

Centro Federal de Educação Tecnológica de Minas Gerais  
*Campus* Divinópolis  
Graduação em Engenharia Mecatrônica

Lucas Alves Bueno

Projeto de um Dispositivo de Identificação de Equipamentos em Ambientes Industriais



Divinópolis  
2021

Lucas Alves Bueno

Projeto de um Dispositivo de Identificação de Equipamentos em Ambientes Industriais

Monografia de Trabalho de Conclusão de Curso apresentada ao Colegiado de Graduação em Engenharia Mecatrônica como parte dos requisitos exigidos para a obtenção do título de Engenheiro Mecatrônico.  
Áreas de integração: Computação, Eletrônica.

Orientador: Prof. Me. Marlon Henrique Teixeira

Divinópolis  
2021

Lucas Alves Bueno

Projeto de um Dispositivo de Identificação de Equipamentos em Ambientes Industriais

Monografia de Trabalho de Conclusão de Curso apresentada ao Colegiado de Graduação em Engenharia Mecatrônica como parte dos requisitos exigidos para a obtenção do título de Engenheiro Mecatrônico.  
Áreas de integração: Computação, Eletrônica.

Comissão Avaliadora:

Prof. Me. Marlon Henrique Teixeira  
CEFET-MG / *Campus* Divinópolis

Prof. Me. Raulivan Rodrigo da Silva  
CEFET-MG / *Campus* Divinópolis

Prof. Dr. Nelson de Figueiredo Barroso  
CEFET-MG / *Campus* Divinópolis

Divinópolis  
2021

A minha família e amigos que, nos momentos de alegria e dificuldade, forneceram apoio incondicional.

# Agradecimentos

Agradeço,

- aos meus pais, Geraldo e Regina, por todos os ensinamentos e por fornecerem os meios necessários para meu desenvolvimento nos estudos, mantendo um lar aconchegante e baseado no respeito e confiança.
- a minha esposa, Cássia, por estar ao meu lado e acreditar na minha capacidade durante todo o período da minha formação, tornando a trajetória mais leve com seu apoio e amor incondicional.
- aos meus amigos e colegas de curso, em especial a Turma 8 e a turma de 2016, por todos os pequenos momentos de alegria diária e por todo conhecimento compartilhado, tornando a graduação um processo menos árduo.
- ao meu orientador Prof. Me. Marlon Henrique Teixeira, por todo suporte fornecido durante a execução deste trabalho.
- aos professores Nelson de Figueiredo e Raulivan Rodrigo, que, na condição de membros da comissão avaliadora, forneceram contribuições importantes para a conclusão deste trabalho.
- por fim, a todos professores e colaboradores que mantêm o funcionamento do CEFET-MG Campus Divinópolis, provendo a infraestrutura necessária para meu estudo e formação.

*"Nossa existência deforma o universo.  
Isso é responsabilidade."*

Neil Gaiman

# Resumo

A identificação e catalogação das informações a respeito de equipamentos em ambiente industrial é um processo lento e exaustivo quando feito de forma totalmente manual. Apesar de existirem soluções para a automação completa deste processo, em diversos setores da indústria tal procedimento não é viável, por fatores como: espaço físico reduzido; fluxo não regular de equipamentos; orçamento reduzido. Portanto, para estes casos, faz-se necessário o desenvolvimento de uma solução de automação parcial, através de um dispositivo operado manualmente. Partindo desta premissa, este trabalho tem por objetivo o desenvolvimento de um dispositivo de identificação de equipamentos em ambiente industrial via comunicação NFC com integração à uma aplicação web via wi-fi. Serão apresentadas as etapas para o desenvolvimento do projeto, assim como os resultados obtidos e o planejamento para continuidade.

Palavras-chave: Identificação de Equipamentos, Dispositivo Portátil, Aplicação Web.



# Abstract

The identification and cataloging of information about equipment in an industrial environment is a slow and exhausting process when done completely manually. Although there are solutions for the complete automation of this process, in several sectors of the industry this procedure is not feasible, due to factors such as: reduced physical space; non-regular flow of equipment; reduced budget. Therefore, for these cases, it is necessary to develop a partial automation solution, through a manually operated device. Based on this premise, this work aims to develop an equipment identification device in an industrial environment via NFC communication with integration to a web application via wi-fi. The stages for the development of the project will be presented, as well as the results obtained and the planning for continuity.

Key-words: Equipment Identification, Portable Device, Web Application.



# Sumário

<b>Lista de Figuras</b>	<b>xii</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>Lista de Acrônimos e Notação</b>	<b>xiv</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Definição do problema . . . . .	1
1.2 Motivação . . . . .	2
1.3 Objetivos do trabalho . . . . .	2
1.3.1 Objetivos Gerais . . . . .	2
1.3.2 Objetivos Específicos . . . . .	2
1.4 Organização do texto . . . . .	3
<b>2 Fundamentos</b>	<b>4</b>
2.1 Revisão Bibliográfica . . . . .	4
2.1.1 Microcontroladores . . . . .	4
2.1.2 Comunicação Sem Fio . . . . .	5
2.1.3 Aplicações Web . . . . .	6
2.2 Estado da Arte . . . . .	7
2.3 Fundamentação Teórica . . . . .	8
2.3.1 ESP32 . . . . .	8
2.3.2 NFC . . . . .	9
2.3.3 HTML . . . . .	10
2.3.4 PHP . . . . .	10
2.3.5 SQL . . . . .	11
2.3.6 CSS . . . . .	11
<b>3 Desenvolvimento e Resultados</b>	<b>13</b>
3.1 Definição das Funcionalidades . . . . .	13
3.2 Materiais e Métodos . . . . .	14
3.2.1 Eletrônica . . . . .	14
3.2.2 Softwares . . . . .	19
3.2.3 Fluxo das Informações . . . . .	20
3.2.4 Banco de Dados . . . . .	21
3.2.5 O Formato NDEF . . . . .	21

---

3.3	A Aplicação Web . . . . .	23
3.3.1	Desenvolvimento da Aplicação . . . . .	24
3.4	O Dispositivo de Identificação . . . . .	26
3.4.1	Definição da Interface do Dispositivo . . . . .	26
3.4.2	Desenvolvimento do Protótipo . . . . .	26
3.4.3	Projeto da Carcaça . . . . .	31
3.4.4	Procedimento de Uso . . . . .	35
3.4.5	Rotina de Leitura . . . . .	36
3.4.6	Rotina de Escrita . . . . .	38
3.5	Orçamento . . . . .	40
<b>4</b>	<b>Considerações Finais</b>	<b>41</b>
4.1	Conclusões . . . . .	41
4.2	Propostas de Continuidade . . . . .	42
<b>A</b>	<b>Códigos da Aplicação Web</b>	<b>43</b>
A.1	ESP-SQL.php . . . . .	43
A.2	index.php . . . . .	44
A.3	Exportar.php . . . . .	46
<b>B</b>	<b>Código do Microcontrolador</b>	<b>48</b>
	<b>Referências</b>	<b>55</b>

# Lista de Figuras

2.1	PIC 16F15244. Fonte: (MICROCHIP, 2021)	5
2.2	ESP32. Fonte: (ROBOCORE, 2020)	5
2.3	Pinagem ESP32. Fonte: (FELIPEFLOP, 2018)	8
3.1	ESP32. Fonte: (ROBOCORE, 2020)	14
3.2	Módulo Leitor NFC PN532. Fonte: (FELIPEFLOP, 2021a)	15
3.3	Etiqueta Adesiva NFC. Fonte: (FELIPEFLOP, 2021b)	15
3.4	Display LCD 16x02. Fonte: (FELIPEFLOP, 2021c)	16
3.5	Bateria Li-Ion 18650. Fonte: (FELIPEFLOP, 2021d)	17
3.6	Curva de Descarga. Fonte: (TENERGY, 2009)	18
3.7	Conversor boost ajustável. Fonte: (FELIPEFLOP, 2021e)	18
3.8	Módulo carregador de bateria. Fonte: (FELIPEFLOP, 2021f)	19
3.9	Fluxo de dados. Fonte: Elaborado pelo autor	20
3.10	Estrutura do banco de dados. Fonte: Elaborado pelo autor	21
3.11	Estrutura do formato NDEF. Fonte: (O'REILLYMEDIA, 2022)	22
3.12	Exemplo de mensagem no formato NDEF. Fonte: Elaborado pelo autor	23
3.13	Interface da Aplicação Web. Fonte: Elaborado pelo autor	24
3.14	Interface da Aplicação Web. Fonte: Elaborado pelo autor	25
3.15	Tabela exportada em formato .xls. Fonte: Elaborado pelo autor	25
3.16	Design da interface do Dispositivo. Fonte: Elaborado pelo autor	26
3.17	Montagem do protótipo. Fonte: Elaborado pelo autor	30
3.18	Esquemático do dispositivo. Fonte: Elaborado pelo autor	30
3.19	Desenho interno das duas metades da carcaça. Fonte: Elaborado pelo autor	31
3.20	Desenho desenho da carcaça. Fonte: Elaborado pelo autor	32
3.21	Desenho desenho da carcaça. Fonte: Elaborado pelo autor	32
3.22	Desenho técnico da frente da carcaça. Fonte: Elaborado pelo autor	33
3.23	Desenho técnico do fundo da carcaça. Fonte: Elaborado pelo autor	34
3.24	Fluxograma de funcionamento do dispositivo. Fonte: Elaborado pelo autor	35
3.25	Telas de conexão com wifi. Fonte: Elaborado pelo autor	36
3.26	Telas de seleção de modo. Fonte: Elaborado pelo autor	36
3.27	Telas de confirmação de envio. Fonte: Elaborado pelo autor	37
3.28	Telas de confirmação de nova leitura. Fonte: Elaborado pelo autor	37
3.29	Tela final. Fonte: Elaborado pelo autor	37
3.30	Telas de conexão com wifi. Fonte: Elaborado pelo autor	38
3.31	Telas de seleção de modo. Fonte: Elaborado pelo autor	38
3.32	Tela de gravação. Fonte: Elaborado pelo autor	39

3.33 Tela de falha na escrita. Fonte: Elaborado pelo autor . . . . .	39
3.34 Telas de confirmação de nova escrita. Fonte: Elaborado pelo autor . . . . .	39
3.35 Tela final. Fonte: Elaborado pelo autor . . . . .	40

# Lista de Tabelas

2.1	Características da Comunicação NFC . . . . .	9
3.1	Características Elétricas . . . . .	17
3.2	Relação de pinos utilizados no ESP32 . . . . .	27
3.3	Orçamento estimado para os gastos do projeto . . . . .	40

## Lista de Acrônimos e Notação

CSS	Folhas de Estilo em Cascata, do inglês <i>Cascading Style Sheets</i>
GPIO	Entrada/Saída de Uso Geral, do inglês <i>General Purpose Input/Output</i>
HTML	Linguagem de Marcação de Hipertexto, do inglês <i>Hypertext Markup Language</i>
PHP	PHP Pré-processador de Hipertexto, do inglês <i>PHP: Hypertext Preprocessor</i>
I2C	Circuito Inter-integrado, do inglês <i>Inter-Integrated Circuit</i>
LCD	Tela de Cristal Líquido, do inglês <i>Liquid Cristal Display</i>
LED	Diodo Emissor de Luz, do inglês <i>Light-Emitting Diode</i>
NFC	Comunicação por Campo de Proximidade, do inglês <i>Near-Field Communication</i>
NDEF	Formato de troca de dados NFC, do inglês <i>NFC Data Exchange Format</i>
PIC	Controlador de Interface Programável, do inglês <i>Programmable Interface Controller</i>
PWM	Modulação por Largura de Pulso, do inglês <i>Pulse Width Modulation</i>
RFID	Identificação por Rádio Frequência, do inglês <i>Radio-Frequency Identification</i>
SoC	Sistema em um Chip, do inglês <i>System on a Chip</i>
SQL	Linguagem de Consulta Estruturada, do inglês <i>Linguagem de Consulta Estruturada</i>

## Introdução

A identificação e coleta de dados referentes à dispositivos diversos, assim como a manutenção de um sistema de registro de tais informações é uma necessidade presente em vários setores da indústria, sobretudo no controle de estoques. De acordo com BORGES; CAMPOS; BORGES (2010) a gestão de estoques é uma das atividades mais importantes de uma manufatura, gerando grande impacto financeiro através do aumento da eficácia e eficiência das operações da Organização.

No contexto dos setores de manutenção, a gestão de estoque também tem grande importância. "A falta de peças de reposição em estoque aumenta o tempo de inatividade dos equipamentos em um sistema de produção, adia os procedimentos de manutenção e gera custos desnecessários à empresa."(CAMPOS; SIMON, 2019).

Atualmente, com o avanço da Indústria 4.0, existem diversas soluções comerciais e estudos envolvendo identificação de equipamentos em linhas de produção. Isso se deve ao fato deste modelo de indústria integrar recursos como a Internet das Coisas, a computação em nuvem e o aprendizado de máquina, que permitem grande eficiência na obtenção e compartilhamento de dados.

Tais soluções são, em sua maioria, planejadas para grandes cadeias fixas de produção, permitindo pouca ou nenhuma mobilidade. Entretanto, em setores da indústria nos quais os produtos são movidos e organizados manualmente, faz-se necessário um método de identificação de equipamentos que seja portátil e de operação manual simples.

Neste contexto, este trabalho tem como proposta o projeto de um dispositivo portátil de identificação de equipamentos via NFC (Near Field Communication) e comunicação via wi-fi com uma aplicação *web*.

### 1.1 Definição do problema

A identificação e catalogação das informações a respeito de equipamentos em ambiente industrial é um processo lento e exaustivo quando feito de forma totalmente manual. Este

processo pode ser automatizado por completo, com o uso de esteiras ou outros métodos de deslocamento dos equipamentos, sensores fixos para a identificação e um sistema computacional para organização dos dados. Porém, em diversos setores da indústria a automação completa não é viável, por fatores como: espaço físico reduzido; fluxo não regular de equipamentos; orçamento reduzido. Portanto, o tempo e esforços excessivos necessários no processo de identificação, combinados à dificuldade em implementar uma solução de automação completa, é o problema a ser solucionado por este projeto.

## **1.2 Motivação**

A motivação para o desenvolvimento deste projeto surgiu através da experiência do autor na condição de estagiário no setor de manutenção de uma empresa do ramo de telecomunicações. No setor em questão são recebidos, em grandes quantidades e oriundos de diversas cidades, dispositivos como roteadores, fontes, amplificadores de sinal, entre outros. Estes precisam então ser catalogados por tipo e cidade de origem. Então, estes dados devem ser repassados manualmente para uma planilha no computador. A grande quantidade de equipamentos recebidos torna este processo demorado e exaustivo.

## **1.3 Objetivos do trabalho**

São objetivos do trabalho a ser desenvolvidos:

### **1.3.1 Objetivos Gerais**

O objetivo deste trabalho é desenvolver um dispositivo de identificação de equipamentos em ambiente industrial via NFC com integração à uma aplicação web via wi-fi.

### **1.3.2 Objetivos Específicos**

- Definir as funcionalidades do dispositivo;
- Definir e especificar os materiais e softwares necessários;
- Projetar o módulo eletrônico;
- Definir a forma de interação com o dispositivo;
- Definir a interface da aplicação web;
- Definir o procedimento de uso;
- Projetar a carcaça do dispositivo;

- Programar o código do microcontrolador;
- Desenvolver a aplicação web;
- Realizar testes de funcionamento;

## **1.4 Organização do texto**

O presente texto é dividido em quatro capítulos. No corrente capítulo são apresentados a definição do problema e a motivação para o desenvolvimento do trabalho. Além disso, são listados os objetivos gerais e específicos do projeto.

No segundo capítulo é apresentada a fundamentação do trabalho. Constam neste trecho a revisão bibliográfica, o estado da arte e a fundamentação teórica.

Em seguida, no terceiro capítulo, é apresentada a metodologia. Essa parte do trabalho é dedicada a justificar as escolhas de projeto, além de apresentar cada etapa do desenvolvimento do mesmo.

## Fundamentos

O presente capítulo apresenta, inicialmente, uma revisão bibliográfica sucinta sobre os temas abordados no trabalho, expondo a evolução tecnológica dos componentes utilizados. Então, será discutido o estado atual da arte. Por fim, os conceitos necessários para o desenvolvimento do projeto serão apresentados.

