

INSTITUTO FEDERAL DE SANTA CATARINA

ALANA BEATRIZ ROSTIROLLA MANDIM

**Sistema IoT para Gestão de Frotas do  
Destacamento de Controle do Espaço Aéreo de  
Florianópolis**

São José - SC

Julho/2023

# RESUMO

A tecnologia é uma ferramenta presente na vida de todos, com o objetivo de auxiliar na execução de tarefas. Substituir trabalhos manuais por soluções tecnológicas é uma forma de minimizar erros e otimizar processos. Com o objetivo de melhorar a gestão de frotas do Destacamento de Controle do Espaço Aéreo de Florianópolis (DTCEA-FL), este trabalho irá desenvolver um sistema de gestão de frotas utilizando a tecnologia Internet das Coisas (IoT). O propósito é criar uma solução IoT que substitua o formulário manual de controle de entrada e saída dos veículos do destacamento para reduzir o uso de papel e tornar as informações mais ágeis. O uso de um sistema juntamente com ferramentas IoT facilita o armazenamento de dados e reduz os documentos físicos, promovendo mudanças culturais e organizacionais dentre os colaboradores. Para isso, será abordados conceitos de gestão de frotas, IoT e algumas de suas opções de identificação de coisas e pessoas bem como o uso de banco de dados e plataformas *No-Code*. Em seguida será apresentada a metodologia empregada que engloba desde a identificação dos problemas até o desenvolvimento do sistema.

**Palavras-chave:** Gestão de Frotas. Internet das Coisas.

# LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura de três camadas . . . . .	11
Figura 2 – Arquitetura com cinco camadas . . . . .	12
Figura 3 – Link da pagina principal da Wikipédia em QR Code . . . . .	13
Figura 4 – Etapas de modelagem do projeto de banco de dados. . . . .	19
Figura 5 – Top 10 dos SGBDR mais utilizados no mundo. . . . .	21
Figura 6 – Fluxo Cliente-Servidor através da API. . . . .	23
Figura 7 – Diagrama de Uso. . . . .	29
Figura 8 – Componentes do Sistema . . . . .	31

# LISTA DE TABELAS

Tabela 1 – Partes do QR Code. . . . .	15
Tabela 2 – Comparativo do QR Code com RFID. . . . .	17
Tabela 3 – Comparativo de Soluções Tecnológicas em Gestão de Frotas. . . . .	30
Tabela 4 – Cronograma das atividades previstas. . . . .	33

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>5</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>8</b>
<b>2.1</b>	<b>Gestão de Frotas</b>	<b>8</b>
2.1.1	Importância da gestão de frotas para as empresas	9
<b>2.2</b>	<b>Internet da Coisas</b>	<b>10</b>
2.2.1	Arquiteturas e Componentes da IoT	10
<b>2.3</b>	<b>Identificação de coisas e pessoas</b>	<b>12</b>
2.3.1	RFID e QR Code	13
2.3.2	Biometria	16
<b>2.4</b>	<b>Banco de Dados</b>	<b>18</b>
2.4.1	Níveis de abstração de banco de dados	19
2.4.2	Segurança e Controle dos Dados	21
2.4.3	Acesso ao BD via RESTful	22
<b>2.5</b>	<b>Plataforma No-Code para aplicações Web</b>	<b>25</b>
<b>3</b>	<b>PROPOSTA</b>	<b>27</b>
<b>3.1</b>	<b>Levantamento de dados</b>	<b>27</b>
3.1.1	Análise de requisitos e casos de uso	27
3.1.2	Soluções no mercado	29
<b>3.2</b>	<b>Arquitetura da solução proposta</b>	<b>31</b>
<b>3.3</b>	<b>Projeto do Banco de dados</b>	<b>32</b>
<b>3.4</b>	<b>Desenvolvimento do sistema</b>	<b>32</b>
<b>3.5</b>	<b>Testes e validação</b>	<b>32</b>
<b>3.6</b>	<b>Cronograma de Atividades</b>	<b>32</b>
	<b>REFERÊNCIAS</b>	<b>34</b>

# 1 INTRODUÇÃO

A gestão de frotas é essencial em uma empresa que necessita do uso de veículos para a sua atividade fim. É um desafio complexo envolvendo muitos fatores como segurança tanto do motorista como da empresa, custos operacionais, manutenção e logística de transporte. Com tantos desafios apresentados se faz necessário uma busca por soluções que auxiliem nesta árdua tarefa.

Nesta conjuntura, a Internet das Coisas (IoT) surge como uma tecnologia que pode trazer benefícios para a gestão de frotas, pois a mesma permite coleta de dados em tempo real sobre os veículos, condições de tráfego, e dos motoristas. A utilização de sensores e dispositivos IoT pode gerar informações precisas, permitindo um controle mais eficiente das frotas.

A IoT, segundo [Jara et al. \(2014\)](#), tem em sua composição objetos inteligentes cuja diferenciação se dá pelo fato de serem dispositivos físicos e restritos no que diz respeito a sua capacidade que possuem de computação, memória, comunicação e autonomia de energia. Formados tanto por códigos quanto por etiquetas de identificação, esses dispositivos possibilitam que algo em específico seja identificado de maneira única.

[Vermesan et al. \(2009\)](#) explica que a internet das coisas consiste em uma infraestrutura de rede global e dinâmica, cuja capacidade é de autoconfiguração com base nos protocolos de comunicação interoperáveis e padronizados em que as “coisas” virtuais e físicas possuem personalidades virtuais, atributos físicos, identidades, utilização interfaces inteligentes e são totalmente integradas à rede de informação.

Em uma aplicação de controle de manutenção dispositivos IoT podem ser utilizados para monitorar as condições dos veículos como, por exemplo, no Destacamento de Espaço Aéreo de Florianópolis (DTCEA-FL), onde ainda são utilizadas fichas de papel impressas para o controle de entrada e saída de veículos.

No DTCEA-FL há muitas missões programadas onde o militar entra em contato com o Encarregado das Viaturas e informa que tem uma missão prevista para uma determinada data, consulta a disponibilidade de viaturas. Caso tenha alguma viatura disponível, é feita uma “reserva”, na qual a viatura deve estar em condições de realizar a viagem na data e hora informadas, sendo que a mesma não pode ser utilizada no dia da reserva por outro militar. Essa reserva ocorre somente de forma verbal, o encarregado passa a informação aos motoristas do dia anterior para deixar a viatura abastecida e limpa, e os mesmos têm que informar a equipe seguinte. No entanto, por vezes, ocorre esquecimento as atividades previstas não são realizadas.

A questão da reserva da viatura também se enquadra quando, por algum problema mecânico, está impossibilitada de uso. Já ocorreram casos em que o encarregado determinou que não fosse utilizada a viatura e mesmo assim algum motorista fez uso da mesma. Tal fato se deu pela falta de informação, pois o sargento de dia que estava no momento em que a viatura foi “interditada” não informou àquele que o rendeu e facilitou o erro.

Considerando os fatores apresentados, surge a necessidade de criar uma solução para tais problemas, dando ao controle de viaturas credibilidade novamente, contribuindo para a sustentabilidade e trazendo segurança tanto para a instituição quanto para seus militares.

Neste contexto, o objetivo principal deste trabalho é desenvolver um sistema utilizando a tecnologia IoT, visando a resolução dos problemas acima apresentados na gestão de frotas das viaturas pertencentes ao DTCEA-FL, de forma que seja de baixo custo para implementação. Como objetivos específicos tem-se:

- Permitir que o gestor (encarregado) cadastre os motoristas que possuem autorização para utilizar as viaturas de forma rápida e simplificada através de uma interface gráfica;
- Possibilitar que o motorista solicite ou reserve uma viatura;
- Fornecer autonomia ao Sargento de Dia para autorizar ou não as solicitações de utilização dos motoristas;
- Possibilitar que o gestor bloqueie uma viatura de modo que os motoristas não consigam realizar solicitações para a mesma viatura;
- Manter dados armazenados de forma a possibilitar uma geração de relatório permitindo que seja feita a visualização das entradas e saídas das viaturas sempre que necessário;
- Realizar a verificação via sistema da autorização do motorista para sair com a viatura;
- Permitir uma fácil visualização, por parte do sargento de dia, sobre a situação de cada viatura, ou seja, se a mesma se encontra na garagem ou se está em missão.

Com base nos objetivos estabelecidos, a hipótese é de que, para solucionar essa questão, pode-se realizar a implantação de um sistema no qual o motorista realiza a solicitação da viatura através do sistema, o sargento de dia recebe a solicitação e aprova ou não. Tendo a aprovação, basta que o motorista retire a chave da viatura e se dirija com a mesma para a guarda (saída), onde o guarita (soldado de serviço) realiza sua identificação e verifica no sistema se o motorista possui autorização ou não. Em caso de resposta positiva,

o motorista recebe autorização para sua saída. Nos casos de reserva e bloqueio de viatura, o sistema terá essas informações registradas, o que auxilia o sargento de dia e impossibilita erros como o de uso de viatura não autorizada, trazendo transparência e segurança para todos envolvidos. Para a geração de relatório, todos os dados seriam armazenados em um banco de dados para que seja possível gerá-lo sempre que houver necessidade.

Esse trabalho seguirá com os seguintes capítulos: fundamentação teórica, apresentação da concepção e implementação do sistema, teste do protótipo desenvolvido e conclusão. Na fundamentação teórica será apresentada toda informação necessária para a compreensão da tecnologia que será aplicada e trará uma pesquisa de mercado para verificar soluções existentes. Em seguida, serão apresentadas a concepção e implementação do sistema, incluindo componentes de hardware e software. No teste do protótipo serão apresentados os testes realizados com seus respectivos resultados. Ao final do estudo será apresentada a conclusão do trabalho.



