



isec

Engenharia

DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA

**Serviços compartilhados de autoconsumo,
flexibilidade, simulação, análise e impacto na rede**

Relatório de Trabalho de Projeto para a obtenção do grau de
Mestre em Engenharia Eletrotécnica

Especialização em Automação e Comunicações em Sistemas de
Energia

Autor

Álvaro Alves Marques Carpinteiro

Orientador

Dra. Cristina Isabel Ferreira Figueiras Faustino Agreira

Co-Orientador

Dr. Carlos Manuel Borralho Machado Ferreira

Supervisor na empresa Laboratório LEMUR

Dr. Pablo Arboleya Arboleya

Coimbra, abril 2021

INSTITUTO POLITÉCNICO DE
COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

AGRADECIMENTOS

Agradecer ao Instituto Superior de Engenharia de Coimbra, Instituição onde foram adquiridas as ferramentas indispensáveis à realização do exercício de engenharia.

A todo o Departamento de Engenharia Eletrotécnica do Instituto Superior de Engenharia de Coimbra.

Ao laboratório *Lemur* da Universidade de Oviedo, local onde foi desenvolvido todo o projeto.

À Professora Doutora Cristina Isabel Ferreira Figueiras Faustino Agreira pela sua orientação, total apoio e dedicação desde o primeiro dia que embarquei neste projeto.

Ao Professor Doutor Carlos Manuel Borralho Machado Ferreira, na qualidade de orientador.

Ao Professor Doutor Pablo Arboleya Arboleya por me ter permitido fazer parte do seu grupo de trabalho, por toda a transmissão de conhecimento e direção no projeto.

Agradecer aos amigos pela paciência e compreensão que tiveram ao longo da redação deste documento.

À família, um especial obrigado. À minha mãe Isabel Alexandra Marques Cardoso Alves por toda a motivação e confiança que sempre depositou em mim. Ao meu pai Fernando Marques Carpinteiro que sempre acreditou nas minhas capacidades e por todas as folgas que abdicou em prol deste projeto.

Finalmente, à Ana Rita Gaião Gomes por todo o auxílio prestado na elaboração do documento e por trazer felicidade ao dia a dia.

A todas estas pessoas, o meu sincero obrigado.

Álvaro Carpinteiro

RESUMO

Atualmente, as sociedades dependem intensivamente de energia elétrica para desempenharem as tarefas do cotidiano, sendo isto válido em ambientes residenciais, bem como ambientes industriais e públicos.

A tendência futura não envolve reduzir esta dependência, mas sim tornar mais eficiente os processos que dela dependem.

Neste relatório, serão abordadas as temáticas relacionados com cidades inteligentes, onde será proposto um modelo de aquisição de dados garantido por uma rede de contadores inteligentes, armazenamento em bases de dados temporais com particular destaque para o *InfluxDB*, e visualização da informação com recurso a duas ferramentas gráficas: o *Grafana* e o *TigerGraph*.

O projeto foi assente numa rede de testes construída e pensada com o objetivo de serem desenvolvidas estratégias e implementadas metodologias que permitam o estudo de diferentes cenários em matérias de cidades inteligentes.

Os resultados obtidos confrontam tecnologias clássicas, comparativamente a uma abordagem mais atual, permitindo aprofundar caminhos diferentes dos tradicionais e assim obter maior conhecimento de soluções a implementar.

Palavras-chave: Contadores Inteligentes, Cidades Inteligentes, Bases de Dados Temporais.

ABSTRACT

Nowadays, societies rely heavily on electric power to perform daily tasks, which is true in residential environments, as well as in industrial and public environments.

The future trend does not involve reducing this dependency, but rather making the processes that depend on it more efficient.

In this project, thematic related to smart cities will be addressed, where a data acquisition model guaranteed by a network of smart meters will be proposed, storage in temporal databases with particular emphasis in InfluxDB and information visualization using two tools: Grafana and TigerGraph.

The project was based on a test network built and designed with the aim of developing strategies and implementing methodologies that allow the study of different scenarios in matters of smart cities.

The results obtained compare classic technologies, relatively to a more current approach, allowing to deepen paths different from the traditional ones and thus obtaining greater knowledge of solutions to be implemented.

Keywords: Smart Meters, Smart Cities, Time Series Databases

ÍNDICE

AGRADECIMENTOS	i
RESUMO	iii
ABSTRACT	v
ÍNDICE.....	vii
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABELAS	xi
SIMBOLOGIA, SIGLAS E ABREVIATURAS	xiii
1 Introdução.....	1
1.1 Considerações gerais.....	1
1.2 Alvo do trabalho de projeto	2
1.3 Objetivos do trabalho de projeto.....	2
1.4 Organização do Relatório	3
2 Estado de Arte	5
2.1 Introdução às Redes de distribuição	5
2.2 Redes inteligentes (<i>Smart Grids</i>).....	5
2.3 Cidades inteligentes (<i>Smart Cities</i>).....	7
2.4 Contadores inteligentes (<i>Smart meters</i>).....	8
2.5 Transmissão de dados	9
2.6 Contadores inteligentes usados neste projeto	10
2.7 Conclusão.....	11
3 Plataforma de monitorização e supervisão de redes.....	13
3.1 Funcionamento geral da rede de testes	13
3.2 Objetivo global	14
3.3 Explicação do modelo implementado.....	16
3.4 Conclusão.....	16
4 Bases de dados	17
4.1 Termo “ <i>Query</i> ”	17
4.2 <i>Time Series</i>	17
4.3 <i>Time Series Databases</i>	17
4.3.1 Definição e contextualização	18
4.3.2 Tempo.....	18

4.3.3	Principais <i>TSDBs</i> que têm vindo a ganhar destaque	19
4.4	<i>InfluxDB</i>	20
4.4.1	Arquitetura e modo de funcionamento	20
4.4.2	Destaques do <i>InfluxDB</i>	21
4.4.3	Inserir informação na BD	22
4.4.4	Extrair informação da BD	24
4.4.5	Análise de <i>benchmark</i> entre uma BD <i>InfluxDB</i> e uma BD <i>SQL MariaDB</i>	27
4.5	Conclusão	35
5	Estudo e análise da rede de distribuição	37
5.1	Visualização dos dados	37
5.1.1	Dependências e instalação	37
5.1.2	Postos de Transformação ativos	38
5.1.3	Valores instantâneos de consumo de um posto de transformação	38
5.1.4	Potência total da rede	42
5.1.5.1	Durante um dia	42
5.1.5.2	Durante uma semana	42
5.1.5.3	Durante um mês	43
5.1.5.4	Durante o último ano	43
5.1.6	Comparação dos consumos nos últimos dois anos	44
5.1.7	Monitorização dos servidores	44
5.2	Plataforma <i>Web de monitorização</i>	46
5.2.1	<i>Tigergraph</i>	46
5.2.2	<i>Apache Kafka</i>	48
5.3	Conclusão	48
6	Conclusões e Propostas de Desenvolvimentos Futuros	49
6.1	Conclusões	49
6.2	Trabalho Futuro	50
6.3	Contributos Empresariais	50
6.4	Papel da inteligência Artificial	50
	Referências	53
	Anexo A Código em <i>Python</i>	55

ÍNDICE DE FIGURAS

Figura 2.1 – Esquema de funcionamento de um contador eletromecânico.	8
Figura 3.1 – Fluxograma Geral do projeto.	15
Figura 4.1 – <i>TSDBs</i> mais usadas no mundo datado em setembro de 2020.	19
Figura 4.2 – Modelo <i>TICK Stack</i> ,.....	20
Figura 4.3 – importar biblioteca <i>InfluxDBClient</i>	22
Figura 4.4 – todos os parâmetros para conexão a uma base dados <i>InfluxDB</i>	22
Figura 4.5 – Código <i>client</i> para conexão ao <i>host</i>	23
Figura 4.6 – Código para a criação de uma BD	23
Figura 4.7 – Código para a criação de uma nova tabela.....	24
Figura 4.8 – <i>Dashboard Chronograf</i>	25
Figura 4.9 – <i>InfluxDB Shell</i>	26
Figura 4.10 – Código para consulta de informação.....	27
Figura 4.11 – Gráfico Tempo em ms da contagem total de <i>rows</i>	28
Figura 4.12 – Gráfico do tempo em ms da contagem de contadores ativos.....	29
Figura 4.13 – Tempo em ms da soma das potências ativas de entrada.	30
Figura 4.14 – Gráfico do tempo em ms dos registos com a potência reativa > 0.2VAR	31
Figura 4.15 – Gráfico do tempo em ms do cálculo médio da potência ativa.	32
Figura 4.16 – Tempo de escrita em milissegundos.....	33
Figura 4.17 – Tempo de escrita em segundos.	33
Figura 4.18 – Comparação temporal de escrita e leitura/escrita em milissegundos.....	33
Figura 4.19 – Espaço ocupado no disco em MB.	34
Figura 4.20 – Recursos de CPU e memória RAM em <i>InfluxDB</i>	35
Figura 4.21 – Recursos de CPU e memória RAM em <i>MySQL</i>	35
Figura 5.1 – Disposição geográfica dos postos de transformação da rede global.	38
Figura 5.2 – Potência ativa em kW por fase ao longo dos últimos 3 anos.....	39
Figura 5.3 – Potência reativa em kVA por fase ao longo dos últimos 3 anos.....	39
Figura 5.4 – Disposição geográfica dos contadores inteligentes ativo.....	40
Figura 5.5 – Tensão em V por fase ao longo dos últimos 3 anos.....	40
Figura 5.6 – Variações em °C da temperatura do PT ao longo dos últimos 3 anos.....	41
Figura 5.7 – Utilização em % do uso do <i>CPU</i> e memória <i>RAM</i> ao longo dos últimos 3 anos.	41

ÍNDICE DE FIGURAS

Figura 5.8 – consumo total de potência ativa da rede global no dia 25 de agosto de 2020.....	42
Figura 5.9 – curva de potência ativa da rede global entre os dias 6 e 13 de setembro de 2020.....	42
Figura 5.10 – curva de potência ativa da rede global entre os dias 13 de agosto e o dia 13 de setembro de 2020.....	43
Figura 5.11 – curva de potência ativa da rede global compreendida entre agosto de 2019 e agosto de 2020.....	43
Figura 5.12 – Comparação de consumos entre o ano 2019 e 2020.....	44
Figura 5.13 – Bytes recebidos por segundo.....	44
Figura 5.14 – Bytes enviados por segundo.....	45
Figura 5.15 – Utilização do CPU.....	45
Figura 5.16 – Utilização da memória RAM.....	46
Figura 5.17 – Cidade exemplo reconstruída digitalmente a partir de dados GIS.....	47

ÍNDICE DE TABELAS

Tabela 3.1 – Características gerais do SM 5CTM.....	11
Tabela 5.1 – Especificações da máquina de testes	27
Tabela 5.2 – Versões do software usado	28

SIMBOLOGIA, SIGLAS E ABREVIATURAS

SG – *Smart Grid*

REI – *Redes Elétricas Inteligentes*

SC – *Smart City*

CB – *Circuit Breaker*

SM – *Smart Meters*

ML - *machine learning*

AI - *Inteligência Artificial*

DSOs – *Distribution System Operators*

RTP – *Real Time Prices*

GSM – *Global System for Mobile Communications*

PLC – *Powerline Communication.*

PT – *Posto de Transformação*

FO – *Fibra Ótica*

GIS – *Geographic Information Systems*

DQLs – *Data Query Language*

IoT – *Internet of Things*

M2M – *Machine to Machine*

BD – *Base de Dados*

DB– *Database*

TSDBs– *Time series Databases*

RFID – *Radio-Frequency Identification*

SQL – *Structured Query Language*

IE – *InfluxEnterprise*

IC – *InfluxCloud*

TSM – *Time-Structured Merge*

AC – *Alternate Current*

CPU – *Core Unit Processor*

RAM – *Random Access Memory*

Ns – *Nanoseconds*

UTC – *Coordinated Universal Time*

OS – *Operating System*

API – *Application Programming Interface*

AC – *Alternate Current*

DC – *Direct Current*

GND – *Ground*

kB – *kilobyte*

kbit – *kilobit*

kbps – *kilobits per second*

Mbit – *Megabit*

Mbps – *Megabits per second*

1 Introdução

Neste capítulo serão apresentadas as considerações gerais do projeto e toda a organização do documento. São propostos os vários objetivos a alcançar e, apresentados os principais tópicos dos vários capítulos.

1.1 Considerações gerais

É visível um aspeto comum a todas as espécies com maiores taxas de adaptação aos ecossistemas. Esse aspeto reside na necessidade de se criarem comunidades para fazer face às exigências do meio. O ser humano não é exceção, na verdade, dependemos instintivamente de mais seres humanos para viver. Assim sendo, tem-se vindo a refinar a forma e os modos como vivemos em comunidade desde o início das primeiras civilizações.

Nos dias de hoje, onde a energia e a tecnologia são as principais forças motrizes das sociedades, torna-se imperativo dominar e desenvolver ferramentas de forma a lidar eficazmente com esta dependência.

A tendência evolutiva levou à criação de cidades cada vez mais autónomas, sob o ponto de vista tecnológico, aumentando o conforto dos seus habitantes e a proliferação da informação. Porém, com a expansão da população agregada por comunidades e, no limite, na globalidade do planeta, começaram a surgir desafios sobretudo a nível de gestão e processamento dos volumes digitais criados por estas massas humanas, agravados pela forte necessidade energética que todo este sistema consome. (Smil., 2019).

É neste seguimento que surge a exigência de serem criados bons sistemas de supervisão e monitoramento das redes energéticas, que congreguem as mais atuais tecnologias, permitindo a interligação das mais diversas fontes de produção energética, com destaque à crescente procura por energias renováveis, garantindo a segurança, eficácia e estabilidade do sistema visando máximas e ideologias que permitam uma contínua expansão futura. (Child *et al.*, 2019).

Na base da informação e, após ser adquirida, existe a necessidade do armazenamento que, uma vez mais, surge como forma de problema, visto serem necessários avultados recursos de *hardware* e investimento financeiro. Tudo isto dependente de bons recursos de visualização, indispensáveis a uma racional e eficiente tomada de decisões.

Assim sendo, existem três peças fundamentais no sucesso das redes do futuro, sendo estas a aquisição de dados, o seu armazenamento e a sua visualização. Estes três aspetos são nutridos por todos os utilizadores, onde estes conseqüentemente dependem da rede, estabelecendo-se uma lógica de dependência circular, em que cada interveniente desempenha um papel importante, onde quanto maior for a qualidade da informação gerada, melhor é o funcionamento da rede.

1.2 Alvo do trabalho de projeto

A investigação realizada e o projeto desenvolvido tiveram como finalidade o estudo de novos métodos de armazenamento, processamento e análise de informação, no âmbito das cidades inteligentes.

Concretamente, decorrente da massiva quantidade de dados gerados por dispositivos divididos ao longo de uma rede real de distribuição elétrica, o presente projeto propõe solucionar a gestão e consecutivo armazenamento da informação. Isto, com o intuito de que, posteriormente, todos estes dados possam ser utilizados de modo a conferir vantagens energéticas, no domínio da eficiência e aproveitamento das redes de distribuição, beneficiando todos os agentes intervenientes neste processo, com destaque para os operadores da rede de distribuição e consumidores finais da energia.

