



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS QUIXADÁ**  
**CURSO DE GRADUAÇÃO EM SISTEMAS DA INFORMAÇÃO**

**IARLEN DA SILVA AQUILES**

**UTILIZANDO DADOS HISTÓRICOS PARA REALIZAR PREVISÕES EM SMART  
CAMPUS**

**QUIXADÁ**  
**2022**

IARLEN DA SILVA AQUILES

UTILIZANDO DADOS HISTÓRICOS PARA REALIZAR PREVISÕES EM SMART  
CAMPUS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas da informação do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas da informação.

Orientador: Prof. Me. Marcos Dantas Ortiz

QUIXADÁ

2022

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

A667u Aquiles, Iarlen da Silva.  
Utilizando dados históricos para realizar previsões em smart campus / Iarlen da Silva Aquiles. – 2022.  
69 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá,  
Curso de Sistemas de Informação, Quixadá, 2022.  
Orientação: Prof. Me. Marcos Dantas Ortiz.

1. Fiware. 2. Smart Campus. 3. Internet das Coisas. I. Título.

CDD 005

---

IARLEN DA SILVA AQUILES

UTILIZANDO DADOS HISTÓRICOS PARA REALIZAR PREVISÕES EM SMART  
CAMPUS

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Sistemas da informação  
do Campus Quixadá da Universidade Federal  
do Ceará, como requisito parcial à obtenção do  
grau de bacharel em Sistemas da informação.

Aprovada em: \_\_\_\_ / \_\_\_\_ / \_\_\_\_.

BANCA EXAMINADORA

---

Prof. Me. Marcos Dantas Ortiz (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Paulo Antonio Leal Rego  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Jefferson de Carvalho Silva  
Universidade Federal do Ceará (UFC)

À minha mãe, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

## **AGRADECIMENTOS**

Gostaria de agradecer a Deus por ter me dado coragem e paciência para lutar ao longo desta caminhada. Pois, até aqui o Senhor me ajudou.

Ao Prof. MSc. Marcos Dantas Ortiz por me orientar em meu trabalho, onde ele acreditou mais em mim do que eu mesmo, e enquanto eu fraquejava, ele estava lá para incentivar no término do trabalho.

À minha avó Liduina Aquiles, que é mais minha mãe do que a minha própria mãe, que me criou e sempre me incentivou ao estudo e devo esse trabalho à ela e como forma de reconhecimento do investimento que fez em mim.

À Marilene Alves, que sempre esteve comigo e acreditou em mim.

Aos colegas, Jorge e Rafael, que na correria estavam lá para darem uma força quando eu precisava.

Ademais, gostaria de agradecer a todos os professores que contribuíram profissionalmente para a minha carreira e me repassaram os conhecimentos necessários para o desenvolvimento deste trabalho.

Por fim, agradeço a todos que não foram citados, mas que contribuíram diretamente e indiretamente para a finalização deste ciclo.

“O sucesso é a soma de pequenos esforços repetidos dia após dia.”

(Robert Collier)

## RESUMO

O crescente processo de globalização está tornando o desenvolvimento sustentável das cidades um grande desafio e ter a consciência que devem ter cuidado para o consumo sustentável e deixar recursos para as próximas gerações. Só entre os Estados Unidos e a China são consumidas cerca de 60% da energia elétrica gerada no mundo e esta situação se agrava mais pelo motivo de que essas fontes não renováveis degradam o meio ambiente. O principal objetivo das *Smart Cities* é, de fato, tornar as cidades um lugar melhor para se viver agora e, a longo prazo, colocar os cidadãos no centro. Para atingir este objetivo, as *Smart Cities* contam com o paradigma da Internet das Coisas (IoT), pelo que a *Tecnologia da Informação e Comunicação* (TIC) é aplicada de forma intensiva em todas as áreas relacionadas com o bem-estar dos cidadãos, como transportes e mobilidade, saúde, energia ou o ambiente. Isso significa que a sociedade precisa gerenciar melhor os seus recursos para não gerar escassez. Todavia, a Internet das Coisas (IoT) é uma aliada à Inteligência Artificial que traz a ideia de ambientes inteligentes que inserem a computação no mundo físico com o intuito de monitorar e automatizar atividades corriqueiras, identificar padrões e realizar previsões. Neste trabalho é proposta uma solução IoT para o armazenamento de dados históricos dos sensores instalados no Campus de Quixadá para o monitoramento dos ambientes inteligentes e realização de previsões. Deste modo, são tratados os procedimentos de construção e configuração das diferentes partes do projeto, como a implantação do *QuantumLeap* na Nuvem que é responsável de receber a notificação e armazenar o dado no seu banco de dados, utilizando a plataforma *Fiware*. Também foi apresentado a biblioteca para o tratamento e realização das previsões das séries temporais e a integração com o *Dashboard* para a visualização dos dados. Ademais, foram realizados os testes e apresentados os resultados foram promissores, indicando que a solução possui uma grande contribuição ao Campus e aos gestores.

**Palavras-chave:** *Fiware*. *Smart Campus*. Internet das Coisas. Séries temporais.

## ABSTRACT

The growing process of globalization is making the sustainable development of cities a great challenge and to be aware that they must be careful for sustainable consumption and leave resources for the next generations. About 60% of the electricity generated in the world is consumed between the United States and China alone, and this situation is exacerbated by the fact that these non-renewable sources degrade the environment. The main objective of Smart Cities is, in fact, to make cities a better place to live in now and, in the long run, to put citizens at the center. To achieve this goal, Smart Cities rely on the Internet of Things (IoT) paradigm, so Information and Communication Technology (ICT) is intensively applied in all areas related to the well-being of citizens, such as transport and mobility, health, energy or the environment. This means that society needs to better manage its resources so as not to create scarcity. However, the Internet of Things (IoT) is an ally of Artificial Intelligence that brings the idea of intelligent environments that insert computing into the physical world in order to monitor and automate everyday activities, identify patterns and make predictions. In this work, an IoT solution is proposed for the storage of historical data from sensors installed on the Quixadá Campus for the monitoring of intelligent environments. In this way, the procedures for construction and configuration of the different parts of the project, such as the implementation of QuantumLeap in the Cloud, which is responsible for receiving the notification and storing the data in its database, using the Fiware platform. The library was also presented for the treatment and realization of time series forecasts and the integration with the Dashboard for data visualization. In addition, tests were carried out and the results were promising, indicating that the solution has a great contribution to the Campus and managers.

**Keywords:** Fiware. Smart Campus. Internet of Things. Time Series

## LISTA DE FIGURAS

Figura 1 – As novas tecnologias permitem a conectividade de todos os tipos de sensores à Internet. . . . .	20
Figura 2 – Escopo da IoT. . . . .	23
Figura 3 – <i>Visão geral da arquitetura do FIWARE</i> . . . . .	24
Figura 4 – Arquitetura da requisição NGSI. . . . .	25
Figura 5 – Arquitetura do <i>QuantumLeap</i> . . . . .	30
Figura 6 – Dataframe aceito pelo Prophet. . . . .	33
Figura 7 – Gráfico gerado pelo Prophet. . . . .	33
Figura 8 – Arquitetura Smart CEI Moncloa . . . . .	36
Figura 9 – Dashboard Smart CEI Moncloa. . . . .	36
Figura 10 – Arquitetura Smart Place . . . . .	39
Figura 11 – Dashboard Smart Place . . . . .	41
Figura 12 – Smart Meter . . . . .	42
Figura 13 – App Smart Meter . . . . .	43
Figura 14 – Arquitetura simplificada de comunicação . . . . .	45
Figura 15 – <i>Visão dos dados.</i> . . . .	47
Figura 16 – <i>Visão geral da arquitetura da solução</i> . . . . .	49
Figura 17 – Exemplo da tela da aplicação . . . . .	51
Figura 18 – <i>Visão de uma requisição da notificação entre Orion-QuantumLeap.</i> . . . . .	52
Figura 19 – Exemplo da criação da notificação . . . . .	52
Figura 20 – Exemplo utilizando um período de 30 dias . . . . .	54
Figura 21 – Exemplo utilizando um período de 30 dias . . . . .	55
Figura 22 – Gráfico dos dados capturados com o eixo y mostrando os valores e o eixo x mostrando as datas separado por mês. . . . .	56
Figura 23 – Previsão de 45 dias. . . . .	57
Figura 24 – Tendência de 45 dias. . . . .	58

## LISTA DE QUADROS

Quadro 1 – Comparação entre os trabalhos . . . . .	43
Quadro 2 – Dados carregados . . . . .	47

## LISTA DE ABREVIATURAS E SIGLAS

CSV	<i>Comma-separated values</i>
GE	<i>General Enable</i>
IA	<i>Inteligência Artificial</i>
IES	<i>Instituição de Ensino Superior</i>
IoT	<i>Internet das Coisas</i>
KPIs	<i>Indicador-Chave de Desempenho</i>
OMA	<i>Open Mobile Alliance</i>
OSM	<i>OpenStreetMap</i>
RA	<i>Realidade Aumentada</i>
SOA	<i>Service-Oriented Architecture</i>
TIC	<i>Tecnologia da Informação e Comunicação</i>
WS	<i>Web Socket</i>

## SUMÁRIO

1	<b>INTRODUÇÃO</b>	15
2	<b>OBJETIVOS</b>	18
2.1	<b>Objetivo Geral</b>	18
2.2	<b>Objetivos específicos</b>	18
3	<b>FUNDAMENTAÇÃO TEÓRICA</b>	19
3.1	<b>Smart Campus</b>	19
3.1.1	<i>Computação em Nuvem</i>	19
3.1.2	<i>Internet das Coisas</i>	21
3.2	<b>FIWARE</b>	23
3.2.1	<i>NGSI</i>	24
3.2.2	<i>NGSI-LD</i>	26
3.3	<b>Orion Context Broker</b>	26
3.4	<b>QuantumLeap</b>	27
3.4.1	<i>Arquitetura</i>	29
3.4.2	<i>QuantumLeap versus STH-comet</i>	29
3.5	<b>Séries Temporais</b>	30
3.5.1	<i>Diferentes tipos de séries temporais</i>	31
3.5.2	<i>Objetivos da análise de séries temporais</i>	31
3.5.3	<i>Previsão de séries temporais</i>	32
3.6	<b>Prophet</b>	32
3.6.1	<i>Como funciona?</i>	33
4	<b>TRABALHOS RELACIONADOS</b>	35
4.1	<b>Smart CEI Moncloa: An IoT-based Platform for People Flow and Environmental Monitoring on a Smart University Campus (ALVAREZ-CAMPANA <i>et al.</i>, 2017)</b>	35
4.1.1	<i>Arquitetura</i>	35
4.1.2	<i>Sensores e Atuadores</i>	37
4.1.3	<i>Aplicação</i>	37
4.2	<b>Energy Efficiency in Smart Buildings: An IoT-Based Air Conditioning Control System (ROCHA <i>et al.</i>, 2019)</b>	38

4.2.1	<i>Arquitetura</i> . . . . .	38
4.2.2	<i>Sensores e Atuadores</i> . . . . .	39
4.2.3	<i>Aplicação</i> . . . . .	40
4.3	<b>Desenvolvimento de uma nova Metodologia para Previsão do Consumo de Energia Elétrica de Curto Prazo Utilizando Redes Neurais Artificiais e Decomposição de Séries Temporais (AMARAL, 2019)</b> . . . . .	40
4.3.1	<i>Arquitetura</i> . . . . .	41
4.3.2	<i>Sensores e Atuadores</i> . . . . .	42
4.3.3	<i>Aplicação</i> . . . . .	42
4.4	<b>Comparação entre os trabalhos</b> . . . . .	43
5	<b>SOLUÇÃO PROPOSTA</b> . . . . .	45
5.1	<b>Avaliação das tecnologias</b> . . . . .	45
5.2	<b>Treinamento e avaliação da previsão com o Prophet</b> . . . . .	46
5.3	<b>Implantação da solução</b> . . . . .	47
5.3.1	<i>Configurando uma notificação</i> . . . . .	48
5.4	<b>Aplicação de Monitoramento</b> . . . . .	49
5.4.1	<i>Front-end</i> . . . . .	49
5.4.2	<i>Back-end</i> . . . . .	50
5.4.3	<i>Armazenamento Histórico</i> . . . . .	50
5.4.3.1	<i>Assinatura Orion</i> . . . . .	51
5.4.3.2	<i>Inserção de dados</i> . . . . .	51
5.4.3.3	<i>Transmissão de dados</i> . . . . .	52
5.4.3.4	<i>Índice de tempo</i> . . . . .	53
5.4.3.5	<i>Restrições e limitações</i> . . . . .	53
5.4.3.6	<i>Recuperação de dados</i> . . . . .	53
6	<b>AVALIAÇÃO E RESULTADOS</b> . . . . .	54
6.1	<b>Resultados preliminares</b> . . . . .	54
7	<b>RESULTADOS</b> . . . . .	56
7.1	<b>Armazenamento dos dados</b> . . . . .	56
7.2	<b>Conjunto de dados</b> . . . . .	57
7.3	<b>Resultados do Experimento com período de 45 dias</b> . . . . .	57
7.4	<b>Eficácia da solução</b> . . . . .	58

7.4.1	<i>Avaliação da Solução Final</i> . . . . .	58
8	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> . . . . .	59
	<b>REFERÊNCIAS</b> . . . . .	61
	<b>APÊNDICES</b> . . . . .	63
	<b>APÊNDICE A – Docker Compose</b> . . . . .	63
	<b>ANEXOS</b> . . . . .	69

## 1 INTRODUÇÃO

Segundo (ALVAREZ-CAMPANA *et al.*, 2017) com o crescente processo de globalização, o desenvolvimento sustentável das cidades tornou-se um grande desafio para a sociedade atual. Com esse desenvolvimento, muitas iniciativas e projetos foram lançadas sob um novo paradigma das chamadas *Smart Cities*. O principal objetivo das *Smart Cities* é, de fato, tornar as cidades um lugar melhor para se viver agora e, a longo prazo, colocar os cidadãos no centro. Para atingir este objetivo, as *Smart Cities* contam com o paradigma da Internet das Coisas (IoT), pelo que a TIC é aplicada de forma intensiva em todas as áreas relacionadas com o bem-estar dos cidadãos, como transportes e mobilidade, saúde, energia ou o ambiente. Isso significa que a sociedade precisa gerenciar melhor os seus recursos para não gerar escassez.

Segundo Rocha *et al.* (2019) o consumo de energia elétrica por prédios residenciais e comerciais são cerca de 60% da energia elétrica gerada no mundo e esse consumo está dividido entre 40% dos Estados Unidos e 20% da China. Esta é uma situação alarmante, pois a maior parte da matriz energética mundial é composta por fontes não renováveis, como carvão, petróleo e gás natural, que impactam negativamente a Terra.

De acordo com Martins *et al.* (2021) a implementação da sustentabilidade nas *Instituição de Ensino Superior (IES)* não passa apenas por questões como a criação de espaços verdes e a requalificação de áreas de lazer, ou o desenvolvimento de estratégias para reduzir o consumo de papel, plástico, energia e água, ou ainda a criação de programas de incentivo à reciclagem mas também temas de extrema importância como a implementação de um conjunto de ações adicionais visando a criação de um Campus Inteligente, utilizando o conceito de Internet das Coisas (IoT) como ferramenta prática.

Segundo Salhofer *et al.* (2019) mostram uma ideia de *Smart City* onde variados ambientes urbanos sendo monitorados de forma a capturar dados de umidade, temperatura e concentração de partículas finas para a análise de qualidade do ar. De acordo com esse pensamento, tem crescido o interesse pelas chamadas construções inteligentes que monitoram e regulam dinamicamente o uso de energia elétrica, como, por exemplo, com controle automático de dispositivos, alerta de consumo, horário de funcionamento e análise dos dados de um equipamento específico, buscando identificar anomalias a fim de reduzir o consumo e otimizar o uso de dispositivos elétricos.

