

UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação



Dissertação

**Proposta de Framework para o Desenvolvimento de Aplicações IoT no contexto
de Campus Inteligente**

Guilherme Wollmann de Oliveira

Pelotas, 2023

Guilherme Wollmann de Oliveira

Proposta de Framework para o Desenvolvimento de Aplicações IoT no contexto de Campus Inteligente

Dissertação apresentada ao Programa de Pós-Graduação em Computação do Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Mestre em Engenharia de Computação.

Orientador: Prof. Dr. Júlio Carlos Balzano de Mattos
Coorientador: Prof. Dr. Rafael Iankowski Soares

Pelotas, 2023

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

O48p Oliveira, Guilherme Wollmann de

Proposta de framework para o desenvolvimento de aplicações IOT no contexto de campus inteligente / Guilherme Wollmann de Oliveira ; Júlio Carlos Balzano de Mattos, orientador ; Rafael Iankowski Soares, coorientador. — Pelotas, 2023.

84 f.

Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2023.

1. Campus inteligente. 2. Framework. 3. IOT. I. Mattos, Júlio Carlos Balzano de, orient. II. Soares, Rafael Iankowski, coorient. III. Título.

CDD : 005

Guilherme Wollmann de Oliveira

Proposta de Framework para o Desenvolvimento de Aplicações IoT no contexto de Campus Inteligente

Dissertação aprovada, como requisito parcial, para obtenção do grau de Mestre em Engenharia de Computação, Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

Data da Defesa: 31 de março de 2023

Banca Examinadora:

Prof. Dr. Júlio Carlos Balzano de Mattos (orientador)

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Rafael Iankowski Soares (Co-orientador)

Doutor em Computação pela Pontifícia Universidade Católica do Rio Grande do Sul.

Prof. Dr. Adenauer Correa Yamim

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Guilherme Tomaschewki Netto

Doutor em Oceanologia pela Universidade Federal do Rio Grande.

Prof. Dr. Mário Seiji Oyamada

Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

AGRADECIMENTOS

Ao meu orientador Prof. Dr. Júlio Carlos Balzano de Mattos e co-orientador Prof. Dr. Rafael Iankowski Soares pela orientação, competência, profissionalismo e dedicação tão importantes. Agradeço por todos os ensinamentos compartilhados de forma admirável, e por me guiarem na realização deste trabalho.

Aos os colegas Arthur Cardozo Godinho e Yuri Vaz pela ajuda na realização da expansão do estudo de caso, sem o apoio de vocês para a implementação esta ampliação não teria sido realizada, então muito obrigado.

A minha família pelo apoio incondicional que sempre me foi dado, além de todo incentivo e suporte para que sempre pudesse seguir estudando, eles sempre serviram de inspiração para mim e só tenho a agradecer-los por tudo que me foi dado e proporcionado.

A minha companheira Andreza Borchardt a qual se faz presente em minha vida desde antes mesmo do início de minha trajetória acadêmica. Muito obrigado por todo carinho, apoio, paciência e atenção, por sempre estar disposta a ajudar e servir de motivação para que eu continuasse. Que possamos seguir pelo resto da vida nos apoiando nas dificuldades e partilhando as conquistas.

RESUMO

OLIVEIRA, Guilherme Wollmann de. **Proposta de Framework para o Desenvolvimento de Aplicações IoT no contexto de Campus Inteligente**. Orientador: Júlio Carlos Balzano de Mattos. 2023. 84 f. Dissertação (Mestrado em Engenharia de Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2023.

Devido à constante evolução de tecnologias que permitem que objetos se comuniquem por meio da Internet, constituindo a Internet das Coisas (do inglês, *Internet of Things* - IoT), é possível conceber sistemas de monitoramento capazes de auxiliar na redução de desperdícios de recursos tais como energia elétrica, água e insumos em geral. O uso de IoT com o auxílio de tecnologias computacionais possibilita o projeto de sistemas capazes de obter um melhor aproveitamento dos recursos disponíveis em um ambiente, seja ele uma casa, prédio ou mesmo um campus universitário. Além disso, sistemas desta natureza possibilitam propiciar melhores experiências aos usuários, de modo a produzir ambientes conhecidos como *smart* dando origem às casas inteligentes, às construções inteligentes e também aos campus inteligentes. O conceito de Campus Inteligente surge de uma grande variedade de serviços oferecidos para os usuários com a necessidade de reduzir custos e fornecer serviços de alta qualidade não limitados aos aspectos acadêmicos, mas também social, financeiro e ambiental. Este trabalho propõe um Framework que permita construir uma infraestrutura de Hardware/Software para aplicações IoT no contexto de Campus Inteligente, a infraestrutura do Framework é composta por 5 camadas, onde para cada uma foram definidas tecnologias e quais seriam suas responsabilidades dentro do sistema, assim como quais seriam os padrões de comunicação entre as camadas. Este Framework foi concebido com características de escalabilidade e que ofereça suporte às mais variadas aplicações e serviços no contexto de Campus Inteligente. Por fim, de modo a verificar o que foi proposto, é realizado um estudo de caso no campus Anglo da UFPel com base no Framework proposto e então apresentados os resultados obtidos com esta implementação.

Palavras-chave: Campus Inteligente. Framework. IoT.

ABSTRACT

OLIVEIRA, Guilherme Wollmann de. **Proposal of a Framework for the Development of IoT Applications in the context of the Intelligent Campus**. Advisor: Júlio Carlos Balzano de Mattos. 2023. 84 f. Dissertation (Masters in Computer Engineering) – Technology Development Center, Federal University of Pelotas, Pelotas, 2023.

Due to the constant evolution of technologies that allow objects to communicate through the Internet, constituting something known as the Internet of Things (IoT), it is possible to design monitoring systems capable of helping reduce the waste of resources such as electricity, water, and supplies. Using IoT with the help of computational technologies makes it possible to design systems capable of better using the resources available in an environment, be it a house, building, or even a university campus. In addition, systems of this nature make it possible to provide better user experiences in order to produce environments known as *smart*, giving rise to *smart homes*, *smart buildings* and also to *Smart Campus*. The Intelligent Campus concept arises from a wide variety of services offered to users who need to reduce costs and provide quality services, not only to academics but also to social, financial, and environmental services. This work proposes a Framework that allows building a Hardware/Software infrastructure for IoT applications in the context of an Intelligent Campus. The Framework infrastructure is composed of 5 layers, where technologies were defined for each one and their responsibilities within the system, as well as the communication patterns between the layers. It is also desirable that this Framework has scalability and low-cost features. The objective is that the developed work supports the most varied applications and services in the context of an Intelligent Campus. Finally, to verify what was proposed, a case study is carried out on the Anglo campus of UFPel based on the proposed Framework, and then the results obtained with this implementation are presented.

Keywords: Smart Campus. Framework. IoT.

LISTA DE FIGURAS

Figura 1	Número de dispositivos IoT ao longo dos anos, segundo (STATE OF IOT 2022: NUMBER OF CONNECTED IOT DEVICES GROWING 18% TO 14.4 BILLION GLOBALLY, 2022).	14
Figura 2	Arquitetura genérica de um sistema IoT e suas três dimensões, segundo (ALI; ALI; BADAWY, 2015).	19
Figura 3	Categorização dos requisitos de Computação em Nuvem.	20
Figura 4	Arquitetura básica de computação em borda, segundo (YU et al., 2017).	22
Figura 5	Arquitetura típica de computação em borda, segundo (YU et al., 2017).	22
Figura 6	Demonstração de um fluxo de dados no Apache Nifi, segundo (GUPTA, 2022).	25
Figura 7	Visualização em formato de nuvem de palavras de termos chave presentes em resumos de artigos relacionados a Big Data, segundo (DE MAURO; GRECO; GRIMALDI, 2015).	27
Figura 8	Arquitetura Campus Inteligente, segundo (VILLEGAS-CH et al., 2019).	28
Figura 9	Componentes do Framework, segundo (NG et al., 2010).	29
Figura 10	Níveis de Inteligência do Framework, segundo (MOHAMMADI; AL-FUQAHA, 2018)	30
Figura 11	Fluxo do Mapeamento Sistemático, segundo (PETERSEN et al., 2008)	32
Figura 12	Diagrama das <i>strings</i> de busca, segundo (KITCHENHAM et al., 2010).	33
Figura 13	Distribuição do número de artigos por ano.	35
Figura 14	Camadas do Framework proposto.	44
Figura 15	Exemplos de dispositivos da camada de Infraestrutura	46
Figura 16	Exemplo de Arquivo JSON	47
Figura 17	Casos de uso de armazenamento.	50
Figura 18	Camadas do Framework com suas tecnologias.	55
Figura 19	Formato base de arquivo JSON	60
Figura 20	Exemplo de uso prático de arquivo JSON	60
Figura 21	Diagrama do Estudo de Caso	64
Figura 22	Grove Light Sensor	64
Figura 23	NodeMCU V3 Lolin - ESP8266	65
Figura 24	Colunas do banco de dados	65
Figura 25	Dados carregados no Weka	67
Figura 26	Atributos dos dados	67

Figura 27	Atributos dos dados	68
Figura 28	Distribuição dos valores do atributo	68
Figura 29	Frontend Framework	69
Figura 30	Lista de Campus Frontend	70
Figura 31	Informações gerais dos dados	70
Figura 32	Lista de sensores Frontend	71
Figura 33	Placa do estudo de caso	71
Figura 34	Local da coleta de dados	72
Figura 35	ESP32S Node MCU	72
Figura 36	Grove Mini PIR Motion Sensor	73
Figura 37	Grove Loudness Sensor	73
Figura 38	Instalação do nó sensor - Sensor de Movimento	75
Figura 39	Instalação do nó sensor - Sensor de Luminosidade e Sonoro	76

LISTA DE TABELAS

Tabela 1	Modelo PICO aplicado as <i>Research Questions</i>	33
Tabela 2	Critérios de Inclusão e Exclusão	34
Tabela 3	Número de trabalhos por etapa de seleção	34
Tabela 4	Classificação dos trabalhos em tópicos	37
Tabela 5	Tecnologias utilizadas com maior frequência nos trabalhos selecionados	41
Tabela 6	Comparativo de Framework e suas tecnologias	57
Tabela 7	Informações sobre os dados coletados	74

LISTA DE ABREVIATURAS E SIGLAS

RQ	Research Question
IoT	Internet of Things
TIC	Tecnologias da Informação e Comunicação
JSON	JavaScript Object Notation
ETL	Extract Transform Load
SQL	Structured Query Language
HDFS	Hadoop Distributed File System
SaaS	Software como Serviço
IaaS	Infraestrutura como Serviço
PaaS	Plataforma como Serviço
TIC	Tecnologia da informação e comunicação

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos do Trabalho	16
1.2	Organização do Texto	17
2	REFERENCIAL TEÓRICO	18
2.1	Internet das Coisas	18
2.2	Computação em Nuvem	19
2.3	Computação em Borda	20
2.4	MQTT Broker	21
2.5	LoRaWAN	23
2.6	Apache Nifi	24
2.7	Big Data	25
2.8	Campus Inteligente	27
2.9	Framework	28
3	REVISÃO BIBLIOGRÁFICA	31
3.1	Metodologia	31
3.2	Discussão das Questões de Pesquisa (RQs)	35
3.3	Considerações	42
4	PROPOSTA DE FRAMEWORK PARA O DESENVOLVIMENTO DE APLICAÇÕES IOT	43
4.1	Detalhamento do Framework Proposto	43
4.1.1	Infraestrutura	45
4.1.2	Comunicação	48
4.1.3	Armazenamento	50
4.1.4	Processamento	51
4.1.5	Serviços	52
4.2	Considerações sobre o Framework Proposto e Comparativos	54
5	RESULTADOS	62
5.1	Estudo de Caso	62
5.2	Protótipo Desenvolvido	63
6	CONCLUSÕES E TRABALHOS FUTUROS	77
	REFERÊNCIAS	79

1 INTRODUÇÃO

Em uma instituição de ensino, e principalmente pública, um desafio constante para seus gestores é otimizar o consumo de seus recursos a fim de aproveitá-los ao máximo possível. Um dos caminhos para se obter um menor desperdício no uso destes recursos é tornar o ambiente de ensino em um Campus Inteligente. De acordo com (VILLEGAS-CH et al., 2019) muitos campus universitários reconhecem que a integração de tecnologias é o caminho para se obter uma transformação digital. Muitos destes campus trabalham de modo a consolidar as tecnologias de informação e comunicação (TIC) em sua base, tornando possível considerá-los como campus inteligentes (BORAN et al., 2011). Algumas das características de um Campus Inteligente são a criação de um ambiente confortável, sustentável e seguro, que eficientemente consegue contribuir para o aprendizado e a gerência de questões administrativas (AION et al., 2012). De modo a atender estas características, as TICs incluem sistemas de aquisição de dados, computação em nuvem e análise de dados em Campus Inteligentes (NIE, 2013). Ao integrar estas tecnologias torna-se viável o processo de tornar o campus inteligente, dando a este a capacidade de atender com maior eficiência aos desafios de gestão neste ambiente.

Uma dentre as possíveis implementações das TICs é o uso de tecnologias integradas a internet, constituindo a Internet das Coisas (do inglês, *Internet of Things* - IoT), que estão presentes em vários momentos de nosso dia-a-dia, como por exemplo redes de sensores monitorando ambientes, e a partir dos dados coletados e computados localmente e/ou em nuvem, tomando ações de forma autônoma em tempo real de modo a atuar diretamente no ambiente em questão. De acordo com (SARI; CIP-TADI; HARDYANTO, 2017) o conceito de Campus Inteligente já vem sendo moldado em países desenvolvidos a alguns anos, e este é uma tendência na área de IoT, pois ainda de acordo com esse trabalho, o conceito de se construir um Campus Inteligente implica em que a instituição irá utilizar implementações de TICs para monitorar e controlar de forma automática cada área do campus. Além disto é ainda mencionado que IoT tem sido amplamente utilizada na construção de Casas Inteligentes (DING et al., 2011), Campus Inteligente (MIN-ALLAH; ALRASHED, 2020), Construções Inteligen-

tes (AL DAKHEEL et al., 2020) e Cidade Inteligente (CAMERO; ALBA, 2019). Por fim, na Figura 1 é possível observar o crescimento de dispositivos IoT ao longo dos anos nas mais variadas áreas, os dados do gráfico se estendem até o quarto trimestre de 2021 e a partir deste ponto os valores são projeções da quantidade de dispositivos IoT para os próximos anos. É notável o grande crescimento que esta área vem apresentando e segundo especialistas pode-se esperar cada vez mais uso de dispositivos dessa natureza em nosso cotidiano.

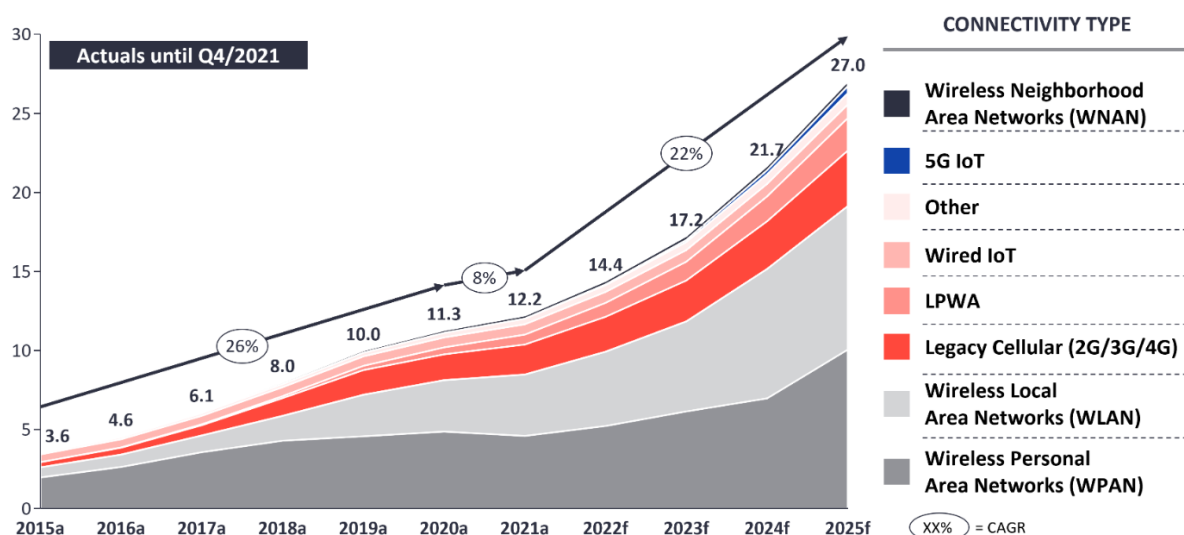


Figura 1 – Número de dispositivos IoT ao longo dos anos, segundo (STATE OF IOT 2022: NUMBER OF CONNECTED IOT DEVICES GROWING 18% TO 14.4 BILLION GLOBALLY, 2022).

Portanto, se faz necessário refletir na construção do Campus Inteligente tendo como base o uso de IoT como uma parte central do processo de tornar o campus inteligente, com isso surge a motivação para a construção de um Framework que facilite a organização dessa estrutura com vários dispositivos IoT, assim como permita a ágil adição de novos componentes. Porém, para que tal objetivo seja possível é importante considerar os desafios de implementar um sistema deste porte. No trabalho feito em (BERROUYNE et al., 2019) são feitas considerações a respeito dos desafios de implementar um sistema com inúmeros e variados dispositivos IoT. Neste trabalho são descritas as potenciais dificuldades de integrar estes elementos IoT ao sistema, dada a heterogeneidade destes dispositivos, as variadas plataformas de computação que podem estar conectadas ao sistema, os diferentes protocolos de comunicação que estes dispositivos utilizam e as linguagens que devem ser utilizadas para configura-los, além disso, também é preciso levar em conta a complexidade para inserir/retirar um elemento do sistema. Também é importante considerar que para que estes elementos possam ser integrados ao sistema, configurados corretamente e tenham suporte quando necessário, é preciso de uma mão de obra especializada, como pessoas capacitadas para realizar calibração de sensores, integração destes a sistemas de com-

putação, configuração de servidores capazes de armazenar e computar estes dados, entre outros e quanto maior for a heterogeneidade dos sensores e elementos que compõem o sistema, maior a demanda de diferentes áreas e pessoas especializadas. Uma possível solução para buscar reduzir essa complexidade gerada pela heterogeneidade de todos os componentes do sistema seria abstrair o sistema, dividindo este em camadas também abstratas, de modo a simplificar o sistema em sua totalidade, o que acaba por motivar a construção de um Framework que sirva de apoio para um projeto baseado em um sistema IoT.

Na literatura é possível encontrar alguns trabalhos que desenvolveram Frameworks, porém cada um para um determinado contexto. Em (VAKALLOUDIS; O'LEARY, 2019) é proposto um Framework para ambientes industriais, onde dois estudos de caso são apresentados para este ambiente. Este Framework é dividido em cinco camadas, estrutura esta que motiva seu uso devido a sua organização em diferentes níveis de abstração, porém trata-se de um projeto que possui uma limitação para seu uso, neste trabalho são suportadas implementações de pequeno a médio porte, sendo escalabilidade um problema presente, problema tal que é levado em consideração no desenvolvimento do Framework para Smart Campus da UFPel.

Já em (YELMARTHI; ABDELGAWAD; KHATTAB, 2016) o Framework proposto visa aplicações IoT de baixo consumo e baixo custo e que seja confiável. A eficiência deste Framework foi comprovada através de implementações de diversas aplicações, principalmente relacionadas a área da saúde. O baixo consumo é um dos pontos de interesse para esta proposta de dissertação. Verifica-se no trabalho citado que o baixo consumo não prejudica o desempenho do sistema ligado ao Framework, porém é importante observar que para realizar a análise dos dados coletados é utilizada uma Raspberry Pi 2, caso seja necessário aumentar a quantidade de sensores atrelados ao dispositivo pode acabar gerando problemas de desempenho.

O trabalho proposto por (WANG et al., 2017) se aproxima bastante do que foi planejado para desenvolver nesta dissertação, ou seja, um Framework para Campus Inteligente com base em conceitos IoT, que possua características de redução no consumo de energia, segurança e proteção em uma perspectiva centrada no usuário. Um ponto importante deste trabalho citado é a preocupação em como é usado o espaço em questão. Do ponto de vista do consumo energético, o sistema trabalha com a capacidade de perceber, identificar, avaliar e reagir aos comportamentos. Na prática, o sistema é capaz de identificar se um certo ambiente está sendo utilizado em dado momento ou não, para com isso possa ligar/desligar as luzes e demais equipamentos ali inseridos.

Em (BARBATO et al., 2015) o sistema projetado junto a um Framework captura dados de vários ambientes utilizando sensores e os armazena em um repositório central que mantém todas as informações relevantes. Este é um ponto importante a ser

tratado na dissertação, principalmente devido ao fato das unidades da UFPel serem geograficamente separadas. Com estas informações é possível que aplicações possam usar tais informações para a criação de modelos ou conjunto de regras para inferir novas informações a partir de algoritmos de aprendizado de máquina, oferecendo serviços inteligentes aos gestores da universidade. Porém o trabalho feito em (BARBATO et al., 2015) não apresenta uma definição de modelo genérico para comunicação dos sensores com o Framework, o que pode gerar uma grande complexidade devido a heterogeneidade de dispositivos IoT que podem estar presentes.

Um dos problemas presentes em um Campus Inteligente que é apontado em (VILLEGAS-CH et al., 2019) trata da dificuldade de estabelecer políticas e implementar aplicações que façam um bom controle do consumo dos recursos energéticos do campus, isso se dá principalmente devido ao grande custo para se adquirir equipamentos capazes de integrar todos os equipamentos em um único sistema de monitoramento. Ainda de acordo com este trabalho uma resposta adequada para isso seria a implementação de um *Big Data Framework* que seria capaz de monitorar as condições de todos os dispositivos, para tal o Framework deve incluir versatilidade e escalabilidade para se adaptar a qualquer campus, independente da infraestrutura utilizada.

1.1 Objetivos do Trabalho

Este trabalho tem o objetivo principal de projetar um Framework que ofereça suporte a implementação de Hardware/Software para aplicações IoT no contexto de Campus Inteligente, de modo que este por sua vez seja escalável e que ofereça suporte a diversas aplicações no contexto de um campus inteligente e os demais serviços que sejam viáveis e pertinentes a um Campus Inteligente. A infraestrutura projetada deve desempenhar um papel de *middleware*, entre os sensores com seus respectivos hardwares necessários para se fazer a obtenção de dados, e um sistema central responsável por prover inteligência e recomendações aos gestores, a partir dos dados coletados. Além disso, idealmente este Framework deve oportunizar a construção de sistemas com características de escalabilidade e independência de tecnologia.

Para que tais objetivos sejam alcançados durante o desenvolvimento deste trabalho buscou-se conhecer detalhadamente as camadas de abstração de um Framework para sistemas IoT, as tecnologias que poderiam ser utilizadas em cada camada, como estas poderiam ser integradas, assim como suas vantagens e desvantagens em relação as outras opções presentes, além de buscar uma forma de validar um sistema tão complexo quanto um Framework para Campus Inteligente de forma mais simples que fosse viável em questão de tempo de desenvolvimento.

Além das características previamente revisadas, que são desejadas para o Framework aqui proposto, também é idealizado que o Framework seja capaz de suportar

diversas aplicações tais como Transporte Inteligente, Banco de Dados Meteorológicos Local Inteligente, Vigilância Inteligente e o que mais for pertinente e viável ao contexto de campus da UFPel. Como estudo de caso deste trabalho foi desenvolvido um pequeno sistema de sensores que monitoram o uso do ambiente em questão para o controle do consumo de energia, também será feita uma busca por levar em consideração a luz natural que já está presente, de modo que, mesmo havendo usuários naquele local, seja possível que não haja a necessidade do uso de luzes elétricas, de modo a tornar o sistema ainda mais eficiente energeticamente. Esse sistema integrado ao Framework capaz de monitorar o uso dos ambientes com a preocupação do consumo energético, também pode levantar dados com relação à questão de ocupação do local. Uma preocupação recorrente é a demanda de salas para a baixa quantidade disponível de locais, parte-se do princípio de que seja possível identificar locais que estão sendo sub utilizados, de modo a fazer uma alocação mais apropriada para os mesmos.

1.2 Organização do Texto

O texto da dissertação foi organizada da seguinte forma: o Capítulo 2 trata do referencial teórico, onde são apresentadas todas as tecnologias e conceitos que serão utilizados e/ou mencionados no decorrer dos demais capítulos. Já o Capítulo 3 descreve detalhadamente a revisão bibliográfica que foi realizada. Após isto o Capítulo 4 apresenta todo desenvolvimento deste trabalho, cada camada do Framework é apresentada de forma detalhada, tanto do ponto de vista estrutural quanto tecnológico, descrevendo quais tecnologias estão presentes em cada camada e como estas se integram com o resto do sistema. No Capítulo 5 são apresentados os resultados obtidos com o estudo de caso realizado e por fim no Capítulo 6 as conclusões sobre o trabalho, assim como são descritos possíveis trabalhos futuros são apresentados.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta de forma resumida o referencial teórico do trabalho contendo as tecnologias e conceitos que foram utilizados no desenvolvimento do trabalho e discutidas no decorrer do texto.