# 2 FUNDAMENTAÇÃO TEÓRICA

## 2.1 Gestão de Frotas

Compreende-se que a gestão de frotas consiste na atividade de gerenciar, administrar ou reger um conjunto de ativos, em específico, os veículos de uma organização ou empresa. A tarefa é bastante ampla e passa por vários serviços, como dimensionamento, cálculo de custos, manutenção mecânica ou roteirização, por exemplo. A tarefa de fazer o cálculo preciso de um modelo de participação da frota nos custos e no patrimônio das instituições não é algo considerado simples, visto que essa parcela tem variação de caso para caso e depende dos propósitos da empresa, por exemplo, se a carga é de passageiros, se é própria, de terceiros; da maneira como são contabilizados e inseridos os veículos na frota, por meio de leasing, aluguel ou financiamento; e da natureza da operação (VALENTE; PASSAGLIA; NOVAES, 1997).

De acordo com (BERTAGLIA, 2017), as empresas que possuem envolvimento com o serviço de transporte tanto com veículos próprios quanto com terceirizados, precisam fazer o controle e ter conhecimento de modo efetivo dos seus ativos e seguir práticas de negócios importantes, visto que é a melhor maneira de manter a competitividade e diminuir seus custos logísticos e, conseqüentemente, o preço do produto final, ficando clara a relevância da administração correta. Um dos elementos considerados mais relevantes na administração de transporte é a gestão das frotas, especialmente se for a sua atividade final da cadeia de abastecimento. Em diversas situações, a movimentação de carga possui um peso considerável na qualidade do serviço prestado e na formação dos custos logísticos. No decorrer dos anos, as empresas que tinham frotas menores tiveram um crescimento e passaram a ampliar a quantidade de veículos. Entretanto, isso não significa que a sua operação continuou sendo feita de modo mais eficiente, o que pode, em diversos casos, aumentar os custos de modo desnecessário e excessivo, ou mesmo perder a qualidade na prestação do serviço (BERTAGLIA, 2017).

A gestão de frotas tem vários aspectos importantes que necessitam receber atenção, como a inutilização da carga, indicadores de produtividade e desempenho, eficiência da frota, planejamento de atividades administrativas, estoque de peças de reposição, manutenção mecânica, consumo de combustível, mão de obra, administração dos ativos, dentre outros. A combinação desses fatores justifica a utilização da tecnologia da informação para administrar os transportes. Com o progresso da tecnologia, várias ferramentas importantes foram elaboradas para o fornecimento de suporte para as tarefas de processo de transporte, como a telemática e os sistemas de otimização de rastreamento de frota e de rotas, por

exemplo (BERTAGLIA, 2017).

### 2.1.1 Importância da gestão de frotas para as empresas

A gestão de frotas tem o papel importante de gerenciar os custos de combustível, de manutenção, de aquisição e renovação dos veículos, assegurando que todos os seus processos aconteçam de forma harmoniosa com as solicitações dos usuários, não ocorrendo atrasos ou mesmo a falta de atendimento.

De acordo com Valente, Passaglia e Novaes (1997), a gestão de frotas é importante para o planejamento, o gerenciamento e a execução de atividades relacionadas aos veículos de uma instituição, sendo exigido do gestor um conhecimento teórico e prático das tarefas que englobam os equipamentos, as tecnologias, os veículos e a mão de obra existentes para o auxílio do controle e da execução desta gestão.

Na gestão são utilizados alguns métodos, técnicas e ferramentas variadas, como softwares, que permitem que as empresas eliminem e/ou minimizem riscos inerentes ao investimento de seus veículos, ampliando a eficiência e a produtividade dos seus serviços (SALLES, 2011).

O controle de forma informatizada viabiliza o encontro de dados gerenciais com confiabilidade, porém poderá ter como resultado um fracasso informatizado caso o sistema seja ineficiente (CAMPOS; BELHOT, 1994). Contudo, podem acontecer vários problemas na gestão, como a depreciação e a manutenção dos veículos, o preço dos combustíveis, os seguros dos veículos, os pagamentos dos motoristas, a concorrência, e os custos de transporte elevados. Além disso, ainda existe a falta de zelo pelos condutores, aumentando o desgaste natural dos veículos, usados por diversas horas seguidas por várias vezes (CLEMENTE, 2013).

COTTI (1989) explica que uma gestão de frotas eficaz não consiste em um processo que necessita de sofisticação ou dificuldade, porém, por diversas vezes se tornam necessárias decisões e análises conduzidas pela preocupação real e o bom senso para a melhoria do rendimento de cada dia para que alcance excelentes resultados.

O cenário passar por modificações e se adapta as inovações que trazem facilidade para o dia a dia do gestor. Para ajudar no trabalho de gestão, existem softwares que fazem a contabilização da manutenção, do abastecimento, da demanda dos serviços, dos seus itinerários, dos tipos de utilização dos veículos e da depreciação dos veículos, viabilizando um controle maior das demandas.

## 2.2 Internet da Coisas

A internet trouxe mudanças para o cotidiano do mundo todo, possibilitando o acesso rápido a informações que não eram acessíveis até aquele momento. Porém, se a internet “das pessoas” pode ser vista como uma revolução, a internet “das coisas” (IoT) pode permitir muito mais (SANTOS et al., 2016).

A IoT é considerada um conceito onde os objetos e os dispositivos do cotidiano são equipados com sensores que possuem a capacidade de manterem uma comunicação entre si de modo inteligente. (SANTOS et al., 2016) explicam que os sensores possuem responsabilidade pela coleta de informações com relação ao contexto no qual os objetos estão localizados, fazendo o armazenamento dos dados e podendo realizar a manipulação do ambiente ou ter uma reação conforme os dados lidos. Estes dispositivos monitoram o ambiente em que está localizado o objeto fazendo a captura dos valores das grandezas físicas como presença, pressão, umidade e temperatura. Nessas aplicações há diversos sensores distintos com capacidade de captura dessas grandezas, além de poder realizar a produção de alguma ação, com o acionamento de comandos que podem ser mecânicos, elétricos ou manuais.

Em sua essência, a IoT é somente um ambiente que faz a reunião de informações de diversos dispositivos (semáforos, smartphones, veículos, computadores, dentre outros) e de aplicações (desde um sistema de controle de tráfego a uma aplicação de mídia social)(SANTOS et al., 2016).

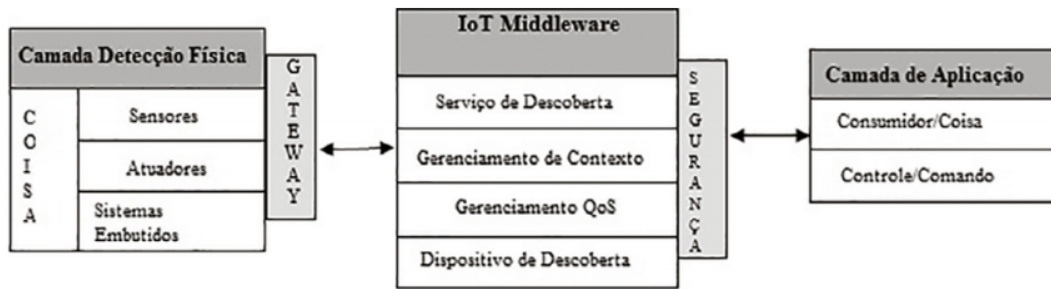
### 2.2.1 Arquiteturas e Componentes da IoT

A definição de Uviase e Kotonya (2018) descreve a arquitetura *Internet of Things Architecture* (IoT-A) como uma estrutura que engloba a especificação dos componentes físicos conectados em rede, a organização e configuração funcional desses componentes, bem como os princípios e procedimentos operacionais associados. Além disso, a arquitetura IoT-A também estabelece os formatos de dados utilizados na operação desses componentes. Essa definição destaca a importância de uma arquitetura sólida e bem definida visando garantir a interoperabilidade e a integração adequada dos dispositivos conectados em uma rede IoT. Ela é responsável por fornecer diretrizes e padrões para a organização e comunicação entre os dispositivos, bem como para a estruturação dos dados que são trocados e processados.

Para compor os diversos modelos de IoT-A existem 5 tipos de camadas diferentes (SATYABRATA, 2022) são elas:

- Camada de Percepção: Nesta camada é onde encontram-se sensores e atuadores, sua principal função é obter informações dos arredores e passar os dados para outra

Figura 1 – Arquitetura de três camadas



Fonte:(MASCHIETTO et al., 2021)

camada como suporte para ações que possam ser tomadas (RFID).

- Camada de Rede: é a camada que recebe os dados vindos da camada de aplicação e envia para a camada *middleware* através de rede 3G, 4G, 5G, WiFi, entre outros. É nela que há a garantia de transferência de dados de forma segura.
- Camada de *middleware*: Responsável por armazenar os conjuntos de dados com base no nome e endereço do dispositivo, composta de recursos de tomada de decisão com base em cálculos, processamento e computação.
- Camada de Aplicação: É a camada responsável pelo gerenciamento de todos os processos de aplicativo com base nas informações passadas pelo *middleware*.
- Camada de Negócios: Trata-se de de uma camada responsável por fazer melhor uso dos dados, trazendo fluxogramas, gráficos, análise de resultados, etc.

Segundo (GLUHAK et al., 2011) a maioria dos experimentos em IoT seguem uma estrutura de duas ou três camadas. A arquitetura de duas camadas é composta pela a camada de percepção e de aplicação. Já a de três camadas, conforme Figura 1, é acrescentado a camada de *middleware* visando a garantida da segurança dos dados. Destaca-se o fato de que essa arquitetura possui uma comunicação restrita e altamente interligada com as camadas de percepção e *middleware*. Sua dependência direta dessas camadas, sem uma camada de rede desacoplada, acaba impactando negativamente a eficiência e escalabilidade das comunicações na IoT.