No trabalho de Martins *et al.* (2021) usando essas tecnologias de sensores para medir o uso do espaço em tempo real, aplicativos operacionais podem ser desenvolvidos para melhorar

a qualidade do espaço, mas também coletar informações sobre utilização de espaços, satisfação do usuário e consumo de energia, apoiando várias decisões de gerenciamento do campus, como reforma de edifícios, otimização dos horários de limpeza, horário de funcionamento e compra de quantidades nas instalações alimentares. Com esta informação, definem-se quatro perspectivas para os *stakeholders*, através das quais se pode medir o valor estratégico para potencializar a qualidade da área ao mesmo tempo que permite colaboração e inovação e apoia a sua imagem e cultura; Valor funcional para aumentar a flexibilidade e satisfação do usuário ao mesmo tempo em que possibilita suporte acomodativo para suas atividades; Valor financeiro para a redução de custos e crescimento de receita; e valor físico para reduzir e controlar o fluxo de pessoas.

A previsão de séries temporais pode ser um desafio, pois há muitos métodos diferentes que você pode usar e muitos hiperparâmetros diferentes para cada método. O *Prophet* é uma biblioteca de código aberto projetada para fazer previsões para conjuntos de dados de séries temporais univariadas. É fácil de usar e projetado para encontrar automaticamente um bom conjunto de hiperparâmetros para o modelo em um esforço para fazer previsões habilidosas para dados com tendências e estrutura sazonal por padrão.

Dito isso, este trabalho busca a identificação de previsões através de padrões existentes na coleta de dados de consumo de energia elétrica, uso da lampada, ar condicionado e rede *Wi-fi*. Para isso, amostras são capturadas dos sensores e será salvas em duas ferramentas de armazenamento, consulta e recuperação de dados espaço-temporais. Além disso, o trabalho recorre ao *middleware FIWARE* e ao *QuantumLeap* para o gerenciamento e armazenamento dos dados históricos.

Foi empregada a Ferramenta *QuantumLeap* disponibilizada pelo *FIWARE* para implementação e configuração nas máquinas virtuais. Assim, este trabalho é destinado aos gestores de recursos que visam reduzir e monitorar o consumo de energia, iluminação adequada e temperatura e umidade aceitáveis. E ainda, este trabalho possui relevância para pesquisas que usam e aplicam o conceito de *middlewares* e computação ubíqua em um contexto de *Smart Building*.

Este trabalho está organizado da seguinte forma: no Capítulo 2, são apresentados os objetivos; no Capítulo 3 é apresentada a fundamentação teórica; no Capítulo 4 são listados os trabalhos relacionados e um comparativo entre eles, no Capítulo 5 é abordada a metodologia realizada. No Capítulo 6, é apresentado a avaliação e os resultados do projeto. No Capítulo 7 são apresentados os resultados. E por último, no Capítulo 8 é apresentada a conclusão e trabalhos

futuros.

## 2 OBJETIVOS

### 2.1 Objetivo Geral

O objetivo deste trabalho é a evolução de uma plataforma de monitoramento<sup>1</sup> do Campus de Quixadá que vai prover o armazenamento dos dados históricos de consumo de energia dos blocos e temperatura/umidade das salas inteligentes e utilizar a previsibilidade para tomada de decisão.

### 2.2 Objetivos específicos

São objetivos específicos deste trabalho:

- a) Estudar e projetar o armazenamento histórico dos dados.
- b) Apresentação de Gráficos comparativos como auxílio à tomada de decisão.
- c) Evolução de uma *Dashboard* para visualização, controle e configuração do ambiente inteligente.
- d) Utilizar a biblioteca *Prophet* para realizar as previsões das séries temporais.

---

<sup>1</sup> <https://github.com/iotsmartcampus/Smart-Campus>

### 3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são detalhados os pontos mais importantes para o entendimento deste trabalho. Na Seção seção 3.1, é tratado o conceito de *Smart Campus* onde são abordadas as principais ideias, definições e explanando sobre as tecnologias. Na Seção 3.1.2, são detalhados os princípios da Internet das Coisas, onde é explicado as principais aplicações. Na Seção seção 3.2, é tratado o *middleware FIWARE* e os serviços utilizados neste trabalho. Na Seção 3.3, é apresentado o *Orion*<sup>1</sup> e sua utilização. Na Seção 3.4, é apresentado o *QuantumLeap*, é explicado sobre seu uso e sua arquitetura. Na Seção 3.5, é explicado sobre o conceito de série temporal. Semelhantemente, na Seção 3.6, são detalhados os procedimentos para prever dados de séries temporais com *Prophet*<sup>2</sup>.

#### 3.1 Smart Campus

De acordo com (DONG *et al.*, 2020) o desenvolvimento do campus inteligente não seria possível sem a inovação em tecnologia. Na literatura, o principal tecnologias que apoiam a revolução do campus inteligente incluem computação em nuvem, *Internet das Coisas* (IoT), *Realidade Aumentada* (RA) e *Inteligência Artificial* (IA). O princípio dessas tecnologias e seus benefícios para o campus inteligente são analisados nesta seção, o princípio de RA não será abordado.

A Figura 1 mostra algumas aplicações do poder do *Smart Campus* e algumas áreas que podemos utilizar as tecnologias. Por exemplo, esta Figura mostra que são utilizados os sensores de luminosidade e intensidade de iluminação, de movimento, controle do ar condicionado, medidor de umidade e temperatura. Esses sensores são bem usuais quando se deseja montar um ambiente inteligente.

O futuro é previsível. Dialogar com o futuro é necessário. A transformação digital nos conecta com os Mercados do Futuro, Menezes ().

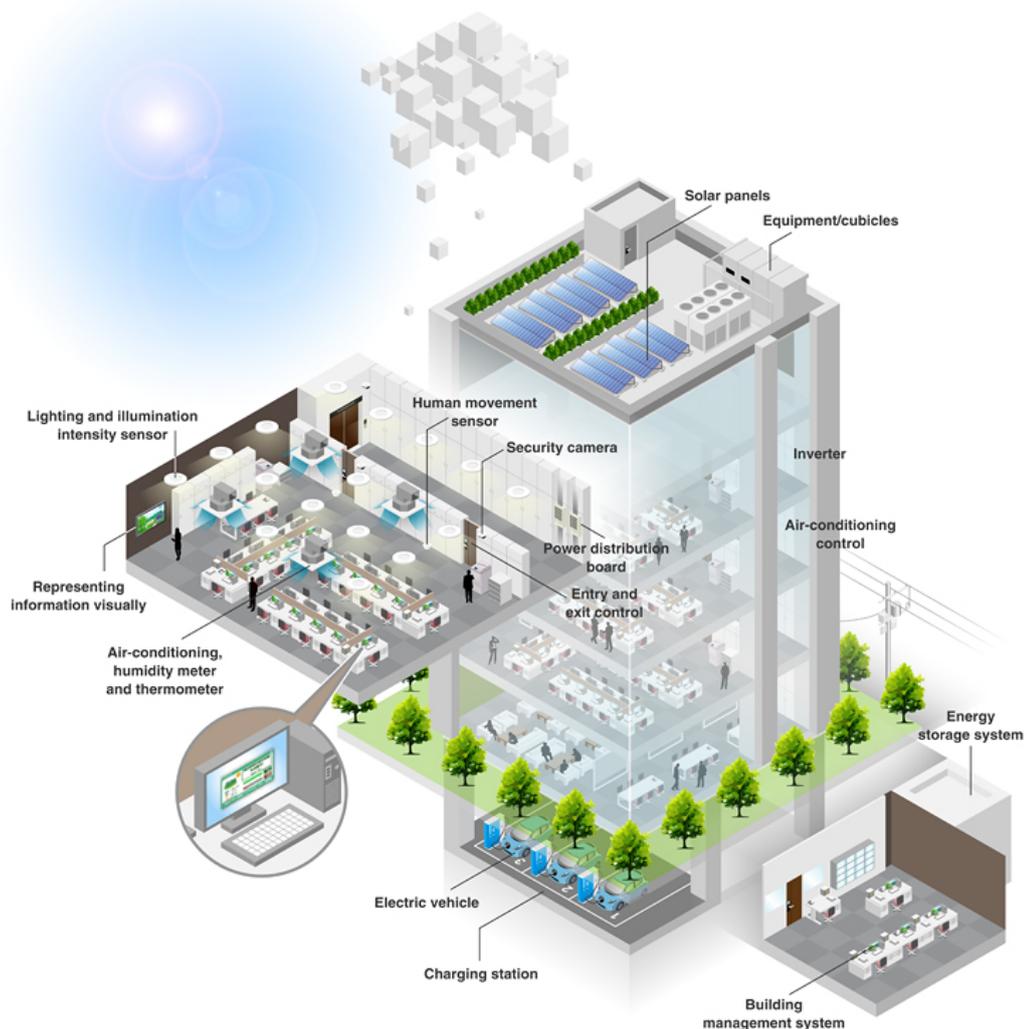
##### 3.1.1 Computação em Nuvem

De acordo com (PEDROSA; NOGUEIRA, 2011. Disponível em: <https://ic.unicamp.br/ducatte/m095352-120531-t2.pdf>. Acesso em: 01 jul. 2022.) a computação na nuvem ou *Cloud Computing*

<sup>1</sup> <https://fiware-orion.readthedocs.io/en/master/>

<sup>2</sup> <https://research.facebook.com/blog/2017/02/prophet-forecasting-at-scale/>

Figura 1 – As novas tecnologias permitem a conectividade de todos os tipos de sensores à Internet.



Fonte: adaptado da werk (2018).

é um novo modelo de computação que permite ao usuário final acessar uma grande quantidade de aplicações e serviços em qualquer lugar e independente da plataforma, bastando para isso ter um terminal conectado à “nuvem”.

Segundo (DONG *et al.*, 2020) a computação em nuvem é um modelo de computação distribuída que permite acesso de rede conveniente e sob demanda a um ambiente compartilhado de recursos de computação configuráveis (como redes, armazenamento e aplicativos) que podem ser rapidamente dimensionados, fornecidos e liberados mediante solicitação do usuário com interação mínima com o fornecedor. Em comparação com o computação convencional, a infraestrutura onde hardware e software são de propriedade e mantidos pelas organizações em suas instalações, a computação em nuvem permite atividades de aprendizagem em um ambiente não estruturado.

Segundo (PEDROSA; NOGUEIRA, 2011. Disponível em: <https://ic.unicamp.br/ducatte/mo401/095352-120531-t2.pdf>. Acesso em: 01 jul. 2022.) a nuvem é representada pela internet, isto é, a infraestrutura de comunicação composta por um conjunto de hardwares, softwares, interfaces, redes de telecomunicação, dispositivos de controle e de armazenamento que permitem a entrega da computação como serviço. A nuvem permite a que os alunos tenham acesso rápido a recursos de aprendizagem online e serviços a qualquer hora e em qualquer lugar, com escalabilidade infinita, maior conveniência e menor custo. A computação em nuvem vem desburocratizar a compra de grandes maquinários, assim, diminuindo as barreiras de entrada e democratizando a criação e testes de novas aplicações.

### 3.1.2 *Internet das Coisas*

A expressão Internet das Coisas IoT, cunhada em 1999 por Kevin Ashton, o pioneiro da tecnologia britânico que cofundou o *Auto-ID Center* no *Massachusetts Institute of Technology*, está se tornando cada vez mais popular. Ao abrir a *IoT Week 20131* com uma mensagem de vídeo pré-gravada, onde Ashton insistiu na percepção de que a IoT está aqui agora; não é o futuro, mas o presente segundo (BAUER *et al.*, 2013).

No trabalho de (DONG *et al.*, 2020) trata IoT como uma tecnologia incorporada com eletrônicos, dispositivos de detecção inteligentes, Internet e tecnologias de comunicação avançadas, IoT estende a conectividade com a Internet em dispositivos físicos e objetos cotidianos. Isso é previsto que o futuro paradigma da computação irá além do modo móvel tradicional baseado em *smartphones* e portáteis, e evoluir para um ambiente cercado por sistemas inteligentes e objetos em rede.

De acordo com (MADAKAM *et al.*, 2015) A Internet das Coisas é uma nova mudança de paradigma na área de TI. A frase “Internet das Coisas”, que também é brevemente conhecida como IoT, é cunhada a partir das duas palavras, ou seja, a primeira palavra é “Internet” e a segunda palavra é “Coisas”. A Internet é um sistema global de redes de computadores interconectadas que usam o conjunto de protocolos padrão da Internet (TCP/IP) para atender bilhões de usuários em todo o mundo.

Uma rede aberta e abrangente de objetos inteligentes que têm a capacidade de se autoorganizar, compartilhar informações, dados e recursos, reagindo e agindo diante de situações e mudanças no ambiente, (MADAKAM *et al.*, 2015)

Assim, com efeito, a Internet das Coisas é uma combinação de um impulso tecnológico e um impulso humano para conectividade cada vez maior com qualquer coisa que aconteça

no ambiente imediato e mais amplo com energia em uma única máquina para o ambiente: o ambiente como interface. Essa combinação *push-pull* o torna muito forte, imparável, rápido e extremamente disruptivo segundo (BAUER *et al.*, 2013).

Para chegar, se chegar, a uma definição da Internet das Coisas, e diferenciá-la de outras redes de sistemas interconectados, podemos partir do conjunto de funcionalidades e características do objeto (ou “coisa”), que tanto pode ser físico quanto virtual. Isso define a IoT como um sistema complexo e também suporta a criação e o design dos objetos, dos subsistemas e dos processos internos da IoT segundo (FAZION FILHO, 2016).

As funcionalidades de um objeto pertencente à internet das coisas são nove, distribuídas em três conjuntos:

1. Características
2. Relações
3. Interface

No conjunto das características do objeto na IoT, existem as seguintes atribuições segundo (FAZION FILHO, 2016):

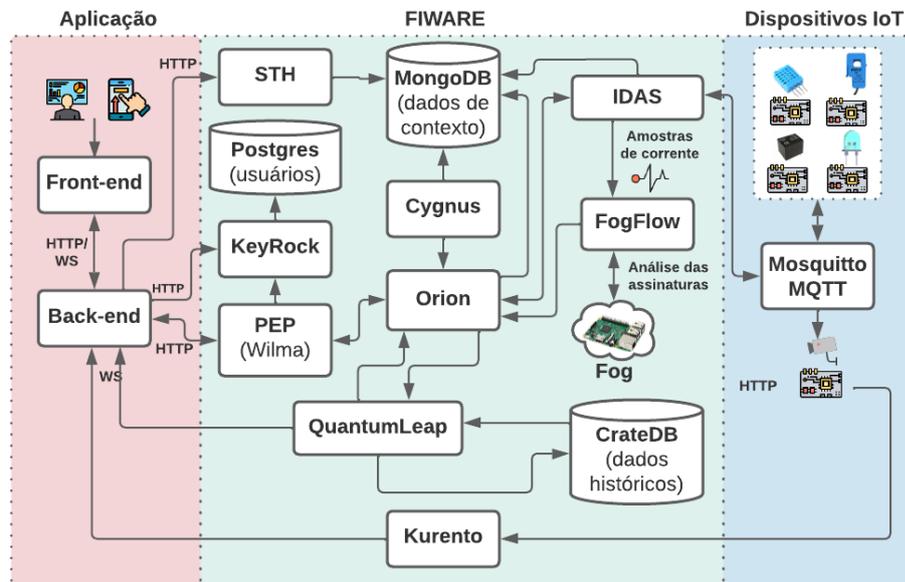
1. **Processamento**, que se refere à capacidade de processamento computacional inserida no objeto, ou “inteligência”, capaz de fazê-lo agir e responder às requisições da IoT e às suas aplicações;
2. **Endereçamento**, que se refere à capacidade do objeto de ser encontrado na IoT, ou seja, de ser localizado na rede por meio do roteamento;
3. **Identificação**, que se refere à identidade de cada objeto, fazendo-o único em toda a rede IoT;
4. **Localização**, atributo relacionado ao local físico em que o objeto se encontra, à sua posição no mapa geográfico.

