



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO AMAZONAS
CAMPUS MANAUS DISTRITO INDUSTRIAL
TECNOLOGIA EM MECATRÔNICA INDUSTRIAL



ASSUNÇÃO ALEME DA CRUZ JUNIOR

SISTEMA DE CONTROLE DE ACESSO UTILIZANDO INTERNET DAS COISAS E
TECNOLOGIA RFID

MANAUS - AM
2024

ASSUNÇÃO ALEME DA CRUZ JUNIOR

SISTEMA DE CONTROLE DE ACESSO UTILIZANDO INTERNET DAS COISAS E
TECNOLOGIA RFID

Trabalho de Conclusão de Curso apresentado ao Curso de Tecnologia em Mecatrônica Industrial do Campus Manaus Distrito Industrial, Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (CMDI/IFAM), como requisito parcial para obtenção do grau de Tecnólogo em Mecatrônica.

Orientador: Dr. Vitor Bremgartner da Frota

MANAUS – AM

2024

Dados Internacionais de Catalogação na Publicação (CIP)

C962s Cruz Junior, Assunção Aleme da.
Sistema de controle de acesso utilizando Internet das Coisas e tecnologia RFDI/ Assunção Aleme da Cruz Junior. — Manaus, 2024.
49f.: il. color.

Monografia (Graduação) — Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, *Campus* Manaus Distrito Industrial, Curso de Tecnologia em Mecatrônica Industrial, 2025.
Orientador: Prof.º Vitor Bremgartner da Frota, Dr.

1. Internet das Coisas. 2. Iot. 3. Controle de Acesso. 4. RFID. I. Frota, Vitor Bremgartner da. II. Instituto Federal de Educação, Ciência e Tecnologia do Amazonas. III. Título.

CDD 621.381

Elaborada por Oziane Romualdo de Souza (CRB111/ nº 734)



Ministério da Educação
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas
Campus Manaus Distrito Industrial
Departamento de Automação e Produção
Tecnologia em Mecatrônica Industrial



TERMO DE APROVAÇÃO

SISTEMA DE CONTROLE DE ACESSO UTILIZANDO INTERNET DAS COISAS E
TECNOLOGIA RFID

por

ASSUNÇÃO ALEME DA CRUZ JUNIOR

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 20 de dezembro de 2024 como requisito parcial para a obtenção do título de Tecnólogo em Mecatrônica Industrial. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Vitor Bremgartner da Frota
Prof.(a) Orientador(a)

Prof. Me. Hugo Alves Velozo
Membro titular

Prof. Dr. Alyson de Jesus dos Santos
Membro titular

AGRADECIMENTOS

A realização deste trabalho marca o encerramento de uma importante etapa da minha vida acadêmica, e não poderia deixar de expressar minha profunda gratidão a todos que, de alguma forma, contribuíram para que este momento fosse possível.

Primeiramente, agradeço à minha dedicação aos estudos e à coragem de confrontar desafios que, muitas vezes, pareciam difíceis. Cada obstáculo vencido foi um aprendizado que me trouxe até aqui, moldando meu crescimento pessoal e profissional.

À minha mãe, meu porto seguro, que sempre acreditou em mim. Sua força e apoio inabaláveis me impulsionaram a seguir em frente nos momentos mais difíceis.

À minha irmã, por todo o carinho e cuidado, por sempre me mimar de maneiras que aquecem o coração e tornam a caminhada mais leve.

Ao meu pai, que, com sua sabedoria, me ensinou a viver com a vida, mostrando que os desafios não são inimigos, mas oportunidades de aprendizado e crescimento.

Aos meus professores e ao meu orientador Vitor Bremgartner, que desempenharam um papel essencial em meu processo de aprendizado. Suas orientações, paciência e dedicação foram fundamentais para que eu adquirisse o conhecimento e a confiança necessários para chegar até aqui.

E à Samyllis, pela presença constante em minha vida, pela ajuda nos momentos em que precisei, mesmo quando eu duvidava das minhas próprias capacidades, e por compartilhar comigo alegrias e dificuldades. Sua amizade e apoio foram fundamentais para que eu alcançasse esta conquista.

A todos vocês, dedico este trabalho com imensa gratidão. Sem cada um, este sonho não teria se tornado realidade.

"O que seria da vida se não tivéssemos a coragem de tentar algo?" (Vincent Van Gogh)

RESUMO

Este trabalho tem como objetivo desenvolver um sistema que permita a autenticação e validação de acesso do usuário, utilizando as tecnologias IoT (Internet das Coisas) e RFID (Identificação por Radiofrequência), em conjunto com o microcontrolador ESP32 e um dispositivo como celular ou computador. O usuário, por meio de uma etiqueta RFID, realizará a validação de seu acesso, que será consultado em tempo real em um banco de dados hospedado no Firebase. O sistema, então, procederá com a autenticação e decidirá se o acesso é autorizado ou não. Para atingir este objetivo, foi necessário integrar recursos de hardware e software, com a utilização do Firebase Realtime Database para o armazenamento e gerenciamento de dados. A plataforma de desenvolvimento escolhida foi a IDE Arduino, e a placa ESP32 foi utilizada como microcontrolador principal, com o auxílio de atuadores para controlar o acesso. Este trabalho visa proporcionar ao usuário a capacidade de monitorar e controlar acessos de forma prática e eficiente.

Palavras-chave: IoT; Controle de acesso; Firebase; RFID.

ABSTRACT

This work aims to develop a system that allows user authentication and access validation, using IoT (Internet of Things) and RFID (Radio Frequency Identification) technologies, along with the ESP32 microcontroller and a device such as a mobile phone or computer. The user, through an RFID tag, will validate their access, which will be checked in real-time in a Firebase-hosted database. The system will then proceed with the authentication and decide whether access is authorized or not. To achieve this goal, it was necessary to integrate hardware and software resources, using Firebase Realtime Database for data storage and management. The chosen development platform was the Arduino IDE, and the ESP32 board was used as the main microcontroller, with the assistance of actuators to control access. This work aims to provide the user with the ability to monitor and control access in a practical and efficient manner.

Keywords: IoT; Access control; Firebase; RFID.

LISTA DE ILUSTRAÇÕES

Figura 1 — Detalhes da utilização do RFID.....	18
Figura 2 — Estrutura de um transponder.....	19
Figura 3 — Exemplo de sistema utilizando dispositivo leitor	20
Figura 4 — Modelo de comunicação SPI.....	22
Figura 5 — Modelo de comunicação SPI mestre e escravos	22
Figura 6 — Nó JSON.....	24
Figura 7 — ESP32.....	27
Figura 8 — Pinagem do Esp32.....	28
Figura 9 — Diagrama de blocos funcionais	29
Figura 10 — Módulo RC522	30
Figura 11 - Micro servo Tower Pro SG90	31
Figura 12 — Interface Arduino IDE	32
Figura 13 — Fluxograma	34
Figura 14 — Arquitetura do sistema	35
Figura 15 — Esquema do protótipo	36
Figura 16 — Diagrama do Protótipo	37
Figura 17 — Código de Configuração.....	38
Figura 18 — Importação SDK.....	38
Figura 19 — Tela Inicial.....	39
Figura 20 — Tela de cadastro	39
Figura 21 — Monitorar acessos.....	40
Figura 22 — Bibliotecas.....	40
Figura 23 — Configuração do Firebase	41
Figura 24 — Regras do Firebase.....	41
Figura 25 — Código em loop	42
Figura 26 — Função para cadastro	42
Figura 27 — ID do RFID enviado à variável tempUID	43
Figura 28 — Captura do UID em Firebase	43
Figura 29 — UID enviado à caixa de texto	43
Figura 30 — Captura de dados em determinado nó.....	44

Figura 31 — Gerar log	44
Figura 32 — Nó em registro de acesso	45
Figura 33 — Captura da Array	45
Figura 34 — Registro de acessos.....	45
Figura 35 — Circuito montado	46
Figura 36 — Tag válida.....	46
Figura 37 — Confirmação de registro.....	47
Figura 38 — Acesso negado	47

LISTA DE TABELAS

Tabela 1 — Dados técnicos ESP32.....	29
Tabela 2 — Especificações técnicas do módulo RC522	31
Tabela 3 — Custo do protótipo	48
Tabela 4 — Produtos semelhantes de empresas	48

LISTA DE ABREVIATURAS E SIGLAS

IoT	Internet of Things (Internet das Coisas)
RFID	Radio Frequency Identification
ESP32	Espressif Systems 32-bit Microcontroller
NTP	Network Time Protocol
SDK	Software Development Kit
JSON	JavaScript Object Notation
HTTP	HyperText Transfer Protocol
Wi-Fi	Wireless Fidelity
API	Application Programming Interface
BaaS	Backend as a Service
ROM	Read-Only Memory
SDA	Serial Data
SCK	Serial Clock
MISO	Master In Slave Out
MOSI	Master Out Slave In
GPIO	General Purpose Input/Output
WIFI	Wireless Fidelity

SUMÁRIO

1 INTRODUÇÃO	14
1.1 JUSTIFICATIVA	15
1.2 DELIMITAÇÃO DO TEMA	15
1.3 OBJETIVOS	16
1.3.1 Geral	16
1.3.2 Objetivos específicos	16
2 REFERENCIAL TEÓRICO	16
2.1 INTERNET DAS COISAS	16
2.2 TECNOLOGIA DE IDENTIFICAÇÃO POR RADIOFREQUÊNCIA	18
2.2.1 Princípio de funcionamento	20
2.2.2 Tipos de etiquetas	20
2.3 PROTOCOLO SERIAL PERIPHERAL INTERFACE	21
2.3.1 Comunicação no Protocolo SPI	22
2.4 FIREBASE.....	23
2.4.1 JSON e Realtime Database	23
2.5 HTML e CSS	24
2.6 JAVASCRIPT	25
2.7 BOOTSTRAP	25
2.8 TRABALHOS RELACIONADOS	25
3 MATERIAIS E MÉTODOS	26
3.1 MATERIAIS	26
3.1.1 Hardware	26
3.1.1.1 ESP32	27
3.1.1.2 Módulo RC522.....	30
3.1.1.3 MICRO SERVO	31
3.1.2 Software	32
3.1.2.1 Arduino IDE	32
3.1.2.4 Notepad++.....	33
3.2 MÉTODOS	33
3.2.1 Arquitetura do projeto	33
3.2.2 Desenvolvimento	37
3.2.2.1 Plataforma web.....	38
3.2.2.2 Desenvolvimento ESP32	40
4 RESULTADOS	46
4.1 CUSTO DO PROJETO E COMPARAÇÃO.....	48
5 CONCLUSÃO	49
5.1 TRABALHOS FUTUROS.....	49
REFERÊNCIAS	50

APÊNDICE A — Código do ESP32.....	53
-----------------------------------	----

1 INTRODUÇÃO

A necessidade de segurança no ambiente em que vivemos sempre foi um problema enfrentado pela sociedade. A vida não está relacionada apenas à segurança pessoal, mas também de manter um local seguro quando não se está nele ou próximo o suficiente para tê-lo em vista (Cardoso, 2014).

