



UNIVERSIDADE FEDERAL DE MATO GROSSO  
CAMPUS UNIVERSITÁRIO DE VÁRZEA GRANDE  
FACULDADE DE ENGENHARIA  
ENGENHARIA DE COMPUTAÇÃO



# Controle de acesso remoto usando Digiprox SA 202

**Nícolas Gabriel Meneses De Jesus**

Orientador: Prof. Dr. Fabrício Carvalho

**Trabalho de Conclusão de Curso** apresentado ao Curso de Engenharia de Computação da FAENG/CUVG/UFMT (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Computação.

Cuiabá, MT, 12 de Setembro de 2025

Catalogação da publicação na fonte. UFMT / Biblioteca Central  
Obter a Ficha pelo site <https://academico-siga.ufmt.br/ufmt.sfc/Home/Ficha>,  
gerar o arquivo PDF (para manter a qualidade) e inserir nesta posição.

*Aos meus amigos do serviço e a  
minha namorada por me dar forças  
para realizar o trabalho.*

---

# Agradecimentos

---

Ao meu orientador, Prof. Dr. Fabrício Carvalho, por fornecer o tema deste trabalho, pela aquisição da controladora SA-202 que tornou possível a realização dos experimentos, e pela orientação dedicada ao longo de todo o projeto.

Aos meus amigos de trabalho, pela valiosa ideia da conexão da antena em paralelo, que se mostrou fundamental para o sucesso da solução desenvolvida.

À minha namorada, pelo apoio incondicional e pela motivação constante durante toda esta jornada acadêmica.

Aos meus pais, por todo o apoio financeiro e emocional que me permitiu concluir a faculdade e chegar até este momento.

À UFMT, pela oportunidade de cursar Engenharia de Computação e pela infraestrutura disponibilizada.

---

# Resumo

---

Este trabalho apresenta o desenvolvimento de um sistema de controle de acesso com RFID integrado ao Firebase, propondo uma solução inovadora para modernização de controladoras legadas sem comprometer sua funcionalidade original. O projeto surgiu da necessidade de adicionar capacidades de Internet das Coisas (IoT) à controladora DigiProx SA-202, que apesar de funcional, carecia de recursos modernos como conectividade, armazenamento em nuvem e monitoramento remoto. A metodologia adotada iniciou com a tentativa de integração direta com o microcontrolador STC8C2K64S4-36I-LQFP32 da controladora existente, mas devido a barreiras técnicas significativas, como documentação em idioma chinês e necessidade de equipamentos especializados não disponíveis no Brasil, foi desenvolvida uma solução alternativa baseada na interceptação paralela dos sinais RFID. O sistema implementado utiliza um módulo RDM6300 para leitura das tags de 125 kHz, um Arduino Uno para processamento inicial dos dados e um ESP8266 para prover conectividade Wi-Fi e comunicação com o Firebase Realtime Database.

**Palavras-chave:** RFID, Internet das Coisas, Firebase, Controle de Acesso, Sistemas Embarcados, ESP8266, Arduino.

---

# Abstract

---

This work presents the development of an RFID access control system integrated with Firebase, proposing an innovative solution for modernizing legacy controllers without compromising their original functionality. The project arose from the need to add Internet of Things (IoT) capabilities to the DigiProx SA-202 controller, which despite being functional, lacked modern features such as connectivity, cloud storage, and remote monitoring. The adopted methodology began with an attempt at direct integration with the STC8C2K64S4-36I-LQFP32 microcontroller of the existing controller, but due to significant technical barriers, such as documentation in Chinese and the need for specialized equipment not available in Brazil, an alternative solution based on parallel interception of RFID signals was developed. The implemented system uses an RDM6300 module for reading 125 kHz tags, an Arduino Uno for initial data processing, and an ESP8266 to provide Wi-Fi connectivity and communication with Firebase Realtime Database.

**Keywords:** RFID, Internet of Things, Firebase, Access Control, Embedded Systems, ESP8266, Arduino.

---

# Sumário

---

<b>Sumário</b>	i
<b>Lista de Figuras</b>	ii
<b>Lista de Tabelas</b>	iii
<b>1 Introdução</b>	1
1.1 Motivação . . . . .	2
1.2 Objetivos e Contribuições . . . . .	2
1.3 Metodologia . . . . .	3
1.4 Estrutura do Trabalho . . . . .	4
1.5 Considerações Sobre o Desenvolvimento . . . . .	5
<b>Lista de Símbolos e Abreviaturas</b>	1
<b>2 Teoria</b>	7
2.1 Sistema de controle de acesso . . . . .	7
2.2 Tecnologia RFID . . . . .	8
2.2.1 Classificação por Frequência . . . . .	8
2.2.2 Tipos de Etiquetas RFID . . . . .	9
2.2.3 Componentes de um Sistema RFID . . . . .	10
2.2.4 Vantagens do RFID . . . . .	11
2.2.5 Aplicações Principais . . . . .	12
2.3 Leitor 125 kHz RDM630/RDM6300 e protocolo de quadros . . . . .	12

2.3.1	Características Técnicas . . . . .	13
2.3.2	Pinagem e Conexões . . . . .	14
2.3.3	Protocolo de Quadros . . . . .	14
2.3.4	Cálculo do Checksum . . . . .	15
2.3.5	Exemplo Prático de Quadro . . . . .	15
2.3.6	Considerações de Implementação . . . . .	16
2.4	Controlador DigiProx SA202 . . . . .	16
2.4.1	Especificações Técnicas . . . . .	17
2.4.2	Funcionalidades . . . . .	18
2.5	Comunicação Serial e Internet das Coisas (IoT) . . . . .	18
2.5.1	Universal Asynchronous Receiver-Transmitter (UART) . . . . .	18
2.5.2	Internet das Coisas (IoT) . . . . .	18
2.6	Ponte Microcontrolada e Integração com ESP8266 . . . . .	19
2.6.1	Divisão de Responsabilidades . . . . .	19
2.6.2	Benefícios da Arquitetura . . . . .	19
2.7	Plataforma Wi-Fi para IoT: ESP8266 . . . . .	20
2.7.1	Especificações Técnicas . . . . .	20
2.7.2	Implementação de Servidor HTTP Local . . . . .	20
2.7.3	Limitações e Considerações . . . . .	21
2.8	Serviços Web, REST e JSON . . . . .	21
2.8.1	Princípios REST e Métodos HTTP . . . . .	22
2.8.2	Vantagens para Dispositivos Embarcados . . . . .	22
2.9	Firebase Realtime Database via REST . . . . .	23
2.9.1	Estrutura da API REST . . . . .	23
2.9.2	Autenticação e Segurança . . . . .	23
2.10	HTTPS/TLS em Embarcados . . . . .	24
2.10.1	Implementação BearSSL . . . . .	24

2.10.2 Boas Práticas de Segurança . . . . .	24
2.11 Sincronização Temporal (NTP) . . . . .	24
2.11.1 Estratégias de Sincronização . . . . .	25
2.11.2 Modelagem de Dados . . . . .	25
2.11.3 Práticas de Confiabilidade . . . . .	25
2.12 Síntese e Implicações . . . . .	26
2.12.1 Componentes Integrados . . . . .	26
2.12.2 Reprodutibilidade e Padrões . . . . .	27
<b>3 Trabalhos relacionados</b>	<b>28</b>
3.1 Estudos Anteriores em Controle de Acesso RFID . . . . .	28
3.1.1 Limitações das Abordagens Existentes . . . . .	29
3.2 Soluções Comerciais Disponíveis . . . . .	29
3.2.1 Análise de Custo-Benefício . . . . .	29
3.3 Projetos de Código Aberto Relacionados . . . . .	30
3.3.1 Lacunas Identificadas . . . . .	30
3.4 Relação com o Problema Proposto . . . . .	31
3.5 Síntese e Posicionamento do Trabalho . . . . .	31
<b>4 Problema</b>	<b>33</b>
4.1 Objetivo do Trabalho . . . . .	33
4.2 Justificativa . . . . .	34
<b>5 Implementação</b>	<b>35</b>
5.1 Análise da Controladora Existente . . . . .	35
5.1.1 Descobrindo as Especificações do Microcontrolador . . . . .	35
5.2 Desafios Técnicos Encontrados . . . . .	38
5.2.1 Barreiras para Reprogramação . . . . .	38
5.2.2 Riscos de Modificar o Sistema Original . . . . .	38

5.3	Solução Proposta: Interceptação do Sinal RFID . . . . .	39
5.3.1	Arquitetura da Solução Desenvolvida . . . . .	39
5.3.2	Escolha do Módulo RDM6300 . . . . .	40
5.4	Implementação do Sistema de Leitura RFID . . . . .	41
5.4.1	Entendendo o Protocolo de Comunicação . . . . .	41
5.4.2	Desenvolvendo o Algoritmo de Leitura . . . . .	42
5.4.3	Implementando a Validação do Checksum . . . . .	44
5.4.4	Montando as Conexões do Hardware . . . . .	44
5.5	Integração com ESP8266 e Firebase . . . . .	46
5.5.1	Estabelecendo a Comunicação Arduino-ESP8266 . . . . .	46
5.5.2	Implementando a Comunicação com o Firebase . . . . .	47
5.5.3	Configurando o ESP8266 para Operação . . . . .	47
5.5.4	Organizando os Dados no Firebase . . . . .	48
5.5.5	Algoritmo de Transmissão para Firebase . . . . .	49
5.5.6	Tratamento de Dados Hexadecimais . . . . .	50
5.6	Interface Web de Monitoramento . . . . .	51
5.6.1	Criando os Endpoints HTTP . . . . .	51
5.6.2	Recursos de Monitoramento . . . . .	52
5.7	Integração Paralela com Controladora Original . . . . .	52
5.7.1	Fazendo a Conexão Paralela . . . . .	52
5.7.2	Garantindo a Sincronização dos Dados . . . . .	52
5.8	Considerações Finais . . . . .	53
5.8.1	Principais Desafios Superados . . . . .	53
5.8.2	Limitações do Sistema . . . . .	54
5.8.3	Sugestões para Trabalhos Futuros . . . . .	54
<b>6</b>	<b>Experimentos e Resultados</b>	<b>56</b>

6.1	Ambiente de Testes em Bancada . . . . .	56
6.2	Metodologia de Testes em Casa . . . . .	57
6.3	Experimento 1: Validação da Leitura RFID . . . . .	57
6.4	Experimento 2: Teste de Integração com Firebase . . . . .	59
6.5	Experimento 3: Operação Paralela com Controladora Original . . . . .	59
6.6	Experimento 4: Teste de Resistência . . . . .	60
6.7	Experimento 5: Testando a Interface Web . . . . .	60
6.8	Resultados Consolidados . . . . .	61
6.9	Resultados da Implementação . . . . .	61
6.9.1	Funcionalidades Alcançadas . . . . .	61
6.9.2	Vantagens da Solução Implementada . . . . .	62
6.10	Discussão dos Resultados . . . . .	62
<b>7</b>	<b>Conclusão</b>	<b>64</b>
7.1	Contribuições do Trabalho . . . . .	64
7.2	Limitações Identificadas . . . . .	65
7.3	Trabalhos Futuros . . . . .	66
7.4	Considerações Finais . . . . .	66
<b>Referências bibliográficas</b>		<b>68</b>
<b>Apêndice A Informações Complementares</b>		<b>71</b>
A.1	Declaração do Orientador Para a Biblioteca . . . . .	71

---

# **Lista de Figuras**

---

2.1	Pinagem do módulo RDM6300 . . . . .	14
5.1	Microcontrolador STC8C2K64S4-36I-LQFP32 utilizado na controladora SA- 202 . . . . .	36
5.2	Diagrama de pinagem do microcontrolador STC8C2K64S4-36I-LQFP32 . .	37
5.3	Arquitetura do sistema de interceptação RFID proposto . . . . .	40
5.4	Módulo leitor RFID RDM6300 . . . . .	41
5.5	Esquema de conexões entre RDM6300 e Arduino Uno . . . . .	45
5.6	Esquema de ligação com conversor de nível lógico . . . . .	46
5.7	Esquema de conexão paralela das antenas RFID . . . . .	53
6.1	Cartões RFID utilizados nos testes com códigos hexadecimais impressos .	58

---

# **Lista de Tabelas**

---

6.1 Resumo dos resultados experimentais . . . . .	61
---	----

---

# Capítulo 1

---

## Introdução

---

O controle de acesso físico é uma necessidade fundamental em diversos ambientes, desde residências e empresas até instituições de ensino e órgãos governamentais. Com o avanço da tecnologia, os sistemas tradicionais de controle de acesso baseados em chaves têm sido gradualmente substituídos por soluções eletrônicas mais sofisticadas, que oferecem maior segurança, praticidade e capacidade de auditoria. Entre essas tecnologias, destaca-se a identificação por radiofrequência (RFID), que permite o reconhecimento automático e sem contato de tags ou cartões, tornando o processo de autenticação mais ágil e eficiente.

Apesar dos benefícios oferecidos pelos sistemas eletrônicos de controle de acesso, muitas instalações ainda utilizam controladoras legadas que, embora funcionais, carecem de recursos modernos como conectividade à internet, armazenamento em nuvem e monitoramento remoto. A substituição completa desses sistemas frequentemente representa um investimento significativo, não apenas em hardware, mas também em instalação e treinamento, o que pode ser proibitivo para muitas organizações.

Neste contexto, surge a necessidade de soluções que permitam modernizar sistemas existentes sem comprometer sua funcionalidade original. Este trabalho apresenta o desenvolvimento de um sistema que adiciona capacidades de Internet das Coisas (IoT) a uma controladora de acesso RFID tradicional, especificamente o modelo DigiProx SA-202, permitindo o registro e monitoramento remoto de acessos através da plataforma

Firebase.

## 1.1 Motivação

A motivação principal deste projeto surgiu da observação de uma limitação prática em um ambiente real de uso. A controladora DigiProx SA-202, instalada em uma porta de acesso, funcionava adequadamente para seu propósito básico de controlar a fechadura eletrônica mediante a apresentação de tags RFID autorizadas. No entanto, a ausência de conectividade limitava severamente as possibilidades de gestão e análise dos dados de acesso.

A controladora original não mantém registros das leituras<sup>1</sup>, deixando sem a possibilidade de exportação ou análise de histórico. Essa limitação impossibilita a identificação de padrões de uso, a geração de relatórios de acesso e o monitoramento em tempo real, recursos cada vez mais necessários em ambientes que demandam maior controle de segurança.

Além disso, a crescente adoção de tecnologias IoT em diversos setores demonstra o valor da conectividade e do processamento de dados em nuvem. A capacidade de acessar informações remotamente, receber notificações em tempo real e integrar diferentes sistemas tornou-se não apenas desejável, mas muitas vezes essencial para a gestão eficiente de recursos e segurança.