De acordo com esta definição, a Figura 2 apresenta uma visão geral do escopo da IoT. A nuvem mostra a computação em nuvem e os dispositivos conectados mostram os sensores e atuadores do escopo da IoT. Por exemplo, o usuário pode ter acesso aos dispositivos de onde estiver.



informação armazenada são ferramentas de segurança, modelos avançados de armazenamento de dados, recuperação histórica de informação, ligação a aplicações de terceiros.

Figura 3 – Visão geral da arquitetura do FIWARE



Fonte: Adaptado pelo autor.

Neste trabalho, são empregados alguns *General Enable (GE)*'s do *FIWARE* como mostra a Figura 3. Deste modo, é utilizado o *QuantumLeap* para o armazenamento dos dados históricos, a ferramenta *Cygnus* para visualização dos gráficos para comparação, o *Orion* para gerenciar os dados e comunicação do contexto. Na sequência, são descritos o funcionamento e as propriedades de cada um desses microsserviços, e detalhadas funções que exercem no projeto, além de como se relacionam com as diferentes partes da infraestrutura de comunicação.

### 3.2.1 NGSI

O NGSI é um protocolo desenvolvido pela *Open Mobile Alliance (OMA)* para gerenciar informações de contexto. Ele fornece operações como:

1. Gerencie as Informações de Contexto sobre Entidades de Contexto, por exemplo, o tempo de vida e a qualidade das informações.
2. Acesso (consulta, subscrição/notificação) às Informações de Contexto disponíveis sobre Entidades de Contexto.

A versão *FIWARE* da interface OMA NGSI é uma API RESTful via HTTP. Seu objetivo é trocar informações de contexto. Os dois principais tipos de interação são:

1. consultas únicas para informações de contexto.
2. assinaturas para atualizações de informações de contexto (e as notificações correspondentes).

A Figura 4 mostra que o *Orion* receberá uma solicitação via NGSI e o banco de dados utilizado pelo *Orion* retornará com os dados das entidades de dados, assinaturas e registros.

Figura 4 – Arquitetura da requisição NGSI.



Fonte: adaptado da *FIWARE*.

Para verificar a integridade do serviço é executado o comando abaixo:

```
1 curl -X GET http://localhost:1026/version
```

E a resposta será semelhante à seguinte:

```

1 {
2   "orion": {
3     "version": "3.0.0",
4     "uptime": "0 d, 0 h, 17 m, 19 s",
5     "git_hash": "
6       d6f8f4c6c766a9093527027f0a4b3f906e7f04c4",
7     "compile_time": "Mon Apr 12 14:48:44 UTC 2021",
8     "compiled_by": "root",
9     "compiled_in": "f307ca0746f5",
10    "release_date": "Mon Apr 12 14:48:44 UTC 2021",
11    "machine": "x86_64",
12    "doc": "https://fiware-orion.rtdf.io/en/3.0.0/",
13    "libversions": {
14      "boost": "1_66",
  
```

```

14         "libcurl": "libcurl/7.61.1 OpenSSL/1.1.1g zlib
15             /1.2.11 nghttp2/1.33.0",
16         "libmicrohttpd": "0.9.70",
17         "openssl": "1.1",
18         "rapidjson": "1.1.0",
19         "mongoc": "1.17.4",
20         "bson": "1.17.4"
21     }
22 }

```

### 3.2.2 *NGSI-LD*

O *NGSI-LD* introduz um mecanismo padronizado para persistir e recuperar dados de contexto histórico. Convencionalmente, os agentes de contexto lidam apenas com o contexto atual - eles não têm memória, no entanto, os agentes de contexto *NGSI-LD* podem ser estendidos para oferecer dados de contexto histórico em uma variedade de formatos baseados em *JSON*. As funções temporais são classificadas como uma interface opcional para agentes de contexto *NGSI-LD*, pois a funcionalidade adicional tem um custo e não é obrigatória por padrão por motivos de desempenho.

O resultado é uma série de pontos de dados com carimbo de data/hora usando a *observedAt* propriedade-de-uma-propriedade . Cada ponto de dados com carimbo de data/hora representa o estado das entidades de contexto em um determinado momento. Os pontos de dados individuais não são particularmente úteis por si só, é somente através da combinação de uma série de pontos de dados que estatísticas significativas como máximos, mínimos e tendências podem ser observadas.

### 3.3 Orion Context Broker

O *Orion Context Broker* permite que você gerencie todo o ciclo de vida das informações de contexto, incluindo atualizações, consultas, registros e assinaturas. É uma implementação de servidor *NGSIv2* para gerenciar informações de contexto e sua disponibilidade. Usando o

*Orion Context Broker*, você pode criar elementos de contexto e gerenciá-los por meio de atualizações e consultas. Além disso, você pode assinar informações de contexto para que, quando ocorrer alguma condição (por exemplo, os elementos de contexto foram alterados), você receba uma notificação.

O *Orion Context Broker* é uma implementação do *Publish/Subscribe Context Broker GE*, fornecendo uma interface *NGSI*. Usando esta interface, os clientes podem fazer várias operações:

- a) Consultar informações de contexto. O *Orion Context Broker* <sup>4</sup> armazena informações de contexto atualizadas dos aplicativos, para que as consultas sejam resolvidas com base nessas informações. As informações de contexto consistem em entidades (por exemplo, um carro) e seus atributos (por exemplo, a velocidade ou a localização do carro).
- b) Atualizar informações de contexto, por exemplo enviar atualizações de temperatura.
- c) Seja notificado quando ocorrerem alterações nas informações de contexto (por exemplo, a temperatura mudou).
- d) Registrar aplicativos de provedor de contexto, por exemplo o fornecedor do sensor de temperatura dentro de uma sala.

### 3.4 QuantumLeap

O *QuantumLeap* <sup>5</sup> é um serviço *REST* para armazenamento, consulta e recuperação de dados espaço-temporais *NGSIv2* e *NGSI-LD* (suporte experimental). O *QuantumLeap* converte dados semiestruturados *NGSI* em formato tabular e os armazena em um banco de dados de séries temporais, associando cada registro do banco de dados a um índice de tempo e, se presente nos dados *NGSI*, a uma localização na Terra. Os clientes *REST* podem então recuperar entidades *NGSI* filtrando conjuntos de entidades por meio de intervalos de tempo e operadores espaciais. Observe que, do ponto de vista do cliente, essas consultas são definidas em entidades *NGSI* e não em tabelas de banco de dados. No entanto, a funcionalidade de consulta disponível por meio da interface *REST* é bastante básica e as consultas mais complexas normalmente exigem que os clientes usem o banco de dados diretamente.

<sup>4</sup> <https://github.com/telefonicaid/fiware-orion/blob/master/README.md>

<sup>5</sup> <https://quantumleap.readthedocs.io/en/latest/>

A especificação da *API REST*, apelidada de *NGSI-TSDB*, que o *QuantumLeap* implementa foi definida com o objetivo de fornecer uma interface *REST* independente de banco de dados para armazenamento, consulta e recuperação de séries temporais de entidades *NGSI* que poderiam ser o mais próximo possível da especificação *NGSI* em si. Assim, o *NGSI-TSDB* fornece um mecanismo uniforme e familiar (para desenvolvedores de *FIWARE*) para acessar dados de séries temporais que permite implementar serviços como o *QuantumLeap* para suportar de forma transparente vários *back-ends* de banco de dados. De fato, atualmente o *QuantumLeap* suporta *CrateDB* e *Timescale* como bancos de dados de *back-end*. Neste trabalho está sendo usando o *CrateDB*.

As consultas temporais *NGSI-LD* parecem ter uma semântica que implica que apenas valores numéricos são rastreados em séries temporais. Este nunca foi o caso do *QuantumLeap* que rastreia ao longo do tempo qualquer atributo, pois eles também podem mudar.

A semântica *NGSI-LD* também parece rastrear valores ao longo do tempo de atributos únicos. O *QuantumLeap* para permitir a recuperação de valores de entidade completos em um determinado momento armazena toda a entidade em uma única tabela (isso evita a necessidade de *JOINS* que são notoriamente demorados - mas, por outro lado, geram dados mais esparsos). Ao fazer isso, foi criado para a entidade um índice de tempo único, isso se deve ao fato de que, embora diferentes atributos *dateTime* possam ser definidos e, portanto, consultados, apenas um pode ser usado para indexar séries temporais em todos os bancos de dados de séries temporais modernos (para obter desempenho). Isso implica que temos uma política para calcular esse índice de tempo (seja personalizado e referente a um atributo da entidade, ou usando os metadados de tempo "mais recentes" vinculados à entidade ou a um atributo). O problema é que, se a carga de notificação enviada ao *QuantumLeap* incluir todos os atributos, o *QuantumLeap* "marcará" todos os valores (também os antigos) com esse carimbo de data/hora. Isso significa que a capacidade de rastrear um valor específico de um atributo em um momento depende da notificação real.

Em suma, dado que o objetivo do trabalho é garantir a compatibilidade direta (o armazenamento de dados como *NGSIv2* pode ser consultado no futuro como *NGSI-LD*) e a compatibilidade com versões anteriores (o armazenamento de dados como *NGSI-LD* pode ser consultado como *NGSIv2*), implementando a *API* temporal *NGSI-LD*, pode não ser 100% compatível com as especificações.

### 3.4.1 Arquitetura

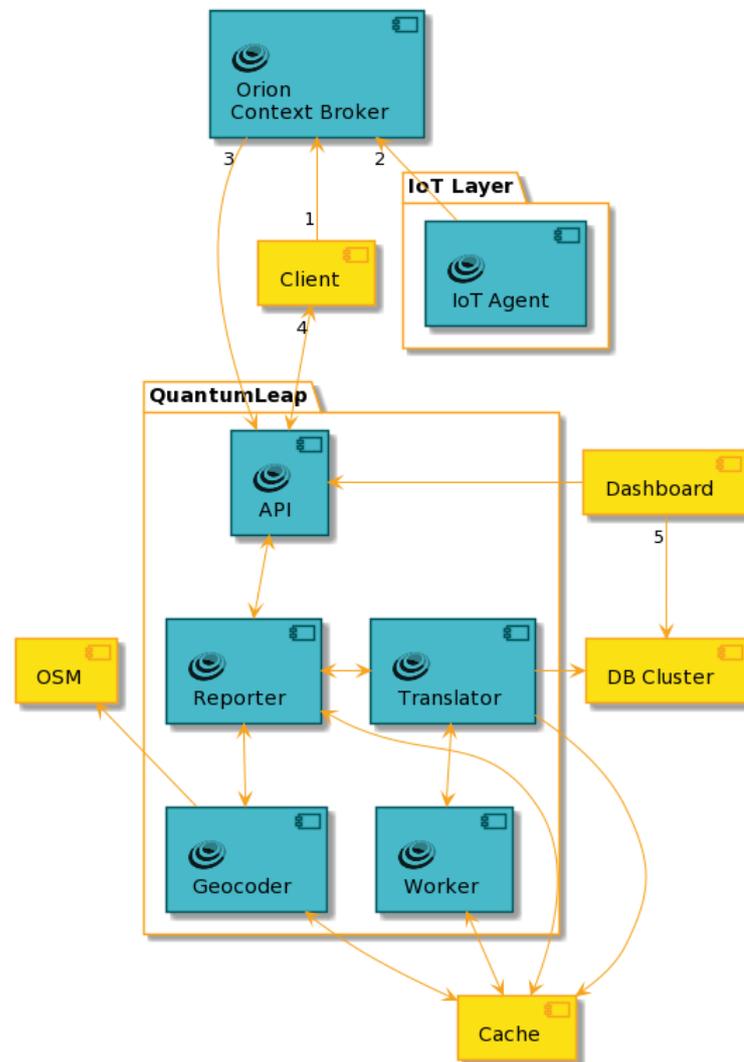
Normalmente, o *QuantumLeap* adquire dados de IoT, na forma de entidades *NGSI*, de uma camada *FIWARE IoT Agent* indiretamente por meio de notificações *NGSI* configuradas antecipadamente com o agente de contexto, Orion. Muitas vezes, ferramentas de visualização como o *Grafana* também são implantadas para visualizar os dados de séries temporais que o *QuantumLeap* armazena no banco de dados.

A Figura 5 ilustra os relacionamentos e interações entre esses sistemas em um cenário típico de implantação do *QuantumLeap*. O componente *Reporter* do *QuantumLeap* analisa e valida os dados da requisição. Além disso, se a geocodificação estiver configurada, o *Reporter* invoca o componente *Geocoder* para harmonizar a representação de localização das entidades notificadas, o que envolve a busca de informações geográficas no *OpenStreetMap* (OSM). Nesse estágio, os dados são imediatamente processados ou armazenados no cache para processamento posterior. No primeiro caso, o *Reporter* passa as entidades *NGSI* validadas e harmonizadas para um componente *Translator*. No segundo caso, o *Reporter* armazena nas entidades *NGSI* validadas e harmonizadas ao *Cache*, que está agindo como uma fila de mensagens. O componente *Worker* lerá as mensagens pendentes a serem processadas e as passará para um componente *Translator*. O *Translator* converte entidades *NGSI* em formato tabular e as mantém como registros de séries temporais no banco de dados. Existe um componente *Translator* em correspondência de cada *back-end* de banco de dados suportado.

### 3.4.2 *QuantumLeap* versus *STH-comet*

Embora o *QuantumLeap* e o *FIWARE STH Comet* compartilhem objetivos semelhantes, o *Comet* não suporta vários banco de dados, suporta apenas o *MongoDB* e também não suporta *NGSIv2*. Embora o *Comet* em si seja um bom software, algumas das necessidades e suposições que motivaram seus desenvolvimentos não são mais atuais, por exemplo, o *STH* não está recebendo mais atualizações. O *QuantumLeap* começou como uma exploração de uma maneira alternativa de disponibilizar dados históricos para o ecossistema *FIWARE* sem se comprometer com um *back-end* de banco de dados específico.

Figura 5 – Arquitetura do *QuantumLeap*



Fonte: adaptado da QuantumLeap (2021).

### 3.5 Séries Temporais

De acordo com (BROCKWELL; DAVIS, 2016) uma série temporal é um conjunto de observações  $x_t$ , cada uma sendo registrada em um tempo específico  $t$ . Uma série temporal discreta é aquela em que o conjunto  $T_0$  de tempos em que as observações são feitas é um conjunto discreto, como é o caso, por exemplo, quando as observações são feitas em intervalos de tempo fixos. As séries temporais contínuas são obtidas quando as observações são registradas continuamente ao longo de algum intervalo de tempo, por exemplo, quando  $T_0 = [0, 1]$ .

Segundo (CHATFIELD, 2000) as aplicações da previsão de séries temporais incluem:

1. Planejamento econômico
2. Previsão de vendas
3. Controle de estoque (ou estoque)

4. Planejamento de produção e capacidade
5. A avaliação de estratégias econômicas alternativas
6. Orçamento
7. Gestão de risco financeiro
8. Avaliação do modelo

Por exemplo, boas previsões de orçamento facilitarão o planejamento dos gestores se prepararem para os próximos meses. No trabalho de (CHATFIELD, 2000) os dados de séries temporais fornecem uma excelente oportunidade para observar o que é chamado de comportamento fora da amostra. Um modelo de série temporal fornecerá previsões de novas observações futuras que podem ser comparadas com o que é realmente observado. Se houver boa concordância, argumenta-se que isso fornece uma verificação mais convincente do modelo do que o ajuste na amostra.