Além da necessidade de manter uma posição segura, há também a necessidade de ter controle em alguns departamentos, apenas pessoal autorizado tem acesso afirmado antes. Embora soluções já conhecidas como portas e fechaduras resolvem o problema de manter um local seguro, elas não fornecem a solução do controle de acesso (Cardoso, 2014).

Com o surgimento de novas tecnologias, o custo dos componentes eletrônicos está cada vez mais baixo, o que permite desenvolver sistemas mais complexos com valores mais atrativos para produção, proporcionando assim um produto mais barato ao usuário final, o que é um fator importante tanto para o comércio quanto para as pessoas, principalmente aquelas com poucos recursos.

Nos últimos anos, processos de identificação automática têm sido amplamente utilizados em diversos serviços industriais, como logística de compras e distribuição, a empresa fabricação e fluxo de materiais. Esses procedimentos automáticos conseguem fornecer informações sobre pessoas, animais, mercadorias e produtos em trânsito (Finkenzeller, 2010).

Este trabalho propõe uma solução baseada na integração de sistemas, responsáveis pela coleta e transmissão de dados por meio de uma plataforma web. Esse sistema captura informações de identificação e envia para processamento e decisão, permitindo um controle de acesso seguro e automatizado.

A proposta combina avanços tecnológicos em hardware e software com a acessibilidade econômica proporcionada pela redução dos custos de componentes eletrônicos, tornando possível o desenvolvimento de sistemas robustos e de baixo custo. Essa solução não apenas atende às demandas de segurança, mas também otimiza processos, ao integrar tecnologias de identificação automática utilizadas em setores como logística e gestão de fluxos.

O sistema apresentado oferece uma abordagem eficiente e escalável, destacando-se pela sua flexibilidade em se adaptar a diferentes cenários e

necessidades. Ele reflete uma convergência entre inovação tecnológica e aplicabilidade prática.

1.1 JUSTIFICATIVA

O controle de acesso é uma questão muito importante em ambientes como empresas, indústrias, instituições de ensino e condomínios, onde a segurança e a restrição de acesso a áreas específicas são necessárias. No entanto, os métodos tradicionais utilizados nesses sistemas, como o uso de chaves físicas, apresentam limitações que comprometem tanto a segurança quanto a gestão do controle de acesso, sendo vulneráveis a possíveis problemas como: perdas, clonagem e compartilhamento por terceiros. Esses fatores geram um risco elevado de acessos indevidos, expondo áreas restritas a brechas de segurança.

Diante desse contexto, é importante que seja desenvolvido um sistema de controle de acesso utilizando as tecnologias IoT e RFID, com monitoramento em tempo real. A necessidade de automação e monitoramento com um sistema de IoT permite que o controle preciso dos cadastros de usuários, como a identificação do RFID, de forma automatizada, tal como possuir um acesso remoto de dados coletados que possibilite ao usuário monitorar e tomar medidas corretivas, mesmo não estando fisicamente no local.

Na realização do mesmo, podemos garantir a otimização da restrição do controle de acesso, oferecendo resultado como controle automatizado por meio do microcontrolador ESP32 de baixo custo. Portanto, este trabalho se justifica pela necessidade de desenvolver um sistema de controle de acesso IoT com a tecnologia RFID, além de proporcionar aos usuários a visualização de acessos e controle de cadastros.

1.2 DELIMITAÇÃO DO TEMA

O trabalho irá abordar o desenvolvimento e implementação de um sistema de monitoramento baseado em IoT específico para o controle de acesso, onde será considerado o monitoramento de parâmetros como a ID da Tag (Etiqueta) RFID. O

foco será a implementação dos sistemas integrados IoT e RFID no Firebase, que é uma plataforma de desenvolvimento de aplicativos do Google.

O presente trabalho irá explorar os aspectos técnicos relacionados à implementação do sistema de monitoramento IoT, como a seleção e integração de atuadores, a comunicação dos dados coletados e a visualização das informações em uma tabela de registros onde identifica o usuário, data e hora do acesso.

1.3 OBJETIVOS

1.3.1 Geral

Desenvolver um sistema de controle de acesso IoT com tecnologia RFID e Firebase.

1.3.2 Objetivos específicos

- a. Facilitar o acesso a locais por meio da conexão RFID
- b. Prover recursos de baixo custo para o desenvolvimento do sistema.
- c. Desenvolver o gerenciamento e controle do sistema.

2 REFERENCIAL TEÓRICO

Neste capítulo é apresentado um breve histórico de tecnologias que possibilitaram o desenvolvimento do sistema de controle de acesso proposto. Também são apresentados conceitos teóricos necessários para o desenvolvimento deste trabalho.

2.1 INTERNET DAS COISAS (IOT)

A Internet das Coisas é um conceito que tem ganhado cada vez mais relevância tanto no campo tecnológico quanto no meio acadêmico. A IoT se baseia na comunicação wireless, permitindo a conexão de diversos objetos, como

sensores, tags RFID, smartphones, computadores, e até objetos de uso cotidiano. Esses dispositivos geram um fluxo de dados, e a conexão entre eles possibilita a troca de informações, formando assim uma rede interligada, conhecida como Internet das Coisas (Wanzeler; Fulber; Merlin, 2016). A IoT refere-se a um sistema de interconexão de dispositivos físicos via internet, permitindo a coleta e o compartilhamento de dados de maneira autônoma. Esses dispositivos podem ser eletrodomésticos inteligentes, sensores, atuadores, entre outros.

Existem seis elementos fundamentais para entender e implementar soluções de IoT: identificação, sensores/atuadores, comunicação, computação, serviços e semântica (Al-Fuqaha, 2015).

Identificação: A identificação é essencial para que os dispositivos sejam reconhecidos e os serviços sejam correspondidos adequadamente às suas demandas. Cada dispositivo deve ser individualmente identificado, para que funcione como um nó exclusivo na rede, permitindo a coleta e a atuação específicas (Al-Fuqaha, 2015).

Sensores e Atuadores: Estes são elementos importantes no desenvolvimento de soluções IoT, responsáveis por coletar dados do ambiente e transmiti-los para os servidores de armazenamento, além de receber comandos e executar ações. Sensores capturam informações do contexto, enquanto atuadores promovem alterações no ambiente em que estão inseridos (Al-Fuqaha, 2015).

Comunicação: A comunicação entre os dispositivos IoT é possibilitada por tecnologias como WiFi, Bluetooth e LoRa, promovendo a conectividade necessária para o sistema (Al-Fuqaha, 2015).

Computação: Refere-se às unidades de processamento, como microcontroladores, microprocessadores, SoCs e FPGAs, bem como plataformas de hardware, como Arduino, Raspberry Pi e ESP32. Esses elementos são responsáveis pelo processamento dos dados e pelo controle dos dispositivos (Al-Fuqaha, 2015).

Serviços: Os serviços IoT podem ser classificados em quatro tipos principais: Serviços de Identificação, que identificam dispositivos físicos no ambiente digital; Serviços de Agregação de Informação, que coletam e sintetizam dados de sensores; Serviços de Colaboração e Inteligência, que utilizam esses dados para tomar decisões e agir conforme o contexto; Serviços Ubíquos, que têm como objetivo

fornecer soluções de colaboração e inteligência em qualquer lugar e a qualquer momento (Al-Fuqaha, 2015).

Semântica: A semântica está relacionada à capacidade de extrair conhecimento dos dados de forma inteligente, englobando a análise e o reconhecimento dos dados, além de associar demandas aos recursos apropriados (Al-Fuqaha, 2015).

Sintetizando as definições apresentadas, IoT é um conceito que promove a conexão de tudo e de todos, sejam reais ou virtuais, à internet. Isso permite que "coisas", como objetos, ambientes, soluções, serviços e aplicações, tornem-se mais inteligentes, facilitando o cotidiano das pessoas e aprimorando a integração entre o mundo físico e o virtual.

2.2 TECNOLOGIA DE IDENTIFICAÇÃO POR RADIOFREQUÊNCIA (RFID)

É um método de identificação através de sinais de rádio que recuperam e armazenam os dados à distância utilizando dispositivos chamados etiquetas RFID ou tags RFID. É utilizado em indústrias, logística, bibliotecas e supermercados devido à sua versatilidade, como é apresentado na Figura 1. Sua principal vantagem é a identificação sem contato direto, o que o diferencia de códigos de barras. É um objeto pequeno composto por uma antena e um microchip: enquanto a antena se comunica com as leitoras e sensores, o microchip pode armazenar dados tanto no formato de leitura quanto no formato de escrita, dependendo da tecnologia adotada.

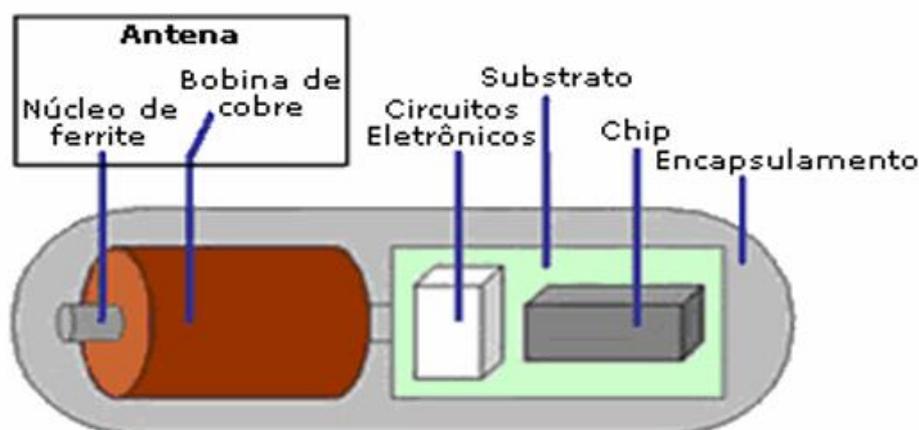
Figura 1 — Detalhes da utilização do RFID



Fonte: Santos (2018).

Uma etiqueta RFID, também denominada tag RFID, trata-se de um transponder que contém chips de silício e antenas, como ilustrado na Figura 2. Esses componentes permitem que a etiqueta responda aos sinais de rádio emitidos por uma base transmissora, o que ocorre principalmente nas etiquetas passivas. Além dessas, há também as etiquetas semi-passivas e ativas, que possuem a capacidade de emitir seus próprios sinais (Morais, 2015).

