



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

## **Sistema de Gerenciamento de Estoque Usando Identificação por Radiofrequência**

Caio Ítallo Souza Mendes e Rodrigo Bornéo de Oliveira

Monografia apresentada como requisito parcial  
para conclusão do Curso de Computação — Licenciatura

Orientadora  
Prof.a Dr.a Edna Dias Canedo

Brasília  
2022



# Dedicatória

*Eu, Rodrigo Bornéo dedico esse trabalho a George Lucas, criador de Star Wars, que idealizou um universo vasto com personagens incríveis e histórias inspiradoras, as quais que por diversas vezes em minha vida serviram de motivação para superar os desafios. Dedico também ao Carl Edward Sagan, cientista, físico, biólogo, astrônomo, astrofísico, cosmólogo, escritor, divulgador científico e ativista norte-americano, pois sua obra moldou minha forma de pensar, enxergar e compreender o universo que nos cerca.*

# Dedicatória

*Dedico a esse trabalho a todos que de alguma forma me ajudaram nessa caminhada através de incetivos palavras de carinhos ou até mesmo me desmotivando o não acreditando na minha capacidade que de algum modo isso me fortaleceu e me fez chegar onde cheguei...*



# Agradecimentos

Agradeço primeiramente ao nosso Senhor que nos ajudou, guiou e iluminou até esse ponto. Agradeço ao meu pai Capitão de Mar e Guerra (T) Paulo Santos de Oliveira e minha mãe Sra. Tânia Maria de Paula Bornéo que sempre me apoiaram e forneceram todo o suporte necessário para poder chegar até este ponto no qual hoje me encontro. Agradeço a minha irmã Bianca Bornéo que por diversas vezes me levantou quando estive caído e corrigiu meu rumo quando estive próximo a me perder. Agradeço ao meu amigo e companheiro de curso Caio Ítallo, o qual lutou muitas batalhas e enfrentou muitos desafios ao meu lado. Agradeço aos amigos que ajudaram durante essa trajetória, destacando dois em especial, Lucas Souza (vulgo Goiano) e Ícaro Rezende, os quais sempre estiveram em prontidão e não mediram esforços para me auxiliar. Agradeço aos meus porquinho-da-india Biscoito II e Sorvete por existirem. Agradeço a mim mesmo por não ter desistido, por ter persistido, por ter tido força e coragem para chegar até a conclusão curso e estar determinado a ir além. Por último mas não menos importante, agradeço a todos aqueles que duvidaram, desestimulam ou até mesmo agiram contra mim por mero capricho ou qualquer outro motivo. Um forte abraço a todos e que a Força esteja com vocês.

# Agradecimentos

Eu Caio Ítallo Souza Mendes, agradeço a Deus que sempre me fortaleceu dando esperança e força para chegar ao fim e concluir esse ciclo. Agradeço também aos meus pais Cleudson Barros e Ivanice Mendes e ao meu irmão Caiã Iuri que sempre me incentivaram acreditando em mim, quando nem eu mesmo acreditava, me guiando nas minhas decisões, escolhas e em busca dos meus objetivos, que sem eles nada disso era possível e essa realização não passaria de um sonho. Quero agradecer especialmente aos meus avós Cairo Humberto e Ivonete Mendes que me estimulam a crescer constantemente me fazendo a ser o melhor em tudo que eu faço e fazer com amor e por me inspirar através de suas histórias de vida. Aos meus tios Cleudirene Barros e Júlio Cesar juntamente com as minhas primas Maria Rita e Maria Júlia por me apoiarem e incentivarem a sempre crescer e ir além nos meus estudos. Ao meu amigo de curso Rodrigo Bornéo no qual juntos enfrentamos muitos desafios e superamos várias barreiras encontradas pelo caminho durante nossa graduação com seu otimismo de sempre acreditar de que éramos capazes de ir longe, além de viabilizar e acreditar no potencial desse estudo sendo meu companheiro de TCC. Ao meu amigo Icaro Rezende que foi uma das minhas grandes referências durante a minha graduação que me ajudou bastante não só na faculdade levando nossa amizade além. Aos meus colegas da empresa júnior de computação CJR que foram super importantes no meu crescimento pessoal e profissional com todos aprendizados e experiências adquiridas. Ao Seu Luiz do Brasília RFID, apoiador diretamente do nosso projeto e da nossa ideia fornecendo os equipamentos necessários para realização do mesmo. E o por último e não menos importante aos meus colegas de trabalho Pedro DI Martino, Eliane Das Graças, Paulo Crapina, Nayara Gomes e Maura Matos pela cooperação diária nas atividades no qual fizeram agregar bastante conhecimento, trocas de experiência, vivencia e inúmeros aprendizados para conclusão desse estudo.

# Resumo

O presente trabalho foi realizado com o intuito de solucionar problemas e desafios da gestão de estoque através do emprego de tags RFID para identificação de ativos, essa problemática leva a perdas significativas para empresas. Para isso foi realizada a pesquisa qualitativa, assim como uma revisão de literatura para identificar os trabalhos existentes e possíveis lacunas que os trabalhos correlatos deixaram em aberto. Dessa forma foi desenvolvido o sistema denominado Sistema Gestor de Estoque. Para seu desenvolvimento foram utilizadas as tecnologias Ruby on Rails para o backend, PostgreSQL para o banco de dados, Bootstrap para o frontend, Heroku como serviço de hospedagem em nuvem para a API, além do sistema operacional Android para poder usar o aplicativo desenvolvido nesse trabalho. Para verificar o funcionamento do sistema, a aplicação foi executada em um cenário real de um pequeno empreendimento do ramo da moda que atua com a venda direta de roupas para o público geral. Embora o sistema seja funcional e possa ser utilizado por qualquer organização, existem problemas relacionados aos altos custos de implementação da tecnologia RFID. Isso é algo a ser superado futuramente, assim como a aplicação desse sistema em escala industrial.

**Palavras-chave:** gestor, estoque, RFID, sistema web, APP, bluetooth, Android, API, Identificação por Rádio Frequência

# Abstract

The present work was carried out in order to solve problems and challenges of inventory management through the use of RFID tags for asset identification, this problem leads to significant losses for companies. For this, qualitative research was carried out, as well as a literature review to identify existing works and possible gaps that related works left open. Thus, the system called "Inventory Management System" was developed. For its development, technologies like Ruby on Rails were used for the backend, PostgreSQL for the database, Bootstrap for the frontend, Heroku as a cloud hosting service for the API, in addition to the Android operating system to be able to use the application developed in this job. To verify the functioning of the system, it was applied in a real scenario of a small enterprise in the fashion industry that works with the direct sale of clothes to the general public. Although the system is functional and can be used by any organization, there are problems related to the high costs of implementing RFID technology. This is something to be overcome in the future, as well as the application of this system on an industrial scale.

**Keywords:** inventory management, RFID, Android, Web App, bluetooth, API, Radio Frequency Identification

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Justificativa do Problema . . . . .	3
1.2	Objetivos . . . . .	4
1.2.1	Objetivo Geral . . . . .	4
1.2.2	Objetivos Específicos . . . . .	4
1.3	Resultados Esperados . . . . .	4
1.4	Metodologia . . . . .	4
1.5	Estrutura do Trabalho . . . . .	6
<b>2</b>	<b>Fundamentação Teórica</b>	<b>8</b>
2.1	Identificação por Radiofrequência (RFID) . . . . .	8
2.2	Partes que compõem o sistema RFID . . . . .	10
2.2.1	Tags . . . . .	10
2.2.2	Antena . . . . .	10
2.2.3	Leitor RFID . . . . .	11
2.2.4	Middleware . . . . .	11
2.2.5	Computador . . . . .	11
2.2.6	Faixas de frequência . . . . .	12
2.3	Padronização dos sistemas de RFID . . . . .	12
2.3.1	Regulamentação sobre o uso de faixas de frequência no Brasil . . .	12
2.3.2	Padronização de Protocolos de RFID . . . . .	13
2.4	Indústria 4.0 . . . . .	15
2.5	Internet das Coisas (IOT) . . . . .	17
2.6	Diferenças entre o sistema RFID e o sistema com Códigos de barras . . .	18
2.6.1	Comparação de etiquetas RFID com códigos de barras tradicionais	18
2.7	Vantagens e desvantagens da utilização da tecnologia RFID . . . . .	19
2.7.1	Dificuldades da aplicação da tecnologia RFID . . . . .	19
2.8	Frameworks Usados no projeto . . . . .	20
2.8.1	Ruby . . . . .	20

2.8.2	Rails . . . . .	21
2.8.3	Gems do Ruby on Rails . . . . .	23
2.8.4	PostgreSQL . . . . .	23
2.8.5	Bootstrap . . . . .	24
2.8.6	Heroku . . . . .	25
2.9	Leitor de tags RFID . . . . .	25
2.9.1	SDK do Leitor . . . . .	28
2.9.2	O Sistema Operacional Android . . . . .	28
2.9.3	Bluetooth Low Energy (BLE) . . . . .	29
2.10	Trabalhos Correlatos . . . . .	30
2.11	Síntese do Capítulo . . . . .	33
<b>3</b>	<b>Desenvolvimento do Sistema</b>	<b>34</b>
3.1	Arquitetura Proposta . . . . .	34
3.2	Sistema . . . . .	34
3.2.1	Preparação do Ambiente . . . . .	34
3.3	Desenvolvimento do Aplicativo Android . . . . .	35
3.4	Desenvolvimento da Aplicação Web . . . . .	36
3.5	Requisitos Funcionais . . . . .	38
3.6	Requisitos Não Funcionais . . . . .	39
3.7	Modelagem do Banco de Dados do Sistema Web . . . . .	40
3.8	Diagrama de Casos de Uso . . . . .	41
3.9	Modelagem do Banco de Dados da API . . . . .	42
3.9.1	API . . . . .	42
3.9.2	Fornecedores . . . . .	43
3.9.3	Clientes . . . . .	44
3.9.4	Produtos . . . . .	45
3.9.5	Categorias . . . . .	46
3.9.6	Compras . . . . .	47
3.9.7	Vendas . . . . .	48
3.9.8	Taxas . . . . .	49
3.9.9	Usuários . . . . .	50
3.9.10	Endereços . . . . .	51
3.10	Síntese do Capítulo . . . . .	52
<b>4</b>	<b>Apresentação do Sistema</b>	<b>53</b>
4.1	Periféricos do Sistema de Captura de Tags . . . . .	53
4.1.1	Leitor RFID Bluetooth . . . . .	53

4.1.2	Tags Passivas . . . . .	54
4.2	Telas do Aplicativo de Captura de Tags . . . . .	55
4.2.1	Conectando no Leitor Via Bluetooth . . . . .	55
4.2.2	Dispositivos Pareados . . . . .	56
4.2.3	Escaneando Tags RFID . . . . .	57
4.2.4	Exibindo o Código Eletrônico do Produto . . . . .	58
4.2.5	Enviando Dados para a API . . . . .	59
4.2.6	Exibindo Dados Salvos Diretamente na API . . . . .	59
4.3	Telas do Sistema Web . . . . .	60
4.3.1	Tela de login . . . . .	60
4.3.2	Tela de Cadastro de Usuário . . . . .	60
4.3.3	Dashboard . . . . .	61
4.3.4	Tela de Index Clientes . . . . .	61
4.3.5	Tela de Cadastro Cliente . . . . .	62
4.3.6	Tela de Editar Categoria . . . . .	63
4.3.7	Tela de Excluir Cliente . . . . .	63
4.3.8	Tela de Exibição de Produto . . . . .	64
4.3.9	Tela de Importação de Produtos . . . . .	64
4.3.10	Tela de busca RFID . . . . .	65
4.3.11	Tela de Verificação RFID . . . . .	65
4.4	Síntese do Capítulo . . . . .	66
<b>5</b>	<b>Resultados da Aplicação do Sistema</b>	<b>67</b>
5.1	Resultados . . . . .	67
5.2	Uso do Sistema em um Contexto Real . . . . .	68
5.3	Comparativo dos Resultados . . . . .	70
5.4	Limitações e Ameaças para Validar o Estudo . . . . .	70
5.5	Síntese do Capítulo . . . . .	72
<b>6</b>	<b>Conclusão</b>	<b>73</b>
6.1	Melhorias para trabalhos futuros . . . . .	73
	<b>Referências</b>	<b>75</b>
	<b>Apêndice</b>	<b>81</b>
<b>A</b>	<b>Códigos Fontes</b>	<b>82</b>

# Lista de Figuras

1.1	Fluxograma metodologia de pesquisa. Fonte: Própria autoria . . . . .	6
2.1	Componentes do sistema RFID. [1] . . . . .	12
2.2	Leitora RFID [1] . . . . .	25
2.3	Fonte dos autores [1] . . . . .	27
3.1	Módulos do sistema. Fonte: Própria dos autores . . . . .	38
3.2	Modelo de Entidade Relacional Sistema Web. Fonte: Própria dos autores .	40
3.3	Modelo de Entidade Relacional Sistema Web. Fonte: Própria dos autores .	41
3.4	Modelo de Entidade Relacional API. Fonte: Própria dos autores . . . . .	42
4.1	Leitor RFID Portátil. Fonte: Própria dos autores . . . . .	54
4.2	Tags de identificação RFID. Fonte: Própria dos autores . . . . .	54
4.3	Tela de buscar dispositivos. Fonte: Própria dos autores . . . . .	55
4.4	Tela de parear dispositivo APP. Fonte: Própria dos autores . . . . .	56
4.5	Tela de de escanear tags APP. Fonte: Própria dos autores . . . . .	57
4.6	Exibir tags no APP. Fonte: Própria dos autores . . . . .	58
4.7	Enviando para API. Fonte: Própria dos autores . . . . .	59
4.8	Tela de tags salvas na API. Fonte: Própria dos autores . . . . .	59
4.9	Tela de login do sistema. Fonte: Própria dos autores . . . . .	60
4.10	Tela de login do sistema. Fonte: Própria dos autores . . . . .	60
4.11	Tela de Dashboard do sistema. Fonte: Própria dos autores . . . . .	61
4.12	Tela index clientes. Fonte: Própria dos autores . . . . .	62
4.13	Tela de criar cliente. Fonte: Própria dos autores . . . . .	62
4.14	Tela de editar categoria. Fonte: Própria dos autores . . . . .	63
4.15	Tela de excluir cliente. Fonte: Própria dos autores . . . . .	63
4.16	Tela que exhibe os produtos. Fonte: Própria dos autores . . . . .	64
4.17	Importação de produtos através de um arquivo CSV. Fonte: Própria dos autores . . . . .	64
4.18	Tela de busca na API. Fonte: Própria dos autores . . . . .	65



4.19	Tela que exibe se a tag já existe no sistema. Fonte: Própria dos autores . .	66
5.1	Camisetas do estoque. Fonte: Própria dos autores . . . . .	68
5.2	Tag RFID Camisetas. Fonte: Própria dos autores . . . . .	68
5.3	Tags Cadastradas. Fonte: Própria dos autores . . . . .	69
5.4	Camiseta Rosa Cadastrada. Fonte: Própria dos autores . . . . .	69

# Lista de Tabelas

2.1	Faixas de frequência de funcionamento.[2] . . . . .	12
2.2	Classes de funcionalidades. [3] . . . . .	15
2.3	Leds indicadores da leitora . . . . .	26
2.4	Parte lateral da leitora . . . . .	27
2.5	Trabalhos correlatos versus nossa proposta. Fonte: Própria dos autores . .	32
3.1	Tabela Bluetoothsearch. Fonte: Própria dos autores . . . . .	42
3.2	Tabela Providers. Fonte: Própria dos autores . . . . .	43
3.3	Tabela Clients. Fonte: Própria dos autores . . . . .	44
3.4	Tabela Products. Fonte: Própria dos autores . . . . .	45
3.5	Tabela Categories. Fonte: Própria dos autores . . . . .	46
3.6	Tabela Buys. Fonte: Própria dos autores . . . . .	47
3.7	Tabela Sells. Fonte: Própria dos autores . . . . .	48
3.8	Tabela Taxes. Fonte: Própria dos autores . . . . .	49
3.9	Tabela Users. Fonte: Própria dos autores . . . . .	50
3.10	Tabela Addresses. Fonte: Própria dos autores . . . . .	51

# Lista de Abreviaturas e Siglas

**ACID** Atomicity, Consistency, Isolation, Durability.

**ANATEL** Agencia Nacional de Telecomunicações.

**ANVISA** Agência Nacional de Vigilância Sanitária.

**API** Application Programming Interface.

**APK** Android Application Pack.

**ARO** Army Research Office.

**BLE** Bluetooth Low Energy.

**CRM** Customer Relationship Management.

**CRUD** Create, Read, Update, Delete.

**CSS** Cascading Style Sheets.

**DARPA** Defense Advanced Research Projects Agency.

**ECMA** European Computer Manufacturers Association.

**ECR** Efficient Consumer Response.

**EEPROM** Erasable Programmable Read Only Memory.

**EPC** Electronic Product Code.

**ERP** Enterprise Resource Planning.

**FIFO** First In First Out.

**GIAI** Global Individual Asset Identifier.

**GLN** Global Location Number.

**GRAI** Global Returnable Asset Identifier.

**GSIN** Global Shipment Identification Number.

**GTIN** Global Trade Item Number.

**HF** High Frequency.

**HTML** Hyper Text Markup Language.

**HTTP** Hypertext Transfer Protocol.

**IDE** Integrated Development Environment.

**IEC** International Electrotechnical Commission.

**IOT** Internet of Things.

**ISO** International Organization for Standardization.

**JSON** JavaScript Object Notation.

**LF** Low Frequency.

**MAC** Media Access Control.

**MIT** Massachusetts Institute of Technology.

**MVC** Model View Controller.

**NSF** National Science Foundation.

**ORMs** Object Relational Mapper.

**PAAS** Platform As A Service.

**PIB** Produto Interno Bruto.

**RADAR** Radio Detection and Ranging.

**RAM** Random Access Memory.

**REST** Representational State Transfer.

**RFID** Radio Frequency Identification.

**SASS** Syntactically Awesome Style Sheets.

**SDK** Software Development Kit.

**SGBD** Data Base Management System.

**SQL** Structured Query Language.

**TIC** Tecnologia da Informação e Comunicação.

**TMS** Transportation Management System.

**UHF** Ultra High Frequency.

**UI/UX** User Interface/User Experience.

**UIT** União Internacional de Telecomunicações.

**UUID** Universally Unique Identifier.

**Wi-Fi** Wireless Fidelity.

**WMS** Warehouse Management System.

**XML** eXtensible Markup Language.

# Capítulo 1

## Introdução

Com o avanço das tecnologias e em diversas áreas da computação, a indústria 4.0 veio com o objetivo de interligar as pessoas com os sistemas e produtos oferecidos, visando trazer ganhos em eficiência enquanto se reduz custos e consumo de energia. Consequentemente, os impactos ambientais também são reduzidos à medida que os processos se tornam mais inteligentes e integrados. Esses processos passam a contar com características de adaptabilidade e maior eficiência na utilização de recursos. Em 2021 o setor industrial foi responsável por alavancar 22,2% do Produto Interno Bruto (PIB) do Brasil [4].

A integração é uma das principais tecnologias na indústria 4.0. Dessa forma, é possível interligar diversos sistemas de computação, assim como aplicações de software, para que essas atuem de forma coordenada. Além da integração temos também o emprego de novos conceitos, como por exemplo *Internet of Things (IOT)*, que consiste em uma rede de dispositivos e máquinas que conectam itens utilizados diariamente à rede mundial de computadores, permitindo assim a troca de informações e dados para a automatização de certos processos [5].

Dessa forma, é possível haver trocas de informações entre sistemas, fator esse que permite a indústria ter uma visão mais abrangente sobre o negócio. O uso da tecnologia permite otimizar e transformar os processos de produção, além de trazer ganhos em eficiência [6]. Dentre essas tecnologias, decidimos trabalhar com o Radio Frequency Identification (RFID) sistema de identificação por radiofrequência, cujo o seu uso na área da logística proporciona uma maior autonomia aos processos de negócios e um maior controle da cadeia de suprimentos [7].

O sistema de RFID é composto por tags que possuem chips aliados a antenas que recebem um sinal em uma faixa de radiofrequência, permitindo a leitura e a escrita de informações. A outra parte do mecanismo é o leitor que tem o papel de ler o que está armazenado no chip. As tags possibilitam a automatização, que consequentemente otimizam a rastreabilidade de produtos, a administração de processos e o gerenciamento de

estoques[2].

As aplicações baseadas em RFID tem por objetivo solucionar problemas relacionados à identificação, automação e auxiliar na tomada de decisão, através do compartilhamento das informações dentro da cadeia de vendas. Além disso, permite a melhora do processo de execução dos negócios [8].

A utilização da tecnologia de RFID em uma cadeia de suprimentos pode gerar benefícios e vantagens na administração e logística. Apesar de seu alto investimento inicial, o retorno vem com a redução no custo de mão de obra ao longo da cadeia [9]. Este, por sua vez, tem potencial para proporcionar um conjunto significativo de benefícios, com maior precisão quanto aos níveis de estoques e maior transparência quanto à demanda em tempo real na cadeia de suprimentos. No final, esses fatores podem contribuir para o aumento da eficiência nas operações de negócios, bem como melhorias na confiabilidade de entrega, tempo de resposta e disponibilidade de produtos .[10].

Para que seja adotada uma nova tecnologia, diversos fatores necessitam ser levados em conta, como por exemplo a infraestrutura, aspectos legais, perfil do mercado local, entre outros. Cláudio Czapski, em uma entrevista para a ComputerWorld em 2009 [11], mencionou a falta de escalabilidade, que não combina com os altos volumes necessários para o RFID. O custo de 20 centavos de real, na época, não era adequado para produtos de baixo custo e também o perfil de cópia do Brasil, que prefere aguardar os resultados em iniciativas estrangeiras para começar os investimentos.