Observa-se que quanto mais camadas são adicionadas à arquitetura, mais complexa vai se tornando. Em contra partida oferece ao cliente um programa de maior qualidade. Na arquitetura de quatro camadas surge a camada de rede, aumentando o leque de possibilidades de implantação tendo em vista que passa a se ter a Internet como meio de envio de dados, ou seja, a estrutura se baseia na comunicação da camada de percepção com a de rede, em seguida passado para a de *middleware* e ao final a de aplicação.

Figura 2 – Arquitetura com cinco camadas



Fonte: (MASCHIETTO et al., 2021)

Por fim, temos a arquitetura de cinco camadas na qual é adicionada a camada de negócio, que é uma camada criada com o intuito de personalizar mais a aplicação, visando utilizá-la para, através das análises de dados obtidas, sugerir melhoras por exemplo. Vale ressaltar que alguns autores colocam essa camada depois da de aplicação e outros antes conforme a Figura 2.

### 2.3 Identificação de coisas e pessoas

A identificação de coisas e pessoas é um fator importante em determinadas áreas, como controle de acesso, autenticação de usuários e controle de estoque. Em cenários não

Figura 3 – Link da pagina principal da Wikipédia em QR Code



Fonte: ([WIKIPEDIA, 2022](#))

digitalizados tudo é feito de forma manual, o que torna os processos lentos e muitas vezes ineficazes. Nos últimos anos, a tecnologia trouxe avanços significativos que contribuíram na área de controle de acesso, por exemplo, tornando-os capazes de auxiliar na identificação de forma eficiente e confiável. As tecnologias QR Code (Quick Response Code), RFID (Radio Frequency Identification), reconhecimento facial, digital e de íris são exemplos dessas tecnologias.

### 2.3.1 RFID e QR Code

O QR Code foi inventado por uma empresa japonesa chamada Denso-Wave em 1994. Naquela época surgiu a necessidade de que a catalogação de peças fossem realizadas de forma mais rápida. Com isso, desenvolveram uma versão bidimensional do código de barras capaz de armazenar caracteres numéricos, alfanuméricos, especiais e pontuações.

Para realizar a leitura dos QR Codes basta um aparelho celular com câmera. Atualmente, muitos dispositivos já vem com o leitor no próprio celular, caso contrário, basta baixar um dos vários aplicativos disponíveis nos sistemas Android e iOS.

Para entender como funciona o desenho do QR Code é necessário dividir em 7 partes, apresentadas na [Figura 3](#): marcadores de detecção de posição e alinhamento, padrões de temporização, informações sobre a versão e formato, códigos de correção de dados e erros e, finalmente, as margens. A seguir algumas definições dessas partes de acordo com [Denso Wave Inc. \(2023\)](#).

- Marcadores de detecção de posição: Esses elementos permitem que o *scanner*

identifique e leia o QR Code de maneira rápida. Eles atuam como indicadores da orientação em que foi impresso, auxiliando na sua identificação e alinhamento adequado.






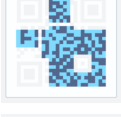

- Marcadores de alinhamento: Empregados para compensar a distorção do QR Code em superfícies curvas. O tamanho e a quantidade desses marcadores podem variar dependendo do volume de informações contidas no código.
- Padrões de temporização: A sequência de módulos pretos e brancos no QR Code define o padrão de informação, conhecido como grade de dados. Por meio dessas linhas, o *scanner* é capaz de identificar o tamanho da matriz de dados.
- Informações sobre a versão: Indicam qual das versões está em uso. Vale ressaltar que são 40 versões.
- Informações de formato: Tem como finalidade facilitar a leitura e decodificação do código, neles são contidos informações sobre a tolerância de erros e o padrão da máscara de dados.
- Códigos de correção de dados e erros: Responsável por armazenar todas as informações, auxiliam na reconstrução de dados perdidos compartilhando o espaço com os módulos de correção de erros.
- Margens: Sempre na cor branca, servem para delimitar o QR Code.

A [Tabela 1](#) apresenta cada parte para melhor visualização.

Os QR Codes se dividem em dois tipos, o estático que armazenam informações “fixas” e permanentes, não qual as informações são codificadas diretamente no código e não podem ser modificadas posteriormente sem que seja gerado um novo código e dinâmico, que é um código que contém informações que podem ser alteradas/atualizadas sempre que necessário sem a necessidade de ser gerado um novo código. O uso de casa um deles depende do que se deseja. O estático é voltado para identificação de usuário, ou quando se quer deixar uma informação não muito extensa para ser compartilhada com um público, já o dinâmico por ter a flexibilidade em seu conteúdo e não necessidade de reemissão de um novo código, é voltado ao uso em redes sociais, cupom de descontos, chaves pix entre outros. A viabilidade do QR Code é possibilitada através de plataformas de gerenciamento de QR Codes dinâmicos, elas possuem ferramentas necessárias desde a criação, atualização até seu rastreamento e análise, essas duas ultimas ferramentas, são muito úteis pois permite saber a quantidade de vezes que o código foi escaneado e quando e onde ocorreu .

Um outro dispositivo muito útil na identificação de coisas é o RFID, considerado como uma variação do código de barras, mas que dispensa o contato visual para identificação já que sua identificação é feita por radiofrequência. Os dados são codificados em etiquetas

Tabela 1 – Partes do QR Code.

Partes	Imagem
Marcadores de detecção de posição	
Marcadores de alinhamento	
Padrões de temporização	
Informações sobre a versão	
Informações de formato	
Códigos de correção de erro e dados	
Margens	

RFID e, para realizar a identificação, as antenas capturam as ondas de rádio. Suas aplicações são em gestão de inventário, rastreamento de itens, controle de acesso entre outros.

O RFID trabalha com quatro faixas de frequência: *Low Frequency* (LF), *High Frequency*(HF) (13,56MHz), *Ultra High Frequency* (UHF) que possuem as seguintes aplicações nas bandas respectivamente, de 100 a 500KHz, 10 a 15MHz, 850 a 950MHz e 2,4 a 5,8 GHz. Cada uma delas tem um uso específico sendo levado em consideração suas vantagens e desvantagens. Em controle de inventario, controle de acesso, identificação de animal, o uso da baixa frequência é mais indicado pelo seu baixo custo, faixa de curta a média leitura baixa velocidade, por exemplo. Já para o monitoramento de veículo em trânsito, o mais indicado é a UHF devido a sua larga faixa de leitura e alta velocidade de leitura, tal fato se dá pelo uso de frequências mais altas [Teleco \(2023\)](#).

A *tag* é um dos componentes da tecnologia RFID, sendo responsável por responder e receber as transmissões. São classificadas nos tipos a ativa e a passiva, onde a ativa possui bateria interna para alimentação e permite tanto a leitura quanto escrita, a passiva



obtem energia através das ondas eletromagnéticas enviadas pela antena do leitor.

No quesito comunicação, ele possui três tipos, sendo eles:

- Half-Duplex (HDX): nessa comunicação, envio e recebimento não pode ocorrer de forma simultânea.
- Full-Duplex (FDX): A transmissão e o recebimento podem ocorrer de forma simultânea.
- Sequencial (SEQ): O sinal não é enviado de forma contínua para a etiqueta. “o sinal de energia é enviado somente enquanto os dados são enviados. Torna-se necessário o armazenamento de energia que será utilizada devido a transmissão de dados em pulsos”(LOUREIRO et al., 2020).

A [Tabela 2](#) apresenta um comparativo das duas tecnologias para facilitar na escolha da melhor opção para cada caso específico. Como pode-se observar o QR Code se destaca no custo e rapidez, tendo em vista que existem sites que permitem a geração de forma gratuita e para realizar a sua leitura, atualmente a maioria dos celulares possuem essa ferramenta, já o RFID tem um custo inicial mais alto devido ao preço das tags, leitores e infraestrutura de suporte. No quesito segurança, alguns tipos de tags RFID possuem recursos de criptografia embutidos para proteger os dados armazenados nas tags. No entanto, as tags RFID de baixo custo e algumas implementações mais antigas podem não ter proteção adequada, o que torna possível a leitura e clonagem não autorizada dos dados já o QR Code ao vincula-lo a uma URL ou conteúdo online, é possível implementar medidas de segurança, como autenticação por senha ou autenticação de dois fatores no *website* ou aplicativo associado a ele. A duração do QR Code se destaca pelo fato de não possuir uma duração definida, tendo em vista que o mesmo vai funcionar em quanto estiver legível, já as tags por mais resistentes que sejam, tendem a danificar mais facilmente e se perder. Analisando a rapidez da leitura, o RFID se destaca relativamente devida a seu uso de radio frequência, permite leituras em tempo real o QR Code por sua vez, depende da captura da imagem pela câmera e o processamento do código o que acaba gerando um *delay*. Vale ressaltar que na tabela o RFID “ganhou” na localização dos itens porém, o QR Code dinâmico, possui a possibilidade de rastreamento.

### 2.3.2 Biometria

A biometria é uma área que se dedica ao estudo e desenvolvimento de técnicas para a identificação e autenticação de indivíduos com base em suas características únicas e distintas, sejam elas físicas ou comportamentais [Biometria \(2023\)](#). Essas características são utilizadas para criar um perfil biométrico que permite reconhecer e verificar a identidade de uma pessoa de forma precisa e confiável. Existem diversas formas de reconhecimento

Tabela 2 – Comparativo do QR Code com RFID.