Figura 2 — Estrutura de um transponder



Fonte: Moraes (2015, p. 4).

Barsotti, Rahal e Silva (2020) falam com entusiasmo sobre o uso da tecnologia RFID em processos de monitoramento e rastreabilidade de itens. Segundo os autores, essa tecnologia é capaz de melhorar processos de gestão de estoque:

A utilização do RFID para monitorar os estoques pode prover maior previsibilidade às empresas, trazendo inúmeros benefícios, como melhorar a previsão sobre os níveis de estoque, reabastecimento imediato nas prateleiras, redução do número da quantidade de itens em estoque, e maior transparência na informação sobre demanda, além de melhorias na confiabilidade de entrega e disponibilidade de produtos (Barsotti; Rahal; Silva, 2020).

Além das aplicações em empresas, a tecnologia RFID também pode ser aplicada em controles de acesso de pessoas em residências, condomínios, escolas, hotéis, possibilitando acessos autorizados a esses locais, bem como o registro de horários de entrada e saída dos mesmos.

2.2.1 Princípio de funcionamento

O princípio do funcionamento da comunicação da tecnologia RFID é bastante simples. O sistema é composto por um transceptor (ou leitora) que transmite uma onda de frequência de rádio através de uma antena para um transponder, mais conhecido por tag. O tag absorve a onda de RF e responde com alguma informação, que é então gerenciada por um sistema computacional, como mostra a figura Figura 3. O termo transponder deriva da expressão TRANSMitter/resPONDER, que revela a função deste componente.

Figura 3 — Exemplo de sistema utilizando dispositivo leitor



Fonte: Adaptado pelo autor (2024), Batouty *et al.* (2020).

2.2.2 Tipos de etiquetas

As etiquetas RFID, ou simplesmente tags, podem ser classificadas de acordo com o seu modo de funcionamento, podendo estas serem do tipo passiva, semi-passiva ou ativa. Essa classificação diz respeito ao modo como essas etiquetas são alimentadas:

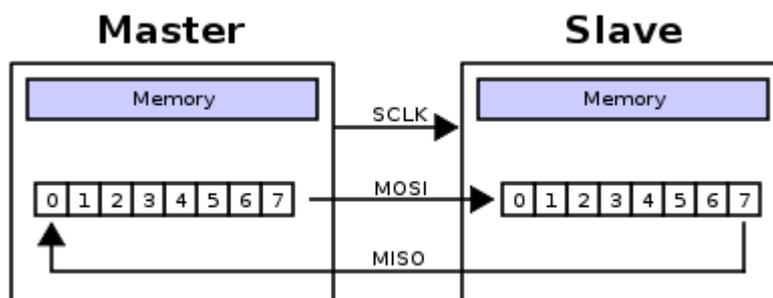
- Tag passiva: Uma tag passiva é um tipo de dispositivo utilizado em sistemas de Identificação por Radiofrequência (RFID) que não possui uma fonte de energia interna, como uma bateria. Em vez disso, ela depende da energia capturada a partir do sinal de rádio emitido por um leitor RFID para funcionar. Normalmente as tags passivas possuem memórias do tipo ROM (Read Only Memory), que possuem um código pré-gravado de fábrica e não podem ser modificadas, além de possuir longa vida útil e baixo custo (Oliani; Silva; Filho, 2014).
- Tag semi-passiva: Esse tipo de etiqueta RFID reúne características das etiquetas ativa e passiva. As tags semi-passivas são equipadas com uma fonte de energia própria, assim como a etiqueta ativa, mas neste caso, ao contrário das etiquetas ativas, a bateria não serve para iniciar uma transmissão de dados. Na semi-passiva, sua bateria serve para fornecer energia elétrica à etiqueta que auxilia na recepção da informação através das ondas de radiofrequência, desta forma, permitindo que este tenha maior capacidade de processamento (Oliani; Silva; Filho, 2014).

2.3 PROTOCOLO SERIAL PERIPHERAL INTERFACE

O SPI (*Serial Peripheral Interface*) é um protocolo de comunicação que permite a comunicação entre um microcontrolador e diversos componentes, formando uma rede. Trata-se de uma especificação de interface de comunicação série síncrona usada para curta distância, sendo amplamente utilizada em sistemas embarcados. Desenvolvido pela Motorola, o SPI se consolidou como um padrão no setor. Os dispositivos que utilizam o protocolo SPI se comunicam em modo full duplex, operando em uma arquitetura *master-slave* (mestre-escravo), na qual há apenas um dispositivo mestre que controla a comunicação. Esse dispositivo mestre é responsável por gerar os sinais necessários para as operações de leitura e escrita.

No contexto do protocolo SPI, tanto o mestre quanto o escravo possuem, internamente, um registro de deslocamento que se conecta por meio de duas linhas de dados. A ilustração na Figura exemplifica esse modelo de comunicação.

Figura 4 — Modelo de comunicação SPI

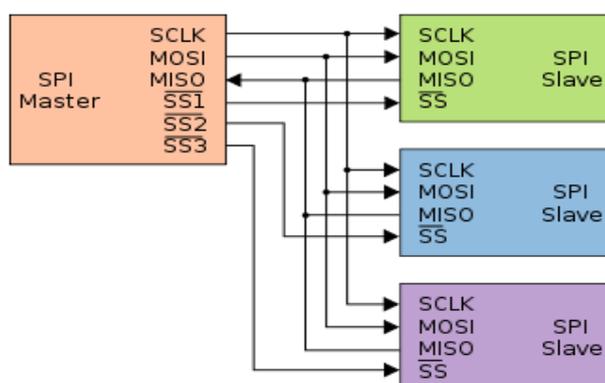


Fonte: Mendonça (2020).

2.3.1 Comunicação no Protocolo SPI

Os bits são deslocados em sincronia com o sinal de relógio (SCLK) gerado pelo dispositivo mestre. Para permitir a conexão de diferentes dispositivos escravos, o mestre controla o estado de linhas de seleção específicas como mostrada na Figura 5. O escravo cuja linha estiver ativa reconhece que os dados transmitidos pela linha MOSI (*Master Output Slave Input*) são direcionados a ele e pode, por sua vez, enviar dados pela linha MISO. Esse mecanismo evita conflitos nas linhas MISO (*Master Input Slave Output*), pois só um escravo estará ativo, enquanto os demais permanecem em alta impedância. A alta impedância indica que, em um ponto do circuito, a corrente que flui é relativamente baixa em comparação à tensão aplicada. A Figura 5 ilustra essa comunicação entre o mestre e os escravos.

Figura 5 — Modelo de comunicação SPI mestre e escravos



Fonte: Mendonça.

2.4 FIREBASE

De acordo com a documentação do Firebase, é uma plataforma fornecida pelo google que facilita o desenvolvimento de aplicações mobile e web. Entre seus principais serviços destaca-se o Realtime Database, um banco de dados não relacional (NoSQL) que possui otimizações e funcionalidades diferentes de um banco de dados relacional, visto que, ao contrário das tabelas utilizadas nos bancos de dados relacionais, possui seus dados armazenados como JSON, dados que são sincronizados em tempo real com todos os clientes conectados (Puffelen, 2016). Além disso, o Firebase é uma plataforma Baas(Backend as a service), que tem como objetivo facilitar o desenvolvimento da aplicação sem a necessidade de realizar configurações manualmente. De acordo com Sharma e Dand:

O BaaS fornece backend para aplicativos móveis, uma interface de programação e ferramentas para diferentes linguagens de programação se integrem ao backend do aplicativo (Sharma; Dand, 2019).

Como citado anteriormente, o Firebase fornece serviços implementados na base da própria ferramenta que permite desenvolvimento de uma aplicação de modo mais ágil, até de uma certa forma, pulando etapas necessárias se fosse utilizado outro serviço. Após a conexão da aplicação ao Firebase (etapa que envolve configuração do ambiente de desenvolvimento, apiKey, URL do banco de dados, entre outros), é possível dar início ao uso dessas ferramentas "prontas".

2.4.1 JSON e Realtime Database

Como citado no tópico acima, o Realtime Database armazena dados em uma estrutura JSON hierárquica, ou seja, permite criação de cada nó(ou *path*) seja um objeto acessível em tempo real. Através da configuração API fornecido no projeto, é possível ler, gravar e sincronizar informações automaticamente entre diferentes dispositivos. Cada elemento na árvore é um nó que pode conter tanto valores primitivos quanto objetos mais complexos como mostra a figura (Puffelen, 2016).

Figura 6 — Nó JSON



Fonte: Puffelen (2016).

2.5 HTML E CSS

Desde o início do uso do computador com a Internet, as páginas da web foram criadas usando a linguagem de marcação HTML (*HyperText Markup Language*). O HTML é essencial para criar e estruturar páginas web. Criado por Tim Berners-Lee e sua equipe do CERN (Organização Europeia para Pesquisa Nuclear) na década de 1990, o HTML desempenhou um papel vital no desenvolvimento da World Wide Web. Originalmente, o HTML permitia apenas marcação de texto simples, mas à medida que a web evoluía, novas funções foram adicionadas para permitir formatar texto, inserir imagens e criar hiperlinks. Com o HTML, os desenvolvedores podem definir os elementos estruturais de uma página, como títulos, parágrafos, listas e tabelas, que fornecem a base para um conteúdo visualmente atraente e navegável (Robbins, 2018).

Já o CSS (*Cascading Style Sheets*), é uma linguagem de estilo poderosa que funciona com HTML para controlar a aparência e apresentação de documentos da web, onde o HTML expressa a estrutura do documento e o CSS define a aparência da estrutura dos elementos da web. Originalmente proposto por Håkon Wium Lie em 1994 e posteriormente desenvolvido em colaboração com Bert Bos, o CSS permite aos desenvolvedores definir estilos consistentes e reutilizáveis para múltiplas páginas web (Robbins, 2018).

2.6 JAVASCRIPT

O JavaScript é uma linguagem de programação desenvolvida por Brendan Eich em 1995, enquanto ele trabalhava na Netscape Communications Corporation. Criada inicialmente para o navegador Netscape Navigator, sua finalidade era tornar processos em páginas web mais dinâmicos e agradáveis para os usuários. Um ano após seu lançamento, a Microsoft adaptou a linguagem para seu navegador, o que contribuiu para a consolidação do JavaScript como uma das tecnologias mais importantes e amplamente utilizadas na internet (Silva, 2015).