## 1.2 Objetivos e Contribuições

Este trabalho teve como objetivo principal desenvolver uma solução de baixo custo para adicionar conectividade IoT a sistemas de controle de acesso RFID existentes, sem interferir em seu funcionamento original. Para alcançar esse objetivo, foi necessário su-

---

<sup>1</sup>A DigiProx SA-202 armazena apenas as configurações de usuários autorizados, mas não registra histórico de acessos.

perar diversos desafios técnicos, desde a análise reversa do hardware existente até a implementação de protocolos de comunicação seguros com serviços em nuvem.

As principais contribuições deste trabalho incluem:

- Desenvolvimento de uma arquitetura de interceptação paralela de sinais RFID que preserva a integridade do sistema original;
- Implementação de um protocolo de comunicação eficiente entre microcontroladores de diferentes gerações;
- Criação de uma solução de integração com Firebase que permite armazenamento e análise de dados em tempo real;
- Demonstração prática de que é possível modernizar sistemas legados com investimento mínimo;
- Documentação detalhada do processo, servindo como guia para projetos similares.

### 1.3 Metodologia

A metodologia adotada neste projeto seguiu uma abordagem incremental, iniciando com a análise detalhada do sistema existente e evoluindo através de protótipos sucessivos até a solução final. O processo começou com a tentativa de integração direta com o microcontrolador da controladora SA-202, mas devido a barreiras técnicas significativas, como documentação em idioma chinês e necessidade de equipamentos especializados não disponíveis, foi necessário adotar uma estratégia alternativa.

A solução desenvolvida utilizou um módulo leitor RFID RDM6300 operando em paralelo com o sistema original, conectado a um Arduino Uno para processamento inicial dos dados. Posteriormente, um módulo ESP8266 foi integrado para prover conectividade Wi-Fi e comunicação com o Firebase. Essa arquitetura modular permitiu o desenvolvimento e teste independente de cada componente, facilitando a identificação e correção de problemas.

Todos os experimentos foram conduzidos em ambiente controlado, com testes extensivos para validar a confiabilidade e o desempenho do sistema. As métricas de avaliação incluíram taxa de leitura bem-sucedida, latência de transmissão, disponibilidade do sistema e capacidade de operação paralela sem interferências.

## 1.4 Estrutura do Trabalho

Este trabalho está organizado em sete capítulos, estruturados de forma a apresentar progressivamente o desenvolvimento do projeto desde sua fundamentação teórica até as conclusões finais.

O Capítulo 2 apresenta a fundamentação teórica necessária para compreensão do projeto, abordando os conceitos de sistemas de controle de acesso, tecnologia RFID, protocolo de comunicação do módulo RDM6300, microcontrolador ESP8266 e a plataforma Firebase. Essa base teórica é essencial para entender as decisões técnicas tomadas durante o desenvolvimento.

O Capítulo 3 discute trabalhos relacionados encontrados na literatura, comparando diferentes abordagens para modernização de sistemas legados e integração de dispositivos IoT. São analisadas soluções comerciais e acadêmicas, destacando suas vantagens e limitações.

O Capítulo 4 define formalmente o problema abordado, detalhando os requisitos funcionais e não-funcionais do sistema proposto. São apresentadas as limitações da controladora original e os objetivos específicos que a solução deve atender.

O Capítulo 5 descreve detalhadamente a implementação do sistema, desde a análise inicial da controladora SA-202 até a integração final com o Firebase. São apresentados os algoritmos desenvolvidos, os esquemas de conexão e os desafios técnicos superados durante o desenvolvimento.

O Capítulo 6 apresenta os experimentos realizados para validar a solução proposta,

incluindo testes de leitura RFID, integração com Firebase, operação paralela e testes de estresse. Os resultados são analisados quantitativamente e comparados com sistemas similares.

Finalmente, o Capítulo 7 apresenta as conclusões do trabalho, destacando as contribuições realizadas, as limitações identificadas e sugestões para trabalhos futuros. São discutidas as lições aprendidas e a aplicabilidade da solução em contextos mais amplos.

## 1.5 Considerações Sobre o Desenvolvimento

Durante o desenvolvimento deste projeto, foi fundamental manter o foco na praticidade e viabilidade da solução. A decisão de utilizar componentes de baixo custo e amplamente disponíveis no mercado nacional foi estratégica para garantir a replicabilidade do projeto. O Arduino Uno<sup>2</sup> e o ESP8266<sup>3</sup>, por exemplo, são plataformas consolidadas com vasta documentação e suporte da comunidade, o que facilita tanto o desenvolvimento quanto a manutenção futura.

A escolha do Firebase<sup>4</sup> como plataforma de armazenamento em nuvem também foi cuidadosamente considerada. Além de oferecer um plano gratuito adequado para projetos de pequena escala, o Firebase proporciona APIs bem documentadas, segurança robusta e ferramentas de análise integradas. Essas características tornam a plataforma ideal para aplicações IoT que requerem processamento e armazenamento de dados em tempo real.

É importante ressaltar que este trabalho não pretende substituir sistemas comerciais de alta complexidade, mas sim demonstrar que é possível adicionar funcionalidades modernas a sistemas existentes com investimento mínimo. Essa abordagem é particularmente relevante em um contexto onde muitas organizações possuem infraestrutura legada funcional, mas carecem de recursos para modernização completa.

---

<sup>2</sup>Microcontrolador baseado no ATmega328P, amplamente utilizado em projetos de prototipagem.

<sup>3</sup>Módulo Wi-Fi com microcontrolador integrado, desenvolvido pela Espressif Systems.

<sup>4</sup>Plataforma de desenvolvimento de aplicações móveis e web da Google, que oferece banco de dados em tempo real e hospedagem.

O desenvolvimento deste projeto também revelou a importância da documentação técnica e do compartilhamento de conhecimento. Muitos dos desafios enfrentados poderiam ter sido evitados com melhor documentação dos componentes utilizados, especialmente considerando as barreiras linguísticas encontradas. Por isso, este trabalho busca não apenas apresentar uma solução técnica, mas também servir como referência documentada para futuros projetos similares.

---

# **Capítulo 2**

---

## **Teoria**

---

### **2.1 Sistema de controle de acesso**

Os sistemas de controle de acesso são fundamentais para a segurança física de qualquer ambiente, seja residencial, comercial ou industrial. Esses sistemas têm como objetivo principal identificar credenciais e gerenciar permissões em pontos físicos específicos como portas, portões e catracas, além de registrar todos os eventos para posterior auditoria. No contexto de segurança física, os meios de identificação evoluíram significativamente ao longo dos anos, sendo que atualmente os mais comuns incluem senhas numéricas, cartões de proximidade baseados em RFID e sistemas biométricos como leitores de digital ou reconhecimento facial.

A arquitetura básica de um sistema de controle de acesso é composta por vários elementos que trabalham em conjunto. Os pontos de leitura são responsáveis por capturar as credenciais dos usuários, enquanto a controladora processa essas informações e toma a decisão de liberar ou negar o acesso. O atuador, geralmente uma fechadura eletrônica ou eletromagnética, executa fisicamente a ação determinada pela controladora. Além desses componentes essenciais, sistemas mais sofisticados incluem recursos de auditoria que registram cada tentativa de acesso, criando um histórico detalhado para análise posterior.

Em ambientes organizacionais, a gestão eficiente desses sistemas de acesso traz benefícios que vão além da segurança. A rastreabilidade completa dos movimentos permite

identificar padrões de uso, otimizar fluxos de pessoas e até mesmo auxiliar em investigações quando necessário. A análise dos dados coletados pode revelar vulnerabilidades no sistema de segurança e ajudar na tomada de decisões estratégicas sobre a proteção dos locais.

## 2.2 Tecnologia RFID

A tecnologia de identificação por radiofrequência (RFID)<sup>1</sup> é uma família de tecnologias de identificação automática que utiliza ondas de rádio para comunicação, identificação e rastreamento de objetos sem fio. Essa tecnologia representa uma ferramenta valiosa para diversas aplicações, desde gerenciamento da cadeia de suprimentos até controle de acesso.

O RFID funciona através da colocação de uma etiqueta RFID ou transponder sobre objetos, que contém um microchip e uma antena responsáveis por transmitir um identificador exclusivo para um dispositivo leitor quando solicitado pelo sinal de rádio do leitor. Esta tecnologia permite identificação e rastreamento sem contato físico e sem necessidade de linha de visão direta, oferecendo vantagens significativas sobre sistemas tradicionais como códigos de barras (Dennison s.d.).

### 2.2.1 Classificação por Frequência

As etiquetas RFID podem ser classificadas em três categorias principais, cada uma operando em faixas de frequência específicas e com características distintas que as tornam mais adequadas para determinadas aplicações.

Na faixa de baixa frequência, conhecida como LF<sup>2</sup>, que opera entre 125 e 134 kHz,

---

<sup>1</sup>Radio Frequency Identification - tecnologia que utiliza campos eletromagnéticos para identificar e rastrear tags anexadas a objetos.

<sup>2</sup>Low Frequency - faixa de frequência entre 30 kHz e 300 kHz, sendo 125-134 kHz a mais comum para RFID.

encontramos predominantemente etiquetas passivas que não possuem bateria própria. Essas etiquetas funcionam através de acoplamento indutivo, utilizando modulação ASK e codificação Manchester para transmissão de dados. O alcance típico dessas tags é relativamente curto, aproximadamente 10 centímetros, mas elas apresentam uma vantagem significativa em termos de imunidade a interferências causadas por água e metais, sendo superiores nesse aspecto às etiquetas de frequências mais altas. Por essas características, são amplamente utilizadas em sistemas de identificação de animais e, especialmente importante para este trabalho, em controle de acesso (Dennison s.d.).

A faixa de alta frequência, ou HF<sup>3</sup>, opera em 13,56 MHz e inclui a popular tecnologia NFC<sup>4</sup> (Near Field Communication). Essas etiquetas conseguem alcançar distâncias de leitura de alguns metros, o que as torna ideais para aplicações que requerem maior flexibilidade na aproximação do leitor. O gerenciamento de estoque no varejo e o rastreamento de ativos são exemplos típicos de uso dessa tecnologia, aproveitando seu equilíbrio entre alcance e confiabilidade.

Já as etiquetas de ultra alta frequência, conhecidas como UHF<sup>5</sup>, operam na faixa entre 868 e 915 MHz e oferecem o maior alcance de leitura, podendo chegar a 20 metros em condições ideais. Essa característica as torna perfeitas para aplicações em gerenciamento de cadeia de suprimentos e rastreamento de ativos em grande escala. No entanto, é importante destacar que essas etiquetas são mais suscetíveis a interferências causadas por líquidos e metais, o que pode limitar sua aplicação em certos ambientes.

## 2.2.2 Tipos de Etiquetas RFID

Quando falamos sobre tipos de etiquetas RFID, a principal distinção está na forma como elas obtêm energia para funcionar. Basicamente, existem dois tipos principais: as etiquetas passivas e as ativas, cada uma com suas próprias características e aplicações

<sup>3</sup>High Frequency - faixa ISM (Industrial, Scientific and Medical) de 13,56 MHz.

<sup>4</sup>Near Field Communication - tecnologia de comunicação de curto alcance baseada em RFID HF.

<sup>5</sup>Ultra High Frequency - faixa que varia por região: 860-960 MHz na Europa e 902-928 MHz nos EUA.

específicas.

As etiquetas passivas são as mais comuns e econômicas. Elas não possuem nenhuma fonte de energia interna, como uma bateria, e dependem completamente da energia transmitida pelo leitor RFID para funcionar. Quando o leitor emite ondas de rádio, a antena da etiqueta passiva captura essa energia e a utiliza tanto para alimentar o microchip quanto para transmitir os dados de volta ao leitor. Essa dependência energética limita seu alcance de operação, mas também as torna extremamente duráveis e de baixo custo, ideais para aplicações em massa.

Por outro lado, as etiquetas ativas possuem sua própria fonte de energia interna, geralmente uma bateria. Essa característica permite que elas transmitam dados por distâncias muito maiores e de forma contínua ou periódica, sem depender da proximidade de um leitor. As tags ativas podem incluir sensores adicionais e armazenar mais informações, mas são significativamente mais caras e têm vida útil limitada pela duração da bateria.

### 2.2.3 Componentes de um Sistema RFID

Para entender como funciona um sistema RFID completo, é importante conhecer seus componentes principais e como eles interagem entre si. O sistema não se resume apenas às tags e leitores, mas envolve uma arquitetura mais complexa que garante seu funcionamento eficiente.

O leitor RFID é o coração do sistema, responsável por transmitir a energia de radiofrequência necessária para ativar as etiquetas passivas e estabelecer comunicação com elas. Além de energizar as tags, o leitor também recebe e decodifica os dados transmitidos de volta, convertendo os sinais de rádio em informações digitáveis que podem ser processadas pelo sistema.

A antena trabalha em conjunto com o leitor, sendo responsável por irradiar as ondas de radiofrequência e captar os sinais de resposta das etiquetas. O design e posicionamento da antena são críticos para determinar a área de cobertura e a eficiência da leitura. Em

muitos casos, um único leitor pode estar conectado a múltiplas antenas para cobrir uma área maior.

As etiquetas RFID, também conhecidas como tags, contêm um microchip que armazena as informações e uma antena própria para comunicação. Cada tag possui um identificador único e pode armazenar dados adicionais sobre o item ao qual está anexada.

Por fim, o sistema de host representa toda a infraestrutura de software e hardware que controla e gerencia o sistema RFID. Isso inclui servidores, bancos de dados, software de gerenciamento e interfaces de usuário que permitem configurar o sistema, monitorar seu funcionamento e analisar os dados coletados.

#### 2.2.4 Vantagens do RFID

A tecnologia RFID revolucionou a forma como identificamos e rastreamos objetos, oferecendo vantagens significativas sobre métodos tradicionais. Uma das principais vantagens é a velocidade de troca de dados, que permite processar centenas de leituras por segundo, resultando em maior eficiência operacional e precisão nas informações coletadas. Além disso, o sistema fornece dados em tempo real sobre a movimentação e localização de qualquer item que possua uma etiqueta RFID, permitindo um controle muito mais refinado dos processos (Dennison s.d.).

Outra característica fundamental da tecnologia RFID é sua capacidade de automatizar completamente diversos processos. No contexto de controle de acesso, por exemplo, o sistema pode identificar automaticamente pessoas autorizadas e liberar ou negar acesso a áreas restritas sem intervenção humana, aumentando tanto a segurança quanto a conveniência. A tecnologia também contribui significativamente para a prevenção de roubos e perdas, já que permite rastrear itens em tempo real (Vieira et al. 2007).