Na década passada não haviam dados concretos que permitissem verificar como empresas estavam utilizando e quais resultados estavam sendo obtidos com a implementação do RFID, pois a tecnologia ainda estava em sua fase inicial de implementação na rede de logística. Porém, atualmente há diversas companhias que obtiveram sucesso e que estão presentes no cotidiano Brasileiro, tais como a Unilever, Boticário e Centauro [12].

Dehoratius and Raman [13] examinaram aproximadamente 3700.000 registros de inventário de 37 lojas, com receita anual de aproximadamente U\$D 10 bi e descobriram que aproximadamente 65% das informações sobre esses registros estavam imprecisas. Essas imprecisões chegam a resultar em prejuízos que reduzem as margens de lucro em mais de 10%. A discrepância entre os registros de inventário e produtos que realmente estão disponíveis para serem comprados pelos consumidores, também chamado de *Inventory Inaccuracy*, é um problema recorrente e inevitável em muitas indústrias [13].

A adoção de tecnologia RFID pode reduzir imprecisões de estoque na cadeia de abastecimento e várias cadeias de varejo exigiram que seus fornecedores adotassem essa tecnologia. O WalMart, por exemplo, implementou um sistema de rastreamento em nível de paletes e caixas baseado em RFID no início de 2005 e exigiu que seus principais fornecedores fornecessem tags de radio frequência em caixas e paletes [13].

O objetivo deste trabalho é desenvolver um sistema web, para auxiliar, controlar e otimizar a gestão de ativos em uma organização. A plataforma fará uso de etiquetas para identificação de produtos com o objetivo de automatizar processos que antes eram feitos de forma manual, facilitando, dessa forma, a identificação e rastreabilidade dos ativos, possibilitando assim um melhor controle da cadeia de vendas de forma automatizada e disponível em diversas plataformas.

## 1.1 Justificativa do Problema

Muitas empresas não obtêm sucesso na implantação do RFID, devido a falhas em suas tecnologias, conhecimento de Tecnologia da Informação e Comunicação (TIC) insuficiente, limites no orçamento, parceiros de negócios sem qualificação, falta de conhecimento da eficiência da tecnologia e como se relaciona isso ao negócio. Portanto, para entender a tecnologia RFID, as organizações devem compreender os conceitos de operação e os componentes do sistema [14].

O fator humano é um ponto importante a ser observado. A resistência dos colaboradores diante da implantação de novas tecnologias é algo bastante significativo, pois senão houver aceitação por parte dos colaboradores que estão atuando diretamente com os sistemas, há uma grande chance de o investimento não apresentar retorno ou até mesmo ser desperdiçado [15]. Além disso, existe uma dificuldade em encontrar sistemas RFID *userfriendly* para todos os usuários diretos e indiretos. Há também diversos problemas relacionados à integração da tecnologia RFID com os sistemas de gestão já atuantes nas empresas como por exemplo *Enterprise Resource Planning (ERP)*, *Warehouse Management System (WMS)*, *Transportation Management System (TMS)* e *Customer Relationship Management (CRM)*. Além disso, vale ressaltar a dificuldade em oferecer treinamento para capacitar os possíveis novos usuários do sistema[15].

Um recente projeto denominado *Efficient Consumer Response (ECR)* analisou posteriormente as causas da perda de estoque. Os resultados mostraram que a retração no setor de bens de consumo de movimento rápido era estimada em 24 bilhões de euros em 2003. Os erros de processo apresentaram 27% de todo o valor de retração, 7% enganos, 28% furtos internos e 38% furtos externos [16]. Assim, é importante que a indústria utilize dos benefícios da tecnologia RFID no gerenciamento e controle de estoques.



## **1.2 Objetivos**

### **1.2.1 Objetivo Geral**

O objetivo deste trabalho é desenvolver um WMS, a fim de investigar qual a eficiência da utilização da tecnologia RFID neste contexto, assim como os ganhos de produtividade obtidos ao implementá-la na área de logística de cadeia de suprimentos.

### **1.2.2 Objetivos Específicos**

Para atingir o objetivo geral deste trabalho, os seguintes objetivos específicos foram definidos:

1. Realizar uma revisão de literatura para identificar as vantagens da utilização da tecnologia RFID;
2. Investigar os efeitos e impactos da implementação da tecnologia RFID em uma organização;
3. Identificar as dificuldades e peculiaridades encontradas na implantação do sistema;
4. Desenvolver um software de gerenciamento de estoque;
5. Realizar testes práticos com o sistema desenvolvido para verificar a viabilidade da implantação; e
6. Realizar ajustes nas funcionalidades do software, caso necessário.

## **1.3 Resultados Esperados**

O resultado final deste trabalho será a entrega de um sistema web de gerenciamento e controle de estoque, o qual faz uso da radiofrequência para identificar os produtos. Esse sistema tem por objetivo ser de fácil manuseio, contar com interface amigável, além de fazer uso de um equipamento com o custo mais acessível, permitindo assim que o uso dessa tecnologia possa ser aplicado e escalado, inclusive, em pequenos negócios.

## **1.4 Metodologia**

A metodologia utilizada neste trabalho foi a pesquisa qualitativa e o método usado foi uma revisão sistemática da literatura. Tendo em vista que essa metodologia se baseia na observação cuidadosa dos ambientes onde o sistema está em uso ou será utilizado, do

entendimento das várias perspectivas dos potenciais usuários do sistema [17]. Para tal fim, foram adotadas as seguintes etapas conforme apresentado na (Figura 1.1):

1. Estudo de conceitos relacionados à RFID;
2. Gestão logística e Indústria 4.0;
3. Levantamento e revisão de literatura de trabalhos similares e relacionados;
4. Análise e sintetização dos conjuntos de trabalhos identificados como relevantes;
5. Estudo do hardware de uma antena, modulo e leitora para a aplicação do sistema RFID proposto no trabalho;
6. Estudo de qual linguagem de programação é mais adequada para o desenvolvimento do sistema proposto;
7. Desenvolvimento do sistema com base na pesquisa de levantamento de dados, linguagem de programação e hardware para suprir a carência nos pontos relatados;
8. Análise de resultados do uso do software desenvolvido para identificar se o software cumpriu com o objetivo proposto; e
9. Recolher feedback e sugestões de melhorias apontadas pelos possíveis usuários do sistema e realizar melhorias na solução, caso necessário.

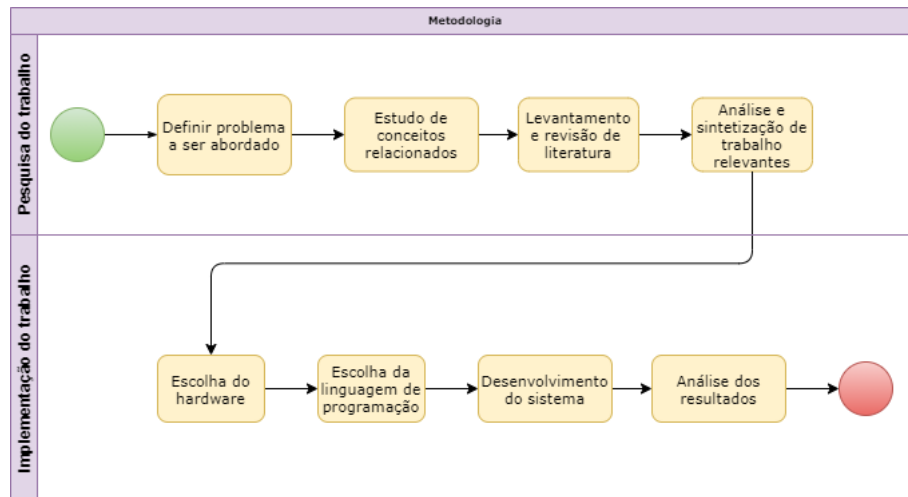


Figura 1.1: Fluxograma metodologia de pesquisa. Fonte: Própria autoria

## 1.5 Estrutura do Trabalho

Este trabalho está organizado da seguinte maneira:

- No **Capítulo 2** é apresentado o referencial teórico necessário para entender a contextualização deste trabalho, sendo este constituído por conceitos do Histórico do RFID, Indústria 4.0, Conceito de IOT, Diferenças entre o sistema RFID e o sistema com Códigos de barras, Aplicação da tecnologia de radiofrequência na indústria, Dificuldades da aplicação da tecnologia RFID, Exemplos de empresas que adotaram a tecnologia RFID, Vantagens e desvantagens de sua utilização.
- No **Capítulo 3** apresentamos a arquitetura proposta para desenvolvimento do sistema. Para que seja possível replicar corretamente os passos que realizamos, definimos a preparação do ambiente, o desenvolvimento do aplicativo Android e Web, além da criação e modelagem dos bancos de dados web e da Application Programming Interface (API).
- No **Capítulo 4** fazemos a apresentação do sistema. Por possuir uma parte que é executada em um dispositivo móvel com o sistema operacional Android e outra parte que roda na web, disponibilizamos screenshots das telas de ambas as plataformas, além de uma breve descrição daquilo que deve ser visualizado.
- No **Capítulo 5** trazemos o caso de uso do sistema, sua aplicação em um empreendimento de pequeno porte, o qual atua no ramo da moda. Assim como os resultados que foram obtidos ao implementar o sistema em um cenário real.

- No **Capítulo 6** apresentamos as conclusões que foram obtidas ao realizar esse estudo.

# Capítulo 2

## Fundamentação Teórica

### 2.1 Identificação por Radiofrequência (RFID)

Na década de 30 o físico Robert Alexander Watson-Watt realizou a descoberta do sistema de Radio Detection and Ranging (RADAR). Essa tecnologia que permitiu realizar a identificação de uma aeronave a quilômetros de distância. Esse sistema foi amplamente utilizado durante a Segunda Guerra Mundial pelos países membros do Eixo, como pelos Aliados. Porém, em seu período inicial de uso não era possível realizar a distinção entre aeronaves inimigas ou aliadas que retornavam de suas missões. Os alemães criaram um padrão de manobras áreas utilizadas pelas aeronaves aliadas. Essa manobra causava mudança no sinal de rádio que permitia diferenciar aliados de inimigos [18].

Um sistema RFID faz uso de ondas eletromagnéticas em uma determinada frequência e com isso, é possível ler e gravar dados em uma etiqueta conhecida também como tag, dados como número de série, data de fabricação, prazo de validade, informações de preço, número do lote, rastreio em tempo real e controle de estoques de mercadorias. Suas aplicações em nossa sociedade possibilitam a troca de informações com diversas tecnologias de informação, tais como sistemas de gerenciamento de banco de dados, rede de computadores, firewalls, computação em nuvem, Internet of Things (IOT) e sistemas de controle em tempo real [9]. Hoje em dia o emprego dessa tecnologia é difundida em diversas áreas e com várias finalidades de uso onde foram empregadas. Fazemos uso dessas tecnologias em:

- **Pedágios:** O uso do RFID em pedágio é feito através de tags com chips que são coladas no para-brisa dos carros, que ao passarem pela cancela onde existe é emitido um sinal, a cobrança é efetuada no sistema e a passagem é liberada [19];
- **Aplicações médicas:** Na medicina o seu uso é feito através de implantes subcutâneos com a finalidade de identificar os pacientes e seus dados pessoais como,

nome, identidade e tipo sanguíneo, alergias a medicamentos e inclusive se o mesmo é portador de uma doença crônica de uma maneira mais ágil [19];

- **Controle de acesso:** No campo de controle de acesso essa tecnologia vem com o desafio de promover a segurança e evitar fraudes, como em grandes eventos, em residências, acesso a cofres e até mesmo em contas bancárias [19];
- **Área Logística:** Na logística, sua aplicação tem sido amplamente difundida, temos inúmeros casos de sucesso, sua utilização está auxiliando no controle de qualidade, controle de estoque de produtos em tempo real, em operações de recebimento, separação, transporte, armazenamento e na expedição de materiais em armazéns e depósitos [19];
- **Linhas de montagens industriais:** Em linhas de montagens industriais a ajuda dessa tecnologia é muito importante, pois assegura o rastreio em sua linha de produção, desde a sua entrada na fábrica até sua saída, evitando extravios [19];
- **Sistemas de proteção antifurtos veicular:** Nos sistemas de proteção antifurto, o sistema RFID é responsável por travar o sistema de ignição, impedindo o funcionamento do carro [19];
- **Uso na pecuária:** Na pecuária, ajuda a prevenir roubos e furtos de animais, principalmente nas regiões de fronteiras, além de identificar a origem da carne e manter o controle de vacinas atualizado [19];
- **Uso em modalidades esportivas:** Esportes que precisam cronometrar o tempo com precisão fazem o uso dessa tecnologia, tais como Formula 1, Ciclismo e Atletismo [19];
- **Uso em produtos farmacêuticos:** A Lei nº 11.903/2009, prevê a rastreabilidade de medicamentos por meio de tecnologia de captura, armazenamento e transmissão eletrônica de dados. A Agência Nacional de Vigilância Sanitária (ANVISA) é responsável por fiscalizar a produção e consumo desses medicamentos [19];
- **Uso em estoques:** Permite um controle de forma mais otimizada de um estoque, assim sabendo quais produtos precisam ser repostos, quais estão tendo mais demanda, protegendo contra furtos e evitando estoque de produtos parados que não estão tendo muitas saídas ultimamente [19];
- **Uso em bibliotecas:** Seu uso em bibliotecas garante com exatidão onde o livro está guardado, em qual seção de prateleira onde se encontra [19].

Esses pontos levantados conseguem nos mostrar, com bastante clareza, que essa tecnologia veio para agregar ainda mais e simplificar a maneira pela qual resolvíamos problemas,

de forma manualmente na organização e controle, coisas simples em nosso dia a dia como uma compra em um supermercado com um cartão, sem necessidade de utilizar senha até em coisas mais complexas, por exemplo a gestão de um controle de estoque com dezenas de milhares de ativos. E isto está cada vez mais presente em nosso cotidiano.

## 2.2 Partes que compõem o sistema RFID

### 2.2.1 Tags

Conforme apresentado na Figura 2.1 temos a composição de todas as peças que fazem parte do sistema de RFID, para começar temos a etiqueta ou tag de radiofrequência que é o dispositivo encarregado de receber e emitir ondas de rádio em uma determinada frequência. Ela possui uma Erasable Programmable Read Only Memory (EEPROM). Cada tag possui um identificador único chamado de Electronic Product Code (EPC), responsável por identificar de forma exclusiva uma tag [20]. Essas tags respondem aos sinais de radiofrequência de um leitor, enviando de volta informações quanto a sua localização e identificação, através de um chip, um circuito eletrônico e uma antena interna. Podemos dividir essas tags em três categorias: **tags ativas**, **tags passivas**, **tags semi passivas**. Essas tags possuem tamanhos e formatos variados, podendo também serem fabricados com materiais como PVC, PET, papel e outros, dependendo de sua aplicação [3].

- As **tags ativas** possuem bateria interna, elevando seu custo. Essa é ativada quando recebe um sinal de radiofrequência do leitor RFID. O seu sinal é emitido de forma contínua e pode ser captado em distâncias superiores a 100 metros[21].
- As **tags passivas**, ao contrário das ativas, não possuem uma fonte de energia própria. Com isso seu custo tende a ser menor. Porém, essas só são ativadas ao receber um sinal de radiofrequência, entrando em um campo eletromagnético emitido pela antena. Parte desse sinal, ao entrar em contato com o chip, é transformado em corrente elétrica para ser transformado em um sinal de resposta, sendo possível a transmissão ou gravação das informações[21][22].
- E por fim as, **tags semi passivas**, possuem fonte de energia própria. Entretanto, essa fonte de energia só é ativada ao receber um sinal de onda eletromagnética da antena, permitindo assim a troca de informações entre o sistema[21].

### 2.2.2 Antena

As antenas são as responsáveis pela comunicação entre a etiqueta e o leitor. O formato da antena e o local a ser colocada são fatores fundamentais para o alcance e um bom

funcionamento. O sistema RFID atua de maneira analógica. Seu sinal é mais passível a interferências e ruídos. Os principais causadores dessas interferências são a proximidade com líquidos, objetos metálicos, umidade elevada no ambiente, temperaturas extremamente quentes ou frias e dispositivos sem fio[23].

### **2.2.3 Leitor RFID**

O leitor é o dispositivo responsável pela captura e processamento dos dados armazenados em uma tag. Alguns modelos de leitores combinado com etiquetas específicas permitem a gravação de dados. Quanto maior a potência do leitor maior é a intensidade do campo eletromagnético emitido por ele. Devido o leitor emitir um campo eletromagnético ele, opera em faixas de frequência que pode variar entre Low Frequency (LF), High Frequency (HF) e Ultra High Frequency (UHF), essas faixas de frequência são definidas pelos regulamentos e normas de cada país. O leitor também é responsável por gerenciar o fluxo de dados entre as tags e o computador. O leitor pode se comunicar com o computador através de uma porta serial, uma conexão Ethernet, via USB, Bluetooth ou até mesmo através de um rede sem fio.[23].

### **2.2.4 Middleware**

O middleware é o componente encarregado de processar as informações obtidas pelo leitor. Este componente é o software que permite o registro da comunicação entre tag e leitor[23].

### **2.2.5 Computador**

Responsável pela comunicação de dados entre o leitor e as tags RFID. O computador pode estar conectado em rede e transferir essas informações para um servidor central onde são coletadas informações de diversos leitores RFID para serem tratadas e servirem a uma finalidade.



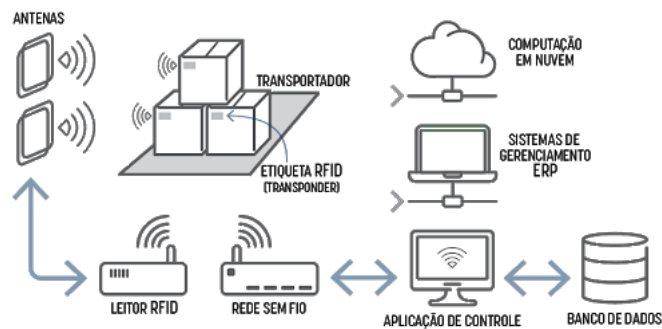


Figura 2.1: Componentes do sistema RFID. [1]

## 2.2.6 Faixas de frequência

A tecnologia RFID possui 3 faixas de frequência para sua operação, sendo elas baixa (*Low Frequency*), intermediária (*High Frequency*) e alta (*Ultra High Frequency*). Os sistemas de RFID são distinguidos por 3 faixas: baixa, intermediária e alta. Cada uma delas com sua especificidade em suas aplicações. Conforme a Tabela 2.1 podemos mostrar com mais objetividade as faixas de frequência, características, alcance e para onde e suas aplicações.

Tabela 2.1: Faixas de frequência de funcionamento.[2]

Faixa de frequência	Características	Alcance	Algumas aplicações
<i>Low Frequency</i> de 100 a 500KHz	Baixo raio de alcance de leitura, Baixo custo	Até 0.5 metro	Identificação de Animais, Depósitos de Almoxxarifados, Garagens de Veículos
<i>High Frequency</i> de 10 a 15 MHz	Médio raio de alcance de leitura, Custo razoável	Até 1 metro	Smartcards
<i>Ultra High Frequency</i> de 850 a 950MHz 2,4 a 5,8GHz	Grande raio alcance de leitura, Alto custo	Dezenas de metros	Monitoramento de veículo em trânsito

## 2.3 Padronização dos sistemas de RFID

### 2.3.1 Regulamentação sobre o uso de faixas de frequência no Brasil

No Brasil a autarquia pública responsável pela regulamentação e autorização das telecomunicações em todo território é a Agencia Nacional de Telecomunicações (ANATEL), criada em 1997 pela Lei Geral de Telecomunicações, Lei nº 9.472/1997 com o poder de

incentivar o desenvolvimento das telecomunicações do país. Entre algumas atribuições da ANATEL podemos citar:

- gerenciar o espectro e o uso de radiofrequências, conforme os requisitos.
- assegurar que as normas e diretrizes sejam seguidas pelos provedores de serviços de telecomunicações em relação aos equipamentos que utilizam.
- agilizar ou confirmar a certificação de produtos ao mesmo tempo em que segue as normas e diretrizes estabelecidas pela organização.

O espectro eletromagnético possível para comunicação varia de 8,3 kHz a 3000 GHz. A utilização dessas bandas é regida por tratados e acordos internacionais que foram aprovados pela União Internacional de Telecomunicações (UIT). No Brasil, a Agência Nacional de Telecomunicações (ANATEL) publica anualmente um Plano de Atribuição, Destino e Distribuição de Faixas de Frequências de Encargo. Este documento deve detalhar o uso das faixas de radiofrequência associadas aos diversos serviços e atividades de telecomunicações[24].

### 2.3.2 Padronização de Protocolos de RFID

Com a popularização do uso do RFID, surgiu a necessidade de padronizar os protocolos, pois cada fabricante fazia o uso da tecnologia da forma que achasse conveniente, abrindo margem para diversos tipos de erros de incompatibilidade, pois operavam de forma que chegavam a interferir em outras faixas de frequências, dificultando a instalação e o bom funcionamento do sistema como um todo. A partir desse momento órgãos fiscalizadores enxergaram que tinham a necessidade de criar um modelo de padronização internacional. Essa normalização da regra veio com intuito de garantir que equipamentos e sistemas operassem de forma integrada, assegurando que usuários e fabricantes falassem a mesma língua. Organizações como a International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), GS1, European Computer Manufacturers Association (ECMA) foram as grandes responsáveis por trás desse feito. Segue a lista das normas ISO sobre o uso do RFID [25],[26].