2.1 Internet das Coisas

De acordo com (ALI; ALI; BADAWEY, 2015) a Internet das Coisas (do inglês, *Internet of Things* - IoT), surge como uma nova tecnologia que é utilizada para expressar uma rede moderna de conexões sem fio. Apesar da definição variar conforme o contexto, formalmente IoT pode ser definido como uma infraestrutura de rede dinâmica global com autoconfiguração e comunicação interoperável. De outra forma também pode ser dito que IoT é a habilidade de conectar tudo ao nosso redor, desde a máquinas, quaisquer dispositivos, celulares, carros, entre outros, até mesmo cidades e estradas a Internet, com um comportamento inteligente e levando em consideração a existência dos conceitos de autonomia e privacidade. Este trabalho apresenta na Figura 2 a arquitetura de um sistema IoT mostrando como pode ser dividida em três dimensões principais que podem ser utilizadas para classificar qualquer trabalho relacionado ao tema. A intersecção destas três dimensões dá origem ao que podemos chamar de infraestrutura IoT.

O trabalho realizado por (LIU; BAIOCCHI, 2016) sugere duas definições para IoT, que variam conforme a complexidade do sistema. Para sistemas de baixa complexidade a definição pode ser expressa como IoT sendo uma rede que conecta *coisas* únicas e identificáveis que são capazes de monitorar/atuar e potencialmente possuem capacidades de serem programáveis. Através dessa capacidade de monitorar é possível captar o estado dessa *coisa* e este pode ser alterado a qualquer momento, de qualquer lugar por qualquer outra *coisa*. Para sistemas de maior complexidade a definição trata de IoT como uma rede complexa, adaptativa e autoconfigurável que interconecta *coisas* a Internet através de protocolos padrões de comunicação. Essas *coisas* tem representação física ou virtual no mundo digital, capacidade de monito-

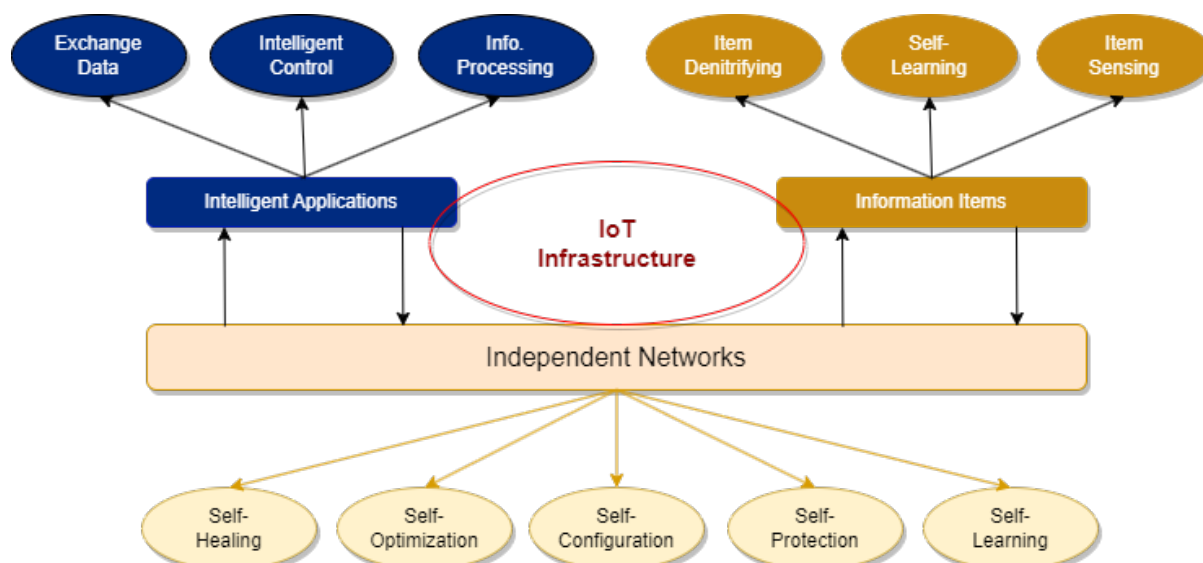


Figura 2 – Arquitetura genérica de um sistema IoT e suas três dimensões, segundo (ALI; ALI; BADAWEY, 2015).

rar/atuar, são programáveis e possuem identificação única. A representação contém informações tais como, a identificação da *coisas*, status, localização e qualquer outra informação relevante. Tais *coisas* oferecem serviços com/sem a interação humana, através da captura de dados, comunicação e capacidade de atuação, este serviço é explorado devido ao uso de interfaces inteligentes e está disponível em qualquer lugar, a qualquer hora, e para qualquer um levando em consideração a segurança envolvida.

De forma geral as definições concordam implicitamente que IoT tem quatro pilares fundamentais: capacidade de monitorar, atuar, computar e comunicar, sendo assim um sistema IoT pode ser visto como um conjunto de sensores/atuadores. Os sensores permitem que o sistema tome conhecimento de eventos acontecendo nos mais variados ambientes, estas informações são processadas e então armazenadas em bancos de dados locais e/ou na nuvem, conforme a necessidade do sistema. Além disto, o sistema também é capaz de atuar diretamente no ambiente através de dispositivos atuadores, utilizando os dados coletados para que decisões sobre a atuação dos elementos seja feita.

2.2 Computação em Nuvem

De acordo com o trabalho apresentado em (TAURION, 2009) não existe uma única e precisa definição para *Cloud Computing*, porém uma das formas de descrevê-la é como um conjunto de recursos disponibilizados na internet, tais como, capacidade de processamento, armazenamento, conectividade, plataformas, aplicações e serviços. De modo geral, a computação em nuvem pode ser vista como uma virtualização de um centro de processamento de dados (em inglês, *data center*).

O trabalho feito em (ARUMUGAM et al., 2010) classifica a computação em nu-

vem conforme a disponibilidade de recursos em três diferentes tipos de serviços. O primeiro destes é fazer com que a infraestrutura de hardware seja oferecida como serviço, este é chamado de *Infrastructure as a Service (IaaS)*. Um exemplo deste tipo seria o Amazon EC3/S3. O segundo tipo consiste em prover uma plataforma, ou seja, o sistema operacional juntamente com os softwares necessários para o usuário. Essa forma de *Cloud Computing* leva o nome de *Plataform as a Service (PaaS)* tendo como exemplo prático deste tipo o *Google App Engine*. Por fim, o terceiro e último tipo prove a aplicação como um serviço, juntamente com o hardware necessário. Este tipo de serviço é chamado de *Software as a Service (SaaS)*, onde o Google Docs é tido como um exemplo.

Outro fator a ser considerado ao utilizar *Cloud Computing* são os requisitos funcionais e não funcionais que o usuário necessita. O trabalho feito em (ALHAMAD; DILLON; CHANG, 2010) destaca como requisitos não funcionais: escalabilidade, tempo de resposta e disponibilidade. Os requisitos funcionais, por sua vez, definem o sistema e seus componentes, estes ajudam a descrever o comportamento desejado do sistema e suas funções, por exemplo, o que se espera de saída a partir de uma certa entrada. Ainda neste trabalho é ilustrado na Figura 3 como são classificados os serviços em nuvem de acordo com os requisitos do ponto de vista do usuário.

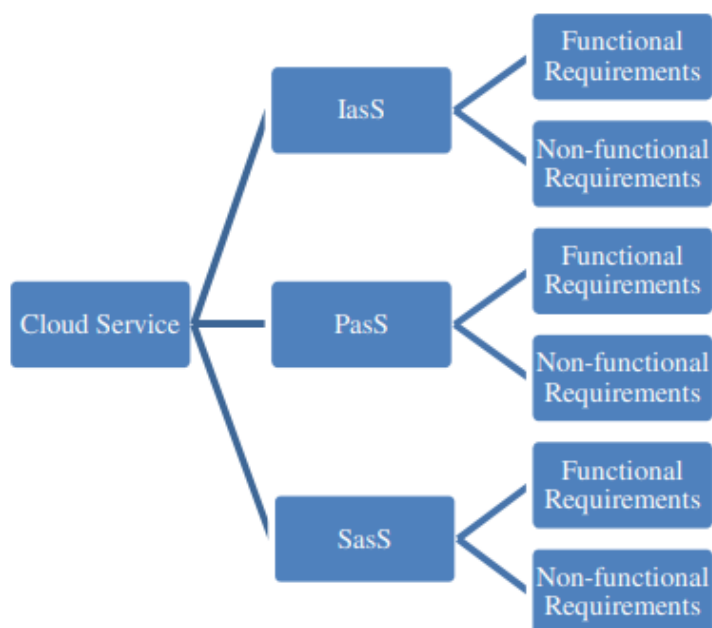


Figura 3 – Categorização dos requisitos de Computação em Nuvem.

2.3 Computação em Borda

Milhares de dispositivos estão a todo momento produzindo dados e efetuando trocas de mensagens através de complexas redes de comunicação, de modo a mitigar o efeito de congestionamento que é gerado por essa comunicação em massa de envio

e recebimento de dados foi criada a computação em borda. De acordo com o trabalho feito em (YU et al., 2017) a computação em borda, diferente da computação em nuvem, migra o armazenamento e a computação dos dados para as camadas mais próximas aos usuários. Logo, com um certo número de nós capazes de realizarem esse tratamento dos dados, antes realizados em nuvem, distribuídos ao longo da rede de dispositivos, pode diminuir a imensa carga computacional que antes ficava atribuída somente a base de dados centralizada. Além disso, esta estrutura distribuída pode balancear o tráfego de dados evitando picos no fluxo nas redes IoT, diminuindo o tempo de transmissão dos dados da rede para com a nuvem e com os usuários.

Ainda, (YU et al., 2017) apresenta uma estrutura básica de um sistema IoT de computação em borda onde é possível observar na Figura 4 que os serviços de computação estão mais próximos dos dispositivos dos usuários que dos servidores em nuvem, mesmo estes tendo um poder menor de computação que os dispositivos em nuvem ainda conseguem prover um serviço de melhor qualidade e menor tempo de latência para os usuários da aplicação.

De maneira geral a computação em borda pode ser dividida em três camadas: a de borda, do meio e a mais distante do usuário, a Figura 5 mostra uma arquitetura típica de um sistema de computação em borda de modo a ilustrar isto. A camada de borda é composta por dispositivos como sensores, atuadores, entre outros, esta camada promove maior interação e responsividade ao usuário, com a computação dos dispositivos próximos a essa camada é possível a prestação de serviços de aplicação em tempo real para os usuários. Os dispositivos da camada do meio dão suporte a grande parte do tráfego de dados presente na rede, na computação em borda a maioria da computação dos dados esta presente nesta camada, deste modo provendo para os usuários um melhor desempenho e menor latência na computação e armazenamento dos dados. Por fim, a camada mais distante do usuário é composta pelos servidores em nuvem. Estes por estarem mais distantes dos dispositivos de borda apresentam uma maior latência na transferência de dados, porém estes provém uma maior capacidade computacional e capacidade de armazenamento de dados, por exemplo, estes servidores oferecem um processamento massivo de dados em paralelo, mineração de big data, entre outros serviços.

2.4 MQTT Broker

Um importante ponto a se considerar quando se deseja um sistema com dispositivos sensores que controlam atuadores é o tempo que leva para que essa comunicação seja feita. Uma solução capaz de agilizar este processo de comunicação entre os dois dispositivos é o MQTT Broker.

Conforme o trabalho apresentado em (JUTADHAMAKORN et al., 2017) o MQTT é

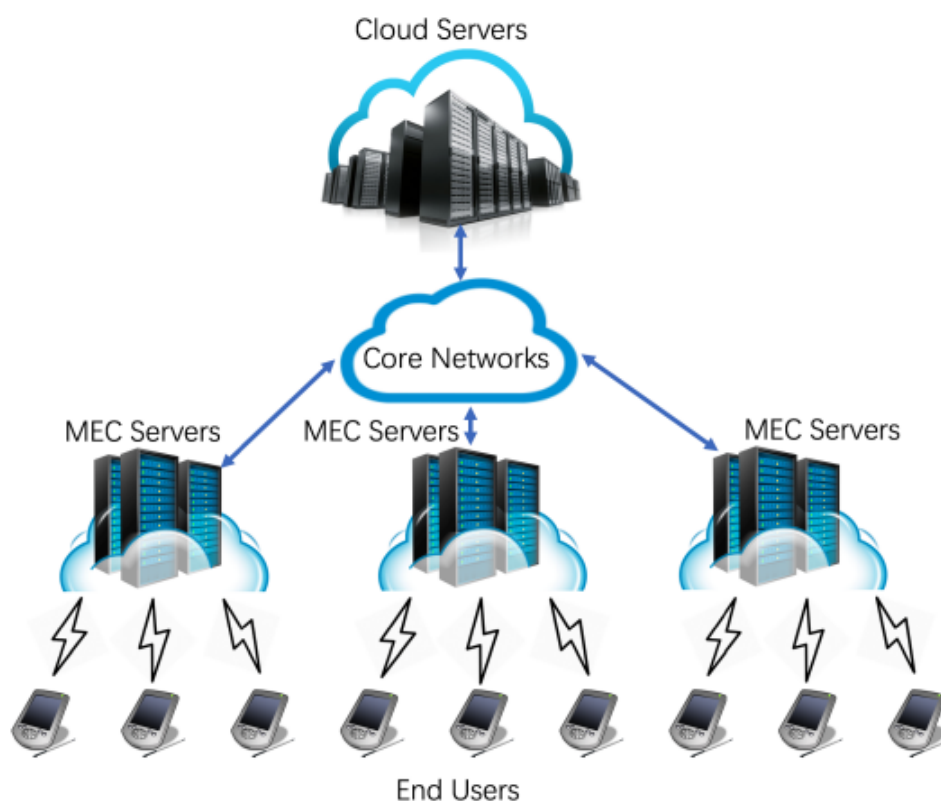


Figura 4 – Arquitetura básica de computação em borda, segundo (YU et al., 2017).

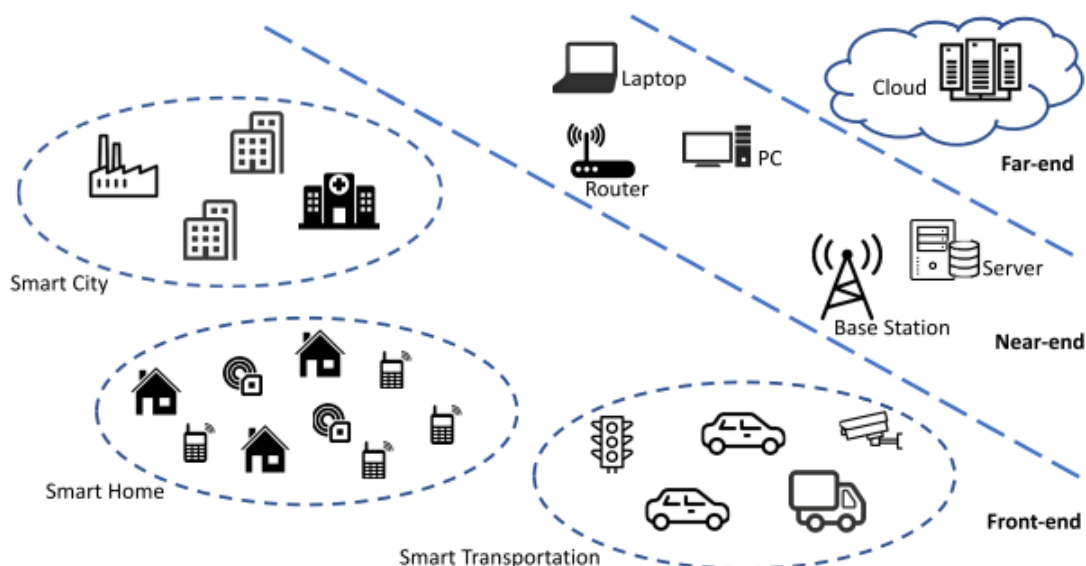


Figura 5 – Arquitetura típica de computação em borda, segundo (YU et al., 2017).

um protocolo leve de troca de mensagens criado para uso de comunicações máquina a máquina, o protocolo funciona com um sistema de tópicos, onde os dispositivos se conectam de modo a publicar nesses tópicos ou se inscrevem para receber tudo que foi publicado nos mesmos. O MQTT Broker é foi criado a partir de um modelo de publicação e assinatura baseado em tópicos, onde os dispositivos conectados a este que desejam enviar mensagens a outros dispositivos não precisam ter conhecimento

destes outros elementos da rede, basta que estes publiquem e/ou assinam determinado tópico do MQTT Broker, sendo este o responsável por fazer todo intermédio da troca de mensagens dos elementos da rede. De forma resumida, o princípio básico de funcionamento do MQTT Broker seria composto pelos seguintes passos:

- Dispositivo se conecta ao MQTT Broker;
- Caso este deseja publicar algo, envia uma mensagem como Publish para determinado tópico;
- Caso este deseja acessar alguma informação, envia uma mensagem como Subscriber para determinado tópico;

Todo processo restante de recebimento e envio de mensagens é feito pelo MQTT Broker, reduzindo alguma possível carga de trabalho dos dispositivos da rede. Por fim, ainda neste trabalho desenvolvido em (JUTADHAMAKORN et al., 2017) são consideradas sugestões para questões de escalabilidade desta tecnologia, como o Framework desta dissertação foi desenvolvido com preocupações referentes a características de escalabilidade é necessário caso haja necessidade de conectar uma grande quantidade de dispositivos ao Broker isto seja possível, para tal são consideradas duas soluções: aprimorar o desempenho de uma máquina que seja responsável por hospedar o MQTT Broker, ou utilizar um sistema de clusters, de modo a distribuir a quantidade de sensores entre múltiplos MQTTs Brokers.

2.5 LoRaWAN

O crescimento no campo da Internet das Coisas levou ao desenvolvimento de muitas aplicações que utilizam redes sem fio para comunicação, como sistemas de cidades inteligentes e agricultura inteligente, conforme o trabalho feito em (COTRIM; KLEINSCHMIDT, 2020) redes de Longo Alcance de Baixa Potência (LPWANs) atendem vários requisitos que sistemas IoT necessitam, tais como, eficiência energética, baixo custo, grande área de cobertura, e implantação em larga escala. A rede LoRaWAN é uma das tecnologias derivadas da LPWAN e está dentre uma das mais pesquisadas e implementadas, devido à facilidade de desenvolvimento de redes privadas com um padrão aberto. Ainda conforme o trabalho de (COTRIM; KLEINSCHMIDT, 2020) LoRaWAN é um protocolo da camada de enlace de dados desenvolvido com base na tecnologia de modulação LoRa, desenvolvido pela LoRa Alliance em 2015, seu objetivo é prover uma solução de baixo consumo para dispositivos alimentados por bateria.

Segundo o trabalho feito em (BOUGUERA et al., 2018) sensores sem fio permitem uma vasta gama de aplicações, como detecção de eventos, monitoramento de

eventos, entre outros. Estas aplicações têm o propósito de coletar informações sobre determinado evento ou fenômeno. Tais sensores podem ser colocados em ambientes de difícil acesso e por tanto é esperado que estes operem por um longo período sem intervenção humana. Estes sensores devem executar tais tarefas para a maioria das aplicações: monitorar o evento, processar localmente a informação coletada e transferir esta informação ao ponto de acesso, cada tarefa destas consome energia e de modo a otimizar este consumo pode ser proposto um modelo de consumo de energia.

Além disto, neste trabalho é proposto um modelo consumo de energia baseado em tecnologia de modulação LoRa e LoRaWAN, e conforme os autores o baixo consumo destas tecnologias associado com sua capacidade de longo alcance de comunicação coloca estas no topo das tecnologias LPWAN, além disto de acordo com sua modulação única, por fim é definida a tecnologia LoRa como sendo esta uma tecnologia muito versátil, que pode ser adaptada a diferentes tipos de ambientes e classes de aplicação o que acaba por incentivar ainda mais seu uso.

2.6 Apache Nifi

Considerando a possível grande quantidade de sensores presentes no Framework, principalmente por este ser desenvolvido buscando ter características de escalabilidade, é necessário que haja um suporte para uma fácil integração e visualização de todos os sensores da rede. Para tal é utilizado o Apache Nifi, segundo o trabalho feito em (SAMAL; PANDA, 2017) ao utilizar este software é possível automatizar o fluxo de dados no sistema, o Nifi oferece suporte para transportar e manter o rastreo dos dados, o usuário pode ajustar os fluxos sem utilizar comandos via linhas de código, podendo apenas utilizando sua interface gráfica para tal. Ainda de acordo com este trabalho está são algumas dos conceitos básicos do Apache Nifi:

- **FlowFile:** Cada objeto que esta se movendo no sistema, como por exemplo um arquivo JSON com dados de um sensor, é representado por um FlowFile e o Apache Nifi mantém dados adicionais relacionados a este arquivo, como por exemplo o tamanho dele e data de criação.
- **Processador do FlowFile:** Este processador é responsável por fazer o roteamento, transformação e todas as eventuais mudanças neste objeto FlowFile, onde estão os dados coletados e os que o Nifi adiciona, este Processador pode ser usado também para alterar o conteúdo do arquivo original, caso necessário.
- **Controle de Fluxo:** atua como um agente intermediário para transmitir as FlowFiles entre os Processadores, este mantém o registro de como os processadores estão conectados de modo que as FlowFiles só sejam enviadas para onde seja necessário

No trabalho feito em (GUPTA, 2022) é utilizado Apache Nifi para o controle do fluxo de dados da aplicação, na Figura 6 é possível observar um exemplo simples de um fluxo de dados neste software, a esquerda temos um processador que gera FlowFiles, objeto básico do Apache Nifi, um exemplo prático disto seria um arquivo FlowFiles composto com dados de sensores, os dados de sensores ao serem enviados para o Apache Nifi são convertidos em FlowFiles para que possam ser transportados pelo sistema, como este exemplo não tem uma entrada de dados foi utilizado um gerador de FlowFiles. Estas FlowFiles que estão sendo geradas pelo elemento a esquerda são enviadas para o elemento a direita, que seria um Processador de FlowFiles, neste exemplo o Processador de FlowFiles envia estas para um banco de dados da Amazon, o AWS S3, no Framework desenvolvido nesta dissertação o processador poderia, por exemplo ser utilizado para enviar os dados para o Apache Hive. Estes dois elementos são conectados por um elemento central, chamado de Conexão, que atua de modo a coordenar este processo de envio de FlowFiles, monitorando as filas que podem ser geradas no envio de arquivos, assim como a taxa com que estes são enviados, dentre outros pontos que este pode monitorar para garantir o fluxo de dados desejado.

Este é um simples exemplo de como é o fluxo de dados no Apache Nifi, porém é possível visualizar um pouco de sua interface e como os elementos são organizados nesta, por fim é importante considerar que o Apache Nifi consegue garantir a entrega dos dados, mesmo com alta demanda, o que traz uma característica de tolerância a falha no sistema.

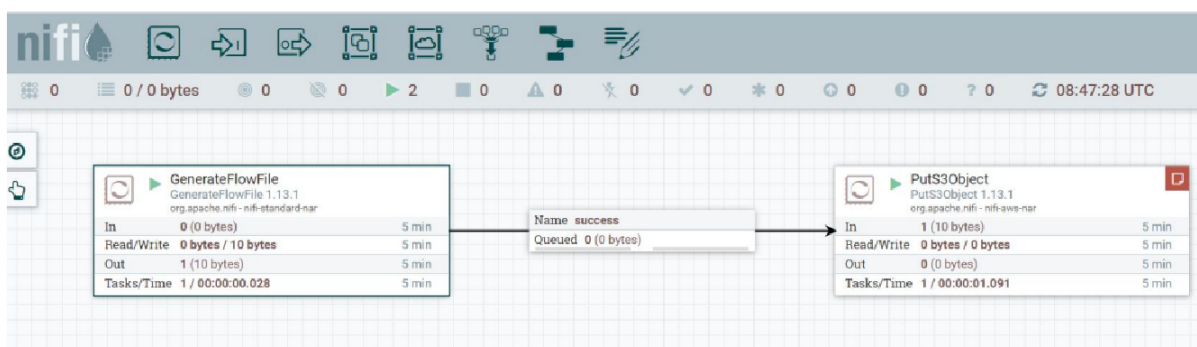


Figura 6 – Demonstração de um fluxo de dados no Apache Nifi, segundo (GUPTA, 2022).

O Apache Nifi tem suporte para o uso de arquivos JSON como dados de entrada e é possível que seja integrado ao Apache Hive,

2.7 Big Data

O termo Big Data é encontrado como algo sem uma definição exata sobre o que se refere, conforme o trabalho feito em (WARD; BARKER, 2013) o termo se tornou ubíquo, estando presente tanto na academia, indústria e mídia, este não possui uma definição unificada, devido à falta de uma definição consistente o termo se torna ambí-

guo e acaba por desencorajar seu uso. Ainda de acordo com este trabalho citado Big Data é predominantemente associada com duas ideias, armazenamento de dados e análise de dados, com isto surge a questão do porquê Big Data se difere notavelmente das técnicas de processamento de dados convencionais. Algumas definições são encontradas na literatura de modo a responder isto, a empresa Intel é uma das poucas organizações a descrever de forma concreta o termo, Big Data estaria diretamente conectada a organizações, este descreveria a produção de 300 terabytes de dados semanalmente, diferente de outras definições a Intel descreve Big Data através da quantificação de dados gerados, a empresa Microsoft apresenta outra definição para o termo, em sua literatura Big Data é o termo utilizado para descrever o processo de aplicar sério poder computacional, o que há de melhor em termos de aprendizado de máquina e inteligência artificial, em uma quantidade massiva e geralmente complexa de dados.