### ***3.5.1 Diferentes tipos de séries temporais***

De acordo com (CHATFIELD, 2000) uma série temporal é um conjunto de observações medidas sequencialmente ao longo do tempo. Essas medições podem ser feitas continuamente ao longo do tempo ou tomadas em um conjunto discreto de pontos de tempo. Por convenção, esses dois tipos de séries são chamados de séries temporais contínuas e discretas, respectivamente, embora a variável medida possa ser discreta ou contínua em ambos os casos. Em outras palavras, para séries temporais discretas, por exemplo, é o eixo do tempo que é discreto.

Segundo (CHATFIELD, 2000) para uma série temporal contínua, a variável observada é tipicamente uma variável contínua registrada continuamente em um traço, como uma medida da atividade cerebral registrada em uma máquina de EEG. O método usual de analisar tal série é amostrar (ou digitalizar) a série em intervalos de tempo iguais para fornecer uma série temporal discreta. Pouca ou nenhuma informação é perdida por este processo, desde que o intervalo de amostragem seja pequeno o suficiente e, portanto, não há necessidade de dizer mais nada sobre séries contínuas.

### ***3.5.2 Objetivos da análise de séries temporais***

A característica especial dos dados de séries temporais é que as observações geralmente não são independentes e, portanto, a análise deve levar em conta a ordem em que as

observações são coletadas. Efetivamente, cada observação na variável medida é uma observação bivariada com o tempo como a segunda variável de acordo com (CHATFIELD, 2000).

Para (CHATFIELD, 2000) os principais objetivos da análise de séries temporais são:

1. Descrição: Para descrever os dados usando estatísticas resumidas e/ou métodos gráficos. Um gráfico de tempo dos dados é particularmente valioso.
2. Modelagem: Encontrar um modelo estatístico adequado para descrever o processo de geração de dados.
3. Previsão: Estimar os valores futuros da série.
4. Ao controle: Boas previsões permitem que o analista tome medidas para controlar um determinado processo, seja ele um processo industrial, uma economia ou qualquer outra coisa.

### 3.5.3 *Previsão de séries temporais*

Para (ZHANG, 2003) a previsão de séries temporais é uma área importante de previsão na qual observações passadas da mesma variável são coletadas e analisadas para desenvolver um modelo que descreva a relação subjacente. O modelo é então usado para extrapolar a série temporal para o futuro. Essa abordagem de modelagem é particularmente útil quando há pouco conhecimento sobre o processo de geração de dados subjacente ou quando não há um modelo explicativo satisfatório que relacione a variável de previsão com outras variáveis explicativas. Muito esforço tem sido dedicado ao longo das últimas décadas para o desenvolvimento e aprimoramento de modelos de previsão de séries temporais.

## 3.6 Prophet

O *Prophet* é um procedimento para prever dados de séries temporais com base em um modelo aditivo em que as tendências não lineares são ajustadas à sazonalidade anual, semanal e diária, além dos efeitos de feriados. Funciona melhor com séries temporais com fortes efeitos sazonais e várias temporadas de dados históricos. O *Prophet* é robusto a dados ausentes e mudanças na tendência e normalmente lida bem com valores discrepantes.

A entrada para *Prophet* é sempre um dataframe com duas colunas: *ds* e *y*. A coluna *ds*(datastamp) deve ter um formato esperado pelos Pandas, de preferência AAAA-MM-DD para uma data ou AAAA-MM-DD HH:MM:SS para um carimbo de data/hora. A coluna *y* deve ser

numérica e representa a medida que desejamos prever como é mostrado na Figura 6.

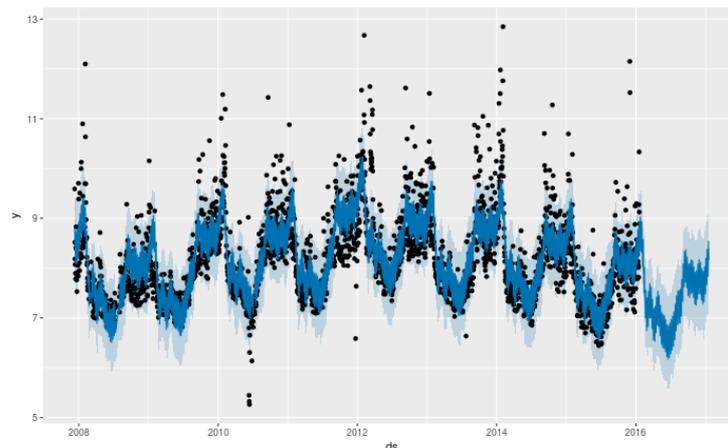
Figura 6 – Dataframe aceito pelo Prophet.

	DS	Y
<b>0</b>	10-12-2007	9.590761
<b>1</b>	11-12-2007	8.519590
<b>2</b>	12-12-2007	8.183677
<b>3</b>	13-12-2007	8.072467
<b>4</b>	14-12-2007	7.893572

Fonte: adaptado Prophet.

### 3.6.1 Como funciona?

Figura 7 – Gráfico gerado pelo Prophet.



Fonte: adaptado Prophet.

Em sua essência, o procedimento do *Prophet* é um modelo de regressão aditivo com quatro componentes principais:

1. Uma tendência de curva de crescimento linear ou logística por partes. O Prophet detecta automaticamente as mudanças nas tendências selecionando os pontos de mudança dos dados.
2. Um componente sazonal anual modelado usando a série de *Fourier*.
3. Um componente sazonal semanal usando variáveis *dummy*.
4. Uma lista fornecida pelo usuário de feriados importantes.

A Figura 7 mostra as visualizações de página em escala de log da página da Wikipedia de Peyton Manning<sup>6</sup> que foi baixado usando o pacote `wikipediatrend`<sup>7</sup>. Como Peyton Manning é um jogador de futebol americano, pode-se notar que a sazonalidade anual desempenha um papel importante, enquanto a periodicidade semanal também está claramente presente.

---

<sup>6</sup> <https://en.wikipedia.org/wiki/PeytonManning>

<sup>7</sup> <https://cran.r-project.org/web/packages/wikipediatrend/index.html>

## 4 TRABALHOS RELACIONADOS

Na literatura, podem ser encontrados vários trabalhos que se relacionam tanto no que diz respeito ao uso de ferramentas e tecnologias IoT quanto na utilização de técnicas de aprendizado de máquina aplicadas na identificação de padrões, mais precisamente, na classificação e obtenção de dados e utilização desses dados para realizar previsões. Dessa forma, este capítulo apresenta, de modo detalhado, trabalhos compatíveis com essas duas vertentes, onde são, principalmente, explorados conceitos e técnicas implementadas neste trabalho.

### 4.1 Smart CEI Moncloa: An IoT-based Platform for People Flow and Environmental Monitoring on a Smart University Campus (ALVAREZ-CAMPANA *et al.*, 2017)

No trabalho (ALVAREZ-CAMPANA *et al.*, 2017) é desenvolvido plataforma IoT que foi implantada nas escolas de engenharia da Universidade Politécnica de Madrid no Campus Moncloa que representa um bom exemplo de bancada de testes ou experimentação com serviços *Smart City*. Esse trabalho apresenta as principais características desta plataforma, dando especial ênfase aos desafios tecnológicos enfrentados e às soluções adotadas, bem como na funcionalidade, serviços e potencial que a plataforma oferece.

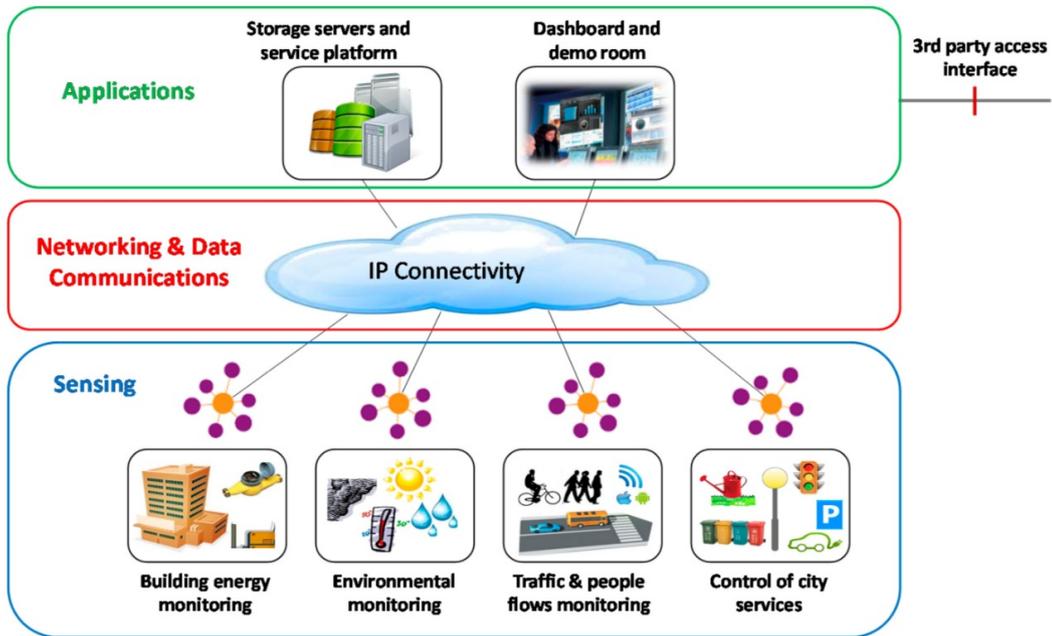
#### 4.1.1 Arquitetura

Portanto, o objetivo da arquitetura inicial, ilustrada na Figura 8, era implantar uma plataforma que incorporasse a infraestrutura de TIC necessária para fornecer um conjunto de serviços básicos, juntamente com uma rede de sensores que permitisse demonstrar a capacidade e escalabilidade da solução adotada. Assim, a plataforma precisava atender aos seguintes requisitos principais:

- a) deve ser de baixo custo e escalável.
- b) deve ser baseado em soluções padronizadas e abertas. que permitam incorporar novos serviços à plataforma de forma integrada.

A Figura 9 mostra a visão geral dos dados fornecidos pelo *Dashboard*, mostra a visão do serviço de monitoramento ambiental acessível a todos. No lado esquerdo da Figura 9, pode ser visto um resumo em tempo real dos parâmetros monitorados. No lado direito da Figura 9, podem ser vistos os gráficos semanais para cada parâmetro. Esses gráficos permitem observar uma clara correlação entre os parâmetros monitorados, dependendo se é dia ou noite. Ressalta-se

Figura 8 – Arquitetura Smart CEI Moncloa



Fonte: adaptado da Alvarez-Campana *et al.* (2017).

que não se observa grande diferença entre dias de semana e finais de semana, pois as bibliotecas das escolas estão abertas durante o final de semana. Os usuários cadastrados também podem acessar os dados coletados de sensores específicos selecionando-os.

Figura 9 – Dashboard Smart CEI Moncloa.



Fonte: adaptado da Alvarez-Campana *et al.* (2017).

### 4.1.2 *Sensores e Atuadores*

A camada de sensoriamento é composta por um conjunto de sensores (e eventualmente também atuadores) que estão distribuídos geograficamente pelo campus. Esses aparelhos são responsáveis por medir a parâmetros de interesse para os serviços que são entregues ou os experimentos que são realizados no topo da plataforma. A Figura 8 mostra alguns exemplos de redes de sensores relacionadas a serviços que podem potencialmente ser testados e entregues no Smart CEI Moncloa, como monitoramento de energia predial, monitoramento ambiental, monitoramento de tráfego e fluxo de pessoas ou controle de serviços da cidade (por exemplo, estacionamento, irrigação de iluminação).

### 4.1.3 *Aplicação*

A camada de aplicação é onde residem a lógica de serviço e os algoritmos. Ele é composto pelos servidores de armazenamento, a plataforma de serviço e o painel. Os servidores de armazenamento são responsáveis pelo armazenamento dos dados gerados pelos sensores e para fornecer acesso a esses dados às aplicações que assim o requerem. Eles fornecem recursos de *Big Data* para lidar com grandes volumes de dados, bem como suporte ao formato *Open Data* para aplicativos de terceiros. A plataforma de serviços é responsável por executar as aplicações baseadas em *Service-Oriented Architecture* (SOA). O painel é acessível via web, que permite acesso onipresente aos dados de controle e visualização da plataforma. Além disso, há uma sala de demonstração localizada na Escola de Engenharia de Telecomunicações e equipada com grandes telas que permitem mostrar as capacidades da plataforma para estudantes, pesquisadores ou outros interessados. No que diz respeito aos serviços, o *Smart CEI Moncloa* inclui um conjunto de serviços iniciais com a ideia de incluir progressivamente novos ao longo do seu ciclo de vida. Os serviços piloto iniciais são:

- a) Monitorização do fluxo de pessoas, que permite contar pessoas e aplicações associadas, como análise de padrões de movimento, locais com maior trânsito de pessoas, tempo de permanência em locais relevantes, etc.
- b) Monitorização ambiental, que permite analisar vários parâmetros ambientais, como temperatura, umidade, luminosidade, nível de ruído ou composição do ar, tanto no interior como no exterior.

## 4.2 Energy Efficiency in Smart Buildings: An IoT-Based Air Conditioning Control System (ROCHA *et al.*, 2019)

No trabalho (ROCHA *et al.*, 2019) relata o uso contínuo de aparelhos elétricos de alta potência, como ar condicionadores em edifícios comerciais e residenciais tem contribuído para o uso ineficiente dos recursos energéticos. Para enfrentar esse cenário, os edifícios inteligentes focam em minimizar o consumo de energia e melhorar a experiência e a produtividade dos usuários nesses ambientes. Com o objetivo de otimizar o uso de condicionadores de ar visando a eficiência energética, este trabalho apresenta o *Smart Place*, um sistema de gerenciamento de ambientes baseado em Internet das Coisas (IoT) para controle automático desses equipamentos.

Nesse sistema, sensores e câmeras de vídeo coletam dados de temperatura, umidade e presença de pessoas nos espaços monitorados. Esses dados são parâmetros para realizar intervenções nos aparelhos de ar condicionado, a fim de evitar mantê-los ligados quando o ambiente não estiver sendo utilizado. O sistema também oferece interface Web para gerenciamento de dispositivos e ambientes monitorados, além de ser integrado à plataforma *FIWARE* como *middleware* subjacente. Esse artigo descreve o *Smart Place*, sua arquitetura, e seu funcionamento na Universidade Federal do Rio Grande do Norte (UFRN), Natal, Brasil. O artigo também discute os benefícios resultado da intervenção automática realizada pela *Smart Place*, que conseguiu economizar 61,8% no consumo de energia em comparação com o controle manual tradicional em um conjunto de salas de aula.