### 2.1 Revisão Bibliográfica

#### 2.1.1 Microcontroladores

Atualmente, os microcontroladores estão presentes em quase todos os dispositivos eletrônicos controlados digitalmente, como em máquinas de lavar, fornos de micro-ondas, televisores e nos veículos (LIMA; VILLACA, 2012). Porém, para atingir o nível de desenvolvimento atual, foram necessários diversos avanços tecnológicos ao longo dos anos, desde os primeiros estudos a respeito de semicondutores.

Em 1969 dá-se início ao desenvolvimento dos primeiros microprocessadores pela Intel Corporation, motivada pela demanda da empresa japonesa BUSICOM, que trabalhava no desenvolvimento de uma calculadora. Marcian Hoff, da Intel, propôs o desenvolvimento de um dispositivo que funcionasse de acordo com um programa, não sendo limitado apenas ao projeto da calculadora. Dessa forma, em 1971, a Intel lançou seu primeiro processador de 4 bits, o 4004, capaz de processar operações a 6Khz (SOUZA, 2006).

Com o avanço desenvolvimento dos microprocessadores surgiram os primeiros microcontroladores. Estes se diferem dos anteriores por possuírem em um único dispositivo, além da unidade de processamento, memória e componentes para envio e recebimento de dados (SOUZA, 2006). Nesta categoria destaca-se a família de microcontroladores PIC (*Peripheral Interface Controler*), da empresa Microchip, que possuem uma grande diversidade de recursos, capacidades de processamento, custo e flexibilidade de aplicações (CORTELETTI, 2006). A Figura 2.1 mostra um PIC 16F15244.

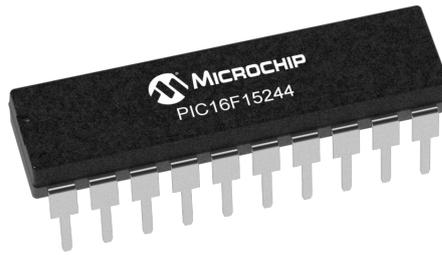


Figura 2.1: PIC 16F15244. Fonte: (MICROCHIP, 2021)

Produzido pela empresa Espressif Systems, o ESP32 é, de acordo com KURNIAWAN(2019), um chip de baixo custo que integra soluções Wi-Fi e Bluetooth em um único dispositivo. Ainda segundo KURNIAWAN(2019), estas características fazem do ESP32 uma boa opção para o desenvolvimento de aplicações de Internet das Coisas (IoT). A Figura 2.2 mostra um ESP32.

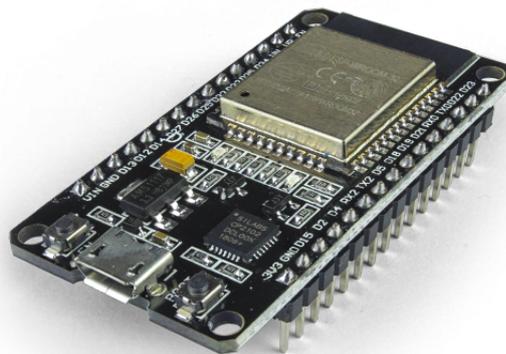


Figura 2.2: ESP32. Fonte: (ROBOCORE, 2020)

### 2.1.2 Comunicação Sem Fio

Os princípios da tecnologia de transmissão sem fio podem ser determinados por uma série de descobertas que ocorreram devido à grande busca do conhecimento e desenvolvimento no decorrer dos séculos (LUGLI; SOBRINHO, 2012).

No início do século XX, de acordo com LUGLI; SOBRINHO(2012), o inglês Jonh Ambrose Fleming e o norte-americano Lee De Forest encontraram pela primeira vez uma forma de modular e amplificar sinais sem fio para o envio de transmissão de voz.

Em 1971, ainda segundo LUGLI; SOBRINHO(2012), um grupo de pesquisadores da Universidade do Havaí criou o pacote First-switched de rede de comunicação de rádio, intitulado “Alohanet”. Esta foi a primeira rede local sem fio, composta por sete computadores.

Já na década de 1990, surge o grupo de trabalho 802.11, responsável por desenvolver uma norma padrão para comunicação sem fio entre computadores. O protocolo 802.11 seria publicado em 1997 pelo órgão regulador IEEE(Institute of Electrical Electronics Engineers). Em 1999, surge a associação de empresas Wi-Fi Alliance, utilizando o padrão 802.11 e licenciando produtos baseados na tecnologia sem fio.

Outra categoria de comunicação sem fio amplamente utilizada é a das comunicações em curtas distâncias. Nesta categoria, destaca-se inicialmente a tecnologia de identificação por rádio frequência RFID(Radio-Frequency IDentification). A origem primitiva do RFID data da Segunda Guerra Mundial. Na ocasião, alemães descobriram que podiam diferenciar aviões aliados e inimigos no radar se os pilotos da sua força aérea movimentassem seus aviões ao retornar a base. Esta manobra alterava o sinal de rádio refletido pelos aviões de volta ao radar. Essencialmente, este foi o primeiro sistema passivo de comunicação RFID (ROBERTI, 2005).

O RFID como conhecemos atualmente surgiu mais tarde. De acordo com ROBERTI (2005), em 1973, o empresário californiano Charles Walton recebeu uma patente para um transponder passivo utilizado para destrancar uma porta sem utilizar uma chave. Walton licenciou a tecnologia para a fabricante de tranças Schalge e para outras empresas, disseminando a tecnologia.

De acordo com a *Near Field Communication Organization* (2017), a Comunicação por campo de proximidade NFC (Near Field Communication) é uma subcategoria da tecnologia RFID, porém com amplitude de comunicação menor, para fins de segurança. Em 2004, Nokia, Sony e Philips formaram o *NFC Forum*, grupo dedicado a promover a segurança, facilidade de uso e popularidade do NFC. O grupo também estabelece os padrões que permitem que o NFC opere entre diferentes dispositivos.

As primeiras especificações para a produção de tags NFC, ainda de acordo com a *Near Field Communication Organization*, foram estabelecidas pelo *NFC Forum* em 2006. As tags NFC são pequenos objetos, como adesivos, que contém informação que pode ser lida por dispositivos compatíveis com a tecnologia. Algumas tags permitem também que sejam escritas novas informações através de um dispositivo ativo.

### 2.1.3 Aplicações Web

De acordo com URYUTIN(2018), a evolução das aplicações para a web pode ser dividida em sete etapas, sendo elas:

- Início da década de 1990: surgimento das primeiras páginas web estáticas, contendo texto e, posteriormente, imagens e arquivos de áudio e vídeo.
- 1995: é apresentada a linguagem Java Script, a qual tornou a internet mais rápida e adicionou elementos dinâmicos para as páginas da web.

- 1996: é introduzido o Macromedia Flash, permitindo a inclusão de animações nas páginas da web. Nesta época houve o crescimento dos jogos interativos.
- 1999: o conceito de aplicação web aparece na linguagem Java.
- 2005: páginas estáticas dão lugar às páginas dinâmicas. É introduzido o Ajax, permitindo usuários a trabalharem em aplicações web de forma mais rápida e melhor.
- 2014: é criado o HTML5, aprimorando os padrões já existentes do HTML. Novos tipos de multimídia são suportados. Torna-se possível criar aplicações web independentes de browsers ou outras plataformas.
- 2015: Alex Russel anuncia as aplicações web progressivas (PWA), as quais combinam as características oferecidas pelos browsers mais modernos em conjunto com as funcionalidades das aplicações móveis.

## 2.2 Estado da Arte

Estudos a respeito de identificação utilizando a tecnologia NFC ainda são escassos, principalmente em comparação com estudos semelhantes que utilizam a tecnologia RFID, mais antiga e bem difundida. Além disso, se tratando de aplicações específicas para ambiente industrial, os estudos são ainda mais raros. Porém, segundo SALLINEN; STROMMER; YLISAUKKO-OJA (2008), “comparado a outras formas de comunicação de curta distância, como Bluetooth ULP (Ultra Low Power) ou Zigbee, o NFC provê vantagens no ambiente industrial”, como, por exemplo, consumo de energia e tempo requerido para conexão mais baixos.

No estudo em questão, os autores apresentam um cenário de uso do NFC em sistemas de produção como um método simples de comunicação. A ideia proposta é utilizar uma interface NFC desenvolvida para permitir uma leitura flexível de sensores e máquinas no ambiente industrial.

Em 2013, RAMANATHAN; IMTIAZ apresentam um estudo a respeito do uso da tecnologia NFC para fornecer informações de localização de uma planta de produção. No estudo, conclui-se que a tecnologia tem grande potencial em aplicações industriais, pois o NFC pode operar em ambientes severos com o auxílio de tags robustas, que possuem encapsulamento mais resistente.

Fora do ambiente industrial, em 2014, CONSTANTINO; NERIS propõem o uso da tecnologia NFC para auxiliar deficientes visuais na identificação de livros. No estudo é proposto o uso de uma aplicação android em conjunto com etiquetas NFC nos livros.

Já em 2015, FERNANDES propõe um sistema de identificações de objetos e pessoas para bancadas laboratoriais. No trabalho em questão é proposto o uso de leitores da tecnologia

NFC em cada bancada laboratorial, tornando possível que os usuários se identifiquem utilizando smartphones compatíveis.

Por fim, em 2018, MEDEIROS; SARTORI apresentam o protótipo de um aplicativo para identificações de animais utilizando NFC. No trabalho é proposta a aplicação de microchips compatíveis com NFC em animais domésticos, contendo a identificação dos mesmos. Em conjunto com uma aplicação mobile, MEDEIROS; SARTORI propõem que esta solução seja utilizada para facilitar a identificação de animais perdidos e abandonados.

## 2.3 Fundamentação Teórica

### 2.3.1 ESP32

ESP32 é uma série de microcontroladores ou *System on Chip* (SoC) com baixo consumo de energia e de baixo custo desenvolvidos pela Espressif. O dispositivo possui um processador *Xtensa dual-core* que opera em 160MHz ou 240MHz, 520 kBytes de memória SRAM, Wi-Fi 802.11 b/g/n e bluetooth 4.2. Além disso, conta com 36 sinais de GPIO, sinais PWM e conversores analógico-digital. A identificação de cada entrada e saída do dispositivo pode ser vista na Figura 2.3

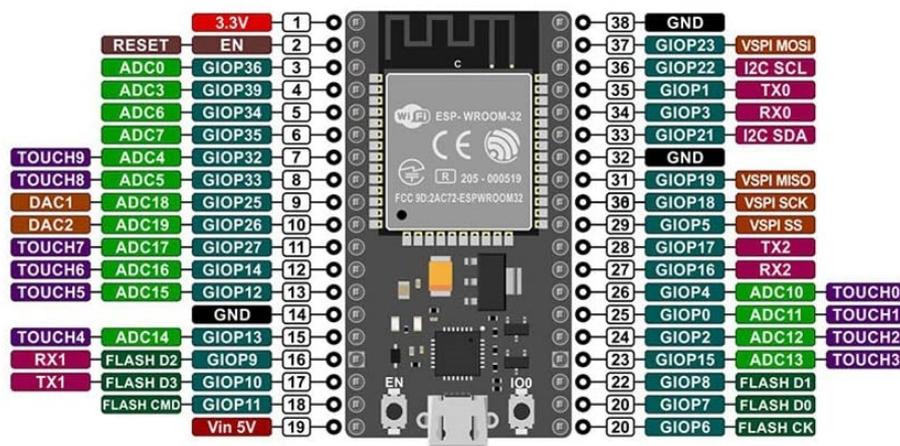


Figura 2.3: Pinagem ESP32. Fonte: (FELIPEFLOP, 2018)

Diversas plataformas são compatíveis como Ambiente de Desenvolvimento Integrado (IDE) para desenvolvimento de aplicações para o ESP32. Entre elas destacam-se a Lua RTOS, a opção oficial *Espressif IoT Development Framework* e, principalmente, a IDE do Arduino com núcleo ESP32 para Arduino.

### 2.3.2 NFC

A comunicação NFC é uma tecnologia ainda considerada emergente e desenvolvida sobre o padrão RFID, sendo, portanto, compatível com esta. Permite um link de comunicação sem contato e sem fio entre dispositivos próximos (menos de 4cm de distância) a uma taxa de dados máxima de 424Kbps. A comunicação pode ser ativa, passiva ou ambos (MOTLAGH, 2012). A Tabela 2.1 mostra as características técnicas da comunicação NFC.

Tabela 2.1: Características da Comunicação NFC

Característica	NFC
Tipo de Rede	ponto-a-ponto
Amplitude	até 20cm
Frequência	13,56MHz
Taxa de <i>bits</i>	424kbs
Consumo de Corrente	< 15mA(leitura)

De acordo com DESAI; SHAJAN(2012), os dispositivos NFC podem operar em três diferentes modos, sendo eles:

- **Leitor/escritor:** o dispositivo NFC se comporta como um leitor de tags NFC, como os cartões inteligentes sem contato e as etiquetas RFID.
- **Ponto a ponto:** permite a troca de informações entre dois dispositivos NFC.
- **Emulação de cartão:** permite que um dispositivo NFC funcione como um cartão inteligente sem contato.

Em cada um dos modos supracitados, de acordo com FILHO; CAVALCANTE (2010) os dispositivos NFC podem ter comportamento ativo ou passivo, com as seguintes características:

**Comportamento ativo** - informações codificadas utilizando código Miller e 100% de modulação. Iniciador e alvo se comunicam gerando alternadamente seus campos magnéticos.

**Comportamento passivo** - codificação Manchester com taxa de modulação de 10% e transmissão até 424Kbps. O iniciador provê um campo magnético portador, o qual induz corrente na antena do alvo, energizando-o, e este por sua vez responde modulando o campo.

### 2.3.3 HTML

O HTML (*HyperText Markup Language*) é uma linguagem de marcação, considerada o bloco de construção mais básico da web. Sua função é definir o significado e a estrutura do conteúdo web. A sigla pode ser traduzida como Linguagem de Marcação de HiperTexto, na qual o “Hipertexto” refere-se a um texto com blocos interconectados contendo palavras, imagens, sons, tabelas e outros elementos (XAVIER, 2019).

A marcação HTML inclui elementos especiais, como `<head>`, `<title>`, `<body>`, `<header>`, entre outros. O nome de um elemento dentro de uma *tag* é insensível a maiúsculas e minúsculas, isto é, pode ser escrito em maiúsculas, minúsculas ou ambas simultaneamente. Abaixo é mostrado um código HTML contendo a definição dos trechos correspondentes ao cabeçalho e ao conteúdo da página.

```
1 <html >
2   <head >
3     <!-- CABECALHO -->
4   </head >
5   <body >
6     <!-- CONTEUDO DA PAGINA -->
7   </body >
8 </html >
```

Listing 2.1: Exemplo de código HTML

### 2.3.4 PHP

De acordo com o *PHP Group* (2022), o PHP (*PHP: Hypertext Preprocessor*) é uma linguagem de código aberto amplamente utilizada, especialmente adequada para desenvolvimento web. A sintaxe é similar às linguagens C, Java e Perl e pode ser embutida em um código HTML. A função principal da linguagem é permitir a criação rápida de páginas web geradas de forma dinâmica.

Um grande diferencial da linguagem PHP quando comparada com C ou Perl é que as páginas em PHP contêm HTML embutido, evitando a necessidade de vários comandos para exibir HTML. O uso das instruções especiais de início e fim `<?php` e `?>` permite incluir trechos em PHP a qualquer momento. A seguir, é mostrado um exemplo de código HTML com uso do PHP embutido para exibir um texto na tela.

```
1 <html >
2 <head >
3 <title>Teste PHP</title >
4 </head >
5 <body >
6 <?php echo '<p>Exemplo de texto na tela</p>'; ?>
7 </body >
```

Listing 2.2: Exemplo de código PHP

### 2.3.5 SQL

A linguagem SQL (Structured Query Language) possui recursos para definir a estrutura de dados, modificar dados em um banco de dados e especificar restrições de segurança e integridade. A linguagem foi desenvolvida nos anos 70 na IBM, originalmente chamada de SEQUEL (Structured English Query Language). Atualmente, o padrão SQL é regulado pelo ANSI (American National Standards Institute)

De acordo com (FERNEDA, 2008) O SQL pode ser dividido em dois conjuntos principais de comandos:

**Data Definition Language (DDL):** fornece comandos para definição e modificação de esquemas de relação, remoção de relações e criação de índices. Os principais comandos que fazem parte da DDL são: CREATE, ALTER, DROP.