Os scripts escritos em JavaScript e executados em navegadores possibilitam a atualização de partes do conteúdo de uma página web sem a necessidade de recarregá-la completamente. Por exemplo, após o envio de um formulário, técnicas como AJAX (*Asynchronous JavaScript and XML*) permitem essa funcionalidade. Isso viabiliza a criação de uma ampla gama de softwares completos e totalmente funcionais para diversas finalidades (Silva, 2015).

2.7 BOOTSTRAP

O Bootstrap é um framework de código aberto para desenvolvimento HTML, CSS e Javascript. Com este framework, é possível acelerar e dinamizar o desenvolvimento web, pois existem diversos elementos prontos e estilizados disponíveis (Andrei I, 2023).

O sistema oferece uma grande variedade de temas e se integra a qualquer linguagem de programação. Seus padrões estão alinhados aos princípios de usabilidade e às tendências de design para interfaces. Essa padronização possibilita que os sites apresentem um visual amigável, agradável e atraente (Andrei I, 2023).

2.8 TRABALHOS RELACIONADOS

Na literatura existente, há diversos projetos e diferentes aplicações para a importância do controle de acesso. Um estudo relevante de Silva (2022) conduzido pela implementação de um sistema de controle de acesso ao Campus Campina Grande do Instituto Federal da Paraíba contribuiu ao mesmo, a substituição de

gerenciamento de controle de acesso manual de chaves físicas por um sistema automatizado, neste trabalho, o autor destaca a influência positiva de condições controladas pelo sistema, resultando em um melhor desempenho, além de desempenhar um processo rápido e não burocrático.

3 MATERIAIS E MÉTODOS

Neste capítulo será apresentada detalhadamente a metodologia de desenvolvimento do sistema de controle de acesso proposto neste trabalho de conclusão de curso. Cada etapa do processo será detalhada a seguir:

- a. Desenvolver a estrutura de hardware necessária para integrar leitores RFID, o microcontrolador ESP32, display e atuador, com o objetivo de gerenciar o controle de acessos.
- b. Implementar um sistema de comunicação de dados entre a ESP32 e o Realtime Database do Firebase, para armazenamento e gerenciamento das informações coletadas.
- c. Criar uma interface web para cadastro de usuários e monitoramento, permitindo a visualização dos acessos em tempo real.
- d. Testar e validar o funcionamento do sistema, com base na leitura e processamento dos dados dos cartões RFID.

3.1 MATERIAIS

3.1.1 Hardware

Hardware refere-se a todos os componentes físicos, sejam internos ou externos, de um computador ou celular, que definem as capacidades de um dispositivo e como ele pode ser utilizado. Embora o hardware precise de um software para operar (ou vice-versa), ele é um elemento distinto e de grande relevância.

Dessa forma, o hardware abrange qualquer componente ou equipamento físico, tanto no âmbito da Tecnologia da Informação como em outros contextos. Isso inclui desde máquinas e ferramentas, como uma fresadora ou uma chave inglesa,

até dispositivos como celulares e computadores. Itens como talheres, peças, dobradiças, engrenagens, parafusos, chips e processadores são todos considerados hardware.

No contexto da Tecnologia da Informação, o termo se aplica aos componentes gerais de dispositivos, incluindo processadores, placas-mãe, memória RAM, unidades de armazenamento (como HDs, SSDs e memória flash), além de dispositivos de entrada e saída, como teclado, mouse, monitores, caixas de som, controles remotos e videogames.

3.1.1.1 ESP32

Neste trabalho optou-se pelo uso da Placa ESP-WROOM-32 WiFi com 30 Pinos, que é um microcontrolador desenvolvido pela empresa chinesa Expressif Systems, esse se trata de uma evolução da ESP8266, trazendo mais recursos e capacidades de processamento, se trata de um circuito totalmente integrado e que permite a conexão com redes WiFi e comunicação via protocolo TCP/IP . O ESP32 é constituído por um robusto processador, que pode ser single ou dual-core de 32 bit (com dois núcleos físicos de processamento), Além de ser um microcontrolador de baixo custo e ter potência de processamento superior a outros microcontroladores, é bastante comum que seja utilizado em projetos acadêmicos e pesquisas. Também um limitador de tensão de 5V para proteção caso a alimentação exceda o valor recomendado (Silva, 2021).

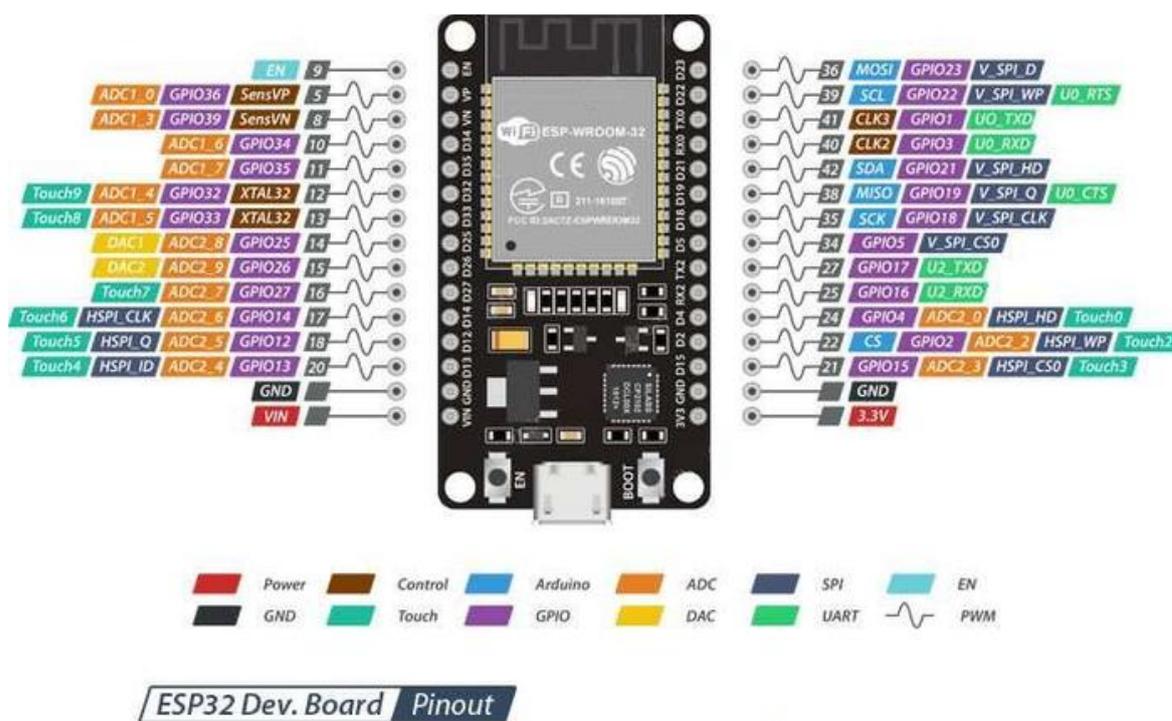
Figura 7 — ESP32



Fonte: O autor (2024).

É importante observar os pinos do ESP32 e como eles podem ser utilizados, os pinos GPIO (*General Purpose Input Output*) são portas que servem como interface de integração entre o microcontrolador e periféricos, tais como sensores e atuadores, ou também outros circuitos integrados e microcontroladores. Na Figura 8 é possível ver a pinagem de um ESP32 - WROOM. Possui um total de 30 pinos GPIO's com níveis de tensão 0V e 3.3V e para níveis digitais 0 e 1 respectivamente, com corrente máxima de 12mA por GPIO.

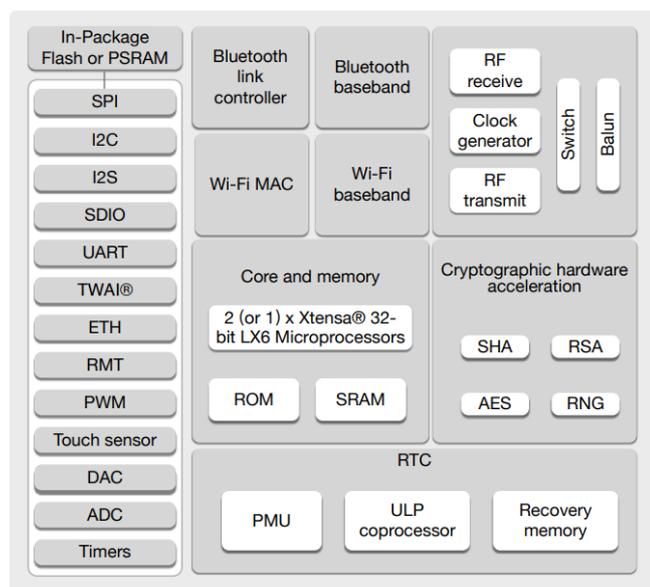
Figura 8 — Pinagem do Esp32



Fonte: Albuquerque (2021).

Os periféricos presentes no módulo ESP32 podem ser melhor visualizados no diagrama de blocos como ilustrado na Figura 9, imagem que pode ser facilmente encontrada em ficha de dados.

Figura 9 — Diagrama de blocos funcionais



Fonte: ESPRESSIF SYSTEMS (2024).

O ESP32 pode ser programada em variadas linguagens de programação, as mais comuns são: C/C++, pois a maioria dos frameworks e bibliotecas são escritos em C/C++ tornando uma escolha popular para o desenvolvimento na plataforma por possuir maior quantidade de recursos e suporte disponíveis; Arduino, a ESP32 é compatível com a plataforma Arduino IDE, inclusive sendo a plataforma de estudo para esse trabalho, outras linguagens como MicroPython, Lua e JavaScript.

Na Tabela 1 pode ser observado algumas das especificações técnicas da Esp32.

Tabela 1 — Dados técnicos ESP32

Dimensões	27,5 x 51,0 x 7,0 mm
Tensão de Alimentação	4,5 a 12V DC
Tensão de Nível Lógico	3,3 V DC
Corrente de Consumo	80 à 500 mA
Temperatura de Operação	-40 à 85 C°

Fonte: ESPRESSIF SYSTEMS (2024).

3.1.1.2 Módulo RC522

No presente trabalho foi utilizado o módulo RC522. É importante citar que ao utilizar o RC522 é essencial para projetos que exigem identificação sem contato, como controle de acesso, automação e rastreamento de objetos. Também conhecido como MFRC522, é um dispositivo compacto e acessível amplamente utilizado em projetos de eletrônica para leitura e gravação de tags RFID, operando na frequência de 13,56 MHz. Equipado com o chip MFRC522, ele se comunica com microcontroladores, como o ESP32, através da interface SPI. Além de sua facilidade de integração e baixo custo, o RC522 oferece suporte ao padrão ISO/IEC 14443, permitindo a leitura de cartões e etiquetas compatíveis com a tecnologia NFC. Tem como princípio de funcionamento uma antena embutida no módulo que gera um campo eletromagnético. Esse campo é utilizado para ativar as tags RFID passivas que estão próximas ao módulo, quando a tag é aproximada, o módulo envia um comando de leitura para a tag RFID, recebe e transmite dados ao microcontrolador via SPI.

