

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

Eduardo Vasconcelos Marques

Projeto de um Dispositivo de Identificação de Pacientes em Ambientes Hospitalares



Divinópolis  
2021

Eduardo Vasconcelos Marques

Projeto de um Dispositivo de Identificação de Pacientes em Ambientes Hospitalares

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  
Co-orientador: Prof. Raulivan Rodrigo da Silva

Divinópolis  
2021

Eduardo Vasconcelos Marques

Projeto de um Dispositivo de Identificação de Pacientes em Ambientes Hospitalares

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

Prof. Dr. Lucas Silva de Oliveira  
CEFET-MG / *Campus* Divinópolis

Divinópolis  
2021

Dedico este trabalho à minha família e amigos que sempre me apoiaram durante toda trajetória acadêmica.

# Agradecimentos

Agradeço,

- Primeiramente a Deus por sempre estar presente em minha vida com suas bênçãos diárias.
- Aos meu pais, Artur e Adriana e minha irmã Juliana por todo incentivo, amor e esforços realizados para ajudar na conclusão de mais essa etapa em minha vida. Vocês são peças essenciais para mim, amo vocês!
- Ao meu avô Geraldo, a pessoa com o maior coração que eu conheço, e minhas avós Nininha e Ignêz, que sempre estão com um sorriso estampado no rosto mostrando uma alegria de viver incomparável. Ao Juliano, Rosângela, João Pedro e Bruno por todos os momentos de descontração, risadas e aprendizado.
- A Júlia que sempre acreditou na minha capacidade e acompanhou de perto todo o desenvolvimento deste trabalho, além de sempre estar ao meu lado quando precisei.
- Aos meus amigos e companheiros de curso, em especial a Turma 8, por toda convivência diária sempre com muita persistência, companheirismo e principalmente alegria, mesmos nos momentos mais difíceis do curso.
- Ao pessoal do estágio, em especial ao Ricardo pelos enormes ensinamentos passados e conselhos sobre a vida corporativa e também a Lorena, Fábio e Natália por me auxiliarem a amadurecer como profissional.
- Aos professores e funcionários do CEFET-MG Campus Divinópolis, pela extrema dedicação, colaboração e atenção.
- Enfim, a todos os que por algum motivo contribuíram em minha trajetória acadêmica.

*Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes.*

Marthin Luther King

# Resumo

A identificação de pacientes é um processo bastante crítico no âmbito hospitalar. Segundo o *The United Kingdom National Patient Safety Agency*, entre 2003 e 2006, foram registrados 236 acidentes relacionados à má relação de informações em pulseiras de identificação e, esses erros foram citados em mais de 100 pesquisas como causa raiz de óbitos. Partindo da necessidade de deixar o processo de identificação de pacientes mais seguro, o objetivo deste trabalho é o desenvolvimento de um dispositivo capaz de identificar um paciente tanto pela impressão digital quanto por uma *TAG RFID*, gerando assim uma maior precisão no processo. Feita a identificação por um dos métodos citados acima, o dispositivo exibirá informações sobre o paciente em um *display* LCD presente no próprio dispositivo e ou em um dispositivo Android equipado com um aplicativo que também será desenvolvido.

Palavras-chave: Identificação de Pacientes, Dispositivo móvel, Aplicativo Android.

# Abstract

The identification of patients is a very critical process in the hospital environment. According to The United Kingdom National Patient Safety Agency, between 2003 and 2006, 236 accidents were recorded related to poor information on wristbands and these errors were cited in more than 100 surveys as the root cause of deaths. Starting from the need to make the patient identification process safer, the objective of this work is the development of a device capable of identifying a patient both by fingerprint and by an RFID TAG, thus generating greater precision in the process. Once identified by one of the methods mentioned above, the device will display information about the patient on an LCD display present on the device itself or on an Android device equipped with an application that will also be developed.

Key-words: Patient Identification, Mobile Device, Android App.



# 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 . . . . .	2
1.2 Motivação . . . . .	2
1.3 Objetivos do trabalho . . . . .	2
1.3.1 Objetivos Gerais . . . . .	2
1.3.2 Objetivos Específicos . . . . .	2
1.4 Estado da Arte . . . . .	3
1.5 Metodologia . . . . .	4
1.6 Organização do texto . . . . .	4
<b>2 Fundamentos</b>	<b>5</b>
2.1 Revisão Bibliográfica . . . . .	5
2.2 Fundamentação Teórica . . . . .	11
2.2.1 Normas Hospitalares . . . . .	11
2.2.2 Materiais . . . . .	12
2.2.3 ESP32 . . . . .	13
2.2.4 RFID . . . . .	13
2.2.5 Biometria . . . . .	15
2.2.6 Firebase . . . . .	18
2.2.7 API . . . . .	18
2.2.8 Node.js . . . . .	19
2.2.9 Flutter . . . . .	20
2.2.10 Protocolo I2C . . . . .	20
<b>3 Desenvolvimento e Resultados</b>	<b>22</b>
3.1 Materiais e Métodos . . . . .	22
3.1.1 Eletrônica . . . . .	22
3.1.2 Estrutura . . . . .	28
3.1.3 Softwares . . . . .	28
3.2 Comunicação sem fio . . . . .	29

---

3.3	Base de dados . . . . .	30
3.3.1	Firebase . . . . .	30
3.3.2	REST API . . . . .	30
3.4	O Aplicativo . . . . .	32
3.4.1	Características do Aplicativo . . . . .	32
3.4.2	Desenvolvimento do Aplicativo . . . . .	33
3.5	O Dispositivo . . . . .	36
3.5.1	Desenvolvimento . . . . .	36
3.5.2	Circuito de Alimentação . . . . .	38
3.5.3	Esquemático do Dispositivo . . . . .	39
3.5.4	Projeto da Case . . . . .	39
3.6	Procedimentos de Uso . . . . .	40
3.7	Custos do Projeto . . . . .	42
<b>4</b>	<b>Considerações Finais</b>	<b>43</b>
4.1	Conclusões . . . . .	43
4.2	Propostas de Continuidade . . . . .	44
<b>A</b>	<b>Código da REST API</b>	<b>45</b>
<b>B</b>	<b>Código do Aplicativo</b>	<b>47</b>
<b>C</b>	<b>Telas do Aplicativo</b>	<b>48</b>
<b>D</b>	<b>Código do Microcontrolador</b>	<b>50</b>
<b>E</b>	<b>Telas do Dispositivo</b>	<b>58</b>
	<b>Referências</b>	<b>60</b>

# Lista de Figuras

2.1	Arduíno Uno. Fonte: (ARDUINOPORTUGAL, 2017)	6
2.2	ESP32. Fonte: (ROBOCORE, 2020)	7
2.3	Arquitetura da rede <i>Wi-Fi</i> . Fonte: (REIS, 2018)	9
2.4	Princípio do funcionamento do RFID. Fonte: (CUNHA, 2016)	10
2.5	Pinagem ESP32. Fonte: (ZIMMERMANN, 2018)	13
2.6	Princípio do funcionamento do RFID. Adaptado de: (HADDAD; RIZZOTTO; MALDONADO, 2016)	14
2.7	Configuração interna de um tag RFID passiva. Fonte: (NOGUEIRA, 2019)	14
2.8	Cristas e vales em uma impressão digital. Fonte: (SILVA, 2015)	16
2.9	Tipos de minúcias encontradas em uma impressão digital. Adaptado de: (MALTONI <i>et al.</i> , 2009)	16
2.10	Diagrama de funcionamento de um sensor de impressões digitais. Fonte: (MALTONI <i>et al.</i> , 2009)	17
2.11	Diagrama de um sensor óptico. Fonte: (MALTONI <i>et al.</i> , 2009)	18
2.12	Topologia do protocolo I2C. Fonte: HOA (2020)	21
3.1	Fluxograma de funcionamento de dispositivo	23
3.2	FPM10A DY50. Fonte: VLADCONTROL	24
3.3	MFRC522. Fonte: BANGGOOD	24
3.4	TAG de papel. Adaptado de: YARONGTECH	25
3.5	Pulseira com TAG inclusa. Adaptado de: OASES	25
3.6	Display 20x4. Fonte: SAINSMART	26
3.7	Bateria 5V. Fonte: ALIBABA	27
3.8	Módulo de carga TP5046. Fonte: MERCADO LIVRE	27
3.9	Diagrama da comunicação sem fio.	30
3.10	Diagrama do banco de dados no Firebase.	31
3.11	Fluxograma de funcionamento do aplicativo.	32
3.12	Esboço das telas do aplicativo	33
3.13	Tela de login do aplicativo.	34
3.14	Tela do menu do aplicativo.	34
3.15	Tela receber dados do aplicativo.	35
3.16	Tela pesquisar dados do aplicativo.	36
3.17	Tela de exibição de dados do aplicativo.	36
3.18	Foto das conexões	38
3.19	Modo de economia de energia	39
3.20	Circuito de alimentação do dispositivo.	39

3.21	Esquemático final do dispositivo. . . . .	40
3.22	Desenho da case com as dimensões. . . . .	41
3.23	Desenho interno da case com a posição dos componentes. . . . .	41
C.1	Telas referentes ao login . . . . .	48
C.2	Tela do menu . . . . .	48
C.3	Telas referentes ao recebimento de dados . . . . .	49
C.4	Telas referentes a pesquisa de pacientes . . . . .	49
C.5	Telas referentes a exibição dos dados do pacientes . . . . .	49
E.1	Conectando ao wi-fi . . . . .	58
E.2	Menu . . . . .	58
E.3	Esperando leitura . . . . .	58
E.4	Exibição dos dados . . . . .	59
E.5	Paciente não encontrado . . . . .	59
E.6	Modo de economia de energia . . . . .	59

# Lista de Tabelas

3.1	Tabela de corrente por componente. . . . .	26
3.2	Tabela de conexão do ESP-32 com os componentes. . . . .	37
3.3	Tabela de custos (orçamento). . . . .	42

# Lista de Acrônimos e Notação

3D	Três Dimensões
ABNT	Associação Brasileira de Normas Técnicas
AFIS	Sistema de Identificação Automatizada de Impressões Digitais, do inglês <i>Automated Fingerprint Identification System</i>
ANVISA	Agência Nacional de Vigilância Sanitária
API	Interface de Programação de Aplicação, do inglês <i>Application Programming Interface</i>
CCD	Dispositivo de carga acoplada, do inglês <i>Charge-Coupled Device</i>
CMOS	Semicondutor de Metal-Óxido Complementar, do inglês <i>Complementary Metal-Oxide Semiconductor</i>
DSP	Processador Digital de Sinal, do inglês <i>Digital Signal Processor</i>
FBI	Departamento Federal de Investigação dos Estados Unidos, do inglês <i>Federal Bureau of Investigation</i>
GPIO	Entrada/Saída de Uso Geral, do inglês <i>General Purpose Input/Output</i>
HTTPS	Protocolo de Transferência de Hipertexto Seguro, do inglês <i>Hyper Text Transfer Protocol Secure</i>
I2C	Circuito Inter-integrado, do inglês <i>Inter-Integrated Circuit</i>
IFF	Identificação amigo ou inimigo, do inglês <i>Identification Friend or Foe</i>
IoT	Internet das Coisas, do inglês <i>Internet of Things</i>
ISM	Industrial, Científico e Médico, do inglês <i>Industrial, Scientific and Medical</i>
LED	Diodo Emissor de Luz, do inglês <i>Light-Emitting Diode</i>
LiPo	Polímero de Lítio, do inglês <i>Lithium Polymer</i>
MISO	Entrada do Mestre e Saída do Escravo, do inglês <i>Master Input Slave Output</i>
MOSI	Saída do Mestre e Entrada do Escravo, do inglês <i>Master Output Slave Input</i>
NPSA	Agência Nacional de Segurança do Paciente, do inglês <i>National Patient Safety Agency</i>
PNSP	Programa Nacional de Segurança do Paciente
RFID	Identificação por Rádio Frequência, do inglês <i>Radio-Frequency Identification</i>
RST	Reset
SCL	Serial Clock
SDA	Serial Data
SGBD	Sistema de Gerenciamento de Banco de Dados
URL	Localizador Uniforme de Recursos, do inglês <i>Uniform Resource Locator</i>
USP	Universidade de São Paulo

## Introdução

O processo de identificação de pacientes é o ponto de partida para execução de outras atividades e é aplicado em todos os ambientes de prestação do cuidado de saúde como, por exemplo, unidades de internação, ambulatório, salas de emergência e centros cirúrgicos. Tal procedimento deve garantir a segurança sendo capaz de fornecer, no mínimo, duas informações relevantes para o reconhecimento a quem se refere o tratamento como doença, medicamentos, tipo sanguíneo, etc, e é assegurado por um protocolo formulado pela Agência Nacional de Vigilância Sanitária (ANVISA, 2013).

Apesar disso, os erros podem aparecer e são potencializados por fatores como mudanças de leito e, até mesmo, o estado de consciência do paciente. De acordo com uma pesquisa realizada entre janeiro de 2013 e agosto de 2015 pelo Instituto de Pesquisa de Pronto Atendimento<sup>1</sup> (ECRI, 2016), foram identificados 7.613 casos de erros relacionados a má identificação de pacientes em 181 instituições médicas monitoradas em diferentes países. Dentre essas ocorrências, mais de um terço tiveram como causa confusões no registro do diagnóstico e 22% na realização de tratamentos e procedimentos.

Para tentar minimizar a ocorrência desses erros e garantir o atendimento correto ao paciente, grande parte dos serviços de saúde utilizam como ferramenta as pulseiras de identificação. Esse método de reconhecimento tem como norma, o regulamento da Agência Nacional de Segurança do Paciente do Reino Unido<sup>2</sup> (NPSA, 2005), que estipula diversos requisitos tais como: tamanho adequado (recém-nascidos, obesos, condições clínicas), conforto (material anti-alérgico, flexível, impermeável e lavável), durabilidade (fácil limpeza, resistente a fluidos corporais, sabões e soluções alcoólicas), técnica de impressão e aplicabilidade (gerada e impressa por meio eletrônico, fácil leitura, espaço suficiente para gravar identificadores completos e inequívocos), cor (branca), letra (preta), identificadores/qualificadores (nome completo: sobrenome em primeiro lugar com letra maiúscula, data de nascimento e número de registro no sistema nacional de saúde).

<sup>1</sup>Do inglês Emergency Care Research Institute.

<sup>2</sup>Do inglês The United Kingdom National Patient Safety Agency.

Devido a essa grande quantidade de fatores críticos e de alto risco no processo de identificação de pacientes, percebe-se a necessidade de incorporação de novas tecnologias, produtos e práticas de gestão nesse processo (NETO, 2011). Pensando nisso, este trabalho tem como proposta, o projeto de um dispositivo de identificação de pacientes por meio de impressão digital e também por uma *TAG RFID (Radio-Frequency Identification)*.

## 1.1 Definição do problema

Como descrito na seção anterior, por ser realizada de forma manual e por conter diversos pontos críticos, a identificação de pacientes em leitos hospitalares está submetida a possíveis erros que podem inclusive levar o paciente a óbito. Sabendo disso, percebeu-se a necessidade de aprimorar a execução desta identificação e observou-se que a modernização deste processo tende a solucionar grande parte dos problemas ocasionados por falhas no mesmo.

## 1.2 Motivação

Este trabalho tem como motivação o aumento da precisão na identificação de pacientes em leitos hospitalares. Esse processo é essencial para a realização de todos os procedimentos de forma correta e com isso, evitar as consequências geradas por falhas de reconhecimento.

Com o desenvolvimento de um dispositivo para tal fim, além de minimizar o problema com a identificação, o operador do aparelho teria acesso a todos os dados e histórico do paciente, facilitando e agilizando o processo de atendimento.

## 1.3 Objetivos do trabalho

São objetivos do trabalho a ser desenvolvidos:

### 1.3.1 Objetivos Gerais

Este trabalho tem como objetivo projetar um dispositivo de identificação de pacientes em ambientes hospitalares, onde, feita a identificação, as informações do paciente serão exibidas para a pessoa de posse do dispositivo em um *display* LCD e em um aplicativo Android.

### 1.3.2 Objetivos Específicos

- Estudar e compreender o processo e as normas para identificação de pacientes;
- Pesquisar e escolher os componentes adequados para utilização em ambiente hospitalar;
- Projetar a parte eletrônica em conjunto com os componentes;

- Dimensionar e desenvolver o modelo 3D da case do dispositivo;
- Desenvolver um aplicativo Android capaz de exibir os dados dos pacientes;
- Implementar a comunicação entre o dispositivo e o aplicativo Android.

## 1.4 Estado da Arte

Os estudos sobre a identificação de pacientes ganharam bastante notoriedade na última década. Segundo TASE *et al.* (2013), a temática desse tema se encontra “intrinsecamente relacionada à qualidade nos serviços de saúde e vem sendo amplamente referida e discutida pelos setores prestadores de serviços de saúde, pelas entidades de classe e pelos órgãos governamentais.” Esses autores desenvolveram um estudo que visava destacar os elementos presentes na identificação por meio de pulseiras e a partir disso, realizar uma reflexão sobre a implementação do processo nos hospitais.

Com esse estudo, TASE *et al.* (2013), concluíram que existem lacunas tangentes à implementação e monitoramento dos protocolos de identificação dos pacientes. A partir disso, os autores enxergaram uma oportunidade de evolução nesse processo.