Conforme é apresentado em (DE MAURO; GRECO; GRIMALDI, 2015) o termo Big Data é definido através de muitos conceitos vagos, este é usado para descrever um largo escopo de conceitos, desde a capacidade tecnológica de armazenamento, agregação e processamento de dados, até a mudança cultural que está invadindo os negócios e a sociedade. Além disso, a ambiguidade presente entre pesquisadores e desenvolvedores que utilizam o termo dificulta um desenvolvimento eficiente de uma definição. O autor deste trabalho utilizou 1473 trabalhos relacionados a Big Data de modo a identificar em seus resumos quais eram os termos-chave mais presentes, na Figura 7 é possível observar uma visualização do resultado desta busca em um formato de nuvem de palavras. Através de uma análise dos termos-chave mais presentes e suas relações mutuas foram identificados quatro tópicos, que segundo o autor a maioria dos trabalhos escritos que tratam sobre Big Data podem ser classificados em algum destes, sendo estes: informação, tecnologia, métodos e impacto. Tecnologia trata de um dos principais fatores para a existência do termo Big Data, que seria decorrente da grande quantidade de informação que pode ser gerada e disponibilizada de forma digital, o termo informação faz referência as tecnologias que permitem sua utilização, a grande quantidade de dados e as complexas operações para o seu processamento implicam em um armazenamento de memória rigoroso e requisitos de desempenho computacional. Métodos tem relação com as técnicas de processamento necessárias para analisar uma vasta quantidade de dados e obter informação relevantes através destes e como estes métodos vão além das técnicas estáticas tradicionais, por fim o termo impacto trata de como Big Data vem impactando nossa sociedade e empresas através do uso de suas tecnologias e métodos associados.

Apesar do trabalho feito (DE MAURO; GRECO; GRIMALDI, 2015) identificar os tópicos principais onde o termo Big Data está relacionado, este não apresenta uma definição concreta, assim como o trabalho citado anteriormente ambos descrevem

Big Data como um termo ainda sem definição, que mesmo estando relacionado a tópicos semelhantes ainda é muito amplo e pode causar ambiguidade em seu uso até mesmo por ser um conceito relativamente novo. Outro ponto é que segundo os autores muitas definições e utilizações do termo ignoram antigas definições, sendo este o principal fator para a ambiguidade do termo.



Figura 7 – Visualização em formato de nuvem de palavras de termos chave presentes em resumos de artigos relacionados a Big Data, segundo (DE MAURO; GRECO; GRIMALDI, 2015).

2.8 Campus Inteligente

De forma geral, a definição de Campus Inteligente parte do conceito de Cidade Inteligente. Conforme o trabalho proposto em (VILLEGAS-CH et al., 2019) Cidades Inteligentes são parte resultados dos avanços feitos na área de TIC, onde graças aos dispositivos IoT, múltiplos dispositivos podem se conectar a Internet e gerar informações que podem ser compartilhadas com os demais membros da rede. Porém, para que uma cidade seja considerada inteligente são necessários muitos esforços técnicos e socioculturais, além da demanda de especialistas em diversas áreas trabalhando visando otimizar recursos. Assim como nas cidades, os desafios são semelhantes no ambiente da universidade e o uso das TICs se faz presente no Campus Inteligente de modo a melhorar a qualidade de ensino do campus e a qualidade de vida dos seus frequentadores. Além disso, o Campus Inteligente propicia uma melhor coexistência entre a universidade e tudo ao seu redor, manejando seus recursos de forma mais adequada e providenciando locais favoráveis ao aprendizado. Este trabalho apresenta de forma visual, como na Figura 8 a proposta de uma arquitetura de Campus Inteligente, composta de 4 estágios, cada um contendo os elementos-chave e os componentes utilizados.

De acordo com (PAGLIARO et al., 2016) campus universitários representam per-

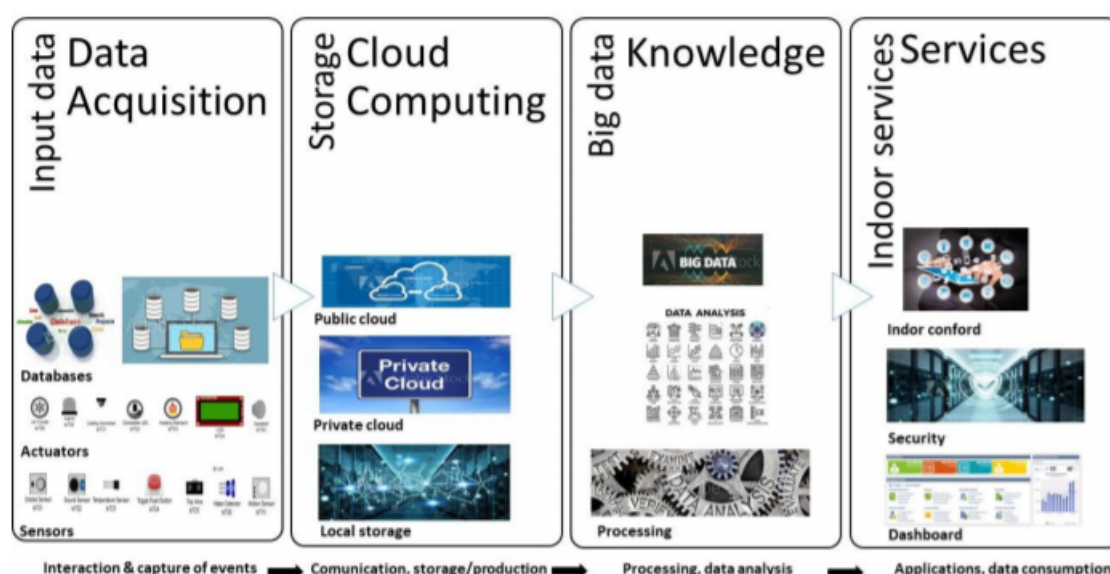


Figura 8 – Arquitetura Campus Inteligente, segundo (VILLEGAS-CH et al., 2019).

feitamente o exemplo de uma célula urbana e são comparáveis a cidade. Apesar de suas diferenças com relação ao tamanho e tipos de estruturas, os campus universitários podem ser comparados a cidades com relação a alguns de seus problemas e desafios, tais como: impacto ambiental, mobilidade interna e externa, falta de serviços fundamentais, causando insatisfação de seus usuários, entre outros. Com isso em consideração, o modelo que define uma Cidade Inteligente é adequado para ser transformado em um modelo de Campus Inteligente. Apesar da definição exata do que define um Campus Inteligente não ser bem explícita, os trabalhos concordam que esta pode ser comparada e então definida a partir do conceito de Cidade Inteligente, porém considerando o contexto e alterando o modelo conforme o necessário.

2.9 Framework

Um Framework IoT pode ser visto como uma camada de intersecção entre uma ou mais aplicações IoT, de acordo com (CHERUVU et al., 2020) esta camada apresenta uma interface de aplicação voltada para redes, pelas quais vários nós de sistemas IoT interagem. Além disso, um Framework IoT é desenvolvido com quatro objetivos primários: reduzir o tempo de desenvolvimento trazendo soluções IoT mais rapidamente ao mercado, reduzir a complexidade aparente de utilizar um sistema IoT, aprimorar a interoperabilidade e portabilidade da aplicação IoT e por fim facilitar os serviços de manutenção deste sistema. Ainda de acordo com este trabalho, o Framework camufla a complexidade de conexão através de métodos de abstração de alto nível de envio de mensagens, de modo a alcançar este objetivo alguns métodos são utilizados, como por exemplo a definição global do formato das mensagens para seu envio, a definição

do formato dos dados utilizados na aplicação, entre outros. De modo geral o Framework IoT simplifica redes IoT ao criar abstrações destas redes, que ocultam grande parte da complexidade presente, para que isto seja alcançado estes Frameworks expõem dados, funções e interfaces que facilitam a interoperação.

Na literatura são encontrados vários exemplos de uso de Frameworks, como no trabalho de (NG et al., 2010) é proposto um Framework holístico que ajuda a definir os diferentes componentes operacionais de um Campus Inteligente a fim de criar categorias de inteligência funcional, como pode ser visto na Figura 9. Estes componentes funcionais inteligentes conectados dão origem ao Framework e possuem muito mais impacto dessa forma que a soma destes vistos separadamente.

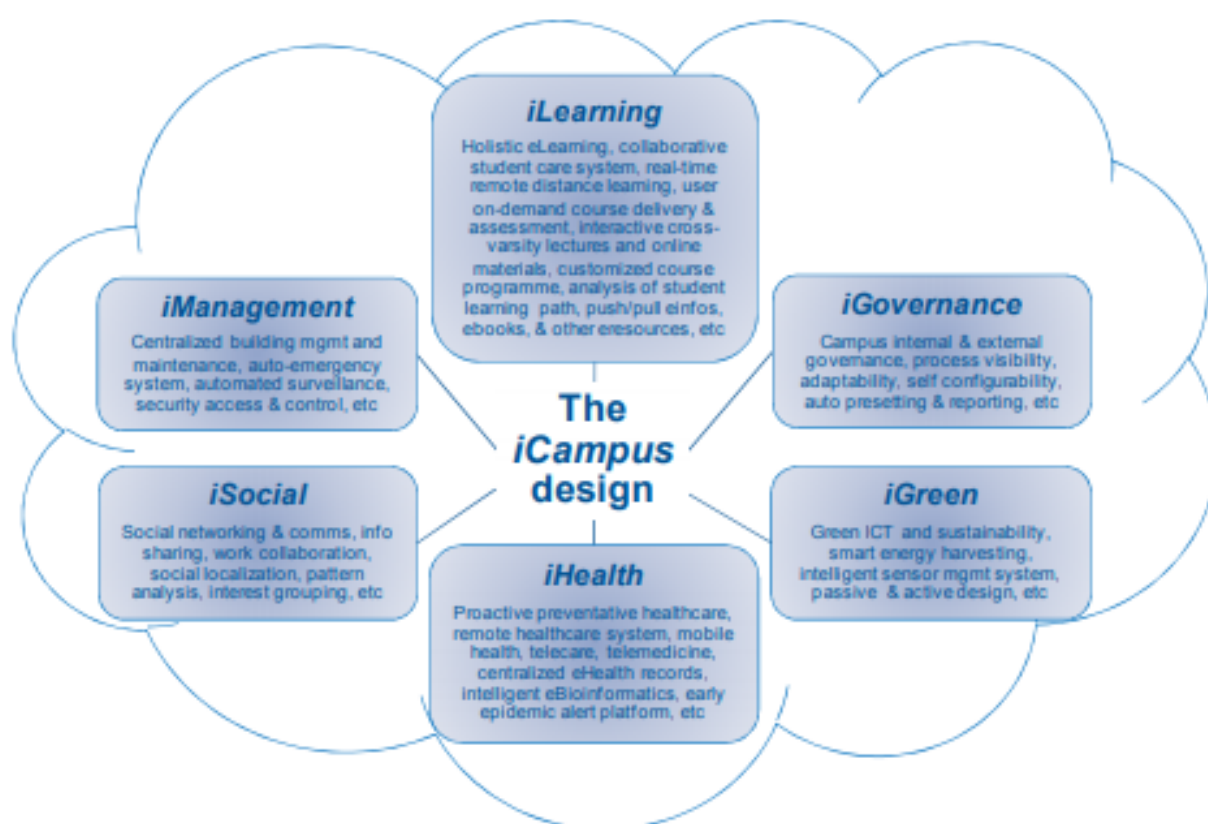


Figura 9 – Componentes do Framework, segundo (NG et al., 2010).

No trabalho feito em (MOHAMMADI; AL-FUQAHA, 2018) é proposto um Framework para Cidades Inteligentes. Este oferece três níveis de inteligência como é mostrado na Figura 10, cada um dos componentes da cidade inteligente é controlado por um software inteligente que se encontra disponível em um ambiente de computação em nuvem ou névoa, dependendo das necessidades do mesmo. A ideia deste trabalho é utilizar muitos dados que não são aproveitados de cidades para melhorar os serviços oferecidos e sua eficácia, para tal usa-se este Framework construído com base em aprendizado profundo por reforço e semi supervisionado. Dada a alteração de contexto em relação aos dois trabalhos citados nessa seção são notáveis as di-

ferências entre os Frameworks propostos, porém ambos apresentam características esperadas de um Framework, como por exemplo apresentar níveis de abstração na rede de dispositivos IoT de modo a facilitar o desenvolvimento.

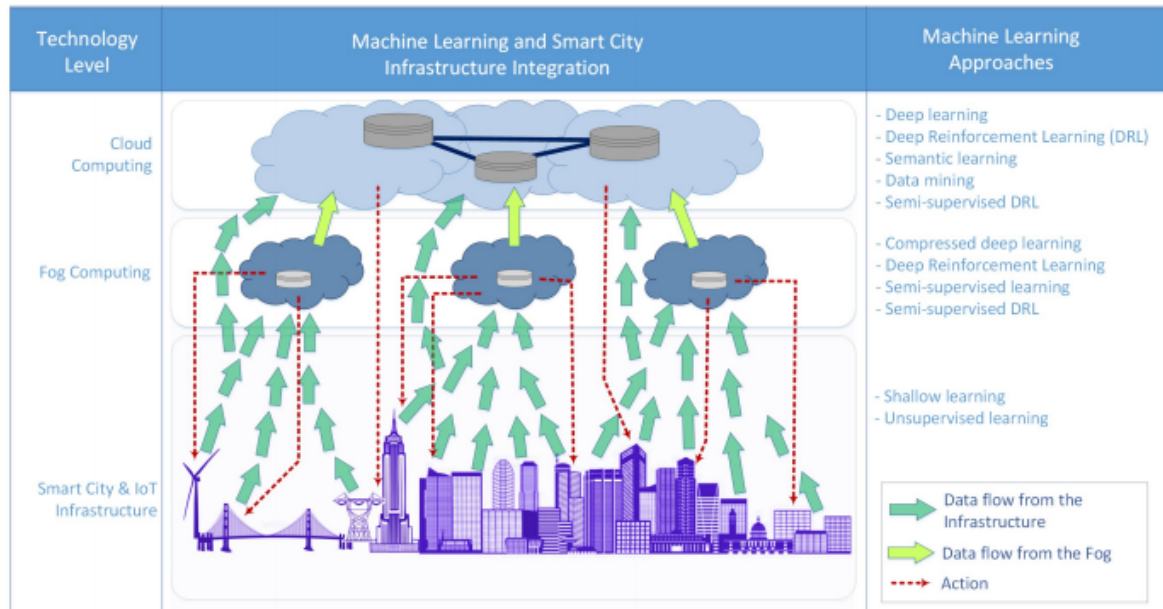


Figura 10 – Níveis de Inteligência do Framework, segundo (MOHAMMADI; AL-FUQAHA, 2018)

3 REVISÃO BIBLIOGRÁFICA

Este capítulo apresneta a revisão bibliográfica desenvolvida utilizando o método de Revisão por Mapeamento Sistemático. É apresentada a metodologia e também a discussão sobre as questões de pesquisa.

3.1 Metodologia

Conforme o trabalho apresentado em (WOHLIN et al., 2013) este tipo de revisão é cada vez mais comum em trabalhos de Engenharia de Software e tem por objetivo apresentar uma síntese dos estudos na área, geralmente em torno de uma questão de pesquisa central, de modo a tentar identificar possíveis lacunas na área, utilizando para tal uma metodologia de pesquisa bem definida.

O método de Mapeamento Sistemático visa principalmente dar uma visão geral da área de pesquisa em questão, identificando a quantidade e o tipo de pesquisa que vem sendo desenvolvida, para tal é utilizado uma metodologia bem estabelecida (PETERSEN et al., 2008). Neste trabalho também é proposto um fluxo de pesquisa (Figura 11) que servirá como base para o desenvolvimento deste trabalho aqui apresentado.

O trabalho feito em (PETERSEN et al., 2008) também determina que o processo de pesquisa seja motivado por uma RQ (*Research Question*). De modo a facilitar a pesquisa, essa questão é dividida em várias outras RQs (*Research Questions*) mais específicas. A RQ central deste trabalho é: "*Como um Framework pode ser utilizado no contexto de um Campus Inteligente?*" e a partir desta são criadas as outras RQs:

- Qual o propósito de utilizar um Framework na construção de um Campus Inteligente?
 - A intenção é descobrir as motivações do uso de um framework em Campus Inteligente.
- Quais são os tópicos de pesquisa mais investigados e como eles mudaram ao longo do tempo?
 - A ideia é agrupar os pesquisadores em áreas.

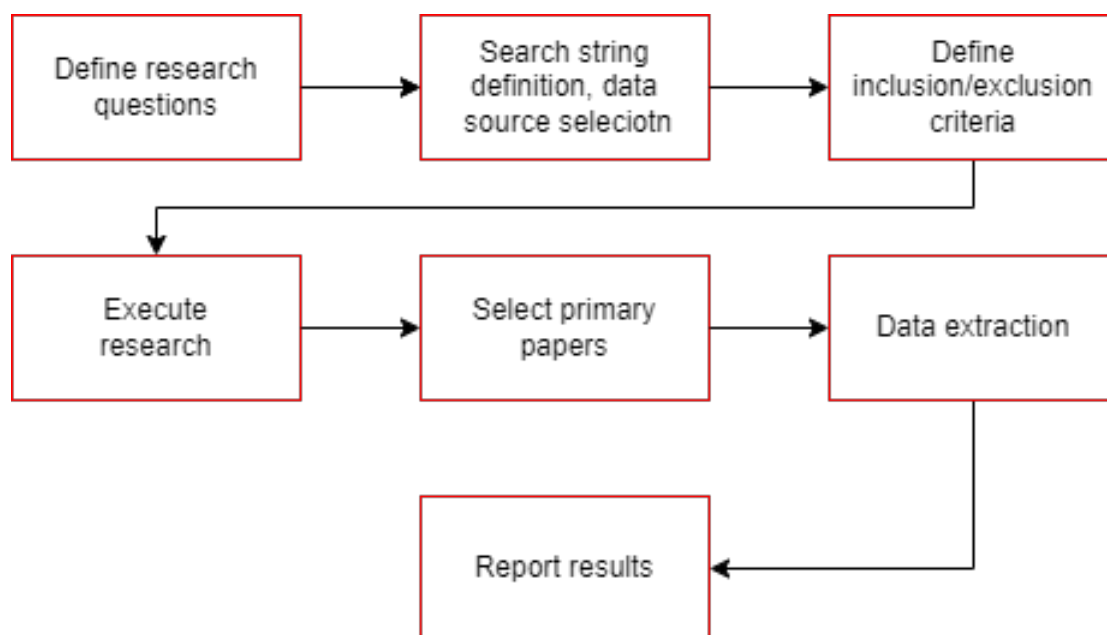


Figura 11 – Fluxo do Mapeamento Sistemático, segundo (PETERSEN et al., 2008)

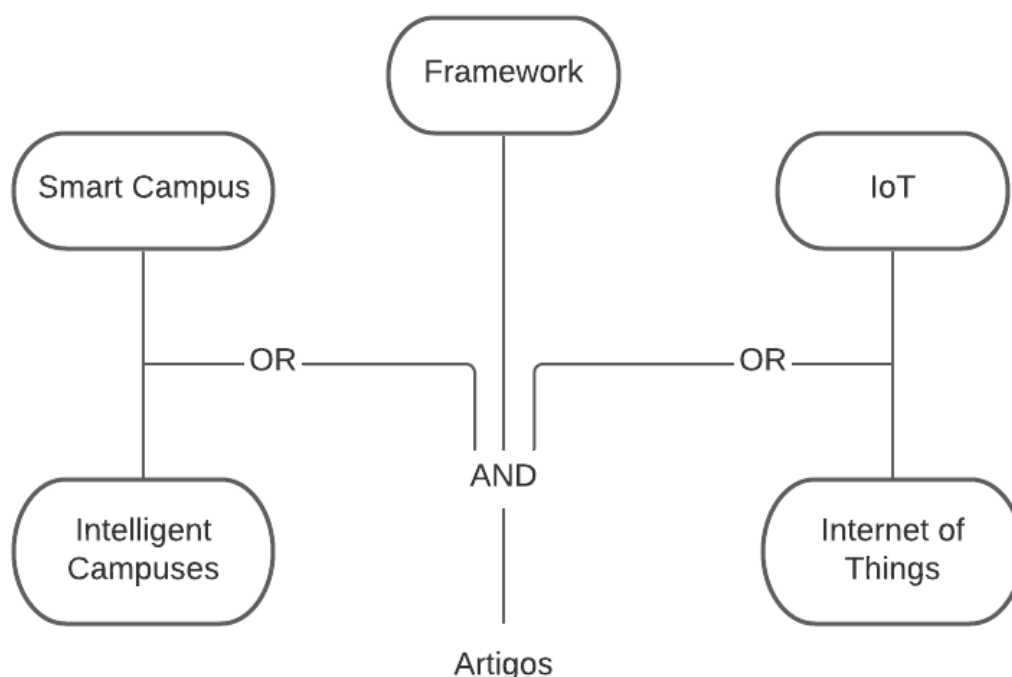
- Quais são os principais desafios de utilizar um Framework em um Campus Inteligente?
 - A ideia é identificar as limitações do Framework nesse contexto e achar oportunidades para este atuar.
- Como foram desenvolvidos outros Frameworks já utilizados no contexto de Campus Inteligente?
 - A intenção é identificar quais linguagens/tecnologias poderiam ser utilizadas para construir o Framework.

A partir das RQs são extraídas as palavras-chave que serão utilizadas para montar as *strings* de busca. O trabalho feito em (KITCHENHAM et al., 2010) sugere o modelo PICO (*Population, Intervention, Comparison and Outcome*) para sistematizar a extração dessas palavras-chave, a pesquisa dos trabalhos será feita com base nessas palavras. A Tabela 1 mostra o resultado da aplicação desse modelo nas perguntas e a Figura 12 mostra o diagrama que serviu como base para que fossem feitas as *strings* de busca.

Com relação às bases de dados que seriam utilizadas para que fossem feitas as buscas dos artigos, o trabalho feito em (KITCHENHAM et al., 2010) sugere que duas bibliotecas digitais e duas bases indexadoras sejam utilizadas como fontes de busca. Bibliotecas digitais contém trabalhos cujos variados autores escolheram como meio para sua publicação, enquanto as bases indexadoras apontam para variadas bibliotecas que contenham trabalhos de interesse. As bibliotecas digitais escolhidas para realizar este Mapeamento Sistemático foram IEEE Xplore e ACM Digital Library, no

Tabela 1 – Modelo PICO aplicado as *Research Questions*

Critério	Objetivo	Palavras
Population	Campus Inteligente / IoT	Campus Inteligente OR Intelligent Campuses AND IoT OR Internet of Things
Intervention	Framework	Framework
Comparisson	not applied	
Outcome	Application of an Framework in Campus Inteligente Context	Campus Inteligente AND Framework

Figura 12 – Diagrama das *strings* de busca, segundo (KITCHENHAM et al., 2010).

que diz relação a bases indexadoras as escolhidas foram Scopus e Engineering Village.

De modo a garantir uma melhor qualidade nos resultados de busca de artigos, foram definidos critérios de inclusão, que garantem que o trabalho selecionado atende as características desejadas para ajudar a responder as RQs, assim como critérios de exclusão para trabalhos que não atendem os requisitos desejados. Estes critérios são definidos de modo a não incluir trabalhos que não sejam relevantes para a pesquisa, a Tabela 2 mostra os critérios utilizados.

A partir das *strings* de busca foram feitas as pesquisas por trabalhos nas bases selecionadas, de modo a realizar a busca utilizou-se de forma manual das ferramentas disponíveis no próprio site da biblioteca/base indexadora, estas ferramentas permi-

Tabela 2 – Critérios de Inclusão e Exclusão

Tipo	Referência	Descrição
Inclusão	IC1	Trabalhos escritos em inglês
	IC2	Publicados a partir de 2000
	IC3	Artigos em que se tenha feito revisão por pares
	IC4	Artigos relacionados a aplicação de um Framework no contexto de Campus Inteligente
Exclusão	EC1	Artigos Duplicados
	EC2	Artigos não disponíveis para Download
	EC3	Artigos não relacionados a aplicação de um Framework no contexto de Campus Inteligente

tem aplicar alguns filtros, facilitando em um primeiro momento para que se reduza o número de artigos fora do escopo desejado, contudo os resultados precisavam ser melhor analisados para que fossem de acordo com os critérios de inclusão/exclusão.

Foram necessárias 4 etapas para que fosse feita a seleção dos artigos desejados. Inicialmente foram utilizadas as *strings* de busca nas bases citadas, aonde a partir da ferramenta disponível era possível utilizar de alguns critérios de inclusão/exclusão na busca pelos trabalhos. Em seguida, iniciou-se a leitura dos títulos e resumos de todos os artigos de modo a selecionar somente o que se aplicava ao contexto desejado. A seguir, foram descartados os artigos duplicados, isso se aplicou somente às bases indexadoras. Por fim, todos os artigos restantes foram lidos inteiramente, as quantidades de artigos resultantes em cada etapa foram sumarizadas na Tabela 3.

Tabela 3 – Número de trabalhos por etapa de seleção

Fonte	Etapas 1	Etapas 2	Etapas 3	Etapas 4
IEEE Xplore	65	19	19	9
ACM Digital Library	34	7	7	1
Scopus	627	31	27	11
Engineering Village	29	9	1	1
Total	755	66	54	22

Com os trabalhos já selecionados foram feitas buscas nestes de modo extrair informações para responder as RQs, alguns trabalhos colaboraram em responder mais de uma das perguntas. A distribuição temporal destes é apresentada na Figura 13.

