoRa	*Long Range - Longo Alcance*
WiFi	*Wireless Fidelity - Fidelidade sem Fio*
UNIFESSPA	*Universidade Federal do Sul e Sudeste Do Pará*
IoT	*Internet of Things*
RFID	*Radio Frequency Identification*
API	*Application Programming Interface*
SIGA-LO	*Sistema Integrado De Gerência de Acesso à Ambientes Laborais*
BACKEND	*Processo interno*
FRONTEND	*Interface de usuário*
RSSI	*Received Signal Strength Indication*

SUMÁRIO

1- INTRODUÇÃO

A pesquisa operacional surgiu da necessidade de analisar cientificamente as operações e melhor alocar os recursos escassos durante a Segunda Guerra Mundial. Hillier (2006) descreve a convocação de cientistas para adotar uma abordagem científica para tratar de problemas táticos e estratégicos, desenvolvendo pesquisas sobre as operações militares que ocorriam na época.

"Portanto, a pesquisa operacional é aplicada a problemas envolvendo como conduzir e coordenar as operações (isto é, as atividades) em uma organização" (Hillier, 2006). Nesse cenário, originam-se os laboratórios operacionais, que utilizam a metodologia da pesquisa operacional proposta por Hillier, combinando áreas distintas como manufatura, telecomunicações, internet das coisas, ciência de dados e outras, com o objetivo de desenvolver melhores soluções para problemáticas apresentadas.

Oracle (2023) descreve a Internet das Coisas (IoT) como uma "rede de objetos físicos incorporados a sensores, softwares e outras tecnologias com o objetivo de conectar e trocar dados com outros dispositivos e sistemas pela internet". O aumento do uso desses dispositivos tem sido tão relevante que o site IoT Analytics (2023) reportou um aumento de 18% nas conexões de aparelhos voltados para IoT, com a expectativa de um novo aumento de 16% em 2023, totalizando aproximadamente 16,7 bilhões de endpoints ativos. Em consonância, o site IoT Analytics (2024) já mostra tendências de crescimento e melhorias no mercado com a inclusão de inteligência artificial aos dispositivos voltados para IoT.

Nesse contexto, a tecnologia LoRa, sigla para Long Range (Longo Alcance, em português), surge como uma importante aliada para sistemas voltados para IoT. Segundo a TCT Brasil (2019), LoRa é uma tecnologia de comunicação sem fio projetada para operar em redes de longo alcance, com baixa largura de banda e consumo de energia reduzido. Ela foi especificamente desenvolvida para atender às necessidades de dispositivos restritos, como sensores e dispositivos da Internet das Coisas (IoT).

O LoRa utiliza técnicas de modulação de espectro espalhado *(Spread Spectrum)* para permitir comunicações de longo alcance, inclusive a nível de ruído, enquanto consome uma quantidade mínima de energia. Isso o torna adequado para aplicações em áreas rurais, urbanas e industriais, onde a eficiência energética e a capacidade de alcançar dispositivos distantes são essenciais.

O módulo ESP32, criado pela empresa ESPRESSIF® (2015-2023), é uma plataforma de desenvolvimento da série ESP voltada para a criação de dispositivos conectados à Internet,

como sensores, controladores, dispositivos de monitoramento, sistemas de automação residencial e projetos de rastreamento. Ele pode utilizar as tecnologias WiFi e Bluetooth integradas no módulo para se comunicar entre si.

O kit TTGO da marca Lílygo® (2023) incorpora a tecnologia do módulo ESP32 com a comunicação via LoRa nas faixas de 868 e 915 MHz. No Brasil, o uso do LoRa é regulamentado pela Resolução N°680 de 27 de julho de 2017 da ANATEL, que dedicou a faixa de banda de 915 MHz para essa tecnologia.

O uso de RFID (Identificação por Radiofrequência) abrange uma ampla gama de aplicações em diversos setores. Finkenzeller (2010), aborda as diversas aplicações do RFID em setores como logística, gestão da cadeia de suprimentos, varejo, saúde e transporte. Ele destaca seu uso para rastreamento de ativos, gerenciamento de estoque, autenticação e coleta de dados, entre outros propósitos. Além disso, explora a integração do RFID com outras tecnologias, como cartões inteligentes sem contato e comunicação de campo próximo (NFC), analisando as possíveis sinergias e aplicações resultantes dessas combinações.

O Galpão de Laboratórios da UNIFESSPA, inaugurado em 6 de abril de 2018, segundo o portal Unifesspa (2018), é um espaço destinado ao desenvolvimento das atividades práticas dos cursos de Engenharia, Sistemas de Informação e Geologia. O local abriga 17 laboratórios de pesquisa operacional e um amplo espaço aberto para o desenvolvimento de projetos.

Com investimentos de quase 4 bilhões de reais, segundo o portal supracitado, cada laboratório é equipado com uma variedade de equipamentos específicos para as atividades e projetos relacionados aos cursos. O acesso a esses equipamentos é restrito a professores, monitores e alunos responsáveis por projetos coordenados por um professor da respectiva área. Essas instalações proporcionam um ambiente propício para a realização de pesquisas e projetos práticos, oferecendo aos alunos a oportunidade de aplicar o conhecimento adquirido em sala de aula e desenvolver habilidades técnicas relevantes para suas áreas de estudo.

Nesse ambiente particular, o atual mecanismo de acesso aos laboratórios operacionais do Galpão é realizado por meio de chaves e trancas convencionais, como demonstrado na Figura 01, que podem gerar transtornos e falhas tanto mecânicas (como chave danificada dentro da tranca) quanto de gerenciamento de acesso, visto que as chaves, além de serem limitadas, são guardadas com os coordenadores, alunos autorizados e agentes responsáveis pela limpeza e segurança desses ambientes.

Figura 01 - Fechadura Mecânica dos Laboratórios do Galpão.

Fonte: O Autor (2023).

No evento de extravio de uma dessas chaves, surge a necessidade de solicitar uma substituição, o que frequentemente resulta em uma cópia de qualidade inferior à original ou na substituição de todo o mecanismo de acesso. Além disso, a ausência de um registro de entrada e saída nesses ambientes cria a possibilidade de acessos não autorizados em momentos que não estão alinhados com o funcionamento do laboratório, colocando em risco os ativos de valor mantidos em seu interior.

Visando melhorar o acesso aos laboratórios, eliminando a necessidade de chaves convencionais e aprimorando a gestão e gerenciamento de acesso, além de reduzir a exposição dos ativos de valor e garantir sua segurança, o presente trabalho tem como objetivo a criação de um protótipo de tranca inteligente como solução para o modelo atual.

Este protótipo autentica usuários utilizando cartões de identificação por radiofrequência (RFID - Radio Frequency Identification) e senhas de acesso para permitir a entrada nos laboratórios e registrar esses acessos em um banco de dados. A comunicação entre os dispositivos da rede será realizada via LoRa, com intermediação dos Gateways para conexão com a Internet e o banco de dados. Além disso, o banco de dados poderá ser acessado por meio de uma API integrada a um sistema web simples, que gerenciará o acesso a esse ambiente laboratorial.

1.1 Justificativa

A proposta de implementação de um sistema embarcado com tecnologia LoRa para controlar o acesso aos laboratórios é uma solução viável e promissora para melhorar a gestão e a responsabilidade dos usuários que utilizam esses ambientes. Atualmente, não há um controle direto sobre quem entra ou sai dos laboratórios. Ao adotar o modelo proposto neste documento, será possível registrar e rastrear de forma precisa as entradas e saídas dos usuários autenticados, mantendo um histórico claro e confiável de suas atividades.

Com a utilização desse sistema, os coordenadores dos laboratórios terão acesso a um banco de dados que armazenará informações sobre as chaves de acesso e as pessoas autorizadas a entrar em cada ambiente. Essas informações serão atualizadas e registradas conforme os usuários acessarem os laboratórios. Assim, em caso de extravio de equipamentos ou componentes, será possível identificar de forma mais precisa quem estava presente no ambiente no momento do incidente, auxiliando na identificação do possível responsável.

No que tange ao quesito de responsabilidade para com o laboratório, a segurança é algo de extrema importância. A finalidade dos sistemas de informação na gestão do acesso de indivíduos a ambientes privados é assegurar uma experiência de acesso fácil e intuitiva para os usuários autorizados, ao mesmo tempo em que impede a entrada de usuários não autorizados (FARIA SANTOS *et al*., 2009).

Além disso, o sistema permitirá um maior controle sobre os horários de acesso aos laboratórios, uma vez que o coordenador terá registros dos momentos em que os usuários entram e saem dos ambientes, bem como a possibilidade de remover, modificar e liberar novos meios de acesso para outros usuários. Isso poderá ajudar a evitar o acesso não autorizado e possibilitar a identificação de eventuais irregularidades.

Segundo Menezes (2018), existem três fundamentos de segurança: a Confidencialidade (que garante o acesso à informação apenas ao usuário autorizado), a Integridade (que impede que a informação seja modificada sem autorização do usuário autorizado) e, por fim, a Disponibilidade (que garante que a informação esteja pronta para entrega no momento solicitado pelo usuário).

Assim, conforme o autor supracitado, é de suma importância a presença de um banco de dados, "pois torna a informação sempre disponível para o usuário ou software que tenha acesso ao sistema. Portanto, ao conceder a entrada ao indivíduo, é importante que seja registrado no banco de dados o identificador do usuário autenticado no sistema e a data/hora em que lhe foi permitido o acesso".

No geral, essa solução tem o potencial de melhorar significativamente a gestão dos laboratórios, proporcionando um maior controle de acesso, registro das atividades e responsabilização dos usuários. Isso contribuirá para um ambiente mais seguro e organizado, promovendo uma melhor utilização dos recursos e equipamentos disponíveis.

A Tranca Inteligente trará uma imensa facilidade de acesso, podendo armazenar acessos em sua memória e manter seu funcionamento mesmo que o servidor fique fora do ar por algum instante. A aplicação web, para acompanhamento e gerenciamento de acesso, e o banco de dados proporcionarão muitos benefícios para os coordenadores dos laboratórios e para a entidade pública, tais como:

- Comodidade: Como o sistema de gestão será realizado de forma online, o coordenador e a entidade poderão acessar a sala e ter conhecimento sobre quem a está acessando, em que momentos e ter controle para liberar ou não o acesso do local que desejarem, sem a necessidade de se deslocar até o local físico.
- Conveniência: Os usuários terão a capacidade de acessar o sistema através de um navegador em qualquer dispositivo digital de sua escolha, como notebooks, desktops e smartphones, desde que estejam conectados à internet. Isso proporcionará maior praticidade na organização e consulta dos registros de acesso aos laboratórios de maneira automatizada, aproveitando a integração com um banco de dados centralizado.
- Gestão: O administrador e o gerente terão a capacidade de acessar o banco de dados da aplicação e conduzir consultas automatizadas. Essa funcionalidade permite a realização de pesquisas destinadas a auxiliá-los na localização das informações desejadas.
- Produtividade: Uma solução mais conveniente para administrar os acessos de qualquer local com conectividade à internet, possibilitando o acompanhamento dos registros de usuários que acessam as instalações laboratoriais.

1.2 Objetivo Geral

Desenvolver um sistema de fechadura inteligente que valide entradas utilizando etiquetas de RFID (Radio-Frequency Identification) ou senhas de acesso, enquanto simultaneamente registra as informações de entrada e saída em um banco de dados por meio de uma API (Interface de Programação de Aplicativos). Esse banco de dados será hospedado

em um servidor central, onde um sistema web será responsável pelo gerenciamento dos acessos e pela consulta das informações armazenadas.

1.3 Objetivos específicos

O Sistema Integrado de Gerência de Acesso e Ativos de Laboratórios Operacionais (SIGA-LO) terá os seguintes objetivos:

- Criar Mecanismo de Liberação de Acesso:
- Desenvolver um mecanismo que controle uma tranca eletrônica e libere a entrada apenas para usuários autenticados via senha de acesso ou cartão RFID.
- Armazenar e Atualizar Dados de Acesso:
- Implementar a capacidade do mecanismo de liberação de acesso para armazenar usuários e atualizar dados de acesso, comunicando-se via LoRa com um banco de dados.
- Criar um banco de dados para o cadastro de prédios, salas, usuários, trancas, tags/códigos de acesso aos laboratórios e seus respectivos responsáveis.
- Registrar entradas e saídas com data, hora e usuário que fez a liberação.
- Criar API com Django:
- Desenvolver uma API usando a framework Django para intermediar e realizar requisições de acesso e manipulação do banco de dados no servidor.
- Possibilitar o consumo dessas informações em uma plataforma web de gerenciamento.
- Implementar Servidor com Django e Docker:
- Configurar um servidor em linguagem Python 3 utilizando a framework Django que consiga ser gerenciado por um container em Docker, permitindo desenvolver o sistema web, API e banco de dados em um mesmo ambiente.
- Desenvolver Sistema Web de Gerenciamento:
- Criar um sistema web simples que permita o acesso via navegador para gerenciamento remoto dos recursos da tranca e dos registros gerados a partir da liberação de acesso.