Esse módulo possui 8 pinos, mas, para o presente projeto serão utilizados 7 pinos. Sendo 2 pinos para trocas de dados, como MOSI enviar dados ao microcontrolador e MISO para receber dados do microcontrolador. Outros 2 pinos são SDA (Serial Data) para iniciar e encerrar a comunicação e, SCK (Serial Clock) para sincronizar a comunicação entre o módulo e o microcontrolador. Além do mais, são utilizados os pinos 3.3V (Alimentação), GND (Terra) e RST (Reset) respectivamente, para alimentação e reset do sistema.

Figura 10 — Módulo RC522



Fonte: Autor (2024).

Tabela 2 — Especificações técnicas do módulo RC522

Tensão de Alimentação	2,5V a 3,3V (Alimentação típica de 3,3V)
Frequência de Operação	13,56 MHz
Interface de Comunicação	SPI, I2C e UART
Distância de Leitura	2 cm a 5 cm
Temperatura de Operação	-20° a +80°C
Antena Integrada	Embutida no próprio módulo

Fonte: NXP Semiconductors (2016).

3.1.1.3 MICRO SERVO

O Micro servo motor SG90, ilustrado na Figura 11, é muito utilizado em projetos de robótica e em sistemas microcontrolados. Esse módulo permite movimentos proporcionais aos comandos recebidos, controlando tanto a rotação quanto a posição. Como um dispositivo de malha fechada, o servo motor opera recebendo um sinal de controle, avaliando sua posição atual e ajustando-se para alcançar a posição desejada. Sendo compacto e leve, o Micro servo SG90 pesa apenas 9 gramas e oferece um torque máximo de aproximadamente 1,6 kg (Maker Hero, 2019).

Figura 11 - Micro servo Tower Pro SG90



Fonte: Autor (2024).

3.1.2 Software

3.1.2.1 ARDUINO IDE

Para o desenvolvimento do projeto o ambiente de desenvolvimento integrado escolhido foi o Arduino IDE, que é uma plataforma de prototipagem eletrônica de código aberto, que permite a criação de projetos interativos de maneira simplificada e acessível, por ser compatível com a ESP32 se tornou uma boa escolha para fazer a programação do projeto, além do mais, é um ambiente que se resume em variáveis declaradas, as instruções são inseridas dentro das funções de “setup()” e “loop()”. Este software é baseado na linguagem de programação C/C++ e oferece inúmeras funções e bibliotecas que facilitam o desenvolvimento de projetos.

Algumas das características principais da interface Arduino IDE, possui uma interface simples, além de possuir uma alta compatibilidade com diversos sistemas operacionais, como Windows, Linux e macOS.

Figura 12 — Interface Arduino IDE



Fonte: Autor (2024).

3.1.2.4 NOTEPAD++

Notepad++ é um software de código aberto, comparado ao bloco de notas do sistema operacional Windows, seu diferencial é suportar diversas linguagens de programação voltado ao desenvolvimento web, além do mais, é um software extremamente leve, servindo como um bom recurso para computadores de baixo custo (Santos).

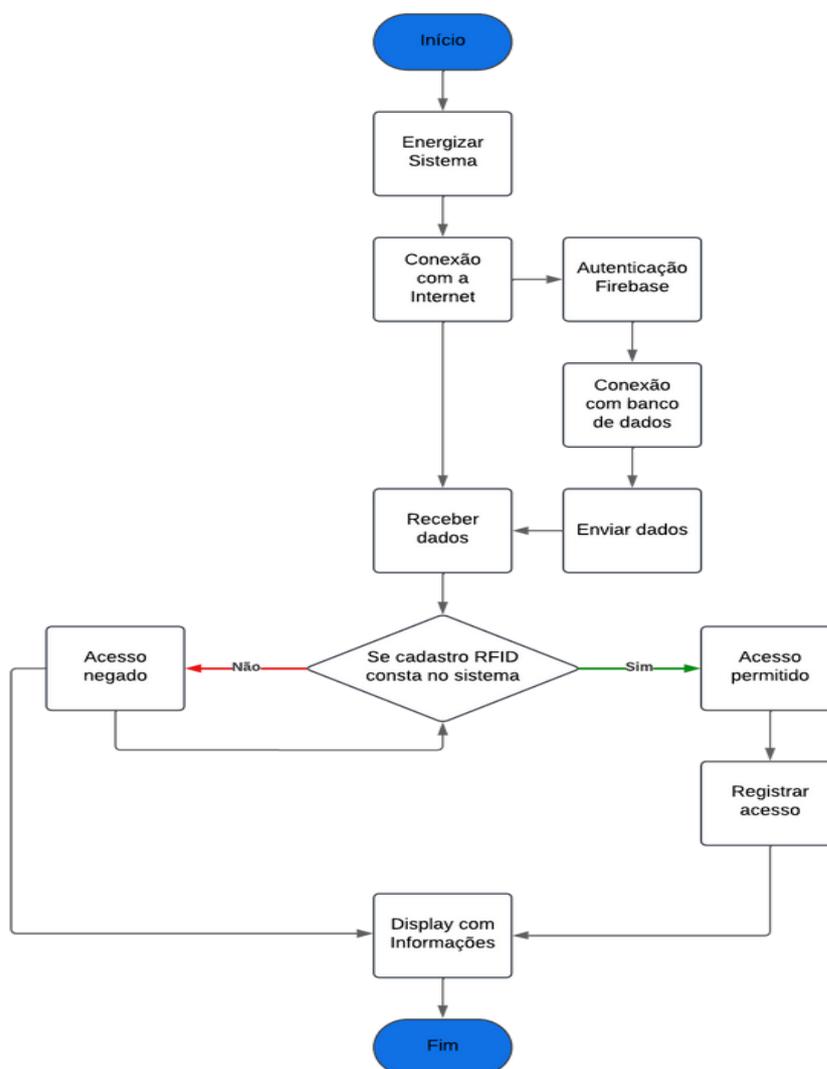
3.2 MÉTODOS

3.2.1 Arquitetura do projeto

O fluxograma descreve o funcionamento de um sistema de controle de acesso baseado em RFID, integrado ao Firebase para armazenamento de dados. O processo começa com a inicialização do sistema ao ser energizado, seguido pela conexão à Internet e realiza autenticação no Firebase. Em seguida, o sistema é integrado ao banco de dados, sendo assim, permitindo a leitura e escrita das informações necessárias.

Assim que está pronto para operar, o sistema começa a receber os dados dos cartões RFID para verificar permissões de acesso. Ao ler um cartão, ele verifica se o cadastro do RFID está no sistema. Caso o cadastro exista, o acesso é permitido, exibe “Acesso Permitido” no display e o evento é registrado no banco de dados. Se o RFID não estiver cadastrado, o sistema nega o acesso e exibe “Acesso Negado” no display. Dessa forma, esse ciclo se repete a cada nova leitura, garantindo que o sistema monitore e registre acessos como mostra a Figura 13.

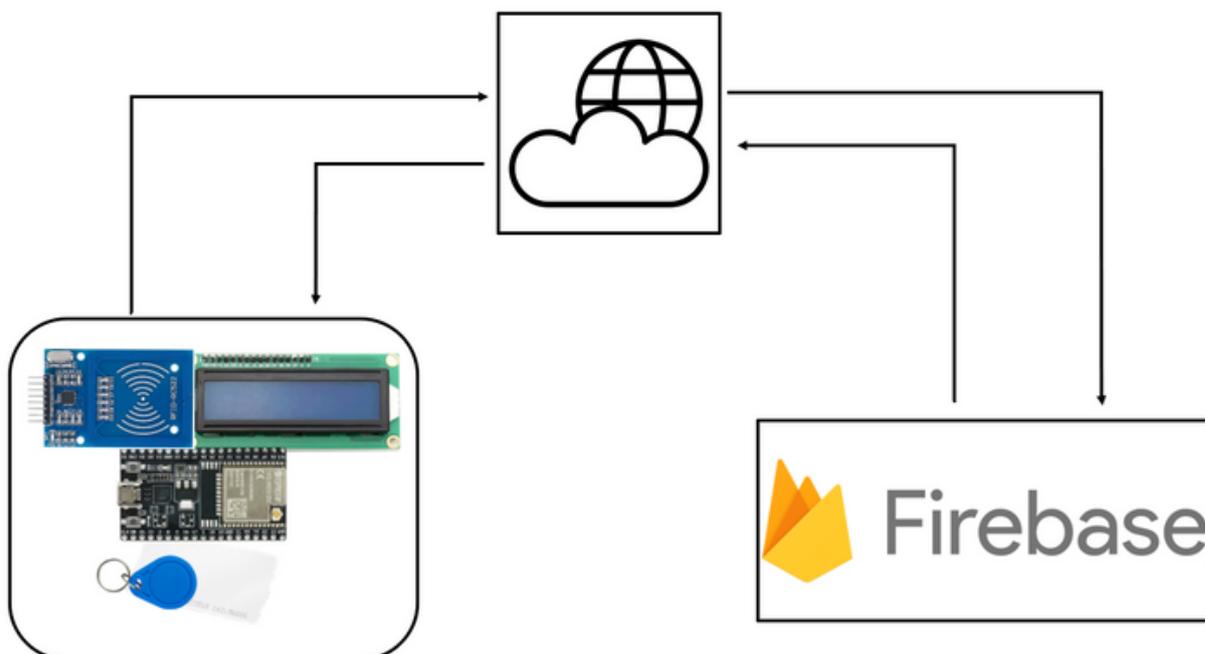
Figura 13 — Fluxograma



Fonte: Autor (2024).

A Figura 14 a seguir, demonstra a arquitetura do sistema de controle de acesso, o ESP32 recebe e envia as informações de UID lidas para o Firebase, onde o sistema verifica se o usuário tem permissão para acessar, armazenando os registros no banco de dados.

Figura 14 — Arquitetura do sistema



Fonte: Autor (2024).

Este esquema e o diagrama do protótipo desenvolvido no software Fritzing mostrado na Figura 15 e na Figura 16, descrição das conexões:

ESP32:

- É o microcontrolador principal do projeto.
- Está alimentado pela porta USB e fornece energia ao circuito através de seus pinos de 3.3V, 5V e GND.