No ano de 2015, foram realizados dois estudos com enfoque nas pulseiras de identificação. No primeiro deles, HOFFMEISTER; MOURA realizaram uma análise em um hospital universitário e perceberam que cerca de 4% dos pacientes do hospital estavam sem a pulseira e outros 12% estavam com a pulseira fora do padrão operacional. Os autores perceberam também que as unidades de internação merecem ser tratadas com maior atenção, pois os resultados obtidos nessa área estavam bem inferiores aos demais.

Já o segundo estudo foi realizado por TASE; TRONCHIN em unidades obstétricas. Os autores perceberam que “91% dos pacientes relataram ter condições de colaborar para evitar os erros que ocorrem nos hospitais, 84% sentiam-se confortáveis em solicitar ao enfermeiro a confirmação dos dados em sua pulseira.” Além disso, cerca de 10% dos pacientes não concordam em utilizar a pulseira durante a permanência no hospital mesmo com as mesmas apresentando indicadores para sua identificação.

Também em 2015, BRITO avaliou os processos de identificação nos serviços de saúde. A autora observou que 41,2% dos profissionais relataram a existência de situações que o paciente pode ficar sem nenhuma identificação, gerando assim uma grande margem para possíveis erros. Em sua pesquisa, outro fato de extrema relevância é que 50% dos enfermeiros entrevistados reconheceram a necessidade de um aprimoramento no processo de identificação.

Por fim, em 2019, NASSAR; VIEIRA, estudaram o uso da tecnologia *RFID* na rastreabilidade hospitalar com ênfase na gestão dos medicamentos a serem enviados para o paciente. Os autores entenderam que o uso do *RFID* representa mais do que um suporte para a realização do processo de medicação, compondo a mediação entre o hospital e os pacientes.

Para os autores, o uso desta tecnologia contribui com a gestão informativa do hospital e oferece agilidade e eficiência nos processos internos, assim, mostrando um enorme potencial de atuação nessa área de pesquisa.

## 1.5 Metodologia

A metodologia adotada na elaboração do trabalho foi pensada de forma com que a evolução dos resultados ocorresse de forma gradativa e para isso, ela foi dividida em quatro partes.

A primeira etapa se consistiu num estudo teórico a respeito do tema, onde foram levantadas as características necessárias que o trabalho devia ter, além de verificar a existência de projetos semelhantes e a evolução no processo de identificação de pacientes que o dispositivo causaria.

Feito isso, partiu-se para a definição dos componentes a serem utilizados no dispositivo e as funcionalidades que o aplicativo deveria possuir. Nessa etapa, foi definido também os *softwares* que seriam utilizados para o projeto.

Tendo todas as especificações do projeto, iniciou-se o desenvolvimento do aplicativo. Esta etapa teve início com a criação do *layout* de todas as páginas e em seguida, implementou-se todas as funcionalidades e a comunicação com a base de dados.

Por fim, a última parte foi a mais longa do trabalho e se consistiu na elaboração do dispositivo em si. O início dessa etapa se deu com a conexão dos componentes ao micro-controlador seguida pela programação do mesmo e comunicação com a base de dados. Após isso, projetou-se o circuito responsável pela alimentação do dispositivo e finalizou-se com a definição da posição de cada componente no projeto da case.

## 1.6 Organização do texto

Este documento é dividido em quatro capítulos.

O presente capítulo apresenta a definição do problema que foi motivação para a produção do trabalho. Além disso, apresenta também os objetivos gerais e específicos do mesmo, o estado da arte e a metodologia utilizada durante o desenvolvimento do projeto.

No segundo capítulo são apresentados os fundamentos do trabalho. Nessa parte, uma revisão bibliográfica é realizada, seguida pela fundamentação teórica de fatores importantes para o desenvolvimento do projeto.

Em seguida, no terceiro capítulo, o desenvolvimento e os resultados são apresentados. Nessa parte, todo o processo de projeto do dispositivo seguido pelos testes e validações são apresentados.

Por fim, o último capítulo apresenta as considerações finais do trabalho em conjunto com as propostas de continuidade.

## Fundamentos

O capítulo a seguir apresenta de forma sucinta uma revisão bibliográfica sobre os temas tratados no trabalho, mostrando a evolução tecnológica dos componentes utilizados e também o estado atual da arte. Após isso, conceitos necessários para o desenvolvimento do mesmo são apresentados.

### 2.1 Revisão Bibliográfica

A partir da evolução da microeletrônica e da capacidade de miniaturizar circuitos e implementar novas funcionalidades, surgiram os primeiros microprocessadores (LIMA; VILLACA, 2012).

O primeiro deles começou a ser projetado em 1969, com a necessidade de uma empresa japonesa, a BUSICOM, no seu projeto de uma calculadora eletrônica. Os engenheiros dessa empresa embarcaram para os Estados Unidos com o objetivo de encontrar Marcian Hoff, da *Intel Corporation*, para expor as necessidades do dispositivo. Tempos depois, em 1971, após comprar a licença da empresa japonesa a *Intel* lançou o 4004, um controlador de 4 bits que conseguia processar 6000 operações por segundo (6kHz) (SOUZA, 2006).

Nos anos seguintes, na mesma época que a *Intel* lançava o modelo 8008, um processador de 8 *bits* que conseguia processar até 300.000 operações por segundos (300kHz), a concorrência começou a surgir. A primeira empresa a disputar o mercado foi a Motorola, com o microprocessador 6800. Em seguida, a *MOS Technology* lançou dois modelos: o 6501 e o 6502. Devido ao preço consideravelmente mais baixo, o 6502 passou a ser o mais utilizado pelo mercado (SOUZA, 2006).

Com o avanço da tecnologia, nos anos 90, a utilização de memórias *flash* e de memórias eletricamente apagáveis e programáveis (EEPROM) tornaram-se disponíveis. Antes disso, para reprogramar um dispositivo, era necessário remover o processador do sistema e fazer a substituição. Segundo AYCOCK (2017) com a remoção dessa limitação “os microcontroladores puderam ser programados e reprogramados enquanto que em circuito, fazendo com que

seus dispositivos pudessem ser atualizados com novos *softwares* sem a necessidade de serem devolvidos ao fabricante.”

Além da facilidade de reprogramação, o uso desse tipo de memória trouxe uma maior confiabilidade aos dispositivos e gerou uma redução de tamanho e custos (TAVARES, 1997). Com esse fato, os microcontroladores começaram a ganhar destaque. Segundo LIMA; VILLACA(2012) a principal vantagem do uso de microcontroladores é a “possibilidade de desenvolvimento rápido de sistemas eletrônicos com o emprego de um pequeno número de componentes.”

Um grande exemplo dessa praticidade, é o Arduino. Sua história começou em 2005, na Itália, onde o professor Massimo Banzi queria ensinar programação aos seus alunos e teve uma grande dificuldade em encontrar uma placa didática, barata e com grande capacidade de processamento FERRONI *et al.* (2015).

Partindo dessa lacuna no mercado, o professor discutiu sua necessidade com o pesquisador sueco David Cuartielles e tiveram a ideia de desenvolver um microcontrolador de fácil uso, barato e que fosse potente o suficiente para atender suas demandas (EVANS; NOBLE; HOCHENBAUM, 2013).

Ainda segundo EVANS; NOBLE; HOCHENBAUM (2013), a popularidade do Arduíno cresceu rapidamente. Com a percepção do público da capacidade na utilização em projetos próprios e na introdução da programação de microcontroladores, o dispositivo alcançou a marca de 300 mil unidades vendidas até 2013. O modelo mais representativo, é o Arduino Uno, representado na Figura 2.1.



Figura 2.1: Arduíno Uno. Fonte: (ARDUINOPORTUGAL, 2017)

Atualmente, existem vários outros microcontroladores inspirados no Arduino. Um bom exemplo disso, é o ESP32 da *Espressif Systems*, mostrado na Figura 2.2. Segundo KURNI-

AWAN (2019), “esse dispositivo é um chip de baixo custo que consiste em um microcontrolador com protocolos de comunicação *Wi-Fi* e *Bluetooth*, o que torna possível, a utilização em conjunto com um aplicativo de Internet das Coisas (IoT).”<sup>1</sup>.

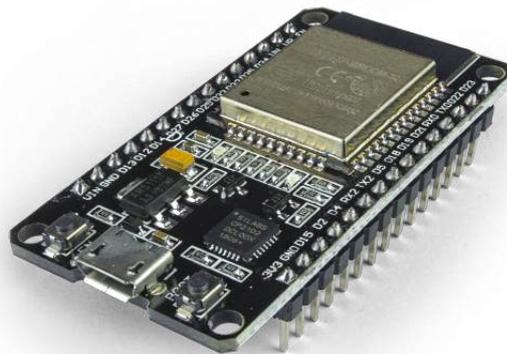


Figura 2.2: ESP32. Fonte: (ROBOCORE, 2020)

Segundo BARRA *et al.*(2018), os aplicativos móveis em geral “visam atender o acesso das pessoas à informação e ao conhecimento, sem restrição de tempo e espaço. A possibilidade da queda de barreiras de tempo e espaço permite também novas formas de comunicação.”

Esses aplicativos podem assumir várias categorizações a partir de suas funcionalidades. Para FLING (2009), essas categorias são:

- Utilitários: Aplicação mais básica. Relógios, previsão do tempo, etc;
- Localização: Aplicativos que auxiliam as pessoas a se localizarem. Mapas, GPS, etc;
- Informação: Possui o objetivo de prover informações ao usuário. Aplicativos de jornais, notícias, etc;
- Produtividade: Oferece maior eficiência ao usuário. Aplicativos de automação, otimizadores de tempo, etc;
- Tela Inteira (*Full Screen*): Aplicativos que prendem a atenção do usuário em vários momentos. Redes Sociais, jogos, etc.

Com o avanço tecnológico, os aplicativos de produtividade vem ganhando bastante destaque, principalmente devido a automação residencial que os mesmos podem oferecer. Para

---

<sup>1</sup>Do inglês Internet of Things.

conectar todo o sistema eletrônico da casa no aplicativo, alguns controladores são instalados e a comunicação entre eles ocorrem através de uma rede sem fio.

A ideia para esses protocolos de comunicação sem fio, surgiu em meados do século XIX, quando o pesquisador inglês Michael Faraday descobriu a indução eletromagnética. Após isso, com os contínuos estudos na área, uma série de invenções foram de extrema importância para chegarmos nas tecnologias da atualidade (LUGLI; SOBRINHO, 2012):

- 1888: Heinrich Rudolf Hertz produziu a primeira onda de rádio;
- 1901: Guglielmo Marconi inventou o telégrafo sem fio;
- 1904: John Ambrose e Lee De Forest criaram uma válvula que modulava e enviava sinais sem fio, possibilitando a transmissão de voz.

Outro avanço muito importante ocorreu em 1971, quando um grupo de pesquisadores da Universidade do Havaí criou um pacote chamado *First-switched*, que foi a primeira rede local de comunicação de rádio sem fio interligando computadores de quatro ilhas (ENGST; FLEISHMAN, 2005).

Alguns anos depois, em 1995, surgiu a primeira tecnologia *Wireless* capaz de conectar telefones móveis e outros aparelho por meio de ondas de rádio, o *Bluetooth*. Segundo LUGLI; SOBRINHO(2012), esse protocolo é capaz de transmitir sinais de dados e voz e "se desenvolveu com uma frequência de rádio aberta, operando na faixa ISM (*Industrial, Scientific and Medical*), com comunicação por salto de frequência FH-CDMA (*Frequency Hopping - Code-Division Multiple Access*), permitindo proteção e fazendo com que a frequência seja dividida em vários canais."

Dois anos depois, em 1997, com a criação do padrão IEEE 802.11, surgiu o *Wi-Fi*. Este padrão define as normas para o uso da rede sem fio e segundo STEFANUTO; SANTOS; TORRES (2016), "sua transmissão é realizada por radiofrequência, espalhando o sinal pelo ar podendo chegar a centenas de metros, dependendo do equipamento e dos obstáculos, tais como árvores e paredes."

As operações desse tipo de rede ocorrem em bandas diferentes estabelecidas de acordo com o governo de cada país. No entanto, algumas faixas ISM que operam nos intervalos 902 MHz - 928 MHz; 2,4 GHz - 2,485 GHz e 5,15 GHz - 5,825 GHz, não necessitam de aprovação governamental (STEFANUTO; SANTOS; TORRES, 2016).

Mesmo com a conexão sendo mais lenta do que em uma rede cabeada, segundo BARBOSA (2016), o *Wi-Fi* é utilizado diariamente por 86% dos brasileiros. Muito disso se deve a mobilidade que o protocolo proporciona. Na Figura 2.3, tem-se a arquitetura de uma rede *Wi-fi*.

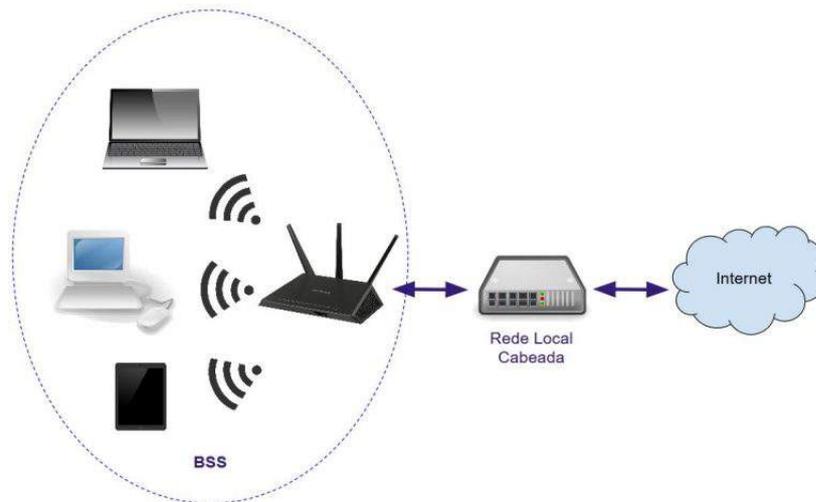


Figura 2.3: Arquitetura da rede *Wi-Fi*. Fonte: (REIS, 2018)

Outra tecnologia que utiliza da comunicação sem fio, é o RFID. No entanto, diferentemente das citadas acima, que tem o objetivo de transmitir dados, essa tecnologia é utilizada para realizar a identificação.

Segundo REI (2010), a origem da utilização da rádio frequência para reconhecimento, data-se da 2ª Guerra Mundial, aonde Ingleses, Japoneses e Alemães “utilizavam os seus radares para saberem, com antecedência, que aviões se aproximavam, mas não identificavam se eram amigos ou inimigos.”

Esse problema foi solucionado pelo inventor do radar, Sir Robert Watson-Watt, que desenvolveu um dispositivo nomeado *Identification Friend or Foe* (IFF). Esse equipamento era um *transponder* acoplado aos aviões, que respondia aos sinais enviados pela estação de rastreamento. Assim, os aviões que não respondiam eram considerados inimigos.

Tempos depois, com o avanço da tecnologia, um sistema de antifurto começou a ser comercializado. Nesse processo, uma etiqueta de 1 bit era fixada nas embalagens dos produtos. Quando as mercadorias chegavam nas prateleiras, o bit assumia o status de ligado e quando ocorria o pagamento no caixa, o operador alterava o status para desligado, assim, os detectores na saída da loja não reportariam o furto (ROBERTI, 2005). É importante notar que esse processo é utilizado até os tempos modernos.

Ainda segundo ROBERTI (2005), o RFID como se conhece hoje começou a surgir somente em 1973, quando Mario W. Cardullo registrou uma patente de um *transponder* passivo usado para destrancar portas sem chave. A partir disso, a tecnologia passou a ser utilizada em vários setores como o agropecuário para controle de gado e o de logística para controle de estoque. Na Figura 2.4, tem-se o princípio de funcionamento dessa tecnologia.

Paralelamente ao RFID, outro método de identificação bastante utilizado na atualidade é a impressão digital. O estudo que comprovou as características dessas dobras de pele encontradas

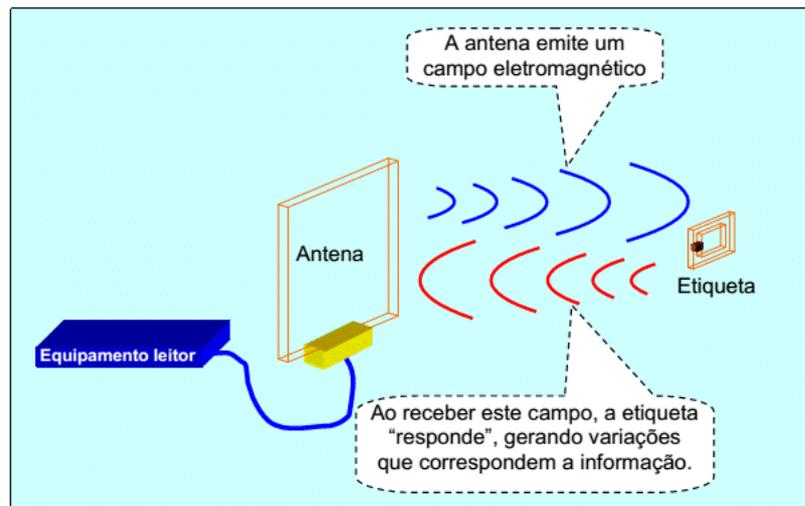


Figura 2.4: Princípio do funcionamento do RFID. Fonte: (CUNHA, 2016)

nos dedos das mãos e dos pés data-se do ano de 1667. Porém, existem relatos que a aplicação desse reconhecimento ocorre desde o período pré-histórico (MAZI; JÚNIOR, 2009).