	QR Code	RFID
Custo de criação	✓	
Rapidez na criação	✓	
Segurança	✓	
Duração do uso	✓	
Rapidez		✓
Localização dos itens		✓

por biometria, seja ela através da digital, reconhecimento facial, reconhecimento por voz, pela íris, entre outros. O fato é que esse tipo de estudo é antigo e vem cada vez mais sendo utilizado no dia-a-dia, seja no controle de acesso ou na segurança pública utilizando sistemas de busca facial. A questão é que as vastas possibilidades abrem margem a dúvidas sobre qual seria a melhor opção.

O reconhecimento facial se baseia em um software que faz a captura do rosto da pessoa e analisa em 2D. Em seguida ele observa a distância entre os olhos, a profundidade das órbitas oculares, a distância entre a testa e o queixo – dados que se julgam particulares de cada pessoa. Após a captura dos dados, o programa realiza uma espécie de extração onde converte a imagem em um código único, para que quando o usuário for se autenticar, ele realizar uma comparação com a imagem obtida na autenticação com o banco de dados. A grande questão deste tipo de reconhecimento é que ele não é 100% preciso. Segundo uma matéria divulgada pelo site [Digital \(2018\)](#), em uma partida da *Champions League* o sistema de reconhecimento facial da polícia errou em 92% ao reconhecer potenciais ameaças, ou seja, ele “condenou” pessoas inocentes.

O reconhecimento através da íris possui algumas etapas. Inicialmente é feita a captura da imagem da íris, em seguida os dados são processados visando separar a imagem da íris das outras imagens do resto do olho. Após o processamento, ocorre a codificação, onde a íris capturada gera um código único que corresponde a ela. Para realizar o reconhecimento, é utilizado um teste de independência estatística entre dois códigos de íris diferentes. A probabilidade de erro deste tipo de reconhecimento é muito baixa, porém possui limitações: ele não consegue fazer um reconhecimento de um fluxo grande de pessoas de forma ágil; e a distância do identificador para os olhos é curta (VALADÃO, 2008).

A impressão digital é formada pela o conjunto de elevações da pele, conhecida como papilas, presentes nos dedos. Foi um dos primeiros tipos de autenticação utilizados devido a sua unicidade. Antes feito por peritos por inspeção visual, atualmente são utilizados software que contam com um amplo banco de dados para armazenar essas informações. O processo de reconhecimento digital segue exatamente as mesmas etapas que os demais: obtenção dos dados, criação de um código correspondente, armazenamento em banco de

dados. Para a validação é feito uma verificação da informação obtida com a presente no banco. Esse método é considerado um dos mais seguras. Em contra partida, suas desvantagens são que sujeiras, dedos machucados, ou alguma condição que afete a pele dos dedos acabam por atrapalhar o reconhecimento. Por exemplo, no desbloqueio de celulares, acontecem casos de ter que repetir a autenticação por algum erro na leitura.

## 2.4 Banco de Dados

Os bancos de dados surgiram como uma solução para lidar com a crescente necessidade de armazenar dados de maneira permanente e organizada. Com o contínuo aumento no volume de dados gerados, tornou-se essencial ter um sistema capaz de gerenciar essas informações de forma eficiente. Foi nesse contexto que os sistemas de gerenciamento de dados (SGBDs) entraram em cena, desempenhando um papel fundamental na extração de informações relevantes para a tomada de decisões. Esses sistemas garantem a integridade dos dados, possibilitam a recuperação rápida e eficiente das informações armazenadas e oferecem recursos avançados para consulta, análise e manipulação dos dados. Dessa forma, os SGBDs se tornaram peças essenciais na gestão de dados em diversas áreas, contribuindo para a eficiência e o sucesso das organizações no mundo moderno.

O uso do termo banco de dados é mais restrito devido três características:

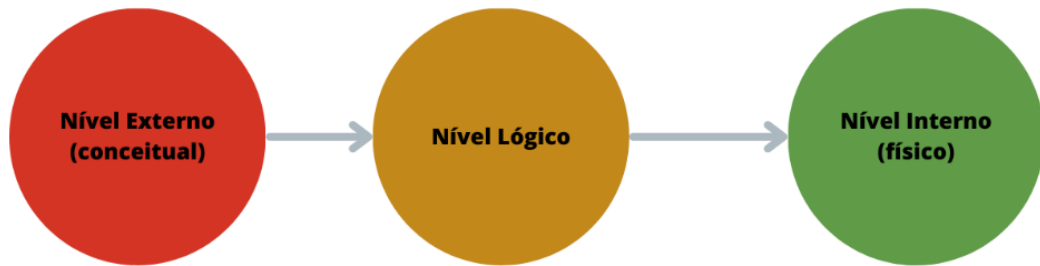
- “Um banco de dados representa uma porção do mundo real, o qual chamamos de minimundo ou Universo de Discurso. Qualquer alteração sofrida por esse minimundo deve ser refletida no banco de dados.
- Um banco de dados é um conjunto lógico e ordenado de dados que possuem algum significado, e não uma coleção aleatória sem um fim ou objetivo específico.
- Um banco de dados é construído e povoado com dados que tem um determinado objetivo, com usuários e aplicações desenvolvidas para manipula-los.” (ALVES, 2014, p. 16)

Pode-se então afirmar que um banco de dados consiste em um agrupamento de dados inter-relacionados, enquanto um SGBD é composto por um conjunto de ferramentas e programas que possibilitam aos usuários criar e manter seus próprios bancos de dados.

Atualmente, existem diversos tipos de banco de dados, onde cada um tem uma característica específica:

- Relacional: O modelo de banco de dados relacional consiste em uma coleção de conjuntos de dados organizados em tabelas, com linhas e colunas, que possuem uma relação claramente definida entre si. A linguagem Structured Query Language (SQL) é utilizada para fornecer uma interface simplificada para interagir com o banco de dados (WADE; CHAMBERLIN, 2012).

Figura 4 – Etapas de modelagem do projeto de banco de dados.



Fonte: Do Autor, utilizando Canva.

- Orientado a Documentos: É um tipo de banco de dados Not Only SQL (NoSQL) que utiliza a estrutura de documentos para armazenar e organizar os dados. Nesse modelo, os documentos são geralmente representados em formatos como JavaScript Object Notation (JSON) ou eXtensible Markup Language(XML), e não em tabelas como no modelo relacional(MACHADO et al., 2017).
- Em memória: é um tipo de banco de dados que armazena e manipula os dados diretamente na memória principal do computador, proporcionando acesso rápido e eficiente aos dados, podendo melhorar significativamente o desempenho e a escalabilidade das aplicações e permitindo a execução de consultas e transações em tempo real. Este dispensa a necessidade de acessar dados em dispositivos de armazenamento secundário, como discos rígidos, resultando em menor latência e maior capacidade de resposta.
- Orientado a Objetos: Esse modelo é um sistema de gerenciamento de banco de dados que permite a criação e modelagem de dados como objetos. Os objetos podem ser referenciados de forma semelhante ao uso de objetos na programação orientada a objetos (KIM, 1990).

#### 2.4.1 Níveis de abstração de banco de dados

Os níveis de abstração de banco de dados são conceitos utilizados para descrever diferentes perspectivas e camadas de um banco de dados. Eles representam diversos níveis de detalhamento e complexidade nas operações e interações com o banco de dados. Esses níveis são utilizados durante o desenvolvimento de um projeto de banco de dados e existem três deles(HEUSER, 2011), conforme pode ser visto na Figura 4.

O nível externo, também conhecido como conceitual, é onde o usuário expõe suas necessidades e define como os dados serão armazenados e relacionados. O nível lógico é o próximo passo no desenvolvimento do projeto, no qual é criado, com base no modelo conceitual, o tipo de banco de dados a ser utilizado, como o Oracle, por exemplo. E o

outro nível é o interno, também conhecido como físico, onde são definidos os tipos de dados a serem armazenados e a implementação da lógica do SGBD.

Ao iniciar o desenvolvimento de um novo sistema, é essencial realizar um levantamento dos requisitos necessários para construir o produto final. Essa análise permite identificar os objetos envolvidos, suas responsabilidades, atributos e seus relacionamentos. Com base nessas informações, surge o modelo conceitual, que é utilizado como guia no desenvolvimento e garante o cumprimento dos requisitos e funcionalidades desejadas. Existem diversos tipos de modelagem conceitual são eles:

- Modelo Entidade-Relacionamento (ER): Amplamente utilizado e representa as entidades (objetos) dos sistemas, seus atributos e relacionamentos entre eles.
- Modelo de Objetos: Baseado na programação orientada a objetos e representa os objetos do mundo real como entidades. Ela também permite a definição de classes, herança, polimorfismos e outros.
- Modelo de Dados Hierárquicos: Os dados são organizados em uma estrutura hierárquica semelhante a uma árvore, com uma única raiz e vários “filhos”.
- Modelo de Dados em Rede: Permite que um nó tenha vários pais. Os dados são organizados em uma estrutura de grafo, com nós representando registros e arestas representando relacionamentos.
- Modelo de Dados Orientado a Documentos: Modelo muito utilizado em banco de dados NoSQL, é baseado no armazenamento de dados em documentos autocontidos e auto descritivos, não existe um relacionamento entre documentos.

Após a fase conceitual é necessário avançar para o nível lógico, onde encontramos a modelagem relacional. Ela se baseia em uma estrutura tabular, onde os dados são organizados em tabelas compostas por linhas (registros) e colunas (atributos). As tabelas representam entidades do mundo real, como produtos, pedidos, e os relacionamentos entre elas são estabelecidos por meio de chaves primárias e chaves estrangeiras, sendo que tais chaves são essenciais para garantir integridade e consistência dos dados por meio de restrições. Na [Figura 5](#) são apresentados os dez sistemas de gerenciamento de banco de dados relacionais (SGBDR) mais populares de acordo com o site [solidITgmbh \(2023\)](#). O cálculo do score é feito através de uma série de parâmetros como número de resultados de buscas no Google e o Bing, interesse geral no sistema utilizando Google Trends, contabilização de perguntas técnicas relacionadas aos bancos em sites como TI Stack Overflow, número de ofertas de emprego em que sistema é mencionado entre outros.