2 – TRABALHOS RELACIONADOS

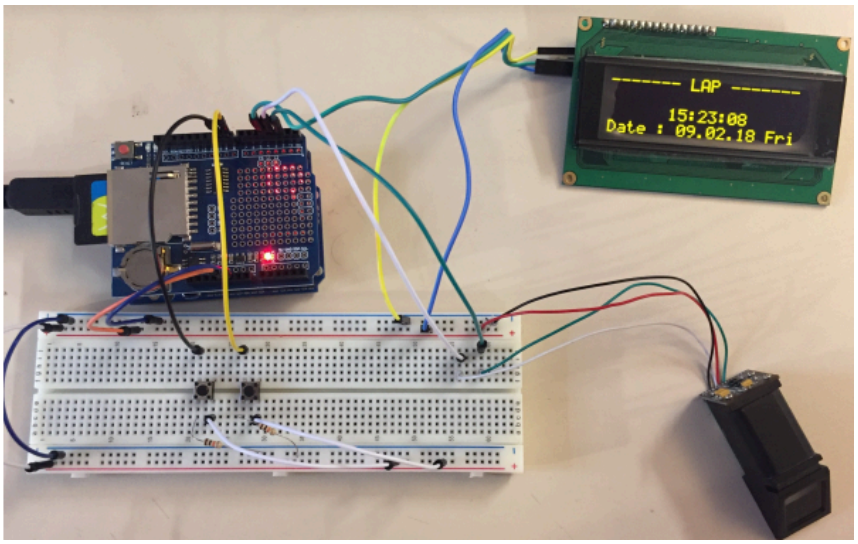
Neste capítulo serão mostrados trabalhos que influenciaram e direcionaram a criação desta ideia, que implementaram soluções semelhantes ao tema apresentado neste projeto. Estes trabalhos serão de suma importância para o desenvolvimento deste trabalho visto que já direcionam algumas formas de arquiteturas que podem ser adotadas, bem como o que pode ser feito diferente e melhorado durante a concepção do protótipo.

2.1 – Implementação de um sistema para acesso pessoal ao Laboratório de Automação Predial do DECAT.

Menezes (2018), tem como objetivo continuar a monografia de Lopez (2014) sobre o uso de um sistema para monitoramento de acesso utilizando Arduino e leitura biométrica, fazendo adições como senha numérica, independência de um computador para fazer a autenticação e cadastro de usuários e uso de um data logger para controle desses indivíduos.

Aqui ele ressalta sobre alguns pontos de instalação de fechaduras eletrônicas que podem ser interligadas a um relé e ativadas por um controlador (“Arduino UNO”) que fará uma verificação em seu dispositivo de armazenamento de dados “data logger” e liberará o acesso, como mostra a Figura 02.

Figura 02 - Implementação do Sistema com liberação via biometria de digital.



Fonte - MENEZES, 2018

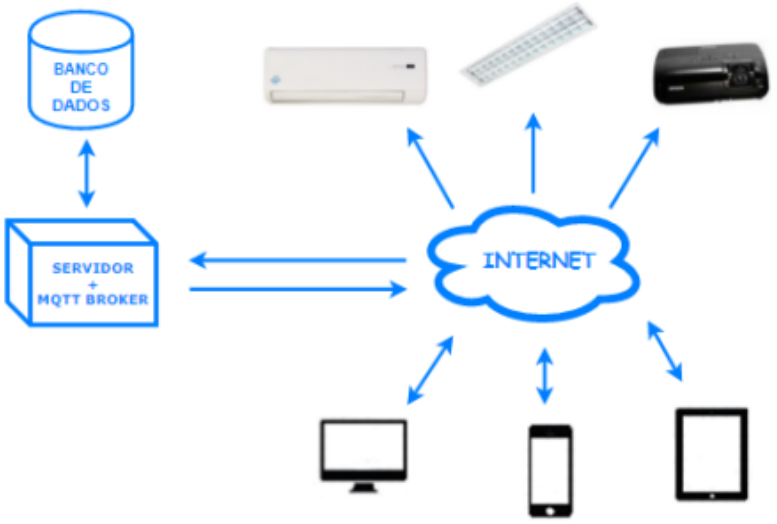
Tal trabalho, apesar de inovador, ainda apresenta a falta de integração com outros dispositivos, trabalhando sozinho e isolado. O autor indica, como sugestão a trabalhos futuros, a criação de um software que se comunique com o banco de dados do dispositivo, trazendo o conceito de IoT. Esta ideia sugerida possibilitou a concepção de trazer essa solução para um

Raspberry Pi 3 que fará a conexão com outros dispositivos de trancas inteligentes, além da conexão com a internet.

2.2 – Um Projeto De Sala De Aula Inteligente Para A Faesa Com O Uso Da Internet Das Coisas E MQTT.

Delfino e Dos Santos (2019), após uma conversa informal com funcionários da FAESA, Instituição de Ensino Superior na cidade de Vitória/SP, pode-se observar o mau uso dos equipamentos por alunos ao observar danos e irregularidades nos equipamentos. Dessa forma, foi idealizado um sistema de automação que utilizará dispositivos de monitoramento remoto, assim promovendo um melhor controle de recursos da infraestrutura de apoio didático em salas de aula da IES.

Figura 03 - Imagem da arquitetura criada pelo autor



Fonte - Delfino e Dos Santos, 2019

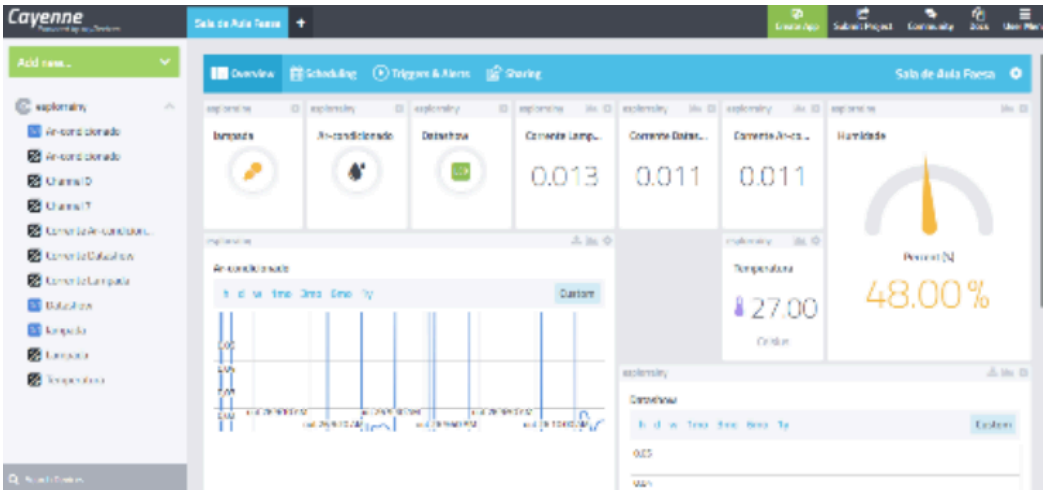
A arquitetura da Figura 03 mostra a utilização de um servidor somado a um *Broker MQTT* que farão controle e alerta de uso dos dispositivos da sala, sempre publicando os acontecimentos em uma aplicação web.

Para controlar os dispositivos inicialmente foi utilizado o módulo Xbee para realizar a comunicação com o servidor, mas devido ao fenômeno de distorção, estavam havendo atrasos de 8 segundos na entrega do sinal, onde na terceira versão de seu protótipo foi trocado pelo módulo NodeMCU, ou ESP8266, usando o WiFi (*Wireless Fidelity*) para comunicação. Dessa forma diminuindo drasticamente o tempo de envio de comando, sendo praticamente imediato o acionamento após a instrução e desnecessária a conexão a um roteador.

Todo o projeto foi posteriormente conectado à interface de comunicação *Cayenne MQTT* (Figura 04), plataforma IoT de arrastar e soltar, desenvolvida pela empresa My

Devices (2023), que permite aos usuários criar rapidamente protótipos e compartilhar suas soluções IoT conectadas.

Figura 04 - Uso do da plataforma Cayenne MQTT pelos autores.



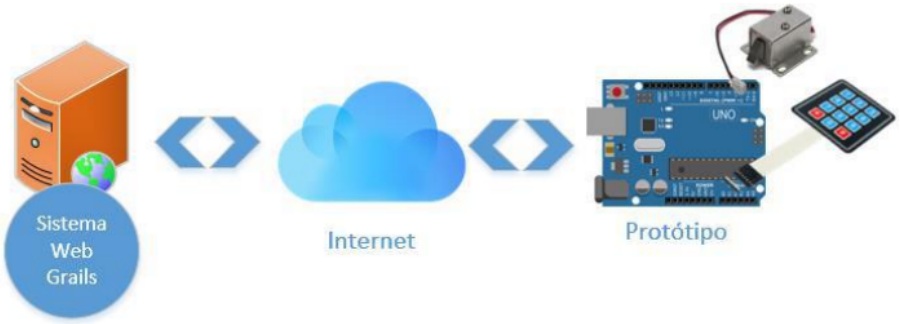
Fonte - Delfino e Dos Santos, 2019

No quesito acompanhamento dos ambientes, a plataforma é excelente visto que mostra dados dos sensores e emite alertas de acontecimentos aos usuários inscritos, Contudo, neste trabalho não foi desenvolvido um banco de dados persistente, assim, em caso de reinicialização do sistema os registros de eventos não podem ser armazenados. Além disso, vale ressaltar que a plataforma *Cayenne* foi descontinuada em Setembro de 2023, inviabilizando o desenvolvimento de novas tecnologias e do projeto em destaque.

2.3 – Desenvolvimento de um protótipo básico de hardware e software para segurança física e controle de acesso em ambientes institucionais que contenham ativos e informações de valor.

Este trabalho (*Lira et al.*, 2023) tem como objetivo criar uma solução que melhore a segurança de ambientes com itens de baixo custo, em que foi desenvolvido um sistema simples de tranca utilizando um arduino acoplado a uma placa de rede com entrada para cabos de ethernet RJ45 que se comunica via internet a um sistema na web que fará a autenticação da entrada, como mostra a Figura 05.

Figura 05 - Arquitetura adotada por Lira et al., 2023.



Fonte - *Lira et al*., 2023.

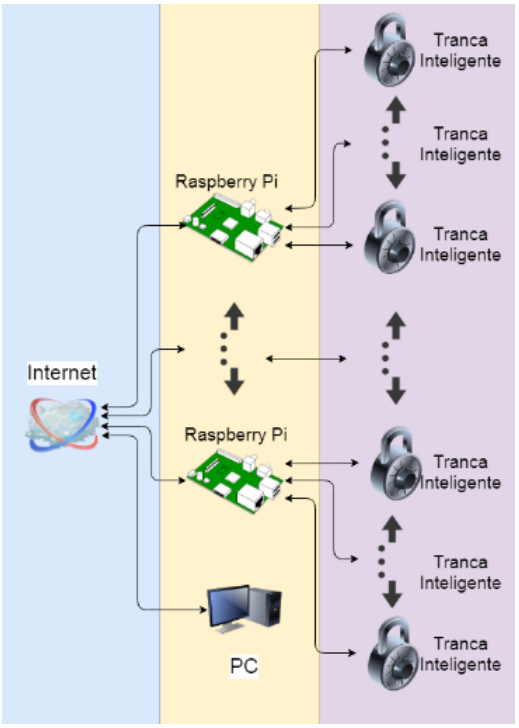
No contexto da arquitetura representada na Figura 05, atualmente, o sistema é projetado para controlar apenas uma tranca eletrônica. No entanto, os autores do trabalho reconhecem a necessidade de expandir esse sistema para gerenciar múltiplos dispositivos de acesso no futuro. Esta expansão é vista como um passo lógico para atender a ambientes com maiores requisitos de segurança e controle de acesso.

Não obstante, é importante notar que o sistema atualmente apresenta uma limitação crítica: sua total dependência de uma conexão de rede à internet para autenticar o acesso. Isso significa que, em situações de perda de conectividade com a internet, o acesso aos dispositivos protegidos se torna obstruído, resultando em potenciais interrupções indesejadas.