Assim, foram implementadas estratégias robustas de armazenamento de dados em tempo real, que permitiram uma visualização mais clara a nível da globalidade da rede. Estes, conferiram alguma facilidade em antecipar e identificar erros ou maus funcionamentos, produzidos por equipamentos que se foram degradando ao longo do tempo, ou até mesmo erros humanos cometidos no processo de instalação ou reparação de determinados segmentos da rede.

1.3 Objetivos do trabalho de projeto

Em síntese, os principais objetivos do relatório realizado foram os seguintes:

- Testar soluções que contribuam para uma gestão eficiente, segura e dinâmica da energia, permitindo uma maior integração de unidades de produção renováveis. Para este efeito, utilizar como suporte a esta pesquisa, uma rede de transporte projetada e construída com dispositivos inteligentes e os mais atuais protocolos de comunicação, constituindo assim, um suporte bastante robusto na investigação do tema. Esta rede, permite testar soluções e progressivamente, uma vez concluída com sucesso a fase de testes, expandir a metodologia e dispositivos, à rede pública.
- Desenvolver soluções de armazenamento eficazes, decorrentes da quantidade enorme de dispositivos inteligentes interligados, levando à migração de Bases de Dados tradicionais a Bases de Dados com uma arquitetura mais atual e totalmente direcionada ao armazenamento de valores variáveis no tempo provenientes de sensores.
- Analisar, tratar e visualizar os dados obtidos, recorrendo a soluções de *software* atuais e robustas, de forma a poderem ser tomadas decisões assertivas.

- Desenvolver uma plataforma de monitorização e supervisão de redes, com recurso a dados extraídos em tempo real, permitindo controlar segmentos da rede desativados, intervenções de mão de obra, consumos atuais e histórico de consumos passados, bem como cálculo de trânsitos de potência. Pretende-se que a plataforma seja consultada via *web*, permitindo a utilização das suas ferramentas em qualquer local, inclusive em trabalho de campo, agilizando dessa forma o acesso à informação.

1.4 Organização do Relatório

Este projeto está dividido em 7 capítulos, tal como se sintetiza seguidamente:

- O Capítulo 1 contém a introdução ao projeto, a sua contextualização, os objetivos, as metas a atingir e a organização do documento.
- O Capítulo 2 inicia-se com o Estado de Arte relativamente ao projeto global, seguindo-se de uma explicação e exposição aos conceitos de *Smart Grids*, *Smart Cities* e *Internet of Things*. Finalizando com uma explicação detalhada da rede onde todo o projeto é baseado, assim como todos os seus componentes e segmentos.
- O Capítulo 3 introduz os conceitos de *Smart Meters*, o que são, a sua aplicação e o porquê de serem usados, assim como as suas vantagens. São ainda expostos os modelos reais utilizados.
- O Capítulo 4 propõe fazer a ponte entre as várias fases do projeto global. Neste capítulo, é apresentado um fluxograma dos vários tópicos que estão em processo de desenvolvimento, permitindo a contextualização da informação do presente relatório com os objetivos globais propostos.
- O Capítulo 5 propõe expor de uma forma clara a organização das bases de dados onde a informação é armazenada, complementadas por testes comparativos entre as várias tecnologias atuais disponíveis no mercado.
- O Capítulo 6 aborda o tema da visualização da informação, os mecanismos usados para a visualização das curvas de carga e os tempos associados desde o requerimento dos dados até à exposição dos mesmos.
- O Capítulo 7 apresenta as conclusões deste trabalho e algumas sugestões para desenvolvimentos futuros.
- O final deste Relatório inclui as referências bibliográficas e os vários anexos mencionados ao longo dos capítulos anteriores.

2 Estado de Arte

Neste capítulo pretende-se enquadrar os principais conceitos que serão desenvolvidos no relatório, iniciando com uma introdução às redes de distribuição, progredindo para o conceito de redes inteligentes, onde serão abordados com detalhe os conceitos de cidades inteligentes e *Internet of Things*, finalizando com os dispositivos usados nas leituras de potências, utilizados neste projeto.

2.1 Introdução às Redes de distribuição

Ao longo dos últimos anos, a forma e o paradigma dos consumos mudou drasticamente, manifestando diretamente um impacto forte nas redes de distribuição (Nasar *et al.*, 2019). Estas alterações estão em simbiose com todo o progresso tecnológico que, inevitavelmente, tem vindo a emergir ao longo das últimas décadas (Marques *et al.*, 2020), motivado pelas metas de sustentabilidade e eficiência energética estabelecidas mundialmente (European Commission, 2020).

Atualmente, a tradicional lógica unidirecional do consumo introduzida em finais do século XIX (Casazza, 2010) onde a energia era produzida, transportada e consumida de uma forma controlada e conjeturável, começou a ficar obsoleta segundo estudos mundiais, em particular um estudo do MIT desenvolvido em 2011 intitulado “*The Future of the Electric Grid*” (Kassakian, *et al.*, 2011), uma vez que, o número de variáveis dinâmicas das redes é incrementado substancialmente de dia para dia.

Concretamente, o aumento de 3000TWh produzidos no mundo, associado à exponencial instalação de infraestruturas de produção solares e eólicas posteriores ao ano 2000 (Ritchie, 2017); a descida dos preços dos bancos de armazenamento de energia de \$1,100/kWh em 2010 para \$156/kWh em 2019 com projeções de \$100/kWh em 2023 (Henze, 2019); e, os 7.2 milhões de veículos elétricos que circulam atualmente (Gersdorf *et al.*, 2020), acabaram por converter muitos consumidores em consumidor/produtor. Resultando em desafios de gestão aos operadores do sistema de distribuição (*DSOs*), forçando os mesmos, a adaptar as redes às necessidades da atualidade.

Para enfrentar as exigências, é imperativa a criação de redes de teste fundamentadas em modelos urbanos reais, onde operem medidores inteligentes (*smart meters*), sensores e mecanismos de automação. Tudo isto baseado em protocolos de comunicação inteligentes, capazes de supervisionar em tempo real a produção, ajustando-a dinamicamente aos consumos de todos os utilizadores da rede. Todas estas soluções combinadas com recursos de modelos de *machine learning* – (ML) e Inteligência Artificial – (AI) dão origem ao conceito de redes inteligentes (*smart grids*), tema que irá ser fortemente abordado ao longo deste projeto, sendo o mesmo catapultado futuramente para o conceito de cidades inteligentes (*smart cities*).

2.2 Redes inteligentes (*Smart Grids*)

Segundo Bernd Styczynski, o autor do livro “*Fundamentals and Technologies in Electricity Networks*” uma *Smart Grid* “é uma rede de eletricidade que pode integrar de forma inteligente as ações de todos os usuários conectados a ela, isto é, produtores, consumidores e

aqueles que se enquadram em ambos, de modo a criar-se um fornecimento de eletricidade sustentável, económico e seguro de maneira eficiente.” (Styczynski *et al.*, 2014).

Assim sendo, *Smart Grids* – (SG) são redes inteligentes com a capacidade de fazer a monitorização, otimização e balanceamento de cargas, ajustando autonomamente a produção aos consumos de uma forma eficiente.

De forma a ser possível gerir ativamente todos os intervenientes da rede, é necessária a existência de sensores que meçam permanentemente todos os setores. O trabalho conjunto dos mesmos com os *smart meters* – (SM) permitem aos *Distribution System Operators* – (DSOs) terem total conhecimento e controlo dos consumos em tempo real. Por outro lado, se os consumidores tiverem também acesso aos valores de consumos da rede, é possível que os mesmos, ajustem os seus hábitos de acordo com as oscilações dos preços do kWh, assim como evitar horas de picos mais acentuados, não sobrecarregando a rede de uma forma desnecessária.

Posto isto, é notório que de acordo com o tamanho da rede, aumenta a sua complexidade, levantando-se algumas questões do ponto de vista de segurança e comunicação.

É irrefutável o contributo que as SGs propõem ao planeta, sendo essa uma das vantagens mais evidentes, uma vez que as SGs se comprometem com a máxima de diminuir o consumo aumentando a eficiência dos equipamentos e meios de transmissão. Para além disso, as fontes de energia renovável constituem um dos principais focos das SGs. Tudo isto aliado a uma maior fiabilidade e qualidade do fornecimento de energia elétrica.

Como em todos os sistemas ou mais diversos setores, existem sempre aspetos menos ideias. Um dos principais obstáculos à progressão destas redes reside exatamente na quantidade monetária envolvida na implementação destes sistemas. De acordo com os dados da *Directorate-General for Energy* - (DG Ener) da União Europeia, os custos de instalação de um SM variam entre os 200€ a 250€. (European Commission, 2020).

De acordo com a entidade supramencionada, estima-se que a taxa de penetração de SM em redes europeias chegue aos 77% em 2024, traduzindo-se esta percentagem numa totalidade de 223 milhões de contadores. (Alaton *et al.*, 2020).

Ao longo do documento são abordadas as funcionalidades que estes dispositivos apresentam, nomeadamente no subcapítulo 2.4, fundamentando o investimento nos SM. Todavia, é de salientar à priori, que estes contadores possibilitam o registo de leituras energéticas e comunicação das mesmas em intervalos de tempo reduzidos, na ordem de minutos, automaticamente aos operadores das redes de distribuição – (ORD) e, com base nesta informação, é possível ser traçado um perfil de consumos individual muito mais rigoroso, tornando a decisão de escolha do tarifário e ciclo horário muito mais assertiva. Desta forma, os SM propõem ampliar a qualidade dos serviços prestados e minimizar custos, sobretudo com deslocações de técnicos especializados ao local, uma vez que remotamente é possível executarem-se funções como a alteração da potencia contratada ou obtenção das leituras reais dos consumos energéticos.

Por último, é de salientar que um meio de transmissão de informação muito usual é a internet, tornando o investimento na segurança uma prioridade, uma vez que a quantidade de dispositivos com periféricos de acesso a toda a rede é enorme, a taxa de vulnerabilidade aumenta drasticamente.

2.3 Cidades inteligentes (*Smart Cities*)

Exposto o conceito de *Smart Grids* – (*SG*), ampliando-o, podemos aproximar-nos do conceito de *Smart City* – (*SC*). As *SCs* são, na verdade, um conceito mais abrangente de *SGs* uma vez que todos os setores da cidade são projetados com base em partilha e interligação de informação.

As *Smart Cities* têm vindo a ganhar um destaque crescente no mundo, sobretudo devido aos avanços igualmente exponenciais de *hardware*, permitindo processarem-se em tempo real volumes de informação que, no passado, seria impossível ambicionar-se desenvolver mecanismos que tivessem poder computacional de gerir as exigências da atualidade.

Numa cidade inteligente, a tecnologia oferece um contributo ativo nos sistemas de fornecimento energético, mobilidade, e espaços comuns, aumentando o conforto dos habitantes e reduzindo os impactos ambientais. Existem algumas tentativas formalizadas de definição de cidade inteligente, nomeadamente uma cidade: “conectando a infraestrutura física, a infraestrutura de tecnologia da informação, a infraestrutura social e a infraestrutura empresarial para alavancar a inteligência coletiva da cidade” (Harrison, C. *et al.*, 2010), ou segundo outros autores, uma cidade inteligente pode ser definida como: “uma cidade inteligente e sustentável é uma cidade inovadora que usa tecnologias de informação e comunicação ou outros meios para melhorar a qualidade de vida, a eficiência das operações, serviços urbanos e a competitividade, garantindo as necessidades das gerações presentes e futuras no que diz respeito aos aspetos económicos, sociais e ambientais ” (ITU, 2016).

As definições apresentadas foram possíveis devido à introdução do conceito *IoT* (*Internet of Things*), este termo está fortemente relacionado com situações em que a conectividade da rede e a capacidade computacional se expandiram para sensores, objetos e materiais de uso diário que anteriormente não possuíam estas funções, permitindo que dispositivos produzam, transfiram e armazenem informação sem intervenção de recursos humanos (Nasar *et al.*, 2019). O conceito fundamental da *IoT* baseia-se na ocorrência simultânea de uma infinidade de interações, nomeadamente, etiquetas de identificação por radiofrequência (RFID), atuadores, sensores, telemóveis entre outros dispositivos, que têm a capacidade de comunicar entre si visando um objetivo comum.

É neste seguimento que os vários setores das cidades começam a ser construídos, totalmente direcionados para a interligação e partilha de informação levando ao surgimento das cidades inteligentes.

O setor energético manifesta-se de maior interesse para o presente relatório, no entanto todos os outros sofrem direta e indiretamente as alterações deste setor, uma vez que todos dependem fortemente de energia para operarem. É o caso particular do setor dos transportes, tema que ganhou também elevado destaque nos últimos anos. A mobilidade elétrica constitui um fator de enorme importância no tema das *SCs*, tornando-se fundamental planear e criarem-se boas infraestruturas para os transportes públicos bem como particulares a fim de garantir o suporte necessário a todos os cidadãos, transformando gradualmente o paradigma, tornando em particular, o carro elétrico um meio de transporte acessível a todas as camadas da sociedade, não só do ponto de vista económico mas também do ponto de vista de redes de distribuição energética e postos de carregamento.

2.4 Contadores inteligentes (*Smart meters*)

Os medidores inteligentes de energia, tornam-se peças fundamentais na ótica das *SG*, uma vez que as mesmas dependem de valores escalares para conseguirem tomar decisões eficientes. Contudo, muitos são os desafios que precisam de ser superados de modo a alcançar uma rede inteligente totalmente funcional.

Neste contexto, o presente capítulo desenvolve uma descrição detalhada das principais funcionalidades que os *SM* devem fornecer.

Após esta etapa, o leitor deve ser capaz de ter uma visão detalhada destes dispositivos, compreender o papel fundamental que desempenham na forma primária de aquisição de dados e, por conseguinte, a exportação dessa informação para ser em tempo real tratada.

Os primeiros dispositivos mais usuais que permitiam fazer medições energéticas eram eletromecânicos. A corrente elétrica, ao passar pelo medidor, alimenta duas bobinas de indução que produzem um fluxo magnético num condutor metálico circular. Por sua vez, este disco, gira a uma velocidade proporcional ao fluxo de potência e, contadas as revoluções, é possível calcular-se com um elevado grau de confiabilidade a potência total consumida por uma carga. A figura 2.1 pretende ilustrar o modo de funcionamento destes dispositivos. (Weranga *et al.*, 2013).

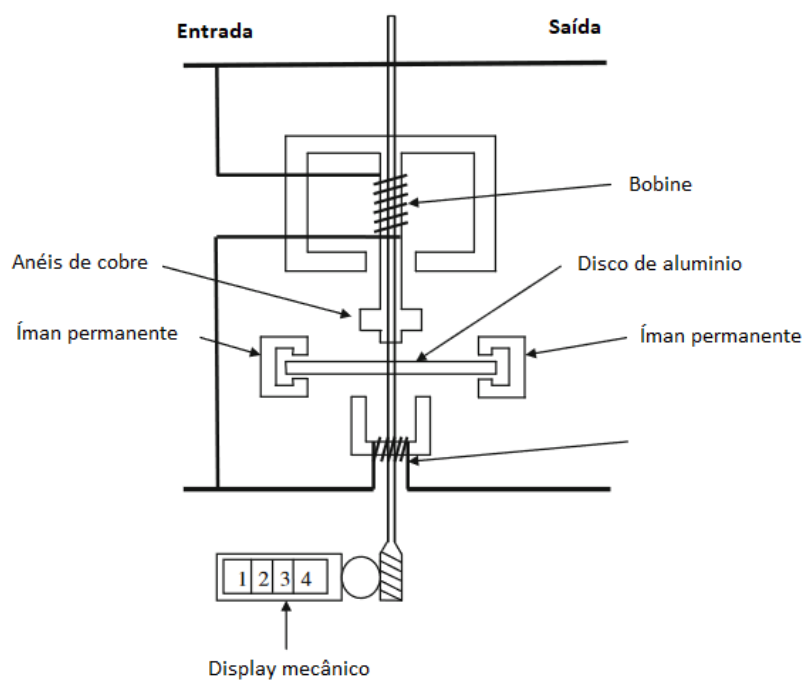


Figura 2.1 – Esquema de funcionamento de um contador eletromecânico.