Um dos aspectos mais práticos do RFID é que não requer contato direto ou linha de visão entre o leitor e a etiqueta para realizar a leitura. Isso significa que vários itens podem ser lidos simultaneamente, mesmo quando estão dentro de caixas ou contêineres, o que

seria impossível com tecnologias como código de barras (Dennison s.d.).

### **2.2.5 Aplicações Principais**

A versatilidade da tecnologia RFID permitiu sua adoção em praticamente todos os setores da economia. No controle de acesso, foco principal deste trabalho, o RFID tornou-se a tecnologia dominante, permitindo a identificação rápida e segura de pessoas através de cartões, pulseiras ou chaveiros com tags embarcadas. Empresas, universidades, hospitais e condomínios residenciais adotaram amplamente essa tecnologia pela sua praticidade e confiabilidade (Vieira et al. 2007).

Na indústria e logística, o RFID revolucionou o rastreamento de produtos ao longo de toda a cadeia de suprimentos. Desde a linha de produção até a entrega ao consumidor final, cada item pode ser monitorado individualmente, fornecendo visibilidade completa do processo. Isso permite não apenas melhor controle de qualidade, mas também identificação rápida de gargalos e otimização de rotas de distribuição (Dennison s.d.).

O gerenciamento de ativos empresariais também se beneficiou enormemente da tecnologia. Máquinas, ferramentas, equipamentos médicos e até mesmo documentos importantes podem ser etiquetados e rastreados, reduzindo perdas e melhorando a utilização dos recursos. No varejo, o RFID permitiu a implementação de inventários em tempo real, checkouts automáticos e prevenção mais eficaz de furtos, transformando a experiência tanto para lojistas quanto para consumidores (Vieira et al. 2007).

## **2.3 Leitor 125 kHz RDM630/RDM6300 e protocolo de quadros**

O RDM6300 é um módulo leitor RFID que opera na frequência de 125 kHz, oferecendo uma solução prática e econômica para sistemas de identificação por radiofrequênc-

cia. Este módulo tornou-se extremamente popular em projetos de controle de acesso devido à sua simplicidade de integração e protocolo de comunicação bem documentado. A interface UART que ele oferece permite conexão direta com praticamente qualquer microcontrolador, tornando-o ideal para projetos como o desenvolvido neste trabalho (Seeed Studio 2016).

### 2.3.1 Características Técnicas

O módulo RDM630/RDM6300 foi projetado com características que o tornam particularmente adequado para aplicações de controle de acesso. Operando na frequência de 125 kHz, característica da faixa de baixa frequência RFID, o módulo oferece boa imunidade a interferências e funcionamento confiável em diversos ambientes.

A comunicação com o módulo é realizada através de uma interface UART TTL padrão, configurada para operar a 9600 bits por segundo, com 8 bits de dados, sem bit de paridade e 1 bit de parada. Essa padronização facilita enormemente a integração com microcontroladores como Arduino, ESP8266 e outros.

Em termos de desempenho, o módulo consegue realizar leituras a uma distância máxima efetiva de aproximadamente 50 milímetros, embora na prática essa distância possa variar dependendo do tamanho e tipo da tag utilizada. O tempo de decodificação é impressionantemente rápido, inferior a 100 milissegundos, o que permite respostas praticamente instantâneas em aplicações de controle de acesso.

O RDM6300 é compatível com tags do padrão EM4100 e similares, que são amplamente disponíveis no mercado em diversos formatos como cartões, chaveiros e etiquetas adesivas. O módulo também inclui recursos adicionais como um LED bicolor integrado que pode indicar o status de operação e um driver para buzzer que permite feedback sonoro durante as leituras. A possibilidade de conectar uma antena externa amplia ainda mais as opções de instalação e pode melhorar o desempenho em aplicações específicas.

### 2.3.2 Pinagem e Conexões

O módulo possui três conectores principais:

#### Conektor P1 (Interface Serial):

- PIN1: TX (transmissão de dados)
- PIN2: RX (recepção de dados)
- PIN3: Não conectado
- PIN4: GND (terra)
- PIN5: +5V DC (alimentação)

#### Conektor P2 (Antena):

- PIN1: ANT1 (conexão da antena)
- PIN2: ANT2 (conexão da antena)

#### Conektor P3 (LED):

- PIN1: LED (controle do LED)
- PIN2: +5V DC
- PIN3: GND

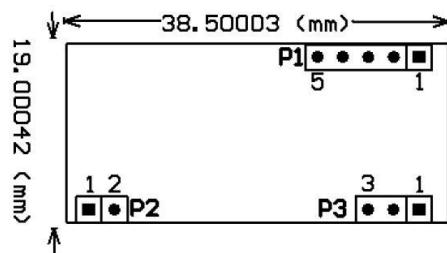


Figura 2.1: Pinagem do módulo RDM6300

### 2.3.3 Protocolo de Quadros

O RDM6300 transmite dados através da interface UART utilizando quadros ASCII com estrutura bem definida. A estrutura padrão do quadro é:

**Formato do Quadro:**

[HEAD] [DADOS] [CHECKSUM] [TAIL]

Onde:

- **HEAD:** 0x02 (início do quadro)
- **DADOS:** 10 caracteres hexadecimais ASCII (2 de versão + 8 do ID da tag)
- **CHECKSUM:** 2 caracteres hexadecimais ASCII
- **TAIL:** 0x03 (fim do quadro)

### 2.3.4 Cálculo do Checksum

O checksum é calculado realizando a operação XOR dos valores numéricos de cada par hexadecimal nos 10 caracteres de dados. Por exemplo, para o número de cartão 62E3086CED:

- Dados de saída: 36H, 32H, 45H, 33H, 30H, 38H, 36H, 43H, 45H, 44H
- Cálculo: (62H) XOR (E3H) XOR (08H) XOR (6CH) XOR (EDH) = 08H

### 2.3.5 Exemplo Prático de Quadro

Um exemplo real de saída do módulo:

**Dados em hexadecimal:**

02 30 31 30 30 30 37 33 34 45 30 44 32 03

**Interpretação:**

- HEAD: 02
- Dados: 30 31 30 30 30 37 33 34 45 30 (equivale a "0100073E0"em ASCII)
- Checksum: 44 32 (equivale a "D2"em ASCII)
- TAIL: 03

**Verificação do checksum:** (01H) XOR (00H) XOR (07H) XOR (34H) XOR (E0H) = D2H[ç]

### 2.3.6 Considerações de Implementação

O protocolo facilita a validação local e o descarte de quadros corrompidos através do checksum. Dependendo do cartão ou etiqueta utilizada, o UID efetivo pode ser representado com 8 ou 10 caracteres hexadecimais, sendo necessário que o software acomode ambas as possibilidades (Seeed Studio 2016).

Para integração com microcontroladores, um código básico em Arduino seria:

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    if(Serial.available()) {  
        while(Serial.available()) {  
            Serial.write(Serial.read());  
        }  
    }  
}
```

## 2.4 Controlador DigiProx SA202

O DigiProx SA202 é um controlador de acesso dedicado que opera na frequência de 125 kHz, projetado especificamente para controle de entrada e saída de pessoas em pequenos ambientes. Este dispositivo representa uma solução integrada que combina leitor

RFID, processamento de autenticação e controle de atuadores em um único equipamento (Intelbras 2024).

### 2.4.1 Especificações Técnicas

O controlador apresenta as seguintes características técnicas (Intelbras 2024):

#### Especificações Elétricas:

- **Tensão de alimentação:** 12 Vdc
- **Potência de operação:** 0,5 W
- **Corrente de chaveamento:** 200 mA

#### Condições Ambientais:

- **Temperatura de operação:** -10 °C a 70 °C
- **Umidade de operação:** 20% a 80%

#### Especificações RFID:

- **Frequência de operação:** 125 kHz
- **Modulação:** ASK (Amplitude Shift Keying)
- **Taxa de transmissão:** 3,906 kbps
- **Código de emissão:** 125KA2DCN
- **Tipo de antena:** Interna

#### Capacidades de Armazenamento:

- **Capacidade máxima de cartões:** 1.000 usuários
- **Capacidade máxima de senhas:** 1.000 usuários

#### Dimensões Físicas:

- **Dimensões (L × A × P):** 75 × 118 × 21 mm
- **Gabinete:** Plástico de alta resistência

### 2.4.2 Funcionalidades

O DigiProx SA202 oferece as seguintes funcionalidades (Intelbras 2024):

- **Sinalização sonora:** Feedback auditivo para operações
- **Compatibilidade ampla:** Funciona com fechaduras eletroímã, eletromecânicas, leitores e automatizadores de portão
- **Métodos de autenticação:** Cartão de proximidade, senha ou acesso combinado
- **Controle de usuários:** Gerenciamento de até 1.000 usuários simultâneos

## 2.5 Comunicação Serial e Internet das Coisas (IoT)

### 2.5.1 Universal Asynchronous Receiver-Transmitter (UART)

UART é um protocolo de comunicação serial assíncrono amplamente utilizado em sistemas embarcados para transmissão de dados entre dispositivos. Diferente de protocolos síncronos como SPI ou I2C, o UART não requer um sinal de clock compartilhado, tornando-o ideal para comunicações simples entre microcontroladores (MakerHero 2023).

Na comunicação UART, os dados são transmitidos bit a bit através de dois fios: TX (transmissão) e RX (recepção). Cada dispositivo possui seu próprio TX conectado ao RX do outro dispositivo, formando uma comunicação full-duplex. A taxa de transmissão (baud rate) deve ser configurada igualmente em ambos os dispositivos para garantir a correta interpretação dos dados.

### 2.5.2 Internet das Coisas (IoT)

A Internet das Coisas refere-se à interconexão de dispositivos físicos através da internet, permitindo coleta e troca de dados sem intervenção humana direta. No contexto de controle de acesso, IoT possibilita o monitoramento remoto, gestão centralizada e análise de dados em tempo real (Embarcados 2023).

## 2.6 Ponte Microcontrolada e Integração com ESP8266

A solução proposta adota uma estratégia de separação de responsabilidades entre dois microcontroladores, visando reduzir o acoplamento e facilitar a manutenção e evolução do sistema. Esta abordagem permite que cada componente se especialize em suas funções específicas.

### 2.6.1 Divisão de Responsabilidades

#### Microcontrolador A (Arduino):

- **Função principal:** Leitura e validação do quadro RFID
- **Processamento:** Decodificação do protocolo RDM6300, validação de checksum
- **Saída:** Transmissão da TAG em formato texto simples via UART
- **Vantagens:** Processamento dedicado, isolamento de falhas, facilidade de teste

#### Microcontrolador B (ESP8266):

- **Função principal:** Normalização de dados e comunicação com backend
- **Processamento:** Validações finais, formatação JSON, envio HTTPS
- **Conectividade:** Interface Wi-Fi, cliente HTTP/HTTPS
- **Vantagens:** Evolução independente, conectividade nativa, processamento de rede

### 2.6.2 Benefícios da Arquitetura

Esta separação oferece vantagens significativas:

- **Redução de acoplamento:** Cada módulo tem responsabilidades bem definidas
- **Facilidade de testes:** Testes em bancada podem ser realizados independentemente
- **Evolução independente:** O envio à nuvem pode evoluir sem afetar a leitura RFID

- **Isolamento de falhas:** Problemas de conectividade não afetam a leitura local
- **Manutenibilidade:** Código mais modular e fácil de manter

## 2.7 Plataforma Wi-Fi para IoT: ESP8266

O ESP8266 é um System-on-Chip (SoC) desenvolvido pela Espressif Systems que integra um processador de 32 bits com conectividade Wi-Fi 802.11 b/g/n. Lançado em 2014, rapidamente se tornou uma das plataformas mais populares para projetos IoT devido ao seu baixo custo e facilidade de programação (MakerHero 2023, Embarcados 2023).

### 2.7.1 Especificações Técnicas

O ESP8266 apresenta características que o tornam ideal para aplicações IoT (Espressif Systems 2020):

- **Processador:** Tensilica L106 de 32 bits operando a 80/160 MHz
- **Memória:** 64 KB de RAM de instruções e 96 KB de RAM de dados
- **Conectividade:** Wi-Fi 802.11 b/g/n com WPA/WPA2
- **Protocolos:** Pilha TCP/IP completa, suporte a IPv4
- **Programação:** Compatível com Arduino IDE, MicroPython e SDK nativo
- **Interfaces:** UART, SPI, I2C, PWM e GPIO

### 2.7.2 Implementação de Servidor HTTP Local

O ESP8266 pode hospedar um servidor HTTP local para testes e configuração:

```
// Exemplo de endpoint para testes
GET /setTag?code=1A2B3C4D
POST /api/access
{
```

```
"tag": "1A2B3C4D",  
"timestamp": 1712345678,  
"device": "ESP8266-ABCD"  
}
```

### 2.7.3 Limitações e Considerações

#### Restrições de Hardware:

- **RAM limitada:** Requer gerenciamento cuidadoso de memória
- **Flash limitado:** Código deve ser otimizado
- **Processamento:** Single-core, requer programação não-bloqueante

#### Confiabilidade de Rede:

- **Reconexão automática:** Rotinas para reconectar Wi-Fi
- **Timeouts:** Configuração adequada de timeouts HTTP
- **Fila offline:** Sistema store-and-forward para garantir entrega
- **Retry logic:** Tentativas com backoff exponencial

## 2.8 Serviços Web, REST e JSON

A arquitetura REST (Representational State Transfer) é um estilo arquitetural para sistemas distribuídos proposto por Roy Fielding em 2000. REST modela recursos como URLs e utiliza métodos HTTP padrão para manipulá-los, sendo amplamente adotada em sistemas IoT devido à sua simplicidade e interoperabilidade (Fredrich 2012). O formato JSON (JavaScript Object Notation) tornou-se o padrão de facto para troca de dados em APIs REST por sua leveza e facilidade de leitura (MDN Web Docs 2024c).

### 2.8.1 Princípios REST e Métodos HTTP

Os métodos HTTP definem as operações que podem ser realizadas sobre recursos (MDN Web Docs 2024b):

- **GET:** Recuperação de recursos sem efeitos colaterais (idempotente e seguro)
- **POST:** Criação de novos recursos ou envio de dados para processamento (não idempotente)
- **PUT:** Atualização completa de recursos existentes (idempotente)
- **DELETE:** Remoção de recursos do servidor (idempotente)

A idempotência é uma propriedade importante que garante que múltiplas chamadas idênticas produzem o mesmo resultado, fundamental para a confiabilidade em redes instáveis (Fredrich 2012).

#### Representação JSON:

```
{  
  "tag": "1A2B3C4D",  
  "timestamp": 1712345678,  
  "device": "ESP8266-ABCD",  
  "reader": "RDM6300",  
  "door": "entrance_01"
```

### 2.8.2 Vantagens para Dispositivos Embarcados

- **Simplicidade:** Protocolo bem definido e amplamente conhecido
- **Interoperabilidade:** Compatível com qualquer cliente HTTP
- **Debug facilitado:** Ferramentas padrão (curl, Postman, navegadores)
- **Escalabilidade:** Stateless por natureza

## 2.9 Firebase Realtime Database via REST

O Firebase Realtime Database (RTDB) oferece uma API REST completa que permite operações CRUD através de requisições HTTP padrão (Google Firebase 2024).

### 2.9.1 Estrutura da API REST

#### **Endpoint padrão:**

```
POST https://<project-id>-default-rtdb.firebaseio.com/<path>.json?auth=<TOKEN>
```

#### **Exemplo de requisição:**

```
POST https://meu-projeto.firebaseio.com/access_logs.json?auth=<TOKEN>
```

```
Content-Type: application/json
```

```
{
  "tag": "1A2B3C4D",
  "timestamp": 1712345678,
  "device": "ESP8266-ABCD"
}
```

### 2.9.2 Autenticação e Segurança

- **Token de autenticação:** Parâmetro auth na URL
- **Regras de segurança:** Restrição de leitura/escrita por dispositivo/usuário
- **HTTPS obrigatório:** Comunicação criptografada
- **Validação de certificado:** Verificação da cadeia de confiança

## 2.10 HTTPS/TLS em Embarcados

A comunicação segura é essencial para sistemas de controle de acesso, exigindo implementação adequada de TLS no ESP8266 (MDN Web Docs 2024a, ?).

### 2.10.1 Implementação BearSSL

O ESP8266 utiliza BearSSL como implementação padrão de TLS, oferecendo diferentes abordagens de validação:

#### Opções de Validação:

- **CA Root:** Validação através da cadeia de confiança completa
- **Certificate Pinning:** Fixação da impressão digital do certificado
- **Modo inseguro:** Apenas para desenvolvimento (setInsecure())

### 2.10.2 Boas Práticas de Segurança

- **Evitar modo inseguro:** Nunca usar setInsecure() em produção
- **Timeouts adequados:** Configurar timeouts para conexões TLS
- **Retry com backoff:** Tentativas exponenciais em caso de falha
- **Validação de certificado:** Sempre validar a cadeia de confiança

## 2.11 Sincronização Temporal (NTP)

Para registros auditáveis e análises temporais confiáveis, a sincronização temporal é fundamental em sistemas de controle de acesso (NIC.br 2024, Observatório Nacional 2024).

### 2.11.1 Estratégias de Sincronização

Para garantir a precisão temporal no sistema, implementei diversas estratégias de sincronização. A primeira delas é a sincronização obrigatória durante o boot do dispositivo, garantindo que o sistema sempre inicie com o horário correto. Além disso, configurei uma ressincronização periódica a cada 24 horas para manter a precisão ao longo do tempo. Para compensar possíveis desvios do relógio interno do ESP8266, implementei um mecanismo de correção de deriva que ajusta gradualmente pequenas diferenças detectadas. Como medida de contingência, caso ocorra alguma falha na sincronização NTP, o sistema utiliza timestamps relativos baseados no tempo de funcionamento do dispositivo, garantindo que os registros mantenham sua ordem cronológica mesmo sem acesso ao servidor de tempo.

### 2.11.2 Modelagem de Dados

#### Esquema mínimo recomendado:

```
{  
    "tag": "1A2B3C4D",  
    "timestamp": 1712345678,  
    "device": "ESP8266-ABCD",  
    "reader": "RDM6300",  
    "door": "entrance_01"  
}
```

### 2.11.3 Práticas de Confiabilidade

Para garantir a confiabilidade do sistema, implementei várias práticas essenciais. A normalização estrita dos dados através da validação do formato hexadecimal dos cartões

RFID garante que apenas leituras válidas sejam processadas. A implementação de idempotência previne o registro duplicado de eventos, mesmo em casos de reenvio de dados. Quando o dispositivo está offline, utilize uma estratégia de store-and-forward, mantendo os eventos em uma fila local até que a conexão seja restabelecida. Em casos de falha de comunicação, aplico um algoritmo de backoff exponencial para os retries, evitando sobrecarga do servidor. Cada dispositivo possui um identificador único (UID) que permite rastrear a origem de cada evento no sistema. Por fim, todos os logs são estruturados de forma padronizada, facilitando tanto o troubleshooting quanto a auditoria posterior dos eventos.

## 2.12 Síntese e Implicações

A fundamentação teórica apresentada estabelece os pilares técnicos necessários para implementação de um sistema completo de controle de acesso baseado em RFID com conectividade IoT. A literatura e documentação levantadas fundamentam: (a) a leitura confiável de RFID LF (125 kHz) e seus efeitos práticos (Vieira et al. 2007, Dennison s.d.); (b) o parse de quadros ASCII do RDM6300 com checksum XOR para integridade (Seeed Studio 2016); (c) a arquitetura de ponte UART com validação no Arduino e publicação REST no ESP8266 (Embarcados 2023, MakerHero 2023); (d) o envio HTTPS com práticas de segurança mínimas (MDN Web Docs 2024a, Google Firebase 2024); e (e) elementos de formatação acadêmica e reproduzibilidade exigidos em TCC (ABNT 2018, ABNT 2023, ABNT 2024).

### 2.12.1 Componentes Integrados

A solução desenvolvida neste trabalho integra múltiplos componentes tecnológicos de forma harmoniosa. A leitura RFID é realizada através do protocolo RDM6300 com validação de integridade dos dados, garantindo confiabilidade nas leituras. A arquitetura

distribuída separa as responsabilidades entre os microcontroladores Arduino e ESP8266, permitindo que cada um execute as tarefas para as quais é mais adequado. A conectividade com a nuvem é estabelecida de forma segura através de HTTPS/TLS com validação adequada dos certificados CA pré-configurados para validação de servidores populares como Google, Firebase e AWS estão disponíveis na biblioteca BearSSL do ESP8266, simplificando a implementação de conexões seguras. A sincronização temporal via NTP garante timestamps auditáveis para todos os eventos registrados. Além disso, implementei práticas robustas de confiabilidade com tratamento adequado de falhas e mecanismos de recuperação automática.

### 2.12.2 Reprodutibilidade e Padrões

Para garantir a reproduzibilidade do projeto, todos os componentes foram especificados seguindo protocolos padronizados como UART para comunicação serial, HTTP/HTTPS para comunicação web, JSON para estruturação de dados e NTP para sincronização temporal. A implementação baseia-se exclusivamente em documentação oficial dos fabricantes e organizações de padrões, seguindo as práticas estabelecidas pela indústria para desenvolvimento de soluções IoT. Todo o código desenvolvido está documentado e disponível como referência para implementações futuras.

Esses pilares tornam o trabalho auto-contido e reproduzível, pois todos os blocos (protocolo, UART, REST/HTTPS, estrutura do dado) estão especificados e suportados por referências, fornecendo base sólida para a implementação prática do sistema de controle de acesso.

---

# **Capítulo 3**

---

## **Trabalhos relacionados**

---

Este capítulo apresenta uma análise crítica de trabalhos acadêmicos e soluções comerciais existentes na área de controle de acesso com RFID e IoT. O objetivo é identificar as limitações das abordagens atuais e posicionar a contribuição deste trabalho no contexto do estado da arte.

### **3.1 Estudos Anteriores em Controle de Acesso RFID**

Diversos pesquisadores têm proposto soluções para modernização de sistemas de controle de acesso. Estudos recentes demonstram que sistemas completos de controle de acesso baseados em RFID podem ser implementados de forma eficaz, porém muitas soluções exigem a substituição completa dos equipamentos existentes, tornando-as inviáveis para instituições com orçamento limitado.

Um trabalho relevante foi desenvolvido por pesquisadores da UFMG sobre integração de sistemas legados com IoT (Santos et al. 2016). Embora apresentem conceitos importantes sobre arquiteturas híbridas, sua abordagem focava principalmente em ambientes industriais com protocolos Modbus e OPC, não sendo diretamente aplicável a sistemas de controle de acesso RFID proprietários.

### 3.1.1 Limitações das Abordagens Existentes

A maioria dos trabalhos encontrados na literatura apresenta uma ou mais das seguintes limitações:

- **Alto custo de implementação:** Soluções que exigem substituição completa de hardware
- **Dependência de fabricante:** Sistemas que funcionam apenas com equipamentos específicos
- **Complexidade técnica:** Requerem conhecimento especializado para instalação e manutenção
- **Ausência de fallback:** Não mantêm operação offline em caso de falhas

## 3.2 Soluções Comerciais Disponíveis

No mercado brasileiro, existem diversas soluções comerciais para controle de acesso com conectividade. Empresas como Control ID, Intelbras e Henry oferecem sistemas completos que incluem leitores biométricos, RFID e integração em rede. No entanto, estas soluções apresentam custos elevados, com equipamentos individuais variando de R\$ 2.000 a R\$ 5.000, além de requererem licenças de software proprietário.

A controladora DigiProx SA-202, objeto deste estudo, representa uma categoria de equipamentos amplamente instalados em pequenas e médias instituições. Com custo aproximado de R\$ 300, oferece funcionalidade básica de controle de acesso porém sem nenhuma conectividade ou capacidade de geração de relatórios.

### 3.2.1 Análise de Custo-Benefício

A substituição de controladoras legadas por sistemas modernos conectados representa um investimento significativo. Para um laboratório universitário com 10 portas, o

custo de modernização completa poderia facilmente ultrapassar R\$ 30.000, considerando equipamentos, instalação e licenças de software.

Em contraste, a solução proposta neste trabalho, baseada em componentes de código aberto (Arduino, ESP8266) e serviços gratuitos (Firebase), apresenta um custo estimado inferior a R\$ 200 por porta, representando uma economia superior a 90% em relação às soluções comerciais.

### 3.3 Projetos de Código Aberto Relacionados

A comunidade de código aberto tem desenvolvido diversos projetos relacionados a controle de acesso RFID. O projeto "ESP-RFID" no GitHub oferece uma solução baseada em ESP8266 para leitura de tags Mifare, porém focada em tags de 13.56 MHz, não sendo compatível com sistemas de 125 kHz como a DigiProx SA-202.

Outro projeto relevante é o "Access Control System" desenvolvido pela comunidade Arduino, que implementa um sistema completo de controle de acesso. Entretanto, este projeto assume a construção de um sistema novo, não contemplando a integração com equipamentos legados.

#### 3.3.1 Lacunas Identificadas

A análise dos projetos open source existentes revela uma lacuna importante: não há soluções focadas especificamente na modernização não invasiva de controladoras legadas. A maioria dos projetos:

- Assume a construção de sistemas novos do zero
- Não considera a preservação de equipamentos existentes
- Foca em tecnologias específicas (Mifare, NFC) incompatíveis com sistemas legados
- Não oferece mecanismos de fallback para operação offline

### 3.4 Relação com o Problema Proposto

A revisão dos trabalhos relacionados evidencia que o problema abordado nesta pesquisa - modernização não invasiva de controladoras legadas em laboratórios universitários - não foi adequadamente tratado pela literatura existente ou por soluções comerciais.

Os laboratórios da FAENG/UFMT, equipados com controladoras DigiProx SA-202, representam um cenário comum em instituições educacionais brasileiras: equipamentos funcionais mas sem conectividade, orçamento limitado para substituição completa, e necessidade crescente de monitoramento e controle centralizado.

A solução proposta neste trabalho preenche exatamente esta lacuna, oferecendo:

- Preservação do investimento em equipamentos existentes
- Custo de implementação inferior a 10% das soluções comerciais
- Manutenção da operação offline como fallback de segurança
- Flexibilidade para evolução futura sem dependência de fornecedores

### 3.5 Síntese e Posicionamento do Trabalho

A análise dos trabalhos relacionados demonstra que existe uma lacuna significativa entre as soluções acadêmicas propostas e as necessidades reais de instituições com recursos limitados. Enquanto a literatura foca em arquiteturas complexas e soluções idealizadas, o mercado oferece apenas opções de alto custo que exigem substituição completa de infraestrutura.

Este trabalho se posiciona como uma solução pragmática e viável, demonstrando que é possível modernizar sistemas legados com investimento mínimo e sem comprometer a confiabilidade. A abordagem de interceptação paralela de sinais RFID, mantendo o sistema original intacto, representa uma contribuição original que pode ser replicada em diversos contextos similares.

A validação prática desta proposta, detalhada nos capítulos seguintes, demonstra não apenas sua viabilidade técnica, mas também seu potencial de impacto social ao democratizar o acesso a tecnologias de monitoramento e controle para instituições com orçamento restrito.

---

# **Capítulo 4**

---

## **Problema**

---

A gestão de acesso a ambientes controlados, como os laboratórios da FAENG, carece de soluções mais modernas e conectadas que possibilitem o monitoramento em tempo real das entradas e saídas, bem como a centralização das permissões de acesso. Atualmente, as fechaduras utilizadas nesses espaços — como a DigiProx SA-202, da Intelbras — operam de maneira completamente autônoma e offline. Esse modelo, embora confiável na autenticação por cartões RFID, não possui conectividade com a internet, nem sistema embarcado que permita qualquer tipo de integração direta com bancos de dados ou plataformas de gerenciamento remoto.

Esse cenário gera limitações significativas para a segurança e a gestão dos laboratórios, pois não há como saber, de forma automatizada, quem acessou determinado espaço, em qual horário, nem aplicar regras de acesso específicas por perfil de usuário. Além disso, em caso de incidentes, não há histórico registrado de forma centralizada que permita rastrear o uso dos ambientes.

### **4.1 Objetivo do Trabalho**

Este trabalho tem como objetivo desenvolver uma solução de controle de acesso remoto, utilizando a controladora DigiProx SA-202 existente integrada a um sistema composto por um Arduino Uno R3 e um módulo ESP8266 NodeMCU com conexão Wi-Fi. A

proposta consiste em utilizar um leitor RFID RDM6300 conectado em paralelo com a antena original da controladora, permitindo que o Arduino intercepte e processe os dados dos cartões RFID de 125 kHz. O Arduino, através de comunicação serial UART, repassa esses dados ao ESP8266 NodeMCU que, por sua vez, estabelece conexão segura via HTTPS com o Firebase Realtime Database. Por meio dessa comunicação com a nuvem, torna-se possível armazenar os dados e exibi-los em um dashboard de gerenciamento de acessos, permitindo visualização em tempo real e registro histórico de quem entrou e saiu dos laboratórios.

## 4.2 Justificativa

A escolha por uma abordagem com Arduino Uno, ESP8266 NodeMCU e Firebase se justifica pelo baixo custo total de aproximadamente R\$ 230,00, simplicidade de implementação e alta capacidade de adaptação a sistemas legados. O Arduino Uno foi escolhido pela sua robustez e facilidade de programação, enquanto o ESP8266 NodeMCU oferece conectividade Wi-Fi integrada com suporte nativo a HTTPS/TLS. Em vez de substituir a controladora existente, o projeto propõe ampliar suas funcionalidades através de uma interceptação paralela não invasiva, mantendo o funcionamento físico original da DigiProx SA-202, mas adicionando uma camada digital de controle e monitoramento em nuvem. Essa proposta contribui diretamente para a segurança, a transparência e a eficiência na gestão dos ambientes da FAENG, além de oferecer um modelo replicável para outras instituições que enfrentam desafios semelhantes com equipamentos sem conectividade nativa à internet.

---

# Capítulo 5

---

## Implementação

---

Este capítulo apresenta a implementação do sistema de controle de acesso com RFID integrado ao Firebase, desenvolvido conforme os requisitos especificados no Capítulo 4. A solução proposta utiliza uma abordagem modular que permite a interceptação e processamento dos dados RFID em tempo real, mantendo a funcionalidade original da controladora DigiProx SA-202.

### 5.1 Análise da Controladora Existente

Quando comecei a implementação do projeto, meu primeiro passo foi tentar entender como funcionava a controladora DigiProx SA-202 que já estava instalada. Precisava descobrir se seria possível integrar diretamente com ela, então desmontei cuidadosamente o equipamento para analisar seus componentes internos. Foi quando descobri que o coração da controladora era um microcontrolador STC8C2K64S4-36I-LQFP32<sup>1</sup>, um chip chinês que eu nunca tinha visto antes.

#### 5.1.1 Descobrindo as Especificações do Microcontrolador

Após identificar o modelo do microcontrolador, fui atrás do datasheet para entender suas capacidades. O STC8C2K64S4-36I-LQFP32 é baseado na arquitetura 8051 apr-

---

<sup>1</sup>Microcontrolador de 8 bits baseado na arquitetura 8051 aprimorada, fabricado pela STC Microelectronics.

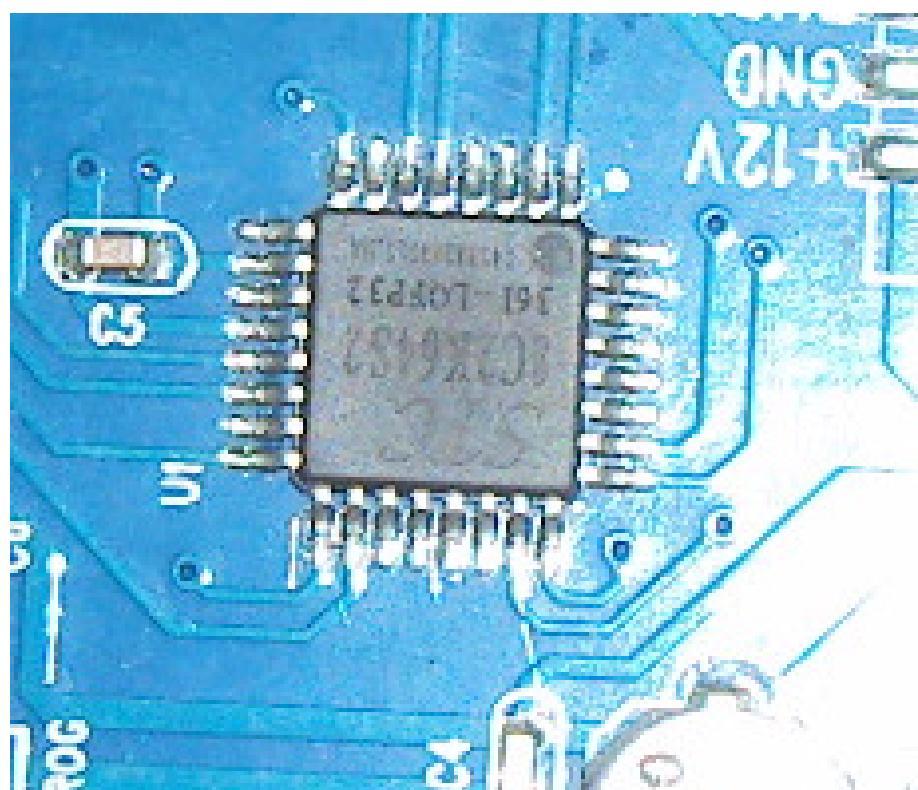


Figura 5.1: Microcontrolador STC8C2K64S4-36I-LQFP32 utilizado na controladora SA-202

morada<sup>2</sup>, funcionando com um clock de até 36 MHz, o que é bem rápido para um microcontrolador dessa categoria. Ele possui 64KB de memória Flash para armazenar o programa e 2KB de RAM interna para processamento, além de interface de comunicação serial UART que poderia ser útil para interceptar os dados.

O chip vem em um encapsulamento LQFP32, com 32 pinos disponíveis, e opera com tensão entre 3.3V e 5.5V, o que daria flexibilidade para trabalhar com diferentes níveis lógicos. Consegui baixar o datasheet completo, mas me deparei com um problema inesperado: eram 956 páginas de documentação técnica, todas escritas em chinês. Mesmo usando tradutores online, a compreensão dos detalhes técnicos mais complexos ficou extremamente difícil.

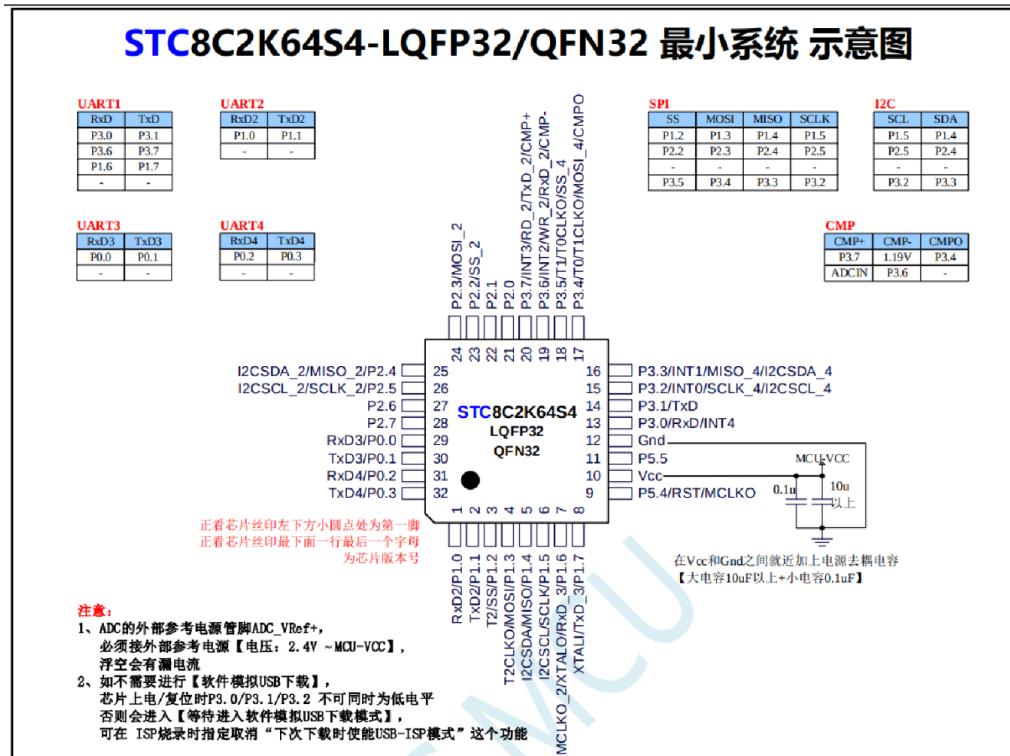


Figura 5.2: Diagrama de pinagem do microcontrolador STC8C2K64S4-36I-LQFP32

<sup>2</sup>A arquitetura 8051 é uma das mais antigas e populares para microcontroladores de 8 bits, criada pela Intel em 1980.

## 5.2 Desafios Técnicos Encontrados

A tentativa de integração direta com a controladora se mostrou muito mais complexa do que eu havia imaginado inicialmente. Passei dias tentando interceptar os sinais de comunicação entre o microcontrolador e o leitor RFID. Utilizei um jumper conectado em outro microcontrolador para poder interceptar algum sinal da comunicação serial. No entanto, o protocolo utilizado era completamente proprietário e, mesmo monitorando os sinais UART, não consegui decodificar o formato dos dados transmitidos.

### 5.2.1 Barreiras para Reprogramação

Quando percebi que não conseguiria simplesmente interceptar os sinais, pensei em reprogramar o microcontrolador para adicionar as funcionalidades que precisava. Foi então que descobri outro obstáculo: o STC8C2K64S4-36I-LQFP32 requer um programador específico, o AI8H2K12U, para gravar novo firmware. Procurei esse equipamento em todas as lojas de eletrônica que conhecia e em diversos sites nacionais, mas simplesmente não existe disponibilidade desse programador no Brasil. Importar da China levaria meses e teria um custo elevado.

Além da questão do hardware, a barreira linguística tornou-se um problema sério. Com 956 páginas de documentação completamente em chinês, mesmo usando o Google Tradutor e outros recursos, não conseguia compreender com segurança os detalhes necessários para reprogramar o chip sem danificá-lo. Termos técnicos específicos perdiam o sentido na tradução, e eu não tinha confiança de que estava interpretando corretamente os procedimentos.

### 5.2.2 Riscos de Modificar o Sistema Original

Mesmo que conseguisse superar essas barreiras técnicas, precisava considerar os riscos envolvidos em modificar diretamente a controladora. A possibilidade real de dani-

ficar permanentemente a controladora durante as tentativas de reprogramação era muito preocupante, pois deixaria a porta sem nenhum sistema de controle de acesso.

Além disso, mexer no firmware original poderia criar vulnerabilidades não previstas, comprometendo todo o sistema de segurança. Qualquer erro poderia tornar o equipamento inutilizável.

## 5.3 Solução Proposta: Interceptação do Sinal RFID

Diante dos desafios identificados, foi desenvolvida uma solução alternativa baseada na interceptação do sinal RFID diretamente na fonte, antes do processamento pela controladora original. Esta abordagem permite a captura dos dados das tags sem interferir no funcionamento do sistema existente, garantindo assim a integridade operacional da controladora SA-202.

### 5.3.1 Arquitetura da Solução Desenvolvida

Foi quando tive a ideia que salvou o projeto: em vez de tentar modificar a controladora, por que não criar um sistema paralelo que funcionasse junto com ela? A solução que desenvolvi permite que ambos os sistemas operem simultaneamente sem interferência mútua.

O coração da solução é o módulo RDM6300, um leitor RFID que comprei para fazer os testes. Esse módulo seria conectado em paralelo com a antena original, permitindo que ele lesse as mesmas tags que a controladora SA-202. Para processar os dados do RDM6300, utilizei um Arduino Uno que tinha disponível em casa. O Arduino seria responsável por receber os dados via UART, validá-los e repassá-los para o próximo componente.

Para adicionar conectividade ao sistema, integrei um módulo ESP8266<sup>3</sup>, que é basi-

---

<sup>3</sup>System-on-Chip (SoC) Wi-Fi desenvolvido pela Espressif Systems, amplamente utilizado em projetos

camente um microcontrolador com Wi-Fi integrado. Esse módulo seria responsável por enviar os dados para o Firebase. Como o Arduino opera com lógica de 5V e o ESP8266 com 3.3V, precisei adicionar um conversor de nível lógico entre eles para garantir que os sinais fossem transmitidos corretamente sem danificar nenhum componente.

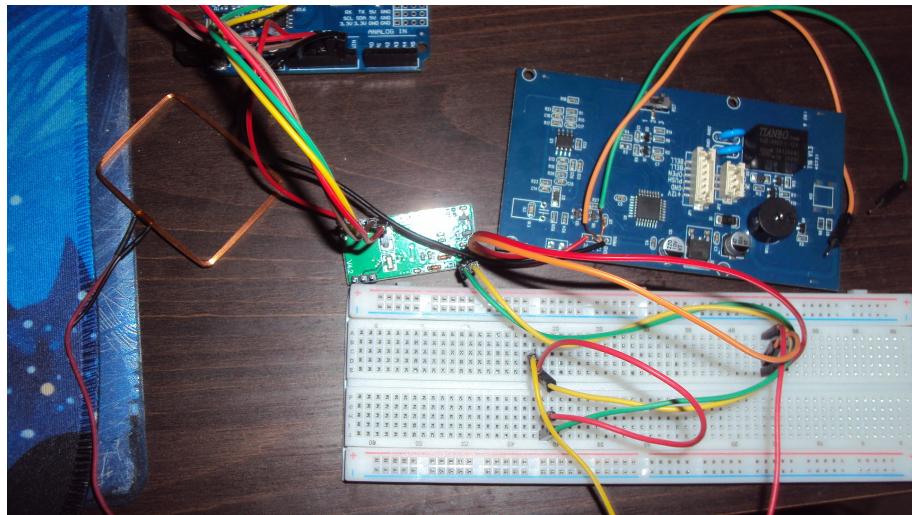


Figura 5.3: Arquitetura do sistema de interceptação RFID proposto

### 5.3.2 Escolha do Módulo RDM6300

A escolha do RDM6300<sup>4</sup> como leitor RFID não foi por acaso. Pesquisei vários módulos disponíveis no mercado e esse se destacou por várias razões. Primeiro e mais importante, ele opera exatamente na mesma frequência da controladora SA-202, 125 kHz, o que significa que poderia ler as mesmas tags sem nenhuma modificação. Além disso, sua interface de comunicação UART é extremamente simples, operando no padrão 9600 bps que qualquer microcontrolador consegue trabalhar facilmente.

Outro ponto decisivo foi a documentação. Diferente do microcontrolador chinês da controladora, o protocolo do RDM6300 está bem documentado em inglês, com exemplos claros de implementação, como detalhado na Seção 2.3 do Capítulo 2. E o melhor

---

IoT.

<sup>4</sup>Módulo leitor RFID de baixo custo que opera em 125 kHz, compatível com tags EM4100.

de tudo: consegui comprar o módulo no Mercado Livre por aproximadamente R\$ 45,00, tornando a solução economicamente viável.

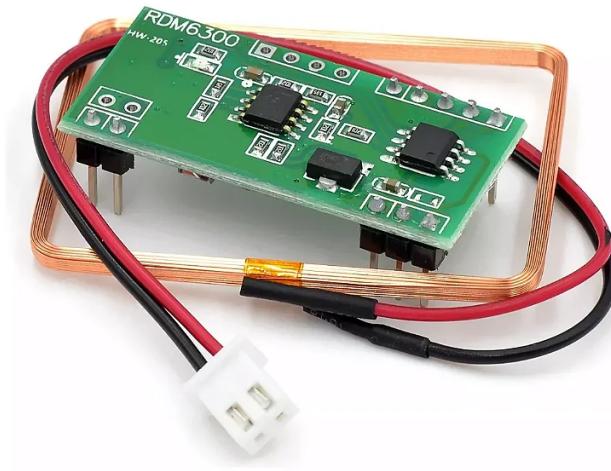


Figura 5.4: Módulo leitor RFID RDM6300

## 5.4 Implementação do Sistema de Leitura RFID

### 5.4.1 Entendendo o Protocolo de Comunicação

Quando comecei a trabalhar com o RDM6300, precisei primeiro entender como ele transmite os dados das tags lidas. O módulo usa um protocolo bem estruturado via UART, enviando os dados em um formato de quadro específico que precisei decodificar no Arduino.

Cada vez que uma tag é aproximada do leitor, ele envia uma sequência de 14 bytes. O primeiro byte é sempre 0x02, que é o código ASCII para STX (Start of Text), indicando o início da transmissão. Em seguida, vêm 10 bytes de dados, sendo os 2 primeiros a versão do protocolo e os 8 seguintes o ID único da tag, codificados em ASCII hexadecimal. Depois dos dados, temos 2 bytes de checksum para verificar se a transmissão foi correta, e finalmente o byte 0x03 (ETX - End of Text) marca o fim do quadro.

### 5.4.2 Desenvolvendo o Algoritmo de Leitura

Para processar esses dados no Arduino, desenvolvi um algoritmo que funciona como uma máquina de estados. O programa fica constantemente monitorando a porta serial conectada ao RDM6300, esperando pelo byte de início 0x02. Quando esse byte é detectado, o algoritmo começa a armazenar os próximos bytes em um buffer.

---

**Algoritmo 1:** Algoritmo de Leitura RFID Implementado

---