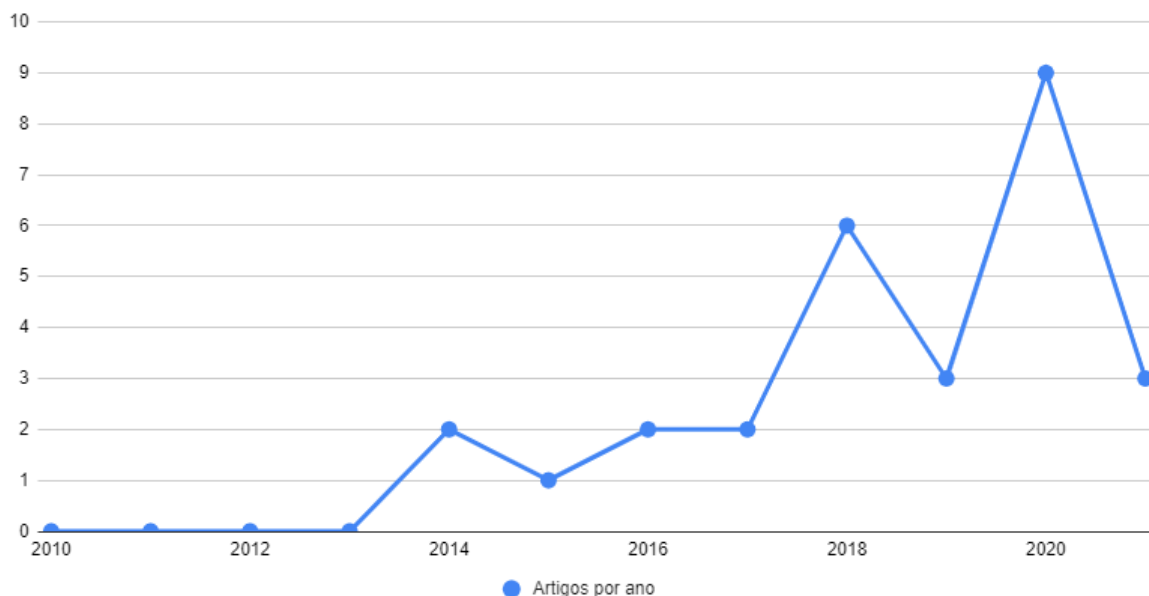


Figura 13 – Distribuição do número de artigos por ano.

3.2 Discussão das Questões de Pesquisa (RQs)

Qual o propósito de utilizar um Framework na construção de um Campus Inteligente?

A motivação principal desta pergunta é identificar motivações para o desenvolvimento de um Framework para o contexto de Campus Inteligente, e se possível, motivações para a construção de um Campus Inteligente.

Muitas cidades estão adotando o conceito de Cidade Inteligente, o qual pode ser estendido e modelado para um Campus, de modo a desenvolver uma melhor qualidade de vida para seus habitantes, reduzir impactos ao meio ambiente e melhorar e eficiência energética (KOSTEPEN et al., 2020). Outro ponto interessante a ser destacado deste trabalho trata sobre uma arquitetura em Campus Inteligente que tinha por objetivo reduzir o consumo de energia do campus, um dos resultados obtidos é que foi possível obter 30% na redução do consumo ao utilizar aplicações que sejam dentro do contexto de Campus Inteligente. De acordo com o trabalho feito em (AGARWAL; GVV; AGARWAL, 2020) Campus Inteligente estão se tornando uma realidade, recorrendo à tecnologia para aumentar a eficiência, tratando-se de sustentabilidade, assim como aumentando a colaboração entre as pessoas e transformando a experiência destes usuários.

O trabalho desenvolvido em (PANDEY; SINGH; RANA, 2020) trata sobre a infraestrutura composta pelo Framework ser a base de construção para um Campus Inteligente, pois com todos os componentes digitais devidamente integrados as informações relacionadas a ensino e aprendizado podem ser facilmente utilizadas/acessadas a partir de qualquer dispositivo. Em (GUO; GUO, 2015) fala-se sobre a importância

da rede do Campus ser gradualmente convertida em uma rede inteligente com um portal unificado, com capacidade de fusão de dados e serviços. Ainda de acordo com este trabalho, o objetivo para uma arquitetura de rede em um Campus Inteligente é conectar todos os serviços de rede, incluindo pontos de acessos menores/estreitos, a uma central de rede de modo a prover um ambiente com fluxo de trabalho inteligente para todos os tipos de sistemas, e para tal é utilizado um Framework constituído de 3 camadas, toda a arquitetura de rede proposta neste trabalho é feita com base neste Framework responsável por unificar o sistema. Conforme o trabalho de (ABUALNAAJ; AHMED; SABOOR, 2020) para se ter uma melhor percepção do que é um Campus Inteligente é preciso aplicar o conceito de Cidade Inteligente de modo a definir o Campus Inteligente com base em três eixos fundamentais, aquisição de dados usando IoT, centralização destes dados e o uso de *big data* para o manuseio e análise destes. A integração destes eixos através de uma infraestrutura centralizada, como um Framework, possibilita a um campus tradicional efetivamente manipular a informação gerada no campus.

As motivações para que seja transformando um campus convencional em um ambiente Inteligente são inúmeras, porém para tal é necessário que o campus seja composto de um sistema capaz de centralizar todas as informações para facilitar seu manuseio e uso, para tal utiliza-se de um Framework modelado para tal contexto.

Quais são os tópicos de pesquisa mais investigados e como eles mudaram ao longo do tempo?

O objetivo com esta pergunta é conseguir agrupar em áreas os trabalhos, facilitando assim a visualização das tendências e identificação de oportunidades. Os tópicos de classificação são resultados do que podem ser observados nos trabalhos. Na Tabela 4 é possível visualizar como foram classificados os trabalhos.

Cada trabalho foi classificado em um ou mais dos seguintes tópicos:

- **Aprendizado Inteligente:** Artigos que propuseram um trabalho que incluía em seu Framework a possibilidade de aprendizado através de *E-learning*, e/ou iniciativas que possibilitem a alunos e professores um melhor ambiente de aprendizado através do uso de dispositivos IoT.
- **Estacionamento/Transporte Inteligente:** Trabalhos em que o Framework dá suporte para o uso inteligente do estacionamento do campus através de automação, podendo também oferecer visualização em tempo real da frota de veículos do campus, de modo a otimizar a vida de quem depende deste meio para se locomover, entre outros tópicos conectados a estes assuntos.
- **Construções Inteligentes:** O Framework contém dados sobre os prédios, salas e demais ambientes dos câmpus, sua ocupação e tudo que mais for conveniente

Tabela 4 – Classificação dos trabalhos em tópicos

Tópicos	Trabalhos
Aprendizado Inteligente	(LIU; MA; JIN, 2018), (PANDEY; SINGH; RANA, 2020), (GUO; GUO, 2015), (LUO, 2018), (PHAM et al., 2020), (CHEN, 2020)
Estacionamento/Transporte Inteligente	(WEBB; HUME, 2018), (PANDEY; SINGH; RANA, 2020)
Construções Inteligentes	(WANG et al., 2017), (DEY et al., 2016), (VILLEGAS-CH; PALACIOS-PACHECO; ROMÁN-CAÑIZARES, 2020), (DEBAUCHE et al., 2020), (PANDEY; SINGH; RANA, 2020), (VALKS et al., 2021), (ABDULLAH; THANOON; ALSULAMI, 2019), (CHEN, 2020)
Mobilidade Inteligente	(PETCOVICI; STROULIA, 2016)
Gestão Inteligente de Recursos	(JOHN et al., 2017), (WANG et al., 2017), (NEWAZ et al., 2014), (MARTINS et al., 2021)
Aspecto Geral	(AGARWAL; GVV; AGARWAL, 2020), (MACIÁ PÉREZ; BERNARDO MARTÍNEZ; LORENZO FONSECA, 2021), (VILLEGAS-CH et al., 2019), (ABUALNAAJ; AHMED; SABOOR, 2020), (KOSTEPEN et al., 2020)

monitorar, como temperatura, ventilação, umidade, entre outros e disponibiliza os dados em sua plataforma.

- **Mobilidade Inteligente:** Trabalhos nos quais o Framework dá suporte para a localização dos usuários que estejam no campus, dentro e fora de prédios e demais estruturas, e mostra rotas de acordo com suas demandas, facilitando a locomoção no campus.
- **Gestão Inteligente de Recursos:** O proposto Framework dava suporte a um controle eficiente dos recursos do campus, como por exemplo otimização no consumo de luz, água, entre outros. Assim como também poderia agregar em seu uso fontes de energia renováveis, como painéis solares, prédios com design sustentável, etc.
- **Aspecto Geral:** Alguns trabalhos propõem Frameworks que contenham em seu projeto 4 ou mais tópicos citados acima, de modo a melhor organizar a visualização estes foram colocados nessa categoria.

No que diz respeito às mudanças ao longo do tempo é possível observar uma maior preocupação no quesito segurança do Framework e das redes onde estão conectados os dispositivos IoT.

Quais são os principais desafios de utilizar um Framework em um Campus Inteligente?

A ideia por trás desta questão é procurar identificar as limitações para o uso de um Framework em um Campus Inteligente, assim como identificar oportunidades para este atuar.

Conforme o trabalho desenvolvido em (WEBB; HUME, 2018) IoT vem gerando cada vez mais excitação com tudo que é possível ser obtido através de seu uso, porém também introduziu um nível inteiramente novo de risco devido a fatores como, variada natureza dos sensores, redes para aquisição de dados e limitações inerentes a questões de controle de segurança relacionadas a sistemas embarcados e computação distribuída. Tais questões estão diretamente atreladas ao Framework, pelo fato deste tratar diretamente com redes compostas por vários dispositivos IoT. Ainda de acordo com este trabalho, a crescente demanda por sensores, dispositivos de computação de borda e redes de aquisição de dados fortalecem a demanda por uma estrutura formal de organização que proteja a infraestrutura tecnológica e garanta que os recursos estão sendo utilizados da forma mais segura e eficiente possível. A recomendação para tal seria o desenvolvimento de um Framework de modo a minimizar os riscos associados a questões de vulnerabilidades de segurança, maximizar a integração entre os dispositivos ligados a tecnologia de informação do campus, entre outros.

O trabalho feito em (VALKS et al., 2021) descreve um pouco do cenário que os campus universitários vem enfrentando ao longo dos anos, onde a combinação de objetivos mais ambiciosos e a constante pressão em questões como recursos humanos, financeiros e energéticos é o motivo pelo qual universidades investem em um manejo cada vez mais eficiente do campus, utilizando para tal a tecnologia de informação. Para tal, um Framework capaz de fazer o manuseio de todas essas informações e dar sugestões administrativas é o recomendado. Entretanto, dada a multiplicidade de informações que os dispositivos IoT são capazes de obter e os diferentes espaços que estes podem atuar, escolher quais dispositivos irão atuar é uma decisão complexa. Adicionalmente estas implementações podem ser custosas e incertas, é importante ao se projetar uma infraestrutura dessa origem escolher uma solução IoT apropriada, especialmente dado o crescente número de dispositivos/sensores. Por fim, o trabalho mostra que 60% das iniciativas IoT não passam da fase de prova de conceito, e somente 26% das iniciativas são consideradas um sucesso. Portanto, assim como construir um Framework que vá de acordo com as demandas do Campus, é necessária uma boa escolha de dispositivos IoT de modo a não afetar a eficiência do sistema.

Em (ABDULLAH; THANOON; ALSULAMI, 2019) o autor propõe um Framework para Campus Inteligente e disserta sobre como este herda os obstáculos que um Campus Inteligente enfrenta, que são categorizados em três áreas: técnica, financeira

e política. No que diz respeito a parte técnica, o maior desafio é ser capaz de planejar uma infraestrutura com um número massivo de dispositivos sem fio e baixo custo em um campus, de modo que concordem com os requisitos de segurança de um sistema deste porte. Isto se torna problemático devido à heterogeneidade dos dispositivos e o ambiente de comunicação massivo em que se encontram. Com relação a questões financeiras, o maior problema é a limitação de recursos que a universidade dispõe para uso, por mais que uma das propostas do sistema seja a redução de gastos para a instituição, a implementação destes requer um investimento inicial, o que muitas vezes pode ser um impeditivo. Ao tratar da parte política, um dos problemas que o sistema pode enfrentar é a oposição de usuários e pessoas que morem em torno ao campus, com base em argumentos relacionados a invasão de privacidade gerados pela infraestrutura, segundo o autor isto é praticamente inconcebível, porém oposição de grupos anti-tecnologia pode ocorrer e deve ser considerado um potencial problema.

É possível observar uma preocupação recorrente no que diz respeito à segurança ao se tratar de um Framework para Campus Inteligente, devido a todos os dispositivos IoT conectados a rede, assim como também preocupações quanto aos gastos referentes a estes dispositivos. Porém, há também em muitos trabalhos a recomendação do uso de um Framework para que se possa viabilizar a construção de um Campus Inteligente da forma mais eficiente possível, tendo em vista a necessidade de ter um sistema que tenha a capacidade de centralizar todas as informações produzidas de modo a facilitar a gestão da instituição, melhorar a qualidade de vida de seus usuários e reduzir o consumo de recursos do campus.

Como foram desenvolvidos outros Frameworks já utilizados no contexto de Campus Inteligente?

O principal objetivo ao responder essa pergunta é identificar quais linguagens e tecnologias já vem sendo utilizadas para construção de Frameworks para Campus Inteligente.

No trabalho desenvolvido em (LIU; MA; JIN, 2018) o Framework foi dividido em três camadas: aquisição de dados, processamento de dados e aplicação dos dados. Na primeira etapa é feita a obtenção de dados através de câmeras, scanners infravermelhos, RFIDs (Radio Frequency Identification) e demais dispositivos IoT. Na segunda etapa, para que seja feito o processamento dos dados são utilizados dois Frameworks de Big Data, o Spark e o Hadoop, ambos projetos de código aberto da *Apache Software Foundation*. Neste projeto o Spark é responsável por analisar todos dados, enquanto o Hadoop é responsável por armazenamentos e gerência destes. Por fim, na terceira etapa, graças ao processamento de dados, informações de várias fontes podem ser disponibilizadas na plataforma de serviço do campus, dados abstratos como avaliação de ensino e administração, suporte logístico, entre outros podem

ser visualmente representados. Com relação à segurança da grande quantidade de dados sendo compartilhados, o Framework pode usar algoritmos de Encriptação de Dados como o SM9, VPN (Virtual Private Network) entre outras tecnologias.

O trabalho apresentado em (DEY et al., 2016) mostra um Framework que conta com diversos sensores e dispositivos IoT para a obtenção de dados, estes por sua vez estão conectados ao Apache Kafka, pois conforme os autores, estes optaram por utilizar um sistema baseado em *JSON*, devido sua baixa complexidade e gastos computacionais. O *Apache Kafka* é responsável por agregar os sensores em grupos, neste caso, cada sala do campus é um grupo composto por seus respectivos sensores. Este software administra o fluxo de dados, porém não faz nenhuma forma de normalização, transformação ou qualquer outro processo adicional nos dados. Os dados que precisavam de transformações eram enviados para o *Apache Nifi*, o qual é um Framework baseado em *browser* capaz de administrar fluxo de dados dinamicamente e fazer todos os processos adicionais nestes. Após os dados serem processados pelo *Apache Nifi* estes enviados para o *Apache Avro*, Framework responsável por normalizar todos dados, deixando-os no formato *JSON*. Por fim, são usados diversos algoritmos de aprendizado de máquina nestes dados para prever a ocupação em prédios.

Em (PETCOVICI; STROULIA, 2016) o trabalho visa desenvolver um sistema capaz de mostrar ao usuário sua localização em tempo real dentro do contexto de Campus Inteligente. Os dados são obtidos dos Smartphones dos usuários do campus. A principal vantagem com isso é não necessitar de uma infraestrutura com propósito exclusivo. O armazenamento de dados é feito utilizando o *PostgreSQL*, ferramenta responsável por gerenciar o armazenamento de dados de bancos objeto relacional. Por fim, os serviços que estão disponíveis em páginas web estão sendo executados no *Apache Tomcat 7*, que é um ambiente de servidor HTTP capaz de hospedar páginas estáticas. O serviço provido pelo Framework proposto neste trabalho também está disponível para sistemas Android.

No trabalho feito em (LUO, 2018) também é utilizado o *Framework Hadoop* para administrar a grande quantidade de dados produzidos pelos vários sensores e demais dispositivos IoT utilizados na obtenção de dados. A rede de aquisição de dados foi projetada utilizando *6LoWPAN*, uma tecnologia para sensores sem fio que é baseada em IPv6 e tem a vantagem de ser de baixo consumo e com pouca demanda de recursos de hardware. No que diz respeito a segurança do sistema, algumas técnicas são utilizadas como por exemplo, o usuário só pode acessar as funções do sistema após ter feito *login*, os usuários possuem diferentes níveis de acessos de acordo com suas permissões e as restrições de cada parte do sistema, entre outros. Por fim, o desenvolvimento da plataforma de dados com a qual o usuário irá interagir é feita com base no *Framework Django Web*, que por sua vez é baseado em *Python*.

De modo a sintetizar o que foi visto nos trabalhos, é apresentado na Tabela 5

algumas tecnologias empregadas com maior frequência nos trabalhos selecionados neste Mapeamento Sistemático. Com base nesta revisão é proposta uma divisão de etapas da construção de um Framework para Campus Inteligente e suas respectivas tecnologias.

Tabela 5 – Tecnologias utilizadas com maior frequência nos trabalhos selecionados

Etapa	Tecnologias
Obtenção de Dados	Sensores/Dispositivos IoT Smartphones
Processamento de Dados	Apache Spark Apache Hadoop Apache Nifi junto ao Kafka/Avro Técnicas de Aprendizado de Máquina
Visualização dos Dados	Apache Tomcat 7 Django Web
Segurança	VPN Sistema de Cadastro de Usuário Padrão de criptografia SM9

Pode ser observado que vários trabalhos utilizam softwares da *Apache Software Foundation*, como o Spark, Hadoop, Nifi, Avro, entre outros. Dentre as justificativas apresentadas para tal é bem frequente ser observado que estes são utilizados por serem softwares de código aberto, além de desempenharem eficientemente suas funções como esperado. No que diz respeito a que tecnologias foram utilizadas para mostrar os dados para os usuários, muitos trabalhos não especificam suas escolhas, assim como quais sensores foram utilizados, visto que os sensores escolhidos acabam por depender do contexto e da necessidade da aplicação. Porém, este Framework busca apresentar com clareza todas as tecnologias utilizadas, assim como justificar as escolhas e exemplificar como poderiam ser integradas no sistema.

Campus Inteligente são cada vez mais presentes e estes são desenvolvidos com base em um sistema central responsável por organizar as informações coletadas, sem essa centralização dos dados torna-se muito mais complexo abstrair tudo que é obtido para que as informações desejadas sejam extraídas.

As funções de um Framework em um Campus Inteligente são variadas, e estes são projetados conforme as necessidades do campus, da mesma forma acontece com a escolha dos sensores/dispositivos IoT utilizados. Há uma preocupação no que diz respeito a segurança de rede quando se faz uso destes dispositivos, além disso, os custos para se obter estes também devem ser considerados, por mais que muitas vezes o projeto tenha por objetivo reduzir gastos, este investimento inicial pode eventualmente ser um problema.

3.3 Considerações

Após cada pergunta ter sido respondida com base nos artigos selecionados no Mapeamento Sistemático, é possível ter uma visão geral do uso de um Framework aplicado no contexto de Campus Inteligente. Assim como existem variados desafios para tal caso de uso, tais como: complexa modelagem e implementação do sistema, preocupações atreladas aos custos para seu desenvolvimento e manutenção, demanda de mão de obra especializada em várias áreas da tecnologia, como modelagem de banco de dados, calibração e instalação de sensores/dispositivos IoT, entre outros. Muitas motivações também se fazem presentes, tais como: redução no consumo de recursos, melhorias na segurança do campus, possibilidade de utilizar metodologias de aprendizado inteligente, possibilitar uma administração mais dinâmica com relação a várias áreas do campus, entre outros.

Também foi possível identificar algumas lacunas na literatura dentre alguns dos trabalhos selecionados, tais como: pouca ou nenhuma característica de escalabilidade, alguns não identificam com clareza as escolhas de tecnologia em cada camada, assim como qual seria a forma definida pelo trabalho para integrar estas, entre outras. Portanto, busca-se neste trabalho um Framework capaz de apresentar as características desejadas já apresentadas, assim como preencher estas lacunas.

4 PROPOSTA DE FRAMEWORK PARA O DESENVOLVIMENTO DE APLICAÇÕES IOT

Este capítulo apresenta a proposta do trabalho que é propor Framework para o desenvolvimento de aplicações IoT no cenário de Campus Inteligente. É apresentada a estrutura em camadas e seu detalhamento e também é apresentada a comparação do Framework proposto com outras propostas da literatura.

4.1 Detalhamento do Framework Proposto

Após ter sido realizada a Revisão por Mapeamento Sistemático, foi possível ter uma visão mais ampla de como outros autores estariam projetando seus Frameworks para Campus Inteligente. A partir deste momento foi feita a divisão em camadas do Framework deste trabalho (Figura 14), com relação ao número de camadas e a responsabilidade de cada uma, este Framework vai ao encontro com o que já está presente na literatura, porém grande parte dos trabalhos encontrados não apresentam com clareza a escolha de tecnologia para cada camada e qual o protocolo para que estas se comuniquem. As tecnologias presentes nas respectivas camadas deste Framework são apresentadas nas seções seguintes. A Figura 14 ilustra a arquitetura do Framework proposto dividido em camadas.

- **Serviços:** Esta camada é responsável por disponibilizar os dados de forma visual para os usuários, sejam estes em um ambiente web ou mobile, permitindo a fácil consulta de tudo que foi coletado, apresentando de diversas formas informações obtidas com base nos dados coletados. Utilizando-se dos dados aqui coletados espera-se ser possível alcançar melhorias na qualidade de vida de alunos/professores/funcionários, otimização no consumo de recursos, administração dinâmica com relação ao uso de salas/laboratórios/auditórios, entre outros serviços.
- **Processamento:** Realiza várias tarefas sobre os dados coletados pela camada de infraestrutura, tais como filtragem, interpretação, inferência, entre outros. Esta

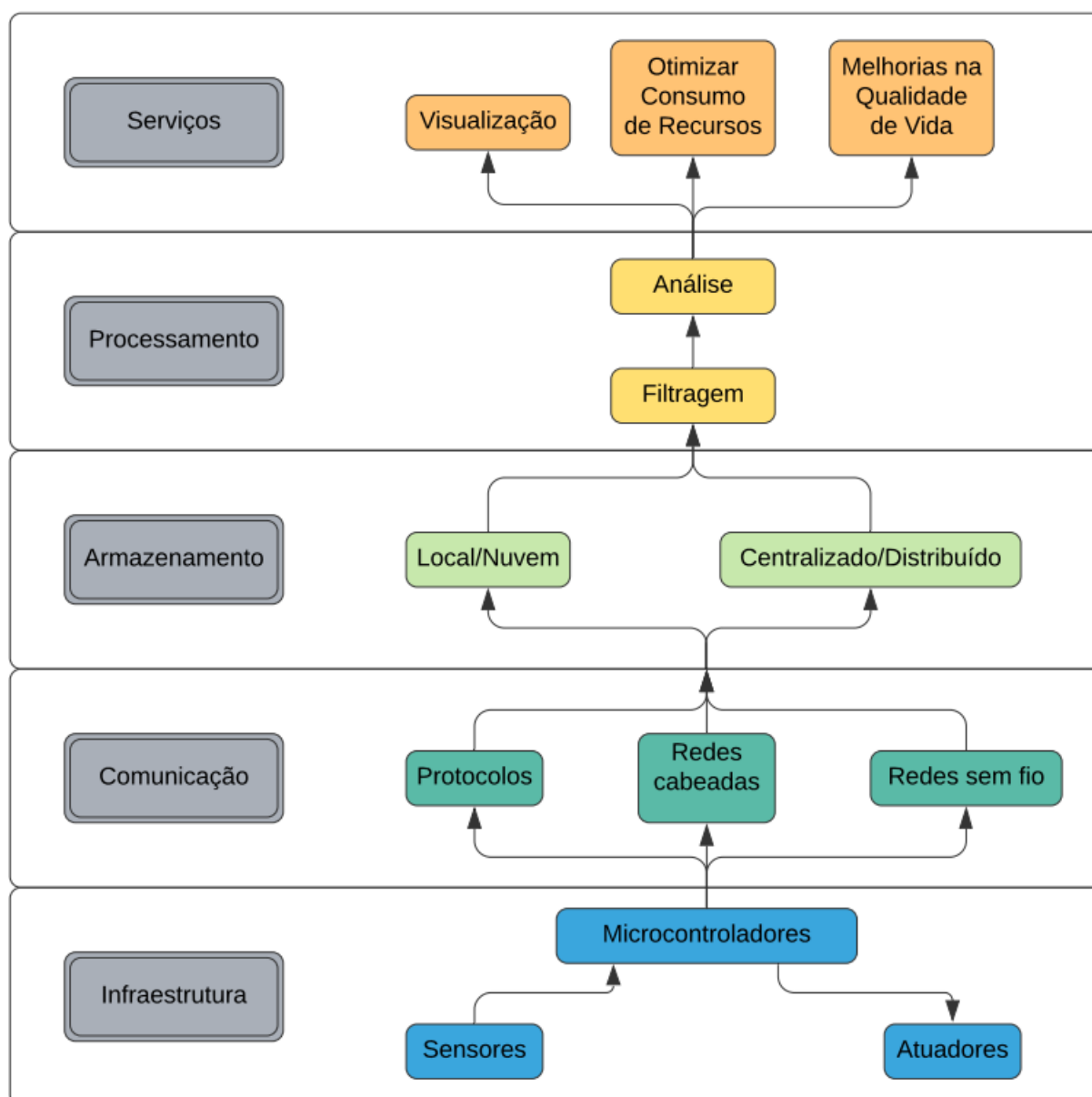


Figura 14 – Camadas do Framework proposto.

camada é formada majoritariamente por algoritmos de inteligência artificial, responsáveis por realizar um processamento dos dados antes que sejam exibidos aos usuários ou para que seja possível fazer alguma sugestão administrativa com base no que foi inferido.