Porém, as exigências de monitorização e controlo da rede elétrica forçaram esta tecnologia a evoluir de modo a fornecer funções adicionais.

Gradualmente, os contadores eletromecânicos começaram a ser substituídos por contadores eletrónicos, baseados em materiais semicondutores, diminuindo o seu volume e peso.

Com a introdução dos microprocessadores, surgiu um portal de opções, tornando estes dispositivos mais eficientes e autônomos, uma vez que, começou a ser possível enviar os valores das contagens para um endereço remoto, minimizando a intervenção humana radicalmente.

Os primeiros contadores com as funcionalidades supramencionadas, permitiam aos *DSOs* fazer contagens a longas distâncias, porém, o avanço não ficou por aqui e, atualmente, estes equipamentos usualmente chamados de contadores inteligentes, adquiriram esta nomenclatura por apresentarem funções com elevado grau de complexidade como o suporte a comunicação bidirecional, armazenamento e gerenciamento de dados, detecção de fugas de energia, balanceamento de cargas e enviar valores de consumo em intervalos de 1 em 1 minuto.

2.5 Transmissão de dados

Um dos principais desafios que as *SG* enfrentam, não está relacionado com a capacidade dos *SM* enviarem informação em intervalos de tempo curtos, mas sim com toda a rede de transporte da informação ser capaz de dar uma resposta eficaz ao volume de informação que por ela circula.

Atualmente, existem várias tecnologias e plataformas de suporte que permitem fazer o envio dos dados. Seguidamente irão ser abordadas as várias soluções de transmissão de dados.

Existem dois meios que servem de suporte à transmissão de informação, via aérea sem cabos ou via física com cabos. No primeiro caso, existe uma vantagem económica para os *DSOs*, uma vez que não é necessária a criação de infraestrutura para a transmissão ser executada. Porém, as distâncias que esta abordagem sugere são bastante inferiores à segunda solução, assim como o volume de informação transmitido. Optando-se por uma via física, a distância de cobertura aumenta radicalmente, bem como a taxa de bits/s. Em ambos os casos, é fundamental não negligenciar o aspeto das interferências do sinal.

Focando nas tecnologias aéreas, uma das formas de se fazer o envio de dados, é com recurso à tecnologia *Bluetooth*, apresentando a vantagem de ser relativamente barata, porém a distância de transmissão ronda um máximo de 10 metros. A frequência de operação é de 2,45GHz e a largura de banda poderá atingir os 3 Mbps. (Zhao *et al.*, 2017).

Seguidamente, outra tecnologia de baixo consumo e baixo custo é o *ZigBee*. Apresenta 3 frequências de operação, 868 MHz, 915 MHz e 2,4 GHz, com larguras de banda encontradas no intervalo de 20 a 250 Kbps a uma distância de 100m. (Burunkaya *et al.*, 2017).

Outra solução poderá ser o *Wi-Fi*, que permite alcances de 250m, opera nas bandas de 2,4GHz ou 5GHz com taxas de emissão até 600Mbps. Contudo, de modo a garantir valores tão elevados, o consumo energético na amplificação e transmissão do sinal é elevado. (Hlaing *et al.*, 2017).

Por último, sendo a tecnologia sem fios mais usual atualmente, encontram-se as redes *GSM*. Estas redes usam predominantemente as frequências 3G, 3.5G e 4G/LTE, permitindo cobrir grandes áreas urbanas a uma taxa de transmissão superior a 100 Mbps. (Nithin *et al.*, 2012).

Dando agora destaque às redes físicas, existem duas formas bastante viáveis de se transmitir dados. Uma delas é a tecnologia *Powerline Communication (PLC)*. Esta tecnologia opera baseada em linhas energéticas já existentes, permitindo enviar informação diretamente pela linha de distribuição energética. Neste domínio, as transmissões PLC dividem-se em dois grupos distintos, a *Narrowband PLC* e *Broadband PLC*. Para o primeiro caso, a frequência de operação é relativamente baixa, encontrando-se entre os 3 a 500Khz, permitindo a transmissão de informação ao longo de grandes distâncias, porém com uma velocidade máxima de transmissão de dados de até 100 kbps. Em segundo, é apresentada uma transmissão PLC que opera em frequências mais altas, na janela dos 1.8MHz a 250MHz, elevando a taxa de transmissão para até 2 Mbps, contudo, as distâncias máximas alcançadas são menores. (Majumder *et al.*, 2004).

Finalizando, existem ainda casos em que a quantidade de informação é de tal ordem elevada que é necessário recorrer à Fibra Ótica, solução que apresenta larguras de banda na casa dos *Gbps*, porém o investimento necessário na transmissão, sobretudo no mecanismo de conversão do sinal digital para sinal ótico é em grande parte dos casos muito elevado.

Neste projeto, os dados provenientes dos SM não são enviados diretamente para bases de dados em servidores, inicialmente são pré processados por dispositivos no interior das várias subestações para posterior envio. Esta abordagem consiste na utilização de um concentrador que agrupa os consumos provenientes de vários SM encaminhando seguidamente os dados para armazenamento central e posterior processamento. Estas comunicações são feitas sob a forma de *PLC* a uma gama dinâmica de frequências reservada aos provedores de energia, localizada entre os 3kHz e os 95kHz, garantindo uma elevada integridade dos valores obtidos.

Nesta fase do projeto, totaliza-se para a rede em questão um total de 8087 SM ativos, cujos valores de potência ativa, potência reativa e valor de tensão de cada um destes dispositivos, são permanentemente transmitidos para posterior processamento, com uma periodicidade de hora em hora.

2.6 Contadores inteligentes usados neste projeto

Dada a dimensão da rede, são usados vários modelos de diferentes fabricantes, todavia, a tecnologia em termos de protocolos de transmissão de dados, modos de leitura, armazenamento e demais funções, mantêm uma tendência geral de similaridade, de forma a permitir a estabilidade e o correto funcionamento da rede.

Dos vários fabricantes, destacam-se a *SIEMENS*, *A.E.G*, *LANDIS*, *ZIV*, *SCHLUMBERGER*, *ROMANILLOS*, *CIA. CONTADORES*, *SAGEMCOM*, *CIRCUTOR*, *ACTARIS*.

A Tabela 2.1 apresenta o *datasheet* de um SM ZIV monofásico.

Modelo: 5CTM	Valores por fase
Precisão de energia ativa / reativa	Classe 1 (IEC 62053-21) / Classe 2 (IEC 62053 23)
Constante de teste de verificação	1000 pulsos / kWh ou kvarh

Valor de referência de corrente (corrente máxima)	5 (80) A
Corrente de arranque	20 mA
Potência absorvida pelo circuito de corrente (Iref)	< 0.2 VA
Valores nominais de tensão	127 a 230 VAC
Consumo	< 2W / 13 VA (EN 62053-61)
Faixa de operação especificada	-20°C to +70°C
Valores nominais do disjuntor embutido (número de operações)	80 A / 250 VAC (106)UC2 de acordo com IEC62055-31
Porta ótica	RS485
Serviço PLC	(PRIME 1.4 in CENELEC Banda A)
Nível de proteção	IP51 (De acordo com IEC 60529)
Dimensões	212,17 mm x 145,85 mm x 64,03 mm

Tabela 2.1 – Características gerais do SM 5CTM

2.7 Conclusão

São irrefutáveis as alterações que as redes de distribuição têm sofrido recentemente. Grande parte deste progresso deriva da necessidade insaciável que o ser humano tem pela busca constante de avanço tecnológico e de todas as regalias que dele advêm, aliado à necessidade de mudança motivada pelos fortes impactos ambientais que se têm observado decorrentes de uma acentuada exploração de recursos, para garantir um estilo de vida cada vez mais exigente.

As cidades inteligentes surgem como tentativa de convergir o melhor dos vários interesses e necessidades das massas. É nesta linha de raciocínio que se desenvolve um trabalho de investigação contínuo, em prol da criação de soluções cada vez mais eficientes e avançadas tecnologicamente, que permitam a implementação de infraestruturas direcionadas para o conceito de *Internet of Things*, e assim permitir a criação de *Smart Cities*.

É ainda de salientar que os *Smart meters* são peças essenciais nas redes atuais e futuras, uma vez que, graças aos mesmos, é possível medir com exatidão os consumos e potências injetadas na rede, permitindo aos operadores do sistema de distribuição ter um controlo total e em tempo real de cada utilizador da rede.

3 Plataforma de monitorização e supervisão de redes

Neste capítulo irão ser abordados com detalhe os objetivos globais, permitindo uma visualização completa de todos os setores e várias fases do projeto.

Este capítulo apresenta a explicação da rede de transporte usada para desenvolver testes e pesquisas relacionadas com o tema de que se tem vindo a tratar, não comprometendo a rede global.

Será ainda apresentada através de um fluxograma, toda a estrutura do projeto, seguindo-se uma explicação ponto por ponto dos vários elementos.

3.1 Funcionamento geral da rede de testes

O presente projeto é todo baseado numa rede real com o objetivo claro de estudar o impacto das novas tecnologias abordadas no ponto 2.1 (Koirala, *et al.*, 2020).

A rede de teste é constituída por 30 Postos de Transformação – (PTs) principais; 10 290 barramentos, dos quais 2 681 são monitorizados e um total de 8087 SM.

Existem 3 tipos de transformadores nesta rede, sendo que o rácio de tensões é 22/0.420 kV e três tipos diferentes de potência 100, 250 e 630 kVA.

As linhas de alimentação que emanam das subestações e convergem em Postos de Transformação possuem circuitos de corte.

Nesta rede, encontram-se cargas monofásicas, bifásicas e trifásicas, sendo que as cargas polifásicas podem ser equilibradas ou desequilibradas. Cada transformador tem conectado em média 150 a 250 SM.

A rede de Baixa Tensão – (BT) é projetada como um sistema de 4 cabos com neutro isolado. A tensão do neutro é monitorizada em cada um dos barramentos, sendo o neutro aterrado apenas no PT.

A tensão nominal é de 416 V (fase a fase) e a frequência 50 Hz.

As centrais de produção energética à exceção de microproduções, não irão ser consideradas neste relatório, é assumido, portanto, que a fonte de produção energética é estável e ilimitada.

É de salientar que a informação *GIS* (*Geographic Information Systems*) foi toda extraída localmente e armazenada em base de dados para permitir fazer a reconstituição da mesma via *software*.

Acrescenta-se que a rede de testes tem dispositivos inteligentes com capacidade sensorial para fazer medições precisas e periódicas da rede. Todas as cargas são monitorizadas através de *smart meters* individuais, neste caso, com uma resolução de 15 em 15 minutos.

3.2 Objetivo global

Uma vez expostos os conceitos de *Smart grid* e *Smart meter*, aliados à rede de testes descrita no subcapítulo 3.1, as possibilidades de estudo começam a ser praticamente ilimitadas.

É nesta sequência que surge a necessidade de ser criada uma plataforma de monitoramento e supervisão de redes que permitisse gerir em tempo real aspetos como consumos dos utilizadores, taxa de ocupação da rede, identificação de barramentos desligados da rede, transformadores fora de serviço, intervenções de mão de obra, cálculos de trânsitos de potência, sempre com a ambição de expandir funções, criando uma ferramenta cada vez mais rica e completa.

Com o intuito de implementar a vasta gama de exigências, manifestou-se a necessidade de segmentar a implementação desta ferramenta em várias fases.

Numa primeira fase, iniciou-se a criação dos mecanismos para a recriação da rede em *software*. Para isso, foi necessário recolher os dados *GIS* e assim, com recurso a ferramentas de *software* que irão ser abordadas no decorrer deste projeto, nasceu o primeiro modelo da rede.

Após ter sido concluída a topologia da rede, foi possível avançar para uma segunda etapa. Nesta, visto já existir suporte, começaram a ser implementadas e aperfeiçoadas funções. Foram adicionadas funcionalidades de busca para localizar transformadores, adicionados contadores, sistemas de proteção e respetivo estado desses sistemas, bem como, começaram a ser feitos os primeiros ensaios de cálculo de trânsitos de potências.

Atingida a terceira fase, fase atual à data do presente relatório, começaram a surgir os desafios de gestão e armazenamento de toda esta informação, desafios de interligação de sistemas, visualização em tempo aceitável de toda a rede e respetivos dados de consumos.

No sentido de dar resposta às contrariedades, foram analisadas soluções e diferentes abordagens, acabando por ser implementada a lógica representada no fluxograma da Figura 3.1, iniciada pela aquisição dos dados provenientes dos *smart meters* que, uma vez agrupados, seguem para o armazenamento numa base de dados temporal. As etapas seguintes asseguram o processamento e visualização dos dados.