Segundo MÁRCICO (2002), o uso desse sistema de identificação gerou uma “grande contribuição para a Polícia quando as cidades eram menores e os criminosos conhecidos, o que possibilitava o confronto das impressões colhidas nos locais de crime com as de um suspeito ou até mesmo podiam ser encontradas nos arquivos monodactilares.”

Devido ao aumento da situação demográfica urbana, tornou-se necessário que o processo de identificação criminal tivesse uma maior eficiência e dinamismo. Assim, com o advento da informática, surgiu uma tecnologia chamada AFIS (Sistema de Identificação Automatizada de Impressões Digitais), que compara digitais a partir de impressões inseridas previamente em um banco de dados.

A ideia para esse sistema começou a surgir nos anos 60, porém somente em 1970, com os esforços realizados pela Agência de Investigação Federal dos Estados Unidos (FBI), que os resultados começaram a aparecer. Neste ano, o FBI entregou para a *Cornell Aeronautics Laboratory* a missão de produzir um dispositivo capaz de digitalizar a impressão digital. Nomeado de FINDER, o dispositivo foi entregue em 1972 (MÁRCICO, 2002).

Esse sensor representou um marco não somente no âmbito policial. Como pode-se perceber nos tempos atuais, o reconhecimento por impressão digital passou a ser utilizado em diversos dispositivos, tendo como um dos principais exemplos os aparelhos celulares, facilitando e agilizando o cotidiano das pessoas.

## 2.2 Fundamentação Teórica

### 2.2.1 Normas Hospitalares

O ambiente hospitalar é um dos mais complexos para a gestão de segurança tanto dos pacientes quanto dos trabalhadores. Devido a essa grande quantidade de riscos comparados com outros setores, existem uma série de normas e protocolos para serem seguidos conforme o Programa Nacional de Segurança do Paciente (PNSP) (SAFESST, 2018).

A partir disso, a Portaria MS/GM nº 529/2013, estabeleceu a implementação de um conjunto de protocolos básicos definidos pela Organização Mundial da Saúde (OMS), sendo esses BRASIL; FIOCRUZ; ANVISA (2014):

1. Protocolo de prática de higiene das mãos em estabelecimentos de Saúde;
2. Protocolo de cirurgia segura;
3. Protocolo de segurança na prescrição, uso e administração de medicamentos;
4. Protocolo de identificação de pacientes;
5. Protocolo de comunicação no ambiente dos estabelecimentos de Saúde;
6. Protocolo de prevenção de quedas;
7. Protocolo de úlceras por pressão;
8. Protocolo de transferência de pacientes entre pontos de cuidado;
9. Protocolo de uso seguro de equipamentos e materiais.

Dois desses protocolos estão diretamente ligados ao tema deste trabalho:

#### **Uso seguro de equipamentos e materiais**

Esse protocolo está diretamente ligado a conservação e esterilização dos equipamentos utilizados em hospitais.

De acordo com o Manual de Segurança no Ambiente Hospitalar, ANVISA (2003), a esterilização “é o processo de destruição ou eliminação total de todos os microrganismos na forma vegetativa e esporulada, através de agentes físicos ou químicos.”

O tipo de desinfecção hospitalar mais utilizado é a por agentes químicos, principalmente pela capacidade de ser empregada em materiais que não suportam as altas temperaturas aplicadas nos processos a vapor e calor seco.

Como descrito por CASSETTARI; SILVEIRA (2018), esse processo é utilizado no Hospital Universitário da USP onde materiais portáteis, como exemplo o oxímetro, são friccionados com álcool 70% para realizar sua limpeza e esterilização.

### **Identificação de pacientes**

Esse protocolo elaborado pela ANVISA (2013), tem a finalidade de “garantir a correta identificação do paciente, a fim de reduzir a ocorrência de incidentes. O processo de identificação do paciente deve assegurar que o cuidado seja prestado à pessoa para a qual se destina.”

Essa identificação deve ocorrer por meio de uma pulseira colocada desde o momento da admissão no leito e para casos que o paciente não possa utilizá-la, o serviço de saúde deve definir como realizar o reconhecimento.

De acordo com esse protocolo, a pulseira deve possuir características como:

**Cor** - A pulseira deve ter a cor branca;

**Tamanho** - A pulseira deve se adequar ao perfil dos pacientes;

**Conforto** - A pulseira não deve ter formas que causam irritação a pele e devem ser produzidas com um material flexível;

**Facilidade de uso** - A pulseira deve ser fácil de limpar, impermeável e resistente a líquidos;

**Registro dos identificadores do paciente** - As informações presentes na pulseira devem ser fáceis de ler e duráveis.

### **2.2.2 Materiais**

A escolha dos materiais é uma importante etapa do projeto, sendo que nela que as características necessárias e a viabilidade são analisadas. De acordo com matéria publicada no site 3DLAB (2018), se tratando de insumos utilizados em impressoras 3D, 4 tipos de filamentos se destacam:

**Filamento PLA** - fabricado a partir de fontes renováveis, o PLA (ácido polilático) é um material de fácil impressão e não é prejudicial para a saúde e o meio ambiente.

Devido a sua dureza superficial elevada em comparação com outros materiais, ele é indicado para peças expostas à esforços de abrasão. O fato de possuir baixa contração, também possibilita o uso em peças grandes e técnicas.

**Filamento ABS** - fabricado a partir do petróleo, o ABS (acrilonitrila butadieno estireno) é o insumo mais antigo utilizado em impressões 3D. Devido a alta resistência e durabilidade, esse material é indicado para peças que tendem a sofrer impactos mecânicos.

**Filamento PETG** - o filamento PETG (polietileno tereftalato glicol) é um dos principais utilizados em impressoras 3D. Além de ser um material bastante resistente, ele não emite gases tóxicos e pode entrar em contato com alimentos.

Além dessas características, o PETG não reage ao álcool e com isso, pode sofrer limpezas profundas e esterilização. Devido a esse fato, o material vem sendo bastante utilizado em produções de suportes de máscaras que ajudam no combate ao corona vírus.

**Filamento Flexível** - esse filamento tem como característica principal, a aplicação em produtos que necessitam de flexibilidade, como pulseiras, palmilhas, etc.

### 2.2.3 ESP32

Como dito na seção anterior, o ESP32 é bastante utilizado em aplicações de IoT. Essa placa conta com um processador *dual-core*, cerca de 500 kBytes de memória SRAM, *Wi-Fi* e *Bluetooth 4.2* de forma integrada além de possuir 36 sinais de GPIO, sinais PWM e conversores Analógico-Digital, como mostrado na Figura 2.5. Outros recursos que diferenciam esse microcontrolador são o sensor de temperatura para monitoramento da temperatura da placa e a interface para sensor de toque, facilitando o uso de teclados e botões capacitivos.

Para realizar a programação do dispositivo é possível utilizar várias IDEs, como a do *CodeBlocks*, *Netbeans*, Lua e principalmente a do *Arduino*.

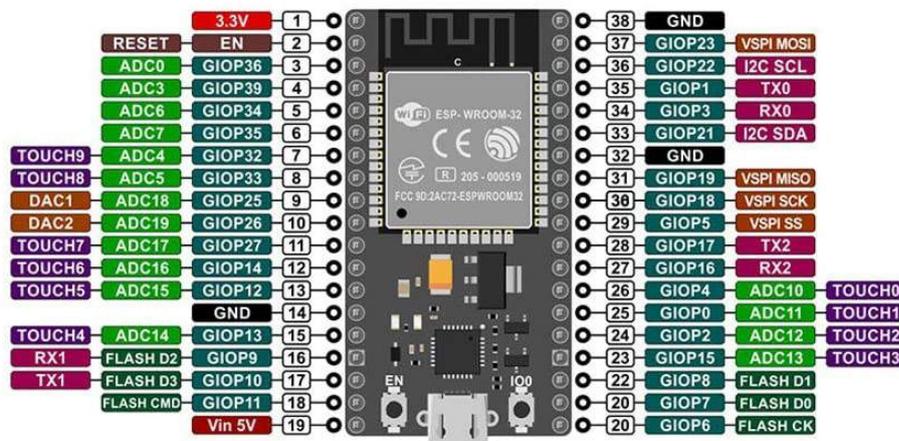


Figura 2.5: Pinagem ESP32. Fonte: (ZIMMERMANN, 2018)

### 2.2.4 RFID

Segundo PEDROSO; ZWICKER; SOUZA (2009), a tecnologia empregada no RFID “é baseada na utilização de ondas eletromagnéticas (de rádio frequência) como meio para comunicar os dados de identificação de algum elemento, tais como produtos, componentes, caixas, paletes, contêineres, veículos, pessoas, ativos, máquinas e serviços.”

Como visto na Figura 2.6, o funcionamento desse sistema é composto por 3 partes:

**Leitura** - A etapa de leitura é constituída por 2 componentes, a TAG e o leitor. A seguir, cada um deles será explicado:

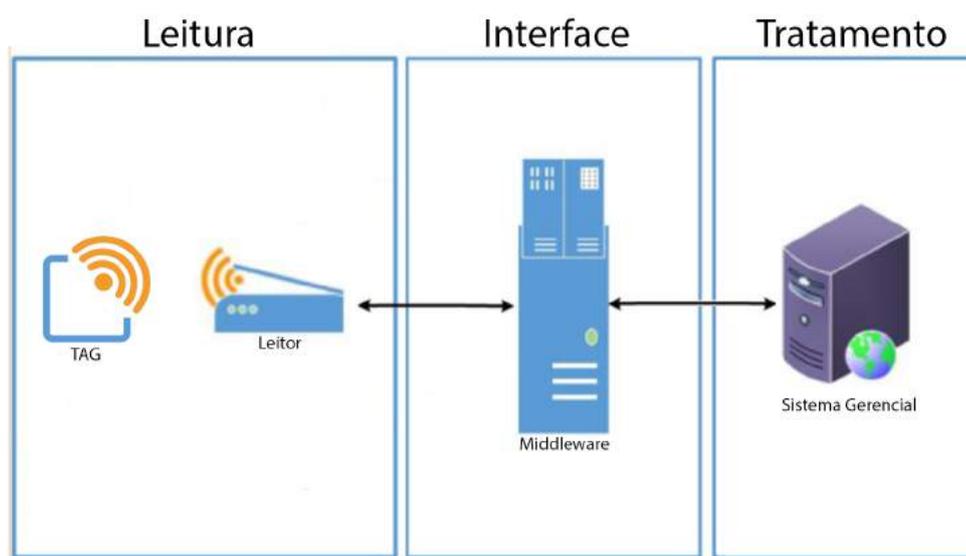


Figura 2.6: Princípio do funcionamento do RFID. Adaptado de: (HADDAD; RIZZOTTO; MALDONADO, 2016)

**TAG** - Composta por um chip e uma antena, usa um sinal de rádio frequência para se comunicar. De acordo com REI (2010), essas etiquetas podem ser classificadas de 4 formas:

- **Passivas:** Com sua configuração interna mostrada na Figura 2.7, esse tipo de TAG não possui fonte própria de alimentação, obtendo energia a partir do sinal emitido pelo leitor. Possui também grande tempo de vida porém sua capacidade de memória e computacional são limitadas. Além disso, são as mais baratas e utilizadas.

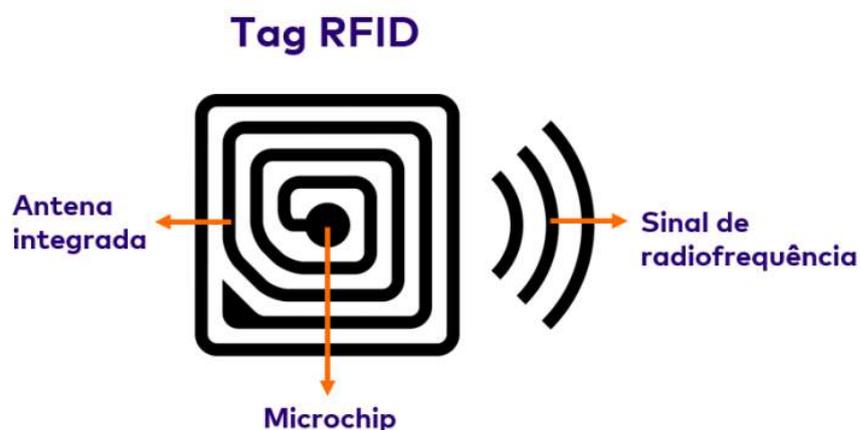


Figura 2.7: Configuração interna de um tag RFID passiva. Fonte: (NOGUEIRA, 2019)

- **Ativas:** Possuem uma bateria interna para fornecimento de energia. Com um tempo de vida limitado a carga da mesma, esse tipo de etiqueta permite

uma maior distância de leitura e também a incorporação de diversos circuitos integrados, podendo inclusive ser interligada a um sistema de GPS.

- Semi-passivas: Possuem o mesmo princípio de funcionamento das etiquetas passivas, porém possui uma bateria interna. Essa alimentação, possibilita a inserção de sensores na mesma.
- Semi-ativas: Tipo de etiqueta que não se encontram em atividade até serem ativadas pelo leitor. A partir dessa ativação, seu funcionamento é idêntico as etiquetas ativas.

**Leitor** - Esse dispositivo serve para obter ou alterar a informação presente na tag e podem ser caracterizados conforme sua mobilidade:

- Fixos: Presentes em pedágios, cais de carga/descarga dentre outras estruturas fixas.
- Móveis: Presentes em empilhadeiras, porta paletes, dentre outras estruturas móveis.
- Portáteis: Presentes em dispositivos que necessitam ser utilizados em diversos lugares. Usados também em situações de falhas dos outros leitores.

Dentre as principais características que um leitor deve possuir, estão a capacidade de detectar e evitar possíveis interferências e o uso de diversos métodos de modulação na emissão e recepção do sinal independente de qualquer alteração no hardware.

**Interface** - O *middleware* é responsável pela interface entre a leitura e o sistema de identificação. Esse *software* recebe os dados do leitor, analisa as informações necessárias junto à base de dados e as retorna para que o diálogo entre o leitor e a etiqueta seja realizado. Por fim, o *middleware* atualiza a base de dados de forma que o sincronismo do processo seja mantido.

**Tratamento** - O sistema gerencial é responsável pelo tratamento da informação, esse sistema processa a informação e disponibiliza aos diversos utilizadores da mesma. Além disso, é responsável também pelo fornecimentos das informações necessárias no *middleware*.

### 2.2.5 Biometria

Como dito na seção anterior, as impressões digitais são formadas por dobras de pele. Esse tecido é constituído pela derme, uma camada mais profunda que apresenta elevações denominadas de papilas e a epiderme, uma camada superficial e transparente que se sobrepõe a derme. (MAZI; JÚNIOR, 2009)

Essas papilas, também chamadas de linhas ou cristas, podem ser encontradas tanto nas superfícies palmares quanto plantares e são separadas por sulcos ou vales caracterizando uma impressão digital como mostrada na Figura 2.8.

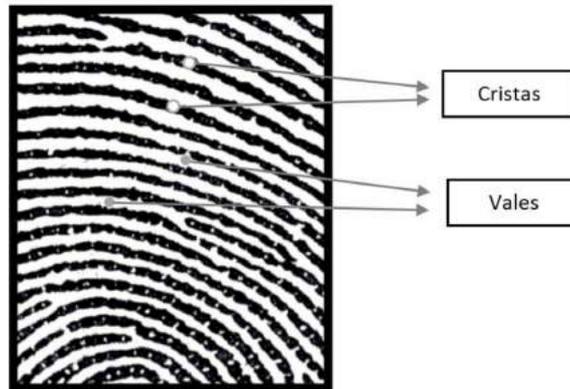


Figura 2.8: Cristas e vales em uma impressão digital. Fonte: (SILVA, 2015)

Segundo MAZI; JÚNIOR (2009), os desenhos formados pelas papilas garantem importantes características, como:

- Unicidade - não existem impressões idênticas;
- Imutabilidade - a característica da impressão se mantém por toda a vida;
- Praticabilidade - os desenhos papilares são facilmente obtidos;
- Classificabilidade - mesmo com infinitas variedades nas minúcias, os desenhos atendem padrões facilitando a classificação.

A combinação dessas minúcias, são utilizadas em abordagens para comparação de digitais e são divididas basicamente em 7 tipos, mostrados na Figura 2.9.



Figura 2.9: Tipos de minúcias encontradas em uma impressão digital. Adaptado de: (MALTONI *et al.*, 2009)

A partir disso, MALTONI *et al.* (2009) propôs três abordagens para se realizar a combinação de biometrias:

- Combinação baseada em correlação: a partir da sobreposição de duas digitais, correlações são buscadas;
- Combinação baseada em minúcias: minúcias são colocadas em um plano 2D e um alinhamento entre elas são buscados;
- Combinação não baseada em minúcias: abordagem realizada a partir da orientação e frequências das cristas e vales. É utilizada em imagens de baixa qualidade.

## Sensores

Diversos tipos de sensores podem ser usados para a aquisição de impressões digitais, mas todos eles compartilham a mesma estrutura padrão mostrada na Figura 2.10.