**Data Manipulation Language (DML):** inclui uma linguagem de consulta baseada na álgebra relacional e no cálculo relacional. Compreende também comandos para inserir, remover e modificar informações em um banco de dados. Os comandos básicos da DML são: SELECT, INSERT, UPDATE, DELETE.

A seguir é mostrado um exemplo de código de criação de tabela em linguagem SQL:

```
1 CREATE TABLE exemplo (  
2   int_exemplo INT AUTO_INCREMENT ,  
3   string_exemplo VARCHAR(10) ,  
4   decimal_exemplo DECIMAL(5,3) ,  
5   PRIMARY KEY (int_exemplo)  
6 );
```

Listing 2.3: Exemplo de código SQL

### 2.3.6 CSS

O CSS (Cascading Style Sheets ) é uma linguagem de estilo utilizada para descrever a apresentação de um documento escrito em HTML ou XML (Extensible Markup Language) padronizada pela W3C (World Wide Web Consortium) (G., 2019).

Esta linguagem é utilizada para definir o estilo de um documento web, definindo cores, fontes, espaçamento, etc. A sintaxe do CSS utiliza palavras em inglês para especificar diferentes propriedades do estilo de uma página web.

A instrução básica CSS consiste em um seletor e um bloco de declaração. Cada declaração contém uma propriedade e um valor, separados por dois pontos (:). Cada declaração é separada por ponto e vírgula (;).

Abaixo é mostrado um exemplo de CSS embutido em HTML, utilizado para definição de alinhamento, fonte e cores:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5 body {
6   background-color: lightblue;
7 }
8 h1 {
9   color: white;
10  text-align: center;
11 }
12 p {
13   font-family: verdana;
14   font-size: 20px;
15 }
16 </style>
17 </head>
18 <body>
19 <h1>TITULO EXEMPLO</h1>
20 <p>Paragrafo exemplo.</p>
21 </body>
22 </html>
```

Listing 2.4: Exemplo de código CSS

## Desenvolvimento e Resultados

Neste capítulo são apresentadas as etapas do desenvolvimento do projeto. Visando uma maior organização textual, realizou-se uma divisão por tópicos.

### **3.1 Definição das Funcionalidades**

A primeira etapa no desenvolvimento do projeto foi a definição clara de quais seriam as funcionalidades desejadas para o dispositivo. Para que sejam supridas as demandas explicitadas na Definição do Problema (Seção 1.1), é necessário que o dispositivo projetado possua os seguintes recursos:

- Realizar diversas identificações em sequência.
- Ser portátil.
- Ser capaz de registrar e organizar as informações.

Porém, apesar destas características já garantirem o funcionamento mínimo esperado, duas funcionalidades extras foram estabelecidas:

- Exibir informações de leitura em um display.
- Alterar os dados de identificação do equipamento.

Dessa forma, permite-se conferir se a informação exibida no display corresponde com o equipamento que está sendo identificado e que, caso contrário, seja possível corrigir ou atualizar a informação.

## 3.2 Materiais e Métodos

Para atender às características estabelecidas na seção anterior, primeiramente, foi definida a forma de identificação através do uso de etiquetas NFC em cada equipamento de interesse. As quais permitem realizar tanto leitura quanto gravação de informações.

Então, para atender ao requisito de portabilidade, foi estabelecido o uso de um sistema embarcado para o processamento das informações, ou seja, que pode operar sem estar conectado a um computador. Além disso, foi definido o uso de alimentação por baterias.

Por fim, para suprir o requisito do registro e organização das informações, foi determinado o desenvolvimento de uma aplicação web que recebe os dados do dispositivo via comunicação wi-fi.

A seções subsequentes serão dedicadas a especificar cada componente e recurso utilizado e justificar tais escolhas.

### 3.2.1 Eletrônica

#### Controlador

A definição dos componentes eletrônicos teve início pelo controlador, pois este deve suportar todas as funcionalidades desejadas para o dispositivo. Portanto, foi determinado o uso do ESP32, mostrado na Figura 3.1. Este dispositivo possui comunicação wi-fi nativa. Além disso, o ESP32 tem suporte a módulos de comunicação NFC e suas dimensões são apropriadas para o uso em um dispositivo portátil. Outras opções seriam o uso de um modelo de Arduino ou um microcontrolador da família PIC. Porém, ambos necessitariam de um periférico para realizar a comunicação wi-fi.

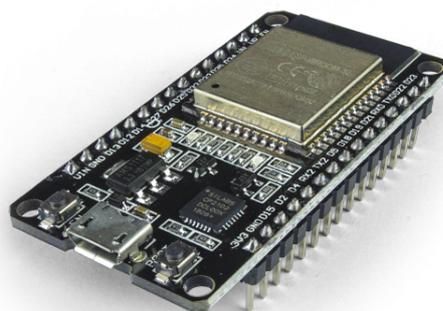


Figura 3.1: ESP32. Fonte: (ROBOCORE, 2020)

### Sensor NFC

O módulo de comunicação NFC escolhido foi o PN532, mostrado na Figura 3.2. Este módulo foi escolhido, principalmente, por ser compatível com o ESP32. Além disso, este componente suporta não somente a leitura, mas também a escrita de informações em etiquetas NFC. Não foram encontradas outras opções comerciais de módulos NFC compatíveis com o ESP32.



Figura 3.2: Módulo Leitor NFC PN532. Fonte: (FELIPEFLOP, 2021a)

### Etiqueta NFC

Como o objetivo do trabalho é utilizar o dispositivo para a identificação de equipamentos industriais diversos, é necessário que a etiqueta escolhida seja versátil, discreta e de fácil aplicação. Portanto, optou-se pelo uso de etiquetas adesivas, conforme as mostradas na Figura 3.3. Estas etiquetas possuem 25mm de diâmetro, pesam apenas 10 gramas e a aplicação é simples devido ao adesivo. Além disso, estas *tags* permitem a regravação de informações indefinidamente.



Figura 3.3: Etiqueta Adesiva NFC. Fonte: (FELIPEFLOP, 2021b)

## Display

A definição do display a ser utilizado no dispositivo foi baseada em três fatores:

- Capacidade de exibir todo o conteúdo necessário.
- Dimensões apropriadas para um dispositivo portátil.
- Compatibilidade com o protocolo de comunicação I2C.

Para atender a estas demandas, foi escolhido o display LCD 16x02 com módulo de comunicação I2C integrado, mostrado na Figura 3.4.



Figura 3.4: Display LCD 16x02. Fonte: (FELIPEFLOP, 2021c)

Este display possui duas linhas com espaço para 16 caracteres simultâneos em cada uma. Dessa forma, a primeira linha será utilizada para exibir o modo de operação do dispositivo (leitura ou escrita) e a segunda linha será dedicada a mostrar o nome do equipamento. As dimensões do display são 80 x 36 x 12mm e, portanto, são adequadas para o uso em um dispositivo portátil. A presença do módulo de comunicação I2C torna possível utilizar apenas duas portas do controlador para a comunicação.

## Alimentação

Devido à portabilidade do dispositivo, a alimentação deve ser feita por baterias. Para definir as especificações necessárias para a bateria foi considerado o consumo de corrente e a tensão de operação de cada componente, de acordo com as informações dos fabricantes. A Tabela 3.1 mostra as características de alimentação de cada componente e a corrente total consumida pelo dispositivo.

Tabela 3.1: Características Elétricas

Componente	Tensão de alimentação	Consumo de Corrente
ESP32	5V	260mA
PN532	5V	10 $\mu$ A
Display LCD	5V	120mA
Total	5V	380,01mA

Portanto, para atender à demanda de corrente do dispositivo, foi escolhida a bateria Li-Ion 18650, mostrada na Figura 3.5.



Figura 3.5: Bateria Li-Ion 18650. Fonte: (FELIPEFLOP, 2021d)

Esta bateria possui as seguintes características:

- Tensão Nominal: 3,7V.
- Capacidade nominal: 2200mAh.
- Tensão de corte: 3V.

A escolha por uma bateria com tensão nominal de 3,7V se deu pois baterias de Li-Ion de 5V não estão entre os valores comerciais disponíveis. A Figura 3.6 a curva de descarga da bateria. Neste gráfico, o eixo Y representa a tensão fornecida pela bateria e volts. O eixo X representa a capacidade percentual de descarga da bateria. As cores das curvas representam a corrente de descarga, conforme a relação abaixo:

- Verde: 2 amperes.
- Ciano: 1 ampere.
- Vermelho: 0.5 ampere.
- Preto: 0.2 ampere.

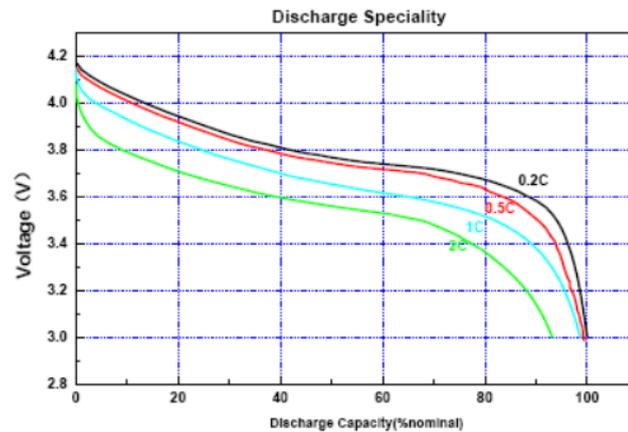


Figura 3.6: Curva de Descarga. Fonte: (TENERGY, 2009)

Nota-se no gráfico que o decaimento até a tensão de corte é suave até 80% da capacidade da bateria. Além disso, nota-se que para o consumo de corrente esperado para o dispositivo (380,01mA) será possível utilizar 100% da capacidade. A autonomia aproximada do dispositivo com esta bateria pode ser calculada como mostrado em (3.1).

$$\text{Autonomia} = \frac{2200mAh}{380,01mA} = 5,79h \approx 5h47min \quad (3.1)$$

Como a bateria possui tensão nominal de 3,7V, torna-se necessário o uso de um conversor *boost step-up* para elevar a tensão a 5V. Para isto será utilizado o modelo comercial de conversor mostrado na Figura 3.7.



Figura 3.7: Conversor boost ajustável. Fonte: (FELIPEFLOP, 2021e)

Este conversor suporta tensão de entrada entre 2V e 24V. Já a tensão de saída é ajustável e pode variar de 2V a 28V, com corrente de saída máxima de 1A. Portanto, o conversor atende às necessidades do projeto.

Para o carregamento da bateria será utilizado o módulo comercial TP4056, mostrado na Figura 3.8.



Figura 3.8: Módulo carregador de bateria. Fonte: (FELIPEFLOP, 2021f)

Este módulo de carregamento possui uma entrada mini USB e opera com tensão de entrada de 5V, o que torna possível a compatibilidade com diversos carregadores comerciais.

### 3.2.2 Softwares

Para o desenvolvimento do trabalho foram utilizados alguns *softwares*. As seções a seguir serão dedicadas a apresentar os *softwares* utilizados e justificar a escolha.

#### Arduino IDE

Para a programação do controlador, foi utilizada a IDE (*Integrated Development Environment*) do Arduino. Esta escolha foi baseada, principalmente, no fato da plataforma possuir compatibilidade com o ESP32, através de um pacote que pode ser instalado gratuitamente. As linguagens utilizadas para programação na IDE do Arduino são C e C++.

#### Visual Studio Code

Para a programação da aplicação web, foi utilizado o Visual Studio Code. Trata-se de um *software* de código-fonte desenvolvido pela Microsoft. A escolha desta plataforma se deu pelo suporte da mesma às principais linguagens de programação web, como HTML e PHP.

#### XAMPP

Para a criação e gerenciamento do servidor local foi utilizada a plataforma XAMPP. A escolha por esta plataforma foi baseada principalmente pelo suporte nativo ao servidor de banco de dados MySQL.

### 3.2.3 Fluxo das Informações

As seções a seguir serão dedicadas a definir como será realizado o fluxo das informações de interesse, desde a obtenção dos dados até a disponibilização destes para o usuário. Também será abordado o fluxo contrário, no qual o usuário registra informações em um equipamento.

A Figura 3.9 mostra um esquemático do fluxo de dados, tanto para obtenção quanto para o registro de informações.

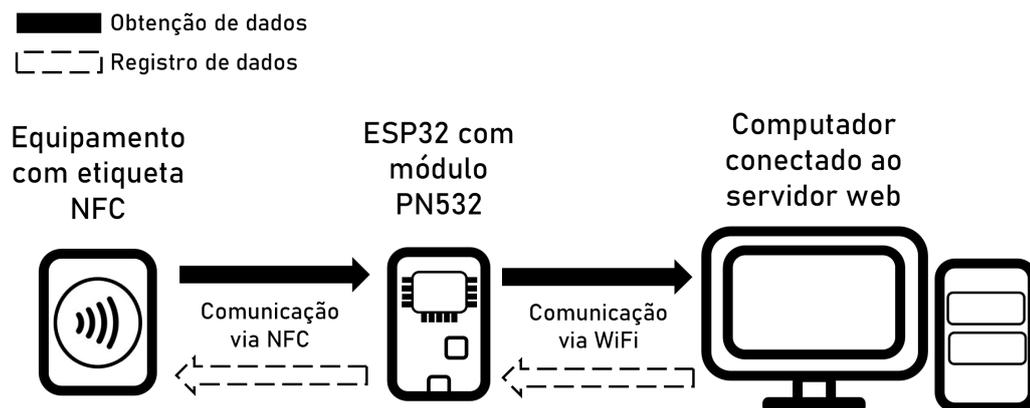


Figura 3.9: Fluxo de dados. Fonte: Elaborado pelo autor

#### Obtenção dos Dados

A obtenção dos dados referentes a cada equipamento de interesse é realizada através da leitura de uma *tag* adesiva NFC fixada no mesmo. Para que isto seja possível, todos equipamentos da empresa ou setor em questão devem estar previamente preparados com etiquetas contendo as respectivas informações. Então, com o uso do módulo PN532, compatível com o controlador ESP32, será possível realizar a leitura dos dados.

#### Envio dos Dados

Após o recebimento dos dados via NFC pelo dispositivo, estes serão armazenados temporariamente na memória do microcontrolador e então enviados via comunicação wi-fi para uma aplicação web. Para isto, será criado um servidor local utilizando a plataforma XAMPP e utilizada a biblioteca *WiFi.h* para conectar o ESP32 ao servidor.

#### Organização e Disponibilização dos Dados

Com o recebimento dos dados pela aplicação web, estes serão organizados em uma tabela de acordo com as informações de interesse, como: tipo de equipamento, local de origem, etc. Então, o usuário poderá exportar os dados para o computador em formato *.xls*, através de uma opção clicável na aplicação.

## Registro de Dados

O registro de novos dados em um equipamento terá início na aplicação web. O usuário digitará em uma aba específica da aplicação as informações que deseja registrar. Então, estas informações serão enviadas via *wi-fi* para o dispositivo. Então, utilizando o módulo PN532, as informações serão gravadas na *tag* NFC do equipamento.

### 3.2.4 Banco de Dados

Como sistema de gerenciamento do banco de dados, foi utilizado o *MySQL*. Essa escolha foi motivada principalmente pela presença nativa deste sistema na plataforma de servidor local XAMPP. Além disso, a linguagem utilizada, SQL, é amplamente difundida e robusta.

Então, utilizando a ferramenta *phpMyAdmin* do XAMPP, foi criada uma base de dados nomeada "tcc". Neste banco de dados foi criada a tabela "lista equip" contendo os seguintes índices:

**LEITURA:** recebe variáveis do tipo *int* auto incrementadas. Servirá como uma contagem do número de leituras realizadas.

**EQUIPAMENTO:** recebe variáveis do tipo *varchar*. Receberá *strings* contendo a identificação de cada equipamento lido.

**ORIGEM:** recebe variáveis do tipo *varchar*. Receberá *strings* contendo a cidade de origem de cada equipamento lido.

A estrutura do banco de dados criado pode ser vista na Figura 3.10.

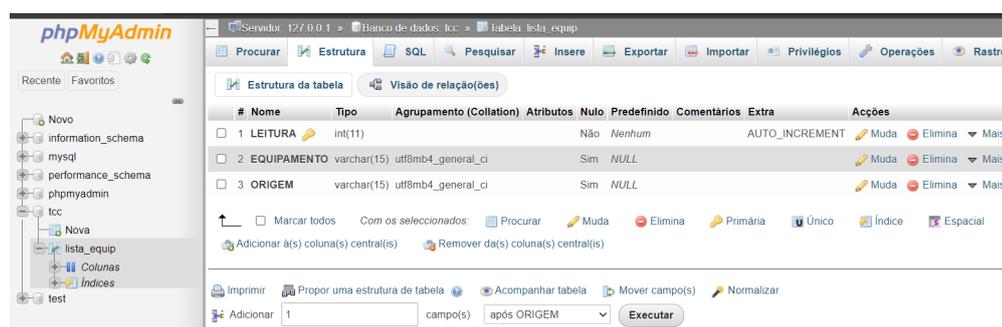


Figura 3.10: Estrutura do banco de dados. Fonte: Elaborado pelo autor

### 3.2.5 O Formato NDEF

O formato utilizado para a leitura e gravação das informações nas tags NFC será o NDEF (*NFC Data Exchange Format*). Este formato é estruturado conforme mostrado na Figura 3.11.

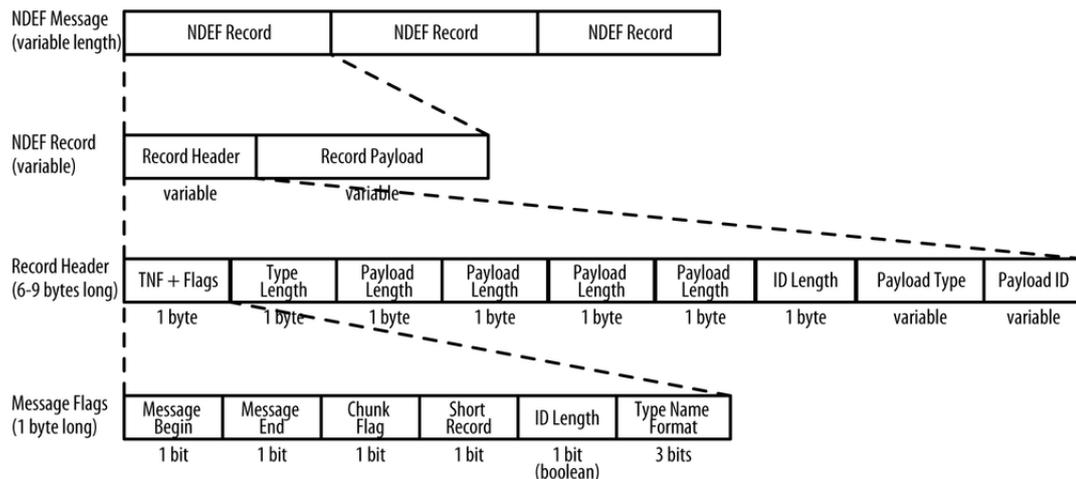


Figura 3.11: Estrutura do formato NDEF. Fonte: (O'REILLYMEDIA, 2022)

A seguir serão detalhados os elementos dessa estrutura:

**NDEF Message:** mecanismo de transporte para as informações (*NDEF Records*). Cada *NDEF Message* pode conter uma ou mais informações (tamanho variável).

**NDEF Record:** cada *NDEF Record* contém uma informação específica (*Record Payload*), além de uma estrutura (*Record Header*) que identifica o conteúdo e o tamanho da informação.

**Record Header:** possui as informações necessárias para interpretar dos dados, contendo os seguintes campos:

- **TNF + Flags:** o TNF (*Type Name Format*) indica em qual formato a mensagem deve ser interpretada (mídia, URL, etc.). Este campo também contém as *Message Flags*, que serão abordadas posteriormente.
- **Type Length:** especifica o tamanho do tipo de *payload*, em *bytes*.
- **Payload Length:** determina o tamanho do *payload*, que pode ser de um *byte* (*short record*) ou quatro *bytes*.
- **ID Length:** determina o tamanho do *Record ID*, se este estiver presente.
- **Record Type:** determina o tipo da informação que será transmitida.
- **Record ID:** valor do *Record ID*, caso este esteja presente.

**Record Payload:** conteúdo que está sendo transmitido.

**Message Flags:** informam como deve ser processado o *NDEF Record*. As *Message Flags* são:

- **MB (Message Begin)**: verdadeira quando o *NDEF Record* atual é o primeiro na *NDEF Message*.
- **ME (Message End)**: verdadeira quando o *NDEF Record* atual é o último na *NDEF Message*.
- **CF (Chunk Flag)**: verdadeira quando o *NDEF Record* está utilizando codificação *chunk*.
- **SR (Short Record)**: verdadeira se o formato *short record* é utilizado para o *payload length*.
- **IL (ID Length is present)**: verdadeira se o campo *ID Length* estiver presente.

A Figura 3.12 mostra um exemplo comentado de uma mensagem em formato NDEF.