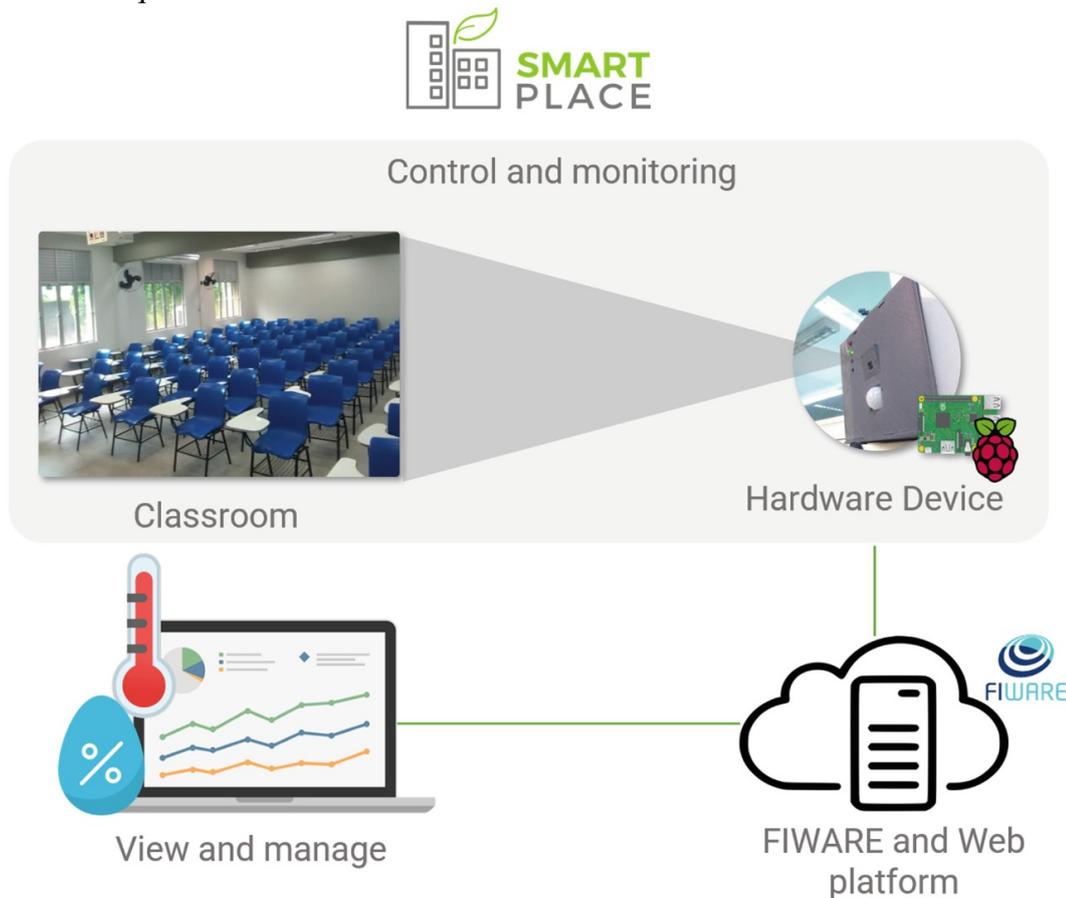
### 4.2.1 Arquitetura

A Figura 10 mostra o dispositivo de hardware que é o componente destinado a monitorar e controlar condicionadores de ar. Este componente é composto por um microcomputador *Raspberry Pi* que:

- a) atua nos condicionadores de ar enviando sinais infravermelhos e
- b) gerencia dados medidos por diferentes sensores conectados ao microcomputador para medir variáveis ambientais como movimento, temperatura e umidade, além de uma câmera para captura de imagens.

O *Raspberry Pi* também executa o algoritmo de decisão que decide sobre a ação de atuação de acordo com os dados recebidos dos sensores e da câmera. Tanto as medições provenientes dos sensores e da câmera quanto as decisões sobre a atuação nos dispositivos de ar

Figura 10 – Arquitetura Smart Place



Fonte: adaptado da Rocha *et al.* (2019).

condicionado são encaminhadas para a plataforma de *middleware* subjacente.

O sistema é composto por:

- I. um dispositivo de hardware, que é responsável por coletar.
- II. dados relacionados ao ambiente e envio de comandos para aparelhos de ar condicionado.
- III. uma plataforma de *middleware*, que fornece serviços como gerenciamento de contexto, gerenciamento de dispositivos e persistência de dados.
- IV. uma plataforma Web capaz de acessar dados persistentes dos dispositivos e disponibilizá-los para os usuários do sistema por meio de interfaces de consulta.

#### 4.2.2 Sensores e Atuadores

O dispositivo de hardware é o componente destinado a monitorar e controlar ar condicionado. Este componente é composto por um microcomputador *Raspberry Pi* que atua nos condicionadores de ar enviando sinais infravermelhos e gerencia os dados medidos por diferentes sensores conectados ao microcomputador para medir variáveis ambientais como movimento,

temperatura e umidade, como bem como uma câmera para captura de imagem. O *Raspberry Pi* também executa o algoritmo de decisão que decide sobre a ação de atuação de acordo com os dados recebidos dos sensores e da câmera. Tanto as medições provenientes dos sensores e da câmera quanto as decisões sobre a atuação nos dispositivos de ar condicionado são encaminhadas para a plataforma de *middleware*.

Para apoiar o processo de decisão, o dispositivo é configurado para acessar um sistema de terceiros, por exemplo, um sistema que registra o uso programado para as salas. Os dados desse sistema são salvos localmente no *Raspberry Pi* para uso offline até o final do período acadêmico atual, quando serão atualizados. Se o sensor de presença detectar movimento na sala, as imagens fornecidas pela câmera não serão usadas para decidir sobre a ativação do ar condicionado.

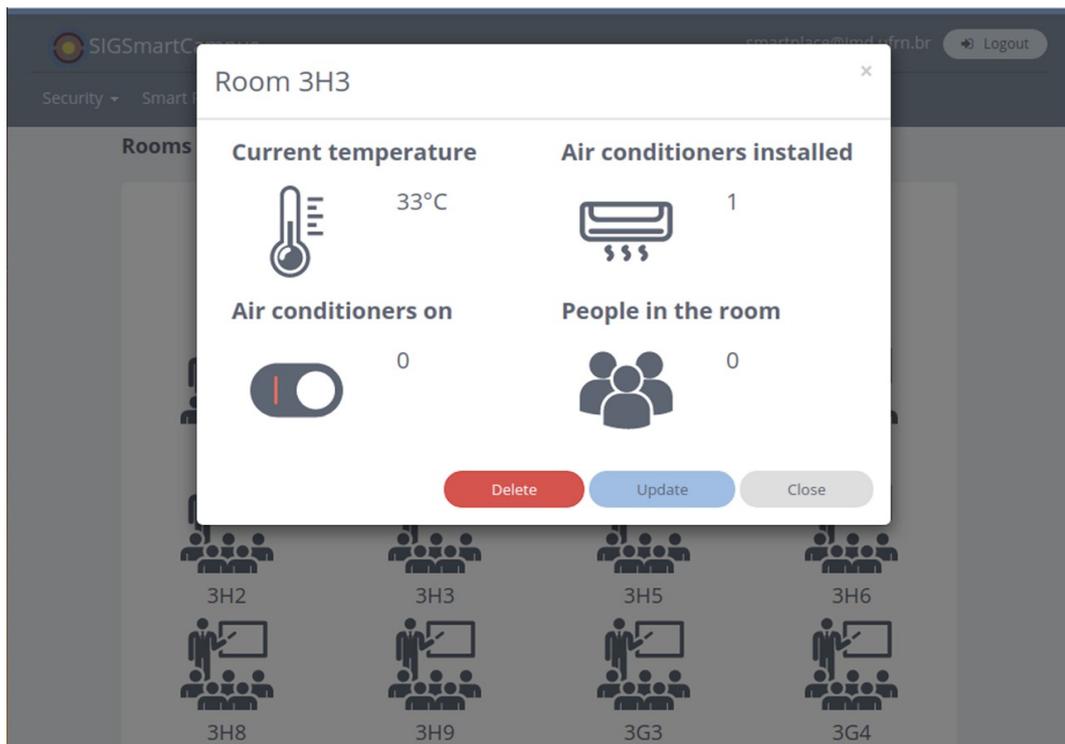
#### 4.2.3 Aplicação

A plataforma *Smart Place Web* fornece serviços para configurar entidades e dispositivos em *FIWARE GEs*, acessar dados históricos e monitorar eventos de aplicativos. Como mostra a Figura 11 por meio da plataforma Web, os usuários podem gerenciar ambientes monitorados e receber alertas sobre a ocorrência de comportamentos inesperados. A plataforma Web também apresenta gráficos de dados de forma condensada para permitir uma melhor análise das informações disponíveis. Os usuários podem definir parâmetros para a execução do sistema e acionar comandos que são enviados aos dispositivos, como ligar ou desligar um ar condicionado, definir a temperatura ambiente desejada, etc.

### 4.3 Desenvolvimento de uma nova Metodologia para Previsão do Consumo de Energia Elétrica de Curto Prazo Utilizando Redes Neurais Artificiais e Decomposição de Séries Temporais (AMARAL, 2019)

Os consumidores residenciais estão cada vez mais ganhando acesso a novas tecnologias de medição e operação remota, com a disseminação de redes inteligentes. Essas oportunidades trouxeram melhorias em termos de qualidade de serviço e o estabelecimento de novos níveis de economia financeira. Estudos recentes demonstraram que um aumento significativo na eficiência energética está relacionado às mudanças nos hábitos de consumo, induzidas por novas tecnologias que dão acesso às informações sobre a qualidade e a quantidade

Figura 11 – Dashboard Smart Place



Fonte: adaptado da Rocha *et al.* (2019).

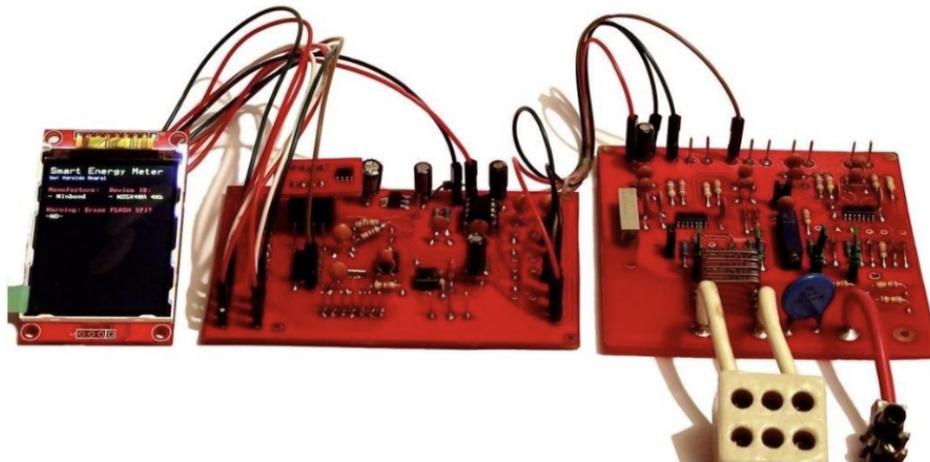
da energia consumida. Nesse contexto, a previsão de consumo de energia de curto prazo é uma ferramenta de conscientização baseada no fornecimento de informações úteis prontas para ajudar na busca dos mais eficientes padrões de consumo e economia de energia. Esta tese apresenta o desenvolvimento de uma nova metodologia para a previsão do consumo de energia residencial, utilizando redes neurais artificiais e decomposição de séries temporais. Vários testes foram realizados para obter a melhor configuração para o problema de previsão do consumo de eletricidade. Os testes também foram realizados comparativamente com os métodos combinados disponíveis, permitindo uma avaliação dos resultados obtidos. Os resultados demonstrados por essa metodologia atingiram um baixo erro percentual médio e um erro percentual absoluto médio baixo no horizonte de previsão, que são as métricas mais utilizadas nesse tipo de estudo.

#### 4.3.1 Arquitetura

O desenvolvimento inicial se baseou no software *Matlab*, utilizando como base os dados coletados com o medidor desenvolvido na dissertação de mestrado. Esse medidor possuía características técnicas que suprem todas as necessidades com relação à medição, dentre elas, exatidão superior a 1%, tanto para medição da tensão como da corrente, amplo range para medição da corrente, processamento de grandezas relacionadas à qualidade da energia como

o fator potência (fp) e a taxa de distorção harmônica relativa às formas de onda de tensão e corrente, se assemelhando muito aos dados fornecidos por um analisador de qualidade da energia convencional.

Figura 12 – Smart Meter



Fonte: adaptado da Amaral (2019).

#### 4.3.2 Sensores e Atuadores

O dispositivo é apresentado na Figura 12 e possui a capacidade de coletar os seguintes dados:

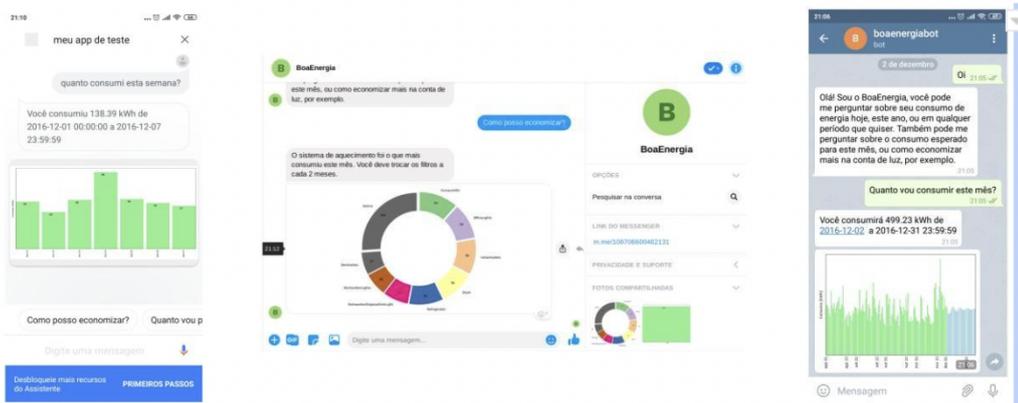
1. Data e hora da coleta;
2. Consumo acumulado total em Wh e VAh;
3. Medição RMS de tensão e corrente;
4. Cálculo das potências aparente, ativa, reativa e reativa de distorção harmônica;
5. Taxa de distorção harmônica (Total Harmonic Distortion - THD) de tensão e corrente; e
6. Formas de onda de tensão e corrente.

#### 4.3.3 Aplicação

A integração da previsão de consumo em um medidor inteligente ou mesmo em um sistema auxiliar torna-se um grande atrativo por unificar o processo em um único dispositivo ou sistema integrado. Como sistema auxiliar pode-se considerar, por exemplo, o controlador central de uma casa inteligente que receberá de um medidor inteligente os dados referentes ao consumo

de energia, ou que realize as medições por conta própria, para então poder realizar a previsão do consumo que pode ser apresentada ao consumidor através de uma interface gráfica dedicada ou mesmo através de um APP para o *smartphone*. Exemplos de apresentação das informações em uma casa conectada com um sistema de gerenciamento são apresentados conforme é apresentado na Figura 13.

Figura 13 – App Smart Meter



Fonte: adaptado da Amaral (2019).

#### 4.4 Comparação entre os trabalhos

No Quadro 1 são listados os trabalhos relacionados além do trabalho proposto. Assim, os trabalhos são comparados nos quesitos de: Objetivo, uso do *middleware*, sensores, arquitetura e algoritmos. Como pode ser observado, os trabalhos são amplamente voltados aos conceitos de IoT.

Quadro 1 – Comparação entre os trabalhos

Trabalho	(ALVAREZ-CAMPANA <i>et al.</i> , 2017)	(ROCHA <i>et al.</i> , 2019)	(AMARAL, 2019)	Este trabalho
Objetivo	Monitoramento de fluxo de pessoas e ambiente	Monitoramento do ar condicionado	Monitorar o consumo de energia elétrica	Monitorar energia, temperatura e umidade
Sensores	Presença via <i>Wi-Fi</i> , temperatura, umidade, luminosidade, barulho	Presença, temperatura, infravermelho, movimento e câmera	Corrente	Corrente, umidade, temperatura, infravermelho
Aplicação	Web	Web	Móvel	Web
Séries temporais	Menciona, mas não processa	Não relatado	TensorFlow, Caffe	<i>QuantumLeap</i>
Algoritmos	Trabalho futuro	algoritmo de decisão	Keras	<i>Prophet</i>
Middleware	Não relatado	<i>FIWARE</i>	Não relatado	<i>FIWARE</i>

Fonte: elaborado pelo autor.