Na etapa final, conhecida como projeto físico, são levadas em consideração particularidades do SGBD que será utilizado na implementação do banco de dados. Nessa fase,

Figura 5 – Top 10 dos SGBDR mais utilizados no mundo.

420 systems in ranking, June 2023

Rank			DBMS	Database Model	Score		
Jun 2023	May 2023	Jun 2022			Jun 2023	May 2023	Jun 2022
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1231.48	-1.16	-56.27
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1163.94	-8.52	-25.27
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	930.06	+9.97	-3.76
4.	4.	4.	PostgreSQL +	Relational, Multi-model ⓘ	612.82	-5.08	-8.02
5.	5.	5.	MongoDB +	Document, Multi-model ⓘ	425.36	-11.25	-55.36
6.	6.	6.	Redis +	Key-value, Multi-model ⓘ	167.35	-0.78	-7.96
7.	7.	7.	IBM Db2	Relational, Multi-model ⓘ	144.89	+1.87	-14.30
8.	8.	8.	Elasticsearch	Search engine, Multi-model ⓘ	143.75	+2.11	-12.25
9.	↑10.	9.	Microsoft Access	Relational	134.45	+3.28	-7.36
10.	↓9.	10.	SQLite +	Relational	131.21	-2.65	-4.22

Fonte: (SOLIDITGMBH, 2023).

além de definir os relacionamentos, tabelas e colunas, também são especificados os tipos de dados para cada coluna e é dada atenção à linguagem utilizada para criar as tabelas no banco de dados (ELMASRI; NAVATHE, 2011). Para melhor compreensão, seguem alguns conceitos de termos utilizados nas modelagens:

- Entidade: uma representação abstrata de um objeto ou conceito do mundo real que possui características e propriedades distintas.
- Atributo: Característica ou propriedade específica de uma entidade.
- Domínio: Conjunto de valores permitidos, ou seja, define se ele é do tipo inteiro, *boolean* e outros.
- Chave primária: é a chave escolhida como principal forma de identificação de cada objeto na entidade. Ela tem como requisito ser única, ou seja, não pode existir outra igual àquela em sua coluna.
- Chave estrangeira: é um campo ou conjunto de campos em uma tabela que estabelece uma relação com uma chave primária em outra tabela. Visa garantir a consistência referencial na tabela e nos dados.
- Relacionamento: define como colunas de uma tabela estão relacionadas às colunas de outras tabelas e descreve as dependências entre elas.

## 2.4.2 Segurança e Controle dos Dados

A segurança em banco de dados sempre é um tema muito levantando nas empresas. Ela concebe medidas que tem por objetivo proteger os dados para que o mesmo não seja

acessado por pessoas não autorizadas. Para que se possa garantir a segurança dos dados, existem três pilares, integridade, disponibilidade e confiabilidade. A integridade visa garantir que as informações terão suas características conservadas. A disponibilidade visa garantir que os dados sempre estarão disponíveis para uso a quem de fato tem autorização e a confiabilidade tem por objetivo impor limites de acesso, ou seja, limitar o acesso a dados de acordo com a necessidade de cada um. [eduardo \(2022\)](#)

Garantir totalmente a segurança em banco de dados é algo praticamente impossível, tendo em vista que a evolução tecnológica e constante conseqüentemente surgem diversas formas de “burlar” os sistemas. Porém existem ferramentas que tem por objetivo impedir ao máximo esses ataques ou falhas. Criptografia, controle de acesso, de ingerência e de fluxo, são medidas que visam proteger os bancos de dados de ameaças à segurança. O controle de acesso é a autenticação, fazer com que usuários tenham login e senha para que possa regular o acesso em um SGBD. O controle de ingerência refere-se ao controle de acesso a bancos de dados estatísticos, que permitem a interpretação de informações estatísticas. Já o controle de fluxo impede acesso de forma descontrolada de dados a usuários não autorizados. Já a criptografia tem como objetivo proteger dados sigilosos, que precisam ser transmitidos por alguma rede de comunicação. [\(PICHETTI; VIDA; CORTES, 2021\)](#).

O administrador de banco de dados (DBA) é a peça chave para gerenciar o SGBD, sendo que ele é responsável por implementar medidas de segurança para proteger o banco de dados contra acessos não autorizados e garantir a integridade dos dados. Isso inclui a criação e gerenciamento de contas de usuários, definição de permissões de acesso, aplicação de políticas de segurança e realização de backups e recuperação de dados.

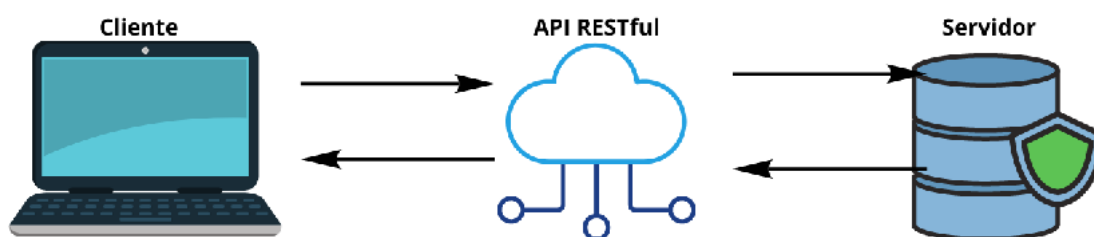
### 2.4.3 Acesso ao BD via RESTful

Antes de conhecer um pouco sobre o que é a API *RESTful*, serão abordados alguns conceitos e nomenclaturas importantes para entender a importância da API no desenvolvimento de aplicações *Web*.

Quando o programador desenvolve um *software* é comum dividi-lo em duas partes denominadas *frontend* e *backend*, onde cada uma tem características e responsabilidades distintas, porém são dependentes uma da outra para o correto funcionamento da aplicação.

Define-se *frontend* como sendo a parte da aplicação voltada para a interação direta do usuário. É a parte responsável por exibir e gerenciar a interface gráfica. Quanto mais intuitiva for, maior a facilidade do usuário na utilização do programa. O foco dessa parte é o *design*, interação, dinamismo e outros fatores que afetem diretamente o usuário. Já o *backend* é a parte da aplicação responsável por desenvolver a lógica, o processamento e a comunicação com o servidor, incluindo todas as requisições feitas pelo usuário, autenticação,

Figura 6 – Fluxo Cliente-Servidor através da API.



Fonte: Do Autor, utilizando Canva.

segurança, regras de negócio e manipulação do banco de dados.

A API *RESTful* desempenha um papel fundamental na integração de dois sistemas, sendo que eles a usam para trocar informações através da internet de modo seguro. Grande parte das aplicações necessitam se comunicar com outras aplicações externas. Ela define as regras de comunicação entre os *software* (Amazon Web Services, 2023). A API *REST* define um conjunto de princípios e diretrizes para a criação de serviços web baseados no protocolo HTTP. Ela utiliza os verbos HTTP, como GET, POST, PUT e DELETE, para realizar operações de criação, leitura, atualização e exclusão de dados (CRUD - Create, Read, Update, Delete).

Analisando o cenário cliente-servidor conforme a Figura 6, o *frontend*, que nesse caso é o cliente, faz requisições à API *REST* para obter ou enviar informações ao *backend*. Tais requisições podem ser feitas através de URLs definidas pela API ao qual correspondem a recursos específicos da aplicação. Já o servidor (*backend*), primeiramente autentica o cliente e dá uma resposta para o cliente informando que ele está autorizado a realizar requisições. Então, o *backend* recebe as requisições, processa-as e retorna as respostas contendo informações indicando se a solicitação foi ou não atendida. Essas respostas vão desde uma consulta a um banco de dados à formatação de resposta de acordo com o formato solicitado, e variam de acordo com o que foi projetado a API.

Conforme o site Amazon Web Services (2023), existem três benefícios das APIs *RESTful*, que são:

- Escalabilidade: Capacidade de um sistema ou aplicação lidar com o aumento na demanda de recursos ou usuários, mantendo o desempenho, confiabilidade e qualidade do serviço. A API *REST* otimiza as interações entre cliente e servidor. A ausência de estado reduz o ônus sobre o servidor, pois ele não precisa armazenar informações de solicitações anteriores dos clientes. Além disso, um cache bem gerenciado pode eliminar parcial ou totalmente certas interações entre cliente e servidor. Esses recursos combinados possibilitam escalabilidade sem criar gargalos de comunicação que afetam negativamente o desempenho do sistema.



- Flexibilidade: Por serem dois sistemas diferentes, tem-se a liberdade de uma constante evolução no desenvolvimento das aplicações, bem como emprego de uma nova tecnologia no *backend* ou *frontend* sem que um afete o outro.
- Independência: O tipo de linguagem de programação aplicado ao cliente não tem qualquer tipo de vínculo com a do servidor, o que torna as duas aplicações independentes uma da outra nesse quesito.

A solicitação do Cliente precisa conter o identificador de recurso exclusivo, no caso o URL, no qual ele especifica qual o caminho para o recurso e o método, ou seja, a especificação do que o cliente de fato deseja. Isto é realizado usando o *Hypertext Transfer Protocol* (HTTP), e são quatro métodos possíveis:

- *GET*: Quando o cliente quer alguma informação que está disponível no servidor.
- *POST*: Utilizado quando o cliente quer salvar alguma informação no servidor.
- *PUT*: Quando já há algum dado salvo no servidor porém o cliente quer fazer alguma alteração nele.
- *DELETE*: Método utilizado para excluir algum dado contido no servidor.