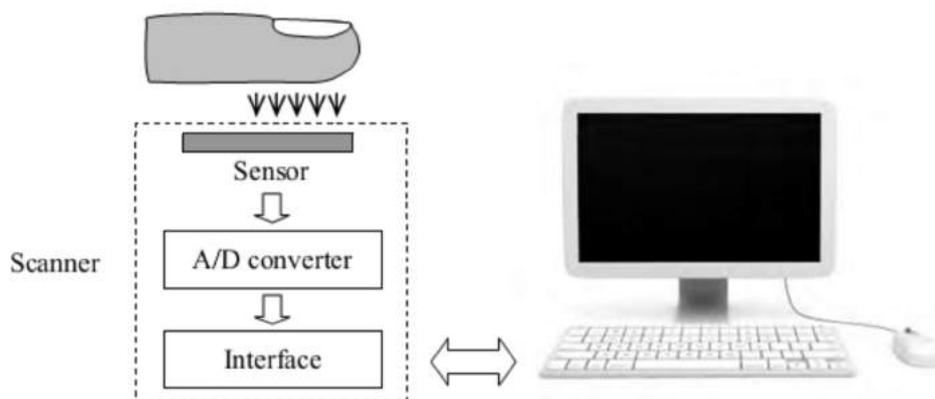


Figura 2.10: Diagrama de funcionamento de um sensor de impressões digitais. Fonte: (MALTONI *et al.*, 2009)

Como descrito por SILVA (2015), “o sensor efetua a leitura do padrão de cristas da superfície do dedo e converte o sinal analógico recebido em um sinal digital por meio do módulo conversor analógico/digital.” Após isso, um módulo de interface realiza a comunicação com o dispositivo externo.

Os sensores ópticos foram os primeiros a serem utilizados para obtenção da impressão digital e ainda são os mais aplicados. Nesse tipo de sensor, quando o dedo entra em contato com o prisma, as cristas encostam na superfície do mesmo e os vales se mantêm a uma certa distância, conforme mostrado na Figura 2.11.

O outro lado do prisma contém uma luz difusa, geralmente um conjunto de LEDs, que é refletida pelos vales e absorvida aleatoriamente pelas cristas. Os raios refletidos são focados por meio de uma lente e chegam a um sensor de imagem CCD ou CMOS (SILVA, 2015).

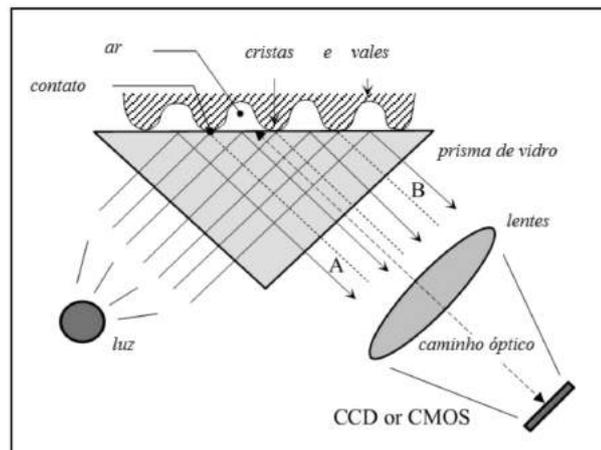


Figura 2.11: Diagrama de um sensor óptico. Fonte: (MALTONI *et al.*, 2009)

### 2.2.6 Firebase

De acordo com SILVA (2018), o *Firestore* é uma “plataforma poderosa da Google para armazenamento e sincronização de dados em tempo real”. Essa ferramenta proporciona uma gama de soluções de desenvolvimento voltadas para a aceleração e integração de recursos baseados em nuvem e em aplicações móveis.

Além disso, cada base de dados no *Firestore* é guardada como uma árvore de objeto JSON, assim, possuindo uma estrutura única que possui diferenças quando comparada com outras bases de dados (CHENG, 2017).

### 2.2.7 API

A sigla API significa Interface de Programação de Aplicações<sup>2</sup> e de acordo com FERNANDES (2018), as API's “são uma forma de integrar sistemas, possibilitando benefícios como a segurança dos dados, facilidade no intercâmbio entre informações com diferentes linguagens de programação e a monetização de acessos”. As possibilidades geradas pelo uso dessa interface, proporcionam aos desenvolvedores a possibilidade de conectar diversos tipos de tecnologias heterogêneas, como por exemplo, diferentes bancos de dados.

Para realizar essa comunicação entre aplicações, as API's utilizam requisições HTTP, que são responsáveis pelas operações básicas necessárias para a manipulação dos dados, sendo elas:

- *POST*: realiza a criação de dados no servidor;
- *GET*: realiza a leitura de dados;
- *DELETE*: exclui as informações;

<sup>2</sup>Do inglês *Application Programming Interface*.

- *PUT*: atualiza os registros.

## REST API

Visando padronizar a comunicação para aproveitar todo o potencial fornecido pelo HTTP, Roy Fielding criou em 2000 na sua tese de doutorado a arquitetura REST. Esse acrônimo significa Transferência de Estado Representacional<sup>3</sup> e se consiste em uma arquitetura associada a aplicações web com princípios e regras que quando seguidas, permitem a criação de um projeto com interfaces bem definidas (PIRES, 2017).

A arquitetura tem algumas características relativas ao modelo estrutural, sendo elas definidas por BRITO (2020) como:

- Cliente servidor: Conceito bem comum que separa as responsabilidades do *frontend* e do *backend*.
- Sem estado: As informações necessárias para o servidor entender a solicitação devem sempre ser informadas, tendo em vista que o estado anterior não é mantido.
- *Cacheable*: A resposta de uma solicitação deve implicitamente ou explicitamente informar se o dado pode ser mantido em cache ou não.
- Interface uniforme: Modelo mais completo, onde deve-se identificar os recursos, manipulá-los através de representações HTTP com mensagens auto-descritivas, sendo que a resposta deve conter links de conteúdos relacionados ao recurso.

### 2.2.8 Node.js

O Node.js é uma tecnologia voltada para a execução de códigos *JavaScript* fora do navegador. Com ele torna-se possível a construção de *web* em geral, desde sites até mesmo API's. Esse fato se deve a união do ambiente de execução fornecido pelo próprio Node.js e o motor de interpretação e execução de *JavaScript* presente no *Google Chrome*, chamado de V8 (DEV MEDIA, 2019).

Ao contrário do que acontece com outros ambientes de desenvolvimento, o Node.js não depende de diversas tarefas para a execução de processos concorrentes na lógica de negócio do servidor. De acordo com SOUSA (2015), “deve-se entender o Node.js como uma plataforma de desenvolvimento de servidores *web* que possibilita a construção de sistemas altamente escaláveis, através de uma abordagem baseada em eventos, entradas e saídas.”

---

<sup>3</sup>Do inglês *Representational State Transfer*

### 2.2.9 Flutter

O *Flutter* é um *framework* criado pela Google para o desenvolvimento de aplicações mobile tanto para *Android* quanto *IOS*. O principal diferencial dessa ferramenta é a utilização de uma abordagem até então única para lidar com as peculiaridades dos componentes nativos de cada plataforma.

De acordo com ANDRADE (2020), ao criar um aplicativo com o *Flutter*, “o código é compilado para a linguagem base do dispositivo, ou seja, as aplicações são realmente nativas e por isso conseguem acessar recursos do dispositivo sem a “ajuda” de terceiros e com o desempenho maior.”

As principais características desse *framework* são:

- Possibilidade de ser utilizado nos principais sistemas operacionais: *Windows*, *Linux* e *MacOS*.
- A partir do mesmo código fonte, aplicativos podem ser gerados tanto para *Android* quanto para *IOS*.
- Uma aplicação criada com o *Flutter*, possui acesso nativo aos recursos do dispositivo.
- Como o código fonte é transformado em código nativo, as aplicações criadas com ele possuem maior desempenho.

#### Dart

A linguagem base utilizada para desenvolvimento dos aplicativo com o *Flutter* é o *Dart*. Ela foi projetada para a programação de ferramentas de desenvolvimento para aplicações modernas e capacitadas para ambientes de alta performance. Essa linguagem se caracteriza por:

- Baseada em compilação de código *JavaScript*;
- Orientada a objetos;
- Sintaxe baseada na linguagem *C*;
- Implementa heranças simples.

### 2.2.10 Protocolo I2C

O protocolo *I2C* (*Inter-Integrated Circuit*), foi desenvolvido pela *Philips* na década de 90 com o objetivo de simplificar a comunicação entre a unidade de controle e os periféricos (KALINSKY, 2001).

Como mostrado na Figura 2.12, esse barramento é composto por dois sinais: o SCL responsável pelo transporte do *clock* (tornando esse um protocolo síncrono) e o SDA, que é o sinal responsável pelo transporte de dados em ambas as direções (CARVALHO, 2016).

Segundo CAMARA (2016), o número de ligações em um único barramento é “limitado tanto pelo tamanho do endereço, que pode ser de 7, 10 e até 16 *bits*; como por restrição de espaço, já que não se pode ultrapassar poucos metros de fios, pois a capacitância total máxima, algo em torno de 400pf, impede o funcionamento correto do barramento.”

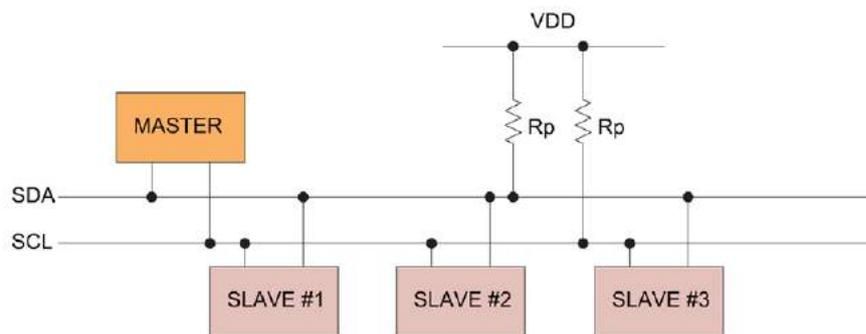


Figura 2.12: Topologia do protocolo I2C. Fonte: HOA (2020)

Os dispositivos conectados por esse protocolo são classificados como mestres (MASTER) ou escravos (SLAVE), onde o mestre é a unidade de controle responsável por coordenar todos os periféricos, chamados de escravos.

Além da facilidade originada pela baixa quantidade de conexões, o I2C possui outras vantagens quando comparado com outras formas de comunicação:

- Capacidade de incluir ou excluir dispositivos no barramento sem afetar os outros dispositivos já conectados.
- Organização em blocos, tornando o esquemático final simples e funcional.
- Taxas de transmissão de dados flexíveis.
- Capacidade de endereçar cada dispositivo do barramento de forma independente.

## Desenvolvimento e Resultados

Neste capítulo as etapas desenvolvidas para alcançar os objetivos propostos são apresentadas. Visando uma maior organização textual, realizou-se uma divisão por tópicos.

O esquema de funcionamento proposto para o dispositivo está representado na Figura 3.1.

### **3.1 Materiais e Métodos**

Nos tópicos a seguir estão representados os materiais e métodos escolhidos para o melhor funcionamento do dispositivo em questão.

#### **3.1.1 Eletrônica**

##### **Controlador**

A escolha dos componentes teve início com o controlador. Para isso, devido ao tamanho compacto e as características de comunicação sem fio embutidas citadas no capítulo anterior, escolheu-se o ESP32.

##### **Sensor de Impressão Digital**

O sensor de impressão digital escolhido foi o modelo FPM10A DY50, mostrado na Figura 3.2. Essa definição se deve a algumas características como tempo de leitura menor que 1 segundo, visor em vidro que possibilita a esterilização pós utilização e capacidade de armazenar 162 impressões.

Esse sensor possui um processador digital de sinal (DSP) de grande precisão, que realiza a renderização, cálculos, localização de recursos e pesquisa de imagens e proporciona muita simplicidade na utilização.

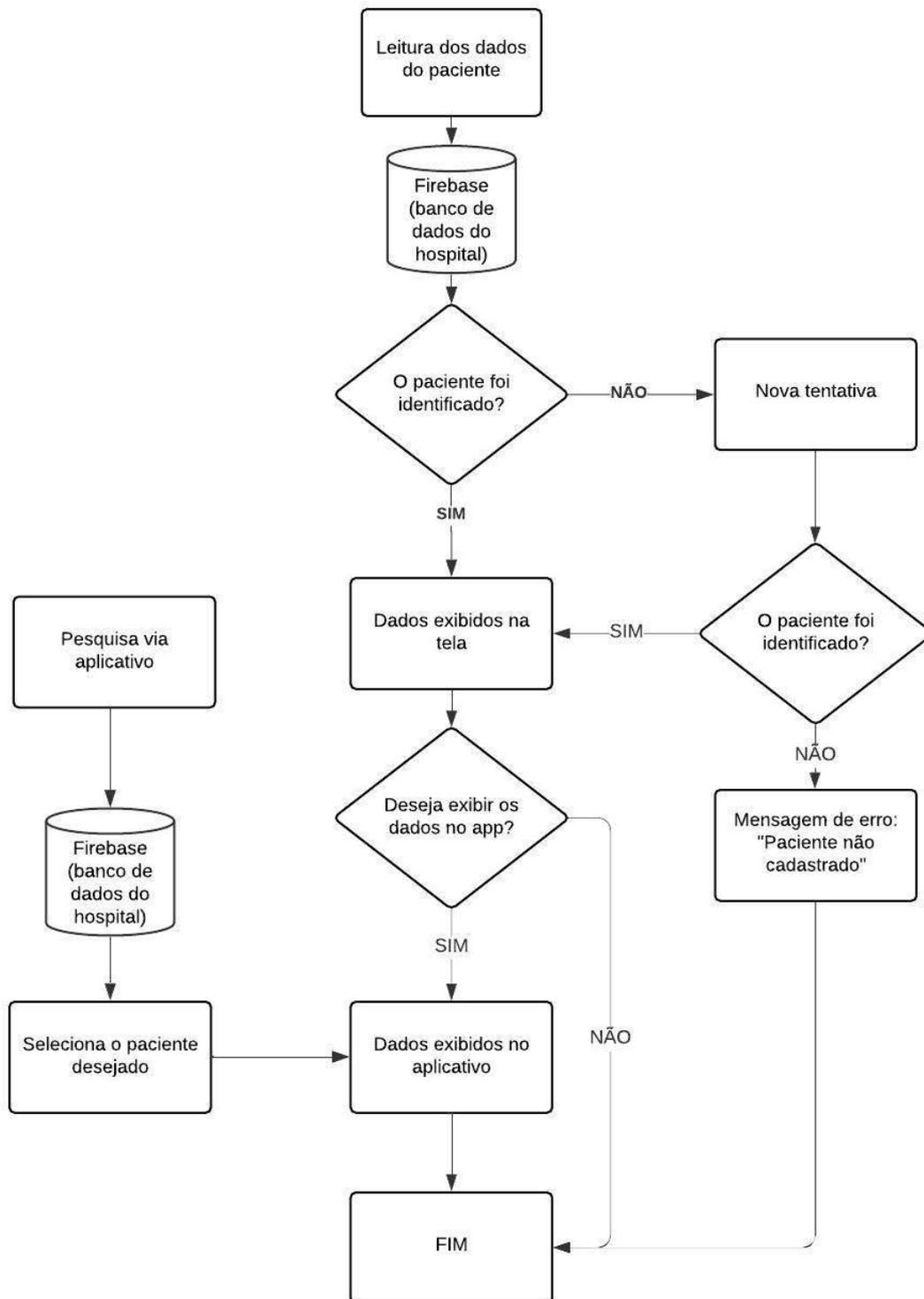


Figura 3.1: Fluxograma de funcionamento de dispositivo

### Sensor RFID

A seguir definiu-se o leitor RFID. O modelo escolhido foi o MFRC522, mostrado na Figura 3.6, que possui um chip com uma tecnologia criada pela *NXP Semiconductors* para cartões



Figura 3.2: FPM10A DY50. Fonte: VLADCONTROL

inteligentes sem contato e cartões de proximidade a uma frequência de 13,56MHz. Além de possuir baixo consumo e tamanho compacto, esse chip permite realizar tanto a leitura quanto a gravação em cartões que seguem o padrão *Mifare*, muito usado no mercado. Alguns exemplos de cartões suportados são: *Mifare1 S50, S70 Mifare1, Mifare UltraLight, Mifare Pro, Mifare Desfire*.



Figura 3.3: MFRC522. Fonte: BANGGOOD

### Tag RFID

Como o sensor atua com a frequência de 13,56MHz, a TAG deve atuar na mesma faixa. Assim, cada instituição poderá escolher a que melhor se adequar a sua necessidade.

No entanto, visando o maior conforto do paciente, recomenda-se os dois seguintes modelos:

- TAG de papel: Mostrada na Figura 3.4, essa TAG resistente à água pode ser acoplada a pulseira padrão já utilizada no hospital. Sua principal vantagem, é a capacidade de

realizar a impressão das informações do paciente direto nela.



Figura 3.4: TAG de papel. Adaptado de: YARONGTECH

- Pulseira com TAG acoplada: Essa pulseira, mostrada na Figura 3.5, possui como diferencial a TAG já acoplada, tornando-a mais resistente. Nesse modelo, a impressão seria feita em etiqueta de papel normal, da mesma forma que é feita na atualidade.



Figura 3.5: Pulseira com TAG inclusa. Adaptado de: OASES

## Tela

A escolha do display foi baseada nas informações que ele tem que exibir. Como determinado pela ANVISA, os meios de identificação de pacientes devem conter no mínimo dois indicadores como nome completo do paciente e data de nascimento no formato curto (DD/MM/AAAA). Pode-se exibir também a filiação, número do prontuário, local de internação, dentre outros. Com isso, o modelo escolhido foi o de 20 caracteres e 4 linhas com adaptador I2C integrado, como mostrado na Figura 3.6.



Figura 3.6: Display 20x4. Fonte: SAINSMART

A presença da tecnologia I2C no display é de extrema importância na economia de portas do controlador. Sem ela, seriam necessárias 6 portas exclusivas para a troca de dados entre a tela e o ESP32, sendo que com ela, somente 2 são necessárias.