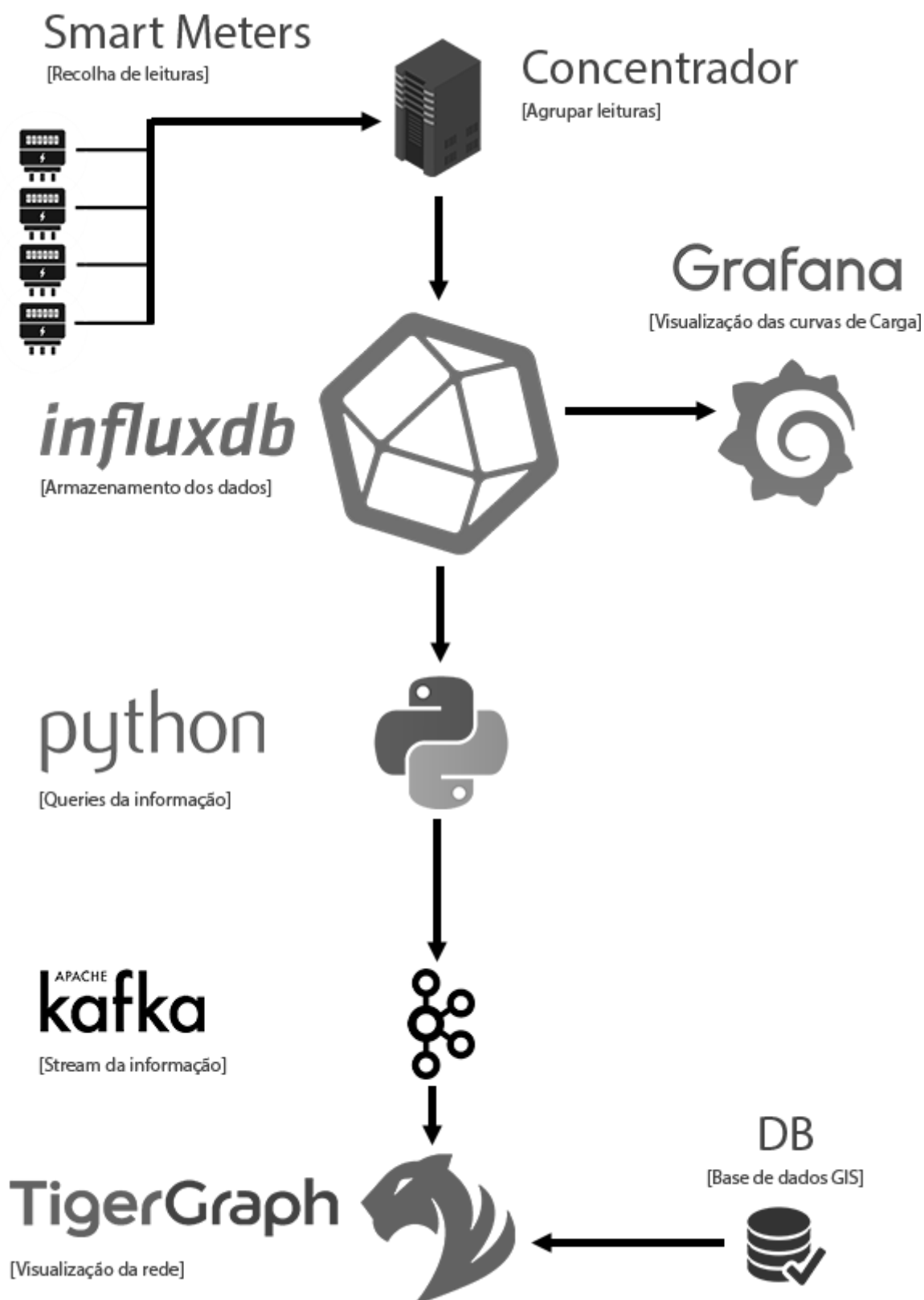


Figura 3.1 – Fluxograma Geral do projeto.

3.3 Explicação do modelo implementado

Baseando a seguinte explicação no fluxograma da Figura 3.1, seguindo um raciocínio descendente, na fase mais primária encontra-se a rede de *smart meters*. São esses dispositivos, como mencionado no capítulo 2, os equipamentos responsáveis pela execução das leituras, que são enviadas e agrupadas para equipamentos, denominados concentradores. Em média cada concentrador está conectado a um total de 150 a 250 SM.

Feita a aquisição dos dados, os mesmos são enviados para um servidor central onde são armazenados. A base de dados escolhida para este fim (*InfluxDB*) baseia-se numa lógica temporal. No capítulo 4 irão ser discutidas as diferentes formas de bases de dados e o porquê de se ter optado por esta tecnologia.

Posto isto, existem dois processos, cada um com caminhos distintos. Uma das opções consiste na visualização das curvas de carga diretamente usando a ferramenta *Grafana*, constituindo esta opção, uma solução prática para consulta de valores de consumos em tempo real da rede. Por outro lado, envergando um caminho mais complexo, segue-se a aquisição da informação recorrendo a linguagem *SQL*. Estas consultas são geridas por código em *Python*, bem como o envio das mesmas para um servidor de *Streaming* em *Apache Kafka*, permitindo assim que as *queries* atinjam a plataforma de visualização final fundamentada em *Tigergraph*.

A fase final em *Tigergraph* contempla duas entradas de dados, sendo que uma delas é externa, proveniente do caminho descrito anteriormente e outra nativa, onde neste último caso, todo o conteúdo *GIS* já está previamente carregado numa base de dados dedicada ao *Tigergraph*.

3.4 Conclusão

Neste capítulo, foi exposta a globalidade do projeto. A implementação total do que foi proposto, carece de tempo e de recursos humanos. Assim sendo, este capítulo pretende abordar todos os pontos do projeto, ainda que, muito do apresentado esteja presentemente concluído.

A diversidade de *softwares*, servidores e proveniência dos dados, dificultam a tomada de decisão nos caminhos e modelos a seguir. Contudo, a abordagem descrita, foi exaustivamente analisada e ambiciona solucionar a maioria dos problemas que foram surgindo e se projeta que venham a aparecer.

Nesta fase do projeto, como descrito anteriormente, já foi ultrapassada a fase de recolha de informação, armazenamento e visualização da mesma. Porém, a introdução de todos os dados na plataforma de monitorização e visualização final ainda constitui um desafio diário. Os esforços atuais residem na criação dessa plataforma, cujo objetivo prende-se por reunir toda a informação num único local, disponível via *web*, a fim de agilizar o trabalho em campo nas redes de distribuição, assim como permitir uma maior vista da rede global, salientando os seus defeitos, avarias, intervenções necessárias ou simplesmente o seu bom funcionamento.

4 Bases de dados

Neste capítulo irá ser dado um enorme destaque às bases de dados temporais. O capítulo será iniciado com uma breve evolução das Bases de Dados – (BD), seguindo-se a definição de *Time Series*, fundamental para entender o funcionamento geral das *Time series Databases (TSDBs)*.

Serão ainda feitas análises de benchmarking e comparações relativamente a outras *TSDBs*.

4.1 Termo “Query”

Ao longo deste documento, serão feitas várias referências ao termo “*query*”. Uma *query* em base de dados é na verdade uma consulta de informação, isto é, um pedido de informação ou um dado em específico à base de dados. A linguagem mais popular utilizada para este fim é a *Structured Query Language (SQL)*.

4.2 Time Series

Antes de se progredir para uma explicação detalhada das bases de dados temporais, existem alguns conceitos que necessitam de ser explicitados previamente, facilitando a compreensão da informação apresentada nos próximos subcapítulos.

De acordo com a definição clássica, uma série temporal é simplesmente uma sequência de números obtidos em intervalos periódicos de tempo. De uma abordagem mais geral, uma série temporal poderá não ser meramente constituída por *floats* ou *integers*, porém, na sua grande maioria, uma série temporal é constituída por um a sequência de valores escalares em tempo discreto.

Assim sendo, destaca-se uma grande relação entre o tempo e a ordem dos acontecimentos, uma vez que a alteração dos eventos, poderia corromper a veracidade das leituras.

Posto isto, é compreensível que as séries temporais existam em qualquer domínio da ciência aplicada e da engenharia que envolva medições numa escala temporal.

4.3 Time Series Databases

Este subcapítulo será iniciado com a definição de *Time Series Databases* seguindo-se em detalhe todos os aspetos relacionados com as mesmas, incluindo a *TSDB* usada no projeto, concluindo com testes precisos de comparação entre duas tecnologias de *BD*.

4.3.1 Definição e contextualização

As bases de dados temporais congregam os conceitos adquiridos no subcapítulo 4.2. Numa análise global, tal como o nome indica, a função principal reside no armazenamento de informação. Os dados são guardados em séries temporais, sendo o tempo, um elemento fundamental na arquitetura das mesmas (Naqvi *et al.*, 2017). São direcionadas sobretudo para o uso direto com valores provenientes de sensores com leituras periódicas.

Numa primeira análise surgiram as bases de dados relacionais. Neste caso, os dados são armazenados sobre a forma de tabelas relacionadas entre si, sendo que cada linha da tabela possui um ID exclusivo, permitindo a sua seleção, identificação e interligação. Cada tabela pode conter várias colunas.

O monopólio das bases de dados tradicionais encontra-se dividido pela *Oracle*, *MySQL* e *Microsoft SQL Server* (DB-Engines, 2020). A linguagem *SQL* é comum a todas e a lógica de armazenamento e aquisição de dados é igualmente idêntica.

Das várias abordagens ao tema no contexto do projeto, um aspeto predominante era exatamente o armazenamento de valores aleatórios provenientes de 8087 dispositivos. Posto isto, e no decorrer da informação adquirida até esta etapa achou-se que fazia pleno sentido, ser implementa uma lógica assente em *TSDBs*.

No subcapítulo 4.4 irá ser apresentada a *TSDB* usada no projeto bem como no subcapítulo 4.4.5, apresentar-se-ão alguns testes de *benchmarking* que irão colocar lado a lado duas bases de dados distintas, fundamentando assim a opção tomada.

É possível implementar algumas funções particulares de *TSDBs* em BD relacionais, todavia a arquitetura de uma e de outra é totalmente distinta, daí existirem diferenças vincadas entre ambas.

No caso das BD convencionais apenas o estado atual dos dados é representado enquanto que nas BD temporais existe uma representação contínua do estado passado, presente e futuro. Por outro lado, para o primeiro tipo de BD o gerenciamento do tempo tem de ser feito por parte da aplicação que o controla, dependência que não é necessária para o caso das BD temporais, uma vez que as mesmas, são responsáveis por essa gestão temporal.

4.3.2 Tempo

É importante perceber que o tempo assume o leme nas *TSDBs*.

As variáveis temporais podem ser distribuídas em *instantes*, *períodos* e *intervalos*. Estas três formas de tempo complementam-se na ordem inversa. O *instante* é um momento aleatório do tempo e está normalmente associado ao registo em BD de um novo valor adquirido num instante de tempo. O *período* compreende o intervalo entre dois instantes de tempo conhecidos. O *intervalo* apresenta semelhanças com o *período* pois ambos estão relacionados com a diferença de tempo entre dois instantes, todavia, neste último caso, é conhecida a diferença temporal dos dois instantes, mas é desconhecido o momento em que o mesmo inicia ou finaliza.

O contexto temporal pode ser representado por diferentes dimensões de tempo, sendo eles tempo de transação, tempo de validade e tempo definido pelo usuário. No primeiro caso, é registado em BD o instante de tempo em que ocorreu uma alteração do estado de um objeto.

No segundo caso, o tempo de validade constitui o tempo em que as amostras foram recolhidas, documentado o passado e o presente dos diferentes estados de um objeto. Finalizando, o tempo de usuário está associado ao gerenciamento feito diretamente pelo gestor da BD ou aplicação onde a BD é incorporada.

4.3.3 Principais TSDBs que têm vindo a ganhar destaque

Com a introdução de um volume enorme de dispositivos que recolhem, processam e enviam informação periodicamente, tornou-se numa necessidade, como tem vindo a ser perceptível ao longo do presente trabalho, de serem criados mecanismos direcionados para o armazenamento da informação gerada por estes dispositivos. Assim, emergiu uma oportunidade pouco explorada, do desenvolvimento de bases de dados construídas especificamente para este fim. Atualmente, a oferta começa a disparar e podem ser destacadas na figura 4.1 (DB-Engines, 2020) as TSDBs mais predominantes no mercado, com particular destaque para o número um, uma vez que é a base de dados utilizada no presente projeto onde todos os detalhes, serão abordados no subcapítulo 4.4. As BDs avaliam-se de acordo com seis parâmetros, sendo eles o número de vezes que a BD é mencionada em websites; a frequência com que é pesquisada nos motores de busca recorrendo ao *Google Trends*; o tipo de discussão técnica sobre o tema; o número de ofertas de trabalho; o número de profissionais de cada BD; e ainda a relevância nas redes sociais (DB-Engines, 2020). Com base nestes fatores, o *InfluxDB* apresenta um valor de 23.34 em setembro de 2020, sendo que a BD que ocupa o segundo lugar, *Kdb+*, apenas foi classificado com 7.43.

□ include secondary database models 34 systems in ranking, September 2020

Rank			DBMS	Database Model	Score		
Sep 2020	Aug 2020	Sep 2019			Sep 2020	Aug 2020	Sep 2019
1.	1.	1.	InfluxDB 📈	Time Series	23.34	+0.46	+4.69
2.	2.	2.	Kdb+ 📈	Time Series, Multi-model 📄	7.43	+0.63	+1.93
3.	3.	3.	Prometheus	Time Series	5.69	+0.33	+2.24
4.	4.	4.	Graphite	Time Series	4.31	+0.20	+0.99
5.	5.	5.	RRDtool	Time Series	3.05	+0.03	+0.48
6.	6.	📈 8.	TimescaleDB 📈	Time Series, Multi-model 📄	2.73	+0.19	+1.33
7.	📈 8.	📉 6.	OpenTSDB	Time Series	2.30	+0.17	+0.45
8.	📉 7.	📉 7.	Druid	Multi-model 📄	2.29	+0.11	+0.56
9.	9.	📈 11.	FaunaDB	Multi-model 📄	1.86	+0.15	+1.43
10.	📈 11.	10.	GridDB 📈	Time Series, Multi-model 📄	0.75	+0.09	+0.26
11.	📉 10.	📉 9.	KalrosDB	Time Series	0.73	+0.05	+0.19
12.	12.	📈 18.	DolphinDB	Time Series	0.69	+0.09	+0.54
13.	13.	📉 12.	eXtremeDB 📈	Multi-model 📄	0.51	+0.06	+0.11
14.	14.	14.	Heroic	Time Series	0.51	+0.06	+0.24
15.	📈 17.	📈 17.	IBM Db2 Event Store	Multi-model 📄	0.43	+0.08	+0.21
16.	16.		Alibaba Cloud TSDB	Time Series	0.41	+0.05	
17.	📈 18.	📉 16.	Riak TS	Time Series	0.37	+0.05	+0.14
18.	📉 15.	📉 13.	Amazon Timestream	Time Series	0.36	-0.01	+0.08

Figura 4.1 – TSDBs mais usadas no mundo datado em setembro de 2020.

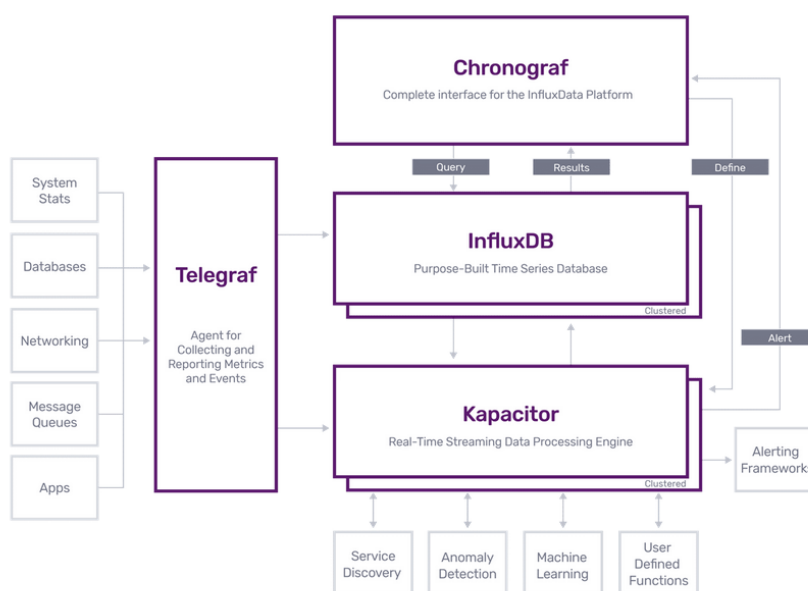
4.4 InfluxDB

Após ter sido abordada a definição e o desenvolvimento do conceito *time series* e, introduzidas as *TSDBs* nos subcapítulos anteriores, o presente subcapítulo focar-se-á na base de dados empregue no projeto. Irá ser dada uma explicação introdutória com um exemplo, seguindo-se de uma explicação detalhada da arquitetura de funcionamento da mesma baseada num fluxograma ilustrativo e clarificador.