No quesito segurança, a API também tem seus pontos fortes. Ela possui alguns tipos de autenticação de HTTP: através da autenticação básica, ou seja, o cliente informa o usuário e a senha no cabeçalho; a do portador, que é com o uso de um *token*; chaves de API, onde o servidor designa uma valor exclusivo para ser usado como chave a um cliente; e o OAuth, que é a combinação de senha com *tokens* e, por ser uma autenticação de dois fatores, é considerada a mais segura.

Assim como as requisições têm uma estrutura, as respostas também possuem. Elas são compostas por três itens:

- Linha de *status*: Contém a informação se foi bem sucedido ou não a requisição.
- Corpo da mensagem: Onde contém a representação da requisição. Normalmente são no formato JSON ou XML.
- Cabeçalhos: Fornecem mais dados sobre a resposta como codificação, data, tipo de conteúdo entre outros.

A ferramenta estuda acima é de grande valia, mas vale ressaltar que nem sempre é obrigatório seu uso. Um exemplo é a comunicação de uma aplicação com um banco de dados em que existe a possibilidade de fazer uso de *plugins* capazes de configurar o *frontend* para comunicar diretamente com o banco de dados. O fato é que, ao optar por esta opção, acaba deixando informações de login e senha “nas mãos do usuário”.

## 2.5 Plataforma No-Code para aplicações Web

A abordagem *no-code* refere-se a um conjunto de ferramentas que permitem a criação de aplicações sem a necessidade de escrever código tradicional e traz a possibilidade de desenvolver software por meio de interfaces visuais e recursos de arrastar e soltar. O movimento *no-code* é impulsionado pela necessidade de superar a barreira da codificação e permitir que pessoas com diferentes habilidades participem do desenvolvimento de software.

As ferramentas *no-code* surgiram trazendo um leque de oportunidades a pessoas que desejavam desenvolver aplicações mas não tinham tempo ou facilidade em aprender linguagens de programação. Porém vale ressaltar que, juntamente, veio o pensamento de que ela “permite aos usuários o desenvolvimento de aplicações sem o conhecimento técnico em programação” (SANTOS, 2022). Afinal, o que é programação? Ela pode ser definida como “o processo de especificar uma sequência de ações para serem executadas por um computador com o objetivo de resolver um problema ou atingir um determinado resultado” (SEBESTA, 2004). Através da programação, os desenvolvedores podem criar software, aplicativos, jogos e sistemas que automatizam tarefas, processam informações e fornecem soluções tecnológicas para uma ampla variedade de problemas.

As plataformas *no-code* permitem o usuário criar aplicações sem a necessidade de conhecimento em linguagem de programação, porém ainda é necessário ter um conhecimento básico em lógica de programação para que possa produzir programas de forma mais simples. Quanto mais complexo o programa que se deseja desenvolver, maior será a necessidade de ter um conhecimento avançado em lógica e outras tecnologias como APIs e banco de dados.

Existem várias ferramentas *no-code* disponíveis no mercado. A seguir alguns exemplos mais populares:

- Bubble
- Adalo
- Webflow
- Airtable

Dentre as quatro elencadas acima, a plataforma Bubble tornou-se mais atrativa para o desenvolvimento do projeto devido às suas funcionalidades e possibilidades de integração com ferramentas externas.

O Bubble.io oferece uma API chamada *API Conector* que tem a finalidade de conectar as aplicações desenvolvidas na plataforma com as APIs externas, permitindo configuração do cabeçalho, parâmetros e manipulação de respostas das APIs no formato

JSON. As chamadas podem ser realizadas através do próprio servidor da Bubble ou diretamente do navegador. Uma outra ferramenta bastante útil é o plugin *SQL Database Connector*, que permite que o aplicativo se conecte diretamente com o banco de dados sem a necessidade de utilização de uma API RESTful, por exemplo. O grande porém é que o desenvolvedor fica limitado na questão de segurança tendo em vista que ele precisa passar o usuário e a senha para a aplicação para que assim consiga se conectar com o banco de dados externo.

## 3 PROPOSTA

Neste capítulo será apresentada uma solução para abordar a dificuldade da gestão de frotas enfrentada pelo encarregado das viaturas do DTCEA-FL. Visando o problema de credibilidade e esquecimento da ficha de viatura, faz-se necessário o uso de um sistema integrado a tecnologias IoT. A ideia é que o sistema bloqueie a saída de viaturas em caso de “esquecimento”, ou seja, se por algum motivo o militar retornar de uma missão tanto na saída quanto no retorno, ele será identificado no portão, onde o próprio sistema já irá cadastrar os respectivos horários, porém ainda é necessário que o motorista finalize sua missão no sistema. Caso o mesmo não o faça, a viatura utilizada por aquele motorista ficará indisponível no sistema até que a missão seja finalizada.

Para que seja possível o desenvolvimento do projeto se fez necessário criar uma metodologia, visando identificar os problemas a serem solucionados, conhecer as características do cliente, buscar ferramentas, arquitetar a estrutura do sistema juntamente com seus componentes. Tais etapas encontra-se nas seções abaixo no qual tem como propósito detalhar a metodologia a ser empregada.

### 3.1 Levantamento de dados

O levantamento de dados é executado após a definição dos objetivos e requisitos do projeto. Será nele que serão levantados dados sobre a frota de veículos do DTCEA-FL, dados sobre os usuários, processos e procedimentos de gestão utilizados. É com base nessas informações que surgem ideias de soluções para o desenvolvimento do projeto.

#### 3.1.1 Análise de requisitos e casos de uso

Conforme definido, o levantamento ocorreu no mês de março de 2023, onde foram obtidas informações essenciais que influenciaram na definição deste projeto. Para melhor entendimento das necessidades do cliente, foi realizado um estudo de caso no qual foi feito o levantamento de dez requisitos funcionais do sistema:

1. O sistema deve permitir que cada motorista realize a solicitação de saída da viatura, informando os dados pertinentes;
2. O sistema deve conter uma identificação de todos os usuários e veículos;
3. O sistema deve permitir que o sargento de dia autorize ou não as solicitações de saída de viatura dos motoristas;

4. O sistema deve permitir que o encarregado das viaturas impossibilite o uso de viatura quando necessário;
5. O sistema deve permitir que cada motorista realize uma solicitação de reserva;
6. O sistema deve permitir que o encarregado autorize ou não a reserva de viatura feita pelo motorista;
7. O sistema deve permitir que o encarregado/comandante/chefe emita e visualize o relatório de entrada e saída das viaturas;
8. O sistema deve permitir que o guarita consiga verificar se a saída ou entrada da viatura está autorizada;
9. O sistema deve permitir que cada motorista possa cancelar a solicitação de saída da viatura e a reserva;
10. O sistema deve registrar a saída, após a identificação do guarita e validação do mesmo quanto à autorização.

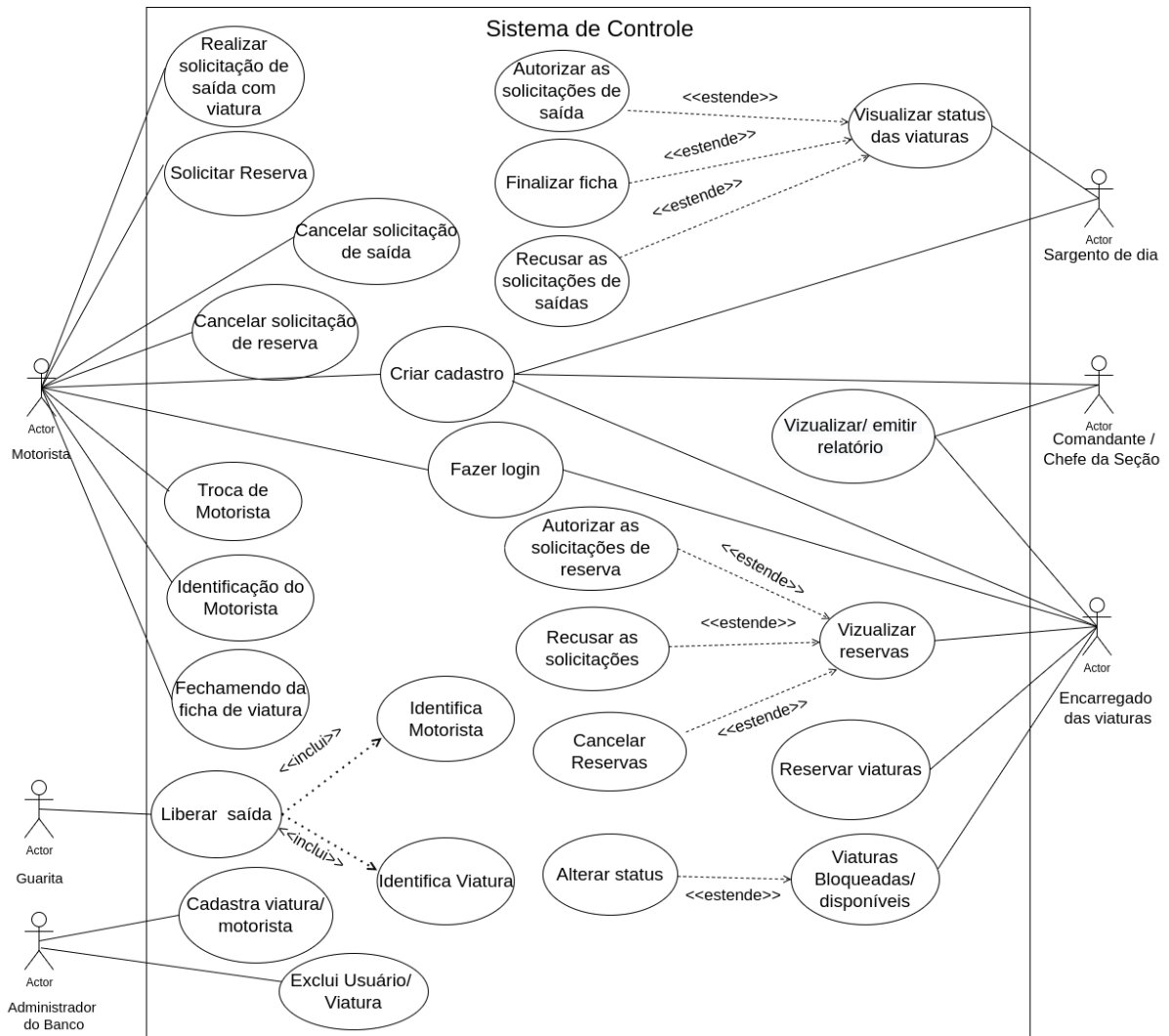
A reserva citada nos requisitos acima é utilizada para casos de manutenções programadas na pista, viagens programadas, ou seja, tem por objetivo auxiliar no planejamento de utilização da viatura bem como impedir que ocorra a utilização da mesma por um motorista não autorizado. Vale ressaltar que, independente da reserva estar autorizada, para se ausentar do destacamento, se faz necessário abrir uma solicitação de saída.