2.4 – Sistema inteligente para controle de acesso e monitoramento de múltiplos ambientes (*class control*).

Buscando melhorar a eficiência energética das centrais de ar e monitorar os múltiplos ambientes e salas de aula, Pereira (2019) desenvolveu um sistema de controle que ao liberar o acesso de um usuário autenticado fará controle das centrais de ar e outros dispositivos em sala. Tal sistema terá uma arquitetura funcional em névoa.

Figura 06 - Arquitetura proposta.



Fonte: Pereira, 2019.

A arquitetura apresentada na Figura 06 está baseada em ter um servidor em névoa que irá fazer a alimentação de dados diária de um conjunto de nós intermediários criados na plataforma Raspberry Pi, que servirá para armazenar dados referentes a acessos, registros e reservas de salas.

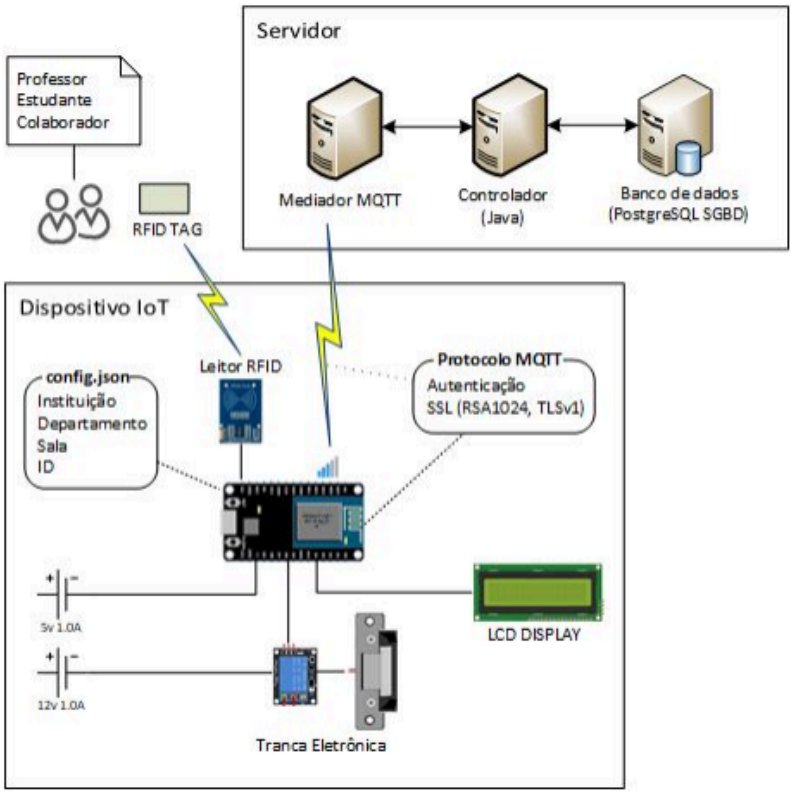
Na tranca eletrônica haverá um controlador modelo ESP8266, ou NodeMCU, que faz controle de todos os dispositivos da sala via controle de tensão via relés, onde ao ler uma tag de acesso no leitor RFID, irá mandar via WiFi ao Raspberry conectado e confrontar os dados com o banco de dados. Caso seja autenticado irá liberar acesso, caso contrário irá devolver usuário não cadastrado.

A aplicação realizada não permite autonomia dos controladores das trancas eletrônicas, criando dependência de verificação em outro dispositivo para autenticar acesso. Em caso de falha de comunicação ou sistema fora do ar, a tranca perde seu uso e o ambiente fica inacessível. É interessante que dados mínimos sejam armazenados nos controladores para gerar autonomia nas trancas e não depender de agente externo para autenticação.

2.5 – Desenvolvimento de Protótipo de um Sistema Embarcado de Fechadura Eletrônica para Controle de Homologação e Acesso

Trazendo o pressuposto da grande dificuldade gerada pelo uso de chaves convencionais, que não gera confiabilidade sobre um controle de acesso, Alessandro de Oliveira (2019) traz uma arquitetura que também está baseada em uso de um servidor MQTT. Da mesma forma, bem como em Delfino e Dos Santos (2019), com o diferencial de buscar persistência nos dados gerados, possibilitando consulta posterior à publicação de ato de acesso realizado, mantendo um histórico intacto de acontecimentos.

Figura 07 - Arquitetura de Alessandro de Oliveira.



Fonte: Alessandro de Oliveira (2019)

A arquitetura escolhida, apresentada na Figura 07, tem 3 subsistemas para controle, autenticação de usuários, emissão de alertas e criação de banco de dados persistente. No primeiro, os dispositivos IoT são programados em linguagem Wiring que controlam a tranca e se comunicam via WiFi com o servidor MQTT. O segundo subsistema, o servidor MQTT, trabalha como mediador de todo o sistema. O terceiro estágio seria o controlador java que fará o recebimento de todas as mensagens enviadas e armazenar no banco de dados, este o quarto subsistema, que fará persistência dos dados de usuário, permissões de acesso, histórico e outros.

Alguns diferenciais com o trabalho proposto neste documento são a linguagem dos controladores, em que foi adotada a C++ devido ao fato de toda a programação poder ser feita no ambiente integrado de desenvolvimento (IDE) “Arduino”, facilitando o desenvolvimento, e a não necessidade de uso de um servidor MQTT, podendo ser utilizada uma API diretamente para enviar e requisitar os dados ao sistema web.

Na maioria dos trabalhos é apresentado como solução para acesso a ambientes o uso de autenticação por biometria, com forte menção aos dispositivos de acesso RFID. Tal tecnologia é um facilitador para acessos rápidos onde há apenas a necessidade de ler a *tag* registrada em um cartão e realizar autenticação com o dado cadastrado. Por outro lado, senhas de acesso ainda são necessárias para, em caso de extravio do cartão, ainda possibilitar o acesso ao local desejado.

Abaixo, na Tabela I, está um comparativo entre este trabalho e os demais correlatos:

Quadro I - Quadro comparativo dos trabalhos relacionados e projeto proposto.

Trabalhos	Autenticação na tranca	LoRa	Servidor Principal	Banco de Dados	API	Sistema Web
2.1	Sim	Não	Não	Não	Não	Não
2.2	Não	Não	Não	Não	Não	Sim
2.3	Não	Não	Sim	Sim	Sim	Sim
2.4	Não	Não	Sim	Sim	Sim	Sim
2.5	Sim	Não	Sim	Sim	Não	Não
Este Trabalho	Sim	Sim	Sim	Sim	Sim	Sim

Outro diferencial do SIGA-LO é sua não necessidade de dependência total à internet para estar operacional, sendo adotado o modelo de comunicação de longo alcance (LoRa), integrado ao controlador ESP 32, consumindo menos energia e mantendo comunicação confiável, mesmo à nível de ruído com o Gateway que possuirá um pequeno banco de dados com dados previamente coletados da última atualização de acessos. Os demais sistemas terão necessidade de conexão à internet, contudo não irão gerar interferência ou falha no funcionamento dos controladores.

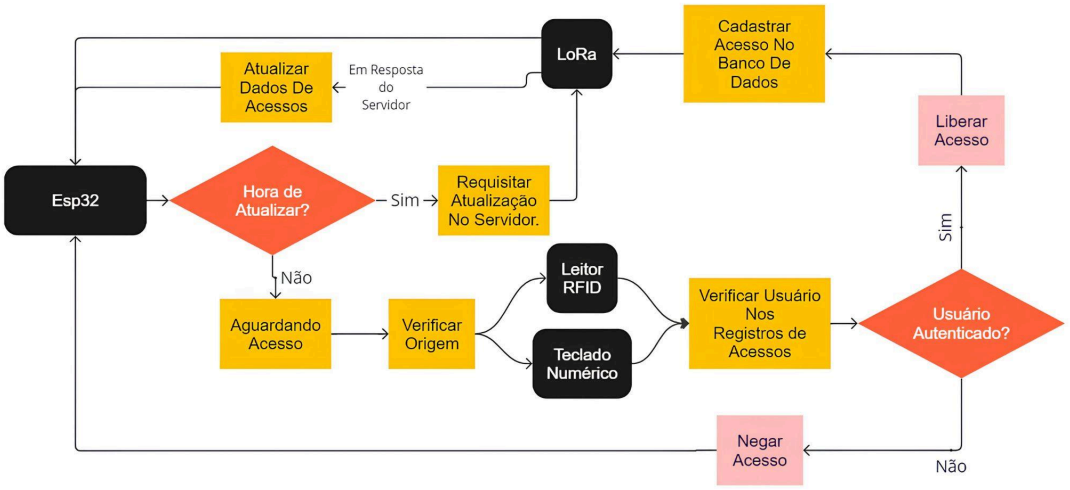
3 – METODOLOGIA

A tranca inteligente será responsável pelo controle de acesso aos laboratórios, autenticando usuários através de cartões RFID e senhas, registrando suas entradas e saídas. Esses registros serão enviados ao gateway via comunicação LoRa. O gateway, por sua vez, além de transmitir essas informações ao servidor central, também buscará dados de acesso atualizados do servidor para manter a tranca sincronizada com as permissões mais recentes. O servidor armazenará todos os dados de acesso e fornecerá as informações necessárias para o gateway. Além disso, será disponibilizado uma interface web para a gestão dos acessos, permitindo aos coordenadores dos laboratórios monitorar e gerenciar as entradas e saídas dos usuários de forma eficiente.

Nesta seção, será detalhado o processo de desenvolvimento do Sistema Integrado de Gerência de Acesso e Ativos de Laboratórios Operacionais (SIGA-LO), apresentando os componentes utilizados, os processos de desenvolvimento e implementação de cada parte do sistema, bem como a integração desses componentes para garantir um funcionamento coeso e eficiente do sistema. A metodologia está dividida em quatro partes principais: a tranca inteligente, o gateway, o sistema web e a integração dos componentes.

3.1 Tranca Inteligente

Figura 08 - foto do esquemático da tranca



Fonte: O Autor (2024).

A tranca inteligente tem como principal objetivo realizar a autenticação do acesso ao laboratório, armazenar dados de acesso de usuários, controlar uma tranca eletrônica e registrar esse acesso para enviá-lo ao banco de dados, como mostrado na Figura 08.

3.1.1 Descrição do Hardware Utilizado

A tranca inteligente é constituída pelos seguintes componentes de hardware descritos na Tabela II.

Quadro II - Componentes de Hardware da Tranca

Componente	Descrição	Modelo
Módulo ESP32	Microcontrolador principal	LoRa32 V1.0
Módulo LoRa	Comunicação de longa distância	SX1276
Leitor RFID	Leitor de cartões e chaveiros RFID de frequência 13,56 MHz	RC522 (13,56 MHz)
Teclado Matricial 4x4 (16 teclas)	Inserção de senhas	4x4 Keypad
Tranca Eletrônica	Abertura e fechamento da porta de acesso ao laboratório	Solenoide
Relé	Envio de sinal de abertura e fechamento da Tranca Eletrônica	5V DC
Expansor 8 bits	Expansão de portas para inclusão do teclado matricial.	PCF8574

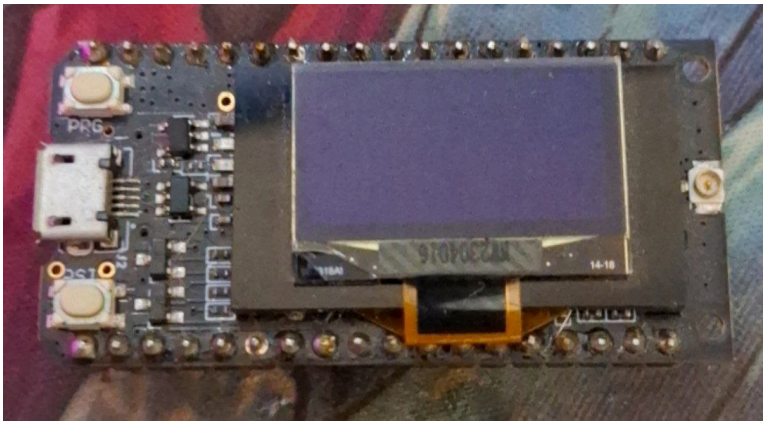
3.1.2 Especificações técnicas e funcionalidades.

Cada componente da tranca inteligente possui especificações técnicas e funcionalidades gerais e específicas.