4.4.1 Arquitetura e modo de funcionamento

O *InfluxDB* é uma base de dados temporal desenvolvida pela empresa *InfluxData* e lançada oficialmente pela primeira vez a 24 de setembro de 2013 (InfluxDB, 2020). É escrita na linguagem de programação *Go* e, é altamente otimizada para lidar com os conceitos abordados nos capítulos anteriores. Esta solução é dividida numa versão *open source*, complementada por funcionalidades extra não *open source* sendo elas *InfluxEnterprise* (IE) e *InfluxCloud* (IC).

Os resultados experimentais abordados no subcapítulo 4.4.5 são inteiramente baseados na versão *open source*. Esta versão funciona conjuntamente com mais ferramentas e, todas juntas, constituem o modelo *TICK Stack* apresentado na Figura 4.2.



**Figura 4.2 – Modelo *TICK Stack*,
Fonte: Documentação oficial (InfluxDB, 2020).**

O *Telegraf* inicia a sigla “*TICK*” e, é um dos pilares da arquitetura desta base de dados uma vez que é responsável pela aquisição e envio de informação, proveniente das mais diversas

fontes. Essas fontes podem ser outras bases de dados como o *MySQL*, *MongoDB*, *Redis* ou diretamente de sensores *IoT*.

Clarificando o parágrafo anterior, o *Telegraf* recolhe informação de diversos *inputs* e guarda esses dados igualmente em diversos *outputs*.

Para reforçar a versatilidade desta ferramenta, existem mais de 200 *plugins* que permitem incorporar o *Telegraf* nos demais projetos e aplicações.

O *InfluxDB* constitui a base de dados *open source*. É responsável por gerir a forma como a informação é guardada e consultada, bem como o envio da mesma para o *Chronograf* ou outros meios que façam uma requisição da informação. Está integrado na sigla “*TICK*” mas é a única das quatro ferramentas que pode operar sem dependências, querendo isto dizer que, apesar de não ser uma boa prática, é possível instalar apenas o *InfluxDB* num servidor e guardar e consultar informação abrangida pela condição temporal.

O *Chronograf* é um interface administrativo que permite de forma rápida através da *web* criar BDs, apagar BDs, consultar as BDs existentes, visualizar os dados inerentes às mesmas sob a forma de gráficos ou tabelas, extrair dados e executar funções matemáticas. Permite ainda gerir opções de autenticações como supervisão de conexões ativas e gerenciamento de usuários.

O *Kapacitor* desempenha o papel fundamental de supervisionar todo o sistema. Proporciona uma arquitetura de *plugins* que permite integrar um motor de deteção de anomalias integrado em bibliotecas de *machine learning* e deteção de padrões, conferindo ainda mais fiabilidade a todo o sistema. Possibilita também de forma rápida a criação de alertas dinâmicos garantindo um maior controlo dos parâmetros da BD.

É de salientar que a linguagem de consulta é semelhante ao *SQL* e, para uma estruturação mais evidente de vantagens, serão apresentados no subcapítulo 5.4.2 os *aspetos* que mais sobressaem nesta *TSDB*.

No subcapítulo 4.4.5 irão ser dados exemplos de *queries* realizadas numa base de dados em *InfluxDB*.

4.4.2 Destaques do *InfluxDB*

De acordo com a documentação oficial (InfluxDB, 2020). as particularidades que mais se destacam são:

- Armazenamento de dados de alto desempenho desenvolvido especificamente para dados de séries temporais. O motor *Time-Structured Merge (TSM)* permite alta velocidade de obtenção, importação, processamento e compressão de dados.
- Escrito inteiramente em *Go*, a plataforma é compilada num único binário sem dependências externas.
- *APIs HTTP* simples e de alto desempenho para escrita e consulta.
- Suporte de *plug-ins* para outros protocolos de interligação de tecnologias.

- Linguagem de consulta semelhante ao *SQL*, adaptada para consultar facilmente dados agregados.
- As *tags* permitem que as séries sejam indexadas para consultas rápidas e eficientes.
- As políticas de retenção expiram automaticamente e com eficiência os dados obsoletos.
- As consultas contínuas calculam automaticamente os dados agregados para tornar as consultas frequentes mais eficientes.

4.4.3 Inserir informação na BD

No projeto foi utilizado código em *Python* para enviar dados para a BD e recolher dados da mesma BD. Na imagem 5.7 é dado um exemplo de código em *Python* que permite a inserção de valores numa BD *InfluxDB*.

Primeiramente, antes de se conseguir inserir dados recorrendo a linguagem *Python* é necessário instalar a biblioteca oficial para esse fim com o comando:

```
$ python3 -m pip install influxdb
```

Após a instalação da biblioteca basta seguir os passos:

1º - Adicionar biblioteca ao código como mostrado na linha 1 da figura 4.3.

```
1. from influxdb import InfluxDBClient
```

Figura 4.3 – importar biblioteca *InfluxDBClient*

2º - Criar uma instância de cliente como mostrado na linha 5 da figura 4.5. Neste caso, como o sistema opera localmente não existem proteções adicionais, em todo o caso o “*client*” deve possuir no mínimo a localização do destino e porta, em particular está direcionado para o “*localhost*” e a porta “*8086*”. É usual em redes com maior grau de proteção ser solicitado o “*username*”, “*password*”, “*ssl*” e “*verify_ssl*” ficando o comando de acordo com a figura 4.4.

```
>>> client = InfluxDBClient(host='mydomain.com', port=8086,  
username='myuser', password='mypass' ssl=True, verify_ssl=True)
```

Figura 4.4 – todos os parâmetros para conexão a uma base dados *InfluxDB*

```
1. def influxdb_connection():
2.     host = 'localhost'
3.     port = 8086
4.     try:
5.         client = InfluxDBClient(host, port)
6.         return client
7.     except:
8.         print("Erro na tentativa de conexão ao host:", host)
9.         return 0
```

Figura 4.5 – Código *client* para conexão ao *host*

3º - Caso não exista nenhuma base de dados criada é necessário executar-se primeiro esse passo. Pode ser criada uma BD de duas maneiras, uma delas é recorrendo diretamente à ferramenta *web Chronograf* ou no caso do projeto, diretamente em *Python* como mostrado na figura 4.6.

```
1. def influxdb_create_table(name_table:chr):
2.
3.     client = influxdb_connection()
4.     client.create_database(name_table)
```

Figura 4.6 – Código para a criação de uma BD

4º - Após ter sido criada a BD já é possível enviar informação. Na figura 4.7 é exemplificada uma forma de enviar leituras para uma BD.

Neste caso optou-se por enviar a informação em formato JSON e, existem alguns campos a ter em consideração.

O primeiro campo diz respeito a “*measurement*”. A “*measurement*” numa lógica de base de dados tradicional funciona com o nome da tabela onde serão guardados os vários valores dos mais diversos tipos de variáveis. Neste exemplo criou-se uma “tabela” para armazenar as leituras dos *smart meters* com o nome de “*meter_measurements*”. Neste seguimento, é necessário guardar vários valores das leituras. Esses valores são organizados no interior do campo “*fields*”, cada “*measurement*” pode conter vários “*fields*” associados. Cada valor integrante do campo “*field*” é referido a uma *string* identificadora, no exemplo tem-se 8 *strings* que dizem respeito a 8 tipos de valores diferentes. É importante referir que cada bloco de dados enviado para o interior da base de dados está associado a um valor específico no tempo, isto é,

no caso do exemplo, os 8 valores ficam associados ao tempo '1599494384012000'. Este tempo está em nano segundos (ns), traduzido para um grupo data hora obtém-se o valor: 07/09/2020, 16:59:44.

É importante notar duas coisas: primeiramente, para a base de dados aceitar a inserção de valores, o campo “*fields*” tem de ter pelo menos um valor válido associado. Em segundo lugar, o *InfluxDB*, por defeito, usa a hora local da máquina onde está instalado em formato UTC. Deste modo é necessário averiguar se todos os *hosts* secundários estão nas mesmas localizações geográficas com as mesmas definições temporais.

```

1. def
   influxdb_sendData(tensao:float,frequencia:float,Phi:float,S:float,P:float,Q:float):
2.
3.     client = influxdb_connection()
4.
5.     cosPhi = math.cos(Phi)
6.
7.     json_body = [
8.         {
9.             "measurement": "meter_measurements",
10.            "timestamp": 1599494384012000,
11.            "fields": {
12.                "reference": "ZF0000001A",
13.                "voltage": tensao,
14.                "frequency": frequencia,
15.                "P": P,
16.                "Q": Q,
17.                "S": S,
18.                "cosPhi": cosPhi,
19.                "stage": True
20.            }
21.        }
22.    ]
23.    client.write_points(json_body, database='InovGrid',
   time_precision='ms')
```

Figura 4.7 – Código para a criação de uma nova tabela

4.4.4 Extrair informação da BD

Para extrair informação da base de dados, num primeiro momento é necessário conhecer a arquitetura e a forma como os dados foram armazenados e que dados se tem efetivamente. Existem 2 formas de se fazer consultas rápidas para compreender o conteúdo das BD.

Uma das formas mais intuitivas, porém, mais exigentes do ponto de vista de *hardware* é recorrer uma vez mais à ferramenta *web Chronograf* explicada no subcapítulo 4.4.1. Na imagem 4.8 é apresentado um exemplo de consulta da potência ativa registada pelos contadores da rede de testes, agrupados de hora em hora num período de 5 dias. O índice 1 representa o

quadro de visualização gráfica da *query*; o índice 2, o campo onde a mesma é escrita e; em 3, são apresentadas todas as bases de dados que integram o modelo *TICK*, no exemplo são apresentadas 3 “*Measurements*” cada uma com os respetivos “*Fields*” e “*Tags*”, agilizando bastante a compreensão da organização dos dados.

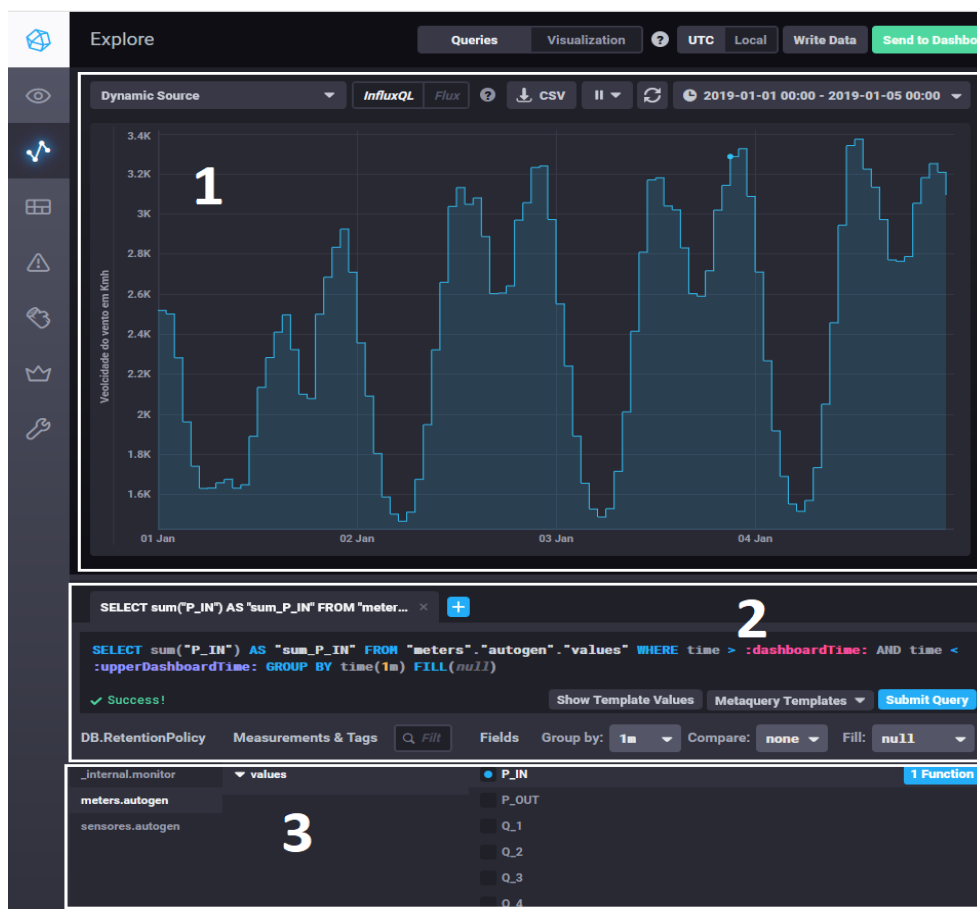


Figura 4.8 – *Dashboard Chronograf*

Outra alternativa menos visual é recorrer diretamente à *engine* do *InfluxDB*. Para isso basta aceder ao *PowerShell* na raiz do *InfluxDB* e correr o comando:

```
> ./influx
```

Uma vez com o terminal a correr o *InfluxDB* já é possível executar alguns comandos para visualizar o interior das bases de dados. Na figura 4.9 são apresentados por comandos os campos mostrados na figura anterior 4.8 porém de uma forma menos gráfica.

O comando:

```
> SHOW DATABASES
```

Mostra todas as bases de dados disponíveis, uma vez mais 3 BD.

O comando:

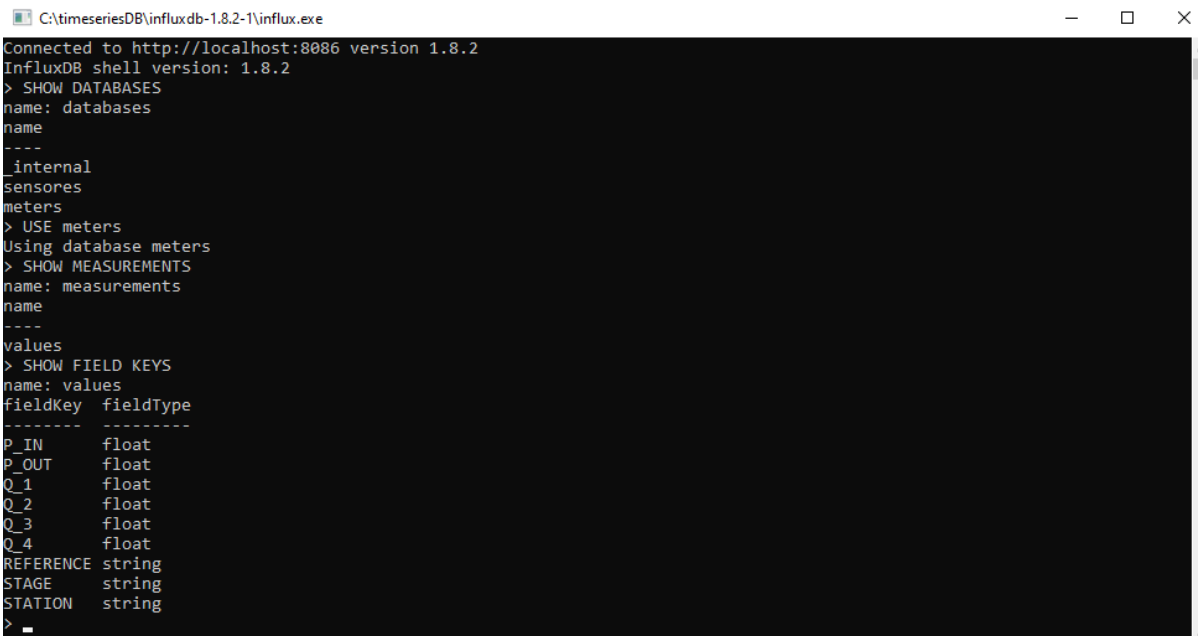
```
> USE meters
```

Seleciona a BD *meters*.