### Alimentação

A bateria a ser utilizada deve ter carga suficiente para fornecer uma boa autonomia ao dispositivo. Para defini-la, calculou-se a corrente consumida por cada componente e a partir disso, estimou-se a corrente total do dispositivo.

Tabela 3.1: Tabela de corrente por componente.

Componente	Corrente
ESP 32	80mA
MFRC522	26mA
DY50	120mA
Display 20x4	40mA
<b>Total</b>	266mA

Como a tensão de alimentação tanto do ESP-32 quanto do LCD é de 5V, escolheu-se a bateria que atendesse essa tensão com a maior capacidade possível. Com isso, uma bateria adequada as necessidades do projeto é a *LiPo* recarregável de 5V que possui 1200mAh, mostrada na Figura 3.7, fornecendo uma autonomia de 4:30h de uso intenso ao dispositivo.

É importante ressaltar que os demais componentes do dispositivo, que tem a tensão de 3.3V, podem ser alimentados com a saída do próprio microcontrolador, que realiza automaticamente a conversão para tal tensão. Além disso, tem-se também que somente com a realização de testes em campo com o dispositivo finalizado que essa escolha poderá ser validada.



Figura 3.7: Bateria 5V. Fonte: ALIBABA

### Módulo de carga

Visando facilitar o processo de recarga da bateria, de modo que a mesma não precise ser retirada do dispositivo, definiu-se a utilização de um módulo de carga com entrada micro USB. O modelo escolhido foi o TP5046, mostrado na Figura 3.8.



Figura 3.8: Módulo de carga TP5046. Fonte: MERCADO LIVRE

Esse módulo conta com proteção contra sobrecarga e termina o ciclo de carga automaticamente, quando a corrente cai para um décimo do valor programado após a tensão flutuante ser atingida.

### Diversos

Além de todos os componentes citados nos tópicos anteriores, tem-se também a utilização de itens básicos de eletrônica, como:

- Cabos diversos
- LED verde
- LED vermelho
- Resistores (400 $\Omega$ , 1K $\Omega$ , 4K $\Omega$  e 8K $\Omega$ )
- Chave On/Off
- Diodo 1N4729A
- Transistor BC548
- Push-Button

### 3.1.2 Estrutura

Visando atender as necessidades do ambiente hospitalar, foi necessário escolher um material não tóxico, resistente ao desgaste e principalmente ao procedimento de esterilização química. Com isso, o material escolhido foi o PETG que além de possuir todas as características necessárias, possui um ótimo custo-benefício.

### 3.1.3 Softwares

Para o desenvolvimento do trabalho, alguns *softwares* tem que ser utilizados. Sendo eles:

#### Arduino IDE

Na programação do controlador, a IDE do Arduino foi utilizada. Essa escolha se deve ao fato de o *software* ser gratuito e de código aberto. A linguagem de programação utilizada nessa IDE é o C++.

#### Visual Code Studio

Utilizado na programação do aplicativo e na construção da API, o *VS Code* é um *software* livre e de código aberto lançado em 2015. Sua escolha se deu pela possibilidade de se programar na linguagem Dart e pela familiaridade do autor com o mesmo.

### Flutter

Em conjunto com o *software* citado acima, o *framework Flutter* também foi utilizado na elaboração da aplicação. Como citado no capítulo anterior, essa ferramenta possibilita o desenvolvimento multiplataforma e facilita a construção de uma interface amigável ao aplicativo projetado.

### SolidWorks

Esse programa de computação gráfica permite a criação de objetos com modelagem 3D a partir da tecnologia CAD 3D. Seu funcionamento se dá a partir de bases de objetos sólidos, que podem ser modeladas para a criação de inúmeros arquivos diferentes. A escolha desse *software* se deu pela experiência do autor gerada no decorrer do curso.

### Node.js

Após a programação da API no *Visual Code*, utilizou-se o Node.js para a realização da implementação da aplicação, ou seja, a compilação, otimização e a interpretação a partir da máquina virtual V8.

### MultiSIM

Esse *software* foi utilizado no projeto e simulação do circuito eletrônico da bateria. Ele foi escolhido devido à sua forma simples de apresentação dos componentes e instrumentos de medição, além de ser gratuito.

## 3.2 Comunicação sem fio

A comunicação entre o dispositivo, aplicativo e base de dados será via *Wi-Fi*. Para isso, como mostrado na Figura 3.9, um roteador será utilizado como forma intermediadora entre os fatores.

Nessa configuração, cada componente assumirá as seguintes características:

**Dispositivo** - O dispositivo enviará e receberá as informações de consulta no *Firebase*, através de uma API. Ele enviará também, as informações do paciente a serem exibidas no aplicativo.

**Aparelho Android com o aplicativo** - No *app*, será configurado uma identificação para o dispositivo. Assim, através do serviço web, o aparelho ficará monitorando se o dispositivo enviou alguma nova informação. O aplicativo também poderá fazer consultas diretas no *Firebase*.

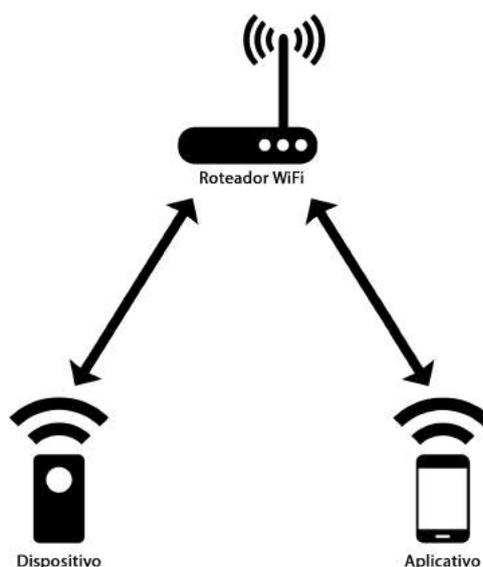


Figura 3.9: Diagrama da comunicação sem fio.

## 3.3 Base de dados

A forma ideal de se obter as informações do paciente é a partir da conexão tanto do dispositivo quanto do aplicativo ao banco de dados da instituição onde o mesmo será utilizado. Com isso, para o desenvolvimento do projeto, decidiu-se a utilização do *Firebase* como uma forma genérica de simular esse armazenamento de dados.

### 3.3.1 Firebase

O primeiro passo para a utilização do *Firebase*, foi a criação do projeto. Nessa parte, algumas informações básicas como o nome da aplicação devem ser preenchidas. Com a conclusão dessa etapa, pode-se começar a utilização da ferramenta em si.

Com isso, a partir da função *Cloud Firestore*, cadastrou-se 5 pacientes genéricos para a realização de testes. Essa função representa um banco de dados flexível e escalonável para desenvolvimentos focados em dispositivos móveis, *web* e servidores.

O diagrama com as informações cadastradas, está representado pela Figura 3.10.

Além dos dados do paciente em si, essa ferramenta também foi utilizada para realizar a autenticação do login dos usuários. A partir da função *Authentication* fornecida pelo próprio *Firebase*, foram cadastrados diversos usuários de teste, realizando a validação pelo e-mail e senha.

### 3.3.2 REST API

Como a conexão com o *Firebase* pode ser realizada de forma direta somente com o aplicativo, tornou-se necessário desenvolver uma API do tipo REST para realizar a troca de dados

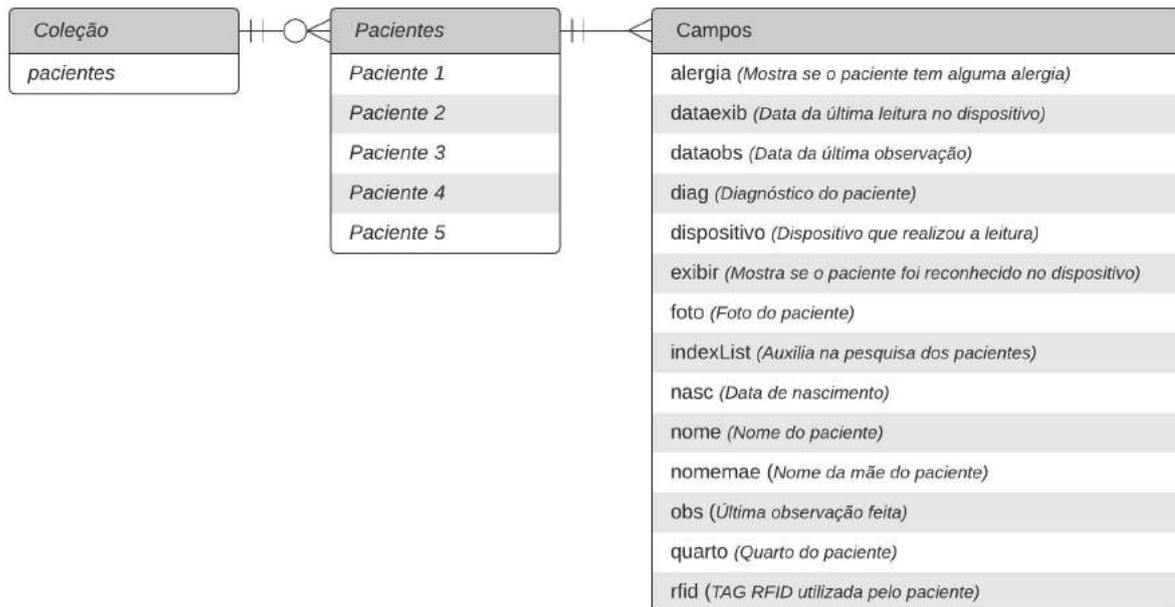


Figura 3.10: Diagrama do banco de dados no Firebase.

com o microcontrolador. Antes de iniciar a construção em si, definiu-se as funcionalidades que essa aplicação devia conter, sendo elas:

- O parâmetro de consulta, será por meio do código da *TAG RFID* a ser enviado junto com a requisição da API. Essa chave será enviada de forma direta para os casos em que o paciente for identificado pela leitura da pulseira e para os casos em que a identificação for por meio da impressão digital, será enviada a *TAG* vinculada a esse paciente.
- Com o código, será identificado o paciente a qual ele pertence.
- Caso o paciente seja identificado, a API retornará o status 200 e caso ocorra algum erro, o status 400.
- Com o paciente identificado a API retornará por meio de um *GET*, os dados a serem exibidos no *display* do dispositivo (nome, data de nascimento, nome da mãe, diagnóstico do paciente e quarto).
- Por fim, será utilizado um *POST* para atualizar os campos utilizados para a exibição do paciente no aplicativo. Nessa parte o campo “exibir” será atualizado para “sim” e o campo “dataexib” será atualizado para a hora que a requisição foi realizada.

Tendo isso definido, pode-se então iniciar o desenvolvimento da aplicação e chegou-se no código mostrado no Apêndice A. O próximo passo, foi realizar o vínculo com a base de dados.

Para isso, foi necessário ativar o *framework Cloud Functions* do próprio *Firebase*, que permite executar automaticamente o código em resposta a solicitações HTTPS.

Com isso, utilizou-se o Node.js para realizar a implementação da aplicação em conjunto com o *Functions*, de onde foi obtido a URL da API em questão.

## 3.4 O Aplicativo

Para o desenvolvimento do aplicativo, definiu-se de forma inicial as funcionalidades do mesmo e o esquema de funcionamento, como mostrado na Figura 3.11.

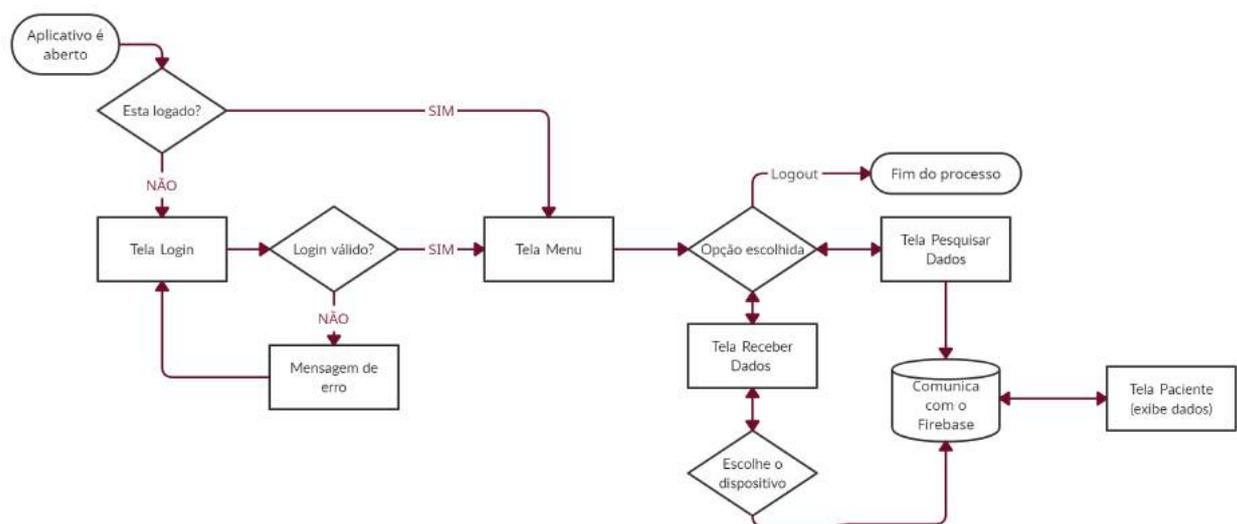


Figura 3.11: Fluxograma de funcionamento do aplicativo.

Após isso, pode-se então iniciar o esboço das telas e das características que cada uma delas devia possuir.

### 3.4.1 Características do Aplicativo

Visando a otimização do processo de utilização, o aplicativo deve possuir uma interface simples e objetiva. Para isso, como mostrado na Figura 3.12, decidiu-se a utilização de apenas 4 telas principais, com a característica de cada uma mostrada a seguir:

**Tela de login** - Primeira tela do aplicativo. Nela, o operador deve fazer o login com e-mail e senha.

**Tela do menu** - Nessa tela, estará presente a logo do hospital em conjunto com dois botões:

- Pesquisar paciente: ao escolher essa opção, a tela de pesquisa será aberta;
- Receber Dados: ao escolher essa opção, o *app* receberá os dados do paciente identificado pelo dispositivo e a tela de exibição será aberta;

**Tela de pesquisa** - O intuito dessa tela é realizar uma busca de forma independente ao dispositivo. Nela, é necessário digitar o nome do paciente desejado e ao clicar em Pesquisar, feita a identificação, a tela de exibição de dados será aberta.

**Tela de exibição de dados** - Nessa tela, os dados do paciente identificado serão exibidos.

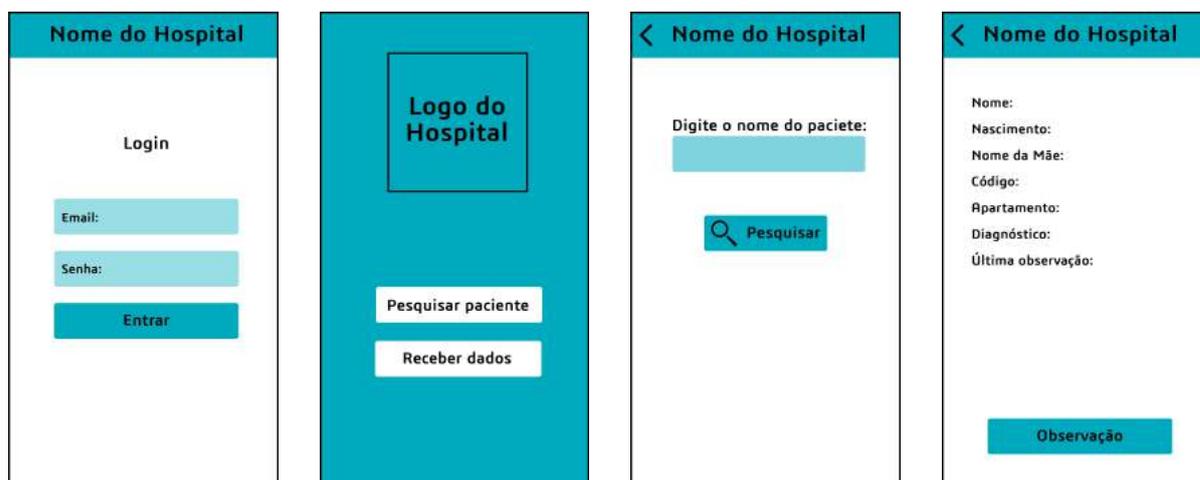


Figura 3.12: Esboço das telas do aplicativo

### 3.4.2 Desenvolvimento do Aplicativo

Com os esboços das telas e as funcionalidades de cada uma delas definidas, pode-se então iniciar o desenvolvimento. Visando uma maior personalização ao aplicativo, definiu-se um nome genérico de um hospital e com isso, os resultados obtidos serão apresentados nas subseções a seguir:

#### Login

Essa tela, mostrada na Figura 3.13, foi pensada visando garantir uma maior segurança aos dados dos pacientes. Assim, somente os usuários cadastrados pela instituição utilizadora do dispositivo poderão acessar as funcionalidades do *app*.

Inserindo um e-mail e senha válidos, será feita a validação dos mesmos e a próxima tela (menu) é aberta. Para os casos que ocorra algum erro, foram desenvolvidas 2 mensagens de alertas (senha errada e usuário não encontrado) além de uma tela auxiliar para recuperação de senha. Essas, podem ser visualizadas na Figura C.1 presente no Apêndice C.