```

18:05:37.487 -> NDEF Message 2 records, 33 bytes A mensagem possui 2 gravações e ocupa 33 bytes
18:05:37.487 -> NDEF Record Primeira gravação
18:05:37.487 -> TNF 0x1 Well Known TNF: Well Know
18:05:37.487 -> Type Length 0x1 1 Type Length: 1 byte
18:05:37.487 -> Payload Length 0x10 16 Payload Length: 16 bytes
18:05:37.487 -> Type 54 T Record Type: 54 T (texto)
18:05:37.487 -> Payload 02 65 6E 45 71 75 69 70 61 6D 65 6E 74 6F 2D 41 .enEquipamento-A Payload: Equipamento-A
18:05:37.487 -> Record is 20 bytes
18:05:37.487 -> NDEF Record Segunda gravação
18:05:37.487 -> TNF 0x1 Well Known TNF: Well Know
18:05:37.534 -> Type Length 0x1 1 Type Length: 1 byte
18:05:37.534 -> Payload Length 0x9 9 Payload Length: 9 bytes
18:05:37.534 -> Type 54 T Record Type: 54 T (texto)
18:05:37.534 -> Payload 02 65 6E 49 74 61 75 6E 61 .enItauna Payload: Itauna
18:05:37.534 -> Record is 13 bytes

```

Figura 3.12: Exemplo de mensagem no formato NDEF. Fonte: Elaborado pelo autor

### 3.3 A Aplicação Web

A primeira etapa para o desenvolvimento da aplicação foi a definição da interface, a fim de contemplar todas as funcionalidades do projeto mantendo a facilidade de uso. Optou-se pela organização das funções em três telas, mostradas na Figura 3.13. As características de cada tela estão expostas a seguir:

**Tela inicial** - Nesta tela são apresentadas duas opções clicáveis:

- Identificação: direciona o usuário para a tela de identificação.
- Registro: direciona o usuário para a tela de registro.

**Tela de identificação** - Esta tela deve ser utilizada em conjunto com o dispositivo no modo de leitura. A cada equipamento lido pelo usuário a planilha será atualizada com novas informações. Esta tela também conta com uma barra lateral com as seguintes opções clicáveis:

- Página inicial: direciona o usuário para a tela inicial.

- Exportar tabela: exporta a tabela atual para o computador.

**Tela de registro** - Esta tela deve ser utilizada em conjunto com o dispositivo no modo de escrita. O usuário insere as informações nos espaços determinados utilizando o teclado do computador. Então, o botão clicável "enviar" transfere os dados inseridos para o dispositivo. Nesta tela a barra lateral contém apenas o atalho para retornar à tela inicial.

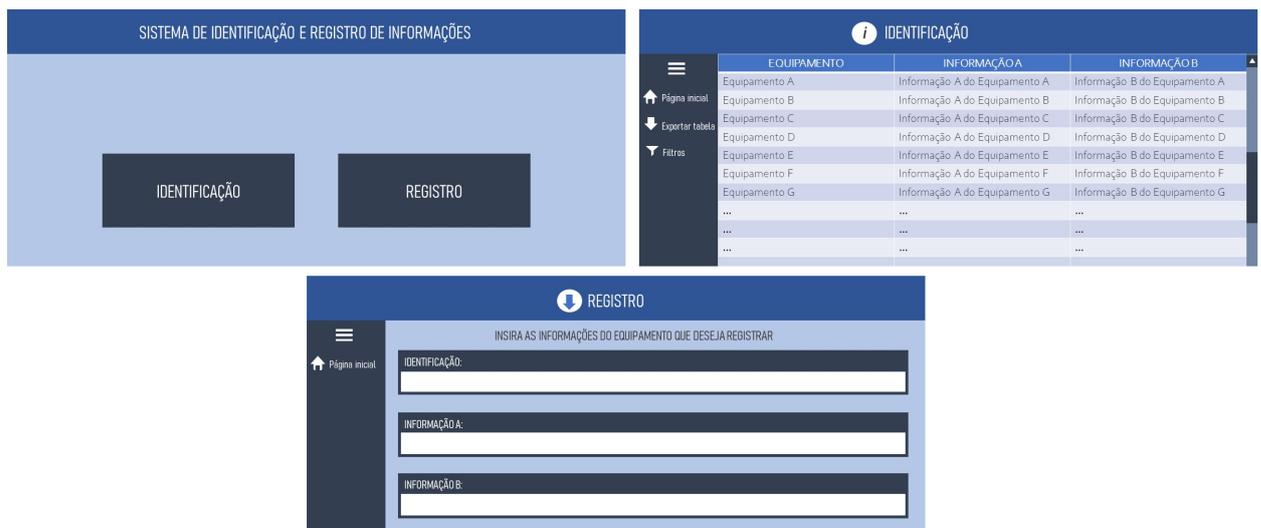


Figura 3.13: Interface da Aplicação Web. Fonte: Elaborado pelo autor

### 3.3.1 Desenvolvimento da Aplicação

Após a definição das funcionalidades e esboço das telas da aplicação pode-se iniciar o desenvolvimento da aplicação. Esta foi toda desenvolvida em um servidor local com o uso do pacote XAMPP.

Devido a limitação de tempo, foi desenvolvida apenas a tela de identificação, pois esta corresponde à função principal desejada no dispositivo. Além disso, conforme explicitado na seção 3.1, a função de escrita foi estabelecida como uma funcionalidade extra. A aplicação é composta por três arquivos na extensão PHP:

**ESP-SQL.php** : Recebe a requisição do ESP32 com os parâmetros *equip* e *ciudad* que contêm, respectivamente, o nome do equipamento lido e a cidade de origem do mesmo. Envia os parâmetros recebidos para o banco de dados.

**index.php** : Arquivo principal da aplicação, responsável por definir a disposição gráfica dos elementos da página e a interface de interação com o usuário. Busca as informações do banco de dados e exibe estas na tela.

**Exportar.php** : Arquivo responsável por realizar o download da tabela exibida na tela em formato *xls*.

Os códigos supracitados podem ser vistos integralmente no Apêndice A.

A Figura 3.14 mostra a interface da aplicação.



LEITURA	EQUIPAMENTO	ORIGEM
107	Equipamento-A	Itauna
108	Equipamento-B	Divinopolis
109	Equipamento-B	Divinopolis
110	Equipamento-A	Itauna
111	Equipamento-B	Divinopolis
112	Equipamento-A	Itauna
113	Equipamento-B	Divinopolis
114	Equipamento-A	Itauna
115	Equipamento-B	Divinopolis

Figura 3.14: Interface da Aplicação Web. Fonte: Elaborado pelo autor

Na região central da página é exibida a tabela com os dados lidos. Na barra lateral está localizada a função de exportar a tabela. Ao clicar nessa opção é acessado o arquivo *Exportar.php* mencionado anteriormente e o download da planilha é realizado. A Figura 3.15 mostra a tabela exportada em formato *.xls*.

	A	B	C
1	<b>LEITURA</b>	<b>EQUIPAMENTO</b>	<b>ORIGEM</b>
2	107	Equipamento-A	Itauna
3	108	Equipamento-B	Divinopolis
4	109	Equipamento-B	Divinopolis
5	110	Equipamento-A	Itauna
6	111	Equipamento-B	Divinopolis
7	112	Equipamento-A	Itauna
8	113	Equipamento-B	Divinopolis
9	114	Equipamento-A	Itauna
10	115	Equipamento-B	Divinopolis

Figura 3.15: Tabela exportada em formato *.xls*. Fonte: Elaborado pelo autor

## 3.4 O Dispositivo de Identificação

### 3.4.1 Definição da Interface do Dispositivo

A primeira etapa no desenvolvimento do dispositivo foi a definição de interface do mesmo. Considerando todas as funcionalidades planejadas, foi definido o *design* mostrado na Figura 3.16.

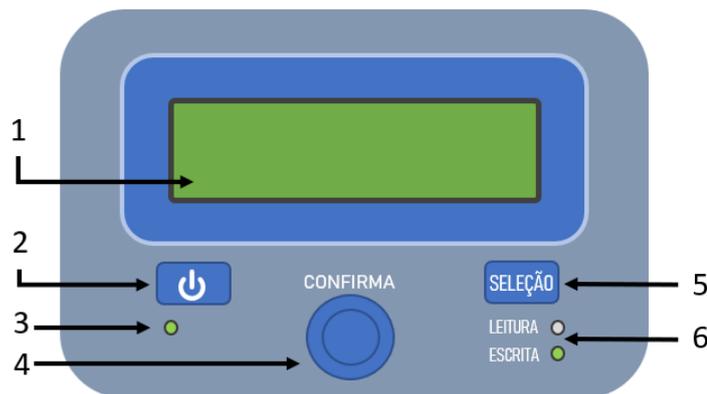


Figura 3.16: Design da interface do Dispositivo. Fonte: Elaborado pelo autor

A seguir estão descritas as funções de cada elemento da interface:

1. **Display LCD** - Mostra as informações lidas de um equipamento ou, no modo escrita, as informações a serem escritas no equipamento.
2. **Botão liga/desliga** - Utilizado para ligar e desligar o dispositivo.
3. **LED indicador** - Ficará aceso caso o dispositivo esteja ligado.
4. **Botão de seleção** - Alterna entre as opções dos menus.
5. **Botão de confirmação** - Confirma a opção selecionada.
6. **LEDs indicadores** - Indicam qual modo de operação está selecionado.

### 3.4.2 Desenvolvimento do Protótipo

A primeira etapa no desenvolvimento do protótipo foi definir quais pinos do ESP32 seriam utilizados para cada componente. Esta definição foi baseada nas necessidades de alimentação e comunicação de cada componente em conjunto com as funcionalidades de cada entrada e saída do ESP32. A pinagem final pode ser vista na Tabela 3.2.

Tabela 3.2: Relação de pinos utilizados no ESP32

Componente	Pino do Componente	Pino do ESP32
LCD	GND	GND
	VCC	VIN
	SDA	D21
	SCL	D22
Módulo NFC	SCK	D18
	MISO	D19
	MOSI	D23
	SS	D5
	VCC	VIN
Botão 1	GND	GND
	IN	D25
Botão 2	OUT	3V3
	IN	D26
	OUT	3V3

Então, deu-se início à programação do código para o ESP32. Inicialmente, foram incluídas as bibliotecas necessárias e foram definidos os pinos para os botões. Então, foi feita a configuração do LCD, do módulo NFC, do *wifi* (SSID e senha) e do servidor local (IP da rede). Para finalizar este trecho do código, foram definidas as variáveis necessárias.

```

1 //BIBLIOTECAS
2 #include <WiFi.h>
3 #include <LiquidCrystal_I2C.h>
4 //...demais bibliotecas no apendice...
5 //BIBLIOTECAS
6 //CONFIG BOTOES
7 #define BUTTON1_PIN 25
8 #define BUTTON2_PIN 26
9 //CONFIG LCD-----
10 int lcdColumns = 16;
11 int lcdRows = 2;
12 LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
13 //CONFIG NFC
14 PN532_SPI pn532spi(SPI, SS);
15 NfcAdapter nfc = NfcAdapter(pn532spi);
16 //CONFIG WIFI-----
17 const char* ssid = "*****"; /*ocultado por segurança*
18 const char* pwd = "*****"; /*ocultado por segurança*
19 IPAddress server(192,168,1,3);
20 WiFiClient client;
21 //VARIABLES-----
22 int lastState = LOW;
23 int currentState = 0;
24 //...demais variaveis no apendice...

```

Listing 3.1: Trecho de código do microcontrolador

A seguir, foi programada a função *setup*. Nesta etapa são inicializados o LCD e o *wifi*. Além disso, esta função contém um menu de seleção em *loop*, para que o usuário escolha o modo de operação (leitura ou escrita), visualizando as opções no LCD e selecionando com os botões.