O comando:

```
> SHOW MEASUREMENTS
```

Apresenta os vários nomes dos *fields*, bem como o tipo de variável associado.



```

C:\timeseriesDB\influxdb-1.8.2-1\influx.exe
Connected to http://localhost:8086 version 1.8.2
InfluxDB shell version: 1.8.2
> SHOW DATABASES
name: databases
name
----
_internal
sensors
meters
> USE meters
Using database meters
> SHOW MEASUREMENTS
name: measurements
name
----
values
> SHOW FIELD KEYS
name: values
fieldKey fieldType
-----
P_IN float
P_OUT float
Q_1 float
Q_2 float
Q_3 float
Q_4 float
REFERENCE string
STAGE string
STATION string
>

```

Figura 4.9 – InfluxDB Shell.

Após a compreensão da arquitetura dos dados, é possível desempenhar-se uma consulta da informação desejada de uma forma mais precisa e objetiva.

Uma vez mais, essa consulta pode ser feita diretamente via *web* com a ferramenta evidenciada na figura 4.8 ou, recorrendo ao *Shell* do *InfluxDB*, ou ainda através de uma quantidade enorme de *plugins* oficiais que mais se adequarem ao projeto em causa.

No exemplo do presente relatório, foi usado o *Chronograf*, o *Shell* e predominantemente a linguagem *Python* com a biblioteca oficial para executar *queries* e, desse modo, trabalhar com a informação. No subcapítulo 4.4.5 serão apresentados exemplos de *queries* reais usadas, todavia, um exemplo simples de criação da infraestrutura em *Python* será exemplificado de seguida.

Os passos 1 e 2 desenvolvidos no subcapítulo anterior 4.4.3 serão usados também na consulta da informação. Assim sendo, como mostrado na figura 4.10, após a importação da biblioteca e criada a função para conexão, basta escrever a *query* desejada no interior de uma variável e, seguidamente, de acordo com a linha 10, executar a *query*. Uma boa prática é encerrar a conexão quando esta deixar de ser necessária.

```

1. class query_influx():
2.
3.     def __init__(self):
4.         self.client = influxdb_connection()
5.         self.mydb = mysql_connection()
6.
7.     def count_i(self):
8.         query = 'SELECT COUNT("P_IN") FROM "meters"."autogen"."values"'
9.
10.        Result = self.client.query(query)
11.        self.client.close()
12.        return Result

```

Figura 4.10 – Código para consulta de informação.

4.4.5 Análise de *benchmark* entre uma BD *InfluxDB* e uma BD SQL *MariaDB*

O presente subcapítulo irá expor testes comparativos entre duas bases de dados de diferentes arquiteturas.

Será analisado o tempo de leitura, escrita e leitura/escrita, recursos de CPU e memória consumida, bem como análise da ocupação em disco dos valores.

Para ser possível executar testes precisos sem a existência de subprocessos no servidor principal foram usadas duas máquinas secundárias e, criadas duas BD com dados extraídos do modelo principal, cada uma num respetivo servidor, uma em *InfluxDB* e outra em *MariaDB*. No total, cada BD armazena 100 000 000 valores, dos quais 60% são dados do tipo *Float*, 10% do tipo *Int*, 10% do tipo *Date/Time* e 20% do tipo *Char*. A arquitetura da BD foi mencionada na figura 5.9. Os valores compreendem a potência ativa de entrada, potência ativa de saída, potências reativas, referência, estação e estado do contador.

Os testes foram feitos em duas máquinas idênticas com as especificações de acordo com a tabela 4.1, já as versões de software usadas encontram-se na tabela 4.2

CPU	Processador AMD Ryzen 5 1600X 12CPU ~3.8GHz
RAM	DDR4 - 2132 MHz G.Skill - 16.384Gb
Disco	SSD 2.5" Samsung 860 Evo leitura: Até 550 MB/s escrita: Até 520 MB/s
OS	Microsoft Windows 10 pro 64 bit

Tabela 4.1 – Especificações da máquina de testes

Chronograf	1.8.6-1
InfluxBD	1.8.2-1
MariaDB	10.4.14

Apache	2.4.46
OpenSSL	1.1.1g
PHP	7.4.9

Tabela 4.2 – Versões do software usado

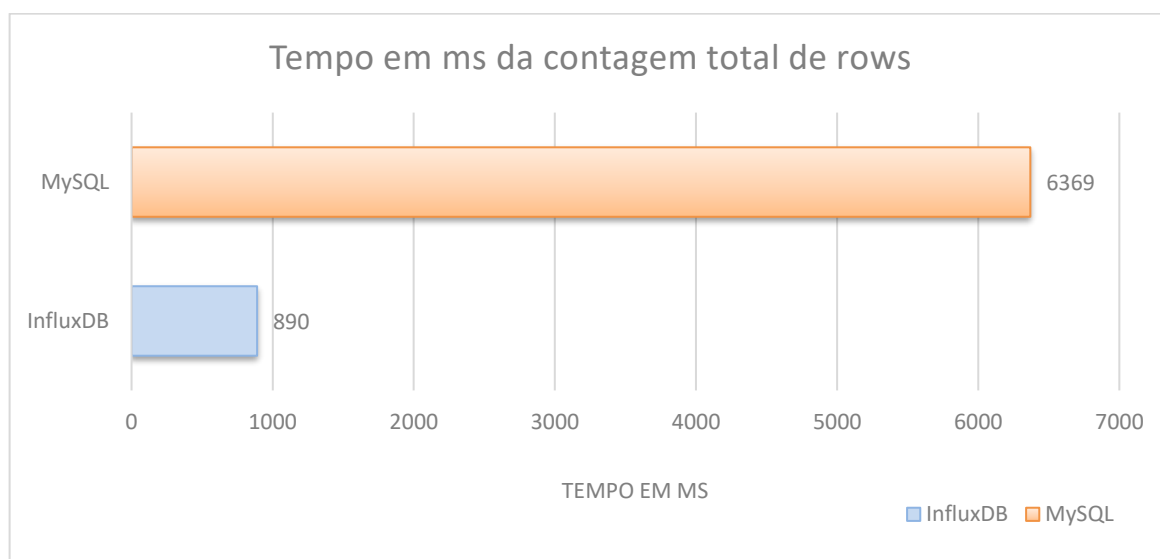
Para contagem de tempo e execução de *queries*, foi usado *software* em *Python* cuja consulta pode ser feita no anexo A. Na totalidade foram feitos grupos de 9 *queries* envolvendo a seleção de dados, cálculos aritméticos e agrupamento de dados em diferentes escalas temporais. Todos os ensaios foram executados 4 vezes, sendo que, os valores apresentados de seguida traduzem a média dos testes. Para reforçar a veracidade dos tempos, foram também comparadas *queries* diretamente na ferramenta *web Chronograf* e *PHP MyAdmin*, desempenhando tempos médios semelhantes aos apresentados em código. Foi usado o *Python* por uma questão de praticidade e fácil organização dos resultados.

1º teste – Tempo de contagem total de registos em cada uma das BD.

Query: Contar o número total de registos de potência de entrada existentes.

```
SELECT
  COUNT(`P_IN`)
FROM
  `measurements`
```

É possível compreender, decorrente da análise do gráfico da figura 4.11 que a BD em *MySQL* levou aproximadamente 6.3 segundos a contar todos os valores de potência ativa de entrada existentes, contra 890 milissegundos em *InfluxDB*. Assim sendo, o mesmo teste, foi aproximadamente executado 7,15 vezes mais rápido para o caso da BD temporal.

Figura 4.11 – Gráfico Tempo em ms da contagem total de *rows*.

2º teste – Tempo de contagem de contadores ativos em diferentes intervalos.

Query: Contar o número de contadores ativos na rede durante o mês de janeiro de 2019 agrupados por hora.

```
SELECT
  COUNT(`STAGE`),
  DATE (DATE_TIME)
FROM
  `measurements`
WHERE
  `DATE_TIME` BETWEEN '2019-01-01 00:00:00' AND '2019-01-31 00:00:00'
GROUP BY
  HOUR (DATE_TIME)
```

O gráfico da figura 4.12 traduz os tempos em ms de contagem de *Smart meters* ativos, porém foi introduzido o conceito temporal na *query*. Desta forma, foram executados 4 ensaios para períodos temporais distintos, aumentando significativamente a totalidade de dados processados à medida que esse intervalo temporal foi expandido. Os tempos foram medidos para um intervalo temporal de 7 dias, 20 dias, 35 dias e 58 dias, apresentando um aumento gradual em tempo de processamento em ambas as bases de dados. Contudo, uma vez mais, a base de dados em *InfluxDB* apresentou resultados 8.78 vezes mais rápidos que a concorrente.

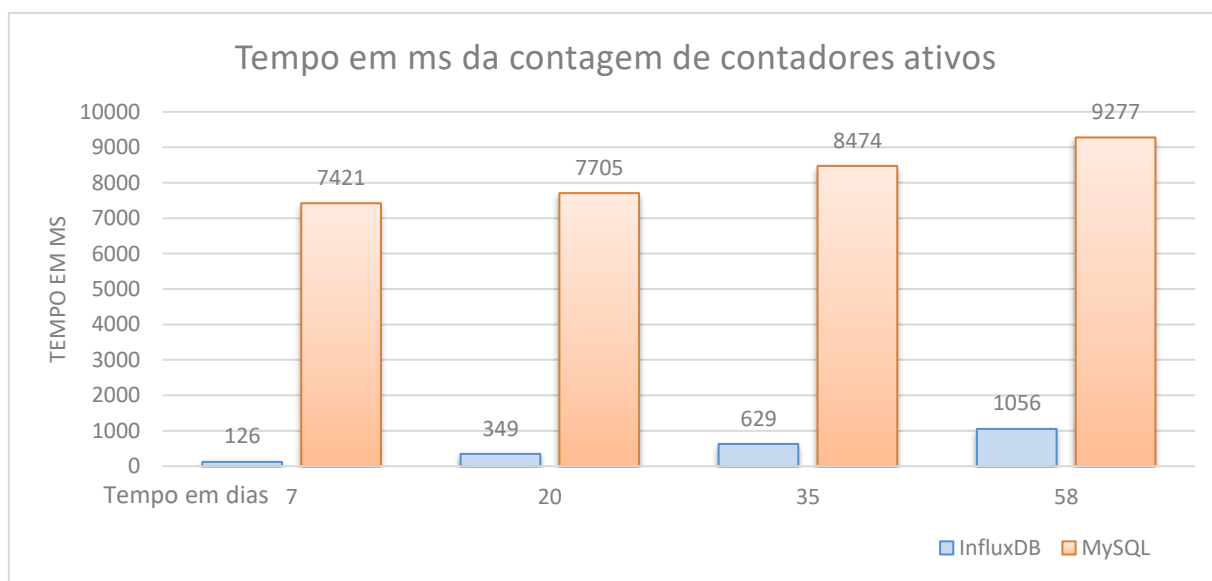


Figura 4.12 – Gráfico do tempo em ms da contagem de contadores ativos.

3º teste – Somar a totalidade da potência ativa agrupada por dia em diferentes intervalos.

Query: Somar a totalidade da potência ativa de entrada durante o mês de janeiro de 2019 agrupada por dia.

```
SELECT
    SUM(`P_IN`),
    DATE (DATE_TIME)
FROM
    `measurements`
WHERE
    `DATE_TIME` BETWEEN '2019-01-01 00:00:00' AND '2019-01-31 00:00:00'
GROUP BY
    DAY (DATE_TIME)
```

A figura 4.13 mostra o tempo total em milissegundos do tempo de cálculo da soma das potências ativas para um período temporal de 7 dias, 20 dias, 35 dias e 58 dias, agrupados por dia. Voltou a verificar-se uma grande vantagem de tempo para a BD temporal. No cenário com maior volume de dados a *InfluxDB* calculou o que lhe foi proposto em 835 milissegundos contra 9.2 segundos em *MySQL*, sendo assim 11.1 vezes mais veloz. Foram desempenhadas outras comparações aritméticas entre as duas bases de dados, contudo todos os resultados foram similares percentualmente a este exemplo, daí não serem mencionados.

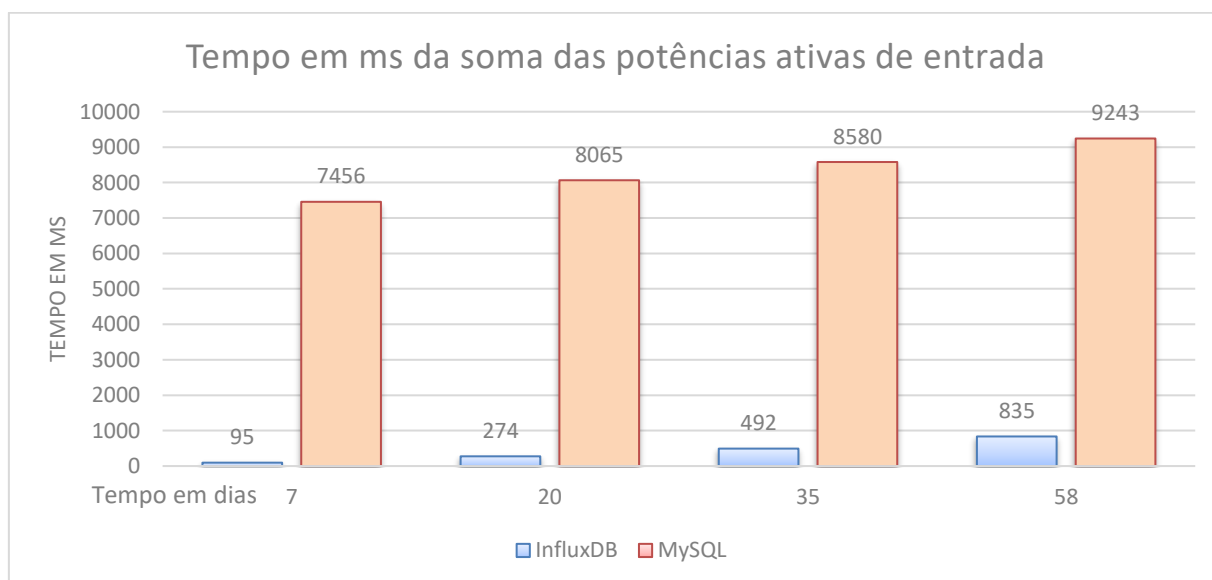


Figura 4.13 – Tempo em ms da soma das potências ativas de entrada.

4º teste – Tempo de seleção de todos os registos onde a potência reativa é superior a 0.2 VAR

Query: Selecionar a totalidade de potências reativas superiores a 0.2 VAR

```
SELECT
  (`Q_1`)
FROM
  `measurements`
WHERE
  `Q_1` > 0.2
```

Na figura 4.14 é apresentado o tempo em milissegundos gastos na seleção de todas as potências reativas superiores a 0.2VAR. É notório que neste ensaio não se usou nenhum tipo de agrupamento de dados ou intervalo temporal, sendo possivelmente esse o motivo pela aproximação em tempos de execução das bases de dados. Desta vez, apesar de continuar a ser superior, o *InfluxDB* apenas conseguiu ser 1.2 vezes superior.