#### Menu

O *display* do menu é bastante simples. Nessa parte, tem-se apenas a seleção da função que o operador deseja seguir:

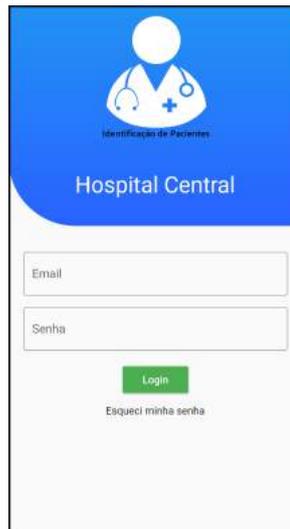


Figura 3.13: Tela de login do aplicativo.

- Receber dados
- Pesquisar paciente

Essa tela, é representada pela Figura 3.14.



Figura 3.14: Tela do menu do aplicativo.

Além disso, foi desenvolvido também um botão (localizado na parte superior esquerda) com a funcionalidade de fazer o *logout* do sistema.

### Receber Dados

A funcionalidade para receber dados, foi desenvolvida visando exibir somente o paciente identificado pelo dispositivo. Para isso, decidiu-se utilizar o parâmetro "exibir" cadastrado na base de dados.

Quando o paciente é reconhecido no dispositivo, a API projetada altera esse parâmetro para “sim”. Com isso, ele será exibido na tela como mostrado na Figura 3.15 e ao clicar no paciente, a tela com os dados do paciente será aberta.



Figura 3.15: Tela receber dados do aplicativo.

Nessa etapa, foi desenvolvido também um botão localizado na parte superior direita com o intuito de sincronizar o aplicativo ao dispositivo utilizado no momento, que pode ser visto na Figura C.5 presente no Apêndice C

### **Pesquisar Pacientes**

Já a tela de pesquisar pacientes é mostrada na Figura 3.16. Nela, ao começar a digitar o nome de algum paciente que esteja cadastrado na base de dados, por meio de filtros, somente o paciente que contém o nome desejado será exibido na tela, fato mostrado na Figura C.5 presente no Apêndice C.

Vale ressaltar que visando a formação de um histórico de identificação realizada pelo dispositivo, nessa etapa os pacientes são ordenados por ordem de atualização do campo “dataexib”, que recebe a hora local atual sempre que a identificação é realizada via API.

### **Tela Paciente**

Por fim, como mostrado na Figura 3.17, tem-se a tela responsável por exibir os dados do paciente selecionado. Nela, além da foto e do nome, são mostrados o quarto, data de nascimento, nome da mãe, código de cadastro, se o paciente possui alguma alergia e o diagnóstico.

Além disso, como visto na Figura 3.17 presente no Apêndice C, é possível também inserir uma observação que é mostrada em conjunto com o horário em que ela foi escrita.

Quando essa tela é fechada, o campo “exibir” é alterado para “não” nos casos que o paciente tenha sido reconhecido pelo dispositivo.

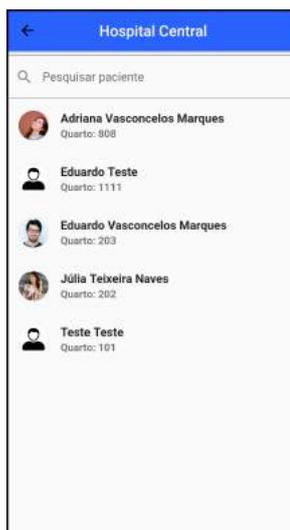


Figura 3.16: Tela pesquisar dados do aplicativo.



Figura 3.17: Tela de exibição de dados do aplicativo.

## 3.5 O Dispositivo

### 3.5.1 Desenvolvimento

O desenvolvimento do dispositivo, teve seu início com a definição dos pinos que cada periférico seria conectado. Essa escolha, levou em conta as necessidades de comunicação de cada componente, e a pinagem final pode ser vista na Tabela 3.2. Em seguida, realizou-se todas as conexões e testou-se o funcionamento de cada elemento, buscando validar o desempenho dos mesmos.

Feito isso, passou-se para a programação geral. Essa parte, se iniciou com a inclusão das bibliotecas necessárias, seguida pela definição dos pinos escolhidos anteriormente, implementação de variáveis globais e por fim, o cadastro do nome da rede *Wi-Fi* e da senha.

Tabela 3.2: Tabela de conexão do ESP-32 com os componentes.

Componente	Pino do componente	Pino do ESP-32
<b>RFID</b>	SDA	2
	SCK	15
	MOSI	23
	MISO	19
	RST	5
<b>LCD</b>	SDA	21
	SCL	22
<b>FingerPrint</b>	RX	17
	TX	16
<b>PushButton</b>	RFID	32
	FingerPrint	12

O próximo passo, foi a função *setup*. Nessa etapa, inicializou-se o LCD, o sensor de impressão digital e também o sensor *RFID*, além de efetivar a conexão com a rede.

Com o microcontrolador apto e todos os periféricos inicializados, foi possível então projetar as funções responsáveis por identificar os pacientes tanto pelo *RFID* quanto pela impressão digital. Essas funções estão representadas a seguir:

**leituraDados():** Responsável pela leitura do *RFID*, essa função espera a aproximação de uma *TAG* e quando isso acontece, realiza a conversão da mesma para o código padrão. Para os casos que a conversão tenha sido efetivada, a função de leitura é requisitada, sendo o código enviado como parâmetro. Além disso, para os casos de erro, um alerta é exibido no LCD do dispositivo.

**checkFingerprint():** Responsável pela leitura da impressão digital, essa função tem um funcionamento parecido com a função de leitura do *RFID*. Ao ser requisitada, o sensor é ativado e fica ativo até que uma digital seja lida. Com isso, para os casos em que ocorra um erro, é exibida uma mensagem no LCD. Quando a digital é reconhecida, a função verifica a posição que a mesma foi salva e realiza um vínculo com a *TAG RFID* cadastrada para o paciente em questão. Após isso, esse código é enviado para a função de exibição de dados como parâmetro.

Tendo essas funções, foi possível então implementar o *loop*, parte do código que sempre está ativo buscando alguma interação. Nessa parte, foi configurado o menu de seleção do modo de leitura composto por dois botões, sendo que o localizado a esquerda ao ser pressionado ativa o modo de leitura via *RFID* e o localizado a direita, o modo de leitura via impressão digital.

Por fim, programou-se a função *exibirdados* que recebe o código da *TAG RFID* como parâmetro. A partir disso, uma requisição com esse código é realizada na API e se o status retornado for igual a 200, tem-se um *Json* com os dados a serem exibidos no LCD. Assim,

realiza-se a separação desse documento em variáveis e também a exibição das mesmo no *display* por 10 segundos. Para os casos em que a API retorna um status diferente de 200, tem-se que ocorreu um erro na identificação do paciente e uma mensagem de alerta é exibida no LCD.

O dispositivo com as conexões realizadas e com a tela de menu exibida, pode ser visto na Figura 3.18. Além disso, todas as telas programadas podem ser visualizadas no Apêndice E.

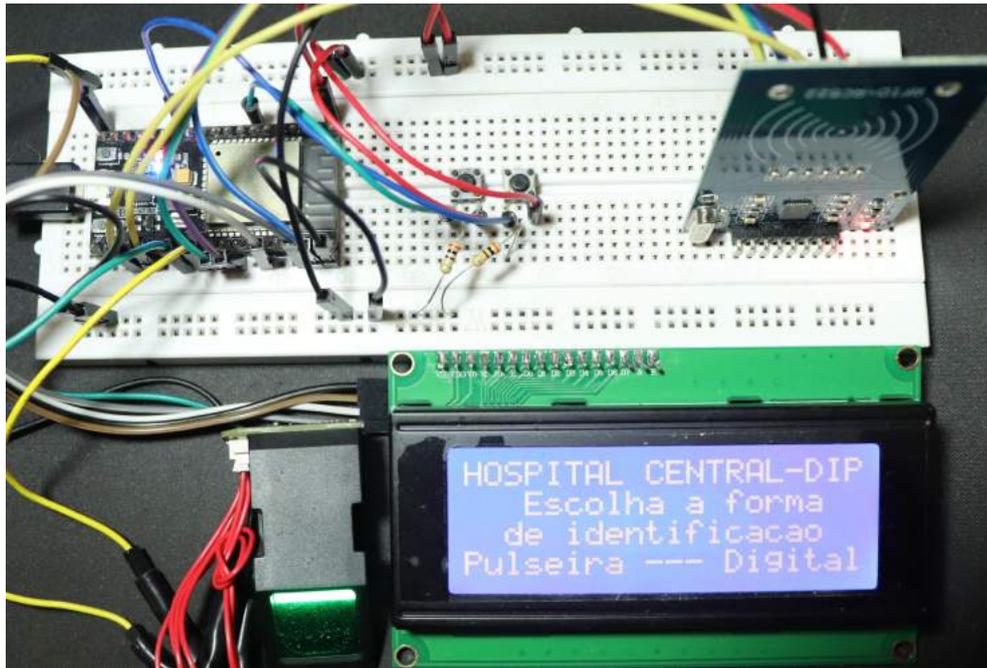


Figura 3.18: Foto das conexões

### Economia de Energia

Visando otimizar o consumo de energia, implementou-se uma funcionalidade nativa do ESP-32 chamada de *deep-sleep*. Esse modo desliga as CPUs, grande parte da memória RAM, todos os periféricos digitais com *clock* e as comunicações via *Wi-Fi* e *Bluetooth* deixando o microcontrolador praticamente desligado e consumindo apenas  $20\mu\text{A}$ .

Essa função foi configurada para ser ativa a partir de 2 minutos de inatividade do dispositivo permanecendo assim até que algum botão seja pressionado, e quando ativada, o único periférico que continua em funcionamento é o LCD, exibindo a mensagem mostrada na Figura 3.19.

### 3.5.2 Circuito de Alimentação

Com o intuito de facilitar a recarga da bateria do dispositivo, projetou-se um circuito que tem como objetivo principal, a conexão da bateria com o módulo de carga. Além disso, esse circuito foi projetado também para informar quando a bateria está com carga baixa,



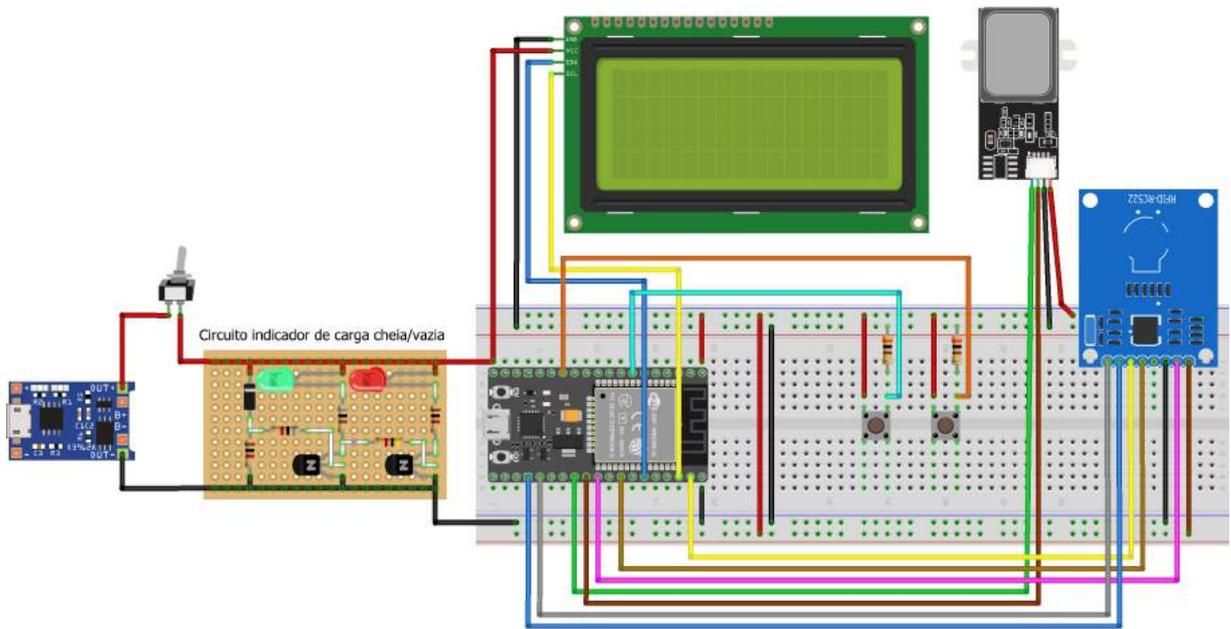


Figura 3.21: Esquemático final do dispositivo.

A partir dessa definição, pode-se então alocar os componentes restantes. Assim, posicionou-se os *pushbuttons* logo abaixo do LCD, de modo com que quando o menu for exibido, cada um fique na posição de seleção desejada e também os LEDs tanto de carga baixa quanto carga completa na parte superior esquerda.

Além disso, na parte lateral esquerda, posicionou-se a chave *on-off* na parte superior e a entrada de carregamento na parte inferior.

Na Figura 3.22 é possível ver o projeto final da case com as respectivas dimensões em milímetros.

Definiu-se também que a melhor forma de se construir a case seria com a produção de duas peças distintas, uma representando a base e outra o topo. Na Figura 3.23, tem-se representado o desenho de ambas as partes com a posição em que cada componente será fixado.

Vale ressaltar que as paredes da case tem 5mm de espessura.

## 3.6 Procedimentos de Uso

Pelo fato de a utilização do dispositivo ser voltada para ambientes hospitalares, os procedimentos de uso tem que ser bastantes rígidos. Com isso, definiu-se os seguintes passos para o uso:

1. O dispositivo deve ser esterilizado;
2. O operador deve ligar o dispositivo;



Figura 3.22: Desenho da case com as dimensões.

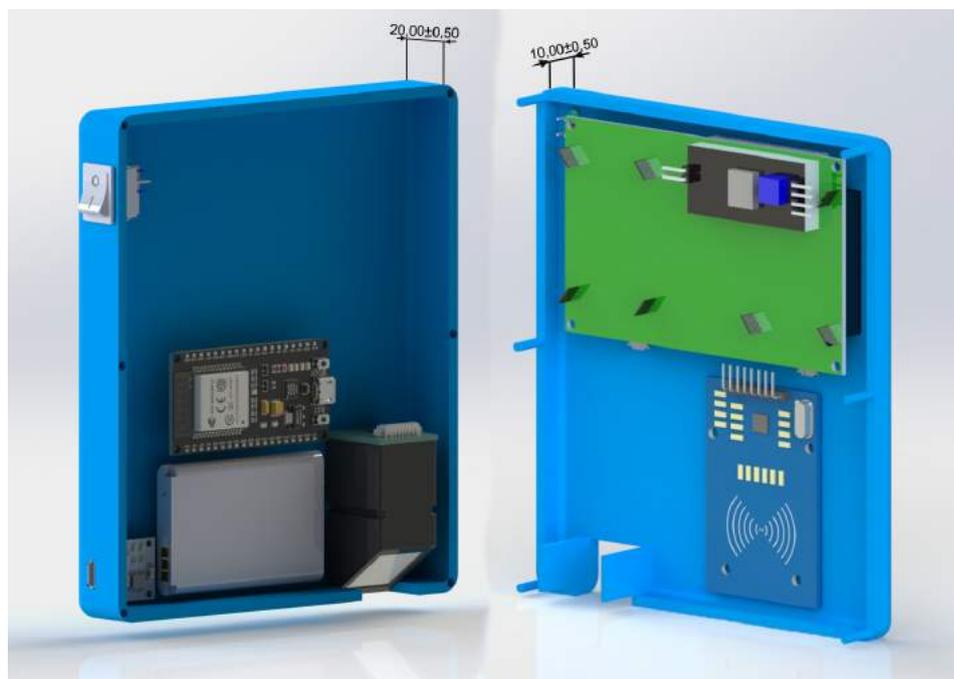


Figura 3.23: Desenho interno da case com a posição dos componentes.

3. A identificação do paciente deve ser realizada pela leitura do RFID ou da impressão digital;
4. As informações do paciente devem ser lidas na tela;
5. Caso o operador deseje ver mais informações sobre o paciente, deve abrir o aplicativo e

escolher a opção desejada;

6. Após o uso o aparelho deve ser desligado e novamente esterilizado.

### 3.7 Custos do Projeto

A Tabela 3.3 a seguir apresenta os custos para a elaboração deste trabalho de conclusão de curso.

Tabela 3.3: Tabela de custos (orçamento).

Produto	Custo (R\$)
<b>Eletrônica</b>	Já possuído
<i>ESP 32</i>	
<i>Display 20x4 com I2C embutido</i>	80,00
<i>Sensor RFID</i>	Já possuído
Sensor de impressão digital	Já possuído
Bateria	45,00
Módulo de Carga TP5046	25,00
Cabos diversos	Já possuído
Botão <i>On-Off</i>	5,00
Componentes diversos (resistores, leds, diodo e transistor)	15,00
<b>Aplicativo</b>	Já possuído
Celular com sistema operacional Android	
<b>Materiais</b>	
Filamento PETG	80,00
<b>Total</b>	250,00

É importante ressaltar que os recursos utilizados para aquisição dos componentes foram do próprio do autor.

## Considerações Finais

Neste capítulo as principais observações e conclusões realizadas no desenvolvimento do trabalho serão apresentadas, bem como as propostas de continuidade.