Módulo RFID-RC522:

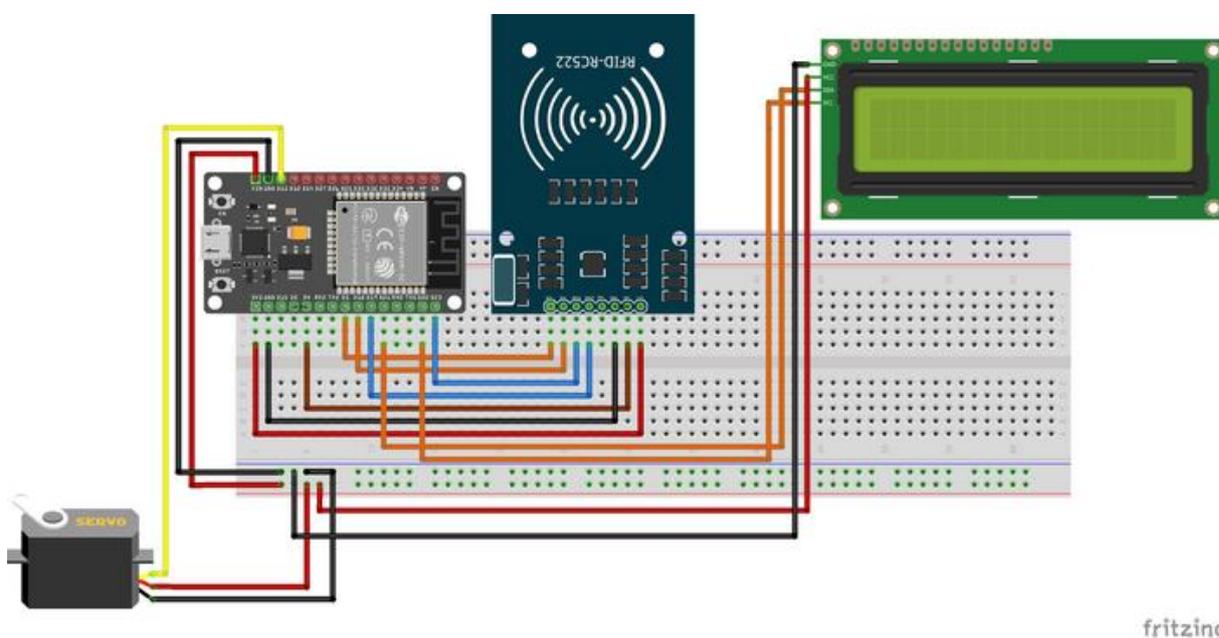
- SDA do RFID ao pino GPIO 5 do ESP32.
- SCK ao pino GPIO 18.
- MOSI ao pino GPIO 23.
- MISO ao pino GPIO 19.
- IRQ não está conectado, pois não é necessário neste caso.
- GND do RFID ao GND do ESP32.
- RST ao pino GPIO 2.

- VCC do RFID ao pino 3.3V do ESP32.

Display LCD I2C:

- GND do LCD ao GND do ESP32.
- VIN VCC ao pino 5V do ESP32.
- SDA ao pino GPIO 21.
- SCL ao pino GPIO 22.

Figura 15 — Esquema do protótipo



Fonte: Autor (2024).

Figura 17 — Código de Configuração

```
const firebaseConfig = {  
  apiKey: "SUA_API_KEY",  
  authDomain: "SEU_DOMINIO.firebaseio.com",  
  databaseURL: "https://SEU_PROJETO.firebaseio.com",  
  projectId: "SEU_PROJECT_ID",  
  storageBucket: "SEU_BUCKET",  
  messagingSenderId: "SEU_SENDER_ID",  
  appId: "SEU_APP_ID"  
};
```

Fonte: Autor (2024).

SDK: O SDK do Firebase é um conjunto de bibliotecas disponibilizado pelo Firebase que simplifica a utilização dos serviços em suas aplicações.

Figura 18 — Importação SDK

```
<script src="https://www.gstatic.com/firebasejs/9.1.3/firebase-app.js"></script>  
<script src="https://www.gstatic.com/firebasejs/9.1.3/firebase-database.js"></script>
```

Fonte: Autor (2024).

3.2.2.1 PLATAFORMA WEB

Neste tópico, apresenta a criação da interface web com o objetivo de realizar cadastros, editar cadastros e monitorar acessos, utilizando o framework Bootstrap para manter a interface responsiva para ser adaptada em diferentes dispositivos.

Tela Inicial: A aplicação foi construída de forma a ter uma interface simples e intuitiva para que seja acessível aos usuários. A tela inicial, ilustrada na Figura 19, possui um menu de fácil acesso, havendo três opções que deseja utilizar.

Figura 19 — Tela Inicial



Fonte: Autor (2024).

Tela de cadastro: Nesta tela de cadastro, ilustrada na Figura 20, são registrados o nome, o UID e o gênero com duas opções disponíveis, como mostrado na figura. No entanto, o campo de UID não é editável, pois o ID do cartão RFID é enviado pelo ESP32 ao aproximar o cartão do módulo RC522.

Figura 20 — Tela de cadastro

A tela de cadastro "Adicionar Usuário" contém os seguintes campos:

- Nome:** Campo de texto com o placeholder "Nome de Usuário".
- UID do Cartão:** Campo de texto com o valor "697fbe18".
- Gênero:** Opções de rádio button para "Feminino" e "Masculino".

Um botão de ação "Adicionar Usuário" em cor verde está posicionado na base da tela.

Fonte: Autor (2024).

Tela de editar cadastro: Após o cadastro, é possível editar ou excluir o usuário, ilustrada na Figura 21. Na edição, é possível modificar o nome, o gênero e atualizar o UID do RFID ao passar um novo cartão pelo módulo RC522.

Figura 21 — Tela de editar ou excluir cadastro

Editar ou Excluir Usuário

Nome	UID do Cartão	Gênero	Ações
assuncao	608bcf55	Masculino	Edit Excluir

Fonte: Autor (2024).

Visualização de acessos: Na tela de monitoramento de acessos, é possível visualizar a data e hora de cada acesso realizado por usuários cadastrados, como mostra a Figura 21.

Figura 21 — Monitorar acessos

Monitorar Acessos

Nome	UID	Gênero	Data e Hora
assuncao	608bcf55	Masculino	31/10/2024 21:48:51
assuncao	608bcf55	Masculino	31/10/2024 21:57:03

Fonte: Autor (2024).

3.2.2.2 DESENVOLVIMENTO ESP32

O desenvolvimento do código foi feito através da Arduino IDE, versão 2.3.2. Para comunicação e processamento de dados, foram utilizadas bibliotecas oficiais e de terceiros para simplificar o desenvolvimento, algumas dessas bibliotecas estão sendo mostradas na Figura 22.

Figura 22 — Bibliotecas

```
#include <WiFi.h>
#include <FirebaseESP32.h>
#include <SPI.h>
#include <MFRC522.h>
#include <NTPClient.h>
#include <LiquidCrystal_I2C.h>
#include <ESP32Servo.h>
```

Fonte: Autor (2024).

Através da biblioteca “FirebaseESP32.h”, para a comunicação entre o Realtime Database e ESP32, é necessário realizar autenticação por meio de “FIREBASE_HOST” e “FIREBASE_AUTH”, como mostra a Figura 23.

Figura 23 — Configuração do Firebase

```
#define FIREBASE_HOST "https://seu-projeto.firebaseio.com"
#define FIREBASE_AUTH "seu_token_de_autenticacao"
```

Fonte: Autor (2024).

Após a autenticação, podemos comunicar com o banco de dados, realizando as seguintes operações, como: leitura e gravação de dados. No entanto, para gerenciar estes dados, é necessário configurar as regras do Firebase para leitura e gravação, como mostra a Figura 24.

Figura 24 — Regras do Firebase

```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

Fonte: O autor (2024).

Loop: Em "loop()", são executadas ações repetitivas, onde uma condição verifica se o cadastro existe no caminho do nó “/acces_control/ + uid”. Caso o cadastro exista, a função “logAccess()” é chamada e permite acesso por meio da mensagem exibida no LCD e aciona o servo; caso contrário, a função “sendUIDToAddUser()” é acionada e também exibe a mensagem “Acesso Negado” no LCD.

Figura 25 — Código em loop

```
String userPath = "/access_control/" + uid;
if (Firebase.getString(firebaseData, userPath + "/uid")) {
  logAccess(uid); // Registrar acesso se o UID estiver cadastrado
} else {
  sendUIDToAddUser(uid); // Enviar UID temporário

  lcd.setCursor(0, 0);
  lcd.print("Acesso");
  lcd.setCursor(0, 1);
  lcd.print("Negado");
  delay(2000); // Aguarda 2 segundos para exibir o UID
  lcd.clear();
}
}
```

Fonte: O autor (2024).

sendUIDToAddUser: É criada a função “sendUIDToAddUser” para enviar o ID do RFID para a tela de cadastro citada no tópico 3.2.2.1.

Figura 26 — Função para cadastro

```
void sendUIDToAddUser(String uid) {
  if (Firebase.setString(firebaseData, "access_control/tempUID", uid)) {
    Serial.println("UID enviado para adicionar usuário: " + uid);
  } else {
    Serial.println("Falha ao enviar UID para adicionar usuário: " + firebaseData.errorReason());
  }
}
}
```

Fonte: Autor (2024).

Ao realizar a leitura do cartão ao módulo RC522, é chamada a função “sendUIDToAddUser”, tem como objetivo enviar o “UID” para o nó “tempUID” a cada leitura, como mostra a Figura 27.

Figura 27 — ID do RFID enviado à variável tempUID



Fonte: Autor (2024).

Após este procedimento, no código HTML utiliza a função `onValue`, que tem como objetivo monitorar o caminho “access_control/tempUID” no Firebase e executa uma ação sempre que o valor muda. O parâmetro “snapshot” traz os dados atuais desse caminho, e “snapshot.val()” retorna o valor atualizado do UID. Isso permite que o valor do UID apareça automaticamente na caixa de texto UID da tela de cadastro sempre que o cartão é lido, conforme ilustrado na Figura 28 e na Figura 29.

Figura 28 — Captura do UID em Firebase

```

const tempUIDRef = ref(database, 'access_control/tempUID');
onValue(tempUIDRef, (snapshot) => {
  const uid = snapshot.val();
  if (uid) {
    document.getElementById('CardID').value = uid;
  }
});

```

Fonte: Autor (2024).

Figura 29 — UID enviado à caixa de texto

Nome:	<input type="text" value="Nome de Usuário"/>
UID do Cartão:	<input type="text" value="697fbe18"/>

Fonte: Autor (2024).

"logAccess": Outra função criada "logAccess", tem como objetivo capturar os dados de um determinado cadastro e registrar para o nó "log".

Figura 30 — Captura de dados em determinado nó