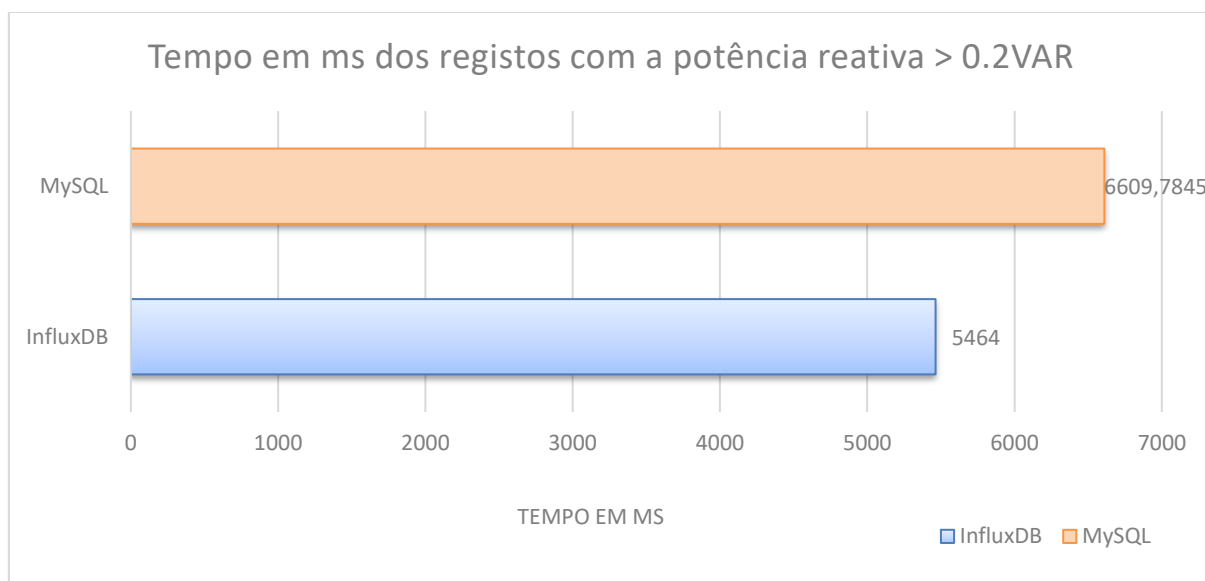


Figura 4.14 – Gráfico do tempo em ms dos registos com a potência reativa > 0.2VAR

5º teste – Tempo de calculo em ms do valor médio da potência ativa

Query: Calcular o valor médio da potência ativa

```
SELECT
  AVG(`P_IN`)
FROM
  `measurements`
```

Após o resultado anterior, o ensaio da figura 4.15 pretende voltar a testar uma *query* sem grupos ou intervalos temporais, todavia ambas as bases de dados irão desempenhar funções matemáticas para calcular o valor médio da potência ativa. Sobressai novamente a diferença temporal de 776 milissegundos em *InfluxDB* para 6.5 segundos em *MySQL*.

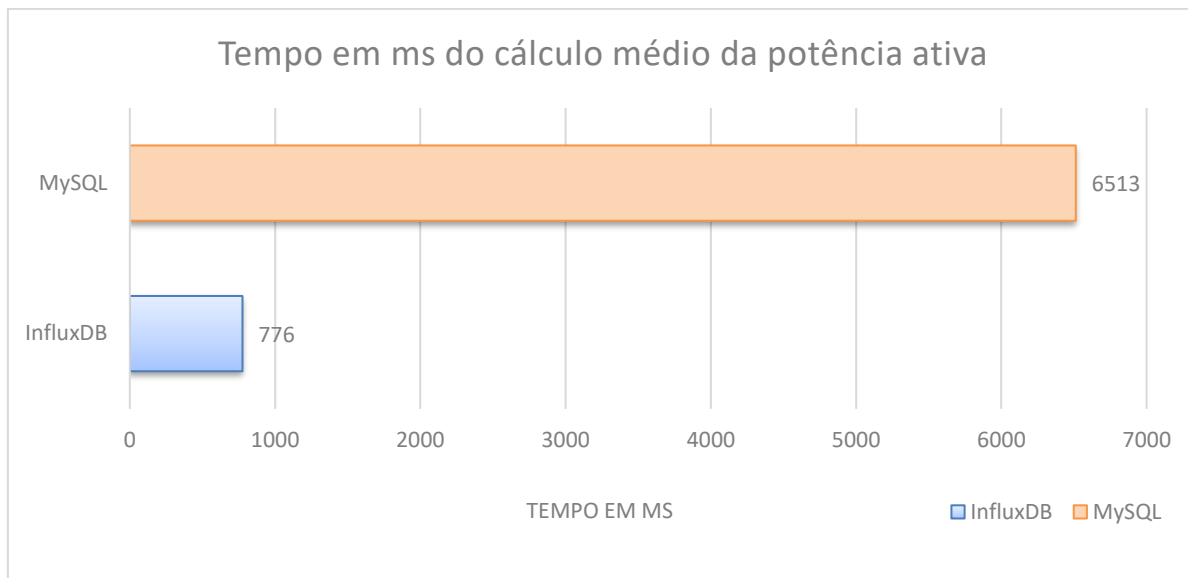


Figura 4.15 – Gráfico do tempo em ms do cálculo médio da potência ativa.

6º teste – Tempo de escrita

Foram executados testes de escrita de valores nas respetivas BD. Os gráficos das figuras 4.16 e 4.17 traduzem em tempo o resultado da escrita de dados. Foram escritos e medidos temporalmente 10 valores, 100 valores, 1000 valores, 10 000 valores e 100 000 valores. Os resultados foram apresentados em dois gráficos para ser perceptível os tempos das quantidades menores de dados. Assim, no primeiro gráfico, o resultado é apresentado em ms e no segundo gráfico em segundos. A inserção de dados foi feita no caso do *InfluxDB* em formato *JSON*. O código para o envio de valores para as BD pode ser consultado no anexo A.

Da análise da informação apresentada, é visível que em velocidade de escrita, o *MySQL* supera o *InfluxDB*. Em valores médios, foi possível inserir 395 valores por segundo em *MySQL* contra 289 em *InfluxDB*.

Existem várias formas de inserir valores em base de dados, neste projeto apenas é contemplado um método, que revelou desfavorecer o *InfluxDB* em matéria de escrita. Todavia, a *TSDB* em questão apresenta uma ferramenta direcionada para esta função, o mecanismo *Telegraf*, abordado no subcapítulo 4.4.1, cujo método de inserção não foi alvo de testes uma vez que o mesmo, é sobretudo direcionado para inserção de dados em tempo real proveniente diretamente dos sensores, não permitindo comparar ativamente este método com um DB relacional.

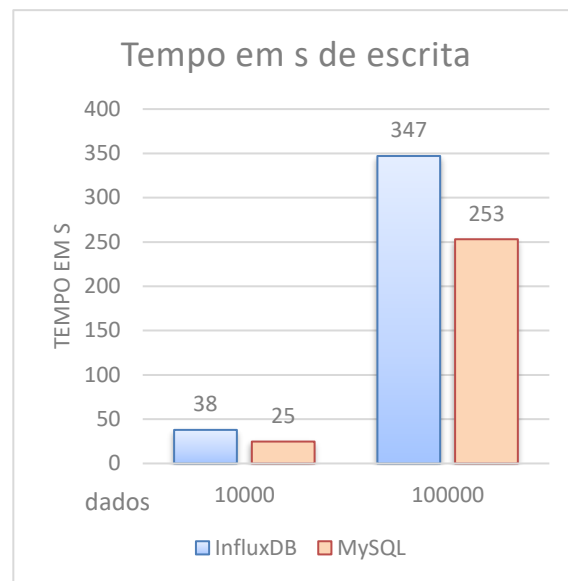
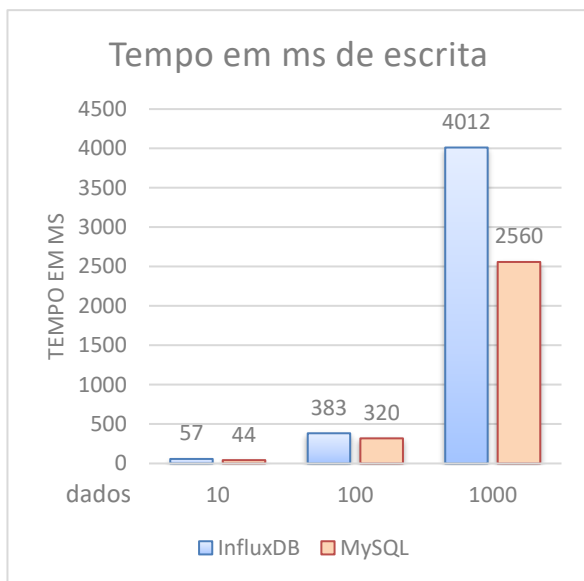


Figura 4.16 – Tempo de escrita em milissegundos. Figura 4.17 – Tempo de escrita em segundos.

7º teste – Tempo de leitura e escrita em simultâneo

Foi repetido o ensaio 5, mas desta vez, durante o tempo de execução da *query* as BD estiveram a receber dados. A inserção dos valores foi do tipo abordado anteriormente e as *queries* foram desempenhadas sob as mesmas condições.

O gráfico da figura 4.17 apresenta os resultados. Conclui-se que o tempo de execução em *InfluxDB* aumentou aproximadamente 6.85% enquanto que o *MySQL* aumentou 26.54%.

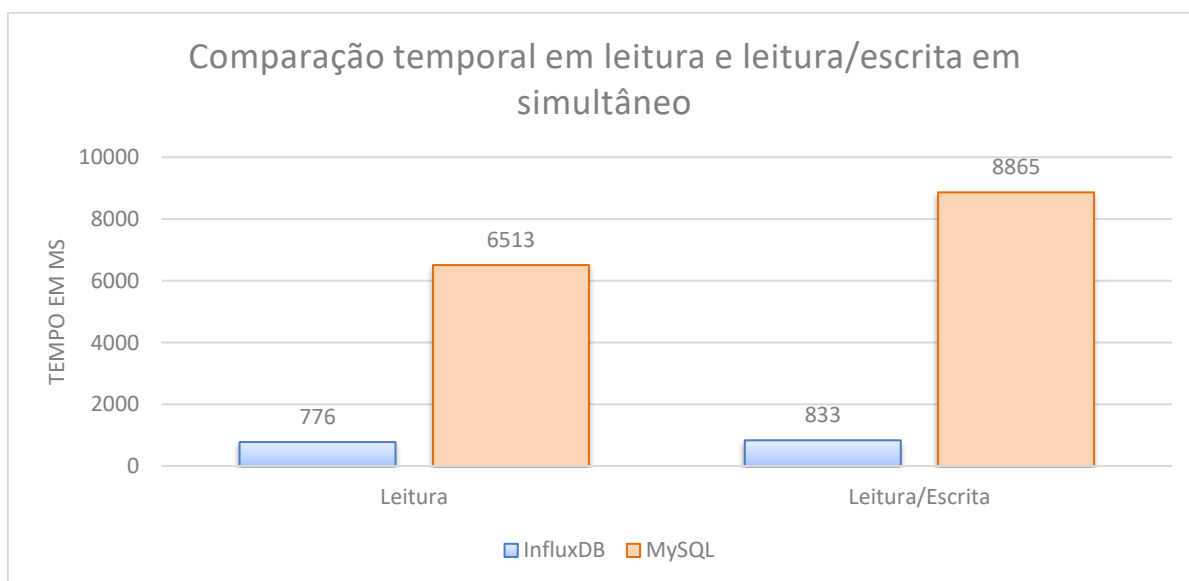


Figura 4.18 – Gráfico de comparação temporal de escrita e leitura/escrita em milissegundos.

8º teste – Espaço ocupado em disco rígido.

Em *Bigdata* a compactação da informação, a forma como é armazenada e o espaço que ocupa em disco, são três fatores de peso com impactos financeiros diretos.

Para as bases de dados em questão, é uma vez mais reforçado que o tipo, quantidade e valores são iguais. Posto isto, no gráfico da figura 4.18 são apresentados em MB as diferenças entre ambas as BD. O destaque é dado ao *InfluxDB*, uma vez que para os mesmos dados, o volume total ocupado é de 441 MB enquanto que em *MySQL* o valor é de 764 MB poupando-se, portanto, em contexto real cerca de 42.3%.

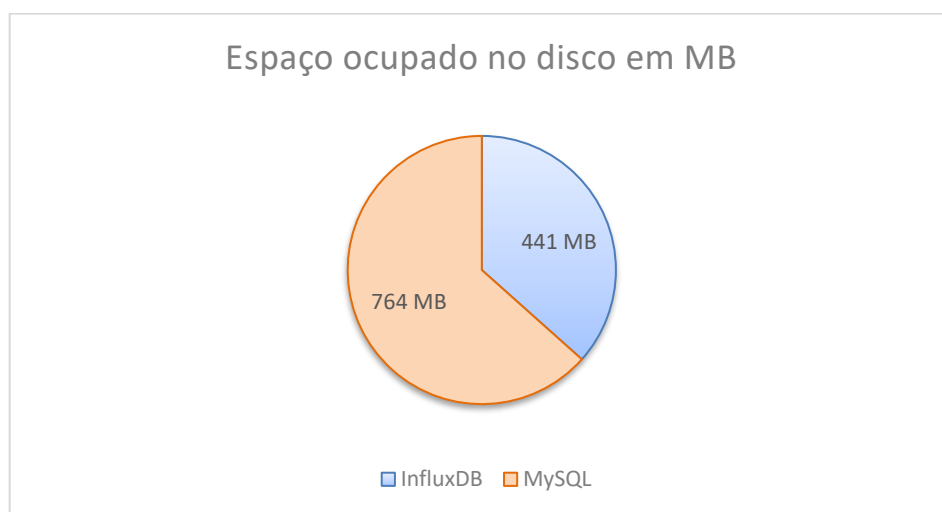


Figura 4.19 – Espaço ocupado no disco em MB.

9º teste – Recursos de CPU e memória RAM

Durante os ensaios foram registados os valores de ocupação percentual de CPU e memória RAM. Na figura 4.20 é apresentado o registo para o servidor em *InfluxDB* e, na figura 4.21, o registo referente ao servidor em *MySQL*. Os valores “*CPU usage*” correspondem à soma de todos os núcleos.

Globalmente, de acordo com a dimensão das bases de dados, é possível observar que ambas as máquinas suportaram em pleno as condições exigidas. Não existem diferenças demasiado vincadas a nível de processador e RAM. Pode verificar-se uma semelhança da ocupação da memória RAM e, uma taxa média de utilização do CPU superior para a BD *MySQL*, todavia, não foram feitos teste de stress que levassem os servidores a um valor limite.