Módulo ESP 32

O ESP 32 (Figura 09) é um microcontrolador com conectividade WiFi e Bluetooth integrados, além de várias portas GPIO para comunicação com outros componentes. Ele possui um processador dual-core de 32 bits, 520 KB de SRAM e suporte para múltiplas interfaces de comunicação.

Figura 09 - ESP 32



Fonte: O Autor (2024)

Objetivo Geral: Prover conectividade entre os componentes e realizar o processamento de dados.

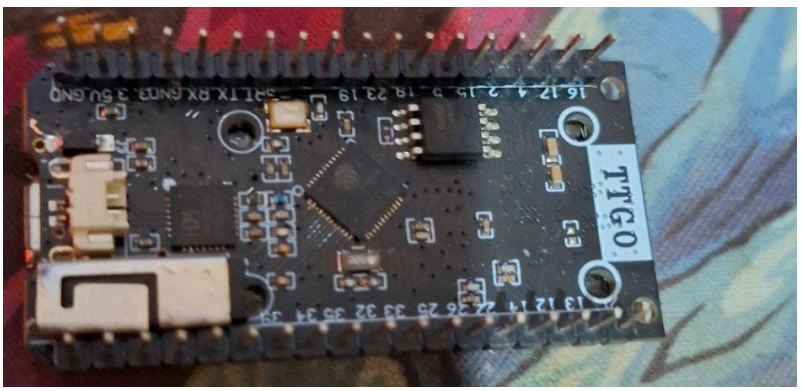
Objetivos Específicos:

- Controlar a tranca eletrônica;
- Armazenar dados de acesso dos usuários homologados;
- Processar entradas de acesso via teclado numérico e RFID, decidindo sobre a concessão de acesso;
- Enviar registros de acesso ao gateway.

Módulo LoRa SX1276

Este módulo () opera nas frequências de 868 MHz e 915 MHz, com alcance de até 15 km em ambientes abertos. Sua taxa de transmissão de dados é baixa, o que garante economia de energia.

Figura 10 - Módulo LoRa SX1276 e ESP 32 no Kit TTGo LoRa32 V1.0



Fonte: O Autor (2024)

Objetivo Geral: Prover comunicação de longa distância com baixo consumo de energia entre o ESP 32 e o gateway.

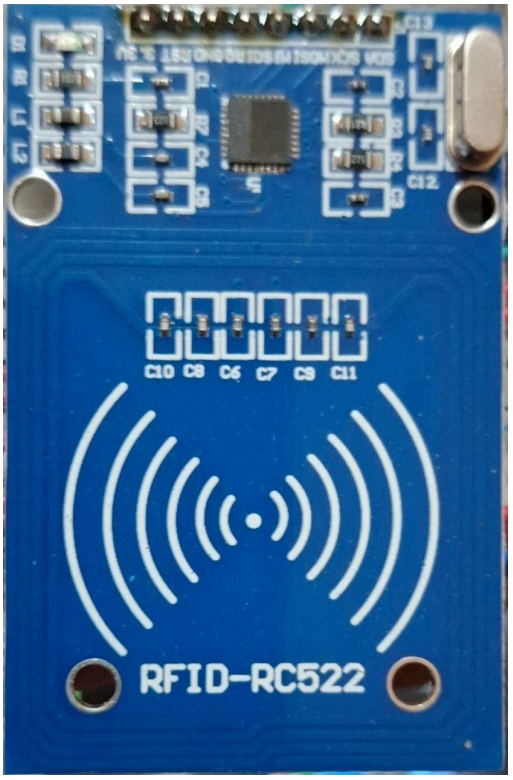
Objetivos Específicos:

- Comunicar-se diretamente com o gateway;
- Receber dados de acesso;
- Enviar registros de acesso ao gateway.

Módulo RFID - MFRC522

Opera na frequência de 13,56 MHz e é compatível com cartões e etiquetas padrão ISO/IEC 14443 (MIFARE, 2021).

Figura 11 - Módulo RFID - MFRC522



Fonte: O Autor (2024)

Objetivo Geral: Autenticar usuários utilizando cartões e chaveiros RFID com frequência de 13,56 MHz.

Teclado Matricial 4x4 (16 teclas)

Este teclado possui uma matriz de 4x4, permitindo a entrada de até 16 caracteres.

Objetivo Geral: Autenticar usuários via senhas alfanuméricas.

Tranca Eletrônica Solenóide

A tranca eletrônica solenoide (Figura 12) funciona de forma normalmente fechada e é ativada por um solenoide. Ela também possui um sensor de presença que garante o fechamento correto.

Figura 12 - Tranca Eletrônica

Fonte: AliExpress, disponível em <https://pt.aliexpress.com/i/33019924390.html>, Acesso em 12/02/2024

Objetivo Geral: Controlar a abertura e fechamento da porta de acesso ao laboratório.

Relé 5V DC

Dispositivo eletromecânico utilizado para comutação.

Objetivo Geral: Enviar sinais para abertura e fechamento da tranca.

Expansor 8 bits PCF8574

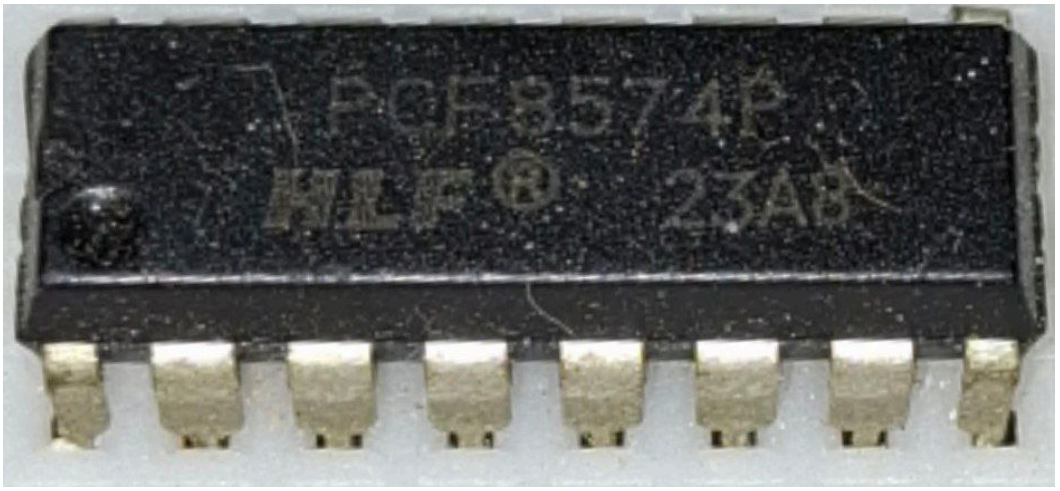
Este circuito integrado (Figura 13) expande as portas do ESP 32 através de uma interface I2C de 8 bits, permitindo o controle de até 8 portas com apenas dois pinos.

Objetivo Geral: Aumentar a eficiência do ESP 32 ao reduzir o número de portas utilizadas.

Objetivos Específicos:

- Receber informações do teclado numérico;
- Processar os impulsos gerados após a digitação de uma tecla no teclado matricial.

Figura 13 - Expansor 8 bits PCF8574



Fonte: O Autor (2024)

3.1.3 Processo de desenvolvimento e implementação.

O desenvolvimento da tranca inteligente envolveu a criação e implementação de diversas rotinas de verificação, cada uma com funções específicas para assegurar o funcionamento eficiente e seguro do sistema. A seguir, detalhamos essas rotinas e seu papel no sistema:

- **Rotina de Análise de Teclado Matricial**: Esta rotina monitora constantemente o teclado matricial para detectar qualquer entrada do usuário. Quando uma tecla é pressionada, a rotina verifica a entrada, processa o comando correspondente e atualiza a tela OLED para fornecer feedback visual ao usuário. Isso é essencial para permitir a interação do usuário com o sistema, especialmente durante a inserção de senhas.
- **Rotina de Análise de Pacotes LoRa**: A comunicação entre a tranca e o gateway é realizada via tecnologia LoRa. Esta rotina verifica se há pacotes de dados recebidos

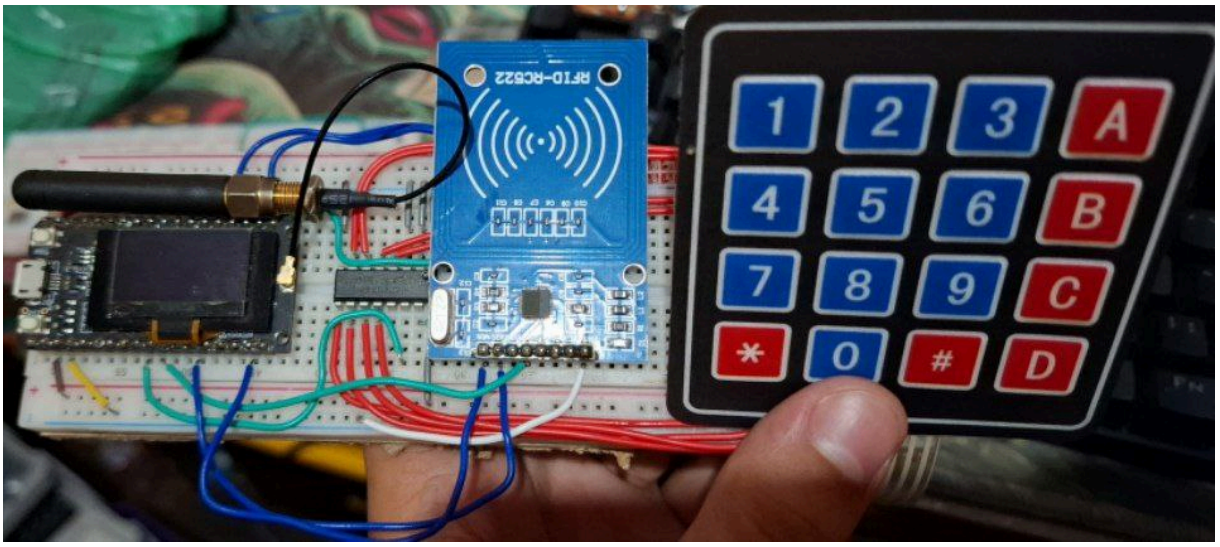
através do módulo LoRa. Quando um pacote é detectado, ele é processado para determinar se contém comandos ou atualizações relevantes para o sistema.

- **Subrotina de Verificação de Pacote LoRa**: Após a recepção de um pacote LoRa, esta subrotina avalia se o pacote é destinado ao dispositivo específico e se sua origem é o gateway autorizado. Isso garante que apenas comandos legítimos sejam executados, evitando interferências ou comandos maliciosos.
- **Rotina de Verificação do Teclado 4x4 de Membrana**: Esta rotina monitora o teclado 4x4 de membrana para detectar tentativas de acesso. Quando o usuário digita uma senha, a rotina verifica a autenticidade da entrada, comparando-a com as credenciais armazenadas na memória da tranca.
- **Rotina de Verificação de Tag RFID**: A autenticação por RFID é uma característica chave do sistema. Esta rotina verifica continuamente a presença de tags RFID próximas ao leitor. Quando uma tag é detectada, seus dados são lidos e comparados com os registros de usuários autorizados.
- **Rotina de Verificação do Gateway**: Para manter a sincronização de dados e garantir atualizações regulares, esta rotina verifica se o gateway está online. Além disso, ela gerencia os intervalos de tempo para a atualização das informações de acesso armazenadas na memória da tranca, garantindo que novos dados de usuários sejam incorporados sem interrupções.
- **Rotina de Análise do Botão de Abertura Interna**: Para permitir a saída segura do laboratório, esta rotina monitora o botão de abertura interna. Quando o botão é pressionado, a tranca é liberada, permitindo a saída dos usuários. Esta função é crucial para a segurança, garantindo que os usuários possam sair facilmente em caso de emergência.

Após o desenvolvimento das rotinas de verificação e controle, o código foi compilado e carregado nos módulos ESP32. A montagem do circuito foi realizada em uma protoboard, conectando todos os componentes de hardware necessários, incluindo sensores, atuadores, módulo LoRa, leitor RFID, teclado matricial, e tela OLED. Este arranjo inicial permitiu a integração dos componentes e a validação do funcionamento do sistema em um ambiente de teste.

A Figura 14 ilustra a integração dos componentes, demonstrando a disposição dos elementos na protoboard e suas conexões.

Figura 14 - Integração de Componentes



Fonte:O Autor (2024).