```

1 void setup() {
2     Serial.begin(115200); //INICIA O MONITOR SERIAL
3     pinMode(BUTTON1_PIN, INPUT_PULLUP);
4     pinMode(BUTTON2_PIN, INPUT_PULLUP);
5     nfc.begin(); //INICIA O MODULO NFC
6     lcd.init(); //INICIA O LCD
7     lcd.backlight(); //LIGA ALUZA DE FUNDO DO LCD
8     WiFi.begin(ssid, pwd); //INICIA O WIFI
9     lcd.setCursor(0,0);
10    lcd.print("Conectando");
11        //...codigo completo nos apendices...
12    while(true){//LOOP DE SELECAO DE MODO
13        lcd.setCursor(0, 0);
14        lcd.print("Selecione o modo");
15        //...codigo completo nos apendices...

```

Listing 3.2: Trecho de código do microcontrolador

Então, foi programada a função *loop*. Esta função é dividida em dois trechos, leitura e escrita. A definição de qual trecho será acessado é feita com o uso de uma variável de seleção cujo valor é definido de acordo com a escolha do usuário na função *setup*.

No trecho correspondente à função de leitura, o módulo NFC é inicializado e aguarda a aproximação de uma *tag*. Quando ocorre a aproximação as informações são lidas e exibidas no display LCD. Então é iniciado um menu em *loop* para que o usuário informe se deseja enviar os dados lidos. Por fim, um novo menu é exibido, para que o usuário informe se deseja realizar uma nova leitura.

```

1 void loop(){//LOOP PRINCIPAL
2     currentState2 = 0;
3     lastState2 = LOW;
4     lastState = LOW;
5     currentState = 0;
6     lcd.clear();
7     if (opcao==0){
8         lcd.setCursor(4,0);
9         lcd.print("LEITURA");
10        while(true){
11            String payloadAsString[2] = {"",""};
12                if (novaleitura==0)
13                    break;
14                lcd.clear();
15                lcd.setCursor(4,0);
16                lcd.print("LEITURA");
17                lcd.setCursor(2,1);
18                if (nfc.tagPresent()){

```

```

19         NfcTag tag = nfc.read();
20         if (tag.hasNdefMessage()) {
21             NdefMessage message = tag.getNdefMessage();
22             if (message.getRecordCount() != 1) {
23                 }
24         //...codigo completo nos apendices...

```

Listing 3.3: Trecho de código do microcontrolador

Já no trecho correspondente à função de escrita, novamente o módulo NFC é iniciado e aguarda a aproximação de uma *tag*. Como a integração entre a aplicação web e o modo de escrita não foi implementada, as informações desejadas devem ser inseridas diretamente no código antes da compilação e do envio para o ESP32. Ao aproximar a *tag* as informações são gravadas na mesma e exibidas no LCD. Por fim, um menu é exibido para que o usuário informe se deseja gravar estas informações em outra *tag*.