- **Armazenamento:** Camada responsável por dar persistência aos dados coletados pela camada de infraestrutura, assim como também os dados gerados pela camada de processamento. Os dados podem ser salvos tanto em um ambiente local, como por exemplo um servidor em um campus, como também em ambientes web como os serviços de nuvem administrados pela Amazon, Google, entre outros. Assim também como é possível uma junção destes modos de armazenamento, distribuindo os dados para persisti-los conforme a necessidade.

- **Comunicação:** Permite a troca de informações entre todos os dispositivos conectados através das diferentes tecnologias contidas nessa camada. As informações são transmitidas dos sensores para que sejam armazenadas, e caso seja necessário, estes sensores possam receber dados das camadas acima. Também permite a comunicação direta entre os dispositivos da camada de infraestrutura por meio de tecnologias que facilitem a comunicação de dispositivos de borda, tal como um broker MQTT.
- **Infraestrutura:** Esta camada do Framework representa todos os elementos que atuam diretamente com o ambiente onde estão inseridos, seja por meio de captura de informações, como dados de temperatura, luminosidade, entre outros, ou por atuar diretamente controlando dispositivos, como lâmpadas, portas com fechadura eletrônica, entre outros.

Nas seções seguintes são apresentadas as camadas do Framework proposto de forma detalhada.

4.1.1 Infraestrutura

Todos sensores e atuadores estão integrados no Framework, proporcionando interconectividade entre todo sistema, todos estes dispositivos IoT distribuídos no campus formam uma rede responsável por captar dados e atuar dentro do necessário. Na Figura 15 é possível observar alguns dentre os mais variados elementos que compõe esta camada do Framework, além destes da imagem outros dispositivos podem ser adicionados ao sistema, desde que utilizem os protocolos estabelecidos pelo Framework, com isso os mais variados eventos podem ser monitorados, pois o Framework é projetado de modo a permitir escalabilidade, inicialmente a proposta trata dos seguintes:

- **Sistemas Autônomos:** Estes sistemas são projetados de modo a diminuir o consumo de recursos dentro do campus, como, por exemplo, energia elétrica. Um exemplo prático para tal seria um sensor em um corredor, onde somente a luz seria acesa na presença de pessoas, evitando assim consumo desnecessário de energia. Também é possível utilizar estes sistemas para controlar as variáveis de um ambiente, como, por exemplo, umidade de uma sala de aula, ou utilizar um atuador para ligar/desligar o ar condicionado, entre outras possibilidades de modo a proporcionar um maior bem-estar aos frequentadores do campus.
- **Controle de Acesso Inteligente:** De modo a facilitar a entrada de pessoas ao campus em horários onde seria necessária alguma forma de autorização é possível utilizar sensores biométricos ou leitores RFID para que o acesso seja liberado sem nenhuma outra forma de interação, o mesmo vale para salas de

aula/laboratórios/auditórios. Qualquer demanda de acesso ao campus ou a locais específicos dentro do mesmo pode ser armazenada no Framework para que o sistema libere o acesso somente a quem for autorizado, tudo de forma autônoma, utilizando sensores biométricos e/ou RFID.

- **Transporte e Estacionamento Inteligente:** Com uso de sensores integrados aos ônibus disponibilizados pela universidade é possível através de um aplicativo e/ou site exibir em tempo real onde se encontram estes veículos de apoio, de modo a facilitar para quem faz uso destes. Outro uso dos sensores seria mostrar para quem pretende usar o estacionamento da faculdade quanto ao número de vagas disponíveis, facilitando para quem deseja encontrar um local para deixar o carro, além de proporcionar dados sobre uso dos mesmos para o sistema.
- **Sistemas de Segurança:** É vital para um bom funcionamento de um campus que este seja o mais seguro possível para seus frequentadores. Com o uso de câmeras integradas ao Framework é possível usar os dados coletados para identificar de forma automática possíveis riscos e eventos que demandem alguma atenção, assim como verificar quais áreas têm maior/menor fluxo de pessoas para que se possa buscar identificar os motivos para tal.

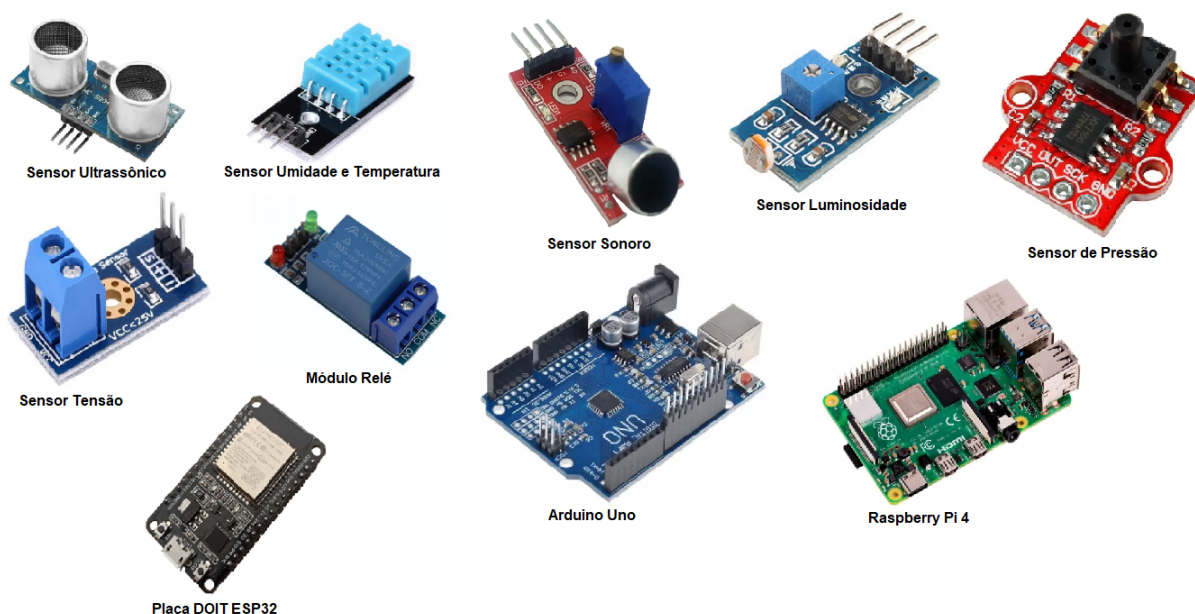


Figura 15 – Exemplos de dispositivos da camada de Infraestrutura

Estes são alguns casos de usos onde os sensores podem ser utilizados, porém, é possível que conforme novas demandas apareçam outros serviços possam ser prestados, dada a escalabilidade do Framework. Com o uso destes dispositivos IoT é possível coletar informações e disponibilizá-las para que sejam tomadas decisões de forma mais rápida e precisa quanto a diferentes aspectos do campus, assim como

possibilitar uma administração mais dinâmica, além de tornar o campus um ambiente mais sustentável ao reduzir o consumo de diversos recursos.

Além dos sensores mencionados também é possível que dados sejam obtidos através de Smartphones dos frequentadores da universidade, para tal é preciso que os usuários estejam conectados a rede do campus e concordem em compartilhar seus dados, como, por exemplo, sua localização no campus. Porém, para que isso seja possível é necessária alguma forma conforme a comunidade, para que aqueles que permitirem tenham alguns dados coletados, como, por exemplo, sua localização no campus, este é um tema com tendência a gerar debates sobre questões de privacidade, portanto não será tratado nessa proposta como algo viável, assim como não será tratado como o sistema deveria ser construído de modo a disponibilizar este recurso.

De modo a facilitar o desenvolvimento do Framework está proposta trata a comunicação dos sensores/atuadores/microcontroladores com as demais camadas utilizando o formato de texto JSON (JavaScript Object Notation). Esta padronização visa facilitar a escalabilidade do Framework, visto que a única preocupação com relação à comunicação seria que ao se comunicar com as outras camadas se utilize do formato JSON no envio/recebimento de mensagens. Desta forma o princípio básico é que a mensagem seja no formato chave/valor, assim como na Figura 16, independente da quantidade de chaves e valores presentes o necessário é que a mensagem seja composta por estes pares, desta forma o sistema será capaz de interpretá-las.

Tendo em vista as diversas camadas, cada uma com suas próprias demandas e especificações, desenvolver um Framework como este iria demandar profissionais especializados em diversas áreas/tecnologias. Buscando reduzir a quantidade de diferentes tecnologias presentes, é proposto o uso do JavaScript tanto para o frontend/backend quanto para os microcontroladores, os quais estão presentes nessa camada. Para tal pode ser utilizado o Johnny-Five, uma plataforma capaz de traduzir código em JavaScript para a linguagem utilizada no Arduino/Esp. Por fim, é importante salientar que não se faz necessário para o funcionamento do Framework o uso do Johnny-Five, para que o código escrito para os microcontroladores seja recompilado em Javascript, este recurso torna-se opcional dado o contexto de uso.

```
1 {  
2   "sensor_id": "001",  
3   "sensor_data": "127",  
4   "date": "2022-03-17T21:08:54.000Z"  
5 }
```

Figura 16 – Exemplo de Arquivo JSON

Com relação às tecnologias presentes nessa camada, é proposto o uso de micro-controladores como Arduino e Esp, também é possível que sejam utilizados micro-computadores, como a Raspberry Pi. Com relação a sensores/atuadores é possível que sejam utilizados todos que forem compatíveis com os microcontroladores e/ou com a Raspberry Pi.

Outra importante observação a ser feita com relação a essa camada do Framework é sobre como os sensores se comunicam com os atuadores. É proposto o uso de um Broker MQTT, este será melhor detalhado na seção a seguir por pertencer a tal camada. O objetivo com o uso do Broker MQTT é facilitar a comunicação entre os sensores/atuadores, permitindo um caminho mais curto para troca de mensagens entre estes, pois deste modo não é necessário enviar a mensagem até as camadas mais acima para que então esta seja processada e volte, somente então sendo recebida por algum componente. Os dispositivos ainda informam ao Framework sobre os eventos ocorridos, porém, podem tomar decisões de forma autônoma e ágil.

Por fim, é necessário considerar que apesar da proposta utilizar o formato de texto JSON para a comunicação das camadas que possuam tal opção, também é possível integrar ao sistema dispositivos que geram stream de dados, tal como câmeras, captadores de áudio, entre outros, de modo a monitorar certos eventos. Mesmo estando utilizando outra forma de transmitir os dados gerados, dadas as tecnologias utilizadas nas outras camadas torna-se possível integrá-las ao sistema, por exemplo, o Apache Nifi é capaz de nativamente tratar a integração com um fluxo de dados vindo de uma câmera.

4.1.2 Comunicação

De forma que os dispositivos IoT possam enviar/receber informações do restante do Framework é necessária uma etapa responsável por esta comunicação. Os dispositivos IoT formam uma grande rede de aquisição/uso de dados, que precisam se comunicar entre estes, de modo a atuar diretamente com o ambiente quando preciso e com serviços de armazenamento de dados, desta demanda surge esta camada.

Considerando a infraestrutura de rede presente nos campus da UFPel atualmente, a comunicação dos sensores/atuadores com o sistema pode ser feita por meio desta. Todos os dispositivos IoT que estejam próximos a uma rede cabeada ou sem fio podem fazer uso destas para que sejam integrados ao Framework.

Para áreas sem acesso a redes de internet é proposto o uso da tecnologia LoRa, permitindo assim uso de sensores em locais mais afastados, utilizando uma tecnologia de baixo consumo e longo alcance. Ainda em relação a locais sem cobertura de sinal de internet provido pelos campus, também é possível o uso de tecnologias como 4g e 5g para inserir os sensores/atuadores na rede, porém é preciso uma avaliação com relação aos custos para tal em comparação a outras opções disponíveis, como LoRa.

De modo a agilizar a comunicação entre os dispositivos da camada de infraestrutura é proposto neste Framework o uso de MQTT Broker. No trabalho feito (JUTADHA-MAKORN et al., 2017) em é proposto um MQTT Broker de baixo custo, escalável e de alto desempenho. A inserção do MQTT Broker neste Framework visa agilizar a comunicação entre sensores e atuadores, sem a necessidade destes somente realizarem troca de mensagens dependentes de mais camadas, como, por exemplo, armazenamento e processamento. Alguns destes dispositivos da camada de infraestrutura podem ser projetados para que atuem dependendo somente dessa troca de mensagens através do broker MQTT, agilizando a tomada de decisões dos mesmos. É considerado que estes enviam mensagens para as outras camadas alertando sobre os eventos ocorridos, de modo que tudo fique registrado no banco de dados do sistema.

Uma preocupação considerada é como estes vários sensores iriam transmitir ininterruptamente, uma grande quantidade de dados simultaneamente na rede, principalmente para garantir escalabilidade para o Framework. Para tal é proposto o uso do software Apache Nifi nesta camada de comunicação, de acordo com (KIM; LEE; GO, 2019) o Apache Nifi é um sistema ETL (Extract, Transform, Load) em tempo real utilizado para coletar grandes quantidades de dados em um ambiente distribuído. Sistemas ETL são utilizados para extrair dados de diferentes fontes, transformá-los caso seja necessário e então estes dados são enviados para que sejam armazenados. Ainda dentro do propósito de padronizar como o sistema troca mensagens entre suas diferentes tecnologias, o Apache Nifi se comunica através de mensagens no formato Chave/Valor, assim como o formato JSON. A finalidade deste software é controlar o fluxo de dados proveniente da camada de infraestrutura, para que estes dados possam então ser armazenados. Ainda conforme o trabalho feito em (KIM; LEE; GO, 2019) o software possui diversas vantagens, tais como:

- Código aberto, sendo este sem custo para uso;
- Possui uma UI própria, facilitando o desenvolvimento e a visualização do fluxo de dados desejado;
- Possui variados recursos para o controle do fluxo de dados, como: sistema para controle de prioridade, controle de vazão, entre outros;
- É possível também realizar processamento em tempo real, desde que não sejam muito complexos, para maior complexidade é necessário a integração com outro sistema;

Apesar das vantagens provenientes do uso deste Software é preciso considerar que este agrega mais complexidade ao sistema, o que acarreta uma maior dificuldade na implementação do Framework, assim como também é necessário alocar hardware

para o uso desta tecnologia. Porém, visando uma maior robustez do sistema, assim possibilitando maior escalabilidade, é visto como necessário utilizar o Apache Nifi neste Framework.

4.1.3 Armazenamento

Após todo sistema de aquisição de dados ser descrito, e como este foi projetado para se comunicar com o restante do Framework é necessário definir como as informações coletadas pela camada de infraestrutura serão armazenadas. De modo geral, é possível que as informações sejam armazenadas de duas formas, localmente em servidores dos campus ou utilizando algum serviço de nuvem.

Em sistemas que empregam exclusivamente um serviço de armazenamento local faz-se necessário o uso de servidores disponíveis no campus para que os dados coletados ali sejam armazenados. Esta escolha de projeto pode evitar gastos com serviços de armazenamento em nuvem. Por outro lado, é preciso considerar que outros campus possivelmente não têm acesso a tais informações. Portanto, é preciso determinar se os dados que estão sendo coletados precisam ser utilizados por outros campus além deste que os coletou. Também é possível utilizar um modelo híbrido, onde é armazenado em nuvem somente aquilo que seja necessário compartilhar com outros locais, e o restante pode ser armazenado localmente, o que demanda um serviço de menor custo de armazenamento em nuvem. Caso seja necessário para um determinado contexto de uso do Framework o compartilhamento de todos os dados coletados, é necessário que estes estejam disponíveis em um serviço de nuvem, de modo a todos os campus terem acesso a estes dados. Independente de qual for o caso de uso de armazenamento, ilustrados na Figura 17, todas as opções tornam viáveis a escalabilidade do sistema, sendo possível aumentar a capacidade de armazenamento, independente de qual for a opção escolhida.

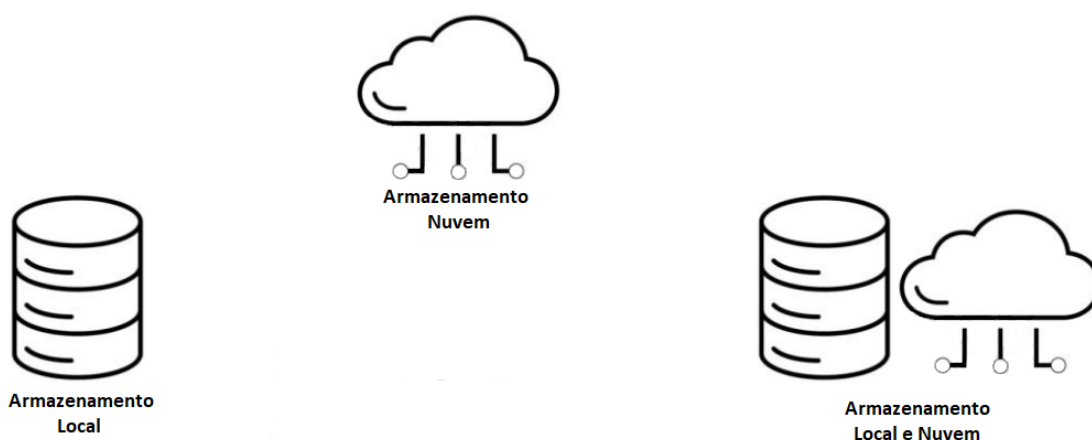


Figura 17 – Casos de uso de armazenamento.

Com relação às tecnologias aplicadas nesta camada, é proposto o uso do Apache Hive como tecnologia responsável pelo armazenamento e organização dos dados.

Conforme o trabalho feito em (HUI et al., 2014) o Apache Hive foi originalmente criado como uma camada de tradução que rodava em cima do MapReduce do Apache Hadoop, seu principal objetivo é facilitar o uso da plataforma do Hadoop, que por sua vez utiliza o modelo SQL como base de seu funcionamento. Com o uso do Hive é possível realizar as operações de busca no banco de dados através de um modelo de grafos direcionais acíclicos, ao contrário do funcionamento padrão do Hadoop que consiste em escrita de declarações de manipulação de dados. Logo, o Hive apesar de ter sua base de funcionamento em cima do Hadoop, permite ao usuário manipular os dados armazenados no sistema HDFS (Hadoop Distributed Filesystem) através de uma interface gráfica, aumentando significativamente a produtividade e facilitando o processo de adaptação ao utilizar esta tecnologia. Ainda conforme o que foi desenvolvido em (HUI et al., 2014) o Hive é amplamente utilizado em várias organizações como Facebook, LinkedIn, entre outros por seu excelente desempenho como aplicação para armazenamento e análise de big data. Assim como o Hadoop, o Apache Hive também é um projeto open source, que vai ao encontro visando reduzir custos ao projetar o Framework.

O Apache Hive pode ser utilizado como um serviço local ou online para a gerência e armazenamento dos dados do Framework, porém é preciso levar em consideração que ao utilizar um serviço na web é comumente atrelado a um custo, como um serviço em nuvem de armazenamento de dados. Como um dos principais objetivos do Framework é reduzir custos, se torna necessário avaliar na implementação se existe esta demanda de armazenamento em um banco de dados online, possibilitando que um campus seja este qual for, tenha acesso aos dados gerados pelos demais campus.

Outra preocupação ao utilizar o Apache Hive no Framework seria sua integração com a camada de Comunicação, que estaria enviando os dados para este, assim como também com a camada de Processamento que estaria fazendo o uso destes dados. Por se tratar de um banco de dados com base em SQL é possível fazer buscas como em qualquer outro sistema deste mesmo modelo, além do Apache Hive fornecer uma interface gráfica para facilitar este processo. Portanto, tanto para o armazenamento quanto para a leitura de dados no Hive é necessário que as camadas adjacentes sejam capazes de realizarem *queries* no formato SQL.

4.1.4 Processamento

Após todo sistema de captura e armazenamento de dados ter sido descrito previamente, é necessária uma camada responsável por computar estes dados e transformá-los em informação com relevância para determinado contexto. Esta camada faz acesso aos dados presentes na camada de Armazenamento para que possa realizar processos de extração, carregamento e transformação (ETL) dos dados. Além disso, é capaz de fornecer informações diretamente para a camada de Serviços caso

necessário, tendo em vista que algumas funções do Framework podem não demandar alguma etapa de Processamento, não fazendo parte do fluxo sempre haver essa demanda da camada de Serviços a esta etapa.

De modo a realizar estes processos de extração, carregamento e transformação, este Framework foi projetado utilizando o Apache Spark como responsável para tais atividades. Conforme o trabalho desenvolvido em (YANG et al., 2020) para o funcionamento do Spark é necessário um gerenciador de clusters e um sistema de armazenamento distribuído, o gerenciador de cluster dentre as opções disponíveis encontra-se o do próprio Spark e o do Hadoop, que vai ao encontro com a camada anterior, por esta utilizar o Apache Hive, que tem como sua base de funcionamento o Hadoop. Com relação ao sistema de armazenamento distribuído, o Spark pode ser integrado com vários sistemas, como Cassandra, Amazon S3 e também com o Hadoop, facilitando sua integração com o Framework. Apesar do Apache Hadoop ser capaz de realizar funções de análise e processamento de dados, este trabalho feito em (YANG et al., 2020) mostra que o Spark em certos cenários é capaz de obter uma performance até 10 vezes melhor que o Hadoop para estas tarefas. Outro ponto que deve ser considerado que também foi discutido pelos autores é o fato do Spark também ser adequado para integração com aplicações de aprendizado de máquina, além de ser capaz de se comunicar diretamente com serviços frontend caso necessário.

Apesar das vantagens presentes ao integrar o Spark com o Framework é necessário apontar que isto também traz desvantagens, conforme o trabalho feito em (VILLEGAS-CH et al., 2019) o Apache Spark tem um maior custo de implementação que o Hadoop/Hive, devido à infraestrutura que este demanda.

Outro ponto importante destacar é que o Apache Spark, assim como o Hive trata-se um software open source, o que viabiliza a construção de um Framework com preocupações relacionadas a baixo custo em seu desenvolvimento. Por fim, é necessário considerar que o Apache Hadoop é capaz de realizar funções como processamento e análise de dados que o Apache Spark estaria desempenhando neste Framework, porém no trabalho feito em (YANG et al., 2020) é observado que o tempo de processamento difere muito entre esses dois softwares para a execução destas tarefas, e isso torna-se cada vez mais evidente conforme aumenta a quantidade de dados, com o Spark realizando o processamento em um tempo até 10 vezes menos que o Hadoop para uma certa quantidade de dados. Portanto, considerando as premissas de escalabilidade desejadas neste Framework foi feito o projeto considerando o uso do Spark, apesar da complexidade adicional agregada por este.

4.1.5 Serviços

De modo que estes dados coletados, armazenados e então analisados tenham alguma função prática, é necessário que sejam utilizados das mais variadas formas,

para que o objetivo do Framework, de melhorar a qualidade de vida no campus, possa ser atingido. Alguns destes serviços foram previamente descritos em detalhes juntamente com a camada de Infraestrutura, especificamente os que tinham uma conexão mais direta com a camada em questão, pois interagem diretamente com os dispositivos IoT. Porém, alguns outros recursos projetados demandam uma interação maior com a camada de Processamento e/ou Armazenamento, estes são descritos a seguir.

- **Dashboard:** Com o intuito de disponibilizar a comunidade acadêmica todos os dados relevantes que foram coletados, este serviço trata-se de uma plataforma que pode ser acessada na web ou através de um aplicativo, onde todos dados obtidos são disponibilizados após sofrerem um processo de filtragem, de modo a mostrar somente o que for relevante. Os dados podem ser visualizados das mais diferentes formas, como gráficos, planilhas, entre outros. De maneira geral é natural concluir que a comunidade de forma geral possa se interessar em sua maior parte por dados como, localização dos veículos de transporte de apoio, feriados e ponto facultativos, entre outros, mais pertinentes aos estudantes. Por sua vez os dados com relação ao consumo de recursos, ocupação de salas e laboratórios, entre outros, possam ser de maior interesse do setor administrativo, de qualquer forma estes podem ocupar o mesmo ambiente, desde que sejam devidamente organizados.
- **Sugestões Administrativas:** Com os dados coletados e devidamente processados pelas ferramentas disponíveis no Framework, como o Apache Spark e os mais variados algoritmos de aprendizado de máquina que forem relevantes e necessários para o contexto de uso, torna-se viável a função de sugestões pelo próprio Framework para os usuários. Alguns exemplos práticos disso, seria o sistema detectar ambientes que estejam consumindo energia elétrica sem necessidade, como corredores sem movimento com luzes ligadas, salas e laboratórios que estariam sem ocupação mesmo havendo demanda por parte do campus naquele momento, entre outros.
- **Avaliações de Ensino:** Através de algoritmos de aprendizado de máquina e dos dados coletados das avaliações dos estudantes e professores, é possível correlacionar estes com as mais variadas características dos ambientes onde estes estão inseridos, de modo a encontrar possíveis melhorias no âmbito do Aprendizado Inteligente.
- **Disponibilizar dados para estudos:** Além de todas as melhorias de qualidade de vida possíveis de serem alcançadas com o uso do Framework, também é viável utilizar da grande quantidade de dados que são coletados para que outros estudos possam ser realizados. O objetivo com este serviço é disponibilizar os

dados que foram coletados para que outras pessoas/grupos de pesquisa possam realizar estudos tendo como base o contexto da UFPel, que nesta proposta é onde o Framework estaria sendo utilizado. Os dados seriam disponibilizados da forma como foram obtidos, logo após terem sido armazenados através do Apache Hive, podendo ter a opção de disponibilizar estes antes e depois da etapa de Processamento, conforme fosse a necessidade de quem iria utilizá-los.