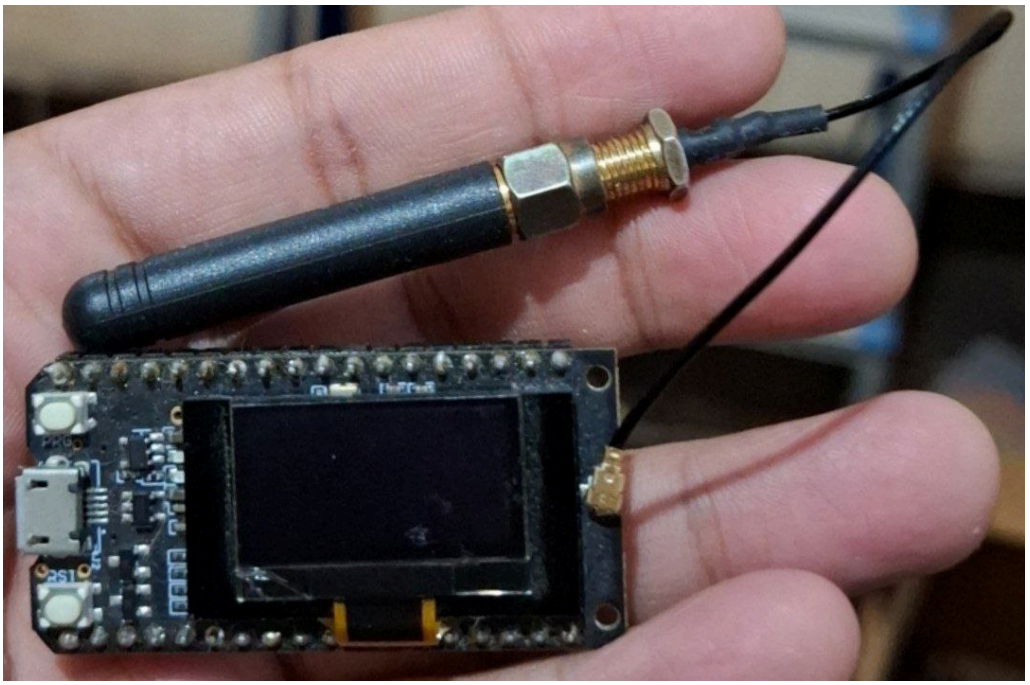
A partir dessa montagem, foram realizados testes funcionais para verificar a precisão das leituras de RFID, a resposta do teclado matricial, a recepção de pacotes LoRa e a atuação da tranca eletrônica. Os ajustes necessários foram feitos para otimizar a comunicação entre os componentes e garantir a estabilidade do sistema.

Essas rotinas, descritas detalhadamente no Anexo 01 - Tranca.ino, formam a base do funcionamento da tranca inteligente, proporcionando um sistema robusto e eficiente para o controle de acesso aos laboratórios.

3.2 Gateway

O gateway (Figura 10) desempenha um papel crucial como intermediário entre a tranca inteligente e o sistema web. Sua principal função é garantir a comunicação eficiente e segura entre a tranca e o servidor, facilitando a atualização dos dados de acesso e o envio de informações de registro.

Figura 10 - Gateway



Fonte: O Autor (2024).

O gateway atua como uma ponte, permitindo que a tranca inteligente se conecte a uma rede local e se comunique com o servidor web através de uma API. O gateway é responsável por dois aspectos principais:

- Atualização de Dados de Acesso: O gateway recebe as atualizações de dados de autenticação e acesso da tranca inteligente utilizando a comunicação via LoRa. Isso inclui informações sobre novos usuários, alterações em permissões de acesso, e atualizações de registros de acesso. Essas informações são então enviadas para o servidor, garantindo que o banco de dados esteja sempre atualizado com os dados mais recentes.
- Envio de Dados de Registro ao Servidor: A tranca inteligente registra cada evento de acesso (entrada e saída) que ocorre. O gateway coleta esses dados de registro e os envia ao servidor web. Isso permite que o sistema web tenha um registro completo e atualizado das atividades nos laboratórios.

Para a implementação do gateway, foi utilizado um kit ESP32/LoRa da Lilygo® (2023). Este kit combina as funcionalidades do módulo ESP32 com a capacidade de comunicação LoRa, proporcionando uma solução robusta para as necessidades de comunicação do sistema. Na Tabela IV pode ser visto a configuração escolhida para o funcionamento da comunicação LoRa

Quadro III - Configuração da Comunicação Utilizando LoRa

Configuração	Ajuste
Faixa de Frequência	928 MHz
NSS	Porta 18
MISO	Porta 19
MOSI	Porta 27
SCK	Porta 05
DIO0	Porta 26
RESET	Porta 14
Ganho Antena	2 dBi
Fator de Espalhamento (*Spread Factor*)	7

Sobre a comunicação com a rede de internet local, o ESP32 consegue realizar a conexão a uma Rede WPA2/Enterprise por meio do código no Anexo 02, que não utiliza um certificado necessário para conectividade mas que possui um formato de autenticação de usuário a rede interna da UNIFESSPA.

Com a combinação desses componentes, o gateway consegue se conectar a uma rede local e realizar requisições ao servidor web através da API. Isso possibilita o gerenciamento remoto das trancas e a sincronização dos dados de acesso em tempo real. Com o gateway corretamente configurado e integrado, o sistema pode operar de maneira coordenada, garantindo que os dados de acesso sejam continuamente atualizados e gerenciados de forma centralizada.

3.3 Sistema Web

O Sistema Web foi desenvolvido utilizando a tecnologia de conteinerização Docker (2013-2024), que permite que aplicações sejam criadas e executadas em qualquer sistema operacional, incluindo Linux, Windows e Mac. A conteinerização utiliza o kernel do Linux e suas funcionalidades, como cGroups e namespaces, para segregar processos, o que facilita a implementação e a execução de aplicações em um ambiente isolado e consistente.

A utilização de Docker (2013-2024) permitiu a criação de containers, que são unidades de software que empacotam o código e todas as suas dependências, garantindo que a aplicação funcione de maneira idêntica em qualquer ambiente. Isso é especialmente útil para o desenvolvimento de aplicações que precisam ser robustas e escaláveis, além de simplificar a distribuição e a implantação do software.

Para o desenvolvimento do Sistema Web, foi utilizada a Framework Django (2005-2024). Segundo Pree (1995), uma framework é um conjunto consolidado de códigos compartilhados entre diversos projetos de software, fornecendo uma funcionalidade genérica que pode ser reutilizada. A Django (2005-2024), especificamente, facilita a criação rápida de aplicativos robustos e escaláveis, com funcionalidades integradas e foco na reutilização de código. Com Django, foi possível desenvolver tanto o Processo Interno (*Back-End*) quanto a Interface de Usuário (*Front-End*) da aplicação, de forma eficiente e eficaz.

Com a tecnologia de conteinerização, também foi possível criar um banco de dados usando PostgreSQL (1996-2024), um sistema gerenciador de banco de dados objeto-relacional desenvolvido como um projeto de código aberto. O PostgreSQL (1996-2024) é conhecido por sua robustez e capacidade de lidar com grandes volumes de dados, sendo uma escolha ideal para o gerenciamento das informações de acesso e autenticação dos usuários dos laboratórios.

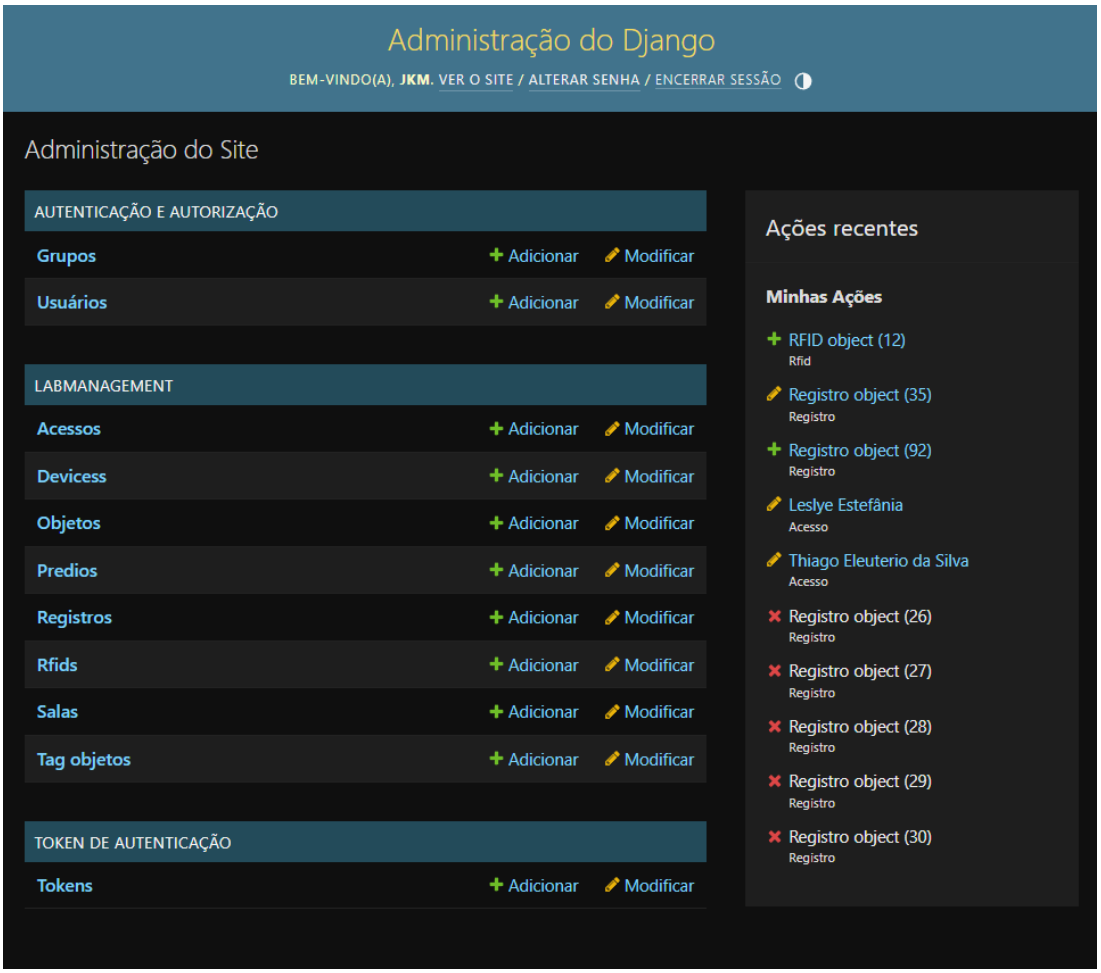
O sistema web desenvolvido possui várias funcionalidades que permitem a gestão eficiente dos laboratórios, incluindo:

- Cadastro de Alunos e Usuários: Permite o registro e a gestão dos dados de alunos e outros usuários que terão acesso aos laboratórios, incluindo a atribuição de credenciais de acesso.
- Gestão de Salas e Laboratórios: Facilita o gerenciamento das diferentes salas e laboratórios.
- Registro e Controle de Acessos: Mantém um registro detalhado de todas as entradas e saídas, permitindo a consulta e o monitoramento em tempo real.

- Interface de Usuário Amigável: Desenvolvida com foco na usabilidade, garantindo que os coordenadores e administradores possam utilizar o sistema de maneira intuitiva e eficiente.

O sistema web (Figura 11), desenvolvido e gerido através de containers Docker, integra-se de maneira eficaz com o banco de dados PostgreSQL e a Framework Django. Essa integração permite uma gestão centralizada e eficiente dos acessos aos laboratórios, oferecendo um sistema robusto e escalável que atende às necessidades da instituição de ensino.

Figura 11 - Interface do Sistema Web



Fonte - O Autor

Utilizando a tecnologia de containers Docker, é possível assegurar que o ambiente de desenvolvimento seja idêntico ao ambiente de produção, eliminando problemas de compatibilidade e garantindo a consistência na execução do software. A Framework Django, com sua arquitetura baseada em módulos reutilizáveis e uma estrutura robusta para o desenvolvimento rápido de aplicações web, fornece uma base sólida para a construção de um sistema escalável e fácil de manter.

O banco de dados PostgreSQL, por sua vez, proporciona um sistema gerenciador de banco de dados objeto-relacional confiável e altamente eficiente. Ele é capaz de lidar com grandes volumes de dados e realizar consultas complexas com rapidez e precisão, características essenciais para a gestão das informações de acesso e autenticação dos usuários dos laboratórios.

A integração harmoniosa entre Docker, Django e PostgreSQL resulta em um sistema web que centraliza todas as operações de gestão de acesso, permitindo aos administradores e coordenadores dos laboratórios monitorar e controlar as atividades de maneira simples e intuitiva. A robustez e escalabilidade dessa solução garantem que ela possa crescer junto com as necessidades da instituição, adaptando-se a novas demandas e desafios sem comprometer a qualidade e a segurança do serviço.