```

1     else if (opcao==1){
2         lcd.setCursor(4,0);
3         lcd.print("ESCRITA");
4         while (true) {
5             if (novaescrita==0)
6                 break;
7             lcd.clear();
8             lcd.setCursor(4,0);
9             lcd.print("ESCRITA");
10            if (nfc.tagPresent()) { /
11                NdefMessage message = NdefMessage();
12                message.addTextRecord("Equipamento-A");
13                message.addTextRecord("Itauna");
14                bool success = nfc.write(message);
15                if (success) {
16                    lcd.setCursor(0,1);
17                    lcd.print("Equipamento-A");
18                } else {
19                    lcd.setCursor(0,1);
20                    lcd.print("Falha na Escrita");
21                //...codigo completo nos apendices...

```

Listing 3.4: Trecho de código do microcontrolador

Na Figura 3.17 pode ser visto o protótipo montado em uma matriz de contato. O código do microcontrolador pode ser visto de forma integral no Apêndice B.

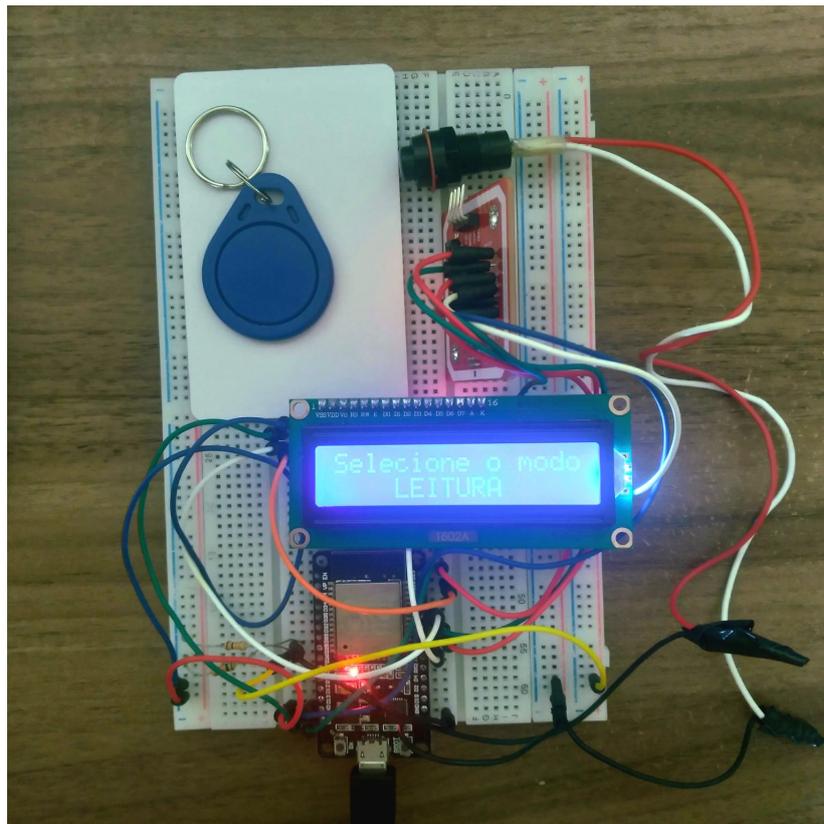


Figura 3.17: Montagem do protótipo. Fonte: Elaborado pelo autor

As conexões podem ser observadas de forma mais clara no esquemático do dispositivo elaborado no *software Fritzing*, mostrado na Figura 3.18.

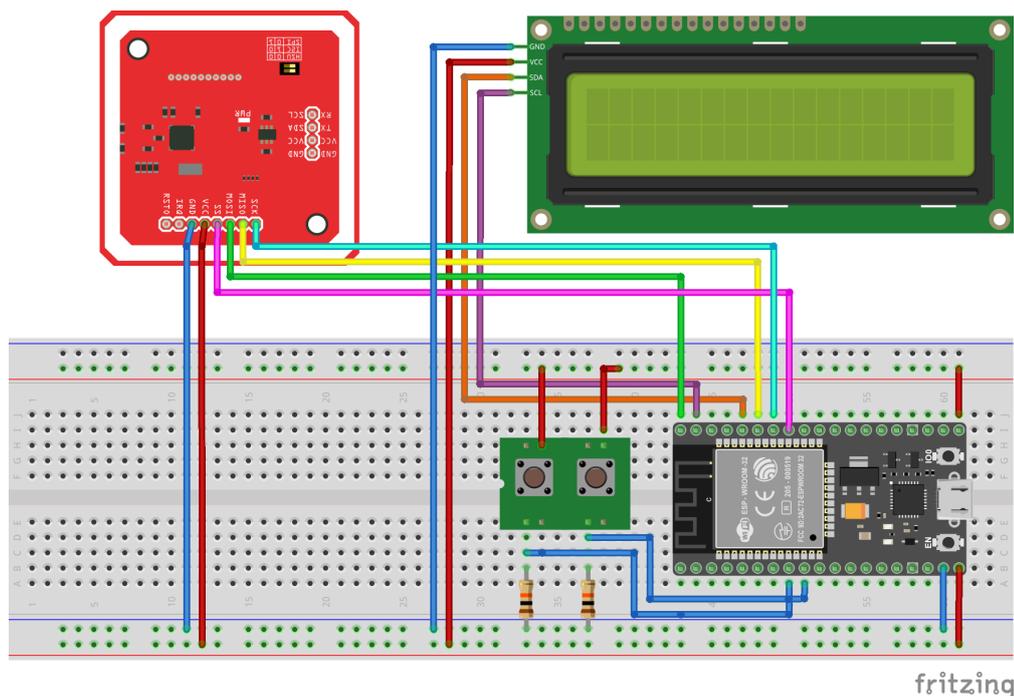


Figura 3.18: Esquemático do dispositivo. Fonte: Elaborado pelo autor

### 3.4.3 Projeto da Carcaça

O projeto da carcaça do dispositivo levou em consideração a melhor posição de cada componente de forma a manter uma boa ergonomia para o operador. Dessa forma, foi definido que a carcaça deveria ter a parte inferior mais estreita, de forma que o dispositivo possa ser segurado com facilidade com uma mão. Portanto, nesta região foram dispostos a bateria, o módulo de carga e o módulo *step-up* em orientação vertical, conforme pode ser visto na Figura 3.19

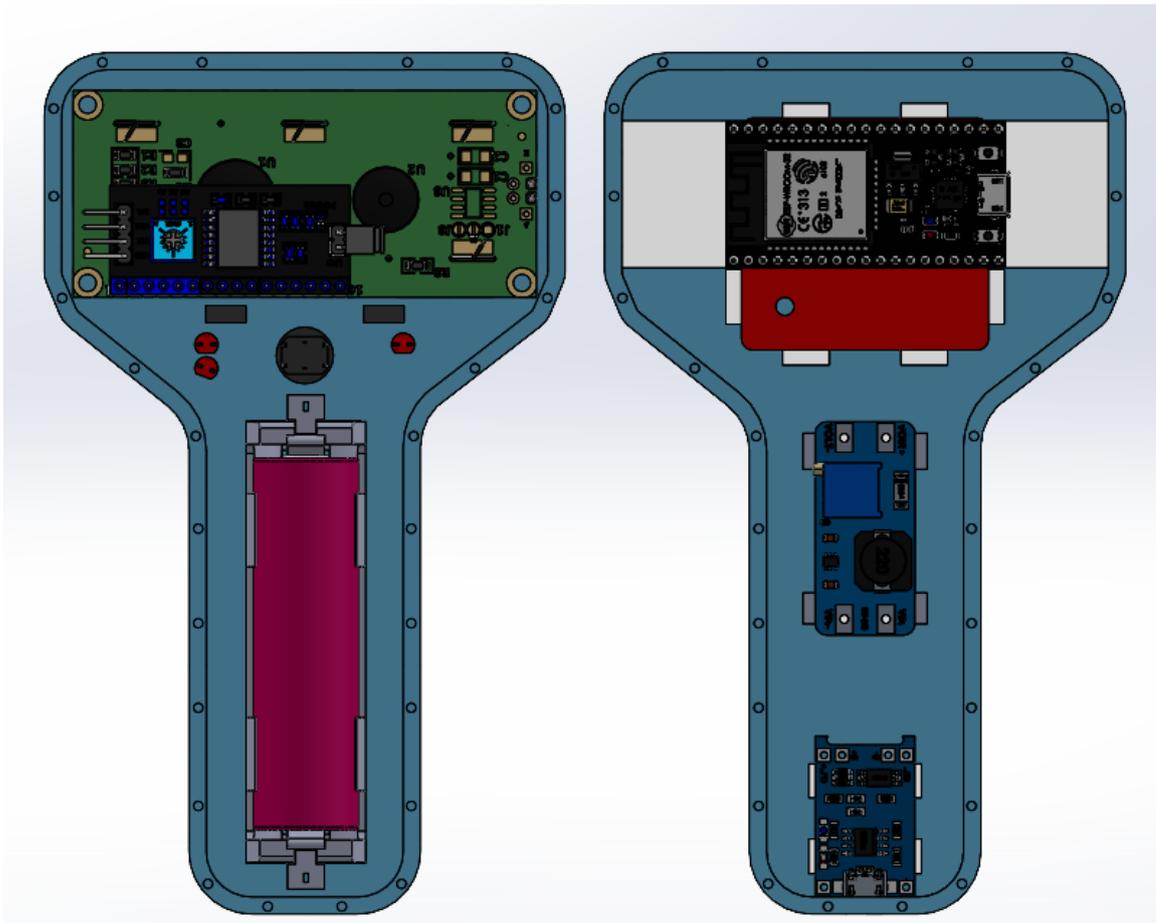


Figura 3.19: Desenho interno das duas metades da carcaça. Fonte: Elaborado pelo autor

Na parte superior foram alocados o display LCD, o ESP32 e o módulo NFC. Os botões e LEDs, em conjunto com o LCD, foram dispostos de forma a seguir a interface proposta na Figura 3.16. O resultado obtido pode ser visto na Figura 3.20

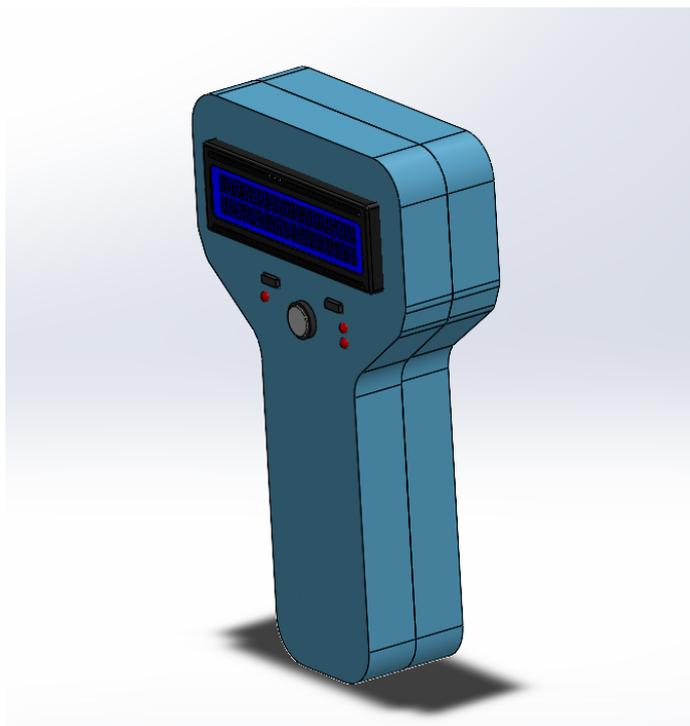


Figura 3.20: Desenho da carcaça. Fonte: Elaborado pelo autor

A carcaça é composta por duas partes, com pontos de encaixe, conforme pode ser observado na Figura 3.21. Além disso, nessa figura é possível notar o entalhe feito na tampa traseira no local do leitor NFC, para indicar ao usuário que esta é a região a ser apontada para as *tags*.

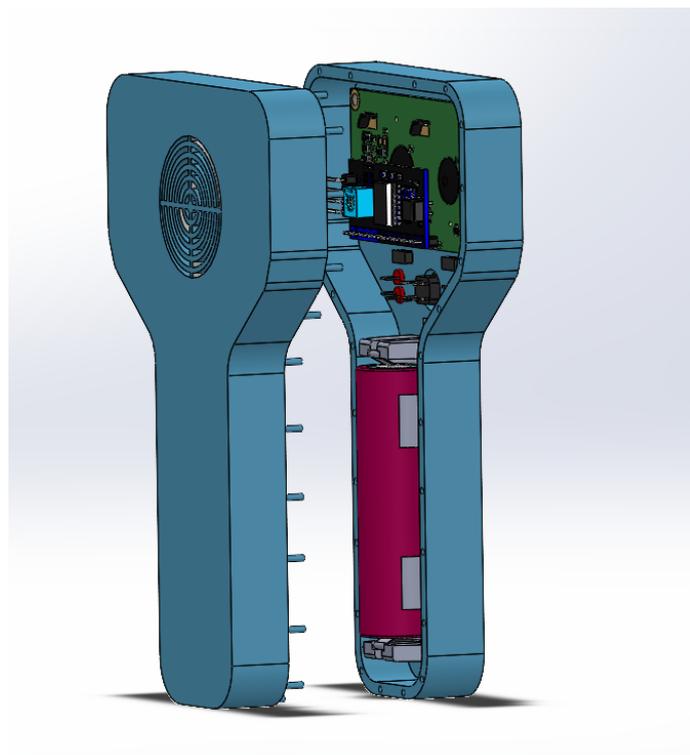


Figura 3.21: Desenho da carcaça. Fonte: Elaborado pelo autor

As Figuras 3.22 e 3.23 mostram, respectivamente, a peça da frente e a peça do fundo da carcaça, com as dimensões em milímetros.

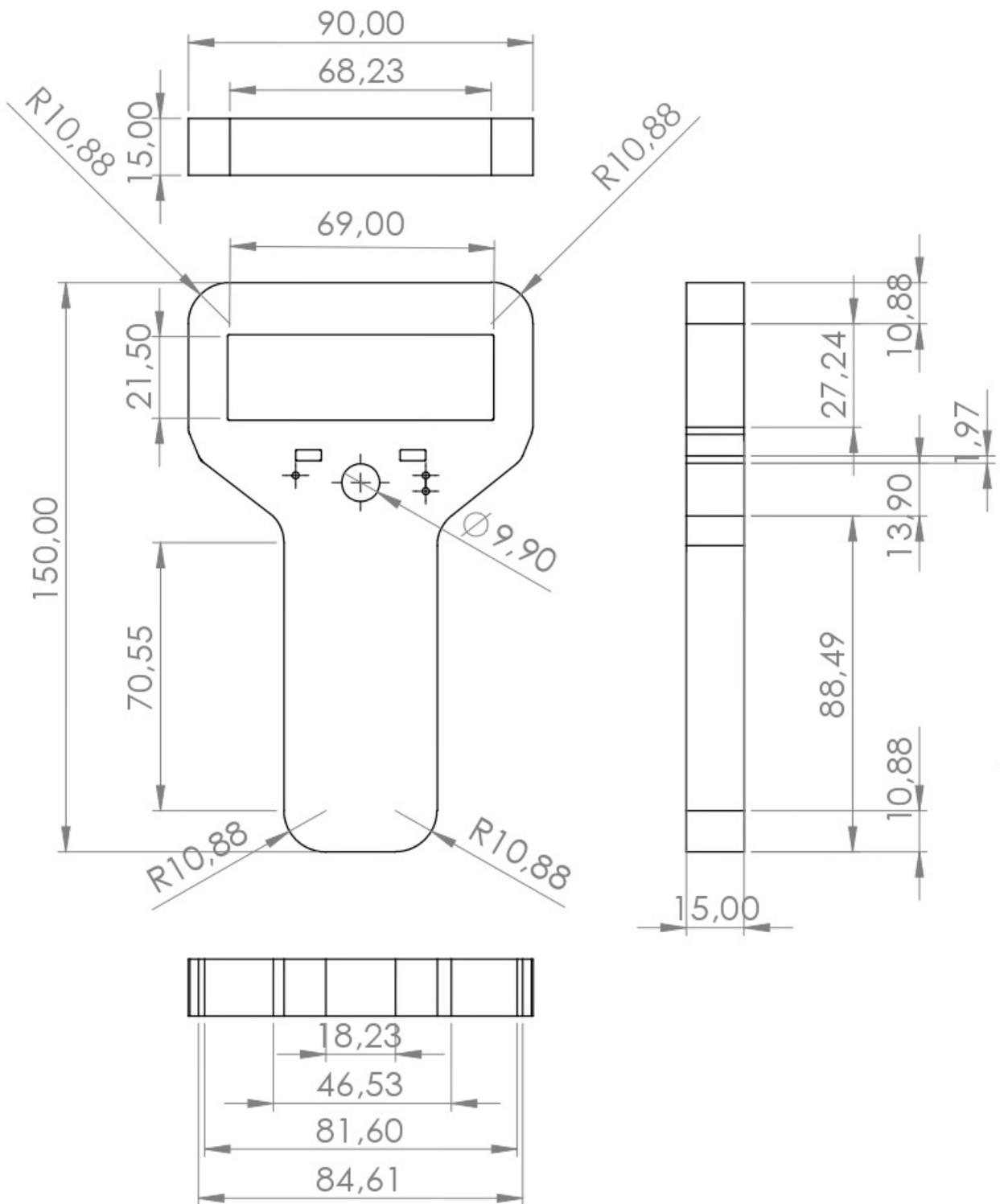


Figura 3.22: Desenho técnico da frente da carcaça. Fonte: Elaborado pelo autor

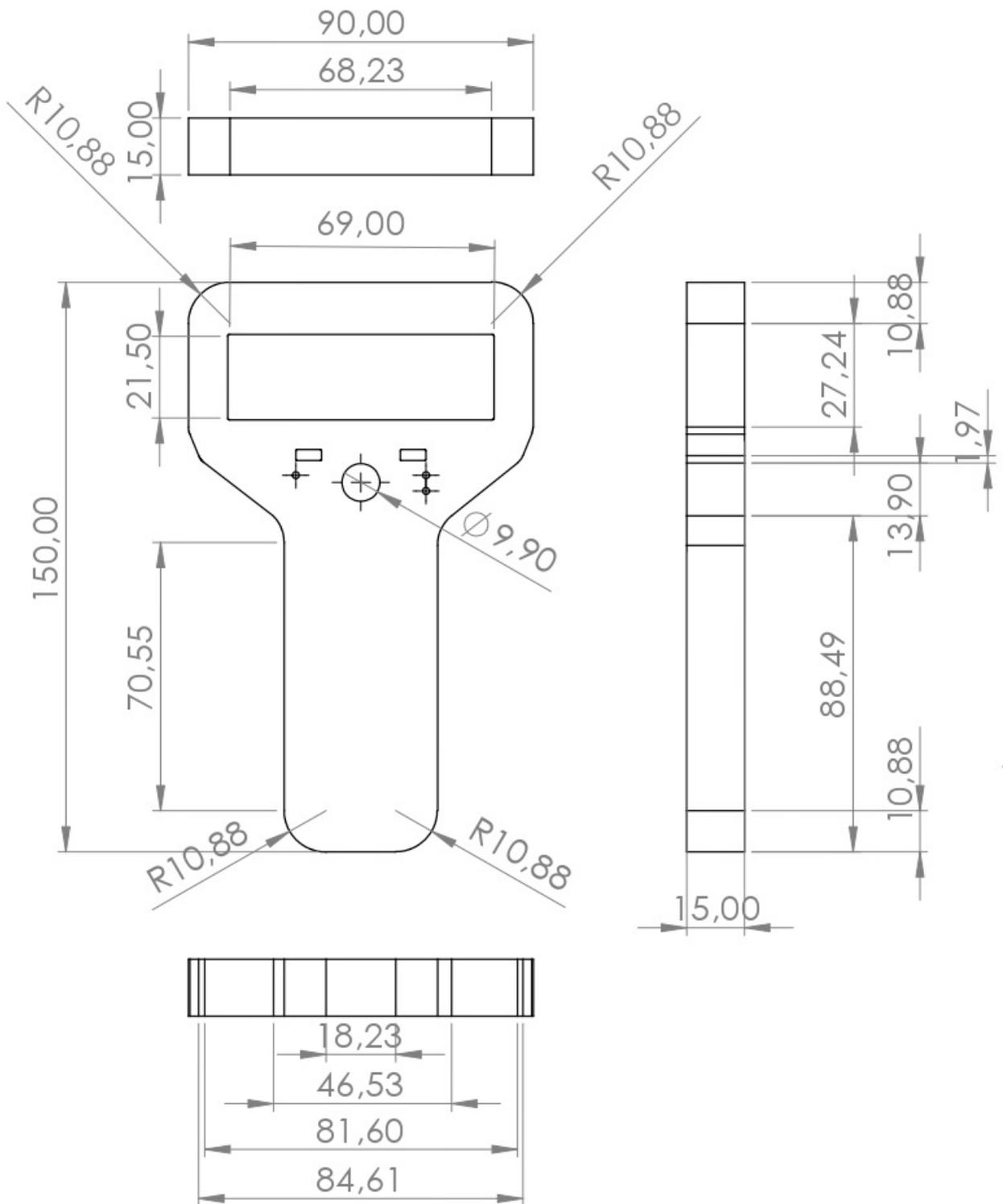


Figura 3.23: Desenho técnico do fundo da carcaça. Fonte: Elaborado pelo autor

### 3.4.4 Procedimento de Uso

A Figura 3.24 mostra o fluxograma de funcionamento do dispositivo.

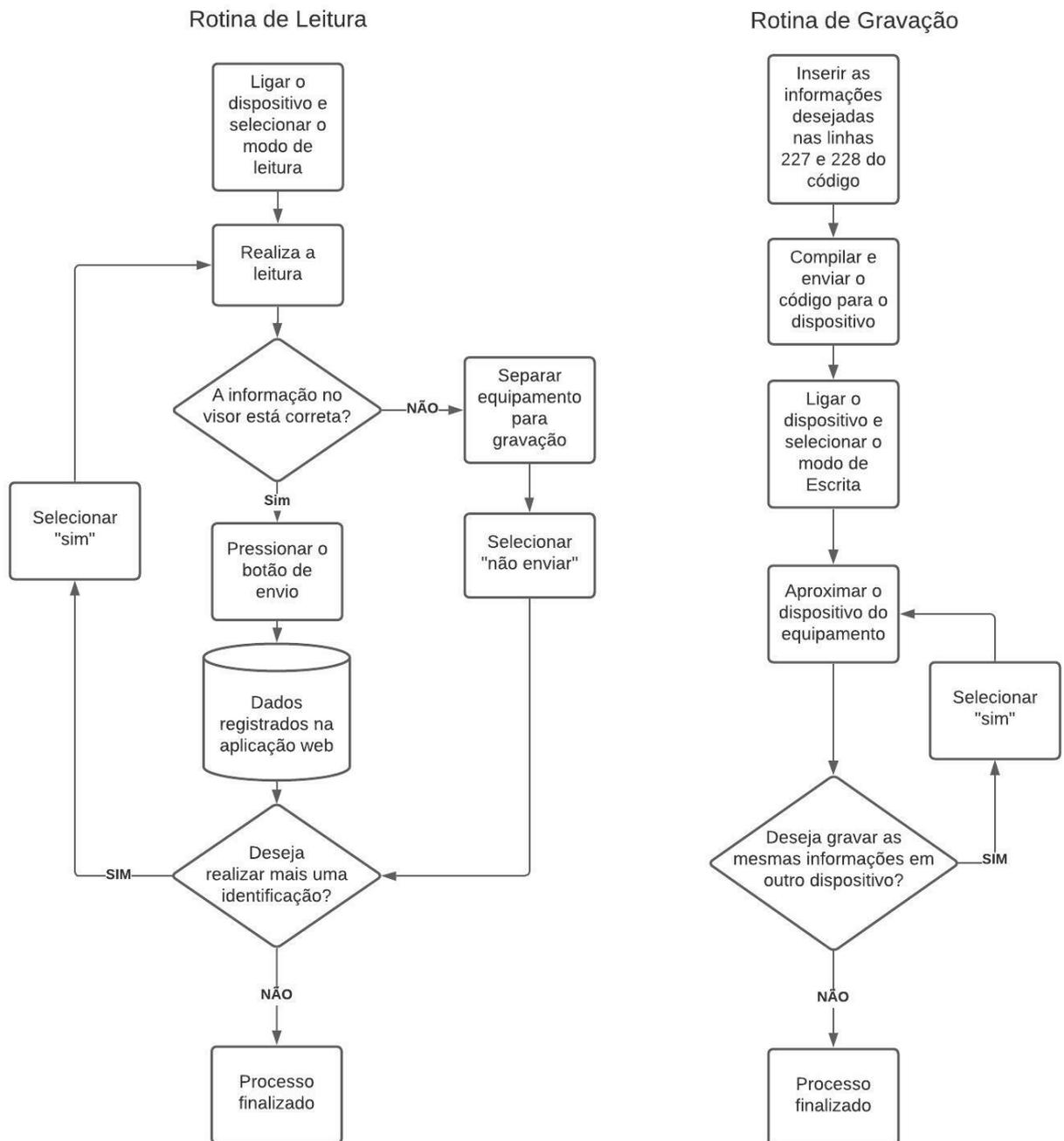


Figura 3.24: Fluxograma de funcionamento do dispositivo. Fonte: Elaborado pelo autor

As seções a seguir são dedicadas a explicar o procedimento de cada rotina, expondo também a informação no visor em cada etapa.

### 3.4.5 Rotina de Leitura

Para realizar a identificação de diversos equipamentos consecutivamente, primeiramente deve-se ligar o dispositivo e aguardar a conexão com o wifi. A mensagem "Conectando" será exibida no visor e, após a conexão, surgirá a mensagem "Conectado!", conforme pode ser observado na Figura 3.25.



Figura 3.25: Telas de conexão com wifi. Fonte: Elaborado pelo autor

Então, será exibida a tela de seleção de modo. O botão de seleção irá alternar entre "LEITURA" e "ESCRITA", conforme mostrado na Figura 3.26. Ao pressionar o botão de confirmação enquanto a opção "LEITURA" estiver sendo exibida o dispositivo entrará no *loop* de leitura.

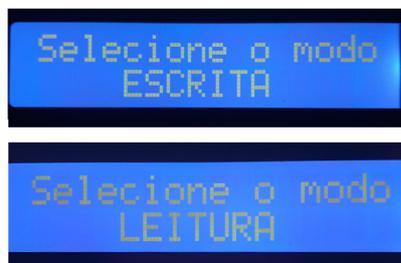


Figura 3.26: Telas de seleção de modo. Fonte: Elaborado pelo autor

Uma vez que o dispositivo esteja no modo de leitura, basta aproximar o dispositivo com tag NFC contendo uma informação no formato NDEF. Então, a informação será exibida no visor, juntamente com um menu de seleção, conforme a Figura 3.27. Ao pressionar o botão de seleção, a opção de enviar irá alternar entre "SIM" e "NÃO". Ao pressionar o botão de confirmação enquanto a opção "SIM" é mostrada a informação lida será enviada ao banco de dados. Caso contrário, a informação será descartada.

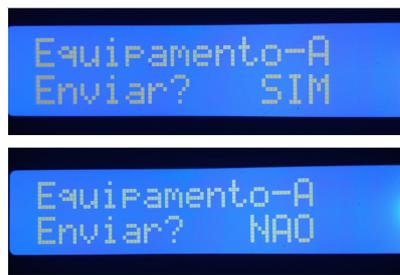


Figura 3.27: Telas de confirmação de envio. Fonte: Elaborado pelo autor

Então, será exibido um menu de seleção para que o usuário informe se deseja realizar uma nova leitura, conforme a Figura 3.28. Novamente, o botão de seleção alterna entre as opções "SIM" e "NÃO". Ao pressionar o botão de confirmação enquanto a opção "SIM" é mostrada na tela o dispositivo irá aguardar a aproximação de um novo equipamento.

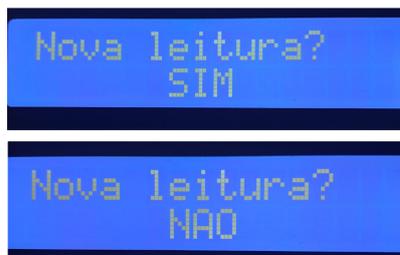


Figura 3.28: Telas de confirmação de nova leitura. Fonte: Elaborado pelo autor

Caso o usuário selecione a opção "NÃO" no menu anterior, será exibida a tela mostrada na Figura 3.29, contendo a mensagem "Fim do procedimento". Essa tela permanecerá no visor até que o dispositivo seja desligado.



Figura 3.29: Tela final. Fonte: Elaborado pelo autor

Após estas etapas todas as informações coletadas estarão organizadas na aplicação web, na qual será possível visualizar e exportar os dados.

### 3.4.6 Rotina de Escrita

Antes de iniciar o processo de gravação, é necessário inserir as informações desejadas diretamente no código (linhas 227 e 228), compilar e enviar o código para o ESP32.

Então, deve-se ligar o dispositivo e aguardar a conexão com o wifi. A mensagem "Conectando" será exibida no visor e, após a conexão, surgirá a mensagem "Conectado!", conforme pode ser observado na Figura 3.30.

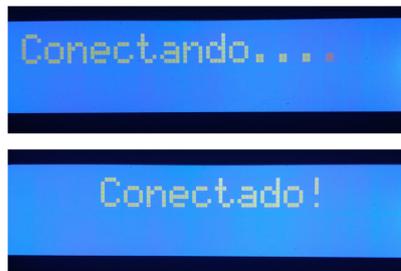


Figura 3.30: Telas de conexão com wifi. Fonte: Elaborado pelo autor

Então, será exibida a tela de seleção de modo. O botão de seleção irá alternar entre "LEITURA" e "ESCRITA", conforme mostrado na Figura 3.31. Ao pressionar o botão de confirmação enquanto a opção "ESCRITA" estiver sendo exibida o dispositivo entrará no *loop* de escrita.



Figura 3.31: Telas de seleção de modo. Fonte: Elaborado pelo autor

Em seguida, basta aproximar o dispositivo contendo uma tag NFC formatada para informações NDEF. O visor exibirá a informação que foi gravada na tag, conforme mostrado na Figura 3.32.



Figura 3.32: Tela de gravação. Fonte: Elaborado pelo autor

Caso a gravação falhe, devido ao alinhamento incorreto entre o módulo NFC e a tag presente no dispositivo, será exibida a mensagem "Falha na Escrita", conforme mostrado na Figura 3.33.



Figura 3.33: Tela de falha na escrita. Fonte: Elaborado pelo autor

Após a gravação da informação, bem sucedida ou não, será exibido um menu para que o usuário informe se deseja gravar a informação novamente, conforme mostrado na Figura 3.34. Ao pressionar o botão de seleção a opção exibida irá alternar entre "SIM" e "NÃO". Ao pressionar o botão de confirmação enquanto a opção "SIM" é exibida o dispositivo aguardará a aproximação de outro equipamento contendo uma tag.



Figura 3.34: Telas de confirmação de nova escrita. Fonte: Elaborado pelo autor

Caso o usuário selecione a opção "NÃO" no menu anterior, será exibida a tela mostrada na Figura 3.35, contendo a mensagem "Fim do procedimento". Essa tela permanecerá no visor até que o dispositivo seja desligado.



Figura 3.35: Tela final. Fonte: Elaborado pelo autor

## 3.5 Orçamento

A Tabela 3.3 mostra os gastos para a elaboração deste trabalho.

Item	Preço Estimado
Display LCD com módulo I2C	R\$28,90
Adesivos NFC	R\$20,00 - R\$30,00
Módulo de leitura e gravação NFC PN532	R\$20,00 - R\$50,00
Bateria Li-Ion 18650 2.200mAh	R\$64,90
Cabos para conexões	R\$15,50
Carcaça para o dispositivo	R\$50,00 - R\$100,00
Microcontrolador ESP32	R\$39,90
<b>Total</b>	<b>R\$239,20 - R\$329,20</b>

Tabela 3.3: Orçamento estimado para os gastos do projeto

Todos os item foram adquiridos com recursos próprios do autor.

## Considerações Finais

### 4.1 Conclusões

O presente trabalho teve por objetivo o desenvolvimento de um dispositivo de identificação de equipamentos em ambiente industrial. O propósito para o desenvolvimento de tal dispositivo é possibilitar a automação parcial do processo de identificação de equipamentos, sobretudo para setores da indústria nos quais a automação completa é inviável.

A primeira etapa para o desenvolvimento do projeto foi, frente a experiência pessoal que motivou o projeto, definir as funcionalidades essenciais do dispositivo, de forma a atender a demanda apresentada. Então, deu-se início à uma revisão bibliográfica, a fim de compreender os recursos tecnológicos disponíveis atualmente para o desenvolvimento do dispositivo. Também foi realizado o estudo do Estado da Arte, para os projetos recentes na área de estudo fossem considerados.

Então, foi iniciado o processo de definição dos componentes e *softwares* necessários para o desenvolvimento do dispositivo. Primeiramente, foram determinados os elementos que compõem a parte eletrônica do projeto, de forma a atender às necessidades deste. Nesta etapa notou-se a necessidade de componentes não previstos nas etapas preliminares, como os módulo conversor *boost step-up* e o módulo de carregamento da bateria. Por fim, foram definidos os *softwares* necessários para cada etapa do projeto.

Posteriormente, foi definido o fluxo ideal das informações, contemplando a obtenção, envio, exibição e registro dos dados. Então, foi definido o uso do sistema de gerenciamento de banco de dados MySQL, finalizando esta etapa com a criação da base de dados.

A próxima etapa foi definição da interface da aplicação web, contemplando tanto o aspecto visual quanto as funcionalidades necessárias. Após a definição da interface desejada, foi iniciado o processo de desenvolvimento da aplicação. Como a opção de escrever informações foi definida como uma funcionalidade extra, o desenvolvimento foi focado na função de exibir os dados de forma organizada e possibilitar o download destes. Então, a tela de leitura de informações proposta foi desenvolvida, contendo apequenas pequenas alterações visuais e no

*layout.*

Então, deu-se início ao projeto do dispositivo. Inicialmente, foi definida a interface de exibição de informações e interação com o usuário, para que ficasse claro as formas de entrada necessárias para garantir uma operação confortável do aparelho. A seguir, foram definidas as entradas e saídas do microcontrolador a serem utilizadas, de forma a atender a necessidade de cada componente. Com a definição dos pinos estabelecida, foi iniciada a programação do microcontrolador, progredindo de forma gradual na inclusão dos componentes e funções. Em paralelo a programação do microcontrolador, o protótipo foi montado em uma matriz de contatos para realização de testes iterativos para correção de falhas na programação.

Com o microcontrolador programado e o protótipo montado, deu-se início a integração entre o dispositivo e aplicação web. Nesta etapa houve dificuldade pois, inicialmente, foram feitas tentativas de utilizar o ESP32 como servidor. Porém, essa funcionalidade causou conflito com o módulo NFC, de forma que o uso de ambos concomitantemente causava muita lentidão na abertura da página web. Então, foi estabelecido o uso de um servidor local para a aplicação web. Então, foi desenvolvido o código *ESP-SQL.php*, que recebe os dados do ESP32 via requisição e os envia para o banco de dados. Foram performados diversos testes e ajustes para se alcançar o resultado desejado.

A etapa final no desenvolvimento do projeto foi o desenho da carcaça do dispositivo. Esta foi planejada para oferecer boa ergonomia no uso com apenas uma mão. Esta decisão de *design* guiou o posicionamento dos componentes na carcaça.

Por fim, conclui-se que o projeto foi desenvolvido de forma a contemplar todas as funções básicas propostas, alcançando resultados satisfatórios. Além disso, a forma de desenvolvimento não impossibilita a inclusão das funções faltantes, que serão abordadas na proposta de continuidade. O protótipo mostrou ter potencial para ser aplicado no problema prático que motivou este trabalho. Além disso, no desenvolvimento do projeto, foram empregados métodos multidisciplinares, principal característica da Engenharia Mecatrônica.

## 4.2 Propostas de Continuidade

- Implementação da função de escrita através da aplicação web, eliminando a necessidade de inserir informações diretamente no código antes de compilar e enviar ao ESP32.
- Substituição do display LCD por um modelo de maiores dimensões (128x64 pixel, por exemplo), a fim de exibir mais informações simultaneamente e elaborar uma interface mais amigável.
- Implementação da opção de configurar a rede wifi a partir da aplicação web, eliminando a necessidade da configuração no código.

## Códigos da Aplicação Web

### A.1 ESP-SQL.php