- **ISO 11784** RFID para animais – estrutura de código[27];
- **ISO 11785** RFID para animais – concepção técnica[28];
- **ISO/IEC 14443** Identificação de cartões – cartões com circuitos integrados sem contato – cartões de proximidade[29];

- **ISO/IEC 15961** Gerenciamento de Itens de RFID – Protocolo de Dados:Interface de Aplicação[30];
- **ISO/IEC 15962** Gerenciamento de Itens de RFID – Protocolo: Regras de Codificação de Dados e Funções de Memória Lógica[31];
- **ISO/IEC 15963** Gerenciamento de Itens de RFID – Identificação única do RF Tag[32];
- **ISO/IEC 15693** Identificação de cartões – cartões com circuitos integrados sem contato – cartões de vizinhança[33];
- **ISO/IEC 18000-1** Define uma arquitetura de referência e os parâmetros a serem normalizado[34];
- **ISO/IEC 18000-2:** Estabelece os parâmetros para a comunicação sem fio pelo ar na faixa de frequências abaixo de 135 kHz[35];
- **ISO/IEC 18000-3:** Estabelece os parâmetros para a comunicação sem fio pelo ar na frequência de 13,56 MHz[36];
- **ISO/IEC 18000-4:** Estabelece os parâmetros para a comunicação sem fio pelo ar na frequência de 2,45 GHz[37];
- **ISO/IEC 18000-6:** Estabelece os parâmetros para a comunicação sem fio pelo ar na faixa de frequências entre 860 MHz e 960 MHz[38];
  - **ISO/IEC 18000-61:** Comunicação tipo A
  - **ISO/IEC 18000-62:** Comunicação tipo B
  - **ISO/IEC 18000-63:** Comunicação tipo C
  - **ISO/IEC 18000-64:** Comunicação tipo D
- **ISO/IEC 18000-7:** Estabelece os parâmetros para a comunicação ativa sem fio pelo ar na frequência de 433 MHz [39].

No Brasil temos um associação chamada GS1 que se trata de uma organização sem fins lucrativos, com atividades de atuação no mundo inteiro. Ela desenvolve padrões globais de identificação, captura e rastreabilidade com sua atuação e emprego em diversos setores que vão desde o escopo de bens de consumo à área governamental, com ferramentas que ajudam na automação de processos da administração pública. Ela foi uma das organizações responsáveis que ajudou a padronizar e normalizar regras para o uso do RFID em todo mundo. Além do EPC usado em aplicações RFID, existem outros padrões de identificação únicos usados também a nível internacional como o Global Trade Item Number

(GTIN), Global Individual Asset Identifier (GIAI), Global Location Number (GLN), Global Shipment Identification Number (GSIN), Global Returnable Asset Identifier (GRAI) entre outros padrões reconhecidos e aceitos internacionalmente.[40][3] Ela também propõe níveis de classes para tags RFID, sendo que, cada classe possui funcionalidades específicas conforme é apresentado na Tabela 2.2:

Tabela 2.2: Classes de funcionalidades. [3]

Classe	Funcionalidade
Classe 0	Tags UHF somente leitura, passivas, pré-programadas
Classe 1	Tags UHF ou HF, de escrita única, de leituras múltiplas
Classe 3	Leitura e escrita com sensores integrados capazes de gravar parâmetros como temperatura, pressão e movimento; pode ser semipassiva ou ativa.
Classe 4	Tags ativas de leitura e escrita com transmissores integrados; pode se comunicar com outras tags e leitores.
Classe 5	Semelhante às tags da Classe 4, mas com funcionalidade adicional; pode fornecer energia para outras tags e se comunicar com outros dispositivos que não leitores.

## 2.4 Indústria 4.0

Indústria 4.0 é o termo usado para promover a ideia de que estamos iniciando uma nova revolução industrial, na qual o seu principal avanço é o aperfeiçoamento de uma série de tecnologias que permitem uma maior conexão entre os meios físico e digital, feita praticamente em tempo real[41]. Para chegarmos até a revolução da indústria 4.0 que encontramos atualmente tivemos que passar por outras revoluções.

A primeira revolução industrial ocorreu durante os séculos XVIII e XIX. O seu advento foi o uso da queima do carvão para ser utilizada como a principal força motriz, mecanizando processos produtivos que anteriormente eram feitos de forma artesanal. Esse período foi marcado pelas construções de ferrovias juntamente com invenção da máquina a vapor[42].

A segunda revolução industrial surgiu entre o final do século XIX e início do século XX. Ficou marcado pelo uso da eletricidade e da linha de montagem, modelos conhecidos como toyotismo e fordismo surgem nessa época como forma de otimização nas linhas de produção, permitindo uma alta produtividade[42].

A terceira revolução conhecida também como Revolução digital teve seu surgimento na década de 60. A grande evolução nessa revolução foi o desenvolvimento dos semicondutores, da computação em mainframe, da computação pessoal assim como o nascimento

e popularização da internet. A terceira revolução industrial foi tão grandiosa que chegou a quebrar paradigmas. Essa evolução veio por meio da facilidade de conexão com a internet em diversos aparelhos, conhecido como IOT como iremos ver futuramente, o aumento na capacidade de processamento dos computadores, a computação em nuvem, o surgimento e o aprimoramento da inteligência artificial e aprendizagem de máquinas [42].

A quarta revolução industrial é composta por três componentes: físico, digital e biológico, todos com impacto direto na economia e na sociedade. Em comparação com as revoluções anteriores que ocorreram ao longo da história e foram identificadas anos após sua ocorrência, a quarta revolução industrial é a primeira a ter seus resultados medidos em tempo real[43].

O sistema de Identificação por Rádio Frequência vem se destacando na indústria 4.0 pela sua praticidade de controle e identificação em larga escala. Seu uso permite que diversos objetos sejam identificados de forma única e conectados entre si para formar o que já é conhecido como Internet das Coisas [44]. O uso desses sistemas traz ganhos significativos para toda a cadeia de suprimentos, pois permite uma melhor gerencia de ativos, aumentando assim o acesso e a qualidade dos dados, que por fim tornam as operações mais transparentes e fluídas[45].

Com a indústria 4.0 cada vez ganhando seu espaço e conquistando cada vez mais adeptos, tecnologias como o RFID vem sendo cada vez mais usadas em diversas aplicações nos mais diversos setores. O RFID teve sua evolução e demorou um tempo até ser aprimorado e ser usado em grande escala e em soluções inteligentes como usamos hoje. Podemos dividir essa evolução em cinco eras[3]:

1. **Proprietária** (primeira era): começou com a redução do tamanho e custo de produção dos transistores, permitindo o desenvolvimento de tags RFID por empresas privadas para uso próprio no controle de seus produtos armazenados. Não havia condições para integrar as soluções na ausência de um padrão[3];
2. **Regulamentação** (segunda era): começou quando as empresas precisavam controlar seus produtos fora de suas próprias instalações. Contêineres eram identificados graças as tags de rádio frequência. Este período foi o pontapé para o desenvolvimento dos padrões para a aplicação universal da tecnologia. A instalação ainda era vista como um custo adicional ao processo pelas empresas[3];
3. **RFID nas Empresas** (terceira era): O diferencial foi a incorporação do RFID nos processos de negócios, com identificação efetuada item a item, envolvendo o controle do processo. A produção em larga escala das tags reduziu os custos finais de produção.[3];

4. **RFID nas Indústrias** (quarta era): a consolidação dos requisitos de segurança e privacidade define esta era. Os produtos são identificados e rastreados em todos os níveis do processo de fabricação. As informações dos produtos são utilizadas estrategicamente por diversas áreas de negócios. Além disso, sensores utilizando RFID e outras tecnologias começaram a coletar dados ambientais para incorporação e melhoria dos processos de negócios[3];
5. **Internet das Coisas** (quinta era): começa com a adoção generalizada do RFID e de outras tecnologias, quando empresas privadas podem fornecer serviços a seus clientes de maneira oportuna e inteligente. O setor público poderá prestar serviços à população de forma eficiente, com base em dados coletados automaticamente pelas cidades. O RFID e outras tecnologias serão descontinuadas, assim como a eletricidade será eliminada[3].

## 2.5 Internet das Coisas (IOT)

A Internet das Coisas envolve uma infraestrutura de rede que usa recursos de comunicação e captura de dados para vincular objetos físicos e virtuais e ela teve o seu surgimento com os sistemas embarcados, com a microeletrônica, comunicação e sensoriamento. O conceito de IOT foi introduzido pelo *Massachusetts Institute of Technology (MIT)* em 1999[46]. Esta foi definida como uma rede de todos os objetos que se comunicam entre si e interagem de forma autônoma via internet, permitindo o monitoramento desses dispositivos via software para aumentar a eficiência de sistemas e processos, viabilizar novos serviços e melhorar a qualidade de vida das pessoas[3],[47].

O RFID é uma tecnologia cujo o desenvolvimento mudará profunda e amplamente a vida das pessoas. Os leitores de RFID operam de acordo com os protocolos de comunicação estabelecidos e, por estarem ligados ao terminal de Internet, podem automaticamente e em tempo real identificar, rastrear e monitorar os objetos com etiquetas fixadas em todo o mundo. Isso é conhecido como a Internet das Coisas.

## 2.6 Diferenças entre o sistema RFID e o sistema com Códigos de barras

### 2.6.1 Comparação de etiquetas RFID com códigos de barras tradicionais

A tecnologia de RFID vem se mostrando muito promissora no futuro dos sistemas de identificação, ganhando seu espaço com as tecnologias já difundidas e conhecidas, como o código de barras [48]. Porém, ambas as tecnologias possuem vantagens e desvantagens.

#### **Comparação de etiquetas RFID com códigos de barras tradicionais**

- **Diferentes princípios:** Um código de barras é um identificador gráfico que representa um conjunto de informações. O princípio da RFID é a comunicação de dados sem contato entre o leitor e a etiqueta para atingir o objetivo de identificar o alvo [49];
- **Distância de leitura:** Os códigos de barras suportam apenas leitura de perto enquanto as etiquetas RFID suportam maior distância de leitura, dependendo da potência de sua antena[49];
- **Aplicativos diferentes:** O código de barras é usado principalmente em circulação de mercadorias, gerenciamento de livros, gerenciamento postal e sistemas bancários, enquanto o RFID é mais amplamente usado na gestão de animais, controle de acesso, controle de estacionamento, automação da linha de produção, gestão de ativos, gestão de lavanderia e gestão de eventos[49];
- **Capacidade de armazenamento diferente:** Os códigos de barras unidimensionais têm capacidade para 50 bytes, códigos de barras bidimensionais têm uma capacidade máxima de 2 a 3000 caracteres e RFID tem uma maior capacidade de memória[49];
- **Eficiência de digitalização diferente:** A frequência de reconhecimento de leitura de código de barras é limitada. Apenas um código de barras pode ser lido por vez enquanto o leitor RFID pode identificar e ler várias etiquetas de radiofrequência ao mesmo tempo[49];
- **Função de leitura e gravação:** O código de barras não tem função de leitura e gravação e não é possível adicionar informações ao código de barras impresso. As etiquetas de radiofrequência podem ser lidas e gravadas[49];
- **Desempenho de segurança:** Os dados RFID podem ser protegidos por tecnologia de criptografia que não é fácil de ser falsificada e alterada. Já as informações do

código de barras é exposta e geralmente usa regras de codificação comuns para que a segurança dos dados não possa ser garantida[49];

- **Vida útil:** O material do código de barras tradicional é o papel. Por isso é fácil ser danificado e violado; Um código de barras não pode ser alterado e não pode ser reutilizado após a impressão; O RFID armazena os dados no chip que é protegido com um invólucro feito em PVC, PET, papel ou outros materiais, dependendo de sua aplicação, podendo gravar e excluir dados diversas vezes aumentando sua vida útil[49];
- **Custo:** O sistema de código de barras é a tecnologia de identificação automática mais econômica e prática atualmente, devido à sua configuração flexível e baixo custo.

Embora o sistema RFID tenha muitas vantagens insubstituíveis, o alto custo de fabricação, e custo de equipamentos se tornou o principal obstáculo ao amplo uso, em comparação com o código de barras.[49].

## 2.7 Vantagens e desvantagens da utilização da tecnologia RFID

### 2.7.1 Dificuldades da aplicação da tecnologia RFID

Para fazer uma análise desse tópico, elencamos com base em uma adaptação do quadro apresentado por Moretti[15].

- Falta de apoio da alta gerência para a implantação do sistema RFID[15];
- Falta de conhecimento técnico avançado na empresa para auxiliar na implantação do sistema de RFID[15];
- Resistência dos colaboradores à implementação de novas tecnologias[15];
- Custo elevado para a implantação do sistema RFID[15];
- Dificuldade em definir o impacto estratégico positivo e o ganho real para os clientes com o resultado da implementação do sistema de RFID[15];
- Dificuldades associadas à comunicação entre diferentes grupos responsáveis pela implantação dos sistemas RFID e a empresa[15];
- Dificuldade de calcular o retorno dos investimentos na implantação de RFID[15];



- Problemas na adequação do layout da empresa para a instalação da infraestrutura necessária à implantação de RFID[15];
- Dificuldade de encontrar sistemas RFID user friendly para todos os usuários diretos e indiretos do novo sistema[15];
- A escassez de estudos de casos na literatura capazes de servir de benchmarking ou apoio para a implantação do sistema RFID[15];
- Dificuldade de entender como será feita a manutenção do sistema RFID depois de sua implantação[15];
- Problemas na migração do sistema de código de barras existente na empresa para o sistema RFID[15];
- Problemas na integração do sistema RFID com os sistemas de gestão já atuantes na empresa ERP e WMS[15];
- Dificuldade de comprovar para a direção da empresa que as melhorias nos indicadores são decorrentes da implantação do RFID[15];
- Dificuldade de garantir a segurança e a privacidade das informações ao se utilizar um sistema RFID[15].

## 2.8 Frameworks Usados no projeto

### 2.8.1 Ruby

Ruby é uma linguagem de script de alto nível, criada por Yukihiro Matsumoto em 1995 com o objetivo de ser mais poderosa do que Perl e mais orientada a objetos do que Python. Linguagens de scripts são frequentemente utilizadas para automatização e execução de tarefas. Os ambientes que podem ser automatizados através de scripting incluem aplicações de software, páginas web dentro de um navegador web, shells de sistemas operacionais, sistemas embarcados e até mesmo games. Ruby se baseia na programação multiparadigma, em outras palavras, suporta programação funcional, orientada a objetos, imperativa e reflexiva [50]. As principais características da linguagem são:

1. Sintaxe bem próxima à língua inglesa, fator esse que torna a linguagem simples e intuitiva.
2. Possibilidade da reutilização de componentes, o que facilita a configuração e reduz o tempo de desenvolvimento.

3. Facilidade em instalar e atualizar bibliotecas com uma linha de comando, através do RubyGems.
4. Possibilidade de emular a herança múltipla, através do uso de Mixins.
5. Linguagem de tipagem dinâmica. Dessa forma, todas as variáveis devem possuir um tipo, mas essa pode vir a ser alterada dinamicamente.
6. Ser multiplataforma, podendo ser executado no Microsoft Windows, Linux, Mac OS X, máquinas virtuais entre outros.

### 2.8.2 Rails

Criado em 2004, Rails é um framework de desenvolvimento projetado para facilitar a programação de sistemas web. Usado em conjunto com a linguagem de programação Ruby, os dois ficaram conhecidos como Ruby on Rails. Sua arquitetura se baseia no padrão Model View Controller (MVC) [51]. Onde a Model é responsável por gerenciar e controlar a forma como os dados se comportam por meio de funções, lógica e regras de negócios estabelecidas. View tem o papel de apresentar as informações de forma visual ao usuário entrando na parte de User Interface/User Experience (UI/UX). Por último, Controller é a camada responsável por intermediar as requisições enviadas pelo View com as respostas fornecidas pelo Model [52], [53]. O Rails já por padrão componentes nativos como:

- **Active Record:** Conecta classes a tabelas de banco de dados relacionais, para estabelecer uma camada de persistência de configuração quase zero para aplicativos. A biblioteca fornece uma classe base que, quando subclassificada, configura um mapeamento entre a nova classe e uma tabela existente no banco de dados. No contexto de um aplicativo, essas classes são comumente chamadas de modelos. Os modelos também podem ser conectados a outros modelos; isso é feito definindo associações [54].
- **Active Model:** O active model fornece um conjunto conhecido de interfaces para uso em classes de modelo. Eles permitem que os auxiliares do action pack interajam com modelos não active record, por exemplo. O active model também ajuda na construção de Object Relational Mapper (ORMs) personalizados para uso fora do framework rails[55].
- **Action Pack:** O action pack é uma estrutura para lidar e responder às solicitações da Web. Ele fornece mecanismos para roteamento (mapeamento de URLs de solicitação para ações), definição de controladores que implementam ações e geração de

respostas. Resumindo, o action pack fornece a camada do controlador no paradigma Model View Controller (MVC)[56].

- **Action View:** é uma estrutura para lidar com a pesquisa e renderização de modelos de visualização e fornece auxiliares de visualização que auxiliam na construção de formulários Hyper Text Markup Language (HTML), feeds atom e muito mais. Os formatos de modelo que o Action View manipula são ERB (Ruby incorporado, normalmente usado para inserir pequenos trechos de Ruby dentro de HTML) e eXtensible Markup Language (XML)[57].
- **Action Mailer:** O action mailer é uma estrutura para projetar camadas de serviço de e-mail. Essas camadas são usadas para consolidar o código para enviar senhas esquecidas, desejos de boas-vindas na inscrição, faturas para cobrança e qualquer outro caso de uso que exija uma notificação por escrito para uma pessoa ou outro sistema[58].
- **Action Mailbox:** O action mailbox roteia os e-mails recebidos para caixas de correio do tipo controlador para processamento no Rails. Ele é fornecido com entradas para Mailgun, Mandrill, Postmark e SendGrid. Você também pode lidar com e-mails de entrada diretamente por meio das entradas internas do Exim, Postfix e Qmail[59].
- **Active Job:** O active job é uma estrutura para declarar jobs e executá-los em uma variedade de backends de enfileiramento. Esses trabalhos podem ser desde limpezas programadas regularmente, cobranças, até correspondências – qualquer coisa que possa ser dividida em pequenas unidades de trabalho e executada em paralelo[60].
- **Action Cable:** O action cable integra perfeitamente web sockets com o resto de sua aplicação Rails. Ele permite que recursos em tempo real sejam escritos em Ruby no mesmo estilo e forma que o resto do seu aplicativo Rails, enquanto ainda são eficientes e escaláveis. É uma oferta de pilha completa que fornece uma estrutura javascript do lado do cliente e uma estrutura Ruby do lado do servidor. Você tem acesso ao seu modelo de domínio completo escrito com o active record ou o Object Relational Mapper (ORM) de sua escolha[61].
- **Action Storage:** O active storage simplifica o upload e referência de arquivos em serviços de nuvem como Amazon S3, Google Cloud Storage ou Microsoft Azure Storage e anexa esses arquivos ao active record. Suporta ter um serviço principal e espelhos em outros serviços para redundância. Ele também fornece um serviço de disco para testes ou implantações locais, mas o foco está no armazenamento em nuvem[62].

- **Active Text:** O action text traz conteúdo e edição de rich text para o Rails. Ele inclui o editor Trix que lida com tudo, desde formatação até links, citações, listas, imagens e galerias incorporadas. O conteúdo rich text gerado pelo editor Trix é salvo em seu próprio modelo rich text que está associado a qualquer modelo active record existente no aplicativo. Quaisquer imagens incorporadas (ou outros anexos) são armazenadas automaticamente usando o active storage e associadas ao modelo rich text incluído[63].
- **Action Support:** O active support é uma coleção de classes utilitárias e extensões de biblioteca padrão que foram consideradas úteis para o framework Rails. Essas adições residem neste pacote para que possam ser carregadas, conforme necessário em projetos Ruby fora do Rails[64].

### 2.8.3 Gems do Ruby on Rails

Uma gem do framework Ruby on Rails pode ser entendida como uma biblioteca de código aberto que contém um código Ruby que desempenha uma função. O uso de gems facilita a programação, pois essa permite utilizar as funcionalidades da gem sem a necessidade de inserir explicitamente o código no corpo do programa que está sendo desenvolvido.

Existem gems, como Rails por exemplo, que são programas Ruby independentes, os quais podem ser executados a partir de comandos. Outras não, funcionam melhor em conjunto com outras, e por essa razão são incrivelmente úteis quando utilizadas em um projeto [65].

As gems utilizadas no desenvolvimento do sistema foram:

1. **Chartkick:** para a criação de gráficos em JavaScript.
2. **Devise:** para soluções de segurança e autenticação.
3. **Httparty:** para requisições em APIs.
4. **WickedPDF:** para gerar PDFs a partir de HTML.
5. **Ransack:** para realizar as buscas de uma maneira mais simples rápida.
6. **Will Paginate:** para formatar a paginação do HTML.
7. **Faker:** para gerar dados fictícios para testes.

### 2.8.4 PostgreSQL

O PostgreSQL é um Data Base Management System (SGBD) de objeto-relacional de código aberto. Esse é derivado do pacote postgres, o qual teve seu início em 1986 na Uni-

versidade da Califórnia em Berkeley como um projeto do professor Michael Stonebraker, e ao longo dos anos tem recebido apoio de agências e instituições como a Defense Advanced Research Projects Agency (DARPA), Army Research Office (ARO) e National Science Foundation (NSF) além de contar com uma comunidade que realiza implementações de novas funcionalidades e atualizações constantes.

O PostgreSQL estende o padrão Structured Query Language (SQL) e adiciona recursos que melhoram a capacidade de armazenamento, dimensionamento, segurança e confiabilidade. Esses fatores são um diferencial, principalmente quando se trabalha com cargas de dados mais complicadas.

Desde de 2001 o PostgreSQL é compatível com Atomicity, Consistency, Isolation, Durability (ACID). Ele foi projetado para rodar em plataformas semelhantes ao UNIX. No entanto, o PostgreSQL também foi pensado para ser portátil, permitindo dessa forma, que esse pudesse ser executado em diversas plataformas, como Mac OS X, Solaris e Windows [66].