Além dos requisitos funcionais, foram criados regras de negócio para a solicitação, cancelamento, aprovação, reserva, disponibilidade para reserva, identificação do sargento de dia, identificação da viatura, fechamento de ficha, acesso ao relatório, inoperância de viatura, entre outras. Todas essas informações foram utilizadas para a confecção do diagrama de uso conforme [Figura 7](#).

A ideia do diagrama é facilitar a visualização do projeto. Foram criados 5 atores:

- Sargento de dia: Pessoa responsável pelo controle diário das viaturas, o mesmo é responsável por gerenciar solicitações de saída e finalizar as fichas fechadas pelos motoristas;
- Motorista: Pessoa habilitada a conduzir as viaturas. Tem a função de abrir uma solicitação de saída para que, após aprovação do sargento de dia, possa fazer uso da viatura. Abrir uma solicitação de reserva que precisa ser aprovada pelo encarregado. O mesmo pode fazer cancelamento, tanto do pedido de saída de viatura quanto o de reserva, porém no caso da reserva, a mesma depois que aprovada só poderá ser cancelada pelo encarregado. Ele precisa sempre que sair e adentrar ao destacamento se identificar na portaria e, ao término de sua missão, realizar o fechamento da ficha;

Figura 7 – Diagrama de Uso.



Fonte: Do Autor, utilizando Draw.io.

- Encarregado: Pessoa responsável por gerenciar reservas, acessar/emitir relatórios de entrada e saída das viaturas e por bloquear ou disponibilizar um veículo no sistema de modo que possa estar disponível para os motoristas solicitarem seu uso;
- Guarita: Indivíduo responsável pela identificação do motorista e viatura para que, após a consulta no sistema, autorize ou não a entrada ou saída da viatura do destacamento;
- Comandante ou Chefe da Sessão : Pessoa que pode visualizar ou imprimir o relatório de saída e entrada das viaturas;

### 3.1.2 Soluções no mercado

Nesta etapa, inicialmente é realizada uma pesquisa de mercado em busca de soluções já existentes que atendam os objetivos e requisitos do projeto. Na falta deste, será feito

a definição dos componente do sistema, tecnologias empregadas, segurança e integração. Vale ressaltar que já se tem como definido que serão utilizadas tecnologia IoT.

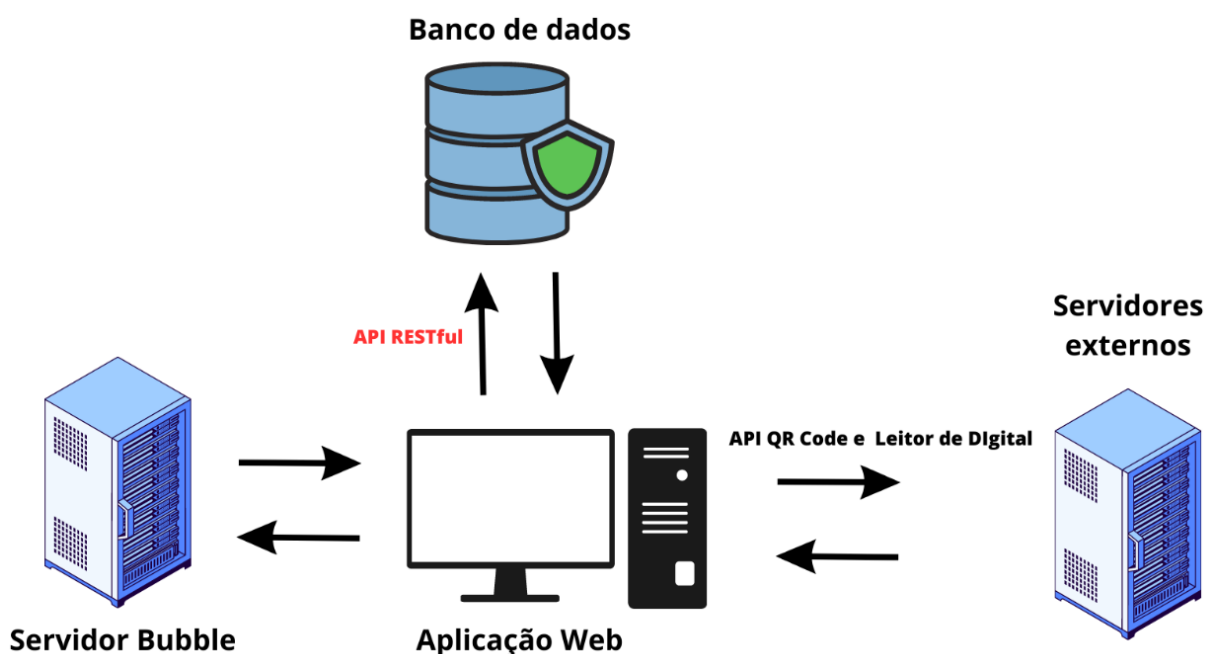
Na Tabela 3 pode-se ser visualizado um resumo da pesquisa de mercado no qual foram estudadas quatro empresas que oferecem o sistema de gestão de frotas na qual cada uma tem suas peculiaridades. Vale ressaltar que todas trabalham com contrato, ou seja, é estabelecido um período de vigência de utilização do *software*.

Tabela 3 – Comparativo de Soluções Tecnológicas em Gestão de Frotas.

Empresas	Cobli	Infleet	Sofit	GeoTab
Rastreamento em tempo real	✓	✓	✓	✓
Gestão de rotas	✓	✓	✓	✓
Monitoramento de consumo de combustível	✓	✓	✓	✓
Controle de manutenção	✓	✓	✓	✓
Solicitação e reserva de veículos			✓	
Identificação do motorista	RFID			NFC
Acompanhamento por câmera	✓	✓		✓

Após análise do dados obtidos na pesquisa, verificou-se que a empresa que mais atenderia os requisitos necessários para os problemas enfrentados pela gestão do DTCEA-FL seria a Sofit, tendo em vista o item solicitação de reserva de veículos. Ao que foi visto dos demais aplicativos, o gestor passava uma rota para o motorista, e o próprio motorista selecionava via aplicativo o carro que ele usaria, pois o grande foco desses aplicativos desenvolvidos tem relação com a gestão de frota mas é mais voltado a empresas que utilizam veículos para entrega, por exemplo. O aplicativo Sofit oferece exatamente dois tipos de solicitações que são necessárias para a gestão atual: solicitação de uso da viatura e solicitação de reserva. Porém no quesito identificação, o mesmo deixou a desejar, pois em nenhum momento é informado como é feita essa certificação de aprovação, como por exemplo, imagine que o motorista João fez uma solicitação para ir ao centro da cidade com um Uno placa X, o gestor recebeu a solicitação e aprovou. Como o rapaz da portaria vai saber que está permitido a saída deste veículo? Supondo que ele tenha acesso ao aplicativo do gestor e ele consiga ver essas informações, ao que parece só constará o nome do motorista e o veículo. Sendo uma empresa de grande porte, como o vigia saberá que de fato o motorista é quem consta na planilha? E é exatamente nesse ponto que o aplicativo deixou de atender um dos requisitos necessários do projeto, que é a identificação do motorista e da viatura na entrada e na saída para que possa ser certificado que existe autorização, bem como registrar a data e horário desses acessos. Com base nessa análise, conclui-se que se faz necessário o desenvolvimento de um software para o atendimento do problema.

Figura 8 – Componentes do Sistema



Fonte: Do Autor, utilizando Canva.

## 3.2 Arquitetura da solução proposta

O sistema será composto por um aplicativo web (*frontend*) desenvolvido na plataforma de desenvolvimento Bubble.io, onde o mesmo fará uso de APIs de QR Code, leitor de digital já disponíveis no mercado e a API RESTful que será implementada para realizar a conexão com o banco de dados de forma segura. Teremos também como componente um banco de dados MySQL (*backend*). Para realizar a leitura do QR Code e da digital será utilizado um celular com acesso à internet. A Figura 8 demonstra como serão as comunicações da aplicação com os demais recursos.