```
void logAccess(String uid) {
    String userPath = "/access_control/" + uid;

    Firebase.getString(firebaseData, userPath + "/uid");
    String rfid = firebaseData.stringData();

    Firebase.getString(firebaseData, userPath + "/name");
    String name = firebaseData.stringData();

    Firebase.getString(firebaseData, userPath + "/gender");
    String gender = firebaseData.stringData();
}
```

Fonte: Autor (2024).

Serão coletadas todas as informações do cadastro por meio da função do objeto "Firebase.getString()", além de utilizar a função da biblioteca "NTPClient.h" para sincronizar o horário com um servidor NTP para obter data e hora atualizados. Com isso, o registro do acesso é criado no Firebase para o nó "log", conforme ilustrado na Figura 31 e na Figura 32.

Figura 31 — Gerar log

```
// Sincroniza com o servidor NTP para obter a data e hora atuais
if (!timeClient.update()) {
    | timeClient.forceUpdate();
}

// Formata o timestamp
String timeStamp = String(timeClient.getFormattedTime());

// Registra as informações no caminho do log
Firebase.setString(firebaseData, logPath + "/nome", name);
Firebase.setString(firebaseData, logPath + "/uid", rfid);
Firebase.setString(firebaseData, logPath + "/genero", gender);
Firebase.setString(firebaseData, logPath + "/timestamp", timeStamp);
```

Fonte: Autor (2024).

Figura 32 — Nó em registro de acesso



Fonte: Autor (2024).

Com isso, podemos gerar a captura dos dados como na tela de cadastro, utilizando a função do objeto “snapshot”, a variável “arr” receberá este array (matriz), e assim, podemos percorrer por meio do mesmo para gerar a tabela de acesso, como mostra na Figura 33 e na Figura 34.

Figura 33 — Captura da Array

```

arr.forEach(function(elem) {
  var row = `<tr>
    <td>${elem.nome || "Nome não definido"}</td>
    <td>${elem.uid || "UID não definido"}</td>
    <td>${elem.genero || "Gênero não definido"}</td>
    <td>${new Date().toLocaleDateString()} ${elem.timestamp || "Hora não definida"}</td>
  </tr>`;
  accessTableBody.innerHTML += row; // Adiciona a nova linha à tabela
});

```

Fonte: Autor (2024).

Figura 34 — Registro de acessos

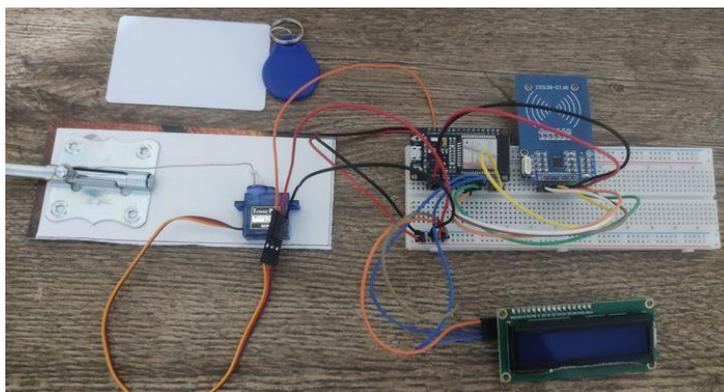
Nome	UID	Gênero	Data e Hora
teste	608bcf55	Masculino	31/10/2024 21:48:51
teste	608bcf55	Masculino	31/10/2024 21:57:03
teste	608bcf55	Masculino	31/10/2024 22:15:57

Fonte: Autor (2024).

4 RESULTADOS

Neste capítulo são apresentados os resultados deste projeto, conforme discutido nos capítulos anteriores. A Figura 35 mostra o circuito montado.

Figura 35 — Circuito montado

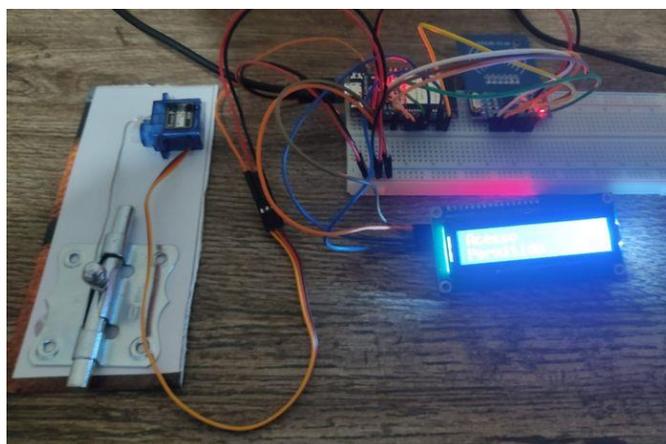


Fonte: Autor (2024).

Na etapa de projeto, foi definido que o ESP32 seria responsável por receber a resposta do banco de dados hospedado na nuvem. A validação completa da UID seria realizada diretamente no banco, enquanto o microcontrolador teria a função de apenas ler a UID da tag, enviá-la para autenticação e receber a permissão indicando aprovação ou reprovação da validação.

A leitura será realizada com a aproximação da tag ao leitor como na Figura 36.

Figura 36 — Tag válida



Fonte: Autor (2024).

Quando uma etiqueta válida, ou seja, cadastrada no banco de dados, for aproximada, o sistema exibirá uma mensagem no display LCD e aciona o Micro servo para liberar entrada, conforme ilustrado na Figura 36. Em seguida, o microcontrolador buscará os dados correspondentes no banco de dados e os enviará como registro de acesso. O monitor serial confirmará o sucesso da operação exibindo uma mensagem de conclusão conforme a Figura 37.

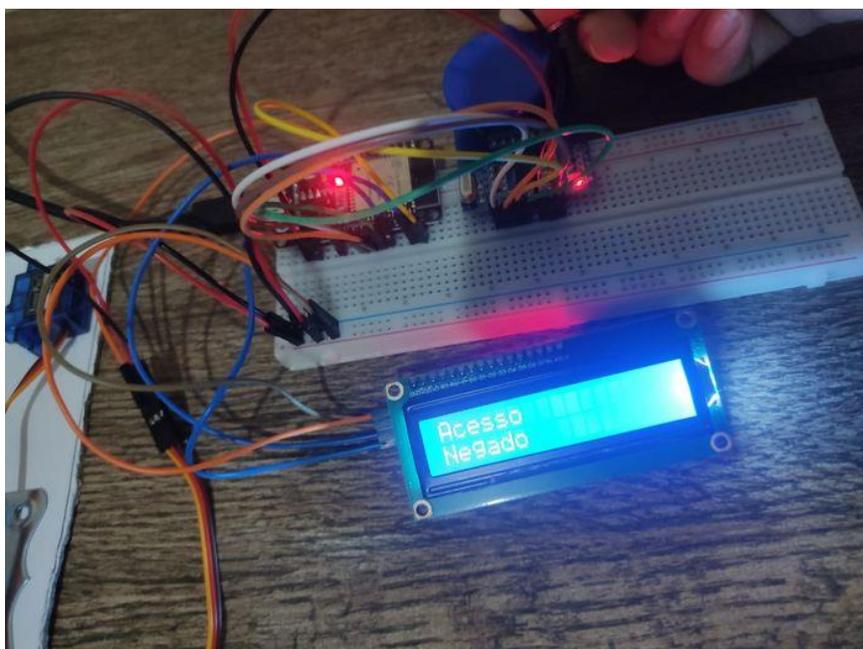
Figura 37 — Confirmação de registro