### **2.8.5 Bootstrap**

Inicialmente conhecido como Twitter Blueprint, o Bootstrap surgiu como um projeto interno do Twitter em meados de 2010, sendo desenvolvido por Mark Otto e Jacó Thornton. Seu propósito inicial era servir um guia de estilo para o desenvolvimento de ferramentas internas na empresa.

Atualmente o Bootstrap é um framework de código aberto voltado para o front-end mantido por sua comunidade, a qual é uma das maiores do Github. Seu uso é voltado para o desenvolvimento de sites e aplicações responsivas [67].

Esse framework possibilita o desenvolvimento de páginas web que se ajustam de forma dinâmica, de acordo com a tela do dispositivo em que está sendo exibida, permitindo assim uma maior compatibilidade com as variadas versões de navegadores web disponíveis no mercado como Google Chrome, Firefox, Internet Edge, Opera e Safari. Para que isso seja possível faz-se uso principalmente de Cascading Style Sheets (CSS) e plugins de JavaScript. Os estilos dentro das aplicações que fazem o uso do Bootstrap ficam com uma aparência mais bonita e moderna para a formatação de texto, tabelas e elementos de formulário [68].

A versão mais recente no momento é o Bootstrap 4. O projeto foi reescrito para levar em consideração duas alterações arquiteturais importantes: migração para o Syntactically Awesome Style Sheets (SASS) e a mudança para o flexbox do Cascading Style Sheets (CSS)[67].

### 2.8.6 Heroku

O Heroku é serviço de hospedagem em nuvem, conhecido também por Platform As A Service (PAAS) que permite o deploy de várias aplicações backend. O Heroku é bastante versátil e permite que um deploy seja feito diretamente por um repositório no GitHub, através da branch mais atualizada do projeto e de forma automática, para testes em produção ou na escalabilidade no seu projeto [69]. Na versão gratuita são disponibilizadas 512 megabytes de Random Access Memory (RAM) no servidor onde sua máquina virtual está executando seu projeto ou aplicação e, se ficar inativa por 30 minutos, o Heroku automaticamente coloca ela em um estado de hibernação [70].

## 2.9 Leitor de tags RFID

Para nosso projeto, optamos por utilizar o leitor de tags portátil CF-H301 do fabricante Chafon, o qual é apresentado na Figura 2.2, pois esse pode ser amplamente aplicado em muitos sistemas de aplicação RFID, como gerenciamento de estoque, sistema de rastreabilidade, identificação de pessoal, controle de acesso e sistema anti-falsificação. As funções dos leds do aparelho são explicadas em detalhes conforme é apresentada na (Tabela 2.3).

Para utilizar a leitora é necessário pressionar o Interruptor de Força *Power Switch* e para realizar a leitura é necessário pressionar o Botão de Gatilho *Trigger Button*, conforme são apresentados na Figura 2.3 e na Tabela 2.4.

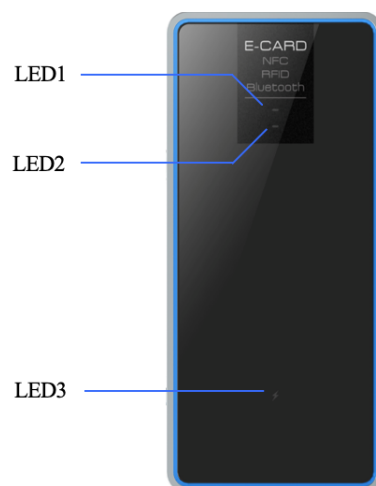


Figura 2.2: Leitora RFID [1]

Tabela 2.3: Leds indicadores da leitora

Item	Descrição
LED1	Luz vermelha acende ao ativar o gatilho. Pisca verde ao ler uma tag com sucesso.
LED2	Luz vermelha pisca lentamente para indicar que o leitor está ligado.
LED3	Vermelho aceso constante significa bateria fraca. Verde aceso durante o carregamento da bateria.

Segundo as características técnicas do leitor detalhadas pelo fabricante, destacamos:

- Design fino e bem portátil [71].
- Bateria econômica, mais de 3 horas de leitura contínua e mais de 1 dia em standby [71].
- Suporte ao modo de trabalho interativo e acionador ativo [71].
- Suporte ao protocolo ISO18000-6C (EPC C1G2) [71].
- Faixa frequência de 902 - 928 MHz ou 865 - 868 MHz [71].
- Transmissão FHSS ou Frequência Fixa [71].
- Antena TX/RX integrada com distância efetiva de até 1 metro [71].
- Potência de saída RF de 0 a 26dbm (ajustável) com suporte RSSI [71].
- Velocidade de leitura 50 tags por segundo [71].
- Tamanho do buffer de tag 370 tags Max 128 bits EPC ou 120 tags Max 496 bits EPC [71].
- Suporta Bluetooth 4.0. Bateria de lítio embutida sem a necessidade de fonte de alimentação externa [71].
- Interface USB1.1. com DLL e software de demonstração para facilitar o desenvolvimento [71].
- Suporta carregamento USB e também o indicador de carregamento, o indicador de bateria fraca [71].

O leitor se comunica com o host por meio da interface de comunicação Bluetooth e completa a operação correspondente de acordo com o comando do host. O parâmetro de comunicação serial padrão é 9600bps, 1 bit de início, 8 bits de dados, 1 bit de parada sem

bit de verificação de paridade. No processo de comunicação, o bit menos significativo de um byte é transmitido primeiro e o byte menos significativo da sequência de dados de comando é transmitido primeiro.

Um procedimento de comunicação é sustentado pelo host que envia comandos e dados ao leitor, o qual retorna o status do resultado, assim como os dados ao host após a execução do comando. O leitor executa um comando após recebê-lo. Somente após a conclusão da execução do comando, o leitor estará disponível para receber o próximo comando. Durante a implementação de um comando, o leitor ignora todos os outros dados de comando recebidos.

O intervalo entre dois bytes consecutivos no bloco de dados de comando deve ser menor que 15ms. Durante o envio do bloco de dados de comando, a sincronização será perdida se o host receber qualquer dado do leitor, assim como o envio será interrompido e a comunicação será reiniciada após 15ms. O leitor conclui a execução do comando após receber o comando do host e retorna os resultados. Durante esse período, ele não processa nenhum dado do host.



Figura 2.3: Fonte dos autores [1]

Tabela 2.4: Parte lateral da leitora

Item	Descrição
Botão de gatilho	Ativa o leitor para ler tags UID e enviar para o host
Interruptor de Força	Mover para a esquerda para ligar e à direita para desligar



### 2.9.1 SDK do Leitor

O fabricante da leitora modelo CF-H301 é a empresa Chafon. No site desta é disponibilizado, na área de "Support - Downloads", um Software Development Kit (SDK) desenvolvido na linguagem Java, a qual roda nativamente em dispositivos Android. Ao executar o Android Application Pack (APK) gerado por esse kit de desenvolvimento oferecido pela fabricante, em um aparelho Android, o celular se conecta ao hardware de leitura via bluetooth e ao selecionar a opção buscar dispositivo, é apresentada ao usuário na tela a visualização de todos os dispositivos com a conexão bluetooth ativa. O nome do device a ser conectado é **UBT19471077B** e seu endereço Media Access Control (MAC) é **00:15:90:91CA:49**. Após ser conectado o Universally Unique Identifier (UUID) das tags são lidas pelo hardware do leitor.

Com esse SDK temos a possibilidade de desenvolver uma solução voltada a uma aplicação específica e de acordo com a necessidade a ser atendida.

### 2.9.2 O Sistema Operacional Android

Inicialmente produzido pela empresa Android.inc e lançado pelo Google em 2007, em conjunto com a Open Handset Alliance que é uma aliança formada por diversas empresas do mercado e do ramo de tecnologia com o propósito de criar padrões open source para telefonia móvel. Entre as empresas participantes estão Google, HTC, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile e Nvidia. A Plataforma Android é composta por um sistema operacional baseado em um núcleo Linux, projetado visando dispositivos móveis com telas sensíveis ao toque como celulares e tablets que operam com processadores ARM. Atualmente já existem versões destinadas a processadores X86, arquitetura de processadores essa que é a mais comum de ser encontrada em computadores pessoais [72].

Suas aplicações, bibliotecas e ferramentas de desenvolvimento desde seu lançamento se tornou atrativa ao interesse de companhias, desenvolvedores e do público em geral. Desde então, esta plataforma de software tem passado por constantes processos de melhoramento, tanto em termos de recursos como o suporte a novos dispositivos de hardware como televisores, relógios inteligentes, assistentes pessoais e até mesmo em veículos automotores [73]. Atualmente, o Android é o sistema operacional móvel mais popular entre os dispositivos de sua categoria, e o mais usado no mercado de smartphones no Brasil [74].

Isso ocorre, principalmente, devido à ampla faixa de preços dos aparelhos que conseguem executar o sistema, desde os de entrada até os dispositivos *high end*, além de contar com uma grande quantidade de aplicativos disponíveis para a plataforma. Desse modo, a escolha desse sistema operacional para a realização do presente trabalho foi devido ao

fato do Android ser o sistema mais acessível e utilizado pelo brasileiro, ser open source e o SDK do leitor ter vindo na linguagem JAVA que é nativa a esse sistema [75].

### 2.9.3 Bluetooth Low Energy (BLE)

A tecnologia bluetooth foi desenvolvida inicialmente pela Ericsson Mobile Communications, em 1994, e veio com o intuito de eliminar o uso cabos nas conexões e na transmissão de dados e arquivos de maneira rápida e segura entre dispositivos eletrônicos, sejam esses portáteis ou fixos. Para que isso seja possível, os dispositivos precisam estar próximos um do outro[76].

Com a ascensão dos smartphones e a queda nos custos desses aparelhos, tais dispositivos ganharam uma grande parte do mercado se tornando populares em todas as classes sociais, e juntamente com isso, os gadgets e acessórios capazes de se conectar com esses dispositivos também cresceram. Eles ficaram conhecidos também como "Beacons" e são capazes de localizar e mandar informações em tempo real de acordo com a proximidade entre eles e o smartphone ou tablet, que por sua vez atua como receptor e processador de dados[76].

Os "Beacons" utilizam a tecnologia sem fio Bluetooth Low Energy (BLE) e, o baixo consumo de energia torna esses dispositivos mais eficientes consumindo até cerca de 20% a menos de energia do que a tecnologia bluetooth convencional [76]. Devido ao baixo consumo de energia ser uma característica principal, isso o torna eficiente para aplicações de reconhecimento de posição.

O sistema de transferência bluetooth funciona por meio de uma arquitetura dividida em camadas, na qual a camada lógica divide-se em enlace lógico e comunicação lógica, e com isso há diversas estratégias de proteção para cada camada do sistema[77].

Projetado para IOT, esta tecnologia consome pouca energia e é muito eficiente, o que a torna ideal para dispositivos que utilizam pequenas células de energia, como baterias de moeda e dispositivos de captação de energia. Os serviços bluetooth dividem-se em três categorias[76]:

- **serviço de controle de dispositivos:** Responsáveis por modificar o funcionamento e modos dos dispositivos.
- **serviços de controle de transporte:** Os quais criam, modificam e liberam portadores de tráfego (Canais e enlace).
- **serviço de dados:** Utilizados para enviar a informação de transmissão através de portadores de tráfego.

## 2.10 Trabalhos Correlatos

Costa [78] propôs um sistema que atua como ferramenta auxiliar para o controle do inventário de loja. Esse sistema possui as funcionalidades de cadastro de produtos, alterar o estado de "disponível" para "Produto pago" e fazer a segurança, ao disparar um alarme caso o produto não esteja pago ao tentar fazer a remoção desse da loja.

Mekruksavanich [79] idealizou um sistema que facilita as compras ao implementar o uso do RFID no carrinho de supermercado. O sistema mostra os produtos disponíveis no mercado e a localização deles, assim como daqueles que o cliente já possui no carrinho. Além disso, também providencia uma plataforma que automatiza a cobrança dos itens no carrinho, tornando assim a experiência de compra mais agradável, conveniente e eficiente, bem como menos demorada para o cliente.

Santos [80] apresentou o desenvolvimento de uma arquitetura abstrata de gerenciamento de dispositivo de IOT, tanto na camada de hardware como na de software que seja capaz de coletar, processar e armazenar dados de leitura de tags RFID, utilizando a frequência UHF em um ambiente de computação em nuvem e microsserviços. A arquitetura proposta é implementada de acordo com a arquitetura padrão em Internet das Coisas e composta por cinco camadas: Camada de Percepção, Camada de Rede, Camada de Middleware, Camada de Aplicação e Camada de Negócio.

Sangari *et al.* [81] apresentaram uma proposta para notificar o administrador sobre o nível de reabastecimento do estoque, assim como a venda instantânea por meio do sistema de mensagens inteligentes. Este procedimento visa ajudar o lojista a monitorar os estoques de maneira remota e inteligente. O sistema proposto reduz o trabalho manual ao utilizar a tecnologia RFID, tornando possível identificar os detalhes do estoque. As contagens dos produtos são atualizadas nas páginas da web e o administrador recebe a notificação quando a contagem do produto diminui.

Adiono *et al.* [82] desenvolveram um sistema composto por um leitor RFID, um smartphone e um banco de dados em nuvem. O smartphone se conecta ao leitor RFID via Bluetooth e ao sistema em nuvem via rede móvel ou Wireless Fidelity (Wi-Fi). O sistema em nuvem é utilizado para receber os dados de compra do site de e-commerce e atualizar seu status para o cliente. O leitor RFID efetua a pesquisa e aponta localização do item comprado. O smartphone integra os dados do leitor RFID com a nuvem e instrui o leitor RFID a pesquisar o item após receber um pedido de compra do sistema. O aplicativo do smartphone irá então atualizar o banco de dados de inventário com base no status do pedido de pesquisa.

Safdar *et al.* [83] propuseram uma estrutura automática do fluxo de um paciente em um hospital, fazendo uso de RFID em cada etapa do atendimento (registro, sinais vitais, médico) para registrar o tempo e a localização do paciente. Ao utilizar esse sistema o

paciente não precisa interagir com a equipe a todo momento, apenas passam seu cartão RFID em cada estação, o que salva seu tempo e localização. O sistema visa auxiliar a gestão do hospital. A experiência do paciente é calculada usando um algoritmo genético que leva essas informações de tempo, dados do sensor e dados de pesquisa como entrada e gera estatísticas otimizadas. Por conta da análise em tempo real, as ações apropriadas podem ser tomadas rapidamente em vez de esperar pela análise manual. Além disso, esse sistema se mostra útil na manutenção do distanciamento social no caso de doenças virais como (SARS, COVID), pois cada paciente tem seu próprio cartão, eliminando assim a necessidade de filas e a interação com outros pacientes durante a espera.

Jumin *et al.* [84] projetaram um sistema que combina robótica e tecnologia RFID para realizar a gestão de armazéns de produtos químicos. Com o intuito de resolver duas questões, a detecção da quantidade restante de produtos químicos e a localização desses. Para detecção, foram utilizados dois indicadores: a indicação de força do sinal recebido e taxa de leitura, como entrada para determinar se a quantidade restante de produtos químicos era suficiente. Para localização, foi utilizada a etiqueta RFID com o movimento da antena do leitor, para que fosse possível obter a posição das etiquetas por ajuste de curva. Os resultados experimentais mostram que o detector alcançou cerca de 93% de precisão da quantidade restante de detecção e 92% de precisão de posicionamento para produtos químicos. Portanto, o detector de radiofrequência pode ser usado em depósitos de produtos químicos para ajudar os gerentes a detectar a quantidade restante de produtos e localizá-los, reduzindo muito sua carga.

Sourour *et al.* [85] apresentaram uma abordagem de armazenamento baseada em classes de produtos para um armazém. Esta abordagem proporciona uma melhor resposta para alocar produtos em locais compatíveis e adequadas no armazém, garantindo a segurança e proteção, tanto da mercadorias quanto das pessoas. A implantação da plataforma RFID e IOT representa uma solução ideal para implementar essa proposta de gestão de armazéns e para criar colaboração entre produtos e prateleiras. O monitoramento das restrições de compatibilidade do produto garantirá uma eficiência de armazenamento com base na classe do produto e reduzirá o tempo de armazenamento ao minimizar o caminho seguido por cada produto, enquanto determina sua rota de armazenamento apropriado.

López *et al.* [86] apresentaram um hardware RFID, de arquitetura básica para o rastreamento de itens médicos e medicamentos, com o intuito de reduzir o risco de erros médicos em hospitais, especificamente no campo da administração de medicamentos. Esse hardware pode ser facilmente integrado aos serviços de informação médica e à infraestrutura de rede já implantada na maioria dos hospitais. Um aplicativo móvel também foi implementado para mostrar informações em tempo real sobre os itens marcados com etiquetas RFID, com o objetivo de torná-lo o mais simples possível para que todos os

usuários possam entender o funcionamento.

Zhang *et al.* [87] apresentaram um sistema de gerenciamento de armazenamento a frio de kiwis com base em RFID. Com esse sistema proposto é possível obter dados da fruta como a variedade, quantidade, local de armazenamento, tempo de armazenamento, local de origem, operador e informações de registro através da tecnologia RFID. Dessa forma, o problema da gestão tradicional de câmaras frigoríficas, o qual depende do registro manual dessas informações, é agilizado. Os problemas de baixa eficiência e baixa confiabilidade são solucionados com a coleta de dados em tempo real das mercadorias, o que torna o processo de armazenamento e entrega de mercadorias mais conveniente. Conforme é apresentado na Tabela 2.5 é uma síntese dos trabalhos correlatos e o principal diferencial em relação ao proposto neste trabalho.

Tabela 2.5: Trabalhos correlatos versus nossa proposta. Fonte: Própria dos autores

	<b>Interface Amigável</b>	<b>Integração APP e Web</b>	<b>Dados armazenados em Cloud</b>	<b>Precisão dos Dados</b>	<b>Controle Eficiente</b>
<b>Nossa proposta</b>	x	x	x	x	x
[78]	-	-	-	x	x
[79]	-	x	x	x	x
[80]	x	-	x	x	x
[81]	-	x	-	x	x
[82]	x	x	x	x	x
[83]	-	x	-	-	-
[84]	-	-	-	x	x
[85]	-	-	x	x	x
[86]	-	-	-	x	x
[87]	-	-	-	x	x

## 2.11 Síntese do Capítulo

Neste **Capítulo 2** destacamos os principais conceitos necessários para uma boa compreensão quanto ao uso e aplicações do RFID, além de apresentarmos normas de padrões de uso e onde essa tecnologia vem sendo utilizada em nosso cotidiano. O RFID é uma tecnologia muito promissora e está em ascensão devido à grande visibilidade da indústria 4.0 e a necessidade da troca rápida de informações decorrente da aplicação de automação por meio do IOT. Apresentamos também outros conceitos que serão necessários para um melhor entendimento sobre as tecnologias que dão suporte para a criação e funcionamento do sistema desenvolvido, tais como: linguagem de programação Ruby, framework Rails, Gems do Ruby on Rails, PostgreSQL, Bootstrap, Heroku, Android, leitora de tags RFID. No capítulo 3 serão abordado as fases do desenvolvimento do WMS.

# Capítulo 3

## Desenvolvimento do Sistema

### 3.1 Arquitetura Proposta

O objetivo deste trabalho é desenvolver um sistema de gerenciamento de estoque aliado à tecnologia de identificação por radiofrequência, onde tags inteligentes são utilizadas para uma melhor organização e gestão dos ativos. Os dados disponíveis nas tags, os quais serão armazenados pelo sistema, serão lidos através de um I-Card espécie de leitor scanner de RFID portátil com conexão Bluetooth que através de um aplicativo os dados serão enviados para um dispositivo móvel e salvo em uma API, visando assim uma interoperabilidade da aplicação. Nesse caso na gestão do estoque, podendo essa ser feita via web ou aplicação mobile. Para tal realização da proposta, implementamos uma API para que os dados sejam lidos e armazenados, o sistema responsável por tratar esses dados e fazer a gestão de forma eficiente, e o aplicativo no qual o aparelho I-Card se conecta para que sejam exibidos as informações das tags e salvas em nosso web service.

A linguagem de programação escolhida para o desenvolvimento do back-end da API e do sistema web foi o Ruby on Rails. Para o usuário ter uma experiência visual melhor e mais agradável, fizemos o uso do Bootstrap que é um framework de front end e se encontra em sua versão 5. E para o aplicativo mobile a fabricante do leitor, a empresa Chafon fornece um SDK feito em Java no qual pode ser modificado para atender as necessidades do projeto.

### 3.2 Sistema

#### 3.2.1 Preparação do Ambiente

Em sistemas basados em UNIX como LINUX e MacOS, a instalação dos softwares podem ser facilitadas e feitas por meio de instalador de pacotes através de linhas de código. Nesse

caso, como os desenvolvedores usaram o sistema operacional MacOS o guia vai seguir o seguinte passo a passo para instalação do mesmo.