No que se refere aos objetivos, a maioria utilizou sensores para monitorar o ambiente, vale salientar que este trabalho e o (AMARAL, 2019) utilizaram os sensores para monitorar o consumo de energia elétrica. Além dos autores (ROCHA *et al.*, 2019) que somente tem o objetivo de monitorar o ar condicionado da sala e desligar se não tiver presença. Vale salientar que este trabalho usa o *QuantumLeap* para realizar o armazenamento dos dados históricos, buscando a facilidade na implantação.

No que diz respeito a utilização de sensores, a maioria utiliza os sensores de temperatura e umidade. Os autores (ALVAREZ-CAMPANA *et al.*, 2017) e os (ROCHA *et al.*, 2019) são os que usam mais sensores para realizar o monitoramento. A maioria dos autores disponibilizam suas aplicações por meio da *Web*, somente (AMARAL, 2019) disponibilizou a aplicação móvel.

Sobre as séries temporais, os autores de (ALVAREZ-CAMPANA *et al.*, 2017) citam sobre o assunto, mas não utilizam no trabalho, citam como trabalho futuro. Os autores de (ROCHA *et al.*, 2019) não relatam sobre o assunto, o autor (AMARAL, 2019) utiliza as redes neurais artificiais e utiliza a decomposição temporal para aplicar aos dados coletados. Neste trabalho utiliza o *QuantumLeap* para processar os eventos complexos e a biblioteca *Prophet* para realizar as previsões.

Os autores de (ALVAREZ-CAMPANA *et al.*, 2017) mencionam os algoritmos nos trabalhos futuros, os autores de (ROCHA *et al.*, 2019) utilizam um algoritmo de decisão simples para ligar ou desligar o ar condicionado, o autor (AMARAL, 2019) utiliza os algoritmos Keras. Neste trabalho é utilizado o algoritmo *Prophet*.

No que diz respeito aos *middlewares*, os autores de (ALVAREZ-CAMPANA *et al.*, 2017) não utilizam um *middleware*, utilizam somente o *Raspberry Pi* para fazer o meio de campo. Os autores (ROCHA *et al.*, 2019) e este trabalho utilizam o *middleware FIWARE*. Somente o autor (AMARAL, 2019) que não especificou o tipo de *middleware*.

## 5 SOLUÇÃO PROPOSTA

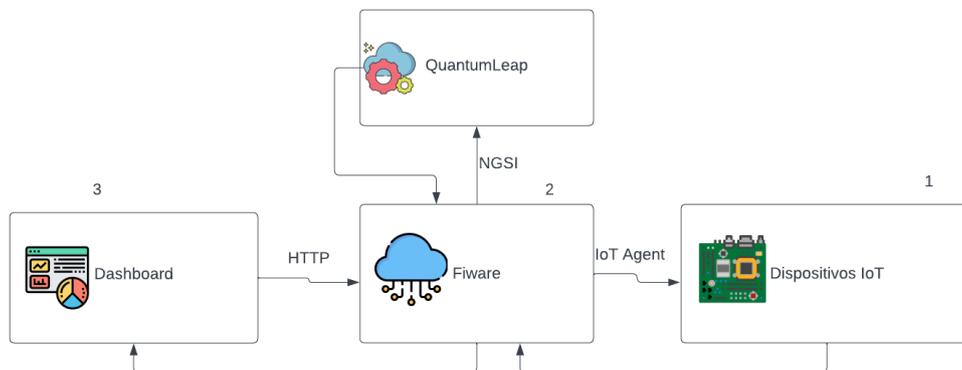
Neste capítulo são discutidas as ações necessárias para a execução da solução proposta. Na Seção 5.1, é explanada a avaliação das tecnologias e arquitetura de comunicação utilizadas no projeto. Na Seção 5.2, mostra alguns passos no uso do *Propher*. Na Seção 5.3, fala do ambiente que foi executado o teste e armazenamento dos dados. Na Seção 5.4, mostra a aplicação desenvolvida. Semelhantemente, na Seção 6 mostra a avaliação da solução proposta do trabalho.

### 5.1 Avaliação das tecnologias

Nesta fase do projeto foi realizado um estudo exploratório para encontrar e validar os serviços de comunicação, armazenamento e gerenciamento dos dados. Foram feitas inúmeras pesquisas e testes para a escolha das tecnologias que fornecem os recursos principais para este trabalho, como por exemplo, o armazenamento histórico e comunicação entre as entidades da arquitetura. Desse modo, foi decidido que a arquitetura das máquinas virtuais do Campus seria empregada de modo a suprir a comunicação de um ambiente IoT, como velocidade de *deploy*, processamento, comunicação e escalabilidade.

Assim, foram definidas as entidades que seriam monitoradas e enviadas ao *QuantumLeap* do ambiente inteligente. A comunicação foi realizada pelo protocolo *NGSI* por requererem recursos mínimos e podem ser usados em microcontroladores pequenos. Os cabeçalhos das mensagens *NGSI* são pequenos para otimizar a largura de banda da rede como mostra a Figura 14. Assim o *NGSI* é responsável para fazer o meio de campo entre o *Orion* e o *QuantumLeap*.

Figura 14 – Arquitetura simplificada de comunicação



Fonte: Elaborada pelo autor.

Na fase de processamento, foi utilizado o Perseo para o processamento dos eventos completos e foi estabelecido o *QuantumLeap* para armazenar as amostras das entidades monitoradas. Na fase da previsão, para calcular a previsibilidade foi escolhido o algoritmo *Prophet* para mostrar uma ideia de como os dados se comportam com determinada entrada.

Neste processo de integração, o *QuantumLeap* foi projetado para realizar o armazenamento dos dados históricos emitidos pelas entidades do *Orion* devidamente conectados pela requisição de conexão como abordado na seção 3.4. Foi feita uma comparação do *QuantumLeap* com o *STH*, mas foi decidido usar o *QuantumLeap* pois os dois tem a mesma função na integração com o *Orion*.

Nesta fase de construção dos dados, foi utilizado o *QuantumLeap* para armazenar o conjunto de dados necessários para construir o conjunto de dados para usar no algoritmo *Prophet* e plotar os gráficos. Para cada entidade, foi definido que o conjunto de dados conteria a data e o valor fundamentadas nas amostras capturadas que precisam ser calculados na previsão.

## 5.2 Treinamento e avaliação da previsão com o Prophet

Primeiro vamos importar as bibliotecas pandas e prophet.

```
1 import pandas as pd
2 from prophet import Prophet
```

Aqui vamos importar o CSV dos dados capturados do ar condicionado do Campus de Quixadá.

```
1 df = pd.read_csv( https://raw.githubusercontent.com/
    iarlenaquiles/smartcampus/main/data-sensor-corrente.csv
    )
2 df.head()
```

No Quadro 2 mostra alguns dados importados do CSV com a data e o consumo do dia do sensor de corrente. Assim, pode-se ter uma ideia do consumo do dia de uma entidade que esteja configurada no *QuantumLeap*. A coluna ds representa a data e a coluna y representa o consumo coletado em amperes(A).

O método *predict* atribuirá a cada linha do *future* com um valor previsto que é nomeado como *yhat*. Quando for passado em datas históricas, ele fornecerá um ajuste na amostra.

Quadro 2 – Dados carregados

DS (Data)	Y (Consumo (A))
04-07-2022	20
05-07-2022	18.4
06-07-2022	15
07-07-2022	17
08-07-2022	20

Fonte: elaborado pelo autor.

O objeto *forecast* é um novo *dataframe* que inclui a coluna *yhat* com a previsão, assim como as colunas para componentes e intervalos de incerteza como mostra a Figura 15.

```

1 forecast = m.predict(future)
2 forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()

```

Figura 15 – Visão dos dados.

	<b>ds</b>	<b>yhat</b>	<b>yhat_lower</b>	<b>yhat_upper</b>
<b>547</b>	2023-07-13	4.087246	1.229102	7.107881
<b>548</b>	2023-07-14	4.054609	1.322172	7.236758
<b>549</b>	2023-07-15	4.088172	1.233599	7.173491
<b>550</b>	2023-07-16	3.888788	0.926713	6.905672
<b>551</b>	2023-07-17	3.976433	1.225980	7.146474

Fonte: adaptado pelo autor.

### 5.3 Implantação da solução

Nesta fase do projeto, foi implantado o serviço de processamento de eventos complexos proporcionados pelo GE *QuantumLeap* do *middleware FIWARE*. Deste modo, as amostras são recebidas do *Mosquitto* e processadas pelo *middleware* e enviada a notificação com o metadados para o *QuantumLeap* onde é armazenado no banco de dados. Neste trabalho, foram utilizados os contêineres *Docker* para otimizar a instalação e configuração do ambiente.

### 5.3.1 Configurando uma notificação

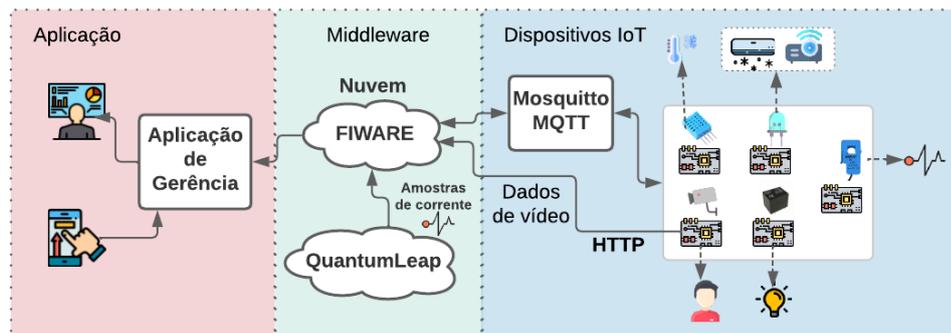
O código abaixo é um exemplo de como é realizado uma *subscription* de uma entidade do *Orion* com o *QuantumLeap*.

```
1 curl -X POST http://orion:1026/v2/subscriptions \
2 -H Accept: application/json -H Content-Type: application
   /json \
3 -d {
4 "description": "A subscription to get info about current
   sensor",
5 "subject": {
6   "entities": [
7     {
8       "id": "urn:ngsi-ld:CurrentSensor:001000001",
9       "type": "CurrentSensor"
10    }
11  ],
12  "condition": {
13    "attrs": [
14      current
15    ]
16  }
17 },
18 "notification": {
19   "http": {
20     "url": "http://quantumleap:8668/v2/notify"
21   },
22   "attrs": [
23     ]
24 },
25 "expires": "2040-01-01T14:00:00.00Z"
26 }
```

## 5.4 Aplicação de Monitoramento

Para atender aos requisitos definidos para a solução, foi desenvolvida uma aplicação de gestão focada em fornecer visualização e controle de dados de consumo de energia de dispositivos IoT. A aplicação já está em execução no Campus Quixadá, utilizando os sensores e atuadores na sala de prototipagem. É dividido em dois módulos principais: *front-end* com painel Servidores reativos (web e móveis) e back-end que lidam com solicitações e interação com *FIWARE*. Foi adicionado o *QuantumLeap* para o armazenamento dos dados históricos. A Figura 16 mostra as três camadas que formam a aplicação, na camada da aplicação temos a aplicação de gerência, na camada do *middleware* temos o *FIWARE* em execução com o *QuantumLeap* e na camada de dispositivos IoT temos os sensores e atuadores do projeto.

Figura 16 – Visão geral da arquitetura da solução



Fonte: adaptado pelo autor.

### 5.4.1 Front-end

A aplicação *front-end* consiste em um *dashboard*, implementado utilizando o *framework* VueJS, que possui monitoramento, controle e análise de consumo de energia, controle de condicionadores de ar, lâmpadas e projetores e verificação do uso da máscara. Os dados de consumo de energia podem estar presente no nível da sala, bloco e campus universitário. A Figura 17a apresenta uma visão do campus com os principais indicadores são exibidos separadamente: consumo de blocos, número de dispositivos ativos e número de dispositivos ativos Há salas de eventos em cada bloco. A Figura 17b mostra o controle de uma sala onde você pode realizar a operação de ligar/desligar do dispositivo e alterar a temperatura do ar condicionado. O painel se comunica com o back-end por meio de serviços *RESTful* e *Web Socket* (WS) de acordo com (AMURIM *et al.*, 2021). Ao *front-end* foram adicionados gráficos dos sensores e foi

ajustado o envio do comando para desligar e ligar os sensores a partir do *Dashboard* que não estava chamando o *sendCommand* corretamente.

#### 5.4.2 *Back-end*

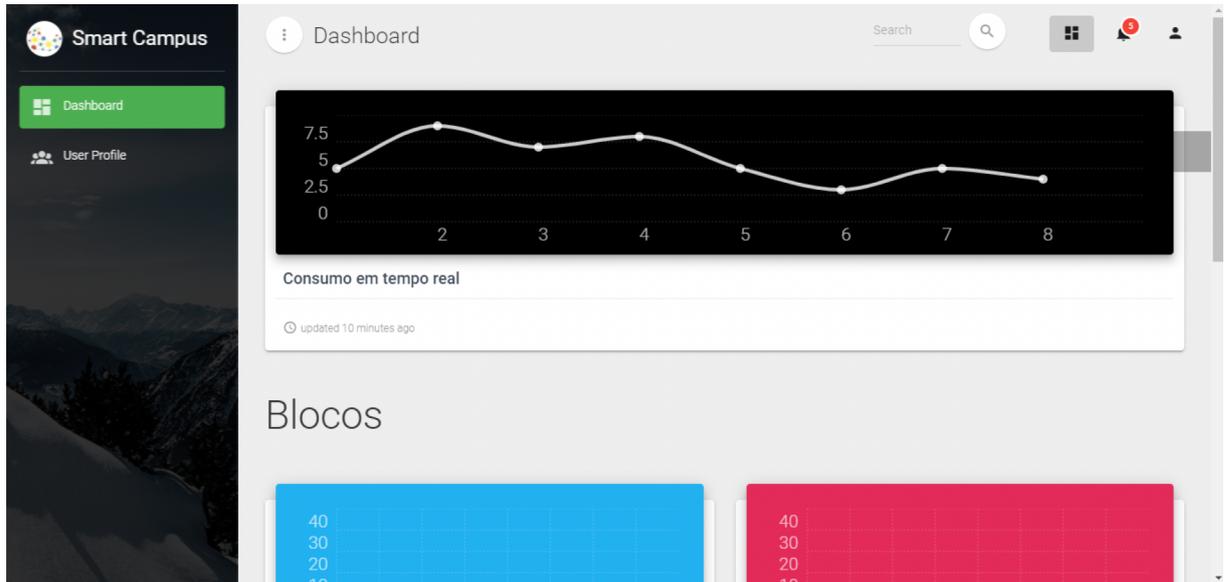
O servidor *Back-end* foi implementado usando *NodeJS* e o *framework Express*. A comunicação com os demais componentes da arquitetura, como ilustrado na Figura 17a, utiliza o protocolo *HTTP* e o *WS* que é uma tecnologia avançada que permite abrir uma sessão de comunicação interativa bidirecional entre o navegador do usuário e um servidor. Os serviços providos pelo *Back-end*, através do protocolo *WS*, são notificações das atualizações em tempo real do consumo de energia, temperatura e umidade. Por outro lado, via *HTTP*, são providos serviços *RESTful* para consulta ao consumo energético de um ou mais blocos e consumo de uma ou mais salas, além do envio de comandos para ligar e desligar os aparelhos de ar-condicionado, lâmpadas e projetores. Ainda via *HTTP*, são providos serviços para mudar a temperatura dos ar-condicionados segundo (AMURIM *et al.*, 2021). Ao *back-end* foi adicionado a integração nas notificações para a inscrição das entidades no *QuantumLeap*.