Em suma, o sistema web desenvolvido com Docker, Django e PostgreSQL representa uma solução completa para a gestão de acessos e administração dos laboratórios. Ele proporciona um ambiente seguro, eficiente e prático, beneficiando todos os usuários envolvidos e contribuindo para a melhoria contínua dos processos operacionais da instituição.

3.4 Integração dos Componentes

A integração dos componentes do Sistema Integrado de Gerência de Acesso e Ativos de Laboratórios Operacionais (SIGA-LO) é essencial para seu funcionamento uníssono e eficiente. Embora a tranca inteligente e o sistema web possam operar de maneira independente, a interdependência entre os componentes garante uma gestão de acesso robusta e atualizada.

O funcionamento do sistema inicia-se com o gateway. Quando ativado, o gateway solicita ao sistema web uma lista dos dispositivos autorizados a se conectar e a requisitar informações. Esta etapa inicial é crucial para assegurar que apenas dispositivos autenticados possam interagir com o sistema. Após receber as informações, o gateway mantém uma rotina de auto-atualização a cada cinco segundos para verificar continuamente quais dispositivos estão autorizados, além de monitorar a conexão com a rede Wi-Fi, reconectando-se automaticamente em caso de desconexão.

A tranca inteligente, ao ser ativada, solicita imediatamente ao gateway as credenciais dos usuários autorizados a acessar o laboratório. Este processo inicial garante que a tranca possua as informações mais atualizadas. Para manter a precisão e a redundância das

informações, a tranca repete essa solicitação ao gateway a cada minuto, atualizando constantemente suas credenciais armazenadas.

Com ambos os dispositivos – tranca e gateway – operacionais e atualizados, o sistema está preparado para gerenciar o acesso dos usuários aos laboratórios. Quando um usuário tenta acessar o ambiente, utilizando RFID ou senha, a tranca verifica em sua memória se as credenciais fornecidas estão autorizadas. Se a verificação for bem-sucedida, a tranca libera o acesso e registra o evento no gateway. O gateway então publica esse registro no sistema web, permitindo que gestores e administradores visualizem as informações em tempo real.

Este fluxo de comunicação e redundância entre a tranca, o gateway e o sistema web garante um controle de acesso seguro e bem gerido. A constante troca de informações e a verificação periódica asseguram que os dados de acesso sejam precisos e atualizados, minimizando riscos de segurança e melhorando a eficiência da gestão dos laboratórios operacionais.

4.RESULTADOS

4.1 Avaliação da Tranca Inteligente

A seguir, são descritos os resultados obtidos a partir dos testes realizados com a tranca inteligente:

- **Tempo de Resposta:** O tempo necessário para a tranca validar as credenciais e liberar o acesso foi de aproximadamente 200 ms, enquanto o tempo médio para registrar o evento no sistema web foi de cerca de 2 segundos, variando conforme a estabilidade da rede LoRa e Wi-Fi.
- **Confiabilidade:** Em 100 tentativas de acesso em períodos de 10 segundos entre cada, a tranca liberou corretamente o acesso em todas as situações, entretanto, apenas 80 dessas interações foram completamente registradas no banco de dados, indicando uma falha de 20% no registro dos acessos.
- **Armazenamento e Redundância:** Em um período de 60 minutos, a tranca conseguiu armazenar 57 de 60 atualizações realizadas. As falhas ocorreram devido à perda de pacotes na comunicação via LoRa ou ao travamento do gateway.
- **Entrada de Dados:** A entrada de dados por meio de Tags RFID foi 100% precisa, sem erros de verificação ou reconhecimento. No entanto, o teclado matricial apresentou falhas, devido tanto ao desgaste físico quanto a um mau contato com o expansor de 8 bits, que ocasionalmente falha ao registrar os pressionamentos. Durante os testes, tanto as entradas via RFID quanto via teclado matricial (50 tentativas cada) resultaram em um total de 100 iterações de acesso.

4.2 Desempenho do Gateway

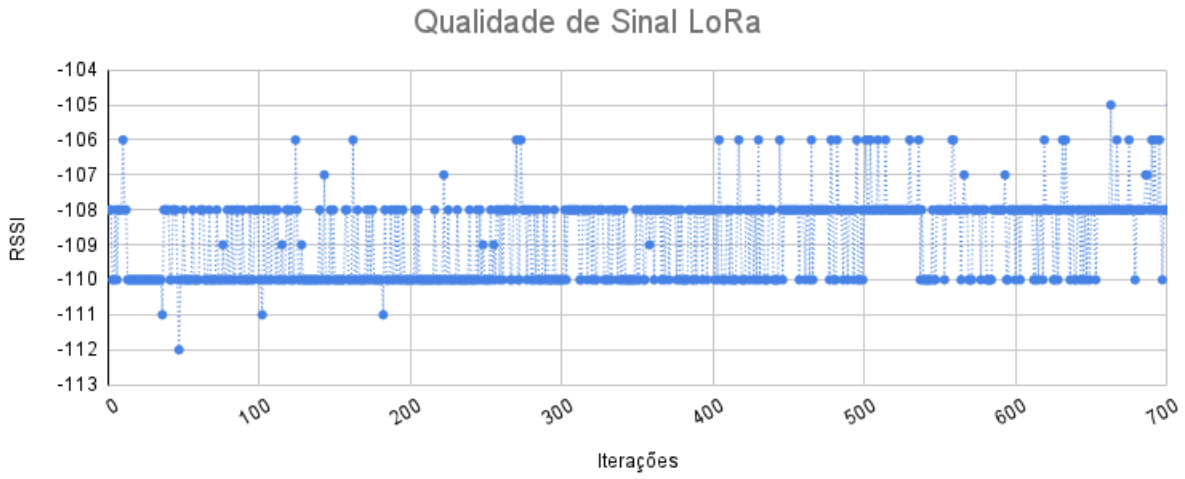
Os resultados referentes ao desempenho do gateway incluem:

- **Latência:** O tempo médio de comunicação entre o gateway e a tranca foi de aproximadamente 1 segundo, enquanto a comunicação com o sistema web levou cerca de 0,5 segundos, dependendo da estabilidade da conexão de internet.
- **Confiabilidade da Conexão:** O gateway manteve uma comunicação ativa e funcional com a tranca e o sistema web, exceto em momentos de falhas de conexão com a internet ou travamentos, que exigiram reinicializações manuais em alguns casos. O

gateway também apresentou reinicializações espontâneas causadas por falhas de energia (*brownout*).

- **Capacidade de Processamento:** Apesar de lidar de maneira eficaz com o fluxo de dados, o gateway apresentou reinicializações durante picos de tentativas de acesso, afetando seu desempenho. Ainda assim, ele conseguiu fornecer à tranca 57 das 60 atualizações solicitadas e registrar 80 dos 100 acessos no sistema web.
- **Qualidade do Sinal LoRa:** A comunicação entre a tranca e o gateway apresentou um Indicação de Força do Sinal Recebido (RSSI - *Received Signal Strength Indication*) médio entre -108 e -110, o que, apesar de fraco, manteve-se dentro dos parâmetros esperados para LoRa, que pode operar com sinais tão baixos quanto -120. Em 700 iterações de teste, o RSSI variou entre -112 e -105, com uma média estável entre -108 e -110. Tais resultados podem ser observados na Figura 12 que retrata a qualidade do sinal para cada iteração realizada.

Figura 12 - Qualidade de sinal LoRa.



Fonte: Autor (2024).

4.3 Integração dos Componentes

Os componentes do sistema SIGA-LO demonstraram uma integração eficaz, conforme os seguintes pontos:

- **Sincronização de Dados**: A comunicação entre a tranca, o gateway e o sistema web foi avaliada pela quantidade de acessos registrados e atualizações realizadas. De 100

tentativas de acesso, 80 foram registradas com sucesso no sistema web, enquanto 57 das 60 atualizações de credenciais foram bem-sucedidas. O índice geral de sincronização foi de 85,63%, considerando um total de 137 sincronizações de um possível total de 160.

- **Redundância e Resiliência**: Em casos de falha na conexão de internet, o gateway conseguiu manter os dados da última atualização da tranca localmente, garantindo continuidade no processo de autorização de acessos. Quando o gateway enfrentou falhas de comunicação, a tranca continuou a operar com as informações previamente armazenadas. Contudo, em caso de desligamento da tranca, os dados armazenados são perdidos, o que impede novos acessos até que o gateway reconecte e atualize as credenciais.

4.4 Desafios e Soluções Encontradas Durante a Integração

Embora funcional, o sistema enfrentou vários desafios durante o processo de integração:

- **Persistência de Dados no Gateway e Tranca**: Ambos os dispositivos armazenam dados temporariamente na memória volátil, resultando em perda de informações em caso de reinicialização ou falha de energia. Esse problema foi particularmente frequente no gateway.
- **Interface do Sistema Web**: A interface do sistema web ainda necessita de melhorias para oferecer uma experiência de usuário mais amigável.
- **Falhas no Teclado Matricial**: O teclado matricial apresentou problemas de contato e reconhecimento dos pressionamentos, o que comprometeu algumas tentativas de acesso.
- **Reinicializações Constantes do Gateway**: As reinicializações espontâneas do gateway indicam que o hardware utilizado pode não ser adequado para a carga de trabalho exigida.
- **Comunicação LoRa**: A comunicação entre o gateway e a tranca via LoRa apresentou boa estabilidade, sendo possível enviar, receber e interpretar mensagens de forma eficaz, mesmo com sinal de baixa intensidade.

5. CONCLUSÃO

Este trabalho apresentou a concepção, implementação e avaliação do Sistema Integrado de Gerência de Acesso e Ativos de Laboratórios Operacionais (SIGA-LO), com foco na integração entre uma tranca inteligente, um gateway e um sistema web. Os resultados obtidos demonstraram que o sistema conseguiu atingir os objetivos propostos, fornecendo uma solução funcional para o gerenciamento de acessos em laboratórios.

A tranca inteligente se mostrou eficaz ao validar credenciais e registrar acessos em um tempo adequado, com tempo de resposta de 200 ms e latência de até 2 segundos para registro no sistema web. Contudo, algumas limitações foram observadas, como falhas na persistência de dados no gateway e na tranca, além de instabilidades no teclado matricial utilizado para entrada de senhas.

O gateway desempenhou um papel central na comunicação entre os dispositivos, embora tenha apresentado travamentos em picos de carga e falhas de conectividade devido à sua qualidade de sinal WiFi e à dependência de uma conexão de internet estável. A troca de mensagens entre a tranca e o gateway via LoRa apresentou resultados satisfatórios, com um RSSI médio de -108 a -110 dBm, dentro dos parâmetros esperados.

A integração entre os componentes alcançou uma taxa de sincronização de 85,63%, com boas práticas de redundância para manter a funcionalidade em caso de falhas temporárias de conexão. Embora o sistema tenha atendido aos requisitos básicos de operação, ainda há oportunidades para aprimoramento.

5.1 Contribuições

O SIGA-LO contribui de forma significativa para a área de automação e gestão de acesso em ambientes laboratoriais, ao integrar tecnologias emergentes como LoRa e Docker, com sistemas robustos como Django e PostgreSQL. A modularidade do sistema permite que cada componente funcione de forma independente, ao mesmo tempo em que assegura uma comunicação integrada para garantir atualizações constantes e confiáveis das credenciais de acesso.

Além disso, o desenvolvimento de um sistema baseado em comunicação LoRa para ambientes com restrições de conectividade amplia as possibilidades de automação em locais onde outras tecnologias podem não ser viáveis. A proposta de redundância na troca de

informações e a persistência de dados mesmo em cenários de falhas adicionam robustez ao sistema, tornando-o uma opção viável para laboratórios que demandam alta confiabilidade e segurança no controle de acessos.

5.3 Trabalhos Futuros

Com base nos resultados e nas limitações identificadas, sugerem-se os seguintes pontos para trabalhos futuros:

1. **Aprimoramento da Persistência de Dados**: Implementar armazenamento permanente no gateway e na tranca, evitando a perda de informações em caso de falhas de energia ou reinicializações. O uso de bancos de dados locais ou armazenamento em EEPROM poderia garantir maior resiliência.
2. **Otimização do Hardware do Gateway**: Substituir o hardware atual por um que ofereça maior estabilidade e capacidade de processamento para lidar com picos de acesso e evitar os travamentos observados.
3. **Melhorias na Interface do Sistema Web**: A interface web, apesar de funcional, necessita de refinamentos para melhorar a experiência do usuário. Investir em uma UX/UI mais intuitiva pode facilitar a adoção do sistema por gestores e administradores.
4. **Estudo de Outras Tecnologias de Comunicação**: Embora o LoRa tenha mostrado bons resultados, explorar tecnologias alternativas como Zigbee ou NB-IoT pode aumentar ainda mais a confiabilidade e reduzir latências, especialmente em ambientes com interferências de sinal.
5. **Teste em Ambientes Reais**: O sistema pode ser testado em larga escala em diferentes tipos de laboratórios, com variações nas configurações físicas e nas necessidades de segurança, a fim de validar sua aplicabilidade em contextos mais diversos.