- **Instalando Homebrew**

```
/bin/bash-c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

- **Instalando a última versão do Ruby**

```
rbenv install 3.0.3  
rbenv global 3.0.3
```

- **Instalando a última versão do Rails**

```
gem install rails -v 7.0.0
```

- **Instalando a última versão do PostgreSQL**

```
brew install postgresql
```

- **Instalando a última versão do Android Studio**

```
brew install --cask android-studio
```

- **Instalando a última Visual Studio Code**

```
brew install --cask visual-studio-code
```

### 3.3 Desenvolvimento do Aplicativo Android

O aplicativo é derivado do SDK disponibilizado pela fabricante do scanner em seu site. Ele foi desenvolvido na linguagem Java e é capaz de rodar somente em dispositivos Android. Para iniciar o projeto houve uma necessidade de atualizar os arquivos do SDK, pois eles não estavam no formato para serem executados no Android Studio, Integrated Development Environment (IDE) responsável por ajudar no desenvolvimento de aplicativos voltado para plataforma Android. O mesmo se encontrava bastante desatualizado e com vários erros, impossibilitando que fosse compilado. Feitas as devidas correções e atualizações no código, o projeto do aplicativo estava pronto para ser iniciado.

O aplicativo tem o seguinte funcionamento: por meio da conexão bluetooth feita através do smartphone, o próprio aplicativo faz a busca por outros dispositivos mostrando na tela, dando a opção de conexão ao usuário e em qual dispositivo ele tá afim de se conectar. Previamente no código fonte do aplicativo já vem todas as permissões de conexões pré estabelecidas, principalmente as conexões com o hardware de leitura para a captura das tags com o smartphone e o scanner de leitura, com o dispositivo móvel devidamente pareado o usuário tem a opção de fazer as leituras das tags que operam na mesma frequência do leitor, exibindo na tela o EPC e quantas vezes por segundo essa tag foi lida.

O SDK é bastante simples, porém cumpre muito bem com o seu propósito que é ser um leitor barato com um bom "custo x beneficio", capaz de ler e identificar tags de radiofrequência que operam na frequência UHF como vimos no funcionamento das faixas



de frequência 2.1. Partindo desse ponto que o aplicativo cumpre com o seu papel, que é somente realizar essas leituras. Surge um desafio: o que fazer com as tags lidas? A solução proposta no desenvolvimento do sistema é de multiplataforma, capaz de garantir um gerenciamento de forma eficiente usando a tecnologia de rádio frequência. Como o SDK disponibilizado pela empresa fabricante do leitor já veio desenvolvida em Java para rodar em smartphones, com o sistema operacional Android temos um impasse, pois restringimos os usuários de outros sistemas operacionais. A saída para driblar esse impasse foi o desenvolvimento de uma API, usando a tecnologia Ruby on Rails no padrão Representational State Transfer (REST), na qual fizesse o envio de todas as tags lidas dentro próprio APP para um servidor na nuvem e retornasse um JavaScript Object Notation (JSON). O benefício de uso da API é que ela pode ser usada em vários outros sistemas preservando sua base de dados, aumentando, dessa forma, o range de usuários que podem utilizar o sistema.

Retorno do JSON da API:

```
{
  "id": 1061,
  "rfidCode": "302EF13EFDD0008000000015",
  "created_at": "2022-02-15T13:21:39.490Z",
  "updated_at": "2022-02-15T13:21:39.490Z"
}
```

### 3.4 Desenvolvimento da Aplicação Web

A aplicação web foi desenvolvida afim de atender os levantamentos de requisitos apontados anteriormente no capítulo 1. O sistema foi criado com o objetivo de automatizar, gerir de forma automatizada clientes, fornecedores, produtos, categorias, compras e vendas. O desenvolvimento da aplicação tem seu backend feito em Ruby on Rails, seu frontend em Bootstrap que é um framework HTML e o seu banco de dados em PostgreSQL. Podemos enxergar o sistema de gerenciamento de estoque como um único grande ecossistema que por sua vez em seu desenvolvimento foi dividido em pequenos ecossistemas no qual chamamos de módulos conforme é apresentado na Figura 3.1, com intuito de simplificar o entendimento das funcionalidades que ira englobar o sistema e o seu desenvolvimento por parte da sua equipe de desenvolvedores no final esses pequenos módulos trabalhando em conjunto formando o sistema por um todo seguindo as boas práticas de engenharia e construção de software segundo Pressman (2011)[88]. A seguir temos os seguintes módulos:

- **Autenticação**

O login do sistema foi feito com a ajuda de uma Gem do Rails chamada Devise. O login é responsável por autenticar o usuário e conceder as permissões para que o mesmo possa ter acesso a todas funcionalidades do sistema.

- **Dashboard**

Na dashboard temos alguns dados e informações em tempo real do sistema, como a quantidade de produtos total cadastrados, acompanhamento mensal através de gráfico sobre o ingresso de novos clientes, compras e vendas. Temos disponível o fluxo de caixa. A entrada de novos produtos é caracterizada como compras, a saída dos produtos como venda e total definido como carteira. E o histórico com data e hora das últimas movimentações de clientes, fornecedores, produtos, compras, vendas e categorias.

- **Cadastros**

O módulo de cadastro engloba todas as tabelas que fazem a operação de Create, Read, Update, Delete (CRUD) que são elas clientes, fornecedores, produtos, compras, vendas e categorias alertas e geração de relatórios para melhor gestão do sistema.

- **Busca RFID**

A busca é feita via API na qual recebe o número do EPC do aplicativo de leitura. O sistema faz a checagem do campo *rfidCode* na tabela de produtos e compara se ele existe API. Se esse número existir somente na API o sistema me mostra que eu não tenho aquele EPC cadastrado em minha base de dados e me possibilita cadastrá-lo. Se o meu EPC tiver tanto na API quanto salvo na tabela *products* no banco de dados, através de uma consulta rápida o sistema vai retornar na tela do usuário que eu tenho esse EPC juntamente com o ID do produto, permitindo a visualização exata do produto que queremos consultar. O campo *rfidCode* na criação do produto é um campo obrigatório, garantindo o preenchimento do mesmo e assegurando essas comparações.

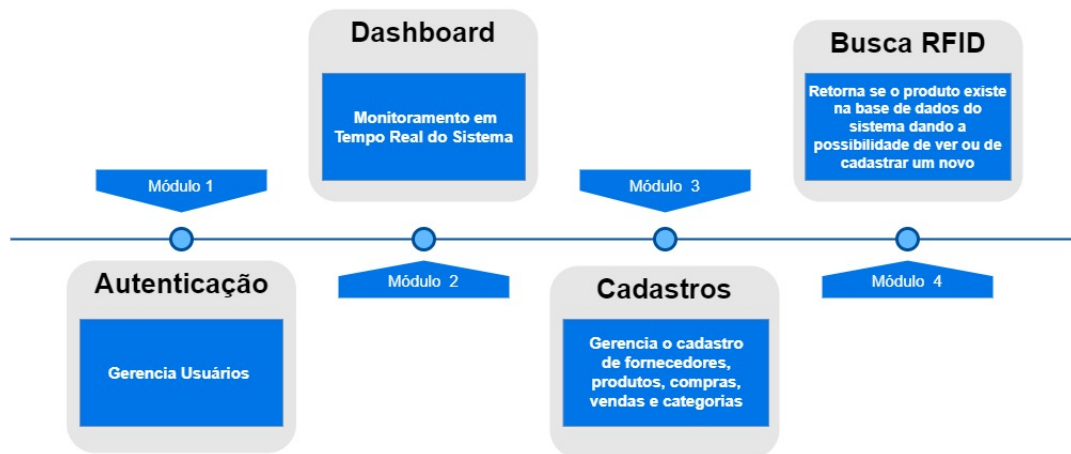


Figura 3.1: Módulos do sistema. Fonte: Própria dos autores

### 3.5 Requisitos Funcionais

Os requisitos funcionais de um sistema são as funcionalidades do sistema em si e como ele reage as entradas fornecidas pelo o usuário, como se comporta em determinadas situações (SOMMERVILLE,2011)[89]. Esses requisitos estão relacionados às atividades que o sistema realiza, como:

- Criar/Excluir/Editar/Consultar nome e endereço de clientes e fornecedores.
- Criar/Excluir/Editar/Consultar produtos e categorias.
- Criar/Excluir/Editar/Consultar compras e vendas.
- Emissão de relatórios de clientes, fornecedores, produtos, compras e vendas.
- Geração de relatório de determinado período de vendas.
- Geração de relatório de determinado período de compras.
- Visualização do fluxo de caixa.
- Importação de produtos através de arquivo CSV.
- Consulta de saldo ou estoque.
- Busca Tag RFID.

## 3.6 Requisitos Não Funcionais

Os requisitos não funcionais são aqueles que não são diretamente relacionados às funções específicas fornecidas pelo sistema. Raramente eles estão associados às características individuais do sistema, sendo aplicados frequentemente ao sistema como um todo (SOMMERVILLE,2011)[89] como:

- Multiplataforma para rodar em Windows, Linux e macOS.
- Pode ser acessado por meio de um navegador.
- Permite somente o acesso de pessoas autorizadas.
- Possui uma interface intuitiva e de fácil utilização para os usuários.
- O sistema deve ser desenvolvido utilizando a linguagem de programação Ruby on Rails, utilizando o banco de dados PostgreSQL e o servidor de nuvem Heroku.
- Responsivo, sua interface se adapta a qualquer tipo de tela.

## 3.7 Modelagem do Banco de Dados do Sistema Web

Como podemos ver na modelagem do sistema apresentada na Figura 3.2, o sistema possui 9 tabelas para executar a aplicação. Para o desenho do Modelo Entidade Relacionamento (MER), foi usada a ferramenta online e gratuita Draw.io. Essa ferramenta permite que seja esboçado vários tipos de diagramas voltados para área de desenvolvimento de softwares de forma prática e fácil, através de uma interface gráfica. Para a criação foi usada o framework Ruby on Rails juntamente com o PostgreSQL. O uso do Rails através do Active Record trás uma camada de abstração bem mais simplificada, permitindo comandos Structured Query Language (SQL) sem usar diretamente comandos SQL. Nesse caso o responsável por fazer essas chamadas seria o próprio Rails usando a linguagem Ruby. Para a administração desse banco foi usado o próprio *Data Base Management System* (SGBD) do PostgreSQL.

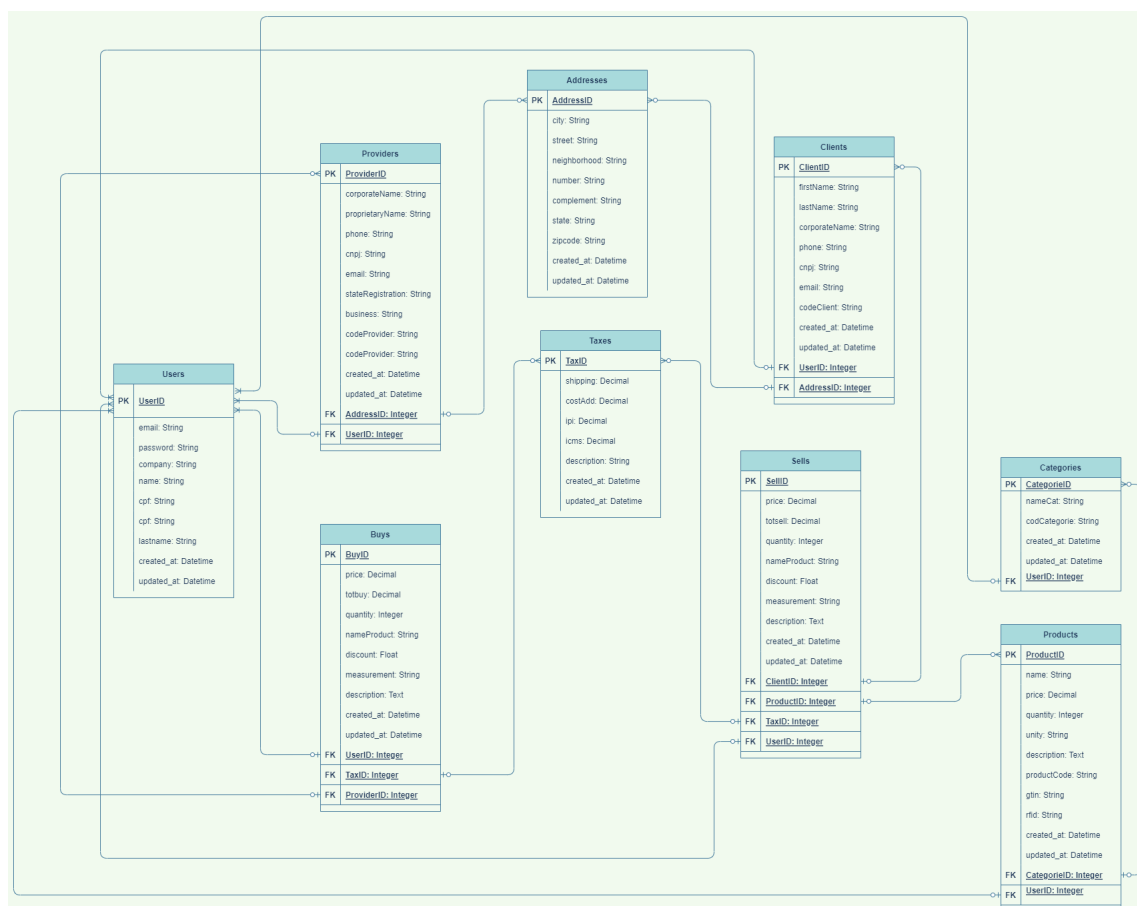


Figura 3.2: Modelo de Entidade Relacional Sistema Web. Fonte: Própria dos autores

### 3.8 Diagrama de Casos de Uso

Considerando os requisitos apresentados na Figura 3.1 com a divisão do sistema projetada em módulos, foi possível delimitar as suas funcionalidades para conseguir atingir o objetivo proposto pelo trabalho de conclusão de curso, conforme é exibido na Figura 3.3 no diagrama de casos de uso, o sistema possui apenas um ator que é o user, o user tem todas as permissões de gerenciamento no módulo de cadastro ações como criar, editar e excluir com interações diretas nos agentes clientes, fornecedores, compras e vendas. O user ainda tem a possibilidade de usar o APP do scanner para enviar as tags para a API que é integrada diretamente ao sistema para saber se tal produto é pertencente ao seu estoque.

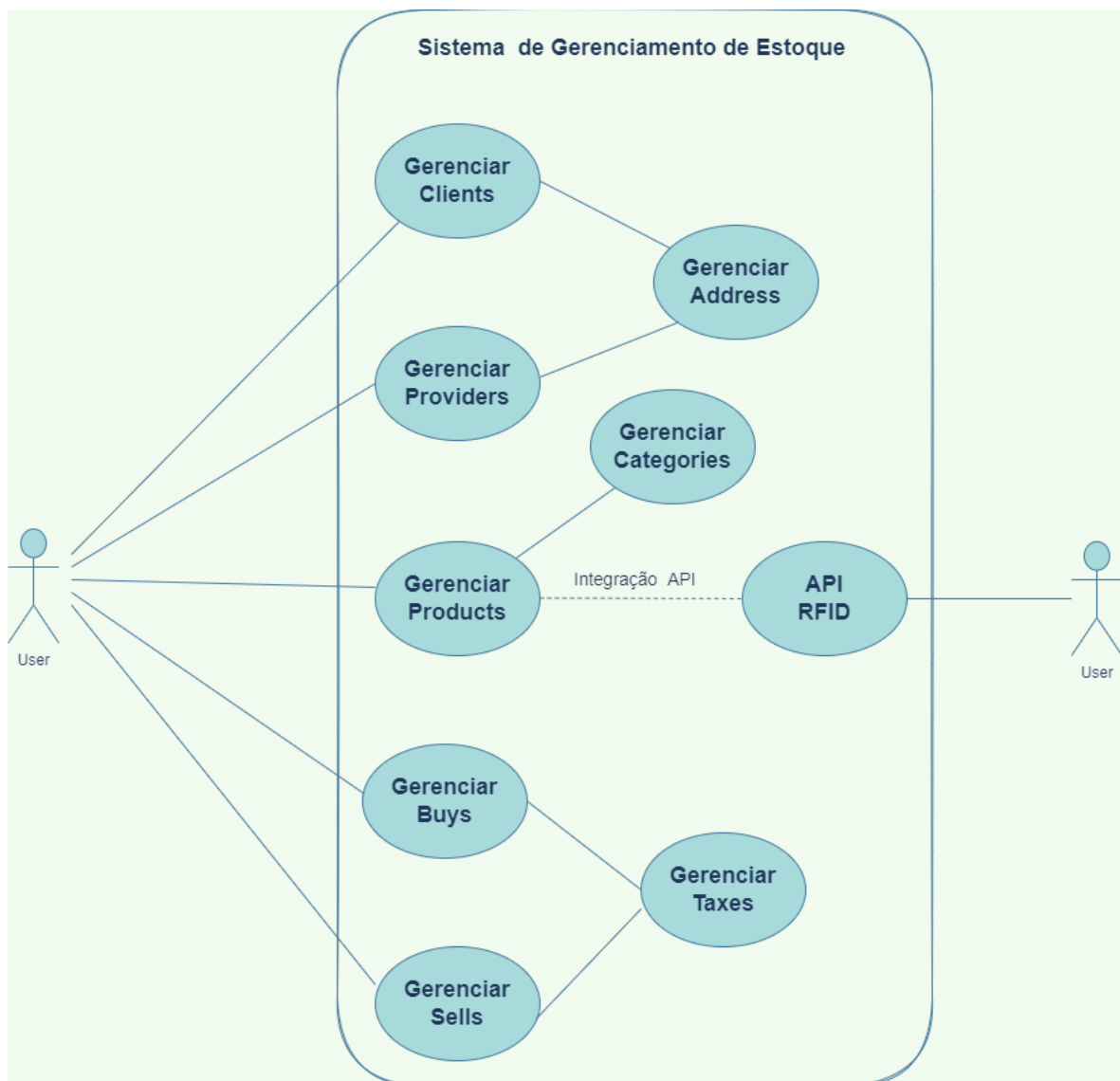


Figura 3.3: Modelo de Entidade Relacional Sistema Web. Fonte: Própria dos autores

## 3.9 Modelagem do Banco de Dados da API

A criação do banco de dados da API segue o mesmo padrão da criação do sistema web, pois eles foram desenvolvidos na mesma linguagem de programação. A API se baseia em somente na Tabela 3.1, com o campo código do RFID que é responsável por armazenar o EPC das tags lidas pelo scanner através do aplicativo e enviado para a API como podemos ver na modelagem Conforme apresentado na Figura 3.4.

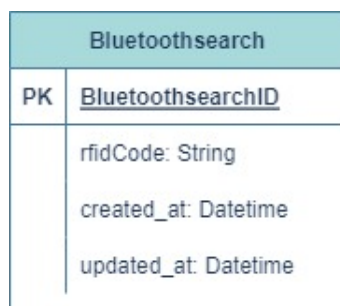


Figura 3.4: Modelo de Entidade Relacional API. Fonte: Própria dos autores

### 3.9.1 API

A API foi desenvolvida conforme é apresentado na Tabela 3.1, que desempenha o papel de armazenar os códigos das tags de RFID lidos pelo scanner enviado através de uma funcionalidade desenvolvida no aplicativo onde o dado a ser salvo é mandado por meio de uma requisição REST do método POST.

Tabela 3.1: Tabela Bluetoothsearch. Fonte: Própria dos autores

Campo	Tipo	Obrigatório	Descrição
Bluetoothsearch_id	INTEGER	Sim	Chave Primária
rfidCode	STRING	Sim	Armazena o código da tag RFID lida pelo scanner

### 3.9.2 Fornecedores

Conforme é apresentado a Tabela 3.2 guarda as informações dos fornecedores, os quais tem uma relação de venda com o usuário. Os campos dessa tabela são formados pelo nome da empresa, nome do proprietário ou representante legal, telefone, CNPJ, e-mail, inscrição estadual, ramo de negócio que a empresa atua e o código do fornecedor que é o ID atribuído a ele de forma automática dentro da tabela. Temos também como chave estrangeira e obrigatória os campos endereço vindo da Tabela 3.10 para complemento de dados e qualidade das informações dos fornecedores e o usuário vindo da Tabela 3.9 se ter o registro de criação do fornecedor.

Tabela 3.2: Tabela Providers. Fonte: Própria dos autores

<b>Campo</b>	<b>Tipo</b>	<b>Obrigatório</b>	<b>Descrição</b>
provider_id	INTEGER	Sim	Chave Primária
corporateName	STRING	Sim	Armazena o nome do fornecedor
proprietaryName	STRING	Sim	Armazena o nome do proprietário da empresa de fornecimento
phone	STRING	Sim	Armazena o número do celular do fornecedor
cnpj	STRING	Sim	Armazena o número do CNPJ do fornecedor
stateRegistration	STRING	Não	Armazena a Inscrição Estadual do fornecedor
e-mail	STRING	Sim	Armazena o E-mail do fornecedor
business	STRING	Não	Armazena o tipo de negócio da empresa do fornecimento
codeProvider	INTEGER	Não	Armazena o ID do fornecedor
description	STRING	Não	Armazena a descrição do fornecedor
address_id	INTEGER	Sim	Chave Estrangeira
user_id	INTEGER	Sim	Chave Estrangeira



### 3.9.3 Clientes

Conforme é apresentado a Tabela 3.3 guarda as informações do cliente, tais como nome, sobrenome, nome da empresa no qual ele é o proprietário ou representante legal, e-mail, CNPJ e algum telefone para contato. Dentro da concepção do sistema o cliente é aquele no qual possui alguma relação de compra com o usuário. Temos também como chave estrangeira e obrigatória os campos endereço vindo da Tabela 3.10 para complemento de dados e qualidade das informações dos fornecedores e o usuário vindo da Tabela 3.9 para se ter o registro de criação do cliente.

Tabela 3.3: Tabela Clients. Fonte: Própria dos autores

<b>Campo</b>	<b>Tipo</b>	<b>Obrigatório</b>	<b>Descrição</b>
client_id	INTEGER	Sim	Chave Primária
corporateName	STRING	Sim	Armazena o nome da empresa do cliente