#### 5.4.3 *Armazenamento Histórico*

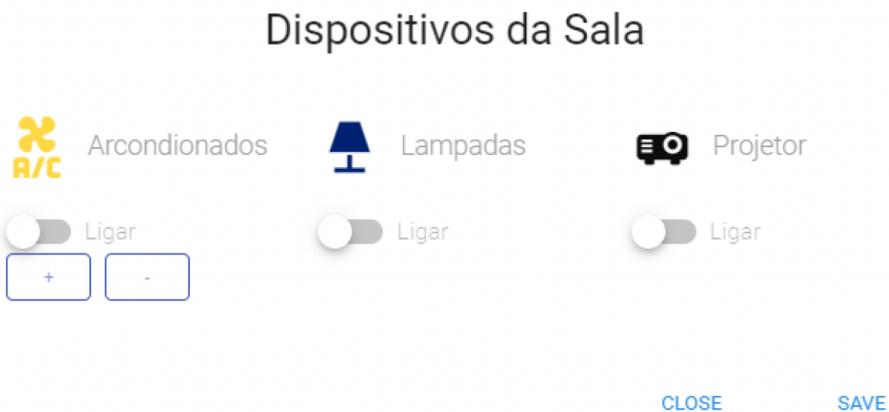
É preciso conectar o *Orion Context Broker* ao *QuantumLeap* por meio de uma assinatura *NGSIv2* para cada tipo de entidade cujos dados históricos deseja monitorar. Os dados históricos para cada tipo de entidade serão adicionados ao banco de dados do *QuantumLeap* desde que a assinatura esteja ativa, configurada corretamente e a entidade na notificação seja compatível com *NGSI*. Caso a assinatura seja removida ou seu status seja alterado, nenhum dado será adicionado após isso. Embora os dados anteriores armazenados no banco de dados do *QuantumLeap* persistam até que sejam removidos externamente usando *APIs* para excluir dados armazenados no *QuantumLeap*. Dessa forma, o fluxo tem a seguinte sequencia:

1. Criar uma assinatura *Orion* para cada tipo de entidade.
2. Inserir/atualizar suas entidades no *Orion*.
3. Os dados históricos serão mantidos no banco de dados da *QuantumLeap*.

Figura 17 – Exemplo da tela da aplicação  
(a) Dashboard.



(b) Modal de gerenciamento.



Fonte: (AMURIM *et al.*, 2021)

#### 5.4.3.1 Assinatura Orion

Como abordado anteriormente, o vínculo entre *Orion* e *QuantumLeap* é estabelecido por meio de uma assinatura que precisa ser criada. Na Figura 18 um exemplo da requisição da assinatura que precisa criar no Orion para estabelecer o link *Orion-QuantumLeap*. A Figura 18 mostra a criação da notificação do sensor de temperatura.

#### 5.4.3.2 Inserção de dados

A Figura 19a tem um exemplo de uma carga útil de inserção para *Orion* que irá gerar uma notificação para *QuantumLeap* com base no exemplo "*Test subscription*" mostrado

Figura 18 – Visão de uma requisição da notificação entre Orion-QuantumLeap.

```
{
  "description": "Test subscription",
  "subject": {
    "entities": [
      {
        "idPattern": ".*",
        "type": "Room"
      }
    ],
    "condition": {
      "attrs": ["temperature"]
    }
  },
  "notification": {
    "http": {
      "url": "http://quantumleap:8668/v2/notify"
    },
    "attrs": ["temperature"],
    "metadata": ["dateCreated", "dateModified"],
    "throttling": 5
  }
}
```

Fonte: QuantumLeap

anteriormente. A Figura 19b mostra uma requisição de criação de uma notificação.

Figura 19 – Exemplo da criação da notificação

(a) Requisição via Orion.

```
{
  "id": "Room1",
  "type": "Room",
  "temperature": {
    "value": 24.2,
    "type": "Number",
    "metadata": {}
  },
  "pressure": {
    "value": 720,
    "type": "Number",
    "metadata": {}
  },
  "colour": {
    "value": "white",
    "type": "Text",
    "metadata": {}
  }
}
```

Fonte: Elaborado pelo autor.

(b) Requisição via QuantumLeap.

```
curl http://localhost:8668/v2/notify -s -S \
-H 'Content-Type: application/json' -d \
{
  "subscriptionId": "5c3edbb331dfg9h71aad5deaaa",
  "data": [
    {
      "id": "Room1",
      "temperature": {
        "value": "10",
        "type": "Number"
      },
      "pressure": {
        "value": "12",
        "type": "Number"
      },
      "type": "Room"
    }
  ]
}
```

#### 5.4.3.3 Transmissão de dados

O *QuantumLeap* usa esquemas de banco de dados para armazenar dados de maneira simples. Essa decisão de *design*, embora não seja eficiente em termos de espaço, uma vez que

muitas vezes muitos valores não mudam entre inserções sequenciais e impondo atributo para ter um tipo consistente de horas extras, aumenta a velocidade de recuperação de entidades completas, eliminando a necessidade de junções que seriam solicitadas de outra forma.

Isso significa que se um atributo de entidade foi recebido na origem como um *Number*, após a inserção, a alteração do mesmo atributo para *Text* falharia. Para mitigar essas falhas, o *QuantumLeap* tenta a conversão de dados, se a conversão não for possível, os valores são substituídos por *None*, para garantir que a inserção no banco de dados não falhe totalmente para a entidade recebida.

#### 5.4.3.4 Índice de tempo

Um elemento fundamental no banco de dados de séries temporais é o índice de tempo. O *QuantumLeap* persistirá o índice de tempo para cada notificação em uma coluna especial chamada *time-index*.

#### 5.4.3.5 Restrições e limitações

O *QuantumLeap* não pode ter dois tipos de entidade com o mesmo nome, mas com letras maiúsculas diferentes. Ex.: *Preprocessor* e *preProcessor*. O mesmo se aplica aos nomes de atributos de uma determinada entidade. Ou seja, os atributos *hotSpot* e *hotspot* serão tratados da mesma forma. Estes são casos raros, mas vale a pena ter isso em mente. Em última análise, a nomenclatura correta de tipos e atributos deve respeitar as diretrizes de nomenclatura.

#### 5.4.3.6 Recuperação de dados

O *QuantumLeap* criará uma tabela para cada tipo de entidade. Os nomes das tabelas são formados com um prefixo (et) mais a versão em minúsculas do tipo de entidade, ou seja, se seu tipo de entidade for *AirQualityObserved*, o nome da tabela correspondente será *etairqualityobserved*. Os nomes das tabelas também devem ser prefixados com o esquema em que são definidos.

## 6 AVALIAÇÃO E RESULTADOS

Depois que todos os passos para a construção, integração e captura dos dados deste projeto foram finalizadas, foram realizados testes para identificação da previsão em um cenário real. Para isso, foi selecionado o uso dos sensores de corrente da sala de teste do Campus de Quixadá, onde o dispositivo foi instalado. A infraestrutura foi executada em uma máquina *Ubuntu 16.04, 64 bits, 2.7 GHz, 6 GB RAM, 60 GB HD*. Dito isso, o *QuantumLeap* armazenou dados do sensor de corrente desde Março à Julho de 2022 e foram feitas previsões de 60, 90, 120 e 365 dias para comparação dos gráficos.

### 6.1 Resultados preliminares

Para testar alguns resultados, foi utilizado alguns períodos para analisar os gráficos. Como mostra o código abaixo, foi utilizado um período de 30 dias para plotar alguns gráficos. Na Figura ?? mostra as datas finais do teste e a Figura 20a mostra as médias e mínimos e máximos dos dados.

```
1 future = m.make_future_dataframe(periods=30)
2 future.tail()
```

Figura 20 – Exemplo utilizando um período de 30 dias

(a) médias.

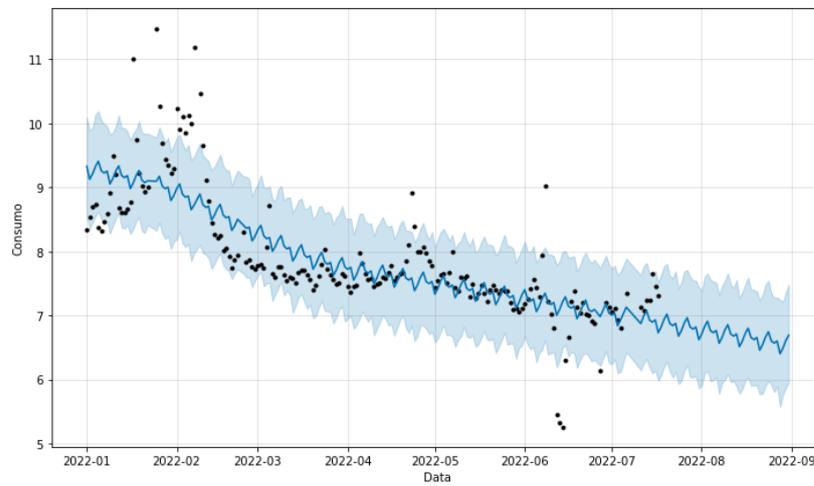
	ds	yhat	yhat_lower	yhat_upper
<b>212</b>	2022-08-12	6.678527	5.918311	7.452196
<b>213</b>	2022-08-13	6.712090	5.931621	7.491641
<b>214</b>	2022-08-14	6.512706	5.744518	7.258566
<b>215</b>	2022-08-15	6.600350	5.910242	7.377642
<b>216</b>	2022-08-16	6.719245	5.957869	7.449868

Fonte: Elaborado pelo autor.

A Figura 21 mostra a sazonalidade semanal associada ao armazenamento dos dados da corrente, bem como a tendência. O uso vai diminuindo ao chegar mais próximo do período de férias do Campus.

Figura 21 – Exemplo utilizando um período de 30 dias

(a) O eixo y mostra o valor da captura e o eixo x mostra a data da captura.



(b) médias.



Fonte: Elaborado pelo autor.

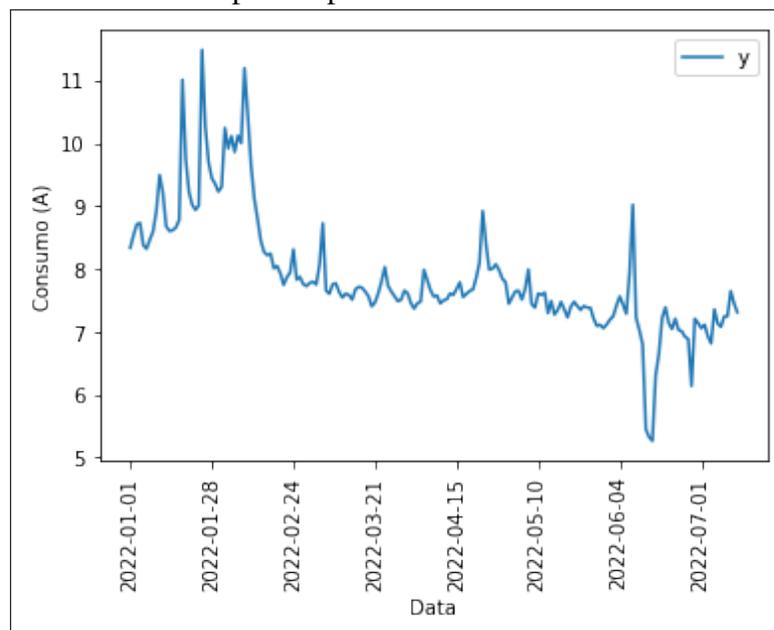
## 7 RESULTADOS

Neste Capítulo são detalhados os resultados provenientes dos testes e experimentos desenvolvidos neste trabalho. Na Seção 7.1, é mostrado o gráfico com os dados capturados no período. Na Seção 7.2 explana sobre o conjunto de dados da solução. Na Seção 7.3 são mostrados 3 gráficos que mostram na Figura 24 os dados da previsão, na Figura ?? mostra um gráfico de tendência e na Figura ?? mostra um gráfico com a tendência semanal. Na Seção 7.4 mostra a eficácia da solução que foi usada para a previsão com os dados coletados. Semelhantemente, na Seção 7.4.1 mostra a avaliação da solução proposta.

### 7.1 Armazenamento dos dados

Após a captura dos dados gerados pelos sensores, foram realizados alguns testes com diversos períodos para certificar a eficácia do algoritmo *Prophet* com os dados armazenados. Com isso, pode-se refazer de modo a observar o detalhamento com um período menor. Dado que uma boa previsão rica em detalhes está associada ao número de amostras capturadas no período. A Figura 22 mostra o gráfico com os dados que foram capturados. Neste gráfico foi utilizado o *matplotlib* para plotar este gráfico.

Figura 22 – Gráfico dos dados capturados com o eixo y mostrando os valores e o eixo x mostrando as datas separado por mês.



Fonte: o autor.

Os dados utilizados neste trabalho, como mostra a Figura 22, foram obtidos do banco

de dados do *QuantumLeap* configurado na máquina virtual do projeto *Smart Campus* do Campus de Quixadá. O intervalo dos dados utilizados está entre 09 de março até 16 de julho.

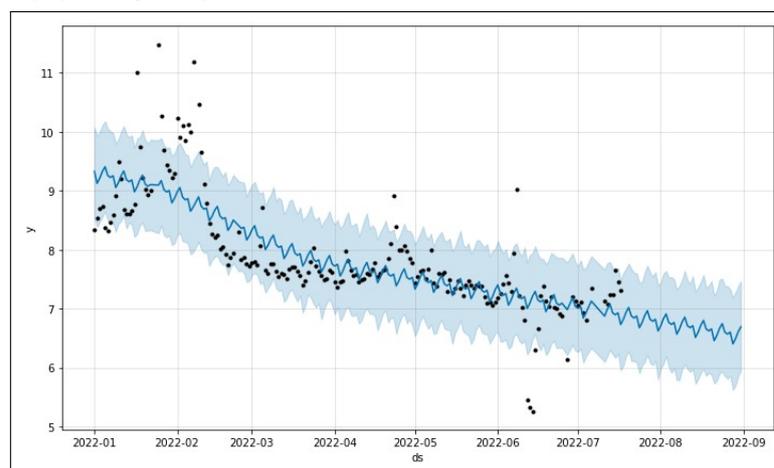
## 7.2 Conjunto de dados

Como dito, o conjunto de dados foi construído a partir dos dados que foram capturados pelas notificações do *Orion* para o *QuantumLeap* que veio do dispositivo IoT. Desta forma, foi gerado um arquivo no formato *Comma-separated values* (CSV) para armazenar o conjunto de dados. A Figura 22 mostra os dados do arquivo CSV resultante. Logo podem ser visualizados os valores gerados pelos sensores de corrente. No Capítulo 5 foi mostrado um exemplo da previsão de 30 dias à frente, no resultado foi selecionado um período de 45 dias.