A IoT-A utilizada será a de 4 camadas, onde teremos:

- Camada de Percepção composta pelas tecnologias IoT QR Code e Leitura Digital;
- Camada de Rede: No qual teremos a Wi-fi e o 4G;
- Camada de Middleware: composta pelo banco de dados e API RESTful;
- Camada de Aplicação: composta pelo aplicativo desenvolvido na plataforma Bubble.io.

A escolha por apenas quatro camadas e não cinco se dá pela necessidade do gestor. Caso a ideia da implantação fosse tentar melhorar algum custo relacionado a viaturas, como por exemplo otimização de viagem, aí sim seria válido fazer uso da IoT-A de cinco camadas, incluindo uma camada de análise de dados.



### 3.3 Projeto do Banco de dados

O banco de dados utilizado pela aplicação será o MySQL. Já se tem um escopo das entidades e atributos e os relacionamentos que serão feitos, porém seu desenvolvimento se iniciará no mês de julho. Inicialmente será feito um diagrama ER para posterior implementação do banco de dados e API RESTful na linguagem Java com ferramentas como o Springboot com o Java Persistence API (JPA).

### 3.4 Desenvolvimento do sistema

Após todas as fases anteriores da metodologia terem sido realizadas, ficará faltando apenas o desenvolvimento do sistema. Nesta etapa será utilizada a ferramenta Bubble.io, no qual serão implementadas as interfaces do usuário, tendo que ser considerado perfis diferentes, sargento de dia, encarregado, administrador, identificador e motorista, por exemplo. Para casa usuário teremos acesso a informações diferentes o que exigirá bastante tempo para que sejam pensadas cada uma de suas particularidades. Para facilitar, será elaborado um fluxograma para o desenvolvimento do mesmo. Acredita-se que com esta ferramenta fica mais fácil a visualização do sistema num todo, bem como auxiliará bastante para a implementação.

Assim que a implementação for realizada, serão implementadas as conexões, ou seja, da aplicação com o banco de dados, com as APIs de QR Code e leitor digital. Feito todo esse processo, o aplicativo estará pronto para testes.

Será desenvolvido o sistema de gerenciamento de frotas, bem como a interface de usuário.

### 3.5 Testes e validação

O sistema será testado para garantir que atenda aos requisitos, bem como a verificação da sua integração de software com banco de dados e o dispositivo IoT. Será importante também avaliar o desempenho das ferramentas de auxílio escolhidas para a leitura do QR code e das digitais se a mesmas estão realizando a identificação correta.

### 3.6 Cronograma de Atividades

Para o desenvolvimento do trabalho, foi criado um cronograma no qual algumas etapas já foram realizadas. O cronograma está disponível na [Tabela 4](#).

- Etapa 1: Identificação do problema e definições definições e requisitos do projeto;

- Etapa 2: Levantamento de dados do cliente;
- Etapa 3: Pesquisa de Mercado e escolha da arquitetura do software a ser desenvolvido;
- Etapa 4: Apresentação da proposta para a banca examinadora;
- Etapa 5: Desenvolvimento do Banco de Dados;
- Etapa 6: Implementação da aplicação conexão com as APIs;
- Etapa 7: Testes e resultados;
- Etapa 8: Apresentação final.

Tabela 4 – Cronograma das atividades previstas.

Etapa	2023						
	Março	Junho	Agosto	Setembro	Outubro	Novembro	Dezembro
1	✓						
2	✓						
3	✓	✓					
4		✓					
5			✓				
6				✓	✓		
7						✓	
8							✓

# REFERÊNCIAS

- ALVES, W. P. *Banco de Dados*. [S.l.]: Saraiva, 2014. 18
- Amazon Web Services. *O que é uma API RESTful?* 2023. Internet. Acessado em: [14-Jun-2023]. Disponível em: <<https://aws.amazon.com/pt/what-is/restful-api/>>. 23
- BERTAGLIA, P. R. *Logística e gerenciamento da cadeia de abastecimento*. [S.l.]: Saraiva Educação SA, 2017. 8, 9
- BIOMETRIA. *Biometria Brasil — biometria.com.br*. 2023. Internet. [Accessed 28-Jun-2023]. Disponível em: <<https://biometria.com.br/>>. 16
- CAMPOS, F. C. d.; BELHOT, R. V. Gestão de manutenção de frotas de veículos: uma revisão. *Gestão & Produção*, SciELO Brasil, v. 1, p. 171–188, 1994. 9
- CLEMENTE, Q. K. *Gestão de frota de veículos*. 2013. 9
- COTTI, L. Formulários para administração racional da frota. *Síntese*, São Paulo, 1989. 9
- Denso Wave Inc. *Conceitos Básicos de QR Codes - Introdução aos QR codes*. qr-code-generator.com, 2023. [Accessed 13-Jun-2023]. Disponível em: <<https://br.qr-code-generator.com/qr-code-marketing/qr-codes-basics/>>. 13
- DIGITAL, C. O. *Reconhecimento facial policial da Champion's League errou 92% das vezes — olhardigital.com.br*. 2018. Internet. [Accessed 12-Jun-2023]. Disponível em: <<https://olhardigital.com.br/2018/05/08/seguranca/reconhecimento-facial-policial-confundiu-2300-pessoas-com-potenciais-criminosos/>>. 17
- EDUARDO. *Segurança em banco de dados: com o que se preocupar - netsupport — netsupport.com.br*. 2022. Internet. [Accessed 28-Jun-2023]. Disponível em: <<https://netsupport.com.br/seguranca-em-banco-de-dados/>>. 22
- ELMASRI, R.; NAVATHE, S. *Sistemas de banco de dados*. 6a edição. São Paulo: Person. 837p, p. 24, 2011. 21
- GLUHAK, A. et al. A survey on facilities for experimental internet of things research. *IEEE Communications Magazine*, IEEE, v. 49, n. 11, p. 58–67, 2011. 11
- HEUSER, C. A. *Projeto de banco de dados*. [S.l.]: Grupa A, 2011. 19
- JARA, A. J. et al. Extending the internet of things to the future internet through ipv6 support. *Mobile Information Systems*, IOS Press, v. 10, n. 1, p. 3–17, 2014. 5
- KIM, W. Object-oriented databases: definition and research directions. *IEEE Transactions on Knowledge and Data Engineering*, v. 2, n. 3, p. 327–341, 1990. 19
- LOUREIRO, G. da S. M. et al. *RFID: Tecnologia - gta.ufrj.br*. 2020. Internet. [Accessed 28-Jun-2023]. Disponível em: <[https://www.gta.ufrj.br/grad/15\\_1/rfid/tecnologia.html](https://www.gta.ufrj.br/grad/15_1/rfid/tecnologia.html)>. 16

- MACHADO, F. T. d. S. et al. Um processo para extração de esquemas conceituais em fontes de dados json baseado em técnicas de similaridade de texto. Universidade Federal de Santa Maria, 2017. 19
- MASCHIETTO, L. G. et al. *Arquitetura e Infraestrutura de IoT*. Grupo A, 2021. Internet. [Accessed 12-Jun-2023]. Disponível em: <<https://app.minhabiblioteca.com.br/#/books/9786556901947/>>. 11, 12
- PICHETTI, R. F.; VIDA, E. S.; CORTES, V. S. M. P. *Banco de Dados*. [S.l.]: Grupa A, 2021. 22
- SALLES, S. R. d. Gestão estratégica de frota leve: o caso de são josé dos campos-sp. Universidade Tecnológica Federal do Paraná, 2011. 9
- SANTOS, B. P. et al. Internet das coisas: da teoria à prática. *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, v. 31, p. 16, 2016. 10
- SANTOS, G. M. d. Estudo do uso de plataformas no-code para geração de mvps. 2022. 25
- SATYABRATA. *Arquitetura de 5 camadas da Internet das coisas*. 2022. Internet. [Accessed 18-Jun-2023]. Disponível em: <<https://acervolima.com/arquitetura-de-5-camadas-da-internet-das-coisas/>>. 10
- SEBESTA, R. W. *Concepts of programming languages*. [S.l.]: Pearson Education India, 2004. 25
- SOLIDITGMBH. *DB-Engines Ranking — db-engines.com*. 2023. Internet. [Accessed 11-Jun-2023]. Disponível em: <<https://db-engines.com/en/ranking>>. 20, 21
- TELECO. *RFID: Faixas de Frequências e Protocolos*. 2023. Internet. [Accessed 28-Jun-2023]. Disponível em: <[https://www.teleco.com.br/tutoriais/tutorialrfid/pagina\\_3.asp](https://www.teleco.com.br/tutoriais/tutorialrfid/pagina_3.asp)>. 15
- UVIASE, O.; KOTONYA, G. Iot architectural framework: connection and integration framework for iot systems. *arXiv preprint arXiv:1803.04780*, 2018. 10
- VALADÃO, R. B. *Biometria - Reconhecimento de íris — gta.ufrj.br*. 2008. Internet. [Accessed 12-Jun-2023]. Disponível em: <[https://www.gta.ufrj.br/grad/08\\_1/iris/index.html](https://www.gta.ufrj.br/grad/08_1/iris/index.html)>. 17
- VALENTE, A. M.; PASSAGLIA, E.; NOVAES, A. G. *Gerenciamento de transporte e frotas*. [S.l.]: Pioneira, 1997. 8, 9
- WADE, B. W.; CHAMBERLIN, D. D. Ibm relational database systems: The early years. *IEEE Annals of the History of Computing*, v. 34, n. 4, p. 38–48, 2012. 18
- WIKIPEDIA. *Código QR – Wikipédia, a enciclopédia livre*. 2022. Internet. [Accessed 28-Jun-2023]. Disponível em: <[https://pt.wikipedia.org/wiki/C%C3%B3digo\\_QR](https://pt.wikipedia.org/wiki/C%C3%B3digo_QR)>. 13