clientName	STRING	Sim	Armazena o nome do proprietário da empresa do cliente
lastName	STRING	Sim	Armazena o sobrenome do proprietário da empresa do cliente
cnpj	STRING	Sim	Armazena o número do CNPJ do cliente
phone	STRING	Não	Armazena o telefone do cliente
email	STRING	Sim	Armazena o Email do cliente
codeClient	INTEGER	Não	Armazena o código ID do cliente
description	STRING	Não	Armazena a descrição do fornecedor
address_id	INTEGER	Sim	Chave Estrangeira
user_id	INTEGER	Sim	Chave Estrangeira

### 3.9.4 Produtos

Conforme é apresentado a Tabela 3.4 é a principal tabela do sistema, uma vez que o sistema foi desenvolvido com o objetivo de ser um WMS. Nessa tabela temos as seguintes informações nome do produto, preço, quantidade, unidade de medida, descrição do produto, imagem, código de barra, código eletrônico do produto, o identificador único dentro do sistema e, por fim a escolha de uma categoria que é a chave estrangeira vinda da Tabela 3.5, de caráter obrigatório na fase de criação de um novo produto e o usuário vindo da Tabela 3.9.

Tabela 3.4: Tabela Products. Fonte: Própria dos autores

<b>Campo</b>	<b>Tipo</b>	<b>Obrigatório</b>	<b>Descrição</b>
product_id	INTEGER	Sim	Chave Primária
name	STRING	Sim	Armazena o nome do produto
price	DECIMAL	Sim	Armazena o valor do produto
quantity	INTEGER	Sim	Armazena a quantidade do produto
unity	STRING	Sim	Armazena a unidade métrica do produto
description	TEXT	Sim	Armazena a descrição do produto
productCode	INTEGER	Sim	Armazena o ID do produto
gtin	STRING	Sim	Armazena o Global Trade Item Number do produto
rfid	INTEGER	Sim	Armazena o número RFID do produto
image	BLOB	Não	Armazena a imagem
categorie_id	INTEGER	Sim	Chave Estrangeira
user_id	INTEGER	Sim	Chave Estrangeira

### 3.9.5 Categorias

Conforme é apresentado na Tabela 3.5 é responsável por armazenar um código gerado pelo sistema e o nome da categoria, que é um campo obrigatório a ser atribuído a um produto com intuito de facilitar em suas classificação e busca dentro do sistema.

Tabela 3.5: Tabela Categories. Fonte: Própria dos autores

<b>Campo</b>	<b>Tipo</b>	<b>Obrigatório</b>	<b>Descrição</b>
categorie_id	INTEGER	Sim	Chave Primária
nameCat	STRING	Sim	Armazena o nome da categoria
codCategorie	DECIMAL	Sim	Armazena o código da categoria
user_id	INTEGER	Sim	Chave Estrangeira

### 3.9.6 Compras

Conforme é apresentado a Tabela 3.6 salva o histórico de compras feitas em nosso sistema com os seguintes campos nome do produto, preço, total de compra, quantidade, unidade métrica, descrição do produto, o nome do fornecedor vindo diretamente da Tabela 3.2 e, em caso de custos adicionais, os campos de IPI, ICMS, frete, custo adicional e descrição do custo vindo da Tabela 3.8 e o usuário vindo da Tabela 3.9.

Tabela 3.6: Tabela Buys. Fonte: Própria dos autores

<b>Campo</b>	<b>Tipo</b>	<b>Obrigatório</b>	<b>Descrição</b>
buy_id	INTEGER	Sim	Chave Primária
price	DECIMAL	Sim	Armazena o valor da compra
totbuy	DECIMAL	Sim	Armazena o valor total
quantity	INTEGER	Sim	Armazena a quantidade de produtos comprados
nameProduct	STRING	Sim	Armazena o nome do produto
measurement	STRING	Sim	Armazena a unidade métrica do produto
description	TEXT	Sim	Armazena a descrição do produto
provider_id	INTEGER	Sim	Chave Estrangeira
tax_id	INTEGER	Sim	Chave Estrangeira
user_id	INTEGER	Sim	Chave Estrangeira

### 3.9.7 Vendas

Conforme é apresentado a Tabela 3.7 salva o histórico de vendas feitas em nosso sistema. O campo "nome do produto" é oriundo da Tabela 3.4. Os campos preço, total de venda, quantidade, unidade métrica, descrição do produto, assim como o nome do cliente são oriundos da Tabela 3.3. Dessa forma, fica garantido que uma venda só possa ser feita o produto exista no banco. Assim sendo, o cliente deve estar obrigatoriamente cadastrado no sistema, permitindo assim um melhor controle e organização interna. Caso hajam custos adicionais, os campos de IPI, ICMS, frete, custo adicional e descrição do custo vindo da Tabela 3.8 e o usuário vindo da Tabela 3.9.

Tabela 3.7: Tabela Sells. Fonte: Própria dos autores

<b>Campo</b>	<b>Tipo</b>	<b>Obrigatório</b>	<b>Descrição</b>
sell_id	INTEGER	Sim	Chave Primária
price	DECIMAL	Sim	Armazena o valor da venda
totsell	DECIMAL	Sim	Armazena o valor total
quantity	INTEGER	Sim	Armazena a quantidade de produtos vendido
measurement	STRING	Sim	Armazena a unidade métrica do produto
description	TEXT	Sim	Armazena a descrição do produto
client_id	INTEGER	Sim	Chave Estrangeira
product_id	INTEGER	Sim	Chave Estrangeira
tax_id	INTEGER	Sim	Chave Estrangeira
user_id	INTEGER	Sim	Chave Estrangeira

### 3.9.8 Taxas

Conforme é apresentado a Tabela 3.8 é uma tabela intermediária que faz a soma dos impostos e taxas adicionais atribuídos a uma venda ou compra. Essa é composta pelos campos: impostos de IPI e ICMS, assim como outros campos referentes ao frete, custo adicional e descrição de custo adicional.

Tabela 3.8: Tabela Taxes. Fonte: Própria dos autores

<b>Campo</b>	<b>Tipo</b>	<b>Obrigatório</b>	<b>Descrição</b>
tax_id	INTEGER	Sim	Chave Primária
shipping	DECIMAL	Sim	Armazena o valor da taxa so frete
costAdd	DECIMAL	Sim	Armazena o valor da taxa do custo adicional
ipi	DECIMAL	Sim	Armazena o valor do imposto de IPI
icms	DECIMAL	Sim	Armazena o valor do imposto de ICMS
description	TEXT	Sim	Armazena a descrição do custo adicional

### 3.9.9 Usuários

Conforme é apresentado a Tabela 3.9 tem como função salvar informações do usuário, quando realizado o cadastro. Por razões de segurança, o sistema permite apenas o acesso de usuários autenticados. Dessa forma, as informações solicitadas para que o cadastro dentro da plataforma seja realizado são nome, sobrenome, nome da empresa, CNPJ e senha.

Tabela 3.9: Tabela Users. Fonte: Própria dos autores

<b>Campo</b>	<b>Tipo</b>	<b>Obrigatório</b>	<b>Descrição</b>
user_id	INTEGER	Sim	Chave Primária
email	STRING	Sim	Armazena o email do usuário
company	STRING	Sim	Armazena o nome da empresa a ser cadastrada
name	STRING	Sim	Armazena o nome do usuário
lastname	STRING	Sim	Armazena a sobre-nome do usuário
password	STRING	Sim	Armazena a senha do usuário

### 3.9.10 Endereços

Conforme é apresentado a Tabela 3.10 é uma tabela de apoio, na qual as chaves estrangeiras da Tabela 3.3 e da Tabela 3.2 exercem a função salvar o endereço dos mesmos nos campos como rua, bairro, cidade, estado, código postal e complemento, caso queira colocar algo.

Tabela 3.10: Tabela Addresses. Fonte: Própria dos autores

<b>Campo</b>	<b>Tipo</b>	<b>Obrigatório</b>	<b>Descrição</b>
address_id	INTEGER	Sim	Chave Primária
city	STRING	Sim	Armazena o nome da cidade do cliente ou do fornecedor
street	STRING	Sim	Armazena o nome da rua onde o cliente ou fornecedor se encontra
neighborhood	STRING	Sim	Armazena o nome do bairro onde o cliente ou fornecedor se encontra
number	STRING	Sim	Armazena o número do estabelecimento onde fica localizado o cliente ou fornecedor.
complement	STRING	Não	Armazena o complemento de endereço do cliente ou do fornecedor.
state	STRING	Sim	Armazena em qual estado do país fica localizada a empresa do cliente ou do fornecedor.
zipcode	INTEGER	Não	Armazena o código postal do endereço do cliente ou do fornecedor .



## 3.10 Síntese do Capítulo

O **Capítulo 3** trata sobre o processo de desenvolvimento do sistema. Disponibilizamos um passo a passo explicando como preparar o ambiente do sistema operacional, a fim de que seja possível replicar a metodologia que utilizamos para o desenvolvimento do sistema. Apontamos os obstáculos encontrados durante o processo, assim como as razões que nos levaram a optar pelo desenvolvimento de dois softwares, um aplicativo Android e uma aplicação web. Esses dois trabalham em conjunto e, dessa forma, permitem tanto a leitura das tags de radiofrequência como a correta gestão do estoque. Explicamos também a modelagem dos bancos de dados, tanto o da API como o do sistema web. Também explicamos os casos de uso, além de detalharmos o banco de dados, o que seus campos representam e suas funções. No capítulo 4 será apresentado o aplicativo em funcionamento, as telas e suas descrições.

# Capítulo 4

## Apresentação do Sistema

Este Capítulo apresenta o sistema gestor de estoque desenvolvido neste trabalho, com o intuito de gerir de forma inteligente e automatizada um estoque ou inventário de um determinado negócio. Esse software facilitará no gerenciamento da atividade de gestão do estoque, permitindo realizar o registro das compras, das vendas, categorização e cadastro de produtos, cadastro de clientes e fornecedores, visualização de relatórios e emissão de notas. Este software foi implementado para operar não somente na plataforma Web, como também em dispositivos móveis.

### 4.1 Periféricos do Sistema de Captura de Tags

#### 4.1.1 Leitor RFID Bluetooth

Conforme apresentado na Figura 4.1, temos o leitor RFID bluetooth portátil UHF que opera nas faixas de frequência 902-928MHz ou 865-868MHz, dentro do protocolo ISO18000-6C do modelo CF-H301 da fabricante chinesa Chafon, com a capacidade de ler 50 tags por segundo. No cenário do funcionamento do sistema o leitor desempenha um papel muito importante, sendo o responsável por conectá-lo ao dispositivo no qual irá enviar os dados das tags lidas e decodificá-lo.



Figura 4.1: Leitor RFID Portátil. Fonte: Própria dos autores

#### 4.1.2 Tags Passivas

Como podemos observar na Figura 4.2, temos os modelos de tags usadas em nosso estudo. São tags do tipo UHF passiva e operam na frequência 860 - 960 MHz, operando dentro do protocolo ISO 18000-6C, cada tag tem um numero identificador único e através desse número podemos identificar e atrelar esse número a novos dados dentro de um sistema e, aliado ao leitor visto na Figura 4.1, temos a possibilidade de ler várias tags dessa ao mesmo tempo, otimizando o tempo de quem tem a tarefa de cadastrar ou ler de forma manual uma a uma.

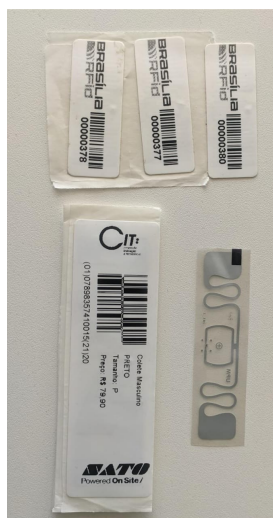


Figura 4.2: Tags de identificação RFID. Fonte: Própria dos autores

## 4.2 Telas do Aplicativo de Captura de Tags

### 4.2.1 Conectando no Leitor Via Bluetooth

Conforme apresentado na Figura 4.3 da tela inicial do aplicativo sendo executado no smartphone, ao clicar em "Buscar dispositivo" temos listadas todas as conexões bluetooth próximas disponíveis para serem pareadas.

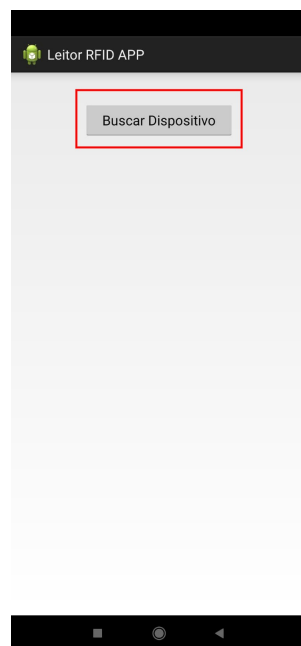


Figura 4.3: Tela de buscar dispositivos. Fonte: Própria dos autores

### 4.2.2 Dispositivos Pareados

Conforme apresentado na Figura 4.4, é apresentado a tela dos dispositivos disponíveis a serem pareados e o nome do dispositivo do leitor é o **UBT19471077B** com o endereço **00:15:90:91:CA:49** a ser conectado ao clicar.



Figura 4.4: Tela de parear dispositivo APP. Fonte: Própria dos autores

### 4.2.3 Escaneando Tags RFID

Conforme apresentado na Figura 4.5, tela na qual o usuário prepara para escanear, tem como resultado a ser exibido o EPC de todas as tags lidas pelo leitor e quantidade de vezes lidas por segundo antes de enviar para a API.

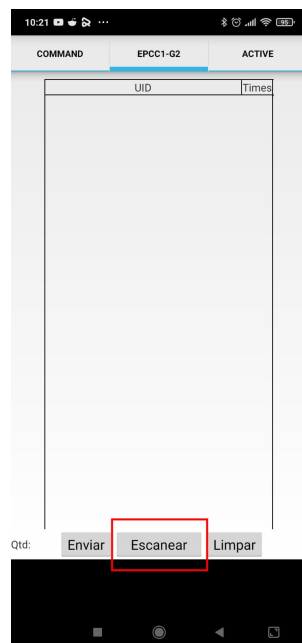


Figura 4.5: Tela de de escanear tags APP. Fonte: Própria dos autores

Como podemos observar na Figura 4.6, temos como resultado o EPC de todas as tags lidas pelo leitor diferente da Figura 4.5 no qual nada era exibido, depois que as tags forem lidas o resultado a ser exibido em tela é a quantidade de vezes lidas por segundo e quantidade total de tags lidas e o próprio EPC como foi dito anteriormente.

COMMAND		EPCC1-G2	ACTIVE
UID			Times
00000000000000000000000001511			8
00000000000000000000000000378			12
E2003411B802011196312232			22
0000000000000000000000000377			19
300832BD2DD9014000000000			15
0000000000000000000000000392			2
E28011606000021193A9020F			12
E28011606000021193A9020E			21
E28011606000021193A9020D			24
202020202042413030333038			4
202020202042413030333037			3
0000000000000000000000000380			12
202020202042413030333039			21
E28011606000021193A8FB01			25

Qtd:20

Enviar

\*Parar

Limpar

58

#### 4.2.5 Enviando Dados para a API

Ao clicar no botão para enviar o EPC, esse valor passa a ser reconhecido pelo aplicativo como um vetor de caracteres do tipo string, entrando em uma estrutura de dados do tipo First In First Out (FIFO), na qual temos a garantia que o EPC de todas as tags que foram lidas também foram enviadas e salvas diretamente no banco de dados da API, estabelecendo um limite de requisições por segundo na hora de seu envio para não prejudicar o desempenho e performance do web service, conforme apresentado na Figura 4.7.

COMMOND		EPPC1-G2	ACTIVE
UID			Times
00000000000000000000000001511	8		
00000000000000000000000000378	12		
E2003411B802011196312232	22		
0000000000000000000000000377	19		
300833B2DD9014000000000	15		
0000000000000000000000000392	2		
E28011606000021193A9020F	12		
E28011606000021193A9020E	21		
E28011606000021193A9020D	24		
202020202042413030333038	4		
202020202042413030333037	3		
0000000000000000000000000380	12		
202020202042413030333039	21		
E28011606000021193A8FB01	25		

Qtd:20

Enviar

\*Parar

Limpar

Figura 4.7: Enviando para API. Fonte: Própria dos autores

### 4.2.6 Exibindo Dados Salvos Diretamente na API

Conforme apresentado na Figura 4.8, temos o retorno do JSON de todas as tags lidas com as respectivas data e horas exatas que foram salvas diretamente na API.