## 7.3 Resultados do Experimento com período de 45 dias

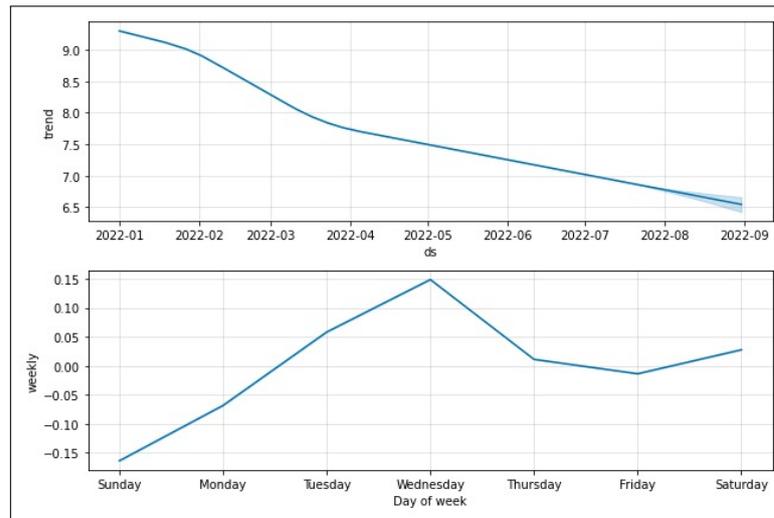
Na Figura ??, podem ser visualizados os dados recriados a partir das amostras capturadas com um período de 45 dias. Na Figura 24 mostra os pontos de distribuição dos dados armazenados, como as últimas datas dos dados armazenados são de julho, ele mostra os pontos até esse mês e soma 45 dias para mostrar o resultado da previsão, por isso que não aparecem os pontos nessa parte. Na Figura ?? podem ser visualizados a tendência, a sazonalidade anual e a sazonalidade semanal da série temporal.

Figura 23 – Previsão de 45 dias.



Fonte: o autor.

Figura 24 – Tendência de 45 dias.



Fonte: o autor.

## 7.4 Eficácia da solução

Como dito no Capítulo 6 foram realizadas uma séries de testes utilizando diferentes períodos. O *Prophet* é otimizado para as tarefas de previsão de negócios que normalmente possuem uma das seguintes características:

1. observações horárias, diárias ou semanais com pelo menos alguns meses (de preferência um ano) de história.
2. fortes múltiplas sazonalidades de “escala humana”: dia da semana e época do ano.
3. feriados importantes que ocorrem em intervalos irregulares que são conhecidos com antecedência (por exemplo, o *Super Bowl*).
4. um número razoável de observações ausentes ou grandes valores discrepantes.
5. alterações de tendências históricas, por exemplo, devido a lançamentos de produtos ou alterações de registro.
6. tendências que são curvas de crescimento não lineares, onde uma tendência atinge um limite natural ou satura.

### 7.4.1 Avaliação da Solução Final

Na avaliação final, os resultados foram comparados com os períodos de 30, 45, 90 e 180 dias, que mostra os gráficos para cada período. Como os dados são de 6 meses, o gráfico em todos os testes sempre está caindo. Isso mostra que a solução proposta precisa de armazenar mais dados para ter um aproveitamento e uma previsão mais acertiva.

## 8 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou a plataforma *web* que serve para o acompanhamento do uso de energia elétrica de seus usuário. Além disso, apresentou o desenvolvimento e integração do *FIWARE* com a *GE QuantumLeap* com um *Web Service* que tem como objetivo criar um centralizador de conteúdo entre diferentes consumidores dos dados.

Disto isso, este trabalho aborda fundamentos de *Smart Campus* e sua aplicação real no Campus de Quixadá, compreendendo a interligação entre sensores e atuadores e módulos de comunicação sem fio para a construção da aplicação e coleta de amostra dos sensores. Além disso, explanou conhecimentos de computação em nuvem, *API Rest* no desenvolvimento de uma aplicação sendo executados na máquinas virtuais utilizando *GEs* do *FIWARE*. Ademais, o uso do *Prophet* para realizar as previsões dos dados históricos. Desta forma, todas as partes do projeto foram concluídas no decorrer do projeto, proporcionando uma solução com embasamento bibliográficos.

Os resultados apresentados no Capítulo 7 mostram que o desenvolvimento e estudo da solução teve seus objetivos alcançados com êxito, assim, este trabalho possibilitou ter uma visão ampla do que pode-se explorar e procurar locais dentro da instituição onde pode ser implantado e monitorado. Além do mais, este trabalho contribuiu com o incentivo ao desenvolvimento de trabalhos sustentáveis, com foco em monitoramento de ambientes inteligentes.

As principais dificuldades encontradas estão relacionadas à rede do Campus, ferramentas com pouca documentação e a literatura com pouco acervo. Contudo, a solução desenvolvida contribuiu para melhorar o acervo e com exemplos de usos das ferramentas, assim, agregando os conhecimentos das tecnologias empregadas.

Dado o estágio atual de desenvolvimento, ainda há vários pontos a serem melhorados no projeto. A aplicação é passível de adição de mais funcionalidades que melhorem a experiência do usuário. Uma dessas funcionalidades, é tornar a aplicação mais colaborativa. Os usuários poderiam compartilhar as configurações de ambientes. Já no sistema de controle, devem ser realizados testes de integração com mais equipamentos e mais salas para capturar mais dados para criar uma previsibilidade mais fiel.

Um trabalho futuro importante seria a implementação de um sistema para controle do ambiente. Esse sistema seria independente, com isso, seria responsável por modificar o ambiente de acordo com parâmetros repassados a ele. Esses parâmetros, em formato *JSON*, ajudariam na configuração e controle de aparelhos dispostos no local, como ar-condicionado, projetores e

lâmpadas. Outro trabalho importante é usar essa grande quantidade de dados para fazer previsões de consumo, nisso, os gestores teriam a ciência da previsão do mês seguinte e tomariam alguma decisão acerca do consumo atual.

## REFERÊNCIAS

- ALVAREZ-CAMPANA, M.; LÓPEZ, G.; VÁZQUEZ, E.; VILLAGRÁ, V. A.; BERROCAL, J. **Smart CEI moncloa: An iot-based platform for people flow and environmental monitoring on a Smart University Campus**. *Sensors*, MDPI, v. 17, n. 12, p. 2856, 2017.
- AMARAL, H. L. M. d. **Desenvolvimento de uma nova metodologia para previsão do consumo de energia elétrica de curto prazo utilizando redes neurais artificiais e decomposição de séries temporais**. Tese (Doutorado) — Universidade de São Paulo, 2019.
- AMURIM, A. D.; SILVA, J. I. da; ORTIZ, M. D.; REGO, P. A.; SOUZA, J. N. de. **Uma solução de IoT baseada no FIWARE para gerenciamento de recursos energéticos e serviços acadêmicos em um campus universitário**. In: WORKSHOP DE COMPUTAÇÃO URBANA, 5. *Anais [...]*. [S.l.], 2021. p. 265–278.
- BAUER, M.; BUI, N.; LOOF, J. D.; MAGERKURTH, C.; NETTSTRÄTER, A.; STEFA, J.; WALEWSKI, J. W. Iot reference model. In: **Enabling Things to Talk**. [S.l.]: Springer, Berlin, Heidelberg, 2013. p. 113–162.
- BROCKWELL, P. J.; DAVIS, R. A. **Introduction to time series and forecasting**. [S.l.]: Springer, 2016.
- CHATFIELD, C. **Time-series forecasting**. [S.l.]: Chapman and Hall/CRC, 2000.
- DONG, Z. Y.; ZHANG, Y.; YIP, C.; SWIFT, S.; BESWICK, K. Smart campus: definition, framework, technologies, and services. *IET Smart Cities*, Wiley Online Library, v. 2, n. 1, p. 43–54, 2020.
- FAZION FILHO, M. **Internet das Coisas**. [S.l.]: Unisul Virtual, 2016.
- MADAKAM, S.; LAKE, V.; LAKE, V.; LAKE, V. *et al.* Internet of things (iot): A literature review. *Journal of Computer and Communications*, Scientific Research Publishing, v. 3, n. 05, p. 164, 2015.
- 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*, MDPI, v. 13, n. 6, p. 3189, 2021.
- MENEZES, J. de. **A vez das Smart Campus: universidades como Laboratórios Vivos**. [S.l.: s.n], 2019. Disponível em: <https://mercadosdofuturo.com.br/a-vez-das-smart-campus-universidades-como-laboratorios-vivos+312492>. Acesso em: 01 jul. 2022.
- PEDROSA, P. H.; NOGUEIRA, T. **Computação em nuvem**, [S.l.], v. 6, 2011. Disponível em: <https://ic.unicamp.br/ducatte/mo401/1s2011/T2/Artigos/G04-095352-120531-t2.pdf>. Acesso em: 01 jul. 2022.
- PIXABAY. **Linha de fomento destina R1.5 bi a projetos de internet das coisas no Brasil**. 2018. Disponível em: <https://epocanegocios.globo.com/Tecnologia/noticia/2018/06/linha-de-fomento-destina-r-15-bi-projetos-para-internet-das-coisas.html>. Acesso em: 09 jul. 2022. Acesso em: 10 jul. 2022.
- QUANTUMLEAP. **Salto Quântico**. 2021. Disponível em: <https://quantumleap.readthedocs.io/en/latest/>. Acesso em: 06 jul. 2022.

ROCHA, F.; DANTAS, L. C.; SANTOS, L. F.; FERREIRA, S.; SOARES, B.; FERNANDES, A.; CAVALCANTE, E.; BATISTA, T. Energy efficiency in smart buildings: An iot-based air conditioning control system. In: SPRINGER. **IFIP International Internet of Things Conference**. [S.l.], 2019. p. 21–35.

SALHOFER, P.; BUCHSBAUM, J.; JANUSCH, M. Building a fiware smart city platform. In: **Proceedings of the 52nd Hawaii International Conference on System Sciences**. [S.l.: s.n.], 2019.

WERK project. **Smart Campus with LoRaWAN**. 2018. Disponível em: <https://projectwerk.gitbook.io/project/assignment/smart-campus-with-lorawan>. Acesso em: 08 jul. 2022.

ZHANG, G. P. Time series forecasting using a hybrid arima and neural network model. **Neurocomputing**, Elsevier, v. 50, p. 159–175, 2003.

## APÊNDICE A - DOCKER COMPOSE

```
1 version: 3.5
2 services:
3   orion:
4     image: fiware/orion:2.0.0
5     hostname: orion
6     container_name: fiware-orion
7     depends_on:
8       - mongo-db
9     networks:
10      - default
11     expose:
12      - 1026
13     ports:
14      - 1026:1026
15     command: -dbhost mongo-db -logLevel DEBUG
16     healthcheck:
17       test: curl --fail -s http://localhost:1026/version
18           || exit 1
19
20   quantumleap:
21     image: orchestracities/quantumleap:0.8.3
22     hostname: quantumleap
23     container_name: fiware-quantumleap
24     ports:
25      - 8668:8668
26     depends_on:
27      - crate-db
28      - redis-db
29     environment:
```

```
30     - CRATE_HOST=crate-db
31     - REDIS_HOST=redis-db
32     - REDIS_PORT=6379
33     - LOGLEVEL=DEBUG
34 healthcheck:
35     test: curl --fail -s http://quantumleap:8668/version
36         || exit 1
37 # Cygnus is configured to persist to STH-Comet
38 cygnus:
39     image: fiware/cygnus-ngsi:2.16.0
40     hostname: cygnus
41     container_name: fiware-cygnus
42     depends_on:
43     - mongo-db
44     networks:
45     - default
46     expose:
47     - "5080"
48     - "5051"
49     ports:
50     - "5051:5051" # localhost:5051
51     - "5080:5080" # localhost:5080
52     environment:
53     - "CYGNUS_MONGO_SERVICE_PORT=5051" # Port that Cygnus
54         listens on for operational reasons
55     - "CYGNUS_MONGO_HOSTS=mongo-db:27017" # Comma
56         separated list of Mongo-DB servers which Cygnus
57         will contact to persist historical context data
58     - "CYGNUS_SERVICE_PORT=5051"
59     - "CYGNUS_API_PORT=5080" # Port that Cygnus listens
60         on for operational reasons
```

```
57     - "CYGNUS_LOG_LEVEL=DEBUG" # The logging level for
      Cygnus
58     healthcheck:
59         test: curl --fail -s http://cygnus:5080/v1/version ||
          exit 1
60
61     # STH-Comet is persisting Short Term History to Mongo-DB
62     sth-comet:
63         image: fiware/sth-comet:2.8.0
64         hostname: sth-comet
65         container_name: fiware-sth-comet
66         depends_on:
67             - mongo-db
68         networks:
69             - default
70         ports:
71             - "8666:8666" # localhost:8666
72         environment:
73             - STH_HOST=0.0.0.0
74             - STH_PORT=8666
75             - DB_PREFIX=sth_
76             - DB_URI=mongo-db:27017
77             - LOGOPS_LEVEL=DEBUG
78         healthcheck:
79             test: curl --fail -s http://localhost:8666/version ||
              exit 1
80
81
82     iot-agent:
83         image: fiware/iotagent-ul:1.7.0
84         hostname: iot-agent
85         container_name: fiware-iot-agent
```

```
86     depends_on:
87         - mongo-db
88     networks:
89         - default
90     expose:
91         - 4041
92     ports:
93         - 4041:4041
94     environment:
95         - IOTA_CB_HOST=orion
96         - IOTA_CB_PORT=1026
97         - IOTA_NORTH_PORT=4041
98         - IOTA_REGISTRY_TYPE=mongodb
99         - IOTA_LOG_LEVEL=DEBUG
100        - IOTA_TIMESTAMP=true
101        - IOTA_CB_NGSI_VERSION=v2
102        - IOTA_AUTOCAST=true
103        - IOTA_MONGO_HOST=mongo-db
104        - IOTA_MONGO_PORT=27017
105        - IOTA_MONGO_DB=iotagentul
106        - IOTA_MQTT_HOST=mosquitto
107        - IOTA_MQTT_PORT=1883
108        - IOTA_DEFAULT_RESOURCE=
109        - IOTA_PROVIDER_URL=http://iot-agent:4041
110     healthcheck:
111         test: curl --fail -s http://iot-agent:4041/iot/about
112             || exit 1
113
114     mongo-db:
115         image: mongo:3.6
116         hostname: mongo-db
```

```
117     container_name: db-mongo
118     expose:
119         - 27017
120     ports:
121         - 27017:27017
122     networks:
123         - default
124     command: --bind_ip_all --smallfiles
125     volumes:
126         - mongo-db:/data
127
128     crate-db:
129         image: crate:4.1.4
130         hostname: crate-db
131         container_name: db-crate
132         ports:
133             - 4200:4200
134             - 4300:4300
135         command: >-
136             crate -Cauth.host_based.enabled=false -Ccluster.name
137                 =democluster
138                 -Chttp.cors.enabled=true -Chttp.cors.allow-origin="*"
139         environment:
140             - CRATE_HEAP_SIZE=2g
141         volumes:
142             - crate-db:/data
143
144     redis-db:
145         image: redis:6
146         hostname: redis-db
147         container_name: db-redis
```

```
148     networks:
149         - default
150     ports:
151         - 6379:6379
152     volumes:
153         - redis-db:/data
154
155
156
157 grafana:
158     image: grafana/grafana:6.1.6
159     container_name: grafana
160     depends_on:
161         - crate-db
162     ports:
163         - 3003:3000
164     environment:
165         - >-
166         GF_INSTALL_PLUGINS=https://github.com/
167             orchestracities/grafana-map-plugin/archive/
168             master.zip;grafana-map-plugin,grafana-clock-
169             panel,grafana-worldmap-panel
170
171     volumes:
172         - grafana:/var/lib/grafana
173
174
175
176 networks:
177     default:
178         ipam:
179             config:
180                 - subnet: 172.18.1.0/24
181
182     volumes:
```

```
177 mongo-db: ~
178 crate-db: ~
179 redis-db: ~
180 grafana: ~
```