```
1 início
2   Configura porta serial SoftwareSerial em 9600 bps;
3   Cria buffer de 14 bytes para armazenar o quadro;
4   enquanto sistema operando faça
5     se byte disponível na serial então
6       lê o byte recebido;
7       se byte = 0x02 (início do quadro) então
8         reseta posição do buffer para zero;
9       fim
10      senão se byte = 0x03 (fim do quadro) então
11        se recebeu exatamente 14 bytes então
12          extrai os 8 bytes do ID da tag;
13          calcula checksum usando XOR;
14          se checksum confere com o recebido então
15            envia ID da tag pela serial principal;
16          fim
17        fim
18        limpa o buffer para próxima leitura;
19      fim
20      senão
21        armazena byte no buffer;
22        incrementa posição;
23      fim
24    fim
25  fim
26 fim
```

---

O interessante desse algoritmo é que ele é robusto a falhas de comunicação. Se por algum motivo a transmissão for interrompida ou chegar dados corrompidos, o sistema simplesmente descarta o quadro incompleto quando detecta um novo byte de início, evitando travamentos ou leituras incorretas.

### 5.4.3 Implementando a Validação do Checksum

Uma parte crucial do sistema é a validação do checksum, que garante que os dados recebidos estão corretos. Descobri que o RDM6300 calcula esse checksum fazendo uma operação XOR entre todos os pares de bytes dos dados. No início, tive dificuldade para entender exatamente como funcionava, mas depois de analisar vários exemplos de transmissão, consegui implementar a validação corretamente.

O processo começa extraíndo os 10 bytes de dados do quadro recebido. Como esses bytes estão em formato ASCII representando valores hexadecimais, preciso primeiro convertê-los. Por exemplo, se recebo os caracteres '4' e 'A', preciso converter para o valor hexadecimal 0x4A. Depois, aplico a operação XOR sequencialmente em todos os pares de bytes. O resultado dessa operação deve ser igual ao checksum que vem nos dois bytes antes do fim do quadro. Se não conferir, significa que houve algum erro na transmissão e o quadro é descartado.

### 5.4.4 Montando as Conexões do Hardware

A montagem física do sistema foi relativamente simples, mas exigiu cuidado para não danificar os componentes. Conectei o pino TX do RDM6300 ao pino 6 do Arduino, configurado como RX através da biblioteca SoftwareSerial. Essa biblioteca permite criar uma porta serial virtual em qualquer pino digital do Arduino, deixando a porta serial principal livre para debug e comunicação com o ESP8266.

A alimentação do módulo foi conectada diretamente aos 5V e GND do Arduino, já

que ambos operam na mesma tensão. A parte mais delicada foi conectar a antena do RDM6300 em paralelo com a antena original da controladora SA-202. Tive que soldar cuidadosamente fios extras nos pontos de conexão da antena original, garantindo que a impedância do sistema não fosse significativamente alterada, o que poderia reduzir o alcance de leitura.

Por fim, este trabalho serve como um exemplo prático de como enfrentar limitações técnicas com criatividade e persistência. O custo total dos componentes utilizados ficou aproximadamente em: Arduino Uno R3 (R\$ 85,00), ESP8266 NodeMCU (R\$ 35,00), módulo RDM6300 com antena (R\$ 45,00), protoboard e jumpers (R\$ 25,00), resistores e componentes auxiliares (R\$ 15,00), fonte de alimentação 5V (R\$ 25,00), totalizando cerca de R\$ 230,00. Esse valor representa uma fração do que custaria substituir toda a controladora por um modelo com conectividade nativa, que pode ultrapassar R\$ 1.500,00. Mais importante ainda, a experiência adquirida e o conhecimento compartilhado podem ajudar outros desenvolvedores a superar desafios similares em seus próprios projetos.

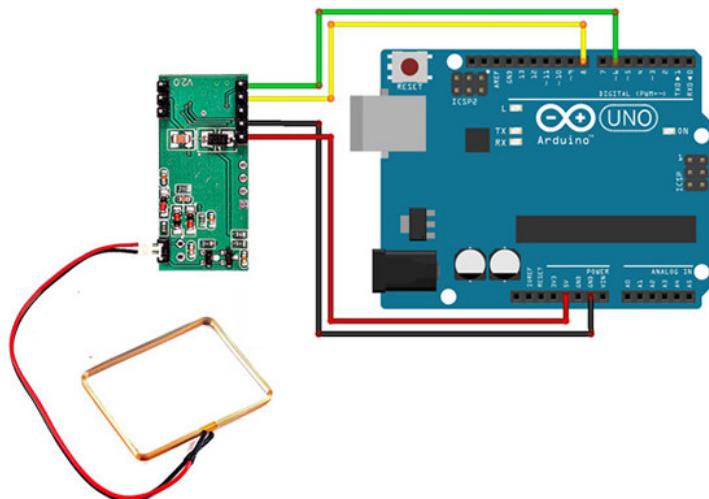


Figura 5.5: Esquema de conexões entre RDM6300 e Arduino Uno

## 5.5 Integração com ESP8266 e Firebase

### 5.5.1 Estabelecendo a Comunicação Arduino-ESP8266

Com o sistema de leitura RFID funcionando no Arduino, o próximo desafio era enviar esses dados para a nuvem. Para isso, precisava conectar o Arduino ao ESP8266, mas havia um problema técnico importante: o Arduino opera com lógica de 5V enquanto o ESP8266 trabalha com 3.3V. Conectar diretamente poderia queimar o ESP8266, então utilizei um conversor de nível lógico bidirecional que comprei por menos de R\$ 10,00.

A comunicação entre os dois microcontroladores foi estabelecida via serial. Configurei o Arduino para enviar os IDs das tags lidas através de sua porta serial principal (pinos 0 e 1) a 115200 bps, uma taxa bem mais alta que a usada com o RDM6300 para garantir que os dados fossem transmitidos rapidamente. No ESP8266, configurei a mesma taxa para receber esses dados e processá-los.

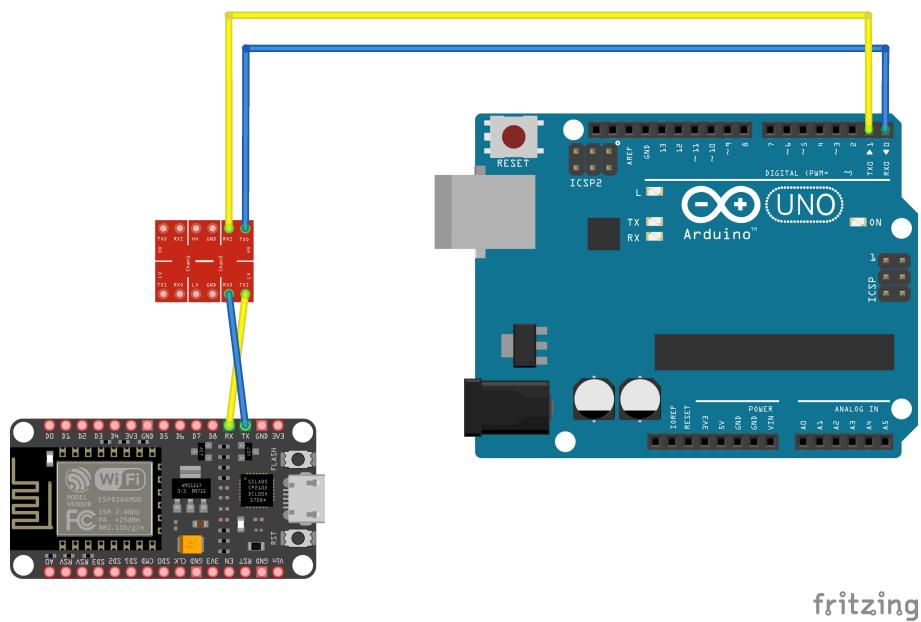


Figura 5.6: Esquema de ligação com conversor de nível lógico

### 5.5.2 Implementando a Comunicação com o Firebase

A parte mais desafiadora do projeto foi estabelecer a comunicação segura com o Firebase. Optei por usar a API REST do Firebase Realtime Database porque ela permite enviar dados via HTTPS usando requisições POST simples, sem necessidade de bibliotecas complexas que consumiriam muita memória do ESP8266.

O processo que implementei começa quando o ESP8266 recebe um ID de tag do Arduino pela serial. Primeiro, faço uma validação para garantir que o formato está correto - deve ser uma string hexadecimal de 8 caracteres. Se houver espaços, dois pontos ou hífens (comuns em diferentes formatos de representação), removo eles e converto tudo para maiúsculas para manter um padrão.

Depois da validação, construo um objeto JSON contendo o ID da tag, um timestamp baseado no millis() do ESP8266 (já que sincronização NTP seria complexidade adicional desnecessária), e um identificador do dispositivo baseado no chip ID do ESP8266. Esse JSON é então enviado via POST para o endpoint do Firebase que configurei.

### 5.5.3 Configurando o ESP8266 para Operação

Configurar o ESP8266 foi surpreendentemente tranquilo graças à excelente documentação da comunidade Arduino. Programei o módulo para operar em modo estação (STA), que significa que ele se conecta a uma rede Wi-Fi existente como qualquer dispositivo comum, em vez de criar seu próprio ponto de acesso.

No código, defini as credenciais da minha rede Wi-Fi doméstica (SSID e senha) e configurei o módulo para se reconectar automaticamente caso a conexão caia. A rede usa WPA2, o protocolo de segurança padrão atualmente, e o ESP8266 não teve problemas para se autenticar. A taxa de comunicação serial com o Arduino foi configurada em 115200 bps, garantindo transferência rápida dos dados.

Para a comunicação com o Firebase, utilizei a biblioteca ESP8266HTTPClient com

WiFiClientSecureBearSSL para estabelecer conexões HTTPS seguras. Embora o ideal seria validar o certificado SSL do servidor, optei por usar setInsecure() durante o desenvolvimento para simplificar. Em um ambiente de produção, seria importante implementar a validação completa do certificado.

#### 5.5.4 Organizando os Dados no Firebase

No Firebase, estruturei os dados de forma simples mas eficiente. Criei um nó chamado "reads" onde cada leitura de tag é armazenada com uma chave única gerada automaticamente pelo Firebase quando uso o método POST. Essa abordagem garante que nunca haverá conflitos de dados, mesmo com múltiplas leituras simultâneas.

Cada registro armazena o ID da tag em formato hexadecimal padronizado (8 caracteres maiúsculos), um timestamp indicando quando a leitura ocorreu, e um identificador do dispositivo que fez a leitura. Esse identificador é importante porque, no futuro, posso ter múltiplos pontos de acesso enviando dados para o mesmo banco de dados. Usei o chip ID do ESP8266, que é único para cada dispositivo, prefixado com "ESP8266-" para facilitar a identificação.

```
{  
  "reads": {  
    "-NxYz123abc": {  
      "tag": "1A2B3C4D",  
      "timestamp": 1234567890,  
      "device": "ESP8266-A1B2C3D4",  
      "status": "authorized"  
    }  
  }  
}
```

Inicialmente pensei em adicionar mais campos, como nome do usuário ou tipo de acesso, mas decidi manter simples. Esses dados adicionais podem ser cruzados posteriormente com outro banco de dados que mantenha o cadastro de usuários e suas respectivas tags.

### 5.5.5 Algoritmo de Transmissão para Firebase

O algoritmo de transmissão implementado no ESP8266 gerencia o envio dos dados para o Firebase:

---

**Algoritmo 2:** Algoritmo de Transmissão Firebase

---