```

1  {
2    "id": 1000,
3    "fullName": "701914110001119102101",
4    "email": "701914110001119102101@hertieai.com",
5    "password": "701914110001119102101",
6    "createdAt": "2022-03-04T17:10:43.301Z",
7    "updatedAt": "2022-03-04T17:10:43.301Z",
8  },
9  {
10   "id": 1001,
11   "fullName": "701914110001119102102",
12   "email": "701914110001119102102@hertieai.com",
13   "password": "701914110001119102102",
14   "createdAt": "2022-03-04T17:10:43.303Z",
15   "updatedAt": "2022-03-04T17:10:43.303Z",
16 },
17 {
18   "id": 1002,
19   "fullName": "701914110001119102103",
20   "email": "701914110001119102103@hertieai.com",
21   "password": "701914110001119102103",
22   "createdAt": "2022-03-04T17:10:43.305Z",
23   "updatedAt": "2022-03-04T17:10:43.305Z",
24 },
25 {
26   "id": 1003,
27   "fullName": "701914110001119102104",
28   "email": "701914110001119102104@hertieai.com",
29   "password": "701914110001119102104",
30   "createdAt": "2022-03-04T17:10:43.307Z",
31   "updatedAt": "2022-03-04T17:10:43.307Z",
32 },
33 {
34   "id": 1004,
35   "fullName": "701914110001119102105",
36   "email": "701914110001119102105@hertieai.com",
37   "password": "701914110001119102105",
38   "createdAt": "2022-03-04T17:10:43.309Z",
39   "updatedAt": "2022-03-04T17:10:43.309Z",
40 },
41 {
42   "id": 1005,
43   "fullName": "701914110001119102106",
44   "email": "701914110001119102106@hertieai.com",
45   "password": "701914110001119102106",
46   "createdAt": "2022-03-04T17:10:43.311Z",
47   "updatedAt": "2022-03-04T17:10:43.311Z",
48 },
49 {
50   "id": 1006,
51   "fullName": "701914110001119102107",
52   "email": "701914110001119102107@hertieai.com",
53   "password": "701914110001119102107",
54   "createdAt": "2022-03-04T17:10:43.313Z",
55   "updatedAt": "2022-03-04T17:10:43.313Z",
56 },
57 {
58   "id": 1007,
59   "fullName": "701914110001119102108",
60   "email": "701914110001119102108@hertieai.com",
61   "password": "701914110001119102108",
62   "createdAt": "2022-03-04T17:10:43.315Z",
63   "updatedAt": "2022-03-04T17:10:43.315Z",
64 },
65 {
66   "id": 1008,
67   "fullName": "701914110001119102109",
68   "email": "701914110001119102109@hertieai.com",
69   "password": "701914110001119102109",
70   "createdAt": "2022-03-04T17:10:43.317Z",
71   "updatedAt": "2022-03-04T17:10:43.317Z",
72 },
73 {
74   "id": 1009,
75   "fullName": "701914110001119102110",
76   "email": "701914110001119102110@hertieai.com",
77   "password": "701914110001119102110",
78   "createdAt": "2022-03-04T17:10:43.319Z",
79   "updatedAt": "2022-03-04T17:10:43.319Z",
80 },
81 {
82   "id": 1010,
83   "fullName": "701914110001119102111",
84   "email": "701914110001119102111@hertieai.com",
85   "password": "701914110001119102111",
86   "createdAt": "2022-03-04T17:10:43.321Z",
87   "updatedAt": "2022-03-04T17:10:43.321Z",
88 },
89 {
90   "id": 1011,
91   "fullName": "701914110001119102112",
92   "email": "701914110001119102112@hertieai.com",
93   "password": "701914110001119102112",
94   "createdAt": "2022-03-04T17:10:43.323Z",
95   "updatedAt": "2022-03-04T17:10:43.323Z",
96 },
97 {
98   "id": 1012,
99   "fullName": "701914110001119102113",
100  "email": "701914110001119102113@hertieai.com",
101  "password": "701914110001119102113",
102  "createdAt": "2022-03-04T17:10:43.325Z",
103  "updatedAt": "2022-03-04T17:10:43.325Z",
104 },
105 {
106   "id": 1013,
107   "fullName": "701914110001119102114",
108   "email": "701914110001119102114@hertieai.com",
109   "password": "701914110001119102114",
110   "createdAt": "2022-03-04T17:10:43.327Z",
111   "updatedAt": "2022-03-04T17:10:43.327Z",
112 },
113 {
114   "id": 1014,
115   "fullName": "701914110001119102115",
116   "email": "701914110001119102115@hertieai.com",
117   "password": "701914110001119102115",
118   "createdAt": "2022-03-04T17:10:43.329Z",
119   "updatedAt": "2022-03-04T17:10:43.329Z",
120 },
121 {
122   "id": 1015,
123   "fullName": "701914110001119102116",
124   "email": "701914110001119102116@hertieai.com",
125   "password": "701914110001119102116",
126   "createdAt": "2022-03-04T17:10:43.331Z",
127   "updatedAt": "2022-03-04T17:10:43.331Z",
128 },
129 {
130   "id": 1016,
131   "fullName": "701914110001119102117",
132   "email": "701914110001119102117@hertieai.com",
133   "password": "701914110001119102117",
134   "createdAt": "2022-03-04T17:10:43.333Z",
135   "updatedAt": "2022-03-04T17:10:43.333Z",
136 },
137 {
138   "id": 1017,
139   "fullName": "701914110001119102118",
140   "email": "701914110001119102118@hertieai.com",
141   "password": "701914110001119102118",
142   "createdAt": "2022-03-04T17:10:43.335Z",
143   "updatedAt": "2022-03-04T17:10:43.335Z",
144 },
145 {
146   "id": 1018,
147   "fullName": "701914110001119102119",
148   "email": "701914110001119102119@hertieai.com",
149   "password": "701914110001119102119",
150   "createdAt": "2022-03-04T17:10:43.337Z",
151   "updatedAt": "2022-03-04T17:10:43.337Z",
152 },
153 {
154   "id": 1019,
155   "fullName": "701914110001119102120",
156   "email": "701914110001119102120@hertieai.com",
157   "password": "701914110001119102120",
158   "createdAt": "2022-03-04T17:10:43.339Z",
159   "updatedAt": "2022-03-04T17:10:43.339Z",
160 },
161 {
162   "id": 1020,
163   "fullName": "701914110001119102121",
164   "email": "701914110001119102121@hertieai.com",
165   "password": "701914110001119102121",
166   "createdAt": "2022-03-04T17:10:43.341Z",
167   "updatedAt": "2022-03-04T17:10:43.341Z",
168 },
169 {
170   "id": 1021,
171   "fullName": "701914110001119102122",
172   "email": "701914110001119102122@hertieai.com",
173   "password": "701914110001119102122",
174   "createdAt": "2022-03-04T17:10:43.343Z",
175   "updatedAt": "2022-03-04T17:10:43.343Z",
1
```

Figura 4.8: Tela de tags salvas na API. Fonte: Própria dos autores



## 4.3 Telas do Sistema Web

### 4.3.1 Tela de login

Ao acessar o sistema pela primeira vez, o usuário encontrará a tela de login conforme apresentado na Figura 4.9 para que possa acessar o sistema. Caso não possua uma conta cadastrada, há a opção de criar um usuário e senha clicando na opção "Cadastrar". Para que não haja a necessidade de estar sempre digitando as mesmas informações de acesso, disponibilizamos a opção de "Lembrar senha". Ao ativar essa opção, os campos ficam previamente preenchidos, sendo necessário apenas clicar em "Entrar".

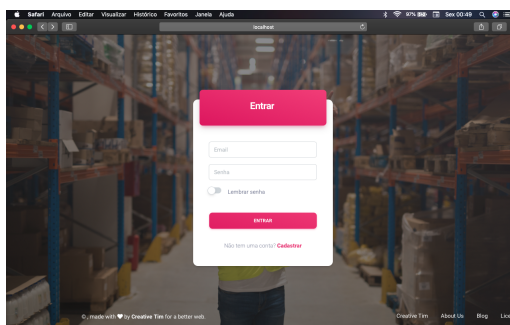


Figura 4.9: Tela de login do sistema. Fonte: Própria dos autores

### 4.3.2 Tela de Cadastro de Usuário

Ao clicar em "Cadastrar" o usuário, é direcionado para a página onde realizará seu registro conforme apresentado na Figura 4.10. Para isso será necessário preencher os campos "Nome", "Sobrenome", "Nome da empresa", "CNPJ", "Email", "Senha" e "Confirme sua senha". Além de preencher obrigatoriamente esses campos, é necessário marcar o "checkbox" de "Eu aceito os Termos e Condições", para que o cadastro possa ser concluído com sucesso.

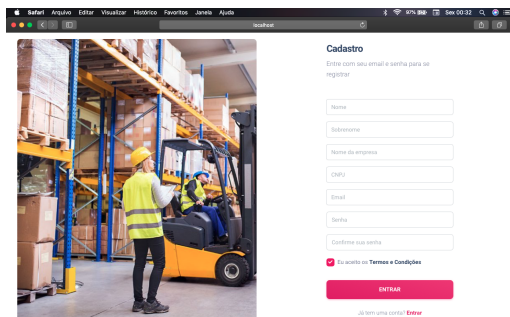


Figura 4.10: Tela de login do sistema. Fonte: Própria dos autores

### 4.3.3 Dashboard

Após realizar o login no sistema, o usuário será direcionado para o painel de Dashboard, conforme apresentado na Figura 4.11, onde há painéis e gráficos sintetizando as principais informações para o gerenciamento do sistema. Nessa tela de exibição temos o demonstrativo de:

1. **Vendas:** a quantidade total de vendas realizadas.
2. **Compras:** a quantidade total de compras realizadas.
3. **Carteira:** a subtração de Vendas por Compras.
4. **Produtos:** a quantidade total dos produtos cadastrados no banco de dados.

Disponibilizamos também um menu lateral para facilitar a navegação entre as categorias do sistema.

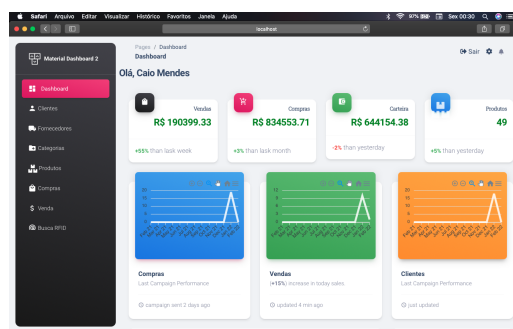


Figura 4.11: Tela de Dashboard do sistema. Fonte: Própria dos autores

### 4.3.4 Tela de Index Clientes

Na view index, conforme apresentado na Figura 4.12 temos as ações que podem ser executadas para a administração do sistema, quais sejam: exibir separadamente, editar, gerar um arquivo PDF com as informações de cada cliente separadamente, e inclusive, deletar o cliente da base de dados.

A index tem por objetivo exibir as principais informações de maneira rápida e facilitada, melhorando assim a visualização, controle e gestão do sistema por parte do usuário.

Como os clientes são um dos pilares fundamentais de qualquer empreendimento, e a satisfação desse para com um produto, serviço, bem, ou até mesmo atendimento, devem ser asseguradas para que um negócio venha a prevalecer.

Por essa razão, decidimos dedicar um módulo do sistema para ser realizada a gestão dos dados do cliente, com o intuito de que isso resulte em um aumento significativo nas

vendas, assim como no faturamento de uma organização que optar por utilizar o sistema gestor de estoque.

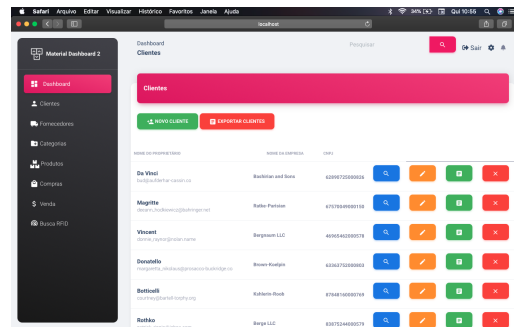


Figura 4.12: Tela index clientes. Fonte: Própria dos autores

### 4.3.5 Tela de Cadastro Cliente

Conforme apresentado na Figura 4.13, temos a opção de cadastrar os dados preenchidos no formulário, para estar criando o cliente com as informações inseridas anteriormente no banco de dados do sistema.

Disponibilizamos, no campo CEP, o botão de "Buscar", o qual faz a requisição do CEP diretamente na API dos Correios, completando de forma automática os campos relacionados ao endereço melhorando a experiência do usuário na usabilidade do software.

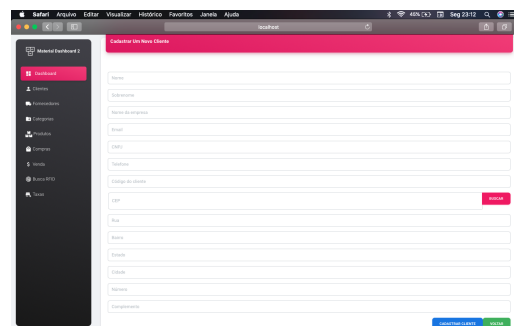


Figura 4.13: Tela de criar cliente. Fonte: Própria dos autores

### 4.3.6 Tela de Editar Categoria

Na tela apresentada na Figura 4.14 temos a opção de editar os dados já preenchidos anteriormente no formulário de cadastro, com o objetivo de fazer alguma atualização no cadastro, caso precise.

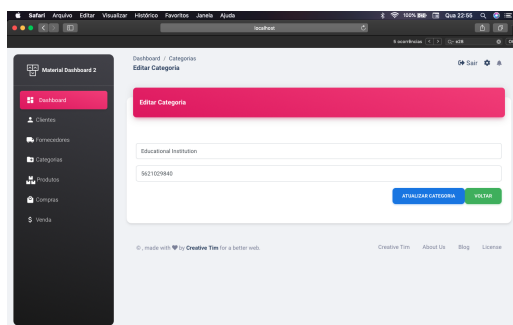


Figura 4.14: Tela de editar categoria. Fonte: Própria dos autores

### 4.3.7 Tela de Excluir Cliente

Como medida para evitar que o usuário venha a apagar algum cliente acidentalmente, colocamos a mensagem de "Você tem certeza?", na qual o usuário necessita selecionar a opção "Ok" para que o usuário seja realmente excluído do sistema, conforme apresentado na Figura 4.15. Caso contrário, basta selecionar "Cancelar" e a ação de excluir o usuário não será realizada.

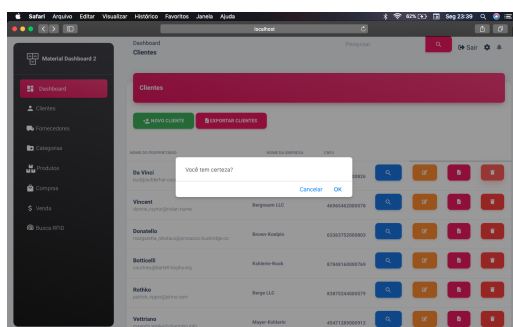


Figura 4.15: Tela de excluir cliente. Fonte: Própria dos autores

#### 4.3.8 Tela de Exibição de Produto

Na tela de exibição é apresentado um resumo sobre os produtos cadastrados. Há uma breve descrição do objeto em questão, a unidade de medida do produto, a quantidade de unidades disponíveis, o preço, além do código de barras e código RFID do produto, conforme apresentado na Figura 4.16.

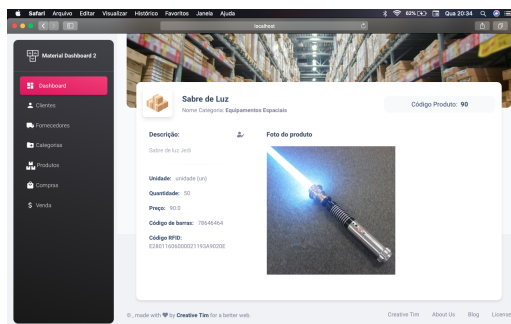


Figura 4.16: Tela que exibe os produtos. Fonte: Própria dos autores

#### 4.3.9 Tela de Importação de Produtos

Caso o usuário assim desejar, disponibilizamos uma view onde é possível fazer o download de uma tabela com extensão .csv, formatada no padrão correto aceito pelo sistema para ser importado diretamente, tornando assim viável o upload de vários produtos ao mesmo tempo e de uma só vez, deixando esse processo bem mais rápido, automatizado e eficiente, conforme apresentado na Figura 4.17.

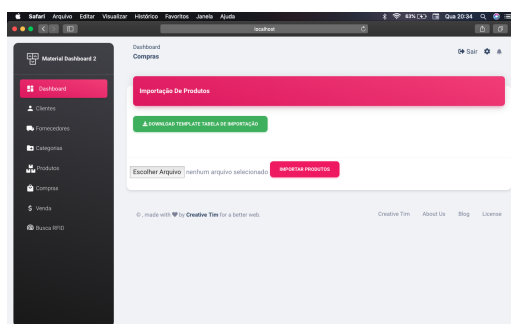


Figura 4.17: Importação de produtos através de um arquivo CSV. Fonte: Própria dos autores

### 4.3.10 Tela de busca RFID

Caso haja dúvida sobre como utilizar o Aplicativo do leitor RFID, disponibilizamos também uma página do sistema onde o usuário pode consultar um passo a passo de como utilizar o dispositivo, conforme apresentado na Figura 4.18.

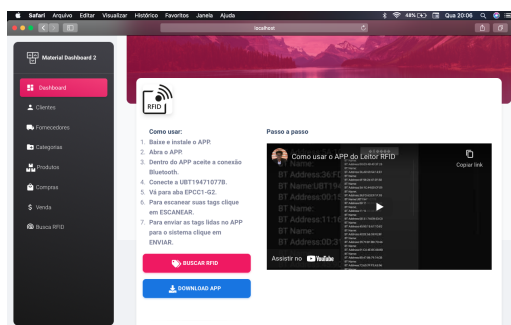


Figura 4.18: Tela de busca na API. Fonte: Própria dos autores

### 4.3.11 Tela de Verificação RFID

Na tela de verificação conforme apresentado na Figura 4.19, é possível o usuário visualizar as tags RFID que foram mandadas do APP para a API, que por consequência o sistema faz uma integração e intercambia os dados. As tags com o botão verde "cadastrar" dão a opção ao usuário de cadastrar um produto diretamente, atribuindo aquela tag EPC ao novo produto. Nessa tela, todas as tags são buscadas diretamente da API e, através do uso de funções e métodos de comparações da linguagem Ruby, conseguimos validar que esse código EPC não existe e não está associado a nenhum dos produtos cadastrados anteriormente no banco de dados, retornando um tipo boolean true para caso existir ou false para caso não existir. Aqui também ocorre outra validação que impede que a mesma tag lida apareça mais de uma vez. Dessa forma, evitamos que possa ocorrer do mesmo EPC ser cadastrado duas vezes. No botão azul de "ver o produto", o produto já existe no banco de dados e através do ID puxado pelo campo rfidCode associado com a tabela de products, é possível ir diretamente para a view do produto cadastrado em questão, escolhido pelo usuário.

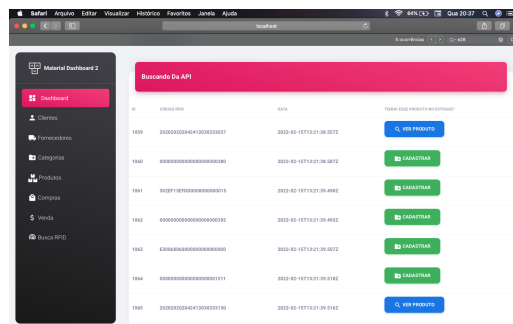


Figura 4.19: Tela que exibe se a tag já existe no sistema. Fonte: Própria dos autores

## 4.4 Síntese do Capítulo

O **Capítulo 4** apresentou o sistema proposto, suas principais funcionalidades e detalhou a integração entre o aplicativo Android e o sistema web. Com o intuito de facilitar o entendimento, pensamos em estruturar esse capítulo em duas partes. Na primeira explicamos sobre o funcionamento e interações que são realizadas no dispositivo Android. Tratamos de tópicos como: a conexão da leitora com o smartphone, a leitura das tags de radiofrequência, o envio dos dados e consulta desses na API hospedada no Heroku. Na segunda parte abordamos as telas do sistema web, as principais funções de cada módulo, informações disponíveis e opções de interação que o usuário pode realizar. No capítulo 5 será apresentada a conclusão.

# Capítulo 5

## Resultados da Aplicação do Sistema

### 5.1 Resultados

Com o intuito de validar o funcionamento do sistema gestor de estoque em um contexto real, o estudo de caso foi realizado em um empreendimento de pequeno porte, o qual será referenciado pelo nome fictício de ASDF. Esta referida empresa atua no departamento de moda, fazendo a venda de camisetas ao público geral. A grande dificuldade de quem está começando é regular exatamente o seu estoque, sem deixar uma grande quantidade de ativos parados, com isso perdendo dinheiro, pois no ramo da logística o estoque parado gera custo. Sendo assim, com uso do sistema teremos um controle de forma digital e facilitada para o usuário gerir suas camisas.

O foco desse estudo é fazer a gestão de um estoque de uma forma mais rápida e arrojada. Para isso, desenvolvemos um sistema onde a gestão dos ativos é realizada por meio da leitura de tags de radiofrequência, de forma que o sistema não dependa somente do RFID. Porém, que esse seja seu principal diferencial, pois no mercado existem diversos provedores de soluções com sistemas de gestão, contudo atrelado a tecnologia de radiofrequência ainda existem poucos. Conforme apresentado na Figura 5.1, para simular um cenário de uso real usamos um pequeno estoque com 15 camisetas sendo que, cada uma tem uma tag que inicialmente não estavam cadastradas em nosso banco de dados. Ao aplicarmos o sistema nesse contexto, observamos que os seguintes resultados foram obtidos:

- Melhoria no controle dos ativos
- Controle financeiro de entrada e saída
- Base centralizada de clientes, fornecedores e produtos
- Confiança por parte dos usuários do sistema



- Controle do estoque em tempo real
- Central de alerta para auxiliar o usuário
- Serviço disponível em nuvem
- Segurança na usabilidade



Figura 5.1: Camisetas do estoque. Fonte: Própria dos autores

## 5.2 Uso do Sistema em um Contexto Real

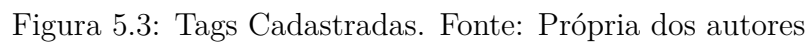
Para a correta utilização do sistema gestor de estoque, o primeiro passo é realizar a sincronização do leitor portátil com um aparelho celular através de uma conexão bluetooth dentro do app "Leitor RFID", o qual foi desenvolvido através desse estudo com o objetivo de automatizar o controle de estoques. O leitor usado é portátil e do modelo de banda UHF que opera nas faixas de frequência 902-928MHz ou 865-868MHz, dentro do protocolo ISO18000-6C. Para iniciar o processo de leitura basta aproximar a leitora perto das etiquetas. Esse dispositivo tem a capacidade de leitura em uma distância máxima de até 1 metro e consegue ler até 50 tags por segundo.

Conforme apresentado na Figura 5.2, temos uma tag UHF passiva atrelada a cada peça do estoque para garantir e facilitar a identificação e controle dentro do sistema de estoque tanto dessa peça quanto de outras.



Figura 5.2: Tag RFID Camisetas. Fonte: Própria dos autores

Conforme apresentado na Figura 5.3, temos uma tela de verificação que interage diretamente com a API buscando os dados e comparando com o que já existe armazenado no banco de dados do sistema, com a finalidade de fazer algumas validações de existência e equiparações de igualdade e para exibir de modo mais amigável ao usuário, retornando na tela dois botões "Cadastrar" caso a tag do EPC não exista e "Ver produto" para caso se aquela tag já esteja cadastrada em algum produto.



69

## 5.3 Comparativo dos Resultados

Conforme mencionado anteriormente no Capítulo 2 e conforme é apresentado na (Tabela 2.5), o RFID é muito promissor e está em crescente ascensão. Porém, ainda não existem muitos trabalhos acadêmicos que se propuseram a utilizar essa tecnologia de RFID voltada para gestão de ativos. Fazendo um comparativo com os sistemas que foram desenvolvidos até o presente momento e listados na (Tabela 2.5), o que foi desenvolvido nesse trabalho apresenta os seguintes diferenciais:

- Controla todos os custos da empresa;
- Ganho de produtividade;
- Confere maior segurança;
- Oferece escalabilidade;
- Gestão de ativos usando RFID;
- Uso de API para armazenamento de dados;
- Emissão de notas de compra e vendas;
- Gestão eficiente do estoque; e
- Geração de relatórios.

## 5.4 Limitações e Ameaças para Validar o Estudo

Durante o estudo foi observado que ao utilizar o sistema gestor de estoque, os resultados apresentados foram positivos para a organização de um negócio, especialmente pelo controle total de informações precisas, alertas e em tempo real de quando o estoque tiver em baixa além disso o sistema faz a gestão do estoque, clientes, fornecedores, compras e vendas. Dentro do nosso cenário de teste e simulações foi possível afirmar que o sistema é funcional, se tratando em pequenos estabelecimentos. Para que seja possível afirmar o mesmo para negócios de grande porte, mais estudos necessitam ser realizados.

- **Alto custo dos equipamentos:** Um dos fatores que é limitador é o alto custo dos equipamentos necessários para a implementação do sistema como as antenas, os módulos RFID e as tags. O valor elevado é devido aos poucos fabricantes de hardware para esse tipo de finalidade, causando um efeito de mercado de oferta e demanda. Para avançar em nosso estudo e driblar essa dificuldade, fomos buscar um hardware de prototipação chinês de menor custo e menor alcance. Porém, dentro do

que precisamos, nós atendeu perfeitamente. Outra dificuldade foi o teste das tags, pois como não tínhamos muitas, o nosso teste foi limitado a 32 tags.

- **Alto custo do software:** Como existe poucos fabricantes desse tipo de hardware, o mercado de software também acompanha essa variante. Geralmente o hardware já vem com o seu software proprietário, limitando ainda mais o desenvolvimento de novas soluções dependendo bastante do fabricante. Nossa solução foi a criação de um sistema e fazer a distribuição open source através de um repositório no Github e distribuir nosso sistema gratuitamente, permitindo até a colaboração de outros desenvolvedores com novas features e novas ideias de implementação.
- **Mão de obra especializada:** É um mercado que está em ascensão e com isso não existe muitos provedores de soluções, restringindo de forma considerável o número de profissionais capacitados para exercer as atividades providas pelo uso da tecnologia de RFID. Com isso, a ideia foi de facilitar o máximo possível o uso do APP e sistema web com interfaces bonitas e intuitivas, afim de, com poucos minutos de uso, o usuário já se sinta familiarizado com o sistema sem grandes dúvidas.
- **Entender a real necessidade do uso da tecnologia RFID em seu negócio:** A implantação de uma nova tecnologia dentro de um negócio ou empreendimento tem que ser algo bem pensado e planejado, envolvendo vários estudos de viabilidade de custo, tanto de aparelhagem, softwares e treinamento de capacitação para os funcionários que forem usar o sistema.