### 4.1 Conclusões

O presente trabalho teve como objetivo o desenvolvimento de um dispositivo de identificação de pacientes em ambientes hospitalares. O primeiro passo para alcançar tal fim, foi a realização de pesquisas acerca do tema, buscando entender sobre as necessidades de projeto, as normas a serem seguidas e evidenciando a importância do dispositivo a ser desenvolvido. Também nesse ponto, percebeu-se que existem poucos estudos relacionados ao desenvolvimento de alternativas no processo de identificação de pacientes.

Feito isso, os componentes a serem empregados no projeto foram definidos e partiu-se para uma revisão bibliográfica, evidenciando um grande avanço tecnológico nos mesmos. Complementando essa etapa, os fundamentos teóricos referentes aos periféricos escolhidos foram descritos, esclarecendo as necessidades operacionais e forma de funcionamento.

Posteriormente, com todas as informações obtidas na produção do referencial teórico, pode-se realizar o desenvolvimento, começando com o aplicativo. Nessa etapa, visando a fluidez do processo, projetou-se de forma inicial todas as telas e somente com isso pronto, as funcionalidades foram implementadas. A principal dificuldade encontrada nessa parte do projeto, relacionou-se com a exibição dos dados dos pacientes vinculados a tela de pesquisa.

Em sequência, com o *app* validado, partiu-se para o desenvolvimento do dispositivo em si. Nessa etapa, percebeu-se a necessidade de se realizar algumas adaptações para o funcionamento do sistema, como a construção de uma API que tornasse possível a comunicação entre o dispositivo em si e a base de dados e também o vínculo entre a impressão digital e a *TAG RFID*, para que a identificação fosse feita. Além disso, devido as grandes peculiaridades do ambiente de utilização do dispositivo, encontrou-se uma grande dificuldade no projeto da case, tanto no posicionamento dos componentes quanto na definição do material de construção.

Por fim, tem-se que o projeto foi desenvolvido de acordo com o cronograma proposto e alcançou resultados satisfatórios e interessantes, principalmente devido ao baixo custo de investimento. Além disso, o trabalho permitiu a utilização das capacidades e conhecimentos obtidos durante o curso de Engenharia Mecatrônica com a interdisciplinaridade das áreas.

## 4.2 Propostas de Continuidade

Com as percepções e dificuldades encontradas durante o decorrer do trabalho, tem-se como propostas de continuidade:

- Substituição do modelo do microcontrolador ESP utilizado, para um que possibilite a inclusão de uma antena ao sistema, assim, aumentando a área de trabalho do dispositivo;
- Substituição do modelo de *display* utilizado para um com uma maior área de exibição, como por exemplo um *display* LCD com Transistor de Película Fina *Touch Shield* que possui 3.5 polegadas além de ser sensível ao toque. Com isso, mais informações poderão ser exibidas no momento da identificação.
- Alteração na base de dados utilizada, visando simular com mais precisão o banco de dados utilizado em instituições hospitalares e também possibilitando o armazenamento das impressões digitais para consultas do dispositivo.
- Construção de outros dispositivos para validar o funcionamento da conexão com o aplicativo.
- Implementação do cadastro da rede *Wi-Fi* no dispositivo.
- Construção de uma página web que possibilite o cadastro tanto de pacientes quanto de usuários.

## Código da REST API

```
1  const functions = require("firebase-functions");
2  const admin = require("firebase-admin");
3  const express = require("express");
4  const cors = require("cors");
5
6  admin.initializeApp();
7  const db = admin.firestore().collection("pacientes");
8
9  const app = express();
10 app.use(cors({}));
11
12 app.route("/test").get((req, res) => {
13   res.status(200).send({ message: "Teste valido" });
14 });
15
16 app.route("/tcc").post((req, res) => {
17   const { rfid } = req.body;
18
19   if (!rfid) {
20     return res.status(400).send({ message: "RFID nao informado" });
21   }
22
23   let paciente = [];
24   db.where("rfid", "==", rfid)
25     .get()
26     .then((docs) => {
27       docs.forEach((doc) => {
28         paciente.push({
29           id: doc.id,
30           nome: doc.data().nome,
31           quarto: doc.data().quarto,
32           diag: doc.data().diag,
33           nasc: doc.data().nasc,
34           nomemae: doc.data().nomemae
35         });
36       });
37     });
```

---

```
38     db.doc(paciente[0].id)
39       .update({
40         exhibir: "sim",
41         dataexib: new Date()
42       })
43       .then(() => {
44         return res.status(200).json(paciente);
45       });
46     })
47     .catch((err) => {
48       return res.status(400).send({ message: `erro. ${err}` });
49     });
50 });
51
52 exports.api = functions.https.onRequest(app);
53 }
```

Listing A.1: Código fonte da API em JavaScript

# Apêndice **B**

## Código do Aplicativo

O código do aplicativo pode ser visto no seguinte link:

<https://github.com/galoduda/dip-tcc/blob/main/lib/main.dart>

## Telas do Aplicativo

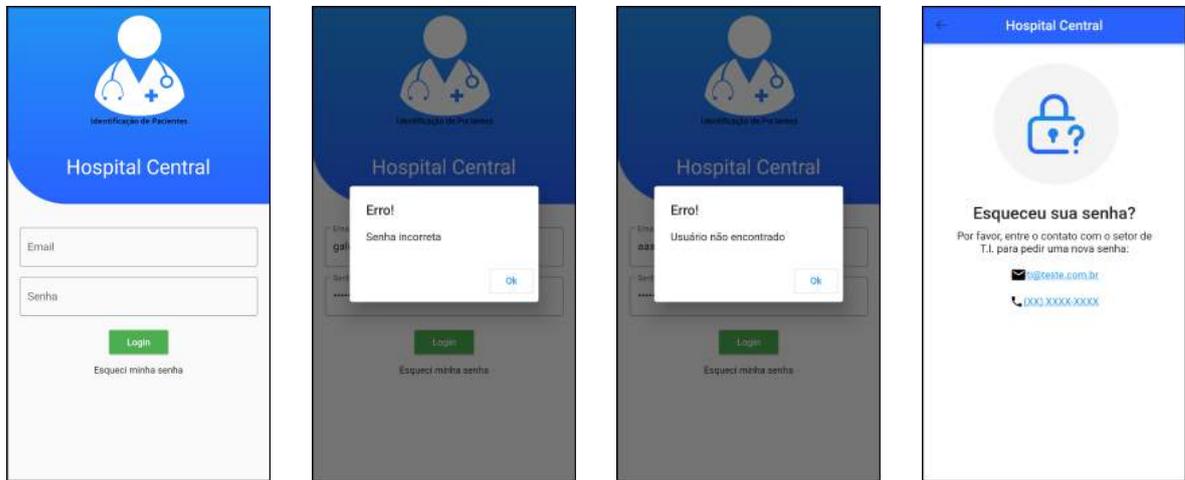


Figura C.1: Telas referentes ao login



Figura C.2: Tela do menu

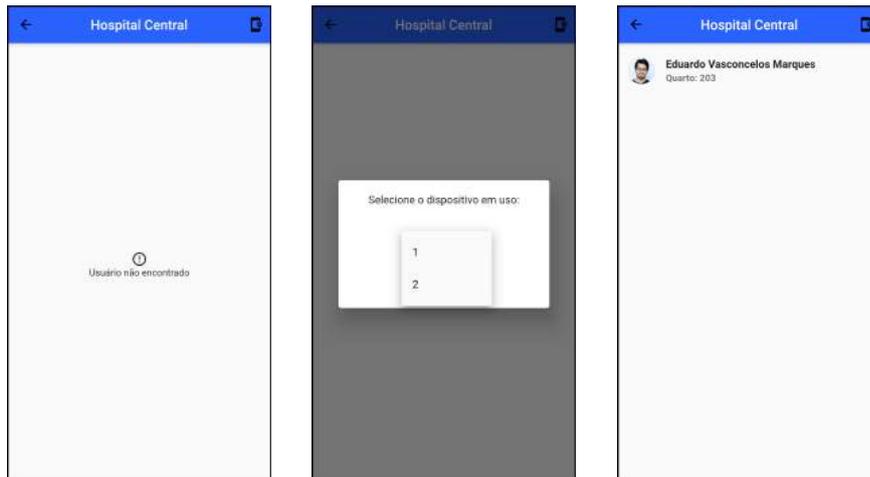


Figura C.3: Telas referentes ao recebimento de dados

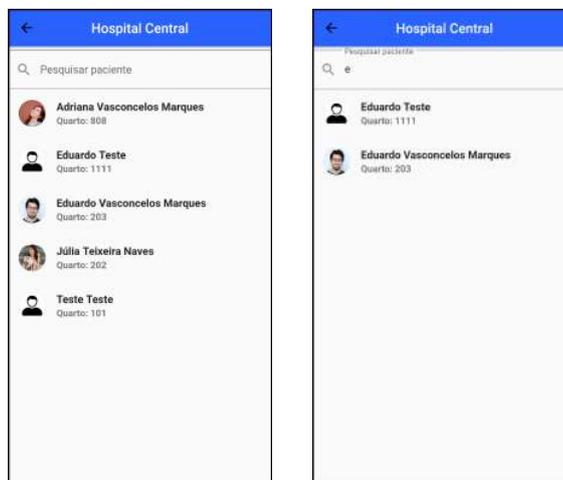


Figura C.4: Telas referentes a pesquisa de pacientes

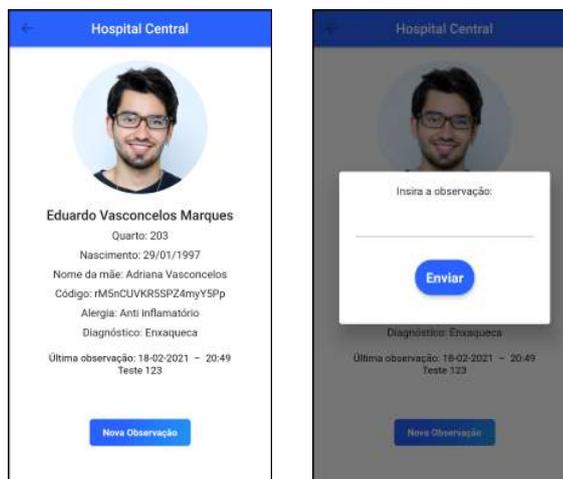


Figura C.5: Telas referentes a exibição dos dados do pacientes

# Apêndice D

## Código do Microcontrolador

```
1 #include <MFRC522.h> //biblioteca responsavel pela comunicacao com
    o modulo RFID-RC522
2 #include <SPI.h> //biblioteca para comunicacao do barramento SPI
3 #include <Wire.h>
4 #include <LiquidCrystal_I2C.h>
5 #include <WiFi.h>
6 #include <Adafruit_Fingerprint.h>
7 #include <ArduinoJson.h>
8 #include <HttpClient.h>
9
10 //Fingerprint
11 const uint32_t password = 0x0;
12 Adafruit_Fingerprint fingerprintSensor = Adafruit_Fingerprint(&
    Serial2, password);
13
14 //DefinindoRFID
15 #define SS_PIN      2
16 #define RST_PIN    15
17
18 #define SIZE_BUFFER      18
19 #define MAX_SIZE_BLOCK  16
20
21 #define butRFID      32
22 #define butDIG      12
23
24 //DefinindoWIFI
25 #define WIFI_NETWORK  "Vasconcelos"
26 #define WIFI_PASSWORD "nina1234"
27
28 #define mask 0x100001000
29
30 int lcdColumns = 20;
31 int lcdRows = 4;
32
33 //esse objeto 'chave' e utilizado para autenticacao
34 MFRC522::MIFARE_Key key;
35 //codigo de status de retorno da autenticacao
```

```

36 MFRC522::StatusCode status;
37
38 // Definicoes pino modulo RC522
39 MFRC522 mfrc522(SS_PIN, RST_PIN);
40 LiquidCrystal_I2C lcd(0x27,20,4);
41 unsigned long timer = 0;
42
43 void setup() {
44     // Inicia a serial
45     Serial.begin(9600);
46     SPI.begin(); // Init SPI bus
47
48     pinMode(butRFID, INPUT);
49     pinMode(butDIG, INPUT);
50
51     lcd.init();
52     lcd.backlight();
53
54     // Inicia MFRC522
55     mfrc522.PCD_Init();
56
57     //Conecta no WiFi
58     WiFi.begin(WIFI_NETWORK, WIFI_PASSWORD);
59     while(WiFi.status() != WL_CONNECTED){
60         delay(500);
61         Serial.print("Conectado no WiFi com IP: ");
62         Serial.println(WiFi.localIP());
63         lcd.clear();
64         lcd.setCursor(0, 0);
65         lcd.print("HOSPITAL CENTRAL-DIP");
66         lcd.setCursor(5, 2);
67         lcd.print("Conectando");
68         lcd.setCursor(4, 3);
69         lcd.print("ao Wi-Fi...");
70     }
71     delay(1000);
72     lcd.clear();
73     //Inicializa o sensor de digitais
74     setupFingerprintSensor();
75     esp_sleep_enable_ext1_wakeup(mask, ESP_EXT1_WAKEUP_ANY_HIGH);
76 }
77
78 void setupFingerprintSensor()
79 {
80     //Inicializa o sensor
81     fingerprintSensor.begin(57600);
82
83     //Verifica se a senha esta correta
84     if(!fingerprintSensor.verifyPassword()){
85         //Se chegou aqui significa que a senha esta errada ou o sensor
            esta problemas de conexao

```

```

86     Serial.println(F("Nao foi possivel conectar ao sensor.
      Verifique a senha ou a conexao"));
87     lcd.clear();
88     lcd.setCursor(0, 0);
89     lcd.print("HOSPITAL CENTRAL-DIP");
90     lcd.setCursor(8, 2);
91     lcd.print("ERRO");
92     lcd.setCursor(4, 3);
93     lcd.print("no sensor de digital");
94     while(true);
95 }
96 }
97
98 void loop(){
99
100     //int opcao = menu();
101     int opcao;
102     //unsigned long timer=millis();
103     Serial.println("Escolha uma opcao");
104     int botaoRfid = digitalRead(butRFID);
105     int botaoDig = digitalRead(butDIG);
106
107     //lcd.clear();
108     lcd.setCursor(0, 0);
109     lcd.print("HOSPITAL CENTRAL-DIP");
110     lcd.setCursor(3, 1);
111     lcd.print("Escolha a forma");
112     lcd.setCursor(2, 2);
113     lcd.print("de identificacao");
114     lcd.setCursor(0, 3);
115     lcd.print("Pulseira --- Digital");
116
117     if(botaoRfid == HIGH){
118         timer =millis();
119         Serial.println("botao RFID pressionado");
120
121         lcd.clear();
122         lcd.setCursor(0, 0);
123         lcd.print("HOSPITAL CENTRAL-DIP");
124         lcd.setCursor(0, 1);
125         lcd.print("");
126         lcd.setCursor(3, 2);
127         lcd.print("Aproxime a TAG");
128         lcd.setCursor(5, 3);
129         lcd.print("no sensor");
130
131         while (mfrc522.PICC_IsNewCardPresent() != true){}
132
133         if ( ! mfrc522.PICC_ReadCardSerial()){
134             return;
135         }

```

```

136     leituraDados();
137
138     // instrui o PICC quando no estado ACTIVE a ir para um
139     estado de "parada"
140     mfrc522.PICC_HaltA();
141     // "stop" a encriptacao do PCD, deve ser chamado apos a
142     comunicacao com autenticao, caso contrario novas
143     comunicacoes nao poderao ser iniciadas
144     mfrc522.PCD_StopCrypto1();
145     timer =millis();
146     }
147     else if(botaoDig == HIGH){
148     Serial.println("botao DIGITAL pressionado");
149     checkFingerprint();
150     timer =millis();
151     return;
152     }
153
154     else if((millis()-timer) >= 120000){
155     Serial.println("TEMPO ESGOTADO");
156     Serial.println(timer);
157     Serial.println(millis());
158     Serial.println(millis()-timer);
159     lcd.clear();
160     lcd.setCursor(0, 0);
161     lcd.print("HOSPITAL CENTRAL-DIP");
162     lcd.setCursor(0, 2);
163     lcd.print("Economia de economia");
164     lcd.setCursor(1, 3);
165     lcd.print("de bateria ativado");
166     delay(2000);
167     esp_deep_sleep_start();
168     return;
169     }
170 }
171
172 //faz a leitura dos dados do cartao/tag
173 void leituraDados(){
174
175     //imprime os detalhes tecnicos do cartao/tag
176     mfrc522.PICC_DumpDetailsToSerial(&(mfrc522.uid));
177
178     //Prepara a chave - todas as chaves estao configuradas para
179     FFFFFFFFh (Padrao de fabrica).
180     for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
181
182     //buffer para colocar os dados ligados
183     byte buffer[SIZE_BUFFER] = {0};
184
185     //bloco que faremos a operacao
186     byte bloco = 1;