```
Conectado ao Wi-Fi!  
Conexão com Firebase OK  
Log registrado com sucesso!
```

Fonte: Autor (2024).

Caso uma etiqueta não esteja cadastrada no banco de dados, o sistema exibirá uma mensagem no display LCD negando o acesso, e o acionamento do servo motor não ocorrerá.

Figura 38 — Acesso negado



Fonte: Autor (2024).

Se nenhuma etiqueta for detectada, o sistema continuará exibindo no LCD a mensagem "Aproxime o cartão", enquanto o banco de dados e o microcontrolador permanecerão em estado de espera pela aproximação de uma tag.

4.1 CUSTO DO PROJETO E COMPARAÇÃO

Neste tópico aborda os custos envolvidos na realização do projeto, detalhando os recursos financeiros necessários para sua execução. Além disso, é feita uma comparação entre os mesmos de outras empresas.

Tabela 3 — Custo do protótipo

Produto	Preço
ESP32 WROOM-32 WiFi - Driver CH9102X - 30 Pinos	R\$39,75
Display LCD 16X2 Azul com Módulo I2C Soldado	R\$32,90
Kit Leitor RFID RC522 Cartão Tag	R\$18,50
Micro Servo 9g	R\$16,50
Protoboard 830 Furos	R\$24,00
Kit Jumpers	R\$22,00
Total	R\$153,65

Fonte: Autor (2024).

Tabela 4 — Produtos semelhantes de empresas

Generic	Kit De Controle De Acesso Rfid De 125 Khz Wg26/34	R\$231,64
Intelbras	Controle Acesso Intelbras Digiprox Sa202	R\$235
Intelbras	Controlador De Acesso 13,56mhz Sa 211 Mf Ip66	R\$521

Fonte: Autor (2024).

Como pode ser observado entre a Tabela 3 e Tabela 4, a comparação entre o protótipo desenvolvido e os produtos do mercado demonstra que o protótipo oferece uma solução mais acessível e custo relativamente mais barato. Enquanto os dispositivos disponíveis no mercado apresentam maior robustez e são voltados para

aplicações profissionais, o protótipo destaca-se como uma ótima alternativa econômica.

5 CONCLUSÃO

A realização deste projeto alcançou com sucesso seu objetivo principal, permitindo a autenticação correta dos usuários para a liberação ou não de acessos. O sistema web integrado ao Firebase também demonstrou uma boa funcionalidade no gerenciamento de cadastros, permitindo a edição e exclusão de registros de forma, bem como a visualização de acessos via interface web.

Os testes realizados comprovaram a obtenção de informações em tempo real ao autenticar os dados com o banco de dados na nuvem. O funcionamento estável e a conexão confiável entre os dispositivos demonstraram que o sistema foi capaz de executar todos os comandos.

Os resultados foram bem satisfatórios já que atenderam aos objetivos propostos e obteve as funcionalidades esperadas. Cada um dos componentes trabalhados nesse trabalho, além do sistema web ser intuitivo e simples para o usuário final. Esses fatos demonstram que os benefícios da IoT tendem a melhorar a confiabilidade da troca de informações entre o ambiente virtual e o físico.

5.1 TRABALHOS FUTUROS

- Interface web para novos usuários realizar o próprio cadastro.
- Autenticação de dois fatores
- Geração de relatórios por data
- Correção de pequenos bugs, como: UID permanecer na caixa de texto do UID da página tela de cadastro mesmo após a realização do cadastro, entre outros.

REFERÊNCIAS

- AL-FUQAHA, Ala et al. Internet of things: A survey on enabling technologies, protocols, and applications. **IEEE communications surveys & tutorials**, v. 17, n. 4, p. 2347-2376, 2015.
- ALBUQUERQUE, Yure. **ESP32 pinout**: Guia Básico de GPIOs. Smartkits. 2021. Disponível em: <https://blog.smartkits.com.br/esp32-pinout-guia-basico-de-gpios/>. Acesso em: 10 out. 2024.
- ANDREI L. **Desvendando o Bootstrap**: O que é e como usar?. Hostinger. 2023. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-bootstrap>. Acesso em: 29 out. 2024.
- BARSOTTI, C. F; RAHAL, E. J; SILVA, M. C. **Análise dos Benefícios da Gestão de Estoque por Meio de Sensores RFID em um Contexto de Indústria**. 2020. 20 p Trabalho de Conclusão de Curso (Engenharia Civil) - Universidade Presbiteriana Mackenzie, São Paulo, 2020.
- BATOUTY, EL *et al.* **Improvement of Radio Frequency Identification Security Using New Hybrid Advanced Encryption Standard Substitution Box**. MDPI. 2020. Disponível em: https://www.mdpi.com/2079-9292/9/7/1168?type=check_update&version=1. Acesso em: 9 out. 2024.
- CARDOSO, R. L. d. B. S. **Desenvolvimento de soluções para acionamento eletrônico de fechaduras "keyless"**: Rfid, keypad e biométrico. 2014.
- ESPRESSIF SYSTEMS. **ESP32 Series Datasheet**. Shanghai, 2024. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Acesso em: 17 out. 2024.
- FINKENZELLER, K. RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication. **John Wiley & Sons**, 2010.
- MAKER HERO. **Micro servo motor SG90 9g Tower Pro**. 2019. Disponível em: <https://www.makehero.com/produto/micro-servo-9g-sg90-towerpro/>. Acesso em: 29 nov. 2024.
- MENDONÇA, Hélio. **HSM. SPI e I2C**. Universidade do Porto. Disponível em: <https://paginas.fe.up.pt/~hsm/misc/old/comp/spi-e-i2c/>. Acesso em: 6 set. 2024.

MORAIS, Rodrigo. **Sistema de controle de acesso ao mestrado em tecnologia da energia utilizando RFID**, f. 41. 2015. 25 p Trabalho de Conclusão de Curso (Bacharel em Engenharia de Computação) - Escola Politécnica de Pernambuco - Universidade de Pernambuco, 2015.

NXP SEMICONDUCTORS. **MFRC522**: contactless reader IC. 2016. 95 p. Disponível em: <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>. Acesso em: 20 out. 2024.

OLIANI, Rogéria; SILVA, Alexandre César R; FILHO, Tércio Alberto dos Santos. Uma abordagem sobre segurança em sistemas RFID. **Anais do Encontro Anual de Computação**. 8 p, 2014. Disponível em: <https://www.enacomp.com.br/biblioteca-de-publicacoes/get-file.php?id=20736>. Acesso em: 9 set. 2024.

PUFFELEN, Frank van. **Have you met the Realtime Database?**. Firebase. 2016. Disponível em: <https://firebase.blog/posts/2016/07/have-you-met-realtime-database>. Acesso em: 29 out. 2024.

ROBBINS, Jennifer. **Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics**. Sebastopol: CA: O'Reilly Media, 2018.

SANTOS, Helson. **Etiqueta RFID: o que é, como funciona e como implementar**. Medium. 2018. Disponível em: <https://medium.com/@helsonsantos/etiqueta-rfid-o-que-%C3%A9-como-funciona-e-como-implementar-d8f42b9a40aa>. Acesso em: 2 set. 2024.

SANTOS, Patrick M. **Notepad++**. Bookdown. Disponível em: https://bookdown.org/patrickcefet/eletronica_geral/notepad.html. Acesso em: 20 out. 2024.

SHARMA, Divya; DAND, Hiren. Firebase as BaaS for College Android Application. International Journal of Computer Applications. **International Journal of Computer Applications**, v. 178, n. 20. 6 p, 2019.

SILVA, Alfredo Rodrigo Sousa da. **IF Access**: Sistema de controle de acesso eletrônico utilizando tecnologia RFID e microcontrolador. Campina Grande, PB, 2022. 47p. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) – Instituto Federal da Paraíba, Campus Campina Grande.

SILVA, Giancarlo. **O que é e como funciona a linguagem JavaScript**. Canaltech. 2015. Disponível em: <https://canaltech.com.br/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/>. Acesso em: 29 out. 2024.

SILVA, Saymon Rafael da Cunha. **Sistema de monitoramento e controle de energia utilizando o microcontrolador ESP32**. 2021 Trabalho de Conclusão de Curso - Universidade Tecnológica Federal do Paraná, 2021.

WANZELER, Tiago; FULBER, Heleno; MERLIN, Bruno. Desenvolvimento de um sistema de automação residencial de baixo custo aliado ao conceito de Internet das Coisas (IoT). **XXXIV Simpósio Brasileiro de Telecomunicações**, Pará, p. 40-44, 2016.

APÊNDICE A — CÓDIGO DO ESP32

Apêndices tem objetivo de melhorar a compreensão textual, ou seja, completar ideias desenvolvidas no decorrer do trabalho.

```
#include <WiFi.h>
#include <FirebaseESP32.h>
#include <SPI.h>
#include <MFRC522.h>
#include <NTPClient.h>
#include <LiquidCrystal_I2C.h>
#include <ESP32Servo.h>

#define FIREBASE_HOST "https://esp32rfid-1f741-default-
rtdb.firebaseio.com/"
#define FIREBASE_AUTH "AlzaSyDFhkiw1Cx4b1eF8E8pWeL985bqi1jY2Lk"
#define SS_PIN 5
#define RST_PIN 4
int servoPin = 13; // Pino do microservo
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", -10800, 60000); // Ajuste o fuso
horário, -10800 para UTC-3
MFRC522 rfid(SS_PIN, RST_PIN);
LiquidCrystal_I2C lcd(0x27, 16, 2); // Endereço I2C e tamanho do display
(16x2)
Servo servo; // Instância do microservo
//const char ssid[] = "LUANA-2G";
//const char pass[] = "56090159";
// Configurações da rede Wi-Fi
const char ssid[] = "M10_4G_PRO";
const char pass[] = "sasasa12";
FirebaseData firebaseData;
FirebaseConfig firebaseConfig;
FirebaseAuth firebaseAuth;
```

```

bool signupOK = false;
void connectWiFi() {
  Serial.print("Conectando ao Wi-Fi");
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(1000);
  }
  Serial.println("\nConectado ao Wi-Fi!");
}
void setup() {
  Serial.begin(115200);
  // Inicializar o servo e definir o pino
  servo.attach(servoPin);
  servo.write(70); // Define o ângulo inicial
  connectWiFi();
  SPI.begin();
  rfid.PCD_Init();
  // Inicializa o cliente NTP
  timeClient.begin();
  // Configuração do Firebase
  firebaseConfig.api_key = FIREBASE_AUTH;
  firebaseConfig.database_url = FIREBASE_HOST;
  if (Firebase.signUp(&firebaseConfig, &firebaseAuth, "", "")) {
    Serial.println("Conexão com Firebase OK");
    signupOK = true;
  } else {
    Serial.println("Erro no cadastro:");
    Serial.printf("%s\n", firebaseConfig.signer.signupError.message.c_str());
  }
  Firebase.begin(&firebaseConfig, &firebaseAuth);
  Firebase.reconnectWiFi(true);
  // Configura e inicializa o display LCD

```

```

lcd.begin();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("Sistema Iniciado");
delay(2000);
lcd.clear();
}
void sendUIDToAddUser(String uid) {
  if (Firebase.setString(firebaseData, "access_control/tempUID", uid)) {
    Serial.println("UID enviado para adicionar usuário: " + uid);
  } else {
    Serial.println("Falha ao enviar UID para adicionar usuário: " +
firebaseData.errorReason());
  }
}
void logAccess(String uid) {
  String userPath = "/access_control/" + uid;
  Firebase.getString(firebaseData, userPath + "/uid");
  String rfid = firebaseData.stringData();
  Firebase.getString(firebaseData, userPath + "/name");
  String name = firebaseData.stringData();
  Firebase.getString(firebaseData, userPath + "/gender");
  String gender = firebaseData.stringData();
  if (!timeClient.update()) {
    timeClient.forceUpdate();
  }
  String logPath = "/log/" + String(millis());
  String timeStamp = String(timeClient.getFormattedTime());
  Firebase.setString(firebaseData, logPath + "/nome", name);
  Firebase.setString(firebaseData, logPath + "/uid", rfid);
  Firebase.setString(firebaseData, logPath + "/genero", gender);
  Firebase.setString(firebaseData, logPath + "/timestamp", timeStamp);
  lcd.setCursor(0, 0);
}

```

```

lcd.print("Acesso");
lcd.setCursor(0, 1);
lcd.print("Permitido");
// Move o servo para a posição de acesso permitido
servo.write(180); // Gira o servo para 180 graus
delay(2000); // Aguarda 2 segundos para exibir a mensagem
lcd.clear();
// Retorna o servo à posição inicial
servo.write(70); // Posição inicial
lcd.setCursor(0, 0);
lcd.print("Registrado!");
delay(2000); // Aguarda 2 segundos para exibir o UID
lcd.clear();
Serial.println("Log registrado com sucesso!");
}
void loop() {
  lcd.setCursor(0, 0);
  lcd.print("Aproxime a tag ");
  if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) {
    return;
  }
  lcd.clear();
  String uid = "";
  for (byte i = 0; i < rfid.uid.size; i++) {
    uid += String(rfid.uid.uidByte[i], HEX);
  }
  String userPath = "/access_control/" + uid;
  if (Firebase.getString(firebaseData, userPath + "/uid")) {
    logAccess(uid); // Registrar acesso se o UID estiver cadastrado
  } else {
    sendUIDToAddUser(uid); // Enviar UID temporário
    lcd.setCursor(0, 0);
    lcd.print("Acesso");
  }
}

```

```
lcd.setCursor(0, 1);  
lcd.print("Negado");  
delay(2000); // Aguarda 2 segundos para exhibir o UID  
lcd.clear();  
}  
}
```