- **Ter processos bem definidos e seguir o sistema risca:** O uso de um sistema de gestão requer um certo nível de maturidade do conhecimento dos processos internos e tê-los bem definidos para um uso mais efetivo e assertivo, gerando resultados que impactem positivamente esses processos, afim de validar o emprego e uso do sistema de gestão. A solução a ser seguida foi um bom levantamento de requisito, conversa com profissionais que trabalham no ramo do varejo e logística e a leitura de vários outros artigos com a mesma temática e entender o processo de gestão de estoque foi fundamental para traçar um workflow para suprir as falhas de gestão de estoque dentro de uma empresa.
- **Saber como usar as informações:** Saber como usar as informações é ponto fundamental, pois não tem efetividade de recolher as informações e não saber como usá-las. O uso das informações tem que ser objetivo e claro, principalmente para poder justificar o seu uso e efetividade. Com isso foi criada uma tela de dashboard com as principais informações vistas de uma maneira muito mais facilitada e interativa com alguns gráficos, trazendo informações instantâneas.

- **Interoperabilidade de sistemas no cadastro das tags:** O compartilhamento das informações em uma base de dados única permite o uso das informações em diversos outros sistemas, principalmente dentro de sistema de gestão. E para garantir essa interoperabilidade entre os sistemas, a solução foi o desenvolvimento de uma API para transitar as informações das tags entre o sistema e o APP, sendo possível ser usado até em outros sistemas.

## 5.5 Síntese do Capítulo

O **Capítulo 5** teve como intuito apresentar os resultados que foram obtidos com o estudo de caso realizado, utilizando o sistema gestor de estoque em um pequeno empreendimento. Para isso, apresentamos os principais resultados obtidos, a forma como o sistema foi utilizado em um contexto real. Além disso, demos destaque para as limitações e dificuldades que encontramos durante a realização do estudo.

# Capítulo 6

## Conclusão

Com a realização desse trabalho, chegamos à conclusão que a tecnologia de identificação por radiofrequência é muito promissora. Porém, os altos custos impendem que essa seja utilizada em larga escala. Com os testes realizados, foi possível observar que, ao utilizar o sistema desenvolvido, a gestão do estoque foi feita de uma maneira muito mais prática. Dessa forma, os recursos empregados para a atividade de gerir um estoque podem ser realocados para outro setor, permitindo assim um melhor aproveitamento da mão de obra.

### 6.1 Melhorias para trabalhos futuros

Durante o processo de pesquisa e desenvolvimento desse projeto, notamos que ainda existe margem para futuras melhorias e listamos algumas delas:

- Melhorar a classificação das categorias usando a Nomenclatura Comum do Mercosul, que é uma convenção de nomenclatura usada em todas as operações de comércio exterior dos países do Mercosul (Brasil, Argentina, Paraguai e Uruguai), pois a criação está muito ampla. Dessa forma, pode haver categorias ambíguas com a mesma finalidade (Ex. Roupas, Camisas, Vestimentas, etc).
- Usar aparelhagem de RFID mais completa e profissional. O leitor usado nesse estudo é pequeno, considerado fraco e utilizado mais para prototipação de soluções e estudo de viabilidade e em caso de um escalonamento do projeto, o leitor do modelo CF-H301 será inviável, tendo em vista que existem outros disponíveis no mercado, mais poderosos em questão de leitura de tags por segundo e até mesmo de longas distâncias.

- O aplicativo mobile do scanner ser integrado juntamente com sistema web, não sendo necessário que os dois sistemas atuem em conjunto para atender o propósito da gestão do estoque, centralizando os dois em somente um.
- Integrar o sistema com a API do Cadastro Nacional de Produtos, cujo objetivo é facilitar o controle sobre os itens comercializados, evitando a duplicidade de informações, proporcionando mais controle e segurança dos dados, garantindo dentro dos padrões globais usados na cadeia de suprimentos a rastreabilidade e identificação.
- Integrar com o sistema de geração de nota fiscal eletrônica (NF-e) do governo, junto a Sefaz (Secretária de Fazenda) órgão responsável responsável por fiscalizar, pagar, arrecadar e controlar os recursos públicos de cada estado.

# Referências

- [1] *Parson - serviços em tecnologia*. <http://parson.com.br/>. (Accessed on 05/20/2021). xii, 12, 25, 27
- [2] *Rfid - identificação rádio frequência*. [https://www.gta.ufrj.br/grad/12\\_1/rfid/links/funcionamento.html](https://www.gta.ufrj.br/grad/12_1/rfid/links/funcionamento.html). (Accessed on 02/09/2021). xiv, 2, 12
- [3] Santos, Alessandro, Leandro Avanço e Matheus Pereira: *TECNOLOGIAS EMERGENTES EM IOT: RSSF, RTLS, RFID CONCEITOS E APLICAÇÕES PARA CIDADES INTELIGENTES E INDÚSTRIA 4.0*. IPT, janeiro 2020, ISBN 978-65-5702-000-5. xiv, 10, 15, 16, 17
- [4] Ministry for Primary Industries: *Importância da indústria - Portal da Indústria - CNI*. <http://www.portaldaindustria.com.br/estatisticas/importancia-da-industria/>, 2021. Online; accessed 28 January 2021. 1
- [5] *Aplicação de rfid para rastreamento de carga e auxílio no processo logístico / anais da sociedade brasileira de automática*. [https://www.sba.org.br/open\\_journal\\_systems/index.php/sba/article/view/1414](https://www.sba.org.br/open_journal_systems/index.php/sba/article/view/1414). (Accessed on 01/31/2021). 1
- [6] Favaretto, Fábio: *Uma contribuição ao processo de gestão da produção pelo uso da coleta automática de dados de chão de fábrica*. São Carlos, 2001. 1
- [7] Bertaglia, Paulo Roberto: *Logística e gerenciamento da cadeia de abastecimento*. Saraiva Educação SA, 2017. 1
- [8] *Tecnologia rfid: o que é e como se aplica na área logística?* <https://www.bloglogistica.com.br/tecnologia/tecnologia-rfid-o-que-e-e-como-se-aplica-na-area-logistica/>. (Accessed on 02/09/2021). 2
- [9] Lin, LC: *An integrated framework for the development of radio frequency identification technology in the logistics and supply chain management*. Computers & Industrial Engineering, 57(3):832–842, 2009. 2, 8
- [10] Pedroso, Marcelo Caldeira, Ronaldo Zwicker e Cesar Alexandre de Souza: *Adoção de rfid no brasil: um estudo exploratório*. RAM. Revista de Administração Mackenzie, 10(1):12–36, 2009. 2
- [11] *Rfid: por que a tecnologia ainda não decolou no brasil? / computerworld*. <https://computerworld.com.br/negocios/idgnoticia-2007-07-20-1602719927/>. (Accessed on 03/22/2021). 2



- [12] *Conheça 10 empresas que usam rfid com sucesso / rfid brasil®*. <https://rfidbrasil.com/blog/conheca-10-empresas-que-usam-rfid-com-sucesso/>. (Accessed on 03/17/2021). 2
- [13] DeHoratius, Nicole e Ananth Raman: *Inventory record inaccuracy: An empirical analysis*. Management science, 54(4):627–641, 2008. 2
- [14] Freitas, Karine Fernandes, Priscilla Cristina Cabral Ribeiro e Nathália Soares Silva: *Adoção de rfid na gestão de estoques e na prevenção de perdas área temática: Logística*. simpósio de excelência em gestão e tecnologia, 2016. 3
- [15] Moretti, Eduardo de Araujo *et al.*: *Estudo das dificuldades observadas na implantação de sistemas rfid*. Unicamp, 2017. 3, 19, 20
- [16] Rekik, Yacine, Evren Sahin e Yves Dallery: *Inventory inaccuracy in retail stores due to theft: An analysis of the benefits of rfid*. International Journal of Production Economics, 118(1):189–198, 2009. 3
- [17] Wainer, Jacques *et al.*: *Métodos de pesquisa quantitativa e qualitativa para a ciência da computação*. Atualização em informática, 1(221-262):32–33, 2007. 5
- [18] Descalzo, Daniel, Flávia Andressa Jacinto da Silva e Graziela Candici Clemente: *Rfid: análise da viabilidade, vantagens e desvantagens da tecnologia e desenvolvimento de um sistema para demonstração e testes*. B.S. thesis, Universidade Tecnológica Federal do Paraná, 2011. 8
- [19] Santos Pinheiro, José Maurício dos: *Identificação por radiofrequência: Aplicações e vulnerabilidades da tecnologia rfid*. Cadernos UniFOA, 1(2):18–32, 2017. 8, 9
- [20] Belhadi, Z, L Fergani, B Poussot e J M Laheurte: *Low computational cost hybrid approach for near-field rfid tag localization*. Em *2019 International Conference on Advanced Electrical Engineering (ICAEE)*, páginas 1–5. IEEE, 2019. 10
- [21] Bolic, Miodrag, David Simplot-Ryl e Ivan Stojmenovic: *RFID systems: research trends and challenges*. John Wiley & Sons, 2010. 10
- [22] Santos, Jonathan Pinheiro dos: *Tecnologia de identificação por radiofrequência (rfid) em acervos bibliográficos: estudo de caso da biblioteca da câmara dos deputados*. BDM UNB, 2013. 10
- [23] Montalvão, Augusto César Pereira da Silva: *Caracterização numérica de antenas para aplicações rfid utilizando o método das ondas-wcip*. Tese de Mestrado, Universidade Federal do Rio Grande do Norte, 2010. 11
- [24] *Radiofrequência — português (brasil)*. <https://www.gov.br/anatel/pt-br/regulado/radiofrequencia#:~:text=Radiofrequ%C3%Aancia%20%C3%A9%20a%20faixa%20do,1997%2C%20%C3%A9%20administrado%20pela%20Anatel>. (Accessed on 04/29/2021). 13
- [25] Faria, Henrique Malloni de: *Projeto de um processador banda base de uma tag passiva de rfid*. FGA UNB, 2015. 13

- [26] Vowels, Susan A: *Understanding rfid (radio frequency identification)*. Em *Ubiquitous and Pervasive Computing: Concepts, Methodologies, Tools, and Applications*, páginas 54–64. IGI Global, 2010. 13
- [27] *Iso - iso 11784:1996 - radio frequency identification of animals — code structure*. <https://www.iso.org/standard/25881.html>. (Accessed on 05/19/2021). 13
- [28] *Iso 11785:1996(en), radio frequency identification of animals — technical concept*. <https://www.iso.org/obp/ui/#iso:std:iso:11785:ed-1:v1:en>. (Accessed on 05/19/2021). 13
- [29] *Iso - iso/iec 14443-1:2018 - cards and security devices for personal identification — contactless proximity objects — part 1: Physical characteristics*. <https://www.iso.org/standard/73596.html>. (Accessed on 05/19/2021). 13
- [30] *Iso - iso/iec 15961:2004 - information technology — radio frequency identification (rfid) for item management — data protocol: application interface*. <https://www.iso.org/standard/30528.html>. (Accessed on 05/18/2021). 14
- [31] *Iso - iso/iec 15962:2004 - information technology — radio frequency identification (rfid) for item management — data protocol: data encoding rules and logical memory functions*. <https://www.iso.org/standard/30529.html>. (Accessed on 05/19/2021). 14
- [32] *Iso - iso/iec 15963:2009 - information technology — radio frequency identification for item management — unique identification for rf tags*. <https://www.iso.org/standard/52124.html>. (Accessed on 05/19/2021). 14
- [33] *Iso - iso/iec 15693-2:2006 - identification cards — contactless integrated circuit cards — vicinity cards — part 2: Air interface and initialization*. <https://www.iso.org/standard/39695.html>. (Accessed on 05/19/2021). 14
- [34] *Iso - iso/iec 18000-1:2008 - information technology — radio frequency identification for item management — part 1: Reference architecture and definition of parameters to be standardized*. <https://www.iso.org/standard/46145.html>. (Accessed on 05/19/2021). 14
- [35] *Iso - iso/iec 18000-2:2004 - information technology — radio frequency identification for item management — part 2: Parameters for air interface communications below 135 khz*. <https://www.iso.org/standard/34113.html>. (Accessed on 05/19/2021). 14
- [36] *Iso - iso/iec 18000-3:2010 - information technology — radio frequency identification for item management — part 3: Parameters for air interface communications at 13,56 mhz*. <https://www.iso.org/standard/53424.html>. (Accessed on 05/19/2021). 14
- [37] *Iso - iso/iec 18000-4:2018 - information technology — radio frequency identification for item management — part 4: Parameters for air interface communications at 2,45 ghz*. <https://www.iso.org/standard/68145.html>. (Accessed on 05/19/2021). 14

- [38] Iso - iso/iec 18000-6:2010 - information technology — radio frequency identification for item management — part 6: Parameters for air interface communications at 860 mhz to 960 mhz. <https://www.iso.org/standard/46149.html>. (Accessed on 05/19/2021). 14
- [39] Iso - iso/iec 18000-7:2014 - information technology — radio frequency identification for item management — part 7: Parameters for active air interface communications at 433 mhz. <https://www.iso.org/standard/57336.html>. (Accessed on 05/19/2021). 14
- [40] Padrões de identificação. <https://www.gs1br.org/codigos-e-padroes/padroes-de-identificacao>. (Accessed on 04/30/2021). 15
- [41] Olsen, Tava Lennon e Brian Tomlin: *Industry 4.0: Opportunities and challenges for operations management*. Manufacturing & Service Operations Management, 22(1):113–122, 2020. 15
- [42] Sakurai, Ruudi e Jederson Donizete Zuchi: *As revoluções industriais até a indústria 4.0*. Rejourvista Interface Tecnológica, 15(2):480–491, 2018. 15, 16
- [43] Prisecaru, Petre: *Challenges of the fourth industrial revolution*. Knowledge Horizons. Economics, 8(1):57, 2016. 16
- [44] Schwab, Klaus: *The fourth industrial revolution*. Currency, 2017. 16
- [45] Moretti, Eduardo de Araujo, Rosley Anholon, Izabela Simon Rampasso, Dirceu Silva, Luis Antonio Santa-Eulalia e Paulo Sérgio de Arruda Ignácio: *Main difficulties during rfid implementation: An exploratory factor analysis approach*. Technology Analysis & Strategic Management, 31(8):943–956, 2019. 16
- [46] Mekruksavanich, Sakorn: *Supermarket shopping system using rfid as the iot application*. Em *2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, páginas 83–86. IEEE, 2020. 17
- [47] Santos, Bruno P, Lucas AM Silva, CSFS Celes, João B Borges, Bruna S Peres Neto, Marcos Augusto M Vieira, Luiz Filipe M Vieira, Olga N Goussevskaia e A Loureiro: *Internet das coisas: da teoria à prática*. Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 31, 2016. 17
- [48] Oliveira, Alessandro de Souza e Milene Franco Pereira: *Estudo da tecnologia de identificação por radiofrequência-rfid*. BDM UNB, 2006. 18
- [49] Rei, António Jorge Laranjeira et al.: *Rfid versus código de barras da produção à grande distribuição*. Faculdade do Porto FEUP, 2010. 18, 19
- [50] *Sobre o ruby*. <https://www.ruby-lang.org/pt/>. (Accessed on 09/02/2022). 20
- [51] *Ruby on rails*. <https://rubyonrails.org/doctrine>. 21

- [52] *Getting started with rails.* [https://guides.rubyonrails.org/v6.1/getting\\_started.html](https://guides.rubyonrails.org/v6.1/getting_started.html). (Accessed on 09/02/2022). 21
- [53] *Welcome to rails.* <https://github.com/rails/rails>. (Accessed on 09/02/2022). 21
- [54] *Active record.* <https://github.com/rails/rails/blob/main/activerecord/README.rdoc>. (Accessed on 09/02/2022). 21
- [55] *Active model.* <https://github.com/rails/rails/blob/main/activemodel/README.rdoc>. (Accessed on 09/02/2022). 21
- [56] *Action pack.* <https://github.com/rails/rails/blob/main/actionpack/README.rdoc>. (Accessed on 09/02/2022). 22
- [57] *Action view.* <https://github.com/rails/rails/blob/main/actionview/README.rdoc>. (Accessed on 09/02/2022). 22
- [58] *Action mailer.* <https://github.com/rails/rails/blob/main/actionmailer/README.rdoc>. (Accessed on 09/02/2022). 22
- [59] *Action mailbox.* <https://github.com/rails/rails/blob/main/actionmailbox/README.md>. (Accessed on 09/02/2022). 22
- [60] *Active job.* <https://github.com/rails/rails/blob/main/activejob/README.md>. (Accessed on 09/02/2022). 22
- [61] *Action cable.* <https://github.com/rails/rails/blob/main/actioncable/README.md>. (Accessed on 09/02/2022). 22
- [62] **Active Storage.** <https://github.com/rails/rails/blob/main/activestorage/README.md>. (Accessed on 09/02/2022). 22
- [63] *Action text.* <https://github.com/rails/rails/blob/main/actiontext/README.md>. (Accessed on 09/02/2022). 23
- [64] *Active support.* <https://github.com/rails/rails/blob/main/activesupport/README.rdoc>. (Accessed on 09/02/2022). 23
- [65] *Encontre, instale e publique rubygems.* <https://rubygems.org/>. 23
- [66] *Postgresql 14.2 documentation.* <https://www.postgresql.org/docs/current/index.html>. 24
- [67] *Equipe e história.* <https://getbootstrap.com.br/docs/4.1/about/overview/>. (Accessed on 25/02/2022). 24
- [68] *About bootstrap.* <https://getbootstrap.com.br/docs/4.1/about/overview/>. (Accessed on 09/02/2022). 24
- [69] *The heroku platform.* <https://www.heroku.com/platform>. 25

- [70] What heroku. <https://www.heroku.com/what>. (Accessed on 09/02/2022). 25
- [71] Chafon: *CF-H301 UHF RFID User's Manual v1.0*. 26
- [72] Patterson, David A e John L Hennessy: *Computer organization and design: the hardware/software interface, (rev. ed. of: Computer organization and design/john l. hennessy, david a. patterson. 1998.)*, 2012. 28
- [73] Gandhewar, Nisarg e Rahila Sheikh: *Google android: An emerging software platform for mobile devices*. International Journal on Computer Science and Engineering, 1(1):12–17, 2010. 28
- [74] Livia Moura, Gustavo Camargo: *Impacto econômico e social do android no brasil*. [https://www.bain.com/contentassets/20e0815cfd784b4a8dead63475b42380/v02\\_b-impactos-do-android-no-brasil.pdf](https://www.bain.com/contentassets/20e0815cfd784b4a8dead63475b42380/v02_b-impactos-do-android-no-brasil.pdf). (Accessed on 23/02/2022). 28
- [75] Mercado em torno do android movimenta r\$ 136 bilhões, 2% do pib / exame. <https://exame.com/tecnologia/mercado-em-torno-do-android-movimenta-r-136-bilhoes-2-do-pib/>. (Accessed on 08/31/2021). 29
- [76] Reck, Marcelo Sala: *Beacons ble-bluetooth low energy-design e análise de um sistema de localização indoor*. Universiade de Caxias do Sul, 2017. 29
- [77] Bonatto, Aurélio e DO Canto: *Bluetooth technology (ieee 802.15)*. Sem publicação, sem publicação, páginas 01–11, 2007. 29
- [78] Costa, Thiago Luiz Leite Ribeiro da: *Sistema de controle de inventário através da tecnologia rfid*. Repositorio CEUB, 2009. 30, 32
- [79] Mekruksavanich, Sakorn: *Supermarket shopping system using rfid as the iot application*. Em *2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT NCON)*, páginas 83–86, 2020. 30, 32
- [80] Santos, Yago Luiz dos: *Implementação de arquitetura baseada em iot para leitura de tags rfid de ultra alta frequência*. BDM UNB, 2019. 30, 32
- [81] Sangari, M Siva, L Kaviya, KL Meenakshi e R Nikaldharan: *Intelligent smart shopping and stock notification using iot*. Annals of the Romanian Society for Cell Biology, páginas 306–311, 2021. 30, 32
- [82] Adiono, Trio, Hans Ega, Hans Kasan e Carrel Suksmandhira Harimurti: *Fast warehouse management system (wms) using rfid based goods locator system*. Em *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*, páginas 1–2. IEEE, 2017. 30, 32
- [83] Safdar, Saria, Shoab Ahmed Khan, Arslan Shaukat e Usman Akram: *Genetic algorithm based automatic out-patient experience management system (gapem) using rfids and sensors*. IEEE Access, 2020. 30, 32

- [84] Zhao, Jumin, Fangfang Xue e Deng-ao Li: *Intelligent management of chemical warehouses with RFID systems*. Sensors, 20(1):123, 2020. <https://doi.org/10.3390/s20010123>. 31, 32
- [85] Trab, Sourour, Eddy Bajic, Ahmed Zouinkhi, Mohamed Naceur Abdelkrim e Hassen Chekir: *RFID iot-enabled warehouse for safety management using product class-based storage and potential fields methods*. Int. J. Embed. Syst., 10(1):71–88, 2018. <https://doi.org/10.1504/IJES.2018.10010505>. 31, 32
- [86] López, Yuri Álvarez, Jacqueline Franssen, Guillermo Álvarez Narciani, Janet Pagnozzi, Ignacio González-Pinto Arrillaga e Fernando Las-Heras Andrés: *RFID technology for management and tracking: e-health applications*. Sensors, 18(8):2663, 2018. <https://doi.org/10.3390/s18082663>. 31, 32
- [87] Zhang, Ting, Chuangle Cao, Yaqiang Liu e Hongtao Yu: *Design and implementation of kiwi fruit cold storage management system based on RFID*. Em *16th International Wireless Communications and Mobile Computing Conference, IWCMC 2020, Limassol, Cyprus, June 15-19, 2020*, páginas 2194–2198. IEEE, 2020. <https://doi.org/10.1109/IWCMC48107.2020.9148540>. 32
- [88] Pressman, Roger S e Bruce R Maxim: *Engenharia de software-9*. McGraw Hill Brasil, 2021. 36
- [89] Sommerville, I.: *Engenharia de software*. Pearson Prentice Hall, 2011, ISBN 9788579361081. <https://books.google.com.br/books?id=H4u5ygAACAAJ>. 38, 39

# Apêndice A

## Códigos Fontes

Abaixo é apresentado os repositórios onde estão localizados todos os códigos fontes implementados neste trabalho:

- **Código Fonte Sistema Web:** <https://github.com/CaioSMendes/wms-rfid-tcc>
- **Código Aplicativo Mobile:** <https://github.com/CaioSMendes/appTccRfid>
- **Código API:** <https://github.com/CaioSMendes/UpcRfidApiUnb>
- **Link da hospedagem da API:** <https://upc-rfid-api-unb.herokuapp.com/>
- **SDK Leitor Chafon CF-H301:** <http://www.chafon.com/DownloadFile.aspx?fid=237>