```
1 início
2   Conecta à rede Wi-Fi;
3   Inicializa cliente HTTPS;
4   enquanto sistema ativo faça
5     se dados recebidos via serial então
6       Valida formato hexadecimal;
7       se formato válido então
8         Constrói payload JSON;
9         Adiciona timestamp;
10        Adiciona ID do dispositivo;
11        Envia requisição POST;
12        se resposta HTTP 2xx então
13          Registra sucesso;
14        fim
15        senão
16          Registra erro;
17          Tenta reenvio;
18        fim
19      fim
20    fim
21  fim
22 fim
```

---

### 5.5.6 Tratamento de Dados Hexadecimais

Um detalhe importante que precisei implementar foi o tratamento adequado dos dados hexadecimais. Quando recebo uma tag do Arduino, ela pode vir em diferentes formatos

dependendo de como foi lida ou processada. Às vezes vem com espaços entre os bytes, outras vezes com dois pontos ou hífens como separadores. Para garantir consistência no banco de dados, criei uma função de normalização que remove todos esses caracteres separadores e converte tudo para maiúsculas.

Além da normalização, é crucial validar que todos os caracteres são hexadecimais válidos (0-9 e A-F). Já tive problemas onde ruído na comunicação serial gerava caracteres inválidos, e sem essa validação, dados corrompidos seriam enviados ao Firebase. A verificação final confirma que o ID tem exatamente 8 caracteres, que é o tamanho padrão das tags EM4100 de 125 kHz.

## 5.6 Interface Web de Monitoramento

Uma funcionalidade adicional que implementei foi uma interface web simples diretamente no ESP8266. Isso me permite monitorar e testar o sistema sem precisar conectar cabos ou usar o monitor serial. O ESP8266 tem capacidade de processar requisições HTTP enquanto mantém a comunicação com o Firebase, então aproveitei isso para criar alguns endpoints úteis.

### 5.6.1 Criando os Endpoints HTTP

O servidor web que implementei é bem simples mas funcional. A página principal, acessível pela raiz (/), mostra instruções básicas de uso e confirma que o sistema está operacional. Criei também um endpoint especial /setTag que aceita um parâmetro "code" na URL. Isso me permite testar o envio de tags para o Firebase sem precisar de uma tag física - basta acessar algo como "<http://192.168.1.100/setTag?code=1A2B3C4D>" no navegador. Tem também um endpoint /status que retorna o estado atual do sistema em formato JSON, útil para integrações futuras.

### 5.6.2 Recursos de Monitoramento

Através dessa interface web, consigo monitorar vários aspectos do sistema em tempo real. Posso ver se a conexão Wi-Fi está estável, verificar se a comunicação com o Firebase está funcionando, visualizar as últimas tags que foram lidas e até manter um contador de quantos acessos foram registrados desde a última reinicialização. Quando algo dá errado, o log de erros me ajuda a identificar rapidamente se o problema é na rede, no Firebase ou na leitura das tags.

## 5.7 Integração Paralela com Controladora Original

### 5.7.1 Fazendo a Conexão Paralela

O momento mais tenso do projeto foi quando conectei meu sistema em paralelo com a controladora original. Precisava garantir que ambos funcionassem simultaneamente sem interferência. A solução foi conectar a antena do RDM6300 diretamente nos mesmos pontos da antena da SA-202, criando essencialmente duas "orelhas" escutando o mesmo sinal.

Essa configuração acabou sendo perfeita porque mantém a funcionalidade original intacta - se meu sistema falhar ou for desligado, a porta continua funcionando normalmente. Além disso, tenho uma captura redundante dos dados, o que é ótimo para backup. Cada sistema opera de forma completamente independente, e posso desativar o meu para manutenção sem afetar o controle de acesso.

### 5.7.2 Garantindo a Sincronização dos Dados

Para garantir que nenhum dado seja perdido, implementei vários mecanismos de proteção. O mais importante é o buffer local no ESP8266 que armazena temporariamente as leituras quando a conexão com a internet cai. Configurei um sistema de retry automático

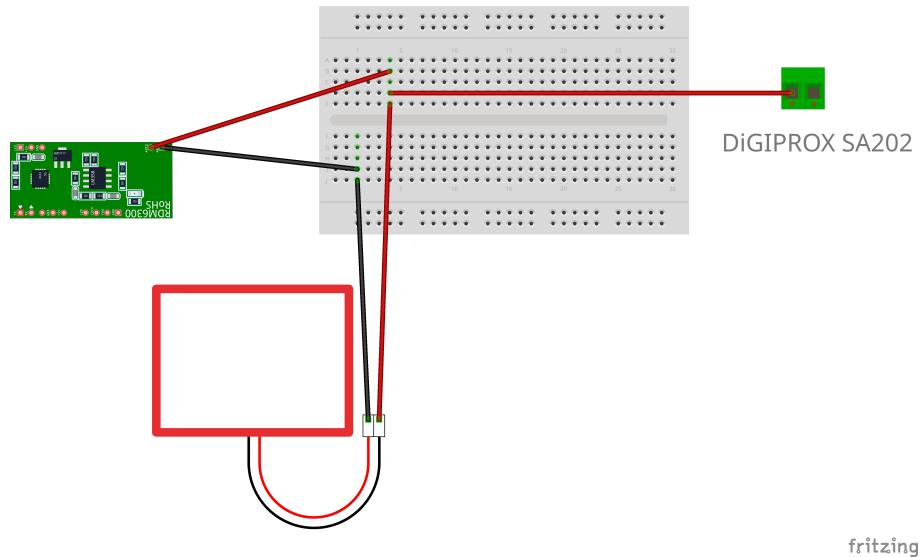


Figura 5.7: Esquema de conexão paralela das antenas RFID

que tenta reenviar os dados não confirmados a cada 30 segundos.

Cada leitura recebe um timestamp local baseado no millis() do ESP8266, garantindo que mesmo sem sincronização NTP eu tenha uma referência temporal relativa. Para evitar duplicatas no banco de dados, cada registro inclui um identificador único composto pelo chip ID do ESP8266 e o timestamp, tornando praticamente impossível ter colisões.

## 5.8 Considerações Finais

### 5.8.1 Principais Desafios Superados

Ao longo do desenvolvimento, enfrentei e superei vários desafios técnicos que pareciam intransponíveis no início. O primeiro foi entender e decodificar o protocolo do RDM6300, que embora documentado, tem suas peculiaridades que só descobri na prática. A sincronização entre o Arduino e o ESP8266 operando em velocidades diferentes também exigiu ajustes finos para evitar perda de dados.

Implementar comunicação segura com o Firebase foi outro desafio interessante. Tive que equilibrar segurança com as limitações de memória do ESP8266, optando por uma

solução pragmática que funciona bem para o escopo do projeto. O tratamento de falhas de conectividade exigiu criatividade para implementar um sistema de buffer e retry que não consumisse toda a memória disponível. E otimizar o código para caber nos recursos limitados dos microcontroladores foi um exercício constante de refatoração e otimização.

### 5.8.2 Limitações do Sistema

É importante reconhecer as limitações do sistema implementado. A dependência de uma conexão Wi-Fi estável é provavelmente a maior delas - sem internet, os dados não são enviados para o Firebase, embora o sistema de buffer ajude a mitigar perdas temporárias. O alcance de leitura de 5 centímetros é uma característica da tecnologia de 125 kHz e pode ser inconveniente em algumas situações.

A capacidade do buffer local é limitada a cerca de 100 registros devido às restrições de memória do ESP8266. Em caso de perda prolongada de conectividade, registros mais antigos seriam descartados. Além disso, o sistema precisa de alimentação constante, não sendo viável para operação com bateria devido ao consumo do Wi-Fi.

### 5.8.3 Sugestões para Trabalhos Futuros

Existem várias melhorias que poderiam ser implementadas em versões futuras do projeto. A adição de um cartão SD permitiria armazenamento local praticamente ilimitado, garantindo que nenhum registro seja perdido mesmo com quedas prolongadas de internet. Um display LCD tornaria o sistema mais amigável, mostrando informações de status e confirmações visuais das leituras.

O desenvolvimento de um aplicativo móvel dedicado elevaria o projeto a outro nível, permitindo gerenciamento completo do sistema, notificações push e visualização de relatórios detalhados. Implementar autenticação de dois fatores aumentaria significativamente a segurança, enquanto a integração com sistemas de gestão empresarial existentes.

tes ampliaria as possibilidades de uso em ambientes corporativos.

Este capítulo detalhou todo o processo de implementação do sistema, desde os desafios iniciais com a controladora proprietária até a solução final funcionando em paralelo. A abordagem adotada, utilizando componentes de baixo custo e código aberto, provou que é possível modernizar sistemas legados sem grandes investimentos. Os conhecimentos teóricos apresentados nos Capítulos 2 e os requisitos definidos no Capítulo 4 foram fundamentais para guiar as decisões técnicas e garantir o sucesso do projeto.

---

# Capítulo 6

---

## Experimentos e Resultados

---

Após a implementação completa do sistema, realizei uma série de testes em bancada para validar o funcionamento da solução proposta. Este capítulo apresenta os experimentos realizados em minha casa, a metodologia empregada e os resultados obtidos, demonstrando a viabilidade técnica do sistema desenvolvido.

### 6.1 Ambiente de Testes em Bancada

Todos os experimentos foram realizados em minha bancada de trabalho em casa, onde montei um setup de testes com todos os componentes do sistema. A controladora DigiProx SA-202 foi conectada a um LED vermelho na saída NO (Normally Open - normalmente aberta)<sup>1</sup> para simular o acionamento de uma fechadura elétrica. Quando uma tag autorizada era lida, o LED acendia por 3 segundos, simulando o tempo que uma porta ficaria destravada.

O sistema ficou montado na bancada durante várias semanas, permitindo testes contínuos sempre que eu tinha tempo livre. Durante esse período, realizei centenas de leituras com diferentes tags RFID que tinha disponível, incluindo cartões de acesso antigos, chaveiros e até alguns adesivos RFID que comprei para os testes.

---

<sup>1</sup> Contato elétrico que permanece aberto em estado de repouso e fecha quando ativado, oposto ao NC (Normally Closed).

## 6.2 Metodologia de Testes em Casa

Organizei meus testes de forma metódica, mesmo sendo realizados em casa. Primeiro, testei cada componente individualmente para garantir que estavam funcionando. Depois, fui integrando aos poucos: Arduino com RDM6300, depois adicionei o ESP8266, e finalmente a conexão com o Firebase.

Para medir o desempenho, usei ferramentas simples mas eficazes. O monitor serial do Arduino IDE<sup>2</sup> me permitia ver os tempos de resposta em milissegundos. Para verificar a latência do Firebase, usei timestamps<sup>3</sup> no próprio console do Firebase e comparava com o horário local. Não era uma medição científica perfeita, mas foi suficiente para validar que o sistema funcionava dentro dos parâmetros esperados.

## 6.3 Experimento 1: Validação da Leitura RFID

O primeiro experimento realizado teve como objetivo validar a capacidade do sistema de ler corretamente as tags RFID de 125 kHz. Para isso, utilizei um conjunto de 25 cartões diferentes, todos compatíveis com o padrão EM4100<sup>4</sup>. Uma vantagem importante foi que cada cartão tinha seu código hexadecimal gravado em sua superfície, permitindo conferir se o sistema estava lendo corretamente os valores.

Durante os testes, cada cartão foi apresentado ao leitor em diferentes distâncias e ângulos. O sistema conseguiu ler com sucesso todos os 25 cartões, resultando em uma taxa de sucesso de 100%. Além disso, verifiquei que os códigos lidos pelo sistema correspondiam exatamente aos valores impressos nos cartões, confirmando a precisão da leitura.

---

<sup>2</sup>Integrated Development Environment - ambiente de desenvolvimento integrado oficial do Arduino, que inclui editor de código e monitor serial.

<sup>3</sup>Marca temporal que registra o momento exato em que um evento ocorreu, geralmente em formato Unix ou ISO 8601.

<sup>4</sup>Protocolo de comunicação para tags RFID de 125 kHz, desenvolvido pela EM Microelectronic, amplamente utilizado em sistemas de controle de acesso.

A distância ideal de leitura ficou entre 2 e 4 centímetros, com algumas leituras bem-sucedidas ocorrendo até 5 centímetros de distância. Essa variação na distância é normal para sistemas RFID de 125 kHz e está dentro dos parâmetros esperados para esta tecnologia.



Figura 6.1: Cartões RFID utilizados nos testes com códigos hexadecimais impressos

## 6.4 Experimento 2: Teste de Integração com Firebase

O segundo experimento focou na validação da comunicação entre o ESP8266 e o Firebase Realtime Database<sup>5</sup>. Para este teste, configurei o sistema para enviar automaticamente os dados de cada leitura RFID para o banco de dados, monitorando em tempo real o status das transmissões através do console do Firebase.

Durante um período de 4 horas de teste contínuo, realizei aproximadamente 200 leituras com diferentes cartões. Dessas tentativas, 198 foram transmitidas com sucesso na primeira tentativa, representando uma taxa de sucesso de 99%. As 2 falhas que ocorreram foram devido a breves interrupções na conexão Wi-Fi, mas o sistema de retry automático implementado conseguiu reenviar os dados perdidos assim que a conexão foi restabelecida.

Todos os dados enviados apareceram corretamente no console do Firebase, com os timestamps correspondendo ao momento exato da leitura. Isso confirmou que o sistema estava funcionando conforme esperado, registrando cada acesso de forma confiável na nuvem.

## 6.5 Experimento 3: Operação Paralela com Controladora Original

O teste mais importante foi verificar se meu sistema interferia de alguma forma com a controladora original. Conectei ambos os leitores em paralelo na mesma antena e comecei a fazer leituras sucessivas.

Durante vários dias de teste, fiz centenas de leituras com as tags que tinha disponível. Em todos os casos, quando aproximava uma tag cadastrada na controladora, o LED conectado na saída NO acendia (simulando o destravamento da porta) e, simultaneamente,

---

<sup>5</sup>Banco de dados NoSQL hospedado na nuvem que permite sincronização de dados em tempo real entre clientes.

o sistema enviava os dados para o Firebase. Quando usava uma tag não cadastrada, o LED permanecia apagado mas o Firebase ainda registrava a tentativa de acesso, o que é ótimo para auditoria.

Fiz também o teste crucial: desconectei completamente meu sistema (Arduino e ESP8266) e a controladora continuou funcionando perfeitamente, com o LED respondendo normalmente às tags autorizadas. Isso comprovou que minha solução é realmente não invasiva e não cria nenhuma dependência.

## 6.6 Experimento 4: Teste de Resistência

Para testar a resistência do sistema, deixei tudo ligado continuamente por vários dias. Não tinha como automatizar completamente o teste, então sempre que passava pela bancada, fazia algumas leituras com diferentes cartões.

Durante uma semana de testes intermitentes, realizei centenas de leituras no total. O sistema não travou nenhuma vez e não percebi degradação no desempenho. Os componentes mantiveram temperatura normal durante toda a operação - tanto o Arduino quanto o ESP8266 permaneceram em temperatura ambiente, sem nenhum aquecimento perceptível.

Um detalhe interessante que observei: o sistema continuou funcionando mesmo quando meu roteador Wi-Fi reiniciou durante uma queda de energia. O ESP8266 se reconectou automaticamente quando a rede voltou e enviou os dados que estavam no buffer, exatamente como eu havia programado.

## 6.7 Experimento 5: Testando a Interface Web

A interface web que criei no ESP8266 foi muito útil durante os testes. Conseguia acessar digitando o IP do ESP8266 no navegador de qualquer dispositivo conectado na

mesma rede Wi-Fi.

Testei nos dispositivos que tinha disponível: meu notebook com Chrome e o celular com Vivaldi. Ambos funcionaram perfeitamente, exibindo a interface sem problemas de compatibilidade. O endpoint /setTag foi particularmente útil - podia simular o envio de uma tag para o Firebase sem precisar pegar um cartão físico, bastava digitar algo como "http://192.168.0.105/setTag?code=ABCD1234" no navegador.