```

```

183 byte tamanho = SIZE_BUFFER;
184
185 //faz a autenticao do bloco que vamos operar
186 status = mfrc522.PCD_Authenticate(MFRC522::
    PICC_CMD_MF_AUTH_KEY_A, bloco, &key, &(mfrc522.uid)); //line
    834 of MFRC522.cpp file
187 if (status != MFRC522::STATUS_OK) {
188     Serial.print(F("Authentication failed: "));
189     Serial.println(mfrc522.GetStatusCodeName(status));
190     lcd.clear();
191     lcd.setCursor(0, 0);
192     lcd.print("HOSPITAL CENTRAL-DIP");
193     lcd.setCursor(0, 1);
194     lcd.print("");
195     lcd.setCursor(6, 2);
196     lcd.print("Erro");
197     lcd.setCursor(3, 3);
198     lcd.print("de leitura");
199     delay(5000);
200     delay(5000);
201     return;
202 }
203
204 //faz a leitura dos dados do bloco
205 status = mfrc522.MIFARE_Read(bloco, buffer, &tamanho);
206 if (status != MFRC522::STATUS_OK){
207     Serial.print(F("Reading failed: "));
208     Serial.println(mfrc522.GetStatusCodeName(status));
209     lcd.clear();
210     lcd.setCursor(0, 0);
211     lcd.print("HOSPITAL CENTRAL-DIP");
212     lcd.setCursor(0, 1);
213     lcd.print("");
214     lcd.setCursor(6, 2);
215     lcd.print("Erro");
216     lcd.setCursor(3, 3);
217     lcd.print("de leitura");
218     delay(5000);
219     return;
220 }
221 else{}
222
223 Serial.print(F("\nDados bloco ["));
224 Serial.print(bloco);Serial.print(F(": "));
225 //String uid= mfrc522.uid;
226 String userid;
227 for (byte i = 0; i < mfrc522.uid.size; i++) {
228     //Serial.print(mfrc522.uid.uidByte[i], HEX);
229     userid += String(mfrc522.uid.uidByte[i], HEX);
230 }
231 Serial.println(userid);

```

```

232     exibirdados(userid);
233     //delay(5000);
234
235     //imprime os dados lidos
236     for (uint8_t i = 0; i < MAX_SIZE_BLOCK; i++){
237         Serial.write(buffer[i]);
238     }
239     Serial.println(" ");
240 }
241
242
243 //Verifica se a digital esta cadastrada
244 void checkFingerprint(){
245     Serial.println(F("Encoste o dedo no sensor"));
246     lcd.clear();
247     lcd.setCursor(0, 0);
248     lcd.print("HOSPITAL CENTRAL-DIP");
249     lcd.setCursor(0, 1);
250     lcd.print("");
251     lcd.setCursor(3, 2);
252     lcd.print("Encoste o dedo");
253     lcd.setCursor(5, 3);
254     lcd.print("no sensor");
255
256     //Espera ate pegar uma imagem valida da digital
257     while (fingerprintSensor.getImage() != FINGERPRINT_OK);
258
259     //Converte a imagem para o padrao que sera utilizado para
260     //verificar com o banco de digitais
261     if (fingerprintSensor.image2Tz() != FINGERPRINT_OK){
262         //Se chegou aqui deu erro, entao abortamos os proximos passos
263         Serial.println(F("Erro image2Tz"));
264         delay(2000);
265         return;
266     }
267
268     //Procura por este padrao no banco de digitais
269     if (fingerprintSensor.fingerFastSearch() != FINGERPRINT_OK){
270         //Se chegou aqui significa que a digital nao foi encontrada
271         Serial.println(F("Digital nao encontrada"));
272         lcd.clear();
273         lcd.setCursor(0, 0);
274         lcd.print("HOSPITAL CENTRAL-DIP");
275         lcd.setCursor(0, 1);
276         lcd.print("");
277         lcd.setCursor(7, 2);
278         lcd.print("Digital");
279         lcd.setCursor(3, 3);
280         lcd.print("nao encontrada");
281         delay(2000);
282         return;

```

```

282 }
283
284 //Se chegou aqui a digital foi encontrada
285 //Mostramos a posicao onde a digital estava salva e a confianca
286 //Quanto mais alta a confianca melhor
287 Serial.print(F("Digital encontrada com confianca de "));
288 Serial.print(fingerprintSensor.confidence);
289 Serial.print(F(" na posicao "));
290 Serial.println(fingerprintSensor.fingerID);
291 if (fingerprintSensor.fingerID == 2){
292     Serial.println(F("Ola Eduardo"));
293     exibirdados("705a7ea6");
294 }
295 if (fingerprintSensor.fingerID == 1){
296     Serial.println(F("Ola Teste"));
297     exibirdados("1111");
298 }
299 if (fingerprintSensor.fingerID == 3){
300     Serial.println(F("Ola Teste2"));
301     exibirdados("4a101fa3");
302 }
303 }
304
305 //Exibir dados
306 void exibirdados(String teste){
307     String IDpaciente = teste;
308     if(WiFi.status() == WL_CONNECTED){
309         HTTPClient http;
310         http.begin("https://us-central1-appdip-96c2a.cloudfunctions.
311             net/api/tcc");
312         http.addHeader("Content-Type", "application/json");
313         StaticJsonDocument<200> docRFID;
314         docRFID["rfid"] = IDpaciente;
315         Serial.println("IDpaciente: " + String(IDpaciente));
316         char buffer[20];
317         serializeJson(docRFID,buffer);
318         int cod = http.POST(buffer);
319         Serial.println(buffer);
320         Serial.println("codigo:"+String(cod));
321
322         if(cod == 200){
323             String dados = http.getString();
324             Serial.println(dados);
325
326             char jsonexibir[500];
327             dados.replace("\n", "");
328             dados.trim();
329             dados.remove(0,1);
330             dados.toCharArray(jsonexibir,500);
331
332             Serial.println(jsonexibir);

```

```

332
333 StaticJsonDocument<192> doc;
334 deserializeJson(doc, jsonexibir);
335
336     const char* nome = doc["nome"];
337     int quarto = doc["quarto"];
338     const char* nasc = doc["nasc"];
339     const char* diag = doc["diag"];
340     String nomemae=doc["nomemae"];
341     nomemae.remove(16);
342
343     lcd.clear();
344     lcd.setCursor(0, 0);
345     lcd.print(nome);
346     lcd.setCursor(0, 1);
347     lcd.print("AP" + String(quarto));
348     lcd.setCursor(10, 1);
349     lcd.print(nasc);
350     lcd.setCursor(0, 2);
351     lcd.print("MAE:" + String(nomemae));
352     lcd.setCursor(0, 3);
353     lcd.print("DIAG:" + String(diag));
354
355     delay(5000);
356     lcd.clear();
357 }
358 else{
359     Serial.println("Paciente nao identificado");
360     lcd.clear();
361     lcd.setCursor(0, 0);
362     lcd.print("HOSPITAL CENTRAL-DIP");
363     lcd.setCursor(0, 1);
364     lcd.print("");
365     lcd.setCursor(6, 2);
366     lcd.print("Paciente");
367     lcd.setCursor(3, 3);
368     lcd.print("nao encontrado");
369     delay(10000);
370 }
371     http.end();
372 }
373 }

```

Listing D.1: Código do microcontrolador

## Telas do Dispositivo



Figura E.1: Conectando ao wi-fi



Figura E.2: Menu



Figura E.3: Esperando leitura



Figura E.4: Exibição dos dados



Figura E.5: Paciente não encontrado



Figura E.6: Modo de economia de energia

## Referências

- 3DLAB. *4 dicas para escolha ideal do filamento para impressora 3D*. [Online; acesso em 13-Outubro-2020], <https://3dlab.com.br/como-escolher-o-filamento-para-impressora-3d/>.
- ALIBABA. [Online; acesso em 17-Março-2021], <https://cutt.ly/xz3rj2P>.
- ANDRADE, A. P. de. *O que é Flutter?* [Online; acesso em 14-Outubro-2020], <https://www.treinaweb.com.br/blog/o-que-e-flutter/>.
- ANVISA, A. *Manual de Segurança no Ambiente Hospitalar*. Brasília: [s.n.], 2003.
- ANVISA, A. *Protocolo de Identificação do Paciente*. [Online; acesso em 25-Fevereiro-2020], <https://www20.anvisa.gov.br/segurancadopaciente/index.php/publicacoes/item/identificacao-do-paciente>.
- ARDUINOPORTUGAL. [Online; acesso em 05-Outubro-2020], <https://www.arduinoportugal.pt/os-reles-estado-solido/>.
- AYCOCK, S. *A história dos microcontroladores*. [Online; acesso em 30-Setembro-2020], [https://www.ehow.com.br/controlador-smbus-fatos\\_94209/](https://www.ehow.com.br/controlador-smbus-fatos_94209/).
- BANGGOOD. [Online; acesso em 25-Outubro-2020], <https://cutt.ly/7z0nWkt>.
- BARBOSA, A. F. *Pesquisa sobre o uso das tecnologias de informação e comunicação no Brasil: tic domicílios e empresas 2016*. São Paulo: Pearson Makron, 2016. (Comitê Gestor da Internet no Brasil.).
- BARRA, D. C. C. *et al.* Métodos para Desenvolvimento de Aplicativos Móveis em Saúde: revisão integrativa da literatura. *Texto contexto - Enfermagem*, [S.l.], n.4 vol.26, 2018.
- BRASIL, M. da Saúde do; FIOCRUZ, F.; ANVISA, A. *Documento de referência para o Programa Nacional de Segurança do Paciente*. Brasília: [s.n.], 2014.

- BRITO, B. *Padrão REST*. [Online; acesso em 15-Março-2021], <https://www.brunobrito.net.br/padrao-rest/>.
- BRITO, M. d. F. P. *Avaliação do Processo de Identificação do Paciente em Serviços de Saúde*. 2015. Dissertação (Mestrado em Ciência da Computação) — Universidade de São Paulo, Ribeirão Preto.
- CAMARA, R. *Protocolo I2C*. [Online; acesso em 29-Outubro-2020], <http://www.univasf.edu.br/~romulo.camara/novo/wp-content/uploads/2013/11/Barramento-e-Protocolo-I2C.pdf>.
- CARVALHO, V. B. *Desenvolvimento e Teste de um Monitor de Barramento I2C para Proteção Contra Falhas Transientes*. 2016. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul, Porto Alegre.
- CASSETTARI, V.; SILVEIRA, I. R. da. *Manual para a Prevenção das Infecções Relacionadas à Assistência*. São Paulo: [s.n.], 2018. (CCIH/HU-USP).
- CHENG, F. *Build Mobile Apps with Ionic 2 and Firebase: hybrid mobile app development*. 1.ed. [S.l.]: Apress, 2017.
- CUNHA, A. [Online; acesso em 07-Novembro-2020], <https://www.embarcados.com.br/rfid-etiquetas-com-eletronica-de-ponta/>.
- DEVMEDIA. *Guia de Node.js*. [Online; acesso em 16-Março-2021], <https://www.devmedia.com.br/guia/node-js/40312>.
- ECRI. *Patient Identification Lessons Learned from ECRI Institute's 2016 Deep Dive*. [Online; acesso em 13-Dezembro-2020], [https://www.ecri.org/components/HRCAlerts/Pages/HRCAlerts092816\\_PatientID.aspx](https://www.ecri.org/components/HRCAlerts/Pages/HRCAlerts092816_PatientID.aspx).
- ENGST, A.; FLEISHMAN, G. *Kit do Iniciante em Redes Sem Fio*. 2.ed. São Paulo: Pearson Makron, 2005.
- EVANS, M.; NOBLE, J.; HOCHENBAUM, J. *Arduino em Ação*. 1.ed. São Paulo: Novatec, 2013.
- FERNANDES, A. *O que é API? Entenda de uma maneira simples*. [Online; acesso em 15-Março-2021], <https://blog.vertigo.com.br/o-que-e-api-entenda-de-uma-maneira-simples/>.
- FERRONI, E. et al. A Plataforma Arduíno e suas Aplicações. *Revista UIIPS*, [S.l.], n.3, 2015.
- FLING, B. *Mobile Design and Development*. Sebastopol: O'Reilly, 2009.

- HADDAD, C. R.; RIZZOTTO, F. H.; MALDONADO, M. U. Revisão Estruturada da Literatura sobre RFID e suas Aplicações na Cadeia de Suprimentos. *Revista Espacios*, [S.l.], n.8 vol. 37, 2016.
- HOA, D. N. T. [Online; acesso em 02-Novembro-2020], <https://toilamkythuat.com/giao-tiep-i2c-trong-vi-dieu-khien-35.html>.
- HOFFMEISTER, L. V.; MOURA, G. M. S. S. de. Uso de pulseiras de identificação em pacientes internados em um hospital universitário. *Revista Latino-Americana de Enfermagem*, [S.l.], n.1 vol. 23, 2015.
- KALINSKY, R. *Introduction to I2C*. [Online; acesso em 29-Outubro-2020], <https://www.embedded.com/introduction-to-i2c/>.
- 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.
- MALTONI, D. *et al. Handbook of Fingerprint Recognition*. 2.ed. [S.l.]: Springer, 2009.
- MÁRCICO, J. E. *Sistema Informatizado de identificação*. 2002. Dissertação (Mestrado em Ciência da Computação) — Faculdade de Tecnologia de Taquaritinga, Taquaritinga.
- MAZI, R. C.; JÚNIOR, A. D. P. Identificação Biométrica Através da Impressão Digital Usando Redes Neurais Artificiais. *Anais do XIV ENCITA 2008*, [S.l.], 2009.
- MERCADO LIVRE. [Online; acesso em 17-Março-2021], <https://cutt.ly/gz3t7HT>.
- NASSAR, V.; VIEIRA, M. A rastreabilidade da medicação do paciente no hospital com a tecnologia RFID. *Revista de Ciencia y tecnología*, [S.l.], n.32, 2019.
- NETO, G. V. *Serviços de assistência direta ao paciente*. 1.ed. Rio de Janeiro: Editora Guanabara Koogan, 2011. (Gestão em Saúde).
- NOGUEIRA, D. [Online; acesso em 12-Outubro-2020], <https://autocorerobotica.blog.br/utilizando-a-tecnologia-rfid-com-arduino/>.
- NPSA, N. *Wristbands for hospital inpatients improves safety. Safer Practice Notice*. [Online; acesso em 25-Fevereiro-2020], <https://www.improvement.nhs.uk/resources/learning-from-patient-safety-incident/>.

- OASES. [Online; acesso em 02-Novembro-2020], <http://m.oaseschina.net/rfid-wristband/paper-rfid-wristband/rfid-disposable-dupoint-paper-wristbands.html>.
- PEDROSO, M. C.; ZWICKER, R.; SOUZA, C. A. de. Adoção de RFID no Brasil: um estudo exploratório. *Revista de Administração Mackenzie*, [S.l.], n.v.10, n.1, 2009.
- PIRES, J. *O que é API? REST e RESTful? Conheça as definições e diferenças!* [Online; acesso em 15-Março-2021], <https://becode.com.br/o-que-e-api-rest-e-restful/>.
- REI, J. *RFID Versus Código de Barras: da produção à grande distribuição*. 2010. Dissertação (Mestrado em Ciência da Computação) — Faculdade de Engenharia da Universidade do Porto, Porto.
- REIS, F. dos. [Online; acesso em 05-Outubro-2020], <http://www.bosontreinamentos.com.br/redes-wireless/arquiteturas-de-redes-locais-sem-fio-wi-fi/>.
- 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 05-Outubro-2020], <https://www.robocore.net/loja/iot/esp32-wifi-bluetooth>.
- SAFESST. *As Normas Regulamentadoras em Hospitais*. [Online; acesso em 15-Outubro-2020], <https://blog.safesst.com.br/os-hospitais-e-as-normas-regulamentadoras/>.
- SAINSMART. [Online; acesso em 25-Outubro-2020], <https://www.sainsmart.com/products/20x4-iic-i2c-twi-lcd-module>.
- SILVA, M. V. da. *Detecção de Impressões Digitais Falsas no Reconhecimento Biométrico de Pessoas*. 2015. Dissertação (Mestrado em Ciência da Computação) — Universidade Estadual Paulista, Bauru.
- SILVA, W. C. S. da. *Aplicações Móveis Nativas com React Native e Firebase: um estudo de caso*. 2018. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal do Maranhão, São Luis.
- SOUSA, F. P. de. *Criação de framework REST/HATEOAS Open Source para desenvolvimento de APIs em Node.js*. 2015. Dissertação (Mestrado em Ciência da Computação) — Universidade do Porto, Porto - Portugal.
- SOUZA, V. A. *A história e as diferenças entre um microcontrolador e um microprocessador*. [Online; acesso em 30-Setembro-2020], [https://www.cerne-tec.com.br/Artigo3\\_Historia.pdf](https://www.cerne-tec.com.br/Artigo3_Historia.pdf).

- STEFANUTO, I.; SANTOS, J. A. M. dos; TORRES, C. T. Evolução das Redes Sem Fio: comparativo entre wi-fi e bluetooth. *Caderno de Estudos Tecnológicos*, [S.l.], n.v.4, n.1, 2016.
- TASE, T. H. *et al.* Identificação do paciente nas organizações de saúde: uma reflexão emergente. *Revista Gaúcha de Enfermagem*, [S.l.], n.3 vol. 34, 2013.
- TASE, T. H.; TRONCHIN, D. M. R. Sistemas de identificação de pacientes em unidades obstétricas e a conformidade das pulseiras. *Acta Paulista de Enfermagem*, [S.l.], n.4 vol. 28, 2015.
- TAVARES, M. C. *Monitor de atividade cerebral baseado em microcontrolador*. 1997. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Santa Catarina. Centro Tecnológico, Florianópolis, SC.
- VLADCONTROL. [Online; acesso em 25-Outubro-2020], <http://www.vladcontrol.com.br/sensores/sensor-de-impressao-digital-fpm10a-dy50/>.
- YARONGTECH. [Online; acesso em 02-Novembro-2020], <https://cutt.ly/lz0n1b5>.
- ZIMMERMANN, D. [Online; acesso em 14-Outubro-2020], <https://www.filipeflop.com/blog/qual-modulo-wifi-esp32-e-ideal-para-meu-projeto/esp32-basico-pinout/>.