```
1 <?php
2
3 $link = mysqli_connect("localhost", "root", "", "");
4
5 // Vari veis para os par metros GET
6 $EQUIPAMENTO = mysqli_real_escape_string($link, $_GET['EQUIPAMENTO
7     ']);
8 $ORIGEM = mysqli_real_escape_string($link, $_GET['ORIGEM']);
9 // Vari veis para os par metros GET
10
11 $dbh; // Database handler: conectar e gerenciar o banco de dados
12 $stmt; // Statement: armazenar uma query do banco e executa-la
13
14 // String de conex o com o banco de dados
15 $connStr = 'mysql:host=localhost;dbname=tcc';
16 // String de conex o com o banco de dados
17
18 try{ // Tenta conectar ao banco
19     $dbh = new PDO($connStr, 'root', '');
20 }
21 catch(PDOException $e){ // Caso ocorra erro
22     echo $e->getMessage(); // Retorna o erro
23     die(); //Encerra a execu o
24 }
25
26 $stmt = $dbh->prepare( // Prepara a query
27     'INSERT INTO lista Equip(EQUIPAMENTO, ORIGEM)
28     VALUES (:EQUIPAMENTO, :ORIGEM)'
29 );
30
31 // Parametros da query recebem os valores das vari veis
32 $stmt->bindValue(':EQUIPAMENTO', $EQUIPAMENTO);
33 $stmt->bindValue(':ORIGEM', $ORIGEM);
```

```

34 // Parametros da query recebem os valores das vari veis
35
36
37 if($stmt->execute()){ // Tenta executar o comando e retorna
    sucesso ou erro
38     echo 'sucesso';
39 }else{
40     echo 'erro';
41 }

```

Listing A.1: Código do arquivo ESP-SQL.php

## A.2 index.php

```

1 <html>
2
3     <! IMPORTA PACOTE DO BOOTSTRAP >
4     <link
5         rel="stylesheet"
6         href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/
           bootstrap.min.css"
7         integrity="sha384-MCw98/
           SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPM0"
8         crossorigin="anonymous"
9     >
10
11     <! IMPORTA PACOTE DO BOOTSTRAP >
12     <link
13         rel="stylesheet"
14         href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.7.2/font/
           bootstrap-icons.css"
15     >
16
17     <! DEFINE O ALINHAMENTO >
18     <style>
19         h1 {text-align: center;}
20     </style>
21
22     <! CABE ALHO >
23     <header style="background-color:#282828;">
24         <div class="container">
25             <h1 class="display-4" style="font-size: 2rem; color: white;
               " > <i class="bi bi-info-circle-fill" style="font-size:
               2rem; color: white;"></i> IDENTIFICA 0 </h1>
26         </div>
27     </header>
28
29     <! CONFIGURA BARRA LATERAL >
30     <head>
31         <title>TCC</title>

```

```
32 <meta charset="UTF-8">
33 <meta name="viewport" content="width=device-width, initial-
    scale=1">
34 <style>
35     .sidenav {
36         height: 100%;
37         width: 160px;
38         position: fixed;
39         z-index: 1;
40         top: 0;
41         left: 0;
42         background-color: #111;
43         overflow-x: hidden;
44         padding-top: 20px;
45     }
46     .sidenav a {
47         padding: 6px 8px 6px 16px;
48         text-decoration: none;
49         font-size: 15px;
50         color: #818181;
51         display: block;
52     }
53     .sidenav a:hover {
54         color: #f1f1f1;
55     }
56     .main {
57         margin-left: 160px;
58         font-size: 28px;
59         padding: 0px 10px;
60     }
61     @media screen and (max-height: 450px) {
62         .sidenav {padding-top: 15px;}
63         .sidenav a {font-size: 18px;}
64     }
65 </style>
66 </head>
67
68 <! CONFIGURA COR DE FUNDO>
69 <body style="background-color:#282828;">
70
71 <! PREENCHE BARRA LATERAL>
72 <div class="sidenav">
73     <a href=""><i class="bi bi-grid-3x3-gap-fill"></i></a>
74     <a href="Exportar.php" >Exportar tabela</a>
75 </div>
76
77 <! PREENCHE TABELA>
78 <div class="main">
79     <table class="table table-striped table-dark">
80         <thead>
81             <tr>
```

```

82         <th scope="col">LEITURA</th>
83         <th scope="col">EQUIPAMENTO</th>
84         <th scope="col">ORIGEM</th>
85     </tr>
86 </thead>
87 <tbody>
88     <?php
89         $link = mysqli_connect("localhost", "root", "", "");
90         mysqli_select_db($link, 'tcc');
91         $query = "SELECT * FROM lista_equip";
92         $result = mysqli_query($link,$query);
93         while($row = mysqli_fetch_array($result)){?>
94             <tr>
95                 <td> <?php echo $row['LEITURA']; ?> </td>
96                 <td> <?php echo $row['EQUIPAMENTO']; ?> </td>
97                 <td> <?php echo $row['ORIGEM']; ?> </td>
98             </tr>
99         <?php } ?>
100         <?php mysqli_close($link); ?>
101     </tbody>
102 </table>
103 </div>
104 </body>
105 </html>

```

Listing A.2: Código do arquivo index.php

## A.3 Exportar.php

```

1 <?php
2
3     header('Content-type: application/vnd-ms-excel'); //define o
4     formato excel
5     $filename="DadosExportados.xls"; //define o nome do arquivo
6     header("Content-Disposition:attachment;filename=\" $filename
7     \")
8 >?>
9
10 <!REPETE O CODIGO REFERENTE A TABELA DO ARQUIVO index.php>
11 <table class="table table-striped table-dark">
12     <thead>
13         <tr>
14             <th style="border:2px solid black;">LEITURA</th>
15             <th style = "border:2px solid black;">EQUIPAMENTO</th>
16             <th style = "border:2px solid black;">ORIGEM</th>
17         </tr>
18     </thead>
19 <tbody>

```

```
20 <?php
21     $link = mysqli_connect("localhost", "root", "", "");
22     mysqli_select_db($link, 'tcc');
23     $query = "SELECT * FROM lista_equip";
24     $result = mysqli_query($link,$query);
25     while($row = mysqli_fetch_array($result)){
26     ?>
27
28     <tr>
29         <td style="border:2px solid black;" > <?php echo $row['
30             LEITURA']; ?> </td>
31         <td style="border:2px solid black;" > <?php echo $row['
32             EQUIPAMENTO']; ?> </td>
33         <td style="border:2px solid black;" > <?php echo $row['
34             ORIGEM']; ?> </td>
35     </tr>
36
37     <?php } ?>
38     <?php mysqli_close($link); ?>
39
40 </tbody>
41 </table>
```

Listing A.3: Código do arquivo Exportar.php

# Apêndice B

## Código do Microcontrolador

```
1 //BIBLIOTECAS
2 #include <WiFi.h>
3 #include <LiquidCrystal_I2C.h>
4 #include <SPI.h>
5 #include <PN532_SPI.h>
6 #include <PN532.h>
7 #include <NfcAdapter.h>
8 #include "SPI.h"
9 #include "PN532_SPI.h"
10 #include "snep.h"
11 #include "NdefMessage.h"
12 //BIBLIOTECAS
13
14 //CONFIG BOTOES
15 #define BUTTON1_PIN 25
16 #define BUTTON2_PIN 26
17 //CONFIG BOTOES
18
19 //CONFIG LCD-----
20 int lcdColumns = 16;
21 int lcdRows = 2;
22 LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
23 //CONFIG LCD-----
24
25 //CONFIG NFC-----
26 PN532_SPI pn532spi(SPI, SS);
27 NfcAdapter nfc = NfcAdapter(pn532spi);
28 //CONFIG NFC-----
29
30 //CONFIG WIFI-----
31 const char* ssid = "*****"; /*ocultado por segurança*
32 const char* pwd = "*****"; /*ocultado por segurança*
33 IPAddress server(192,168,1,3);
34 WiFiClient client;
35 //CONFIG WIFI-----
36
37 //VARIABLES-----
```

```

38 int lastState = LOW;
39 int currentState = 0;
40 int lastState2 = LOW;
41 int currentState2 = 0;
42 int opcao = 0;
43 int enviar = 0;
44 int novaleitura = 1;
45 int novaescrita = 1;
46 int i = 0;
47 String payloadAsString[2];
48 String equip;
49 String cidad;
50 //VARIAVEIS -----
51
52 void setup() {
53     Serial.begin(115200); //INICIA O MONITOR SERIAL
54     pinMode(BUTTON1_PIN, INPUT_PULLUP); //DEFINE MODO DO BOTAO
        COMO PULLUP
55     pinMode(BUTTON2_PIN, INPUT_PULLUP); //DEFINE MODO DO BOTAO
        COMO PULLUP
56     nfc.begin(); //INICIA O MODULO NFC
57     lcd.init(); //INICIA O LCD
58     lcd.backlight(); //LIGA ALUZA DE FUNDO DO LCD
59     WiFi.begin(ssid, pwd); //INICIA O WIFI
60
61     //TELA DE CONEXAO DO WIFI
62     lcd.setCursor(0,0);
63     lcd.print("Conectando");
64
65     while (WiFi.status() != WL_CONNECTED){//LOOP DE
        CONEXAO DO WIFI
66         delay(500);
67         lcd.setCursor((10+i),0);
68         lcd.print(".");
69         i++;
70     }
71     lcd.clear();
72     lcd.setCursor(3,0);
73     lcd.print("Conectado!");
74     delay(1500);
75     //TELA DE CONEXAO DO WIFI
76
77     while(true){//LOOP DE SELECAO DE MODO
78
79         //TELA DE SELECAO DE MODO
80         lcd.setCursor(0, 0);
81         lcd.print("Selecione o modo");
82         currentState = digitalRead(BUTTON1_PIN);
83         currentState2 = digitalRead(BUTTON2_PIN);
84         //DETECTA TOQUE NO BOTAO DE CONFIRMACAO
85         if((lastState2 == LOW && currentState2 == HIGH)|| (

```