Além destes serviços descritos é importante salientar os outros que fazem parte do projeto do Framework e já foram mencionados juntamente com a camada de Infraestrutura, sendo estes: Sistemas Autônomos, Transporte e Estacionamento Inteligente, Controle de Acesso Inteligente e Sistemas de Segurança. Também é necessário apontar que conforme as demandas surgirem é possível fazer as mais diversas adaptações no sistemas para que novos serviços possam ser fornecidos, pois um dos objetivos deste Framework é sua capacidade de escalabilidade e todas as tecnologias presentes foram selecionadas por apresentarem tais características, além de outros fatores.

4.2 Considerações sobre o Framework Proposto e Comparativos

A partir deste ponto, todas as tecnologias do Framework foram descritas em suas respectivas camadas. Na Figura 18 é possível ter uma visão geral das descrições feitas nas seções anteriores, assim como estas estão conectadas nesta proposta.

Apesar de todas as camadas terem sido detalhadamente descritas, são necessárias algumas considerações, principalmente relacionadas ao que se espera do comportamento prático deste Framework. Como foi descrito durante a seção referente a camada de Infraestrutura, alguns dispositivos podem ser conectados a um Broker MQTT de modo a facilitar a comunicação entre os dispositivos desta camada, porém nem todos componentes desta camada podem ter esta necessidade, portanto seu uso dependerá do contexto onde estão inseridos os microcontroladores/sensores/atuadores.

Outro ponto importante a ser destacado é que alguns dos Serviços presentes nesta proposta podem estar interagindo diretamente com o Apache Hive sem a necessidade do Apache Spark. Um exemplo prático disto seriam algumas informações que podem estar contidas no Dashboard, o Frontend estaria fazendo requisições ao Apache Hive sem que estes dados tenham sido previamente processadas pelo Spark, este é um cenário que depende do contexto de uso, porém é um caso prático do comportamento descrito na Figura 18, onde a etapa de Armazenamento está diretamente conectada a etapa de Serviços. Outro cenário que pode ser usado para exemplificar esta conexão direta entre estas duas camadas, seria o caso onde os dados são processados pelo Apache Spark e ao invés de diretamente serem utilizados por algum Serviço,

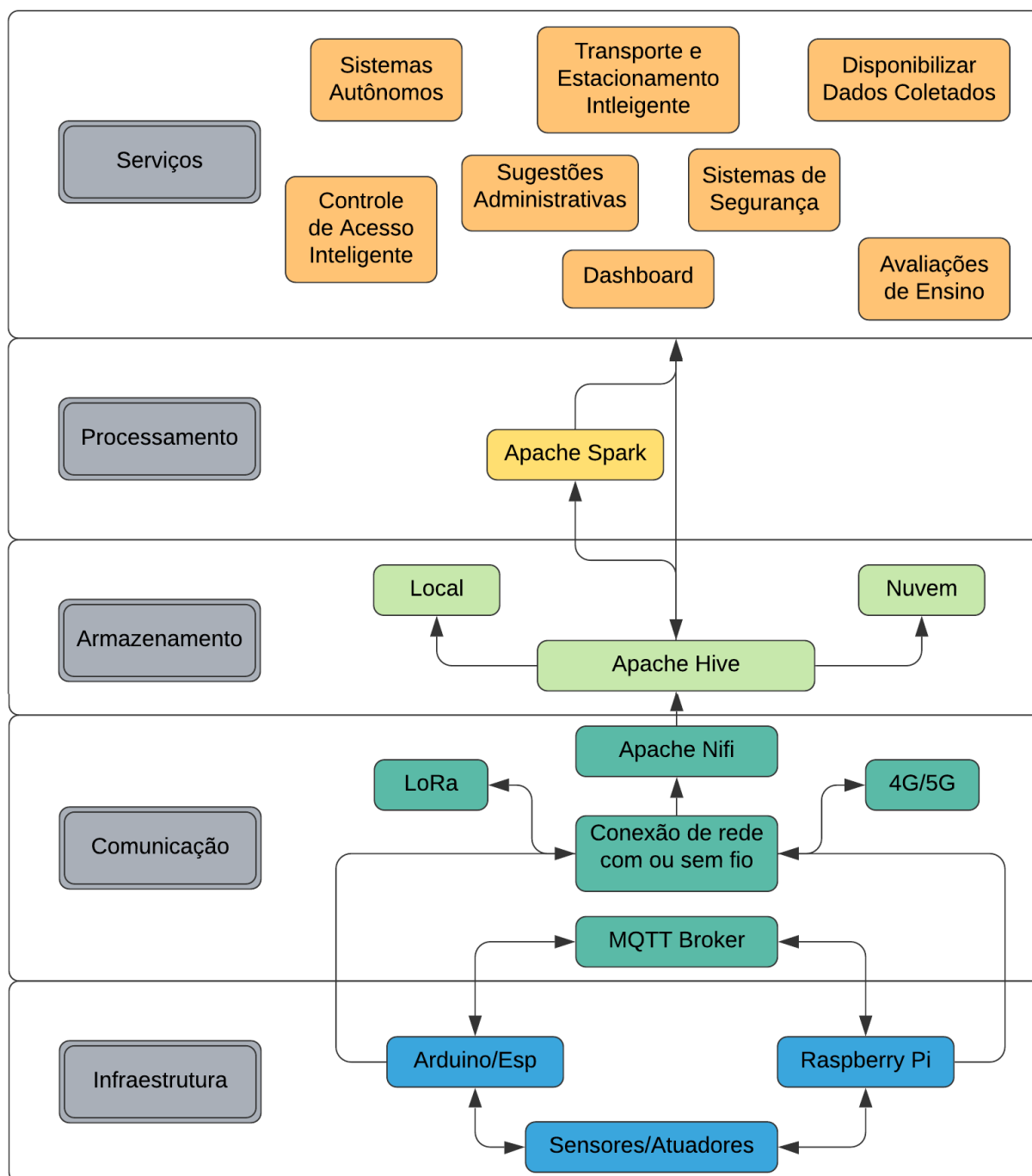


Figura 18 – Camadas do Framework com suas tecnologias.

estes são armazenados novamente no Apache Hive para que possam ser utilizados posteriormente.

De modo a facilitar a visualização de como este trabalho está inserido nesta área de pesquisa, foram selecionados outros trabalhos semelhantes desenvolvidos no contexto de Framework para Campus Inteligente com o intuito de comparar as tecnologias utilizadas e avaliar possíveis vantagens/desvantagens em relação aos trabalhos escolhidos.

O primeiro trabalho (Trabalho 1) escolhido para ser comparado a esta proposta

trata-se do que foi proposto em (LIU; MA; JIN, 2018). O trabalho utiliza tecnologias IoT para coletar e analisar dados de forma compreensiva de professores e alunos em um campus de uma academia militar. Com o uso de dispositivos capazes de coletar dados e softwares como o Apache Spark para a análise destes, o trabalho visa aumentar a supervisão do treinamento educacional, melhorar a precisão dos métodos de ensino, assim como a eficiência e transparência do sistema administrativo do campus.

O trabalho proposto em (DEBAUCHE et al., 2020) (Trabalho 2) também foi escolhido para esta comparação entre Frameworks, é apresentada uma solução para Campus Inteligente baseada no uso de tecnologias como ESP32 e sensores de baixo custo para coletar os dados. Para comunicação é proposto uso de protocolo Wi-Fi em ambientes internos e LoRaWan para áreas externas. O objetivo é que se consiga desenvolver um ambiente onde os recursos sejam utilizados de forma mais efetiva e ofereçam melhorias na qualidade de vida de toda Universidade.

Por fim, o último trabalho (trabalho 3) utilizado para comparar os Frameworks foi o desenvolvido em (VILLEGAS-CH et al., 2019). Esta proposta busca uma forma de prototipação em uma pequena escala um ambiente de Cidade Inteligente, conforme o autor um ambiente em pequena escala que poderia ser utilizado para tal seria um campus de universidade, desta forma é feito um trabalho que integra tecnologias que poderiam ser utilizadas para construção de uma Cidade Inteligente, porém em um campus. Dados são obtidos através de tecnologias IoT, e para a centralização e análise de dados são utilizados softwares e serviços de big data.

É possível observar na Tabela 6 as diferentes tecnologias presentes em cada camada, desta proposta de Framework para Campus Inteligente e os outros 3 trabalhos semelhantes que foram escolhidos.

Como é possível observar na Tabela 6 os trabalhos apresentam diversos pontos em sua construção onde diferem em suas tecnologias. Inicialmente tratando da camada de Infraestrutura, onde de maneira geral estes não variam muito por se tratarem de dispositivos IoT que são responsáveis pela captura de dados e interações diretas com o ambiente onde estão inseridos. Apesar do trabalho proposto nesta dissertação não citar explicitamente tecnologias nesta camada como sensor biométrico, câmeras de vigilância, entre outros, é possível realizar a integração destes ao Framework, devido a suas características de independência de tecnologias.

Com relação à camada de Comunicação, é possível notar que os trabalhos se distinguem quanto às escolhas de tecnologia. O Trabalho 2 apresenta o uso de somente conexão de rede com fio, pois segundo os autores, ao se tratar de um campus de academia militar há uma preocupação em relação a questões de segurança ao se utilizar redes sem fio para enviar/receber os dados. Também é possível observar que o trabalho proposto nesta dissertação apresenta semelhanças com relação ao Trabalho 3 na escolha de tecnologias para esta camada, porém o Trabalho 3 cita o uso de

Tabela 6 – Comparativo de Framework e suas tecnologias

Camada	Framework UFPel	Trabalho 1	Trabalho 2	Trabalho 3
Serviços	Sistemas Autônomos, Controle de Acesso Inteligente, Sugestões Administrativas, Dashboard, Sistemas de Segurança, Avaliações de Ensino, Transporte e Estacionamento Inteligentes, Disponibilizar Dados Coletados	Campus One Card(RIF), Suporte Logístico, Avaliações de Ensino e Aprendizado, Avaliações Administrativas	Controle Autônomo de Ventilação, Redução no Consumo de Energia Elétrica	Redução no Consumo de Energia Elétrica, Melhorias no Sistema de Segurança, Controle de Acesso Automatizado
Processamento	Apache Spark	Apache Spark	Odroid N2, Nvidia Jetson	Apache Hadoop
Armazenamento	Apache Hive, Local e/ou Online	Apache Hadoop, Tachyon	Raspberry Pi 4 SD Card	VNX3200, Cisco UCS MINI
Comunicação	MQTT Broker, Conexão de rede com e sem fio, LoRa, 4G/5G, Apache Nifi	Conexão de rede com fio	MQTT Broker, Conexão de rede com e sem fio, 4G, LoRa, BLE, NFC e Zigbee	N3k-DC, Conexão de rede com e sem fio
Infraestrutura	Arduino, Esp, Raspberry Pi, Sensores e Atuadores Diversos	RFID, Câmeras de Vigilância, Sensores e Atuadores Diversos	ESP32 Wroom, ESP32-CAM, ESP32 Lora, Sensores e Atuadores Diversos	Sensor Biométrico, RFID, Câmeras de Vigilância, Sensores e Atuadores Diversos

algumas tecnologias que não são tratadas nessa proposta aqui apresentada, como, por exemplo, BLE, NFC e Zigbee. Contudo, caso seja necessário a inserção destas tecnologias no Framework seria viável o seu uso, tendo em vista o funcionamento do sistema como um todo, os dispositivos poderiam se comunicar por BLE, por exemplo, caso esta seja uma opção melhor para determinado contexto. De maneira geral este trabalho aqui proposto e o Trabalho 3 apresentam mais opções para formas de comunicação entre o sistema, viabilizam mais facilmente o uso de dispositivos em ambiente externo com difícil acesso a conexões de rede e conexões diretas entre os dispositivos IoT.

A camada de Armazenamento também apresenta distinções entre as tecnologias utilizadas pelos trabalhos. O trabalho desta dissertação utiliza uma tecnologia semelhante ao que foi escolhido no Trabalho 1, porém ao utilizar o Hive ao invés do Hadoop são obtidas algumas vantagens, como, por exemplo, a facilidade para realizar buscas no banco de dados por estar utilizando uma interface gráfica para tal. O Trabalho 2 até o momento vinha apresentando maiores semelhanças com o que está sendo proposto nesta dissertação, porém para esta camada a tecnologia escolhida foi a Raspberry Pi 4 para armazenar os dados, nos cartões de memórias acoplados a esta, conforme o autor deste trabalho o problema gerado por isso é o tempo de vida útil que estes cartões de memória tem até atingirem sua capacidade máxima, que seria em torno de 6 meses, considerando o contexto do trabalho ali feito. Ao utilizar tecnologias como o Apache Hive/Hadoop é possível expandir este tempo limite, considerando que estes podem ser utilizados em sistemas com uma capacidade muito maior de armazenamento, além de também ser possível utilizar serviços em nuvem para salvar os dados, assim como descartar dados muito antigos caso seja necessário para determinado contexto. Também é importante considerar a complexidade adicionada ao Framework por utilizar uma tecnologia como Apache Hive, porém dada a preocupação com a escalabilidade do sistema optou-se pela escolha desta ferramenta.

Com relação à camada de Processamento os trabalhos apresentam similaridades, com exceção ao Trabalho 2 que não utiliza nenhuma ferramenta da empresa Apache. O Trabalho 3 opta por utilizar o Hadoop em sua etapa de Processamento, conforme os autores este se encaixa melhor no contexto de uso de sua proposta por ter um menor custo de implementação que o Apache Spark, porém para o trabalho aqui apresentado é necessário considerar as características de escalabilidade desejadas no Framework, e uma potencial abundância de dados, tais fatores levaram a escolha do Apache Spark por ser capaz de um processamento muito mais rápido que o Hadoop, principalmente ao se considerar maiores quantidade de dados.

Por fim, na camada de Serviços é possível notar que o Framework proposto neste trabalho contém alguns itens a mais se comparado aos demais Frameworks presentes na Tabela 6, isto ocorre devido a esta proposta buscar trazer melhorias na qualidade

de vida nos mais variados setores do campus, além de capacidade para que mais Serviços possam ser implementados caso haja necessidade.

De maneira geral, os 4 Frameworks apresentados na Tabela 6 possuem várias similaridades nas mais variadas camadas, assim como suas particularidades com escolhas de tecnologias, visto que cada um está sendo proposto para determinado contexto. Este Framework aqui proposto, apesar de em alguns pontos agregar complexidade a sua estrutura, se comparado a alguns outros da Tabela 6, da mesma forma oferece maiores opções de escalabilidade em determinados aspectos, assim como preocupações referentes ao uso de uma grande quantidade de dados desde sua implementação, além de oferecer liberdade de escolhas de tecnologias, principalmente na camada de Infraestrutura, contanto que estas respeitem os protocolos de comunicação.

Além disso, é importante destacar que foram utilizados três trabalhos para comparar tecnologias, pois utilizando os resultados do Mapeamento Sistemático somente estes apresentavam descrições completas de todas suas camadas e respectivas tecnologias. Foram encontrados muitos trabalhos que descreviam somente algumas camadas, mas não todo o Framework e como este conectava toda sua estrutura, assim como muitos também apresentavam as camadas e suas conexões, porém sem especificar tecnologia. Como esta proposta buscou descrever todas camadas, suas interações e tecnologias, de modo que pudessem ser feitas comparações, foram selecionados trabalhos que apresentaram formas semelhantes de especificar seu Framework, e estes foram utilizados na Tabela 6.

Também é importante citar algumas definições de uso prático do Framework, de modo que facilite a implementação do sistema utilizando as tecnologias que foram definidas. Na camada de infraestrutura é utilizado um modelo padrão de arquivo JSON para a comunicação entre os dispositivos e o envio de mensagens para o restante do sistema, na Figura 19 é possível observar os campos deste arquivo JSON utilizado como base para comunicação. Os campos presentes são: `sensorId`, que representa o identificador único daquele dispositivo, sendo este definido no momento de instalação e configuração do dispositivo, o campo `content` que representa o conteúdo que está enviado durante a comunicação, o `createdAt` que contém a data e hora de criação daquele conteúdo que foi obtido e por fim o campo `additionalData`, que se trata de um objeto que pode ou não conter campos adicionais, sendo estes quaisquer informações adicionais que sejam necessárias no contexto de uso do dispositivo que está enviando a mensagem, na Figura 20 é possível ver um exemplo prático deste modelo base sendo utilizado, onde este usa o campo `additionalData`. Neste exemplo foram adicionados dois campos para a composição da mensagem, o campo `campus`, onde neste caso representa qual campus da UFPel estaria localizado o dispositivo que envia a mensagem, e o campo `sala` que representa o número da sala onde está

localizado o dispositivo. É importante notar que estes campos são opcionais, caso não seja necessário estes podem não serem utilizados, assim como caso haja necessidade mais campos podem ser adicionados ao objeto `additionalData` com quaisquer informações que forem julgadas pertinentes ao contexto de uso.

```
1 {  
2   "sensorId": "",  
3   "content": "",  
4   "createdAt": "",  
5   "additionalData": {}  
6 }
```

Figura 19 – Formato base de arquivo JSON

```
1 {  
2   "sensorId": "001",  
3   "content": "670",  
4   "createdAt": "2021-01-10 16:25:18",  
5   "additionalData":  
6   {  
7     "campus": "Anglo",  
8     "sala": 415  
9   }  
10 }
```

Figura 20 – Exemplo de uso prático de arquivo JSON

Por fim, na camada de infraestrutura, de modo a configurar novos elementos e adicioná-los ao Framework é necessário que estes tenham um identificador único, para que este possa ser adicionado a mensagem do arquivo JSON, este identificador fica a critério do responsável por configurar o dispositivo no momento de sua instalação e configuração, recomenda-se que seja utilizado algum padrão de identificação para todos os elementos que compõe o sistema, de modo a facilitar posteriormente o acesso aos dados deste dispositivo, ao fazer a busca por seu identificador no banco de dados.

Na camada de comunicação para utilização do Apache Nifi também fica a critério do responsável por configurá-lo e definir os nós do sistema e quais sensores utilizam cada nó, porém da mesma forma recomenda-se o uso de um padrão para todo sistema. Uma possível forma de realizar esta configuração seria agrupar todos os sensores de uma sala em um nó, sendo cada nó do sistema composto por somente

sensores da mesma sala, buscando organizar e padronizar de forma simples todo Framework. Da mesma forma, o MQTT Broker terá o nome dos tópicos definidos no momento de sua configuração, e tal qual o Apache Nifi também é recomendado um modelo padrão para o projeto, onde, por exemplo, os tópicos poderiam ser nomeados conforme as salas, ou andares que estes realizam a comunicação. Por fim, espera-se que a partir destas informações a implementação do Framework com as tecnologias que foram propostas seja facilitada.

5 RESULTADOS

Este capítulo apresenta o caso de uso do Framework proposto realizado um estudo de caso, utilizando como base a infraestrutura descrita neste trabalho.

5.1 Estudo de Caso

O estudo de caso visa verificar, de uma forma simplificada, como é a implementação prática do Framework aqui proposto e desta forma, não foram utilizadas todas as tecnologias descritas nas camadas. Buscou-se fazer o uso de ferramentas que utilizam os mesmos protocolos de comunicação e estas são integradas de forma semelhante, de modo a ilustrar como seria o comportamento do projeto montando este com tudo que foi descrito.

Como a ideia com este estudo de caso é ilustrar de forma comportamental, mesmo que simplificada, todas etapas do Framework, foi implementado um pequeno protótipo que usa todas as camadas, assim como a integração entre estas utilizando as formas de comunicação que foram propostas.

Este estudo de caso tinha como objetivo utilizar um sensor de luz para verificar ao longo do dia a variação de luz em uma sala do campus Anglo da UFPel. A cada 30 segundos era feita uma leitura do valor de luminosidade com este sensor e enviado a um microcontrolador. Este microcontrolador por sua vez fazia requisições a uma base de dados que havia sido criada localmente em um computador no campus Anglo, de modo que estas leituras pudessem ser salvas, contendo como dado o identificador único do sensor, a data e hora da leitura e o valor que havia sido coletado. Além disso, foi feito um servidor Frontend para que os valores pudessem ser acessados através de um Dashboard bem simples, que faria requisições a este mesmo servidor da base de dados e iria mostrar tudo que foi coletado de diversas formas em uma página web, este também estaria instanciado localmente em um computador do campus Anglo. É importante observar que devido ao fato do servidor estar sendo hospedado localmente há uma limitação para seu acesso, este foi configurado de modo que só pudesse ser acessado por dispositivos conectados a mesma rede que este prove acesso. Portanto,

o microcontrolador utilizado e caso mais fossem utilizados, deveriam estar todos conectados a rede onde o dispositivo que hospeda o servidor Backend esta conectado.

Por fim, é possível utilizar os dados coletados em algoritmos de aprendizado de máquina, esta etapa visa somente ilustrar que como os protocolos de comunicação foram seguidos, utilizando o envio de mensagens no formato JSON para os dados e salvando estes em bancos de dados relacionais SQL, é viável inserir algoritmos de inteligência artificial caso seja necessário. Os algoritmos aqui utilizados não visam buscar nenhuma inferência significativa, dada a origem dos dados e a baixa quantidade dos mesmos.

Na Figura 21 é possível observar um diagrama deste estudo de caso, contendo em cada camada a tecnologia utilizada e como esta se conecta com o restante do Framework. A etapa de Processamento não apresenta conexão com os Serviços, pois este estudo de caso buscou somente ilustrar a possibilidade de utilizar algoritmos de aprendizado de máquina de forma simples com os dados coletados, porém, estes dados não são suficientes para que alguma inferência significativa seja feita, portanto, não houve a necessidade para tal conexão.

5.2 Protótipo Desenvolvido

O protótipo é constituído de um sensor de luz Grove Light Sensor (Figura 22), que se trata de uma placa composta por fotoresistor e um circuito integrado amplificador operacional duplo LM358.

Esta placa que contém o fotoresistor foi conectada a placa NodeMCU V3 Lolin (Figura 23), um kit de desenvolvimento que utiliza o microcontrolador ESP8266. O pino referente ao sinal enviado pelo sensor de luz foi conectado ao pino A0 da NodeMCU, de modo que pudessem ser feitas leituras de valores analógicos pelo microcontrolador. De modo que o sensor pudesse receber tensão de alimentação, este foi conectado ao pino de 3v da placa NodeMCU e ao pino de Ground desta. Estes componentes são a representação da camada de Infraestrutura do Framework de forma simplificada, pois é composta de somente um sensor e um microcontrolador.

Com relação à camada de Comunicação foi utilizado conexão de rede sem fio para que as requisições fossem feitas pelo microcontrolador para o backend. O microcontrolador estava conectado a uma rede sem fio do campus Anglo e a cada 30 segundos recebia uma leitura do sensor de luz. Após transformar esses dados coletados em uma string JSON, era feita uma requisição HTTP para o servidor que estava conectado na mesma rede WiFi que a NodeMCU, deste modo o servidor recebia os dados e armazenava-os localmente.