6 – REFERÊNCIAS

UNIFESSPA. Comunidade acadêmica e autoridades políticas prestigiam inauguração do galpão de laboratórios do IGE. 2018. Disponível em: https://unifesspa.edu.br/noticias/2296-comunidade-academica-e-autoridades-politicas-prestigiam-inauguracao-do-galpao-de-laboratorios-do-ige. Acesso em: 25 jun. 2023.

CAYENNE. Cayenne Features - Developer. 2023. Disponível em: https://developers.mydevices.com/cayenne/features/. Acesso em: 14 jul. 2023.

TCT BRASIL. LoRa e protocolo LoRaWAN redes LPWAN. 2019. Disponível em: http://www.tctbrasil.com.br/wp-content/uploads/2019/07/lora-technology-tct-brasil.pdf. Acesso em: 14 jul. 2023.

ESPRESSIF. ESP32. 2015-2023. Disponível em: https://www.espressif.com/en/products/socs/esp32. Acesso em: 10 jul. 2023.

ESPRESSIF. ESP32 - DevKitC. 2015-2023. Disponível em: https://www.espressif.com/en/products/devkits/esp32-devkitc. Acesso em: 14 jul. 2023.

DJANGO SOFTWARE FOUNDATION. Django Documentation. 2005-2024. Disponível em: https://docs.djangoproject.com/en/5.0/. Acesso em: 6 ago. 2024.

LILYGO©. LoRa32 V1.0. 2023. Disponível em: https://www.lilygo.cc/products/lora32-v1-0. Acesso em: 15 jul. 2023.

ORACLE BRASIL. O que é IoT? Disponível em: https://www.oracle.com/br/internet-of-things/what-is-iot/#:~:text=A%20Internet%20das%20Coisas%20(IoT)%20descreve%20a%20rede%20de%20objetos,comuns%20a%20ferramentas%20industriais%20sofisticadas.. Acesso em: 29 jun. 2023.

DOCKER INC. Docker Docs. 2013-2024. Disponível em: https://docs.docker.com/. Acesso em: 6 ago. 2024.

THE THINGS NETWORK. RSSI and SNR. Disponível em: https://www.thethingsnetwork.org/docs/lorawan/rssi-and-snr/. Acesso em: 10 set. 2024.

MIFARE. AN10834 - MIFARE ISO/IEC 14443 PICC selection. 2021. Disponível em: https://www.nxp.com/docs/en/application-note/AN10834.pdf. Acesso em: 23 jul. 2024.

IOT RESEARCH NEWSLETTER. State of IoT 2023: number of connected IoT devices growing 16% to 16.7 billion globally. 2023. Disponível em: https://iot-analytics.com/number-connected-iot-devices/. Acesso em: 29 jun. 2023.

IOT RESEARCH NEWSLETTER. State of IoT spring 2024: 10 emerging IoT trends driving market growth. Disponível em: https://iot-analytics.com/state-of-iot-10-emerging-iot-trends-driving-market-growth/. Acesso em: 16 jul. 2024.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. About. 1996-2024. Disponível em: https://www.postgresql.org/about/. Acesso em: 6 ago. 2024.

ANATEL. Resolução nº 680, de 27 de junho de 2017. 2017. Disponível em: https://informacoes.anatel.gov.br/legislacao/resolucoes/2017/936-resolucao-680. Acesso em: 15 jul. 2023.

SANTOS, K. de F. et al. Um componente de software para integrar leitores de biometria a um sistema de controle de acesso. In: CONGRESSO DE PESQUISA E INOVAÇÃO DA REDE NORTE E NORDESTE DE EDUCAÇÃO TECNOLÓGICA, 2009.

MENEZES, Árly Assis Martins Cordeiro. Implementação de um sistema para acesso do laboratório de automação predial. 2018. 50 f. Monografia (Graduação em Engenharia de Controle e Automação) – Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2018.

DELFINO, Lorrainy Rembiski; DOS SANTOS, Otávio Lube. Um projeto de sala de aula inteligente para a FAESA com o uso da internet das coisas e MQTT. Revista Científica da FAESA, Vitória, ES, v. 15, n. 2, p. 121-142, 2019.

PREE, Wolfgang. Design patterns for object-oriented software development. ACM Press/Addison-Wesley Publishing Co., 1995.

LIRA, R. V.; BATISTA, E. D. de A.; ALVES, L. V. do N.; LOURENÇO, A. dos S. A.; GONZAGA, G. M. Development of a basic prototype of hardware and software for physical

security and access control in institutional environments that contains valuable assets and information. Research, Society and Development, v. 12, n. 4, p. e8812439849, 2023. DOI: 10.33448/rsd-v12i4.39849. Disponível em: https://rsdjournal.org/index.php/rsd/article/view/39849. Acesso em: 27 jun. 2023.

PEREIRA, Vitor; DIAS, Samaherni; QUEIROZ, Kurios de. Sistema inteligente para controle de acesso e monitoramento de múltiplos ambientes (class control). In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SISTEMAS COMPUTACIONAIS (SBESC), 9., 2019, Natal. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 25-30. DOI: https://doi.org/10.5753/sbesc_estendido.2019.8631.

HILLIER, Frederick S.; LIEBERMAN, Gerald J. Introdução à pesquisa operacional. McGraw Hill Brasil, 2013.

FINKENZELLER, Klaus. RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication. John Wiley & Sons, 2010.

LOPEZ, G. N. Sistema para monitoramento de acesso utilizando Arduino e leitura biométrica. Bacharelado em Engenharia de Controle e Automação – Universidade Federal de Ouro Preto, 2014.

PINO, Alessandro Mouras de Oliveira. Desenvolvimento de protótipo de um sistema embarcado de fechadura eletrônica para controle de homologação e acesso. 2019. Disponível em: https://repositorio.ufersa.edu.br/handle/prefix/6643. Acesso em: 15 jul. 2023.