```

86         lastState2 == HIGH && currentState2 == LOW))
87         //DETECTA TOQUE NO BOTAO DE CONFIRMACAO
88         break;
89         //DETECTA TOQUE NO BOTAO DE SELECAO
90         if(lastState == LOW && currentState == HIGH){
91         //DETECTA TOQUE NO BOTAO DE SELECAO
92         lcd.setCursor(4,1);
93         lcd.print("LEITURA");
94         opcao = 0;//VARIABEL DE SELECAO DE MODO
95     }
96     else{
97         lcd.setCursor(4,1);
98         lcd.print("ESCRITA");
99         opcao = 1;//VARIABEL DE SELECAO DE MODO
100     }
101     //TELA DE SELECAO DE MODO
102 }
103
104 void loop(){//LOOP PRINCIPAL
105     currentState2 = 0;
106     lastState2 = LOW;
107     lastState = LOW;
108     currentState = 0;
109     lcd.clear();
110
111     //MODO LEITURA
112     if (opcao==0){
113         lcd.setCursor(4,0);
114         lcd.print("LEITURA");
115         while(true){
116             String payloadAsString[2] = {"",""};
117             if (novaleitura==0)
118                 break; //ENCERRA O LOOP DE LEITURA CASO A VARIABEL
119                 MUDE PARA 0
120             lcd.clear();
121             lcd.setCursor(4,0);
122             lcd.print("LEITURA");
123             lcd.setCursor(2,1);
124             if (nfc.tagPresent()){ //DETECTA SE HA UMA TAG PROXIMA
125                 NfcTag tag = nfc.read(); //LE A TAG
126                 if (tag.hasNdefMessage()) { //DETECTA SE HA
127                     MENSAGEM NA TAG
128                     NdefMessage message = tag.getNdefMessage(); //
129                     SALVA A MENSAGEM NA VARIABEL MESSAGE
130                     if (message.getRecordCount() != 1) { //
131                         VERIFICA SE H MAIS DE UMA INFORMACAO NA
132                         MENSAGEM
133                     }
134                 }
135                 int recordCount = message.getRecordCount();
136                 for (int i = 0; i < 2; i++){ //PERCORRE AS

```

```

131     INFORMACOES DA MENSAGEM
132     NdefRecord record = message.getRecord(i);
133     int payloadLength = record.
134     getPayloadLength();
135     byte payload[payloadLength];
136     record.getPayload(payload);
137     for (int c = 0; c < payloadLength; c++) {
138         payloadAsString[i] += (char)payload[c
139         ]; //SALVA A INFORMACAO[i] NA
140         VARIAVEL payloadAsString[i]
141     }
142     payloadAsString[i].remove(0,3);
143     lcd.clear();
144     lcd.setCursor(0,0);
145     lcd.print(payloadAsString[i]);
146     delay(1000);
147 }
148 String equip = payloadAsString[0]; //SALVA A
149 payloadAsString[0] NA VARIAVEL equip
150 String cidade = payloadAsString[1]; //SALVA A
151 payloadAsString[1] NA VARIAVEL cidade
152 lcd.clear();
153 while(true){
154     currentState = digitalRead(BUTTON1_PIN);
155     currentState2 = digitalRead(BUTTON2_PIN);
156     //DETECTA TOQUE NO BOTAO DE CONFIRMACAO
157     if((lastState2 == LOW && currentState2 ==
158     HIGH)|| (lastState2 == HIGH &&
159     currentState2 == LOW))
160     //DETECTA TOQUE NO BOTAO DE CONFIRMACAO
161     break;
162     lcd.setCursor(0,0);
163     lcd.print(equip);
164     //DETECTA TOQUE NO BOTAO DE SELECAO
165     if(lastState == LOW && currentState ==
166     HIGH){
167     //DETECTA TOQUE NO BOTAO DE SELECAO
168     lcd.setCursor(0,1);
169     lcd.print("Enviar? SIM");
170     enviar = 1; //VARIAVEL DE DEFINICAO DE
171     ENVIO
172     }
173     else{
174     lcd.setCursor(0,1);
175     lcd.print("Enviar? NAO");
176     enviar = 0; //VARIAVEL DE DEFINICAO DE
177     ENVIO
178     }
179     delay(100);
180 }
181 currentState2 = 0;

```

```

171     lastState2 = LOW;
172     lcd.clear();
173     delay(300);
174
175     if (enviar==1){
176         if(client.connect(server, 80)) {
177             //ENVIA A REQUISICAO PARA O BANCO DE
178             DADOS
179             client.print("GET /esptest/ESP-SQL.php
180             ?EQUIPAMENTO="+equip+"&ORIGEM="+
181             cidad+" HTTP/1.1\r\n");
182             //ENVIA A REQUISICAO PARA O BANCO DE
183             DADOS
184             client.println("Host: 192.168.1.3:80")
185             ;
186             client.println("Connection: close");
187             client.println();
188         }
189     }
190     while(true){
191         //DETECTA TOQUE NO BOTAO DE CONFIRMACAO
192         if((lastState2 == LOW && currentState2 ==
193         HIGH)||(lastState2 == HIGH &&
194         currentState2 == LOW))
195         //DETECTA TOQUE NO BOTAO DE CONFIRMACAO
196         break;
197         lcd.setCursor(0,0);
198         lcd.print("Nova leitura?");
199         //DETECTA TOQUE NO BOTAO DE SELECAO
200         if(lastState == LOW && currentState ==
201         HIGH){
202         //DETECTA TOQUE NO BOTAO DE SELECAO
203             lcd.setCursor(6,1);
204             lcd.print("SIM");
205             novaleitura = 1; //VARIABEL DE
206             DEFINICAO DE NOVA LEITURA
207         }
208         else{
209             lcd.setCursor(6,1);
210             lcd.print("NAO");
211             novaleitura = 0; //VARIABEL DE
212             DEFINICAO DE NOVA LEITURA
213         }
214         delay(300);
215         currentState = digitalRead(BUTTON1_PIN);
216         currentState2 = digitalRead(BUTTON2_PIN);
217     }
218 }
219 }
220 }
221 }

```

```

212 //MODO LEITURA
213
214 //MODO ESCRITA
215 else if (opcao==1){
216     lcd.setCursor(4,0);
217     lcd.print("ESCRITA");
218
219     while (true) {
220         if (novaescrita==0)
221             break; //ENCERRA O LOOP DE ESCRITA CASO A VARIAVEL
                MUDE PARA 0
222         lcd.clear();
223         lcd.setCursor(4,0);
224         lcd.print("ESCRITA");
225         if (nfc.tagPresent()) { //DETECTA SE HA UMA TAG
                PROXIMA
226             NdefMessage message = NdefMessage();
227             message.addTextRecord("Equipamento-A"); //ESCREVE
                A PRIMEIRA INFORMACAO NA TAG
228             message.addTextRecord("Itauna"); //ESCREVE A
                SEGUNDA INFORMACAO NA TAG
229             bool success = nfc.write(message);
230             if (success) {
231                 lcd.setCursor(0,1);
232                 lcd.print("Equipamento-A"); //MOSTRA A
                INFORMACAO GRAVADA
233             } else {
234                 lcd.setCursor(0,1);
235                 lcd.print("Falha na Escrita"); //INFORMA NO
                LCD CASO A ESCRITA TENHA FALHADO
236             }
237             delay(1000);
238             currentState2 = LOW;
239             lastState2 = LOW;
240             delay(300);
241             lcd.clear();
242             while(true){
243                 //DETECTA TOQUE NO BOTAO DE CONFIRMACAO
244                 if((lastState2 == LOW && currentState2 == HIGH
                    )||(lastState2 == HIGH && currentState2 ==
                    LOW))
245                     //DETECTA TOQUE NO BOTAO DE CONFIRMACAO
246                     break;
247                 lcd.setCursor(0,0);
248                 lcd.print("Nova escrita?");
249                 //DETECTA TOQUE NO BOTAO DE SELECAO
250                 if(lastState == LOW && currentState == HIGH){
251                     //DETECTA TOQUE NO BOTAO DE SELECAO
252                     lcd.setCursor(6,1);
253                     lcd.print("SIM");
254                     novaescrita = 1; //VARIAVEL DE DEFINICAO

```

```

255         DE NOVA ESCRITA
256     }
257     else{
258         lcd.setCursor(6,1);
259         lcd.print("NAO");
260         novaescrita = 0; //VARIABEL DE DEFINICAO
261         DE NOVA ESCRITA
262     }
263     delay(300);
264     currentState = digitalRead(BUTTON1_PIN);
265     currentState2 = digitalRead(BUTTON2_PIN);
266 }
267 }
268 //MODO ESCRITA
269
270 //TELA FINAL
271 if ((novaleitura==0)|| (novaescrita==0)){
272     lcd.clear();
273     while(true){
274         lcd.setCursor(5,0);
275         lcd.print("Fim do");
276         lcd.setCursor(2,1);
277         lcd.print("procedimento");
278     }
279 }
280 //TELA FINAL
281 }

```

Listing B.1: Código do microcontrolador

# Referências

- BORGES, T. C.; CAMPOS, M. S.; BORGES, E. C. Implantação de um sistema para o controle de estoques em uma gráfica/editora de uma universidade. *Revista Eletrônica Produção Engenharia*, [S.l.], p.237, 2010.
- CAMPOS, R. S.; SIMON, A. T. Benefícios da otimização do estoque de peças de reposição em conjunto com as operações de manutenção. *Uninove*, [S.l.], p.64, 2019.
- CONSTANTINO, E. d. O.; NERIS, V. P. d. A. Acessibilidade em Bibliotecas: utilizando nfc e smartphones para auxiliar deficientes visuais na identificação de livros. *Tecnologias, Infraestrutura e Software*, [S.l.], 2014.
- CORTELETTI, D. *Evolução de Microeletrônica a Micro-Sistemas*. [Online; acesso em 11-Julho-2021], <http://sbprt.ibict.br/dossie-tecnico/downloadsDT/NTE=>.
- DESAI, E.; SHAJAN, M. G. A Review on the Operating Modes of Near Field Communication. *International Journal of Engineering and Advanced Technology (IJEAT)*, [S.l.], p.322–325, 2012.
- FELIPEFLOP. [Online; acesso em 14-Outubro-2020], <https://www.filipeflop.com/blog/qual-modulo-wifi-esp32-e-ideal-para-meu-projeto/esp32-basico-pinout/>.
- FELIPEFLOP. *NDEF message structure, with details about the bytes of the header*. [Online; acesso em 09-Agosto-2021], <https://www.filipeflop.com/produto/kit-modulo-leitor-rfid-nfc-pn532/>.
- FELIPEFLOP. [Online; acesso em 09-Agosto-2021], <https://www.filipeflop.com/produto/tag-etiqueta-adesiva-nfc-1356mhz-x10-unidades/>.
- FELIPEFLOP. [Online; acesso em 09-Agosto-2021], <https://www.filipeflop.com/produto/display-lcd-16x2-i2c-backlight-azul/>.
- FELIPEFLOP. [Online; acesso em 09-Agosto-2021], <https://www.filipeflop.com/produto/bateria-li-ion-18650-2200mah/>.

- FELIPEFLOP. [Online; acesso em 09-Agosto-2021], <https://www.filipeflop.com/produto/conversor-boost-dc-step-up/>.
- FELIPEFLOP. [Online; acesso em 09-Agosto-2021], <https://www.filipeflop.com/produto/modulo-carregador-de-bateria-de-litio-tp4056/>.
- FERNANDES, L. M. C. Sistema de Identificação de Objectos e Pessoas para Bancada Laboratorial. , [S.l.], 2015.
- FERNEDA, E. *Introdução à Linguagem SQL*. [Online; acesso em 31-Janeiro-2022], <https://sites.ffclrp.usp.br/cid/docentes/edberto/Apostilas/Apostila%20SQL.pdf>.
- FILHO, O. L. d. S. F.; CAVALCANTE, S. V. Comunicação NFC (Near Field Communication) entre Dispositivos Ativos. , [S.l.], 2010.
- G., A. *O que é CSS? Guia Básico para Iniciantes*. [Online; acesso em 14-Agosto-2021], <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css/>.
- GROUP, T. P. *PHP Manual*. [Online; acesso em 31-Janeiro-2022], <https://www.php.net/manual/en/preface.php>.
- KURNIAWAN, A. *Internet of Things Projects with ESP32: build exciting and powerful iot projects using the all-new espressif esp32*. 1.ed. Birmingham, Mumbai: Packt, 2019.
- LIMA, C. B. de; VILLACA, M. V. M. *AVR e Arduino: técnicas de projeto*. 1.ed. Florianópolis: [s.n.], 2012.
- LUGLI, A. B.; SOBRINHO, D. G. TECNOLOGIAS WIRELESS PARA AUTOMAÇÃO INDUSTRIAL: wireless\_hart, bluetooth, wisa, wi-fi, zigbee e sp-100. , [S.l.], 2012.
- MEDEIROS, M. T. d.; SARTORI, H. N. PetID: protótipo de um aplicativo para identificação de animais utilizando nfc. , [S.l.], 2018.
- MICROCHIP. [Online; acesso em 12-Julho-2021], <https://www.microchip.com/wwwproducts/en/PIC16F15244>.
- MOTLAGH, N. H. *NEAR FIELD COMMUNICATION (NFC) A TECHNICAL OVERVIEW*. 2012. Dissertação (Mestrado em Ciência da Computação) — UNIVERSITY OF VAASA, Finlândia.
- ORGANIZATION, N. F. C. *About Near Field Communication*. [Online; acesso em 18-Julho-2021], <http://nearfieldcommunication.org/about-nfc.html>.

- O'REILLYMEDIA. *NDEF message structure, with details about the bytes of the header*. [Online; acesso em 03-Fevereiro-2022], <https://www.oreilly.com/library/view/beginning-nfc/9781449324094/ch04.html>.
- RAMANATHAN, R.; IMTIAZ, J. NFC in industrial applications for monitoring plant information. In: FOURTH INTERNATIONAL CONFERENCE ON COMPUTING, COMMUNICATIONS AND NETWORKING TECHNOLOGIES (ICCCNT), 2013. *Anais...* [S.l.: s.n.], 2013. p.1-4.
- ROBERTI, M. *A História da Tecnologia RFID*. [Online; acesso em 02-Outubro-2020], <https://www.rfidjournal.com/the-history-of-rfid-technology>.
- ROBOCORE. [Online; acesso em 12-Julho-2021], <https://www.robocore.net/loja/iot/esp32-wifi-bluetooth>.
- SALLINEN, M.; STROMMER, E.; YLISAUKKO-OJA, A. Application Scenario for NFC: mobile tool for industrial worker. In: SECOND INTERNATIONAL CONFERENCE ON SENSOR TECHNOLOGIES AND APPLICATIONS (SENSORCOMM 2008), 2008. *Anais...* [S.l.: s.n.], 2008. p.586-591.
- SOUZA, V. A. *A história e as diferenças entre um microcontrolador e um microprocessador*. [Online; acesso em 11-Julho-2021], [https://www.cerne-tec.com.br/Artigo3\\_Historia.pdf](https://www.cerne-tec.com.br/Artigo3_Historia.pdf).
- TENERGY. [Online; acesso em 19-Agosto-2021], [https://www.tenergy.com/30005\\_datasheet.pdf](https://www.tenergy.com/30005_datasheet.pdf).
- URYUTIN, O. *A brief history of web app*. [Online; acesso em 24-Julho-2021], <https://aplextor.medium.com/a-brief-history-of-web-app-50d188f30d>.
- XAVIER, T. *O que é HTML e qual sua funcionalidade*. [Online; acesso em 14-Agosto-2021], <https://rockcontent.com/br/blog/html/>.