De modo que os dados pudessem ser recebidos pela camada de Armazenamento, foi feito um servidor backend utilizando a plataforma NodeJS no editor de texto Visual

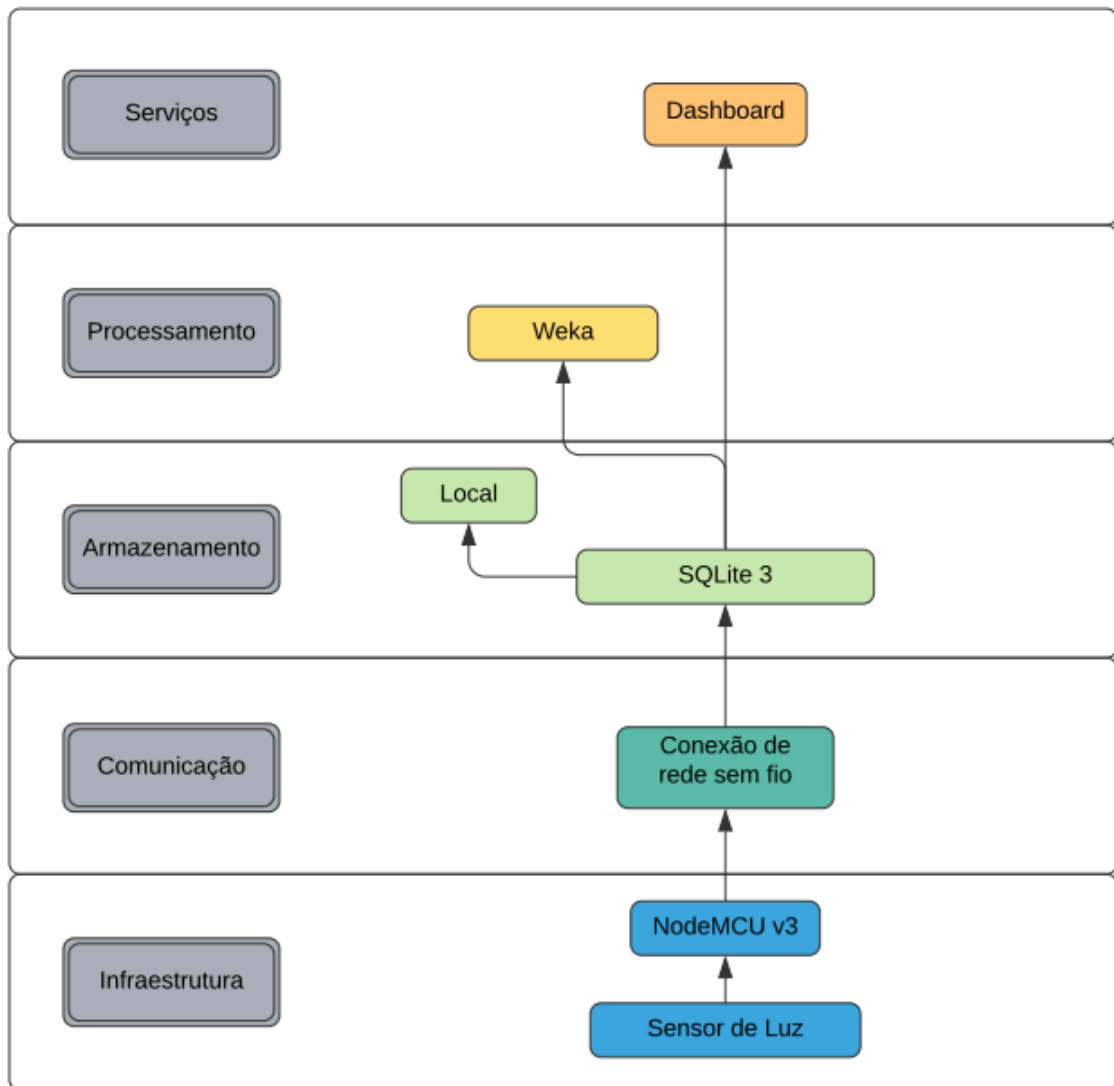


Figura 21 – Diagrama do Estudo de Caso

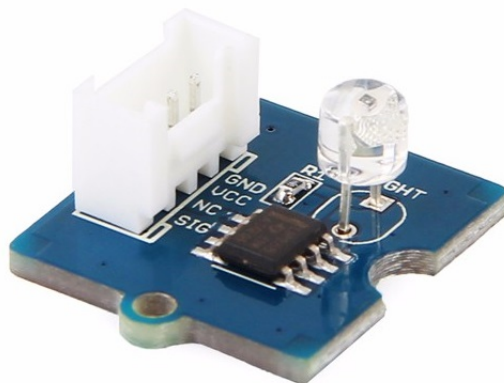


Figura 22 – Grove Light Sensor

Studio Code. Para que fosse implementado o banco de dados foi utilizada a biblioteca SQLite, de modo a ter acesso a um banco de dados SQL, o qual seria utilizado pelo microcontrolador para armazenar os dados e pelo servidor Frontend para a consulta do que havia sido coletado. O banco de dados foi projetado com 4 colunas para as infor-

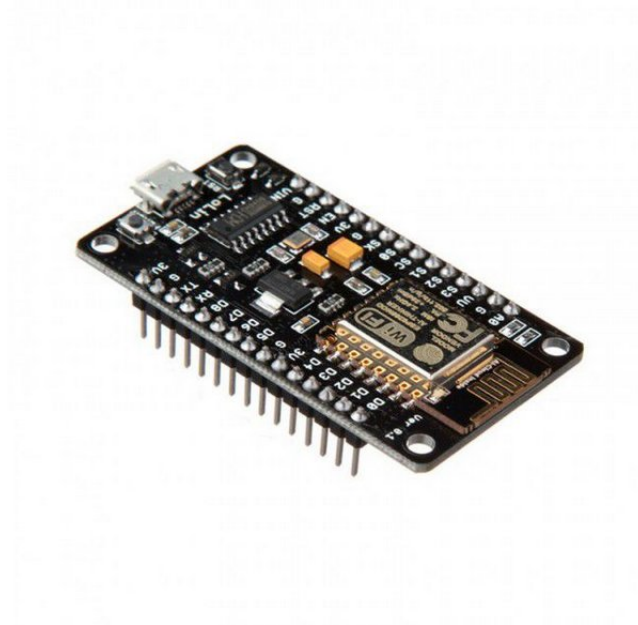


Figura 23 – NodeMCU V3 Lolin - ESP8266

mações (Figura 24), um campo destinado a identificar qual era a linha daquele, dado dentro da base de dados, o identificador do sensor, neste caso é sempre o mesmo, mas poderiam ser adicionados novos sensores a rede caso necessário, a coluna que continha o valor lido, nesse caso de temperatura, e por fim o campo onde era salvo a data/hora da leitura. Por fim, de modo a realizar testes para com as requisições do banco de dados foi utilizado o software Insomnia, o qual consiste em uma ferramenta que visa testar aplicações Web através do envio de requisições e análise da resposta.

id	sensorId	content	createdAt
22	001	668	2021-11-10 16:24:47
23	001	670	2021-11-10 16:25:17
24	001	675	2021-11-10 16:25:48
25	001	675	2021-11-10 16:26:18
26	001	675	2021-11-10 16:26:48
27	001	670	2021-11-10 16:27:18
28	001	675	2021-11-10 16:27:48
29	001	675	2021-11-10 16:28:18

Figura 24 – Colunas do banco de dados

A etapa de Processamento foi feita somente a ilustrar a possibilidade de extrair os dados do banco de dados e utilizá-los em algum software de aprendizado de máquina. Pois, devido às colunas que estão presentes na tabela, realizar algum tipo de inferência nos dados, como, por exemplo, tentar alguma forma de classificação, não traria resultados significativos. Desta forma o estudo de caso buscou somente testar

se é possível utilizar os dados neste formato que foram armazenados de forma prática, e caso houvesse a necessidade do uso de algoritmos de aprendizado de máquina isto seria possível. Também buscou-se ilustrar de forma simplificada como funcionaria o Framework implementado com as tecnologias propostas, pois estas utilizam deste formato de dados para se comunicarem.

Inicialmente foi exportado todo conteúdo da base de dados no formato CSV para facilitar o uso em algoritmos de aprendizado de máquina, devido ao fato destes serem armazenados em uma tabela relacional, neste caso o SQLite 3, basta somente exportá-los usando comandos nativos da biblioteca. Foi escolhido o Weka como software para ilustrar o processamento desses dados, devido a sua interface de fácil uso, a busca por somente verificar se este poderia lidar com o formato dos dados que estavam sendo utilizados de forma prática e por receber como entrada formatos semelhantes ao Apache Spark, como o CSV. Para que os dados possam ser utilizados no Weka estes precisam ter as colunas classificadas em tipos, como numéricos, categóricos, entre outros. Isso é possível ser feito no próprio Weka ou também antes de enviar os dados para este, este estudo de caso optou por converter o formato CSV em ARFF, um formato aceito pelo Weka que traz no arquivo a classificação destas colunas em tipos, facilitando o processo de importar os dados pelo software.

Na Figura 25 é possível observar os dados carregados na interface do software Weka, a esquerda (Figura 26) temos as colunas que são consideradas atributos para que possam ser utilizadas nos algoritmos de aprendizado de máquina, além de algumas informações gerais sobre o dataset, como o número de instâncias. Neste caso o atributo 0 é a coluna referente a data/hora dos dados, o atributo 1 é o identificador do sensor, neste caso assume sempre valor 1, e o atributo 2 que está selecionado é referente aos valores coletados. A direita mais acima (Figura 27) estão contidas as informações referentes ao atributo selecionado, neste caso o atributo 2. Dentre as informações podemos observar algumas como o valor mínimo e máximo que este assumiu ao longo de todos os valores coletados. Por fim, na direita mais abaixo (Figura 28) o software mostra a distribuição de ocorrência dos valores, porém como existem muitos valores distintos isso é mostrado no gráfico através de faixas de valores, e o número de ocorrências nestes intervalos. Os maiores valores se concentram nos pontos extremos, onde havia uma grande intensidade de luz, representado pelos maiores valores, ou muito pouca intensidade, representada pelos valores menores. Os valores presentes no meio representam a transição percebida pelo fotoreistor conforme havia variação de luz.

A partir deste ponto, caso houvesse uma coluna de interesse para realizar algum tipo de inferência, bastava que fossem utilizadas as ferramentas presentes no Weka, de modo que pudesse ser aplicado algum dos algoritmos de aprendizado de máquina presentes.

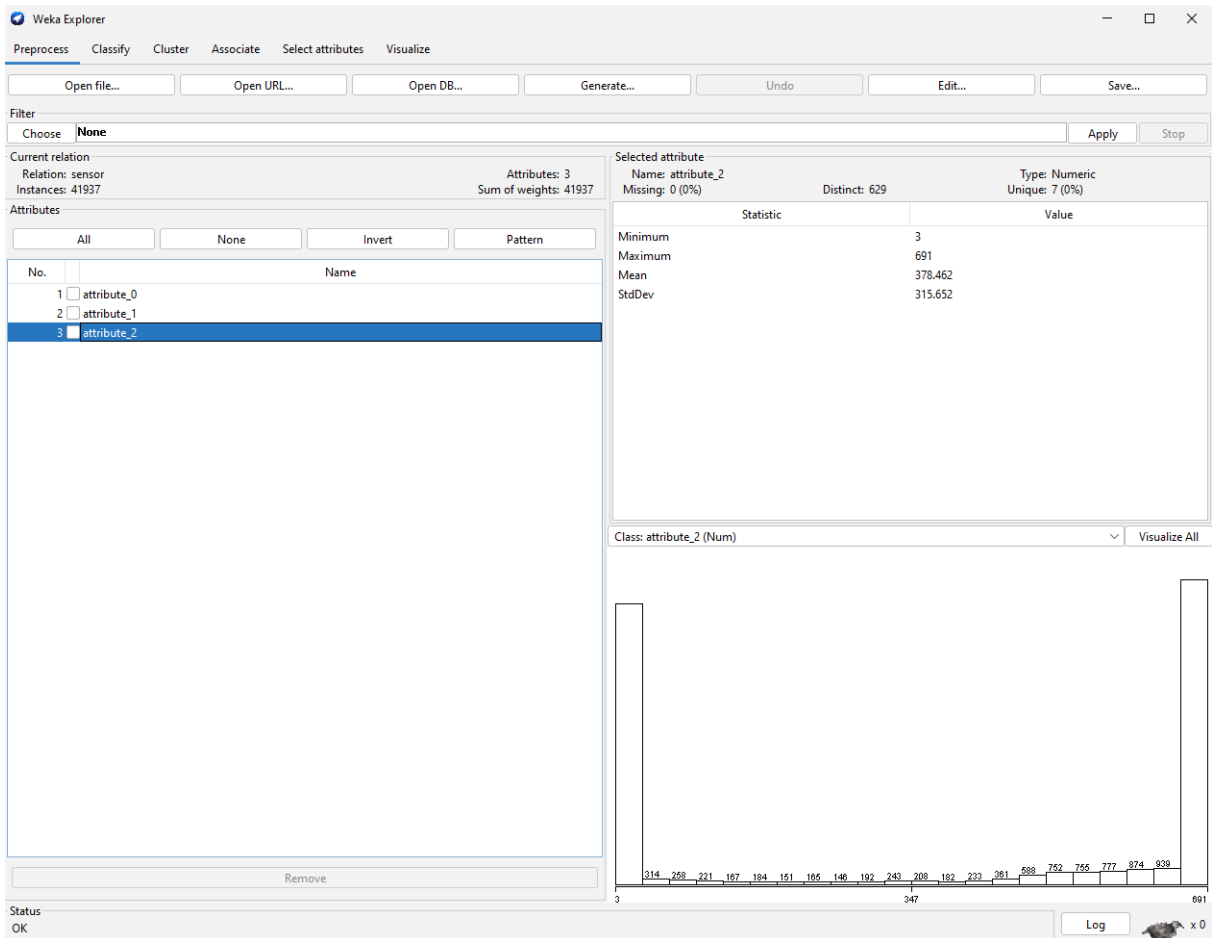


Figura 25 – Dados carregados no Weka

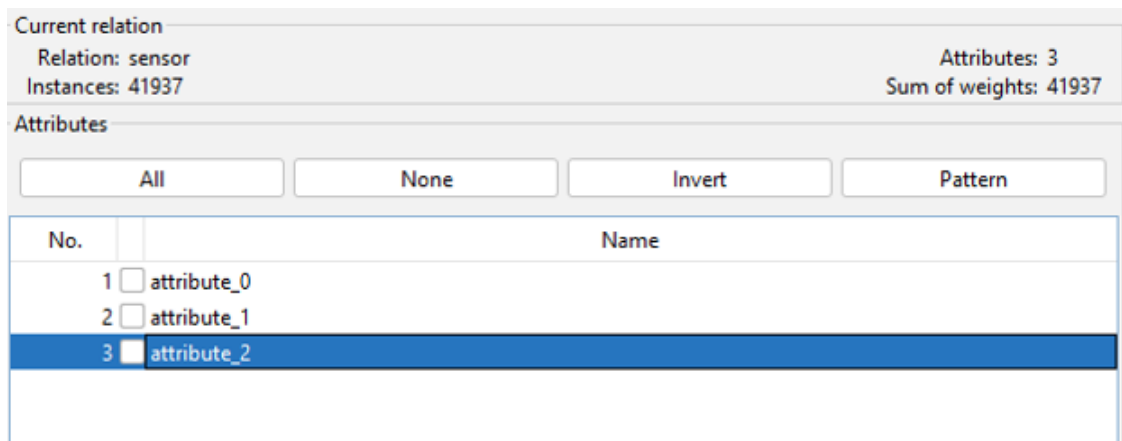


Figura 26 – Atributos dos dados

Por fim, para ilustrar a camada de Serviços, neste estudo de caso foi desenvolvido um Frontend bem simples, que fosse capaz de somente carregar os dados do backend e disponibilizá-los de algumas formas mais visuais aos usuários finais. Este Dashboard foi construído utilizando a plataforma do NodeJS, assim como o backend, também utilizando o editor de texto Visual Studio Code. A biblioteca escolhida para o desenvolvimento foi o React e utilizou-se da linguagem Javascript para todo o código.

O Dashboard (Figura 29) foi dividido entre 3 grandes áreas de informações, à es-

Selected attribute		
Name: attribute_2		Type: Numeric
Missing: 0 (0%)		Unique: 7 (0%)
Distinct: 629		
Statistic	Value	
Minimum	3	
Maximum	691	
Mean	378.462	
StdDev	315.652	

Figura 27 – Atributos dos dados

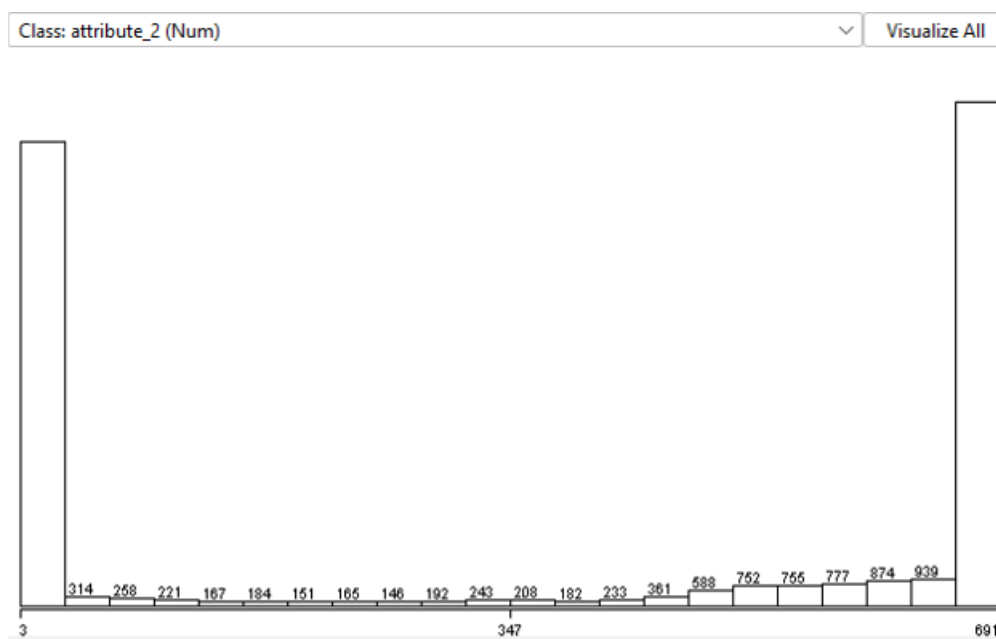


Figura 28 – Distribuição dos valores do atributo

querda estão contidas as localizações onde estariam sendo feitas coletas com o Framework (Figura 30), os que constam ali estão somente para ilustrar como isso poderia ser feito, pois este estudo de caso somente utilizou do Campus Anglo para coletar dados, os demais campus estão presentes de modo a representar uma das formas como este Dashboard poderia ser desenvolvido, onde o usuário poderia navegar entre os campus e verificar os dados coletados. Na parte central (Figura 31) estão contidas algumas informações, tais como: Número de sensores presentes no campus selecionado, Número de dados coletados, Data e hora da primeira coleta e Data e hora da

última coleta. Além destes dados também estão presentes gráficos, onde no histograma é apresentado a quantidade dos valores de luminosidade coletados em faixas específicas, e no gráfico de pizza é apresentado os dados coletados de luminosidades, porém somente em três faixas de valores. Esta coleta em específica foi feita no Campus Anglo, no 3º andar, em um laboratório de pesquisa da computação. Com relação à fonte destes dados é necessário pontuar que todas as informações ali presentes foram disponibilizadas pelo backend do sistema. É importante considerar que é possível utilizar as mais variadas formas de visualizações para os dados, e as que foram escolhidas neste Frontend buscam somente ilustrar como isso poderia ser feito de forma prática, pois a partir do momento que os dados são enviados pelo backend o formato como estes serão entregues ao usuário fica conforme as necessidades do contexto. A última seção do Frontend (Figura 32) mostra a lista de sensores presentes no campus selecionados, de modo que o usuário possa navegar entre eles para verificar os dados coletados por estes, assim como mostrar vários simultaneamente caso desejasse fazer algum tipo de comparação.

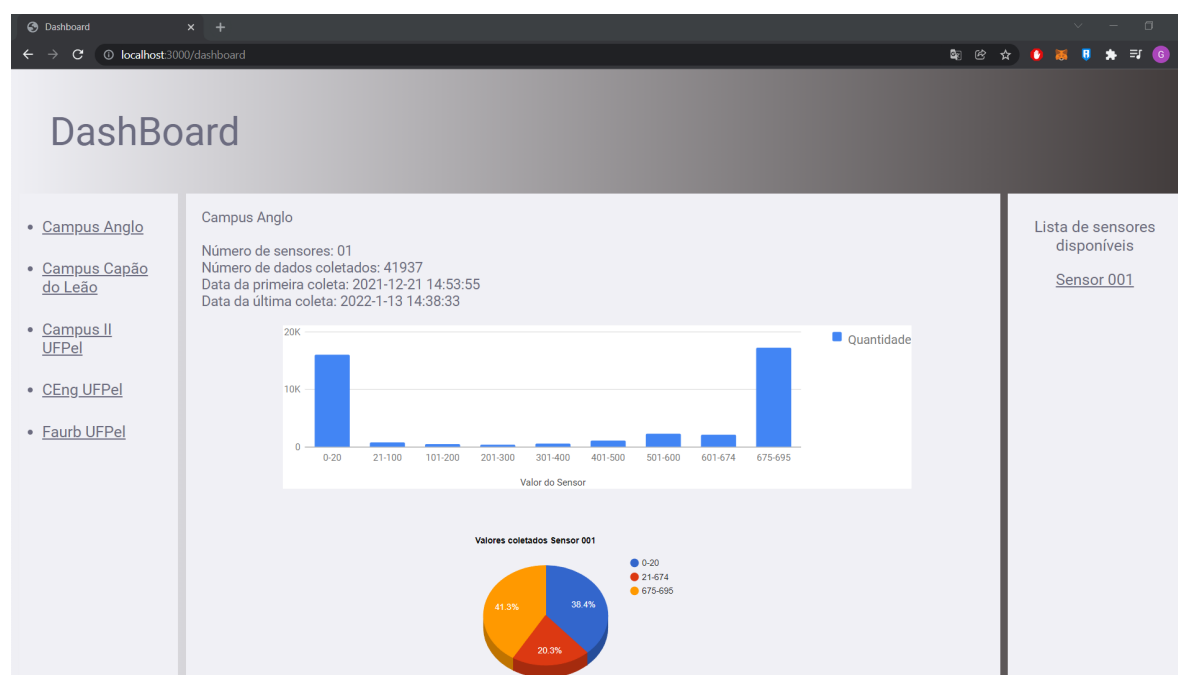


Figura 29 – Frontend Framework

Este estudo de caso, apesar de demonstrar de forma muito simples como estariam distribuídas de forma prática as camadas, foi desenvolvido utilizando como base todos os protocolos e princípios de funcionamento do Framework proposto, tais como: liberdade de escolha de tecnologia para coleta de dados, envio dos dados no formato JSON, armazenamento em um banco de dados no formato SQL, praticidade para que possam ser utilizados algoritmos de aprendizado de máquina, disponibilizar os dados para o usuário final, entre outros. Apesar de ter sido construído de forma simples, este estudo de caso apresenta opções de escalabilidade assim como proposto pelo

- [Campus Anglo](#)
- [Campus Capão do Leão](#)
- [Campus II UFPel](#)
- [CEng UFPel](#)
- [Faurb UFPel](#)

Figura 30 – Lista de Campus Frontend

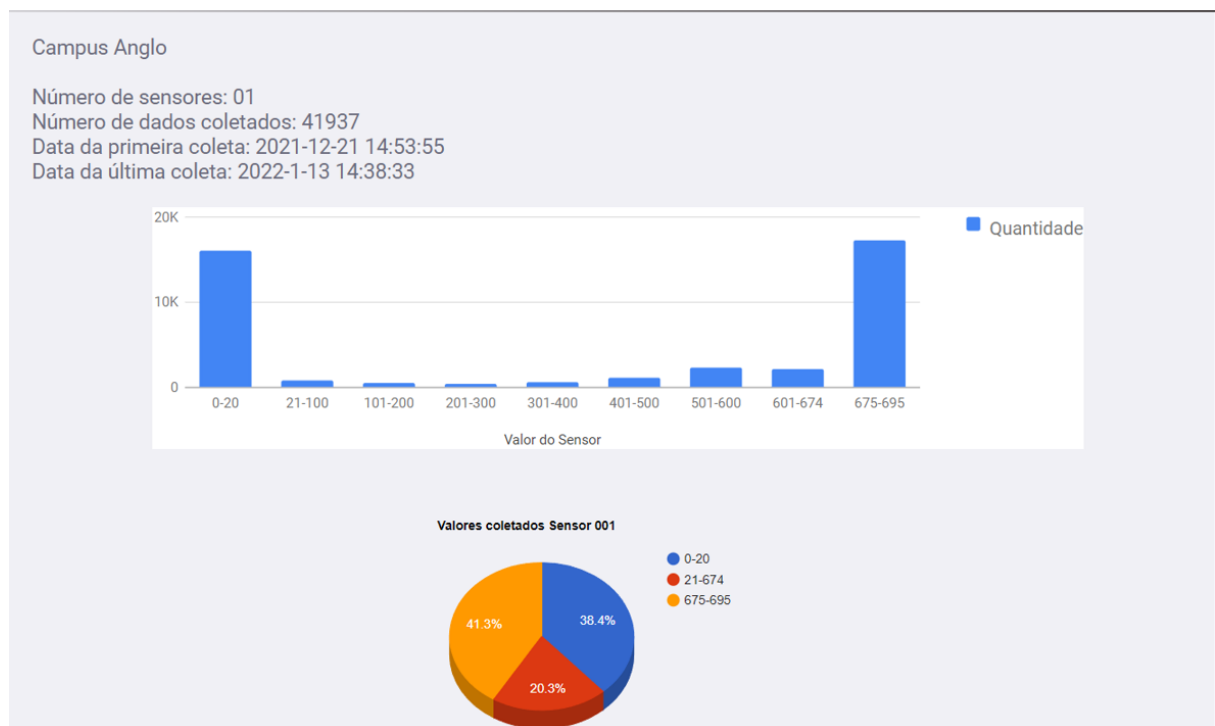


Figura 31 – Informações gerais dos dados

Framework, da forma como este foi implementado no campus Anglo é possível que sejam adicionados mais sensores a rede, coletando os mais variados tipos de dados e armazenando-os, assim como seria possível disponibilizá-los depois para o Weka e o Dashboard. Naturalmente este estudo de caso por não utilizar softwares capazes de tolerarem grandes quantidades de dados, tem sua escalabilidade limitada pelas tecnologias utilizadas. Porém, é válido salientar que com as tecnologias propostas no desenvolvimento é possível obter todas as características de escalabilidade desejadas.