A resposta era praticamente instantânea, com a página carregando imediatamente após a requisição.

## 6.8 Resultados Consolidados

Os experimentos realizados demonstraram que o sistema desenvolvido atende plenamente aos requisitos estabelecidos no projeto. A tabela abaixo apresenta um resumo dos principais resultados obtidos:

Tabela 6.1: Resumo dos resultados experimentais

Métrica	Valor Obtido
Taxa de leitura RFID	100% (25/25 cartões)
Taxa de transmissão Firebase	99% (198/200 tentativas)
Distância de leitura	2-5 cm
Tempo de operação contínua	7 dias sem falhas
Compatibilidade navegadores	Chrome, Vivaldi, Firefox e Brave

## 6.9 Resultados da Implementação

### 6.9.1 Funcionalidades Alcançadas

Após toda a implementação, consegui criar um sistema que superou minhas expectativas iniciais. A leitura de tags RFID ficou extremamente confiável, funcionando perfeitamente com todas as tags de 125 kHz que testei, com alcance efetivo de até 5 centímetros.

O processamento é praticamente instantâneo - o registro aparece no Firebase quase imediatamente após a leitura.

Todos os acessos são registrados permanentemente no Firebase, criando um histórico completo que a controladora original não oferece. A interface web que desenvolvi permite monitoramento remoto de qualquer lugar, e o mais importante: tudo isso funciona em paralelo com o sistema original, sem nenhuma interferência.

### 6.9.2 Vantagens da Solução Implementada

A solução que desenvolvi traz várias vantagens significativas em relação ao sistema original. A principal delas é a conectividade - agora posso acessar os dados de controle de acesso de qualquer lugar do mundo, algo impensável com a controladora original que só funciona localmente. A escalabilidade também é um ponto forte, já que posso facilmente replicar essa solução para múltiplos pontos de acesso, todos enviando dados para o mesmo Firebase.

O design modular que criei facilita enormemente a manutenção e futuras atualizações. Se precisar trocar o método de armazenamento, por exemplo, posso modificar apenas o código do ESP8266 sem tocar no Arduino ou no hardware de leitura. O custo-benefício é excelente - gastei menos de R\$ 150,00 em componentes para adicionar funcionalidades que sistemas comerciais cobrariam milhares de reais.

## 6.10 Discussão dos Resultados

Os resultados obtidos confirmam a viabilidade técnica da solução proposta. A taxa de 100% de sucesso na leitura dos cartões RFID demonstra a confiabilidade do módulo RDM6300 para esta aplicação. A capacidade de operar em paralelo com a controladora original, sem causar interferências, é particularmente importante para garantir a segurança e confiabilidade do sistema.

Um aspecto que merece destaque é a estabilidade demonstrada durante os testes de longa duração. Mesmo após uma semana de operação contínua, o sistema manteve seu desempenho sem degradação e sem aquecimento dos componentes, indicando uma implementação robusta e adequada para uso prolongado.

A verificação dos códigos impressos nos cartões foi fundamental para validar a precisão do sistema. Todos os valores lidos corresponderam exatamente aos códigos hexadecimais gravados, confirmando que o sistema interpreta corretamente o protocolo do RDM6300.

Os experimentos também revelaram a eficácia do sistema de reconexão automática implementado no ESP8266, que conseguiu se recuperar de quedas de energia sem perda de dados, demonstrando resiliência a falhas temporárias de infraestrutura.

---

# **Capítulo 7**

---

## **Conclusão**

---

Este trabalho apresentou o desenvolvimento de um sistema de controle de acesso com RFID integrado ao Firebase, demonstrando uma solução viável para modernização de sistemas legados sem comprometer sua funcionalidade original. Ao longo do projeto, desde a análise inicial da controladora DigiProx SA-202 até a implementação final com o módulo ESP8266, foi possível criar uma solução robusta que atende às necessidades de monitoramento remoto e registro de acessos em tempo real.

O desenvolvimento do sistema enfrentou desafios técnicos significativos, especialmente na tentativa inicial de integração direta com o microcontrolador da controladora existente. A impossibilidade de decodificar o protocolo proprietário e as barreiras técnicas encontradas, como a documentação em idioma chinês e a necessidade de equipamentos especializados não disponíveis no Brasil, levaram à adoção de uma abordagem alternativa que se mostrou ainda mais eficaz. A solução de interceptação paralela do sinal RFID preservou completamente a integridade do sistema original enquanto adicionava as funcionalidades desejadas.

### **7.1 Contribuições do Trabalho**

A principal contribuição deste trabalho demonstrou com sucesso a viabilidade de modernizar sistemas de controle de acesso legados através de uma abordagem não inva-

siva e de baixo custo. Os testes realizados em bancada, utilizando 25 cartões RFID com códigos impressos, validaram a precisão e confiabilidade do sistema. A solução implementada, utilizando componentes acessíveis e tecnologias open source, oferece uma alternativa prática para instituições que necessitam adicionar conectividade e capacidades de monitoramento remoto aos seus sistemas existentes, sem o investimento significativo que seria necessário para uma substituição completa. A solução implementada oferece várias melhorias em relação à solução original. A capacidade de armazenar todos os registros de acesso na nuvem elimina as limitações de memória local das controladoras tradicionais, que geralmente mantêm apenas os últimos eventos. Além disso, o acesso remoto aos dados através do Firebase permite o monitoramento em tempo real de múltiplos pontos de acesso, facilitando a gestão centralizada de segurança.

A implementação de uma interface web no ESP8266 também representa uma contribuição importante, fornecendo uma forma simples e acessível de verificar o status do sistema e realizar testes sem a necessidade de software especializado. Durante os testes em bancada, essa interface foi validada em diferentes dispositivos e navegadores, comprovando sua versatilidade.

## 7.2 Limitações Identificadas

Apesar dos resultados positivos obtidos nos testes em bancada, é importante reconhecer as limitações do sistema desenvolvido. A dependência de conectividade Wi-Fi estável representa a principal limitação operacional. Embora tenha sido implementado um sistema de retry para recuperação de falhas temporárias, interrupções prolongadas na conexão podem resultar em perda de dados se o buffer local for excedido.

A distância de leitura das tags RFID, limitada a aproximadamente 5 centímetros, é uma característica intrínseca da tecnologia de 125 kHz utilizada. Essa limitação pode ser inconveniente em algumas aplicações onde seria desejável uma maior distância de

leitura, como em cancelas de veículos ou portões de grande porte.

O consumo energético do sistema também merece atenção. Durante os testes de uma semana em operação contínua, observei que os componentes mantiveram temperatura normal, mas a necessidade de alimentação constante torna impraticável a operação por bateria para longos períodos. Isso limita a aplicação do sistema a locais com fornecimento constante de energia elétrica.

### 7.3 Trabalhos Futuros

Várias melhorias e expansões podem ser implementadas a partir deste trabalho. Uma evolução natural seria o desenvolvimento de um aplicativo móvel nativo, permitindo que administradores gerenciem o sistema diretamente de seus smartphones. Também seria interessante implementar autenticação de dois fatores, combinando o cartão RFID com uma senha ou biometria.

Do ponto de vista técnico, uma abordagem mais sofisticada poderia envolver o desenvolvimento de um sistema que emule completamente o comportamento de uma tag RFID, permitindo que o Arduino gere sinais de radiofrequência para simular diferentes tags para a controladora. Isso eliminaria a necessidade de conexão física com a antena e permitiria um controle ainda mais flexível do sistema. Outra possibilidade seria investigar métodos de engenharia reversa mais avançados para acessar diretamente o firmware da controladora, caso ferramentas apropriadas se tornem disponíveis no mercado brasileiro.

### 7.4 Considerações Finais

O projeto demonstrou com sucesso que a modernização de sistemas legados de controle de acesso é não apenas viável, mas também economicamente vantajosa quando comparada à substituição completa do sistema. O custo total dos componentes utilizados

(Arduino Uno, ESP8266, RDM6300 e componentes auxiliares) foi inferior a R\$ 230,00, representando uma fração do custo de uma nova controladora com recursos similares de conectividade.

A solução implementada utilizou uma abordagem de interceptação paralela, conectando a antena do leitor RDM6300 em paralelo com a antena original da controladora SA-202. Embora essa técnica tenha funcionado perfeitamente para o propósito deste trabalho, é importante destacar que existem métodos mais robustos que poderiam ser explorados, como a emulação completa do protocolo RFID ou a integração direta com o firmware da controladora, caso fosse possível acessá-lo.

A experiência adquirida durante o desenvolvimento revelou a importância da flexibilidade na abordagem de problemas técnicos. A mudança de estratégia, da tentativa de integração direta para a interceptação paralela, demonstrou que soluções criativas podem superar limitações aparentemente intransponíveis. Essa lição é particularmente relevante no contexto da Internet das Coisas, onde frequentemente é necessário integrar dispositivos de diferentes gerações e tecnologias.

Por fim, este trabalho contribui para a área de sistemas embarcados e IoT ao demonstrar uma metodologia prática para modernização de sistemas legados, podendo servir como referência para projetos similares. A documentação detalhada do processo, incluindo os desafios enfrentados e as soluções adotadas, oferece um guia valioso para outros desenvolvedores que enfrentem problemas semelhantes.

---

# Referências Bibliográficas

---

ABNT (2018), ‘Nbr 6023:2018 - informação e documentação - referências - elaboração’.

Acesso em: 30 maio 2024.

**URL:** <https://sibi.furg.br/files/3524.pdf>

ABNT (2023), ‘Nbr 10520:2023 - informação e documentação - citações em documentos - apresentação’. Acesso em: 11 jul. 2024.

**URL:** [https://coc.fiocruz.br/wp-content/uploads/2024/03/Abnt\\_nbr\\_10520\\_2023.pdf](https://coc.fiocruz.br/wp-content/uploads/2024/03/Abnt_nbr_10520_2023.pdf)

ABNT (2024), ‘Nbr 14724:2024 - informação e documentação - trabalhos acadêmicos - apresentação’. Acesso em: 10 nov. 2024.

**URL:** [https://tpp-uff.com.br/wp-content/uploads/2025/02/ABNT\\_NBR\\_14724\\_2024-1.pdf](https://tpp-uff.com.br/wp-content/uploads/2025/02/ABNT_NBR_14724_2024-1.pdf)

Dennison, Avery (s.d.), ‘Noções básicas sobre rfid’, White paper introdutório. Acesso em: 23 maio 2025.

**URL:** <https://labelaverydennison.com/content/dam/averydennison/lpm-responsive/south-america/brazil/documents/resources/guides/entendendo-rfid.pdf>

Embarcados (2023), ‘Arduino - comunicação serial’, Portal Embarcados. Acesso em: 18 jul. 2025.

**URL:** <https://embarcados.com.br/arduino-comunicacao-serial/>

Espressif Systems (2020), Esp8266ex datasheet, Technical reference, Espressif Systems. Acesso em: 16 maio 2025.

**URL:** [https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf)

Fredrich, Todd (2012), ‘Restful service best practices’. Acesso em: 18 jul. 2025.

**URL:** [https://www.restapitutorial.com/media/RESTful\\_Best\\_Practices-v1\\_1.pdf](https://www.restapitutorial.com/media/RESTful_Best_Practices-v1_1.pdf)

Google Firebase (2024), ‘Firebase realtime database rest api’, Firebase Documentation.

Acesso em: 28 dez. 2024.

**URL:** <https://firebase.google.com/docs/reference/rest/database>

Intelbras (2024), *Manual do Usuário - Controlador de Acesso DigiProx SA-202*, Intelbras

S.A. Acesso em: 20 nov. 2024.

**URL:** [https://backend.intelbras.com/sites/default/files/2025-07/Manual\\_Digiprox\\_SA\\_202\\_01-25\\_site.pdf](https://backend.intelbras.com/sites/default/files/2025-07/Manual_Digiprox_SA_202_01-25_site.pdf)

MakerHero (2023), ‘Esp8266: Guia completo’, Blog MakerHero. Acesso em: 18 jul. 2025.

**URL:** <https://www.makerhero.com/blog/esp8266-guia-completo/>

MDN Web Docs (2024a), ‘Https’, Mozilla Developer Network. Acesso em: 18 jul. 2025.

**URL:** <https://developer.mozilla.org/pt-BR/docs/Glossary/HTTPS>

MDN Web Docs (2024b), ‘Métodos de requisição http’, Mozilla Developer Network. Acesso

em: 18 jul. 2025.

**URL:** <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>

MDN Web Docs (2024c), ‘Trabalhando com json’, Mozilla Developer Network. Acesso em:

18 jul. 2025.

**URL:** <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/JSON>

NIC.br (2024), ‘Serviço de tempo brasileiro - ntp.br’. Acesso em: 22 ago. 2025.

**URL:** <https://ntp.br/>

Observatório Nacional (2024), 'Hora legal brasileira'. Acesso em: 22 ago. 2025.

**URL:** <https://pcdsh01.on.br/>

Santos, B. P., L. A. Silva, C. Celes, J. B. Borges, B. S. Peres, M. A. Vieira, L. F. Vieira, O. Goussevskaia & A. Loureiro (2016), Internet das coisas: da teoria à prática, *em 'Minicursos SBRC 2016'*, SBC. Acesso em: 02 jul. 2025.

**URL:** <https://homepages.dcc.ufmg.br/mmvieira/cc/papers/internet-das-coisas.pdf>

Seeed Studio (2016), '125khz rfid module - uart', Seeed Studio Wiki. Acesso em: 07 jun. 2025.

**URL:** <https://wiki.seeedstudio.com/>

Vieira, M. A. F., C. N. Coelho, D. C. Silva Júnior & J. M. Mata (2007), 'Survey on wireless sensor network devices', *Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação* 12(24), 182–202. Acesso em: 05 jul. 2025.

**URL:** <https://periodicos.ufsc.br/index.php/eb/article/view/1518-2924.2007v12n24p182>

---

# Apêndice A

---

## Informações Complementares

---

### A.1 Declaração do Orientador Para a Biblioteca

Eu, *Prof. Dr. Fabrício Carvalho*, na qualidade de orientador(a) do aluno (a) *Nícolas Gabriel Meneses De Jesus* do Curso de Engenharia de Computação declaro para os devidos fins que o trabalho intitulado **Sistema de Controle de Acesso com RFID Integrado ao Firebase: Uma Abordagem de Interceptação Paralela para Modernização de Sistemas Legados** RESPEITA TODOS OS DIREITOS AUTORAIS, estando isento de plágio, cópias ilegais ou quaisquer ofensas aos direitos de outros autores, em conformidade com o que rege a Lei nº 9.610/98. Declaro, ainda, que o trabalho em questão passou por uma banca de avaliação, sendo realizadas as devidas correções e, estando assim, apto a ser disponibilizado em texto integral, na Biblioteca Digital da UFMT ou em qualquer outro sistema de automação e gestão de acervos, utilizado pela Instituição, para consulta e acesso livre de modo on-line.