7. APÊNDICES

7.1 Anexo 01 - Tranca.ino

```
#include "bibliotecas.h"

//DEFININDO PORTAS PARA O DISPLAY
#define DISPLAY_ADDRESS_PIN 0x3c
#define DISPLAY_SDA 4
#define DISPLAY_SCL 15
#define DISPLAY_RST 16

// Altura da fonte (correspondente a fonte
ArialMT_Plain_16)
const int fontHeight = 16;

// CONTADOR DE PACOTES RECEBIDOS
int counter = 0;

//DEFININDO PORTAS DE COMUNICAÇÃO
DO LORA E DPI
#define SCK 5
#define MISO 19
#define MOSI 27
#define NSS 18
#define RESET 14
#define DIO0 26

//BANDA DE USO DO LORA
#define BAND 928E6

// DISPOSITIVOS DA REDE LoRa
#define LoRaGatewayAddress 0x01
#define LoRaLockerAddress 0x02

// Instantiate Wire for generic use at 400kHz
TwoWire I2Cone = TwoWire(0);
TwoWire I2Ctwo = TwoWire(1);

// Objeto do display
SSD1306 display(DISPLAY_ADDRESS_PIN,
DISPLAY_SDA, DISPLAY_SCL);

//DEFININDO PORTAS PRO RFID
#define SS_PIN 17 //PINO SDA
#define RST_PIN 13 //PINO DE RESET

#define bot 25  //PINO DE RESET
//OBJETO DO LEITOR RFID
MFRC522 rfid(SS_PIN, RST_PIN);

// Set i2c address
PCF8574 pcf8574(&I2Ctwo, 0x20);

#define Password_Length 8
char KeyWritten[Password_Length + 5];

String MasterKey = "DCBA248675913";
char actualKey;
byte keyCount = 0;

// Define o layout do teclado matricial
const byte numRows = 4;
const byte numCols = 4;
char keymap[numRows][numCols] = {
  { '1', '2', '3', 'A' },
  { '4', '5', '6', 'B' },
  { '7', '8', '9', 'C' },
  { '*', '0', '#', 'D' }
};

// Define os pinos do teclado matricial
byte rowPins[numRows] = { 4, 5, 6, 7 };  // Mapeie
os pinos às linhas do teclado
byte colPins[numCols] = { 0, 1, 2, 3 };  // Mapeie
os pinos às colunas do teclado

//Variável Armazenadora de Acessos
//0 - Id; 1 - Nome, 2 - Tag, 3 - Senha
String acessos[20][5];

#define tranca 12

bool controle = false;
bool showKeyWriteen = false;

// Função que inicializa o display
bool displayBegin() {
  // Reiniciamos o display
  pinMode(DISPLAY_RST, OUTPUT);
  digitalWrite(DISPLAY_RST, LOW);
  delay(1);
  digitalWrite(DISPLAY_RST, HIGH);
  delay(1);

  if (!display.init()) {
    Serial.println("Display não inicializado");
    return false;
  }
  // Invertemos o display verticalmente
  display.flipScreenVertically();
  // Setamos a fonte
  display.setFont(ArialMT_Plain_16);
  // Alinhamos a fonte à esquerda
```

```
display.setTextAlignment(TEXT_ALIGN_LEFT);
  // Limpamos a tela do display
  display.clear();

  return true;
}

// Função que inicializa o radio LoRa
bool loraBegin() {
  // Iniciamos a comunicação SPI
  SPI.begin(SCK, MISO, MOSI, NSS);
  // Setamos os pinos do LoRa
  LoRa.setPins(NSS, RESET, DIO0);
  // Iniciamos o LoRa
  if (!LoRa.begin(BAND)) {
    Serial.println("LoRa falhou em iniciar nessa
Frequência");
    return false;
  }

  LoRa.setTxPower(20);
  return true;
}

void setup() {
  Serial.begin(115200);

  // Configurar os pinos do PCF8574
  for (int i = 0; i < 8; i++) {
    if (i < 4) {
      pcf8574.pinMode(i, INPUT_PULLUP);
    } else {
      pcf8574.pinMode(i, OUTPUT);
      pcf8574.digitalWrite(i, HIGH);
    }
  }

  pinMode(bot, INPUT_PULLUP);

  // Inicializa o objeto PCF8574
  if (pcf8574.begin()) {
    Serial.println("Expansor inicializado com
sucesso!");
  } else {
    Serial.println("Falha ao inicializar o
Expansor.");
    while (1)
      ;
  }

  if (!displayBegin()) {
    // Se não deu certo, exibimos falha de display na
serial
    Serial.println("Display failed!");
    // E deixamos em loop infinito
    while (1)
      ;
  }

  if (!loraBegin()) {  // Iniciamos o LoRa
    // Se não deu certo, exibimos falha de LoRa na
serial
    Serial.println("LoRa failed!");
    display.clear();
    display.drawString(0, 0, "LoRa failed!");
    display.display();
    while (1)  // E deixamos em loop infinito
      ;
  }

  rfid.PCD_Init();  //INICIALIZA O LEITOR DE
TAG RFID 01

  pinMode(tranca, OUTPUT);  //DEFINE O PINO
COMO SAÍDA PARA A TRANCA

  for (int i = 0; i < 20; i++) {
    acessos[i][3] = "Not Initialized";
    acessos[i][4] = "Entrada";
  }

  display.clear();
  display.drawString(0, 0, "System");
  display.drawString(2, 1 * fontHeight,
"Verification");
  display.drawString(0, 2 * fontHeight,
"Complete");
  Serial.println("System verification complete");
  display.display();
  delay(500);


  for (int i = 0; i < logo_height; i++) {
    for (int j = 0; j < logo_width / 8; j++) {
      logo_bitmap[i * (logo_width / 8) + j] =
~logo_bitmap[i * (logo_width / 8) + j];
    }
  }
}

// IMPRIME NA TELA TODOS OS ACESSOS
CADASTRADOS NO ESP32
void printAccess() {
  int line;
  for (int i = 0; i < 20; i++) {
    line = 0;
    display.clear();
    Serial.println(acessos[i][0] + " : " + acessos[i][1]
+ " : " + acessos[i][2] + " : " + acessos[i][3]);
```

```
    display.drawString(0, line, acessos[i][0]);
    line++;
    display.drawString(0, line * fontHeight,
acessos[i][1]);
    line++;
    display.drawString(0, line * fontHeight,
acessos[i][2]);
    line++;
    display.drawString(0, line * fontHeight,
acessos[i][3]);
    display.display();
    delay(600);
  }
}

bool feelingLonely = true;
unsigned long timeToPing = millis();

// FAZ A LEITURA DE TAGS RFID
void leituraRFID() {

  if (!rfid.PICC_IsNewCardPresent() ||
!rfid.PICC_ReadCardSerial())  //VERIFICA SE O
CARTÃO PRESENTE NO LEITOR É
DIFERENTE DO ÚLTIMO CARTÃO LIDO.
CASO NÃO SEJA, FAZ
    return;
//RETORNA PARA LER NOVAMENTE

  /***INICIO BLOCO DE CÓDIGO
RESPONSÁVEL POR GERAR A TAG RFID
LIDA***/
  String strID = "";
  for (byte i = 0; i < 4; i++) {
    strID +=
      (rfid.uid.uidByte[i] < 0x10 ? "0" : "") +
String(rfid.uid.uidByte[i], HEX) + (i != 3 ? ":" :
"");
  }
  strID.toUpperCase();
  /***FIM DO BLOCO DE CÓDIGO
RESPONSÁVEL POR GERAR A TAG RFID
LIDA***/

  Serial.print("Identificador (UID) da tag: ");
//IMPRIME O TEXTO NA SERIAL
  Serial.println(strID);                    //IMPRIME
NA SERIAL O UID DA TAG RFID

  rfid.PICC_HaltA();       //PARADA DA
LEITURA DO CARTÃO
  rfid.PCD_StopCrypto1();  //PARADA DA
CRIPTOGRAFIA NO PCD

  display.clear();
  display.drawString(0, 0, "User ID Tag:");
  display.drawString(0, 1 * fontHeight, "=> " +
strID);
  display.display();
  delay(100);

  for (int i = 0; i < 20; i++) {
    if (acessos[i][2] == strID) {
      liberaAe(i);
      break;
    }
  }

  if (controle == false) {
    display.clear();
    display.drawString(0, 0, "Acesso Negado");
    display.drawString(0, 1 * fontHeight, "     =(
");
    display.display();
    delay(100);
  }
}

// LIMPA OS DADOS DE BUFFER DE SENHA
void clearData() {
  while (keyCount != 0) {
    KeyWritten[keyCount--] = 0;
  }
  KeyWritten[0] = 0;
  return;
}

bool debugSenha = false;

// FAZ A ANÁLISE DE SENHAS DIGITADAS
void leituraSenha() {
  if (keyCount == 13 && actualKey == '*') {
    showKeyWriteen = false;
    if (String(KeyWritten) == MasterKey) {
      digitalWrite(tranca, HIGH);
      display.clear();
      display.drawString(0, 0, "Acesso Concedido");
      display.drawString(0, 1 * fontHeight, "     =D
");
      display.display();
      delay(2000);
      digitalWrite(tranca, LOW);
      display.clear();
      display.drawString(0, 0, "Bem Vindo");
      display.drawString(0, 1 * fontHeight,
"Administrador");
      display.display();
      delay(100);
      controle = true;
    } else {
```

```
      display.clear();
      display.drawString(0, 0, "Acesso Negado");
      display.drawString(0, 1 * fontHeight, "     =(
");
      display.display();
      Serial.println(F("Acesso Negado"));
      delay(100);
    }
    clearData();
  } else if (keyCount == 13) {
    Serial.println("Limite de Caracteres atingido");
  } else if (actualKey == '*' && keyCount <= 7) {
    for (int i = 0; i < 6; i++) {
      if (acessos[i][3] == String(KeyWritten)) {
        liberaAe(i);
        break;
      }
    }
    if (controle == false) {
      display.clear();
      display.drawString(0, 0, "Acesso Negado");
      display.drawString(0, 1 * fontHeight, "     =(
");
      display.display();
      Serial.println(F("Acesso Negado"));
      delay(100);
    }
    showKeyWriteen = false;
    clearData();
  } else {
    KeyWritten[keyCount] = actualKey;
    keyCount++;
    showKeyWriteen = true;
    debugSenha = true;
  }
  delay(150);
}

// FUNÇÃO PARA LER A TECLA DIGITADA
DO TECLADO MATRICIAL
char getKeyFromKeypad() {
  // Esta função realiza a varredura das linhas e
colunas do teclado matricial
  for (byte row = 0; row < numRows; row++) {
    // Configura a linha atual como saída em nível
baixo
    pcf8574.digitalWrite(rowPins[row], LOW);

    // Verifica as colunas para uma tecla pressionada
    for (byte col = 0; col < numCols; col++) {
      if (pcf8574.digitalRead(colPins[col]) == LOW)
{
        // Uma tecla foi pressionada
        // Desativa a linha atual
        pcf8574.digitalWrite(rowPins[row], HIGH);

        // Aguarde um breve intervalo para evitar
leituras repetidas
        delay(50);

        // Retorna o caractere correspondente à tecla
pressionada
        return keymap[row][col];
      }
    }

    // Desativa a linha atual
    pcf8574.digitalWrite(rowPins[row], HIGH);
  }

  // Nenhuma tecla foi pressionada
  return '\0';
}

String leftForPost[12][2];
int postCounting = 0;

// LIBERA ACESSO A TRANCA
void liberaAe(int posicao) {
  digitalWrite(tranca, HIGH);
  display.clear();
  display.drawString(0, 0, "Acesso Concedido");
  display.drawString(0, 1 * fontHeight, "     =D
");
  display.display();
  delay(200);
  digitalWrite(tranca, LOW);
  display.clear();
  display.drawString(0, 0, "Bem Vindo");
  display.drawString(0, 1 * fontHeight,
acessos[posicao][1]);
  display.display();
  delay(400);
  Serial.println("Postando no Servidor");
  LoRa.beginPacket();
  LoRa.write(0x01);
  LoRa.write(0x02);
  LoRa.print("ITHinkISawALittleCat|");
  LoRa.print(acessos[posicao][0] + "|");
  LoRa.print(acessos[posicao][4] + "|");
  LoRa.endPacket();
  if (postCounting < 12) {
    leftForPost[postCounting][0] =
acessos[posicao][0];
    leftForPost[postCounting][1] =
acessos[posicao][4];
    postCounting++;
  } else {
    Serial.println(F("Número MAX de objetos a
serem postados roubados atingido!!!"));
```

```
  }
  controle = true;
}

// Refaz o envio do último acesso realizado ao gateway para que o mesmo realize a postagem no servidor
void justRemindTheGatewayItsJobToPost() {
  LoRa.beginPacket();
  LoRa.write(0x01);
  LoRa.write(0x02);
  LoRa.print("ITHinkISawALittleCat|");
  LoRa.print(leftForPost[postCounting][0] + "|");
  LoRa.print(leftForPost[postCounting][1] + "|");
  LoRa.endPacket();
}

// SOLICITA OS DADOS DE ACESSO AO GATEWAY
void sendMeTheAccessHoney() {
  LoRa.beginPacket();
  LoRa.write(0x01);
  LoRa.write(0x02);
  LoRa.print("SendMeTheAccessHoney|");
  LoRa.endPacket();
  Serial.println("SendMeTheAccessHoney");
}

// MANDA UMA MENSAGEM DE CONTROLE A FIM DE ESPERAR RESPOSTA DO GATEWAY
void ping() {
  LoRa.beginPacket();
  LoRa.write(0x01);
  LoRa.write(0x02);
  LoRa.print("AreYouThere|");
  LoRa.endPacket();
  Serial.println("AreYouThere");
}

void accessGrantedStorage(int packet) {
  if (packet == 0) {
    for (int i = 0; i < 10; i++) {
      acessos[i][0] = LoRa.readStringUntil('|');
      acessos[i][1] = LoRa.readStringUntil('|');
      acessos[i][2] = LoRa.readStringUntil('|');
      acessos[i][3] = LoRa.readStringUntil('|');
    }
  } else if(packet == 1){
    for (int i = 10; i < 20; i++) {
      acessos[i][0] = LoRa.readStringUntil('|');
      acessos[i][1] = LoRa.readStringUntil('|');
      acessos[i][2] = LoRa.readStringUntil('|');
      acessos[i][3] = LoRa.readStringUntil('|');
    }
  }
}

// ANALISA UMA MENSAGEM LoRa RECEBIDA
void messageAnalisys(uint8_t address, String message) {
  if (message == "200")
sendMeTheAccessHoney();
  else if (message == "503") Serial.println("503 - Servidor Ocupado, Tente Novamente Mais Tarde");
  else if (message == "200 - HereAreTheAccess 1")
accessGrantedStorage(0);
  else if (message == "200 - HereAreTheAccess 2")
accessGrantedStorage(1);
  else if (message == "200 - Post done") {
    Serial.println("deu bom");
    leftForPost[0][0] = "";
    leftForPost[0][1] = "";
    for (int i = 1; i < postCounting && i < 12; i++) {
      if (i == 1) {
        leftForPost[0][0] = leftForPost[1][0];
        leftForPost[0][1] = leftForPost[1][1];
      } else if (i == 12) {
        break;
      }
      leftForPost[i - 1][0] = leftForPost[1][0];
      leftForPost[i - 1][1] = leftForPost[1][1];
    }
    postCounting--;
  } else if (message == "400 - Post not done")
justRemindTheGatewayItsJobToPost();

  int snr = LoRa.packetSnr();
  int rssi = LoRa.rssi();
  Serial.println("RSSI: " + String(rssi) + "\nSNR: " + String(snr));
}

void loop() {
  int packetSize = LoRa.parsePacket();  // Armazena o tamanho do pacote recebido
  actualKey = getKeyFromKeypad();       // Função personalizada para ler o teclado matricial
  int botPress = digitalRead(bot);

  if (showKeyWriteen) {
    if (debugSenha) {
      Serial.println("A contagem de caracteres inseridos é de " + String(keyCount));
      Serial.println("Senha: " + String(KeyWritten));
      debugSenha = false;
    }
    display.clear();
```

```
   display.drawString(0, 0, "Senha:");
   display.drawString(0, fontHeight,
String(KeyWritten) + "_");
   display.drawString(0, 2 * fontHeight, "Pressione
*");
   display.drawString(0, 3 * fontHeight, "Para
Confirmar");
   display.display();
  } else if (!showKeyWriteen) {
   display.clear();
   display.drawXbm(0, 0, logo_width, logo_height,
logo_bitmap);
   display.display();
  }

  if (packetSize) {
   uint8_t address = LoRa.read();
   if (address == 0x02) {
    uint8_t deviceApplicant = LoRa.read();
    if (deviceApplicant == 0x01) {
     String message = LoRa.readStringUntil('|');
     Serial.println("Received: " + message);
     display.clear();
     display.drawString(0, 0, "Server Answer:");
     display.drawString(0, 1 * fontHeight,
message);
     display.display();
     delay(30);
     messageAnalisys(deviceApplicant, message);
    }
   }
  } else {
   if (actualKey) leituraSenha();
   else {
    leituraRFID();
   }
  }

  if (feelingLonely || millis() - timeToPing >=
60000) {
   ping();
   timeToPing = millis();
   feelingLonely = false;
  }

  if(botPress == LOW){
   digitalWrite(tranca, HIGH);
   display.clear();
   display.drawString(0, 0, "Acesso Concedido");
   display.drawString(0, 1 * fontHeight, "     =D
");
   display.display();
   delay(200);
   digitalWrite(tranca, LOW);
   display.clear();
   display.drawString(0, 0, "Até Logo");
   display.drawString(0, 1 * fontHeight, "     =D
");
   display.display();
   delay(400);
  }

  controle = false;
 }
```

7.2 Anexo 02 - Função para conectar a Rede WiFi WPA2-Enterprise

```
// Inicia a conexão a Rede WiFi WPA2-Enterprise
void wifiBegin() {
  Serial.print(F("Connecting to network: "));
  Serial.println(ssid);

  WiFi.disconnect(true);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, WPA2_AUTH_PEAP, EAP_IDENTITY, EAP_USERNAME, EAP_PASSWORD);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
    counter++;
    if (counter >= 60) {
      Serial.print(F("Não foi possível conectar em "));
      Serial.print(ssid);
      Serial.println(F("! Reiniciando o sistema para tentar conectar no WiFi"));
      ESP.restart();
    }
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address set: ");
  Serial.println(WiFi.localIP());
}
```