Figura 32 – Lista de sensores Frontend

Por fim, é possível observar na Figura 33 como foi montando o circuito contendo o sensor de luz, acima na foto, e o microcontrolador NodeMCU, à direita, assim como haviam sido descritas previamente as conexões. Também foi feito o registro (Figura 34) do local onde a placa estava posicionada para que fosse feita a coleta dos valores, esta havia sido conectada a um computador e estava logo abaixo de uma janela sem cortinas, onde o sensor tinha como capturar toda a variação de luz ao longo dos dias.

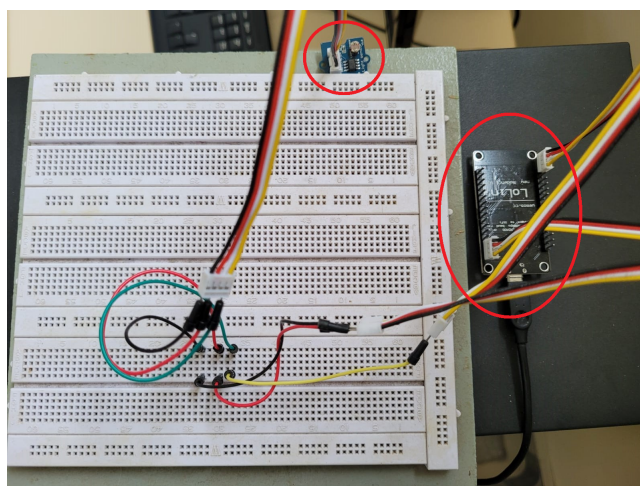


Figura 33 – Placa do estudo de caso

De modo a expandir o estudo de caso e buscar avaliar as características de escalabilidade do Framework implementado, foram feitas duas novas implementações de sensores no Campus UFPel. Os responsáveis por realizar a montagem e configuração do equipamento foram membros do grupo de pesquisa do próprio PPGC/UFPel.

Esta nova implementação foi feita da seguinte forma, cada nó sensor era composto por uma ESP32S Node MCU (Figura 35), uma placa de desenvolvimento baseada



Figura 34 – Local da coleta de dados

no módulo ESP32S WiFi, uma destas possuía dois sensores conectados a placa de desenvolvimento, a outra contava com somente um sensor.

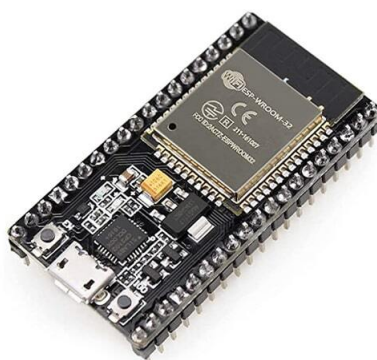


Figura 35 – ESP32S Node MCU

Em uma das ESP32 estavam acoplados um sensor de movimento Grove Mini PIR Motion Sensor (Figura 36), que se trata de um sensor infravermelho passivo, e um sensor de luminosidade, o mesmo sensor que já estava no estudo de caso descrito

anteriormente, na outra foi conectado um sensor sonoro Grove Loudness Sensor (Figura 37), que se trata de uma placa composta por um microfone de eletreto e um circuito integrado amplificador operacional duplo LM358.



Figura 36 – Grove Mini PIR Motion Sensor

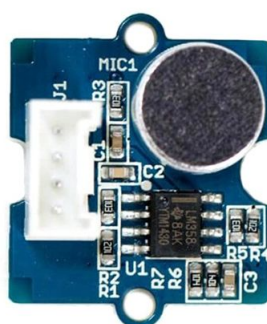


Figura 37 – Grove Loudness Sensor

Para realizar a configuração de modo a integrar no sistema foi necessário atualizar no código de configuração dos sensores a rede a qual iriam estar conectados e qual seria sua identificação no banco de dados. Com isto configurado e os sensores conectados a suas respectivas ESP32 era possível iniciar a coleta de dados. Com relação à identificação dos sensores no banco de dados, o sensor de luminosidade que já estava no estudo de caso se manteve com a identificação 001 no banco de dados, o sensor de presença foi identificado como 002 e por fim o sensor sonoro como 003.

Assim como no estudo de caso exposto anteriormente é preciso que todos os elementos do Framework estejam conectados à mesma rede de internet, além disto um outro ponto importante com relação à coleta de dados é a frequência com que estes são coletados, nesta expansão do estudo de caso os dados são coletados a

cada 5 segundos por cada sensor, gerando uma taxa de amostragem muito maior que a utilizada anteriormente e o sistema operou sem que ocorresse falhas, mesmo com esta maior frequência de entrada de dados.

Não houve necessidade de alterações na estrutura do banco de dados para que os novos sensores fossem adicionados, assim sendo as colunas do banco de dados seguem da mesma forma que na versão anterior do estudo de caso (Figura 24). Assim como também não houve alteração no formato do arquivo JSON enviado ao banco de dados, os campos se mantiveram os mesmos, sendo estes, o `sensorId`, responsável por identificar qual sensor fez a leitura, o `content`, que contém o dado que foi coletado, e por fim o campo `createdAt`, que identifica em qual data e horário foi feita a coleta.

Com relação aos dados obtidos, é possível observar na Tabela 7 algumas informações sobre o que foi coletado pelos sensores, tais como: Número de amostras coletadas por cada sensor, data da primeira e última coleta dos dados.

Tabela 7 – Informações sobre os dados coletados

Sensor	Nº de coletas	Primeira coleta	Última coleta
001	387.616	21/12/2021 às 14:53:55	09/06/2022 às 09:09:53
002	276	20/12/2022 às 15:33:10	20/12/2022 às 16:08:50
003	276	20/12/2022 às 15:33:10	20/12/2022 às 16:08:51
Total	388.168	-	-

É possível observar que o sensor 001 possui muito mais amostras coletadas que os demais, isso se deve ao fato deste ter sido coletado muito tempo antes para coletar dados e também ter permanecido por muito mais tempo em atividade. Por fim na Figura 38 é possível observar o nó sensor que possuía o sensor de movimento, este foi posicionado em um dos corredores do campus Anglo da Universidade Federal de Pelotas. O outro nó sensor que era composto por dois sensores pode ser observado na Figura 39, este manteve-se posicionado no mesmo laboratório onde foi realizado o estudo de caso citado anteriormente.



Figura 38 – Instalação do nó sensor - Sensor de Movimento

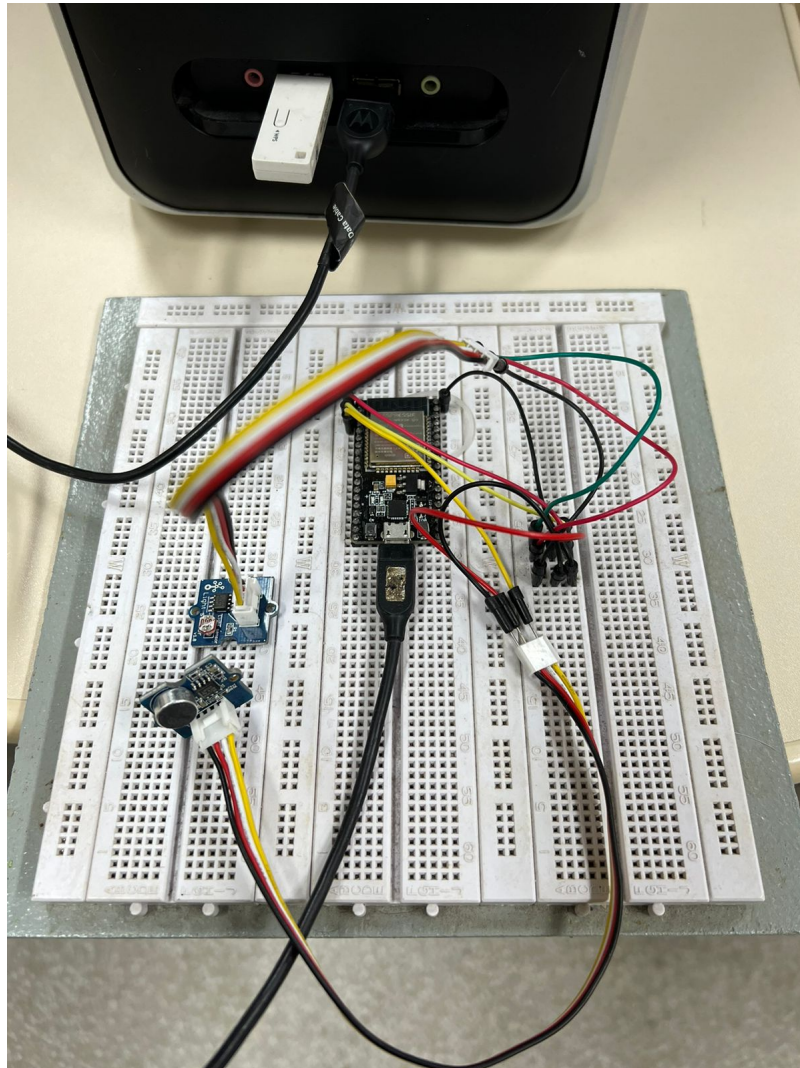


Figura 39 – Instalação do nó sensor - Sensor de Luminosidade e Sonoro

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou a proposta de um Framework que ofereça suporte a implementação de Hardware/Software para aplicações IoT no contexto de Campus Inteligente. Como características desejadas, foi concebido que seja escalável, independência de tecnologia e que ofereça suporte a diversas aplicações no contexto de um campus inteligente e os demais serviços que sejam viáveis e pertinentes a um Campus Inteligente.

A arquitetura do Framework proposto foi composta de 5 camadas: Infraestrutura, Comunicação, Armazenamento, Processamento e Serviços, para as quais foram definidas tecnologias correspondentes, além de quais protocolos de comunicação estas deveriam seguir de modo a enviar/receber dados de outras camadas. Com relação a trabalhos semelhantes encontrados na literatura, o trabalho proposto se destacou em alguns aspectos: o detalhamento das tecnologias utilizadas e como estas deveriam se comunicar no Framework, a possibilidade de utilizar tecnologias menos complexas, que mesmo tendo certas limitações de escalabilidade podem ser utilizadas dependendo do contexto, assim como foi o estudo de caso e por fim a característica já desejada para o Framework de escalabilidade.

Os resultados alcançados ao ser implementado o estudo de caso foram satisfatórios, pois este seguiu os protocolos de funcionamento do Framework e o fluxo projetado ocorreu dentro do esperado, os dados foram coletados, armazenados e disponibilizados para as duas plataformas desejadas, Weka e o Dashboard, de forma simples e ainda haviam opções de escalabilidade. Apesar deste ter sido construído para somente ilustrar de forma reduzida o funcionamento do Framework, é possível notar que todo comportamento esperado foi obtido e as tecnologias utilizadas se assemelham às que foram propostas para o desenvolvimento completo do projeto.

É importante salientar que algumas mudanças podem ser feitas com relação à escolha de tecnologias, pois esta proposta apresenta certa complexidade de implementação, como, por exemplo, o uso do Apache Nifi e do Apache Hive, para realizar a configuração de ambos e ter disponível um ambiente com capacidade computacional suficiente para executá-los são necessários mais recursos, tanto de pessoas capacita-

das quanto de computadores, e caso o contexto de implementação do Framework seja com uma quantidade relativamente pequena de dados é possível optar por soluções menos custosas. Estas podem ser integradas ao sistema substituindo as tecnologias atuais desde que consigam se comunicar com o resto do sistema, independente da camada que se deseje alterar algo.

Em relação aos trabalhos futuros, é possível observar que existem inúmeros estudos de caso que possam ser realizados utilizando este Framework como base, ao implementar o que foi proposto e variar de diversas maneiras protocolos de comunicação e tecnologias, de modo a averiguar qual possui melhor comportamento e custo/benefício dado certo contexto. Além disto, também é possível buscar uma forma de aumentar a quantidade de formas de obtenção de dados, como, por exemplo, capturar dados de frequentadores do campus, de modo que aqueles que permitirem tenham alguns dados coletados, como, por exemplo, sua localização no campus, para que com isso cada vez mais serviços possa ser oferecidos.

REFERÊNCIAS

ABDULLAH, A.; THANOON, M.; ALSULAMI, A. Toward a Smart Campus Using IoT: Framework for Safety and Security System on a University Campus. , [S.l.], 2019.

ABUALNAAJ, K.; AHMED, V.; SABOOR, S. A Strategic Framework for Smart Campus. In: ANNUAL INTERNATIONAL CONFERENCE ON INDUSTRIAL ENGINEERING AND OPERATIONS MANAGEMENT, DUBAI, UAE, 10., 2020. **Proceedings...** [S.l.: s.n.], 2020. p.10–12.

AGARWAL, P.; GVV, R. K.; AGARWAL, P. IoT based Framework for Smart Campus: COVID-19 Readiness. In: FOURTH WORLD CONFERENCE ON SMART TRENDS IN SYSTEMS, SECURITY AND SUSTAINABILITY (WORLDS4), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.539–542.

AION, N.; HELMANDOLLAR, L.; WANG, M.; NG, J. W. Intelligent campus (iCampus) impact study. In: IEEE/WIC/ACM INTERNATIONAL CONFERENCES ON WEB INTELLIGENCE AND INTELLIGENT AGENT TECHNOLOGY, 2012., 2012. **Anais...** [S.l.: s.n.], 2012. v.3, p.291–295.

AL DAKHEEL, J.; DEL PERO, C.; ASTE, N.; LEONFORTE, F. Smart buildings features and key performance indicators: A review. **Sustainable Cities and Society**, [S.l.], v.61, p.102328, 2020.

ALHAMAD, M.; DILLON, T.; CHANG, E. Conceptual SLA framework for cloud computing. In: IEEE INTERNATIONAL CONFERENCE ON DIGITAL ECOSYSTEMS AND TECHNOLOGIES, 4., 2010. **Anais...** [S.l.: s.n.], 2010. p.606–610.

ALI, Z. H.; ALI, H. A.; BADAWY, M. M. Internet of Things (IoT): definitions, challenges and recent research directions. **International Journal of Computer Applications**, [S.l.], v.128, n.1, p.37–47, 2015.

ARUMUGAM, R. et al. DAVinCi: A cloud computing framework for service robots. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2010., 2010. **Anais...** [S.l.: s.n.], 2010. p.3084–3089.

BARBATO, A. et al. An energy management framework for optimal demand response in a smart campus. **ICGREEN (2015)**, [S.l.], v.11, 2015.

BERROUYNE, I. et al. CyprIoT: framework for modelling and controlling network-based IoT applications. In: ACM/SIGAPP SYMPOSIUM ON APPLIED COMPUTING, 34., 2019. **Proceedings...** [S.l.: s.n.], 2019. p.832–841.

BORAN, A. et al. A smart campus prototype for demonstrating the semantic integration of heterogeneous data. In: INTERNATIONAL CONFERENCE ON WEB REASONING AND RULE SYSTEMS, 2011. **Anais...** [S.l.: s.n.], 2011. p.238–243.

BOUGUERA, T. et al. Energy consumption model for sensor nodes based on LoRa and LoRaWAN. **Sensors**, [S.l.], v.18, n.7, p.2104, 2018.

CAMERO, A.; ALBA, E. Smart City and information technology: A review. **cities**, [S.l.], v.93, p.84–94, 2019.

CHEN, X. Exploration and Practice of the Construction of Smart Campus——Taking Sichuan University as an Example. In: THE 4TH INTERNATIONAL CONFERENCE ON DIGITAL TECHNOLOGY IN EDUCATION, 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.36–41.

CHERUVU, S.; KUMAR, A.; SMITH, N.; WHEELER, D. M. **Demystifying internet of things security**: successful iot device/edge and platform security deployment. [S.l.]: Springer Nature, 2020.

COTRIM, J. R.; KLEINSCHMIDT, J. H. LoRaWAN mesh networks: A review and classification of multihop communication. **Sensors**, [S.l.], v.20, n.15, p.4273, 2020.

DE MAURO, A.; GRECO, M.; GRIMALDI, M. What is big data? A consensual definition and a review of key research topics. In: AIP CONFERENCE PROCEEDINGS, 2015. **Anais...** [S.l.: s.n.], 2015. v.1644, n.1, p.97–104.

DEBAUCHE, O. et al. RevoCampus: a Distributed Open Source and Low-cost Smart Campus. In: INTERNATIONAL CONFERENCE ON ADVANCED COMMUNICATION TECHNOLOGIES AND NETWORKING (COMMNET), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.1–10.

DEY, A. et al. Namatad: Inferring occupancy from building sensors using machine learning. In: IEEE 3RD WORLD FORUM ON INTERNET OF THINGS (WF-IOT), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.478–483.

DING, D.; COOPER, R. A.; PASQUINA, P. F.; FICI-PASQUINA, L. Sensor technology for smart homes. **Maturitas**, [S.l.], v.69, n.2, p.131–136, 2011.

GUO, M.; GUO, J. Constructing smart campus network based on ubiquitous sensor technology. In: INTERNATIONAL CONFERENCE ON INFORMATION SCIENCE AND TECHNOLOGY (ICIST), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p.265–268.

GUPTA, M. Google Dataflow orchestration using TOSCA in the hybrid cloud. , [S.l.], 2022.

HUAI, Y. et al. Major technical advancements in apache hive. In: ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 2014., 2014. **Proceedings...** [S.l.: s.n.], 2014. p.1235–1246.

JOHN, T. M.; UCHEAGA, E. G.; BADEJO, J. A.; ATAYERO, A. A. A framework for a smart campus: A case of covenant university. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE AND COMPUTATIONAL INTELLIGENCE (CSCI), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.1371–1376.

JUTADHAMAKORN, P. et al. A scalable and low-cost MQTT broker clustering system. In: INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY (IN-CIT), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.1–5.

KIM, S.-S.; LEE, W.-R.; GO, J.-H. A study on utilization of spatial information in heterogeneous system based on apache NiFi. In: INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGY CONVERGENCE (ICTC), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p.1117–1119.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering—a tertiary study. **Information and software technology**, [S.l.], v.52, n.8, p.792–805, 2010.

KOSTEPEN, Z. N. et al. A Framework for Sustainable and Data-driven Smart Campus. In: ICEIS (2), 2020. **Anais...** [S.l.: s.n.], 2020. p.746–753.

LIU, M.; MA, J.; JIN, L. Analysis of Military Academy Smart Campus Based on Big Data. In: INTERNATIONAL CONFERENCE ON INTELLIGENT HUMAN-MACHINE SYSTEMS AND CYBERNETICS (IHMSC), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. v.1, p.105–108.

LIU, X.; BAI OCCHI, O. A comparison of the definitions for smart sensors, smart objects and Things in IoT. In: IEEE 7TH ANNUAL INFORMATION TECHNOLOGY, ELECTRONICS AND MOBILE COMMUNICATION CONFERENCE (IEMCON), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.1–4.

LUO, L. Data acquisition and analysis of smart campus based on wireless sensor. **Wireless Personal Communications**, [S.l.], v.102, n.4, p.2897–2911, 2018.

MACIÁ PÉREZ, F.; BERNA MARTÍNEZ, J. V.; LORENZO FONSECA, I. Modelling and Implementing Smart Universities: An IT Conceptual Framework. **Sustainability**, [S.l.], v.13, n.6, p.3397, 2021.

MARTINS, P.; LOPES, S. I.; CRUZ, A. M. Rosado da; CURADO, A. Towards a Smart & Sustainable Campus: An Application-Oriented Architecture to Streamline Digitization and Strengthen Sustainability in Academia. **Sustainability**, [S.l.], v.13, n.6, p.3189, 2021.

MIN-ALLAH, N.; ALRASHED, S. Smart campus—A sketch. **Sustainable Cities and Society**, [S.l.], v.59, p.102231, 2020.

MOHAMMADI, M.; AL-FUQAHA, A. Enabling cognitive smart cities using big data and machine learning: Approaches and challenges. **IEEE Communications Magazine**, [S.l.], v.56, n.2, p.94–101, 2018.

NEWAZ, S. S. et al. A web based energy cloud platform for campus smart grid for understanding energy consumption profile and predicting future energy demand. In: INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGY CONVERGENCE (ICTC), 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p.173–178.

NG, J. W. et al. The intelligent campus (iCampus): end-to-end learning lifecycle of a knowledge ecosystem. In: SIXTH INTERNATIONAL CONFERENCE ON INTELLIGENT ENVIRONMENTS, 2010., 2010. **Anais...** [S.l.: s.n.], 2010. p.332–337.

NIE, X. Constructing smart campus based on the cloud computing platform and the internet of things. In: INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND ELECTRONICS ENGINEERING, 2., 2013. **Anais...** [S.l.: s.n.], 2013. p.1576–1578.

PAGLIARO, F. et al. A roadmap toward the development of Sapienza Smart Campus. In: IEEE 16TH INTERNATIONAL CONFERENCE ON ENVIRONMENT AND ELECTRICAL ENGINEERING (EEEIC), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.1–6.

PANDEY, J.; SINGH, A. V.; RANA, A. Roadmap to Smart Campus based on IoT. In: INTERNATIONAL CONFERENCE ON RELIABILITY, INFOCOM TECHNOLOGIES AND OPTIMIZATION (TRENDS AND FUTURE DIRECTIONS)(ICRITO), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.909–913.

PETCOVICI, A.; STROULIA, E. Location-based services on a smart campus: A system and a study. In: IEEE 3RD WORLD FORUM ON INTERNET OF THINGS (WF-IOT), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.94–99.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. In: INTERNATIONAL CONFERENCE ON EVALUATION AND ASSESSMENT IN SOFTWARE ENGINEERING (EASE) 12, 12., 2008. **Anais...** [S.l.: s.n.], 2008. p.1–10.

PHAM, T. V. et al. Proposed Smart University Model as a Sustainable Living Lab for University Digital Transformation. In: INTERNATIONAL CONFERENCE ON GREEN TECHNOLOGY AND SUSTAINABLE DEVELOPMENT (GTSD), 2020., 2020. **Anais...** [S.l.: s.n.], 2020. p.472–479.

SAMAL, B.; PANDA, M. Real time product feedback review and analysis using apache technologies and nosql database. **International Journal of Engineering and Computer Science**, [S.l.], v.6, n.10, p.22551–22558, 2017.

SARI, M. W.; CIPTADI, P. W.; HARDYANTO, R. H. Study of smart campus development using internet of things technology. In: IOP CONFERENCE SERIES: MATERIALS SCIENCE AND ENGINEERING, 2017. **Anais...** [S.l.: s.n.], 2017. v.190, n.1, p.012032.

STATE of IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally. Accessed: 2022-06-26, [https://iot-analytics.com/number-connected-iot-devices/#:~:text=The%20number%20of%20connected%20IoT%20devices%20using%20cellular%20technology%20grew,half%20\(1H\)%20of%202021](https://iot-analytics.com/number-connected-iot-devices/#:~:text=The%20number%20of%20connected%20IoT%20devices%20using%20cellular%20technology%20grew,half%20(1H)%20of%202021).

TAURION, C. **Cloud computing-computação em nuvem**. [S.l.]: Brasport, 2009.

VAKALLOUDIS, A.; O'LEARY, C. A framework for rapid integration of IoT Systems with industrial environments. In: IEEE 5TH WORLD FORUM ON INTERNET OF THINGS (WF-IOT), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p.601–605.

VALKS, B.; ARKESTEIJN, M. H.; KOUTAMANIS, A.; HEIJER, A. C. den. Towards a smart campus: supporting campus decisions with Internet of Things applications. **Building Research & Information**, [S.l.], v.49, n.1, p.1–20, 2021.

VILLEGAS-CH, W. et al. Application of a big data framework for data monitoring on a smart campus. **Sustainability**, [S.l.], v.11, n.20, p.5552, 2019.

VILLEGAS-CH, W.; PALACIOS-PACHECO, X.; ROMÁN-CAÑIZARES, M. An Internet of Things Model for Improving Process Management on University Campus. **Future Internet**, [S.l.], v.12, n.10, p.162, 2020.

WANG, Y. et al. A smart campus internet of things framework. In: IEEE 8TH ANNUAL UBIQUITOUS COMPUTING, ELECTRONICS AND MOBILE COMMUNICATION CONFERENCE (UEMCON), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.498–503.

WARD, J. S.; BARKER, A. Undefined by data: a survey of big data definitions. **arXiv preprint arXiv:1309.5821**, [S.l.], 2013.

WEBB, J.; HUME, D. Campus IoT collaboration and governance using the NIST cybersecurity framework. , [S.l.], 2018.

WOHLIN, C. et al. On the reliability of mapping studies in software engineering. **Journal of Systems and Software**, [S.l.], v.86, n.10, p.2594–2610, 2013.

YANG, C.-T. et al. On construction of an energy monitoring service using big data technology for the smart campus. **Cluster Computing**, [S.l.], v.23, n.1, p.265–288, 2020.

YELMARTHI, K.; ABDELGAWAD, A.; KHATTAB, A. An architectural framework for low-power IoT applications. In: INTERNATIONAL CONFERENCE ON MICROELECTRONICS (ICM), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p.373–376.

YU, W. et al. A survey on the edge computing for the Internet of Things. **IEEE access**, [S.l.], v.6, p.6900–6919, 2017.