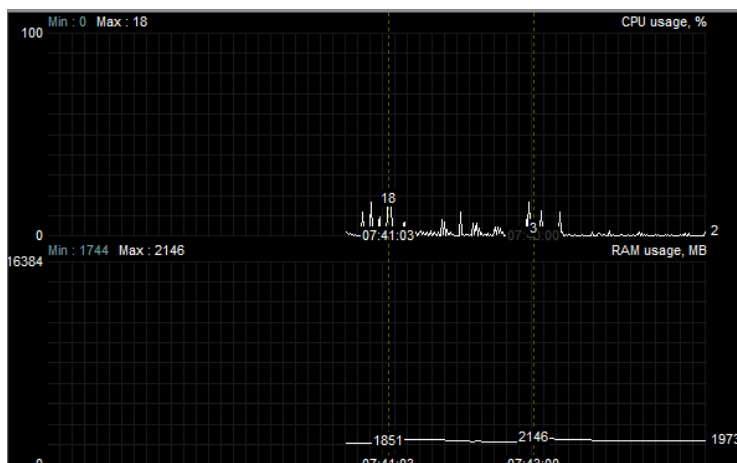


Figura 4.20 – Recursos de CPU e memória RAM em *InfluxDB*.

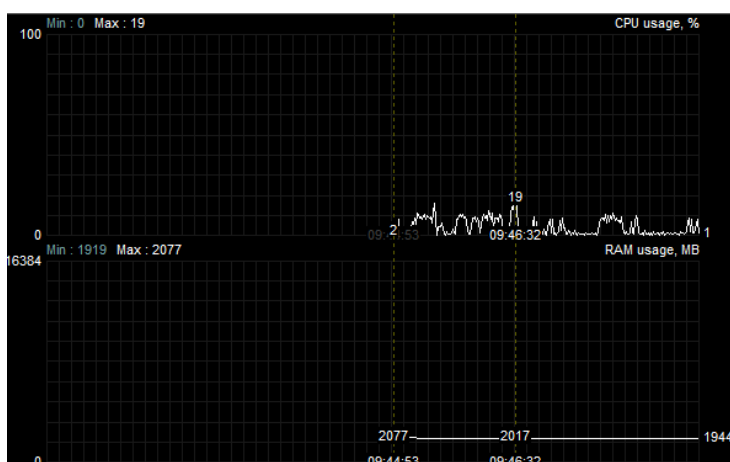


Figura 4.21 – Recursos de CPU e memória RAM em *MySQL*.

4.5 Conclusão

Este capítulo revela a importância das Bases de Dados no projeto. Tornou-se evidente que a escolha das bases de dados tem uma repercussão acentuada nos objetivos que se ambicionam cumprir.

Optou-se por Bases de Dados temporais, isto porque, analisando os nove ensaios, sobressaiu uma diferença temporal de performance entre as duas Bases de Dados, destacando claramente o *InfluxDB* com taxas percentuais bastante superiores, comparativamente ao *MySQL* em funções semelhantes. É de salientar que em aspetos temporais e aritméticos a primeira BD devolve resultados em tempos próximos do instantâneo, inculindo assim uma maior fluidez no projeto. Não obstante, a BD *MySQL*, em termos de escrita, apresentou uma performance superior, o que, em alguns casos, pode constituir uma enorme vantagem, evitando e minimizando perdas de armazenamento de informação.

5 Estudo e análise da rede de distribuição

Após a recolha e armazenamento de dados, é fundamental fazer-se uma boa análise e eficiente processamento dos mesmos, de forma a otimizar processos, detetar anomalias e identificar padrões. Melhora-se assim a tomada de decisões e torna-se a visualização dos dados um mecanismo indispensável no projeto e na grande maioria dos setores.

Este subcapítulo está dividido por duas grandes ferramentas de visualização e, em cada uma das partes será dada uma explicação introdutória bem como o porquê de terem sido aplicadas no projeto, encerrando com ilustrações reais da rede de testes.

5.1 Visualização dos dados

Ao longo do capítulo, verifica-se que a base de dados usada (*InfluxDB*) está fortemente direcionada para algumas ferramentas de visualização. Uma das ferramentas mais usuais e eficientes na visualização de dados provenientes de uma base de dados temporal é o *Grafana*.

Esta plataforma exerce um trabalho excelente na visualização de dados sobretudo na forma de gráficos em tempo real (histogramas, linhas, barras, circulares, mapas de calor). Permite criar *dashboards*, configurar alertas, manipular rapidamente a escala temporal e, usa uma linguagem com sintaxe muito idêntica à linguagem *SQL* facilitando assim a aquisição e consequente exposição de dados.

É de salientar que o acesso à plataforma é feito via *web browser*.

Existem três grandes vantagens que levaram à criação e implementação de um servidor em *Grafana*.

Primeiramente a base de dados possui código aberto de um *plug-in* direto para o *Grafana*, evitando assim questões complexas de segurança, problemas de compatibilidade na visualização e, no limite, *crashes* aleatórios característicos de versões *beta* desenvolvidas com poucos recursos ou com uma arquitetura menos direcionada a *TSDBs*.

Por outro lado, ainda que exista uma aparente imensidão de ferramentas, a arquitetura do *Grafana* minimiza a utilização intensiva do *CPU* e memória *RAM*.

É ainda uma ferramenta *open source*, o que facilita a criação de uma comunidade ampla e participativa, agilizando desse modo a resolução de erros ou dúvidas de implementação.

5.1.1 Dependências e instalação

Segundo a documentação oficial (Grafana, 2020) o *Grafana* pode ser instalado nos quatro sistemas operativos mais comuns sendo eles, *Debian/Ubuntu*, *RPM-based Linux* (*CentOS*, *Fedora*, *OpenSuse*, *RedHat*), *macOS* e *Windows*. As recomendações mínimas são 255MB de memória *RAM* e 1 núcleo de *CPU*. Todos os principais *browsers* são igualmente suportados nomeadamente *Chrome/Chromium*, *Firefox*, *Safari*, *Microsoft Edge* e *Internet Explorer 11*. A porta de acesso por defeito é a 3000.

5.1.2 Postos de Transformação ativos

Os pontos da figura 5.1 representam a totalidade de postos de transformação (PT) da globalidade da rede.

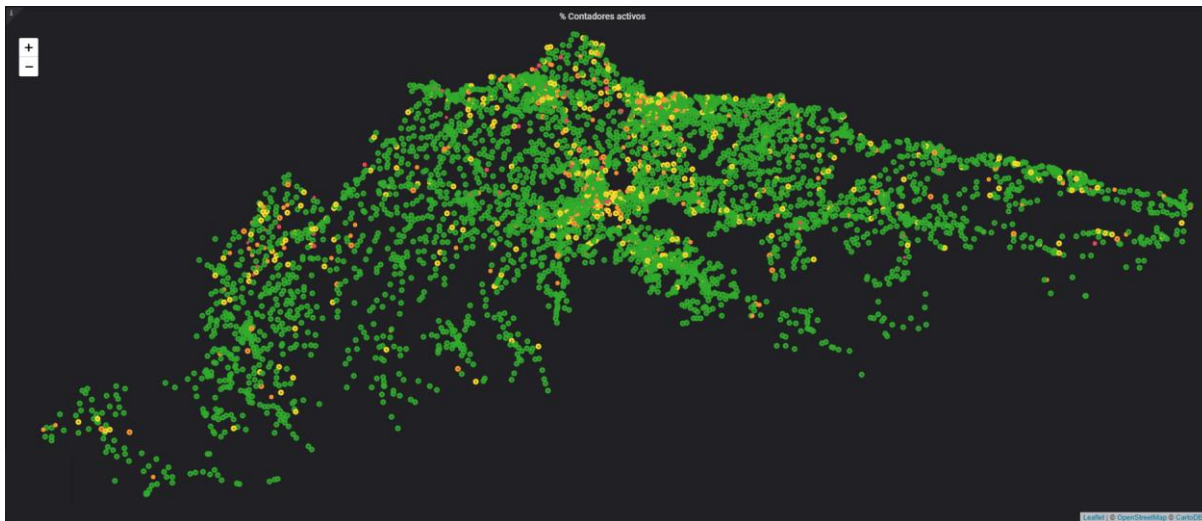


Figura 5.1 – Disposição geográfica dos postos de transformação da rede global.

5.1.3 Valores instantâneos de consumo de um posto de transformação

Neste subcapítulo irão ser apresentadas algumas ilustrações de um PT de referência para demonstrar as capacidades desta ferramenta.

Potência ativa instantânea

Na figura 5.2 é apresentado dividido pelas fases pa, pb e pc os valores instantâneos de consumos de potência ativa ao longo dos últimos 3 anos para o transformador em questão. É possível observar-se que as fases estão desequilibradas, verifica-se que a fase 'a' é mais sobrecarregada atingindo picos de 40kW e um valor médio de 25kW. Já a fase 'b' é a fase menos explorada com picos menos acentuados na ordem dos 20kW e um valor médio de 10kW. A fase 'c' é uma fase intermédia comparativamente às restantes, com picos superiores a 30kW e valor médio de 15kW.

Um dos trabalhos que se tem vindo a desenvolver é exatamente a correção das disparidades de consumos entre fases, tornado a rede mais estável e eficiente.

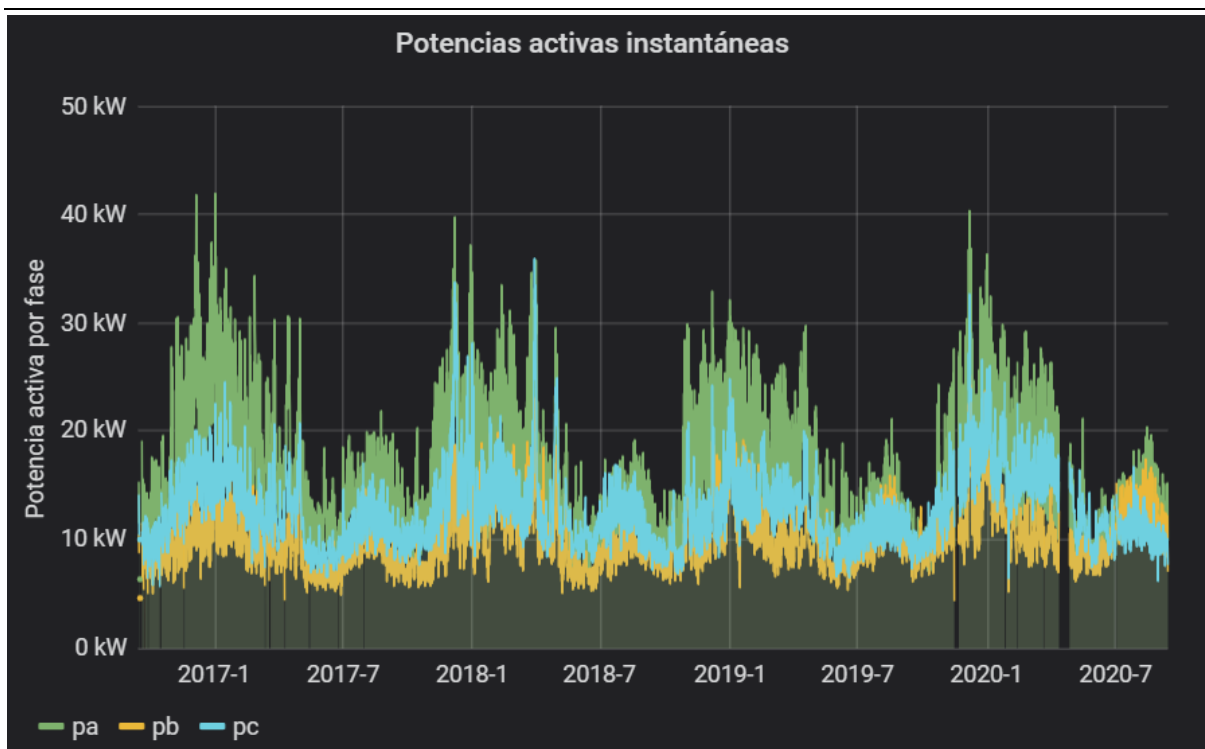


Figura 5.2 – Potência ativa em kW por fase ao longo dos últimos 3 anos.

Potência reativa instantânea

Na figura 5.3, à semelhança da anterior, são mostradas por fase os valores de potência reativa consumidos nos últimos 3 anos. É notória a similaridade de relação entre a fase com maior valor de potência ativa e a fase com maior valor de potencia reativa. Isto é, a fase 'a' apresenta o maior valor em ambas as potências, seguindo-se a mesma lógica para a fase 'b' e 'c'.

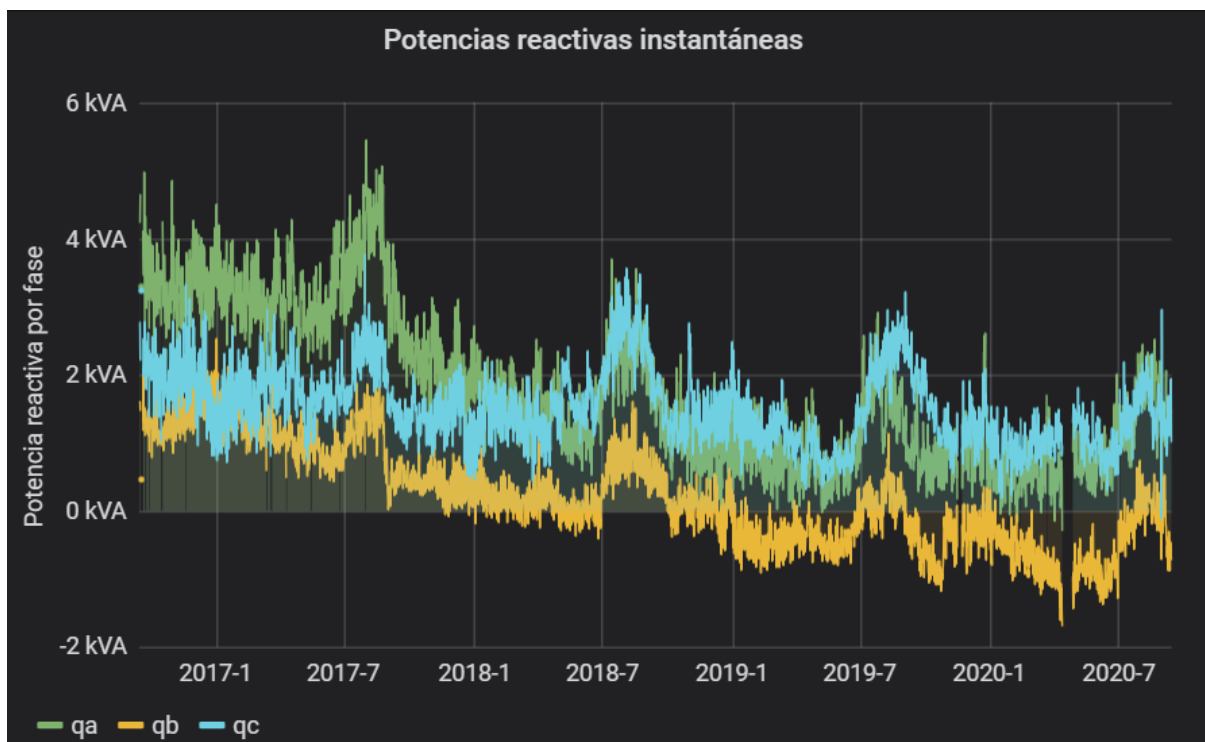


Figura 5.3 – Potência reativa em kVA por fase ao longo dos últimos 3 anos.

Carga em amperes por fase

Partindo do pressuposto que os valores de tensão não oscilam abruptamente é expectável o comportamento evidenciado na figura 6.4, assim sendo, a fase ‘a’ apresenta o maior valor de corrente, seguindo-se a fase ‘b’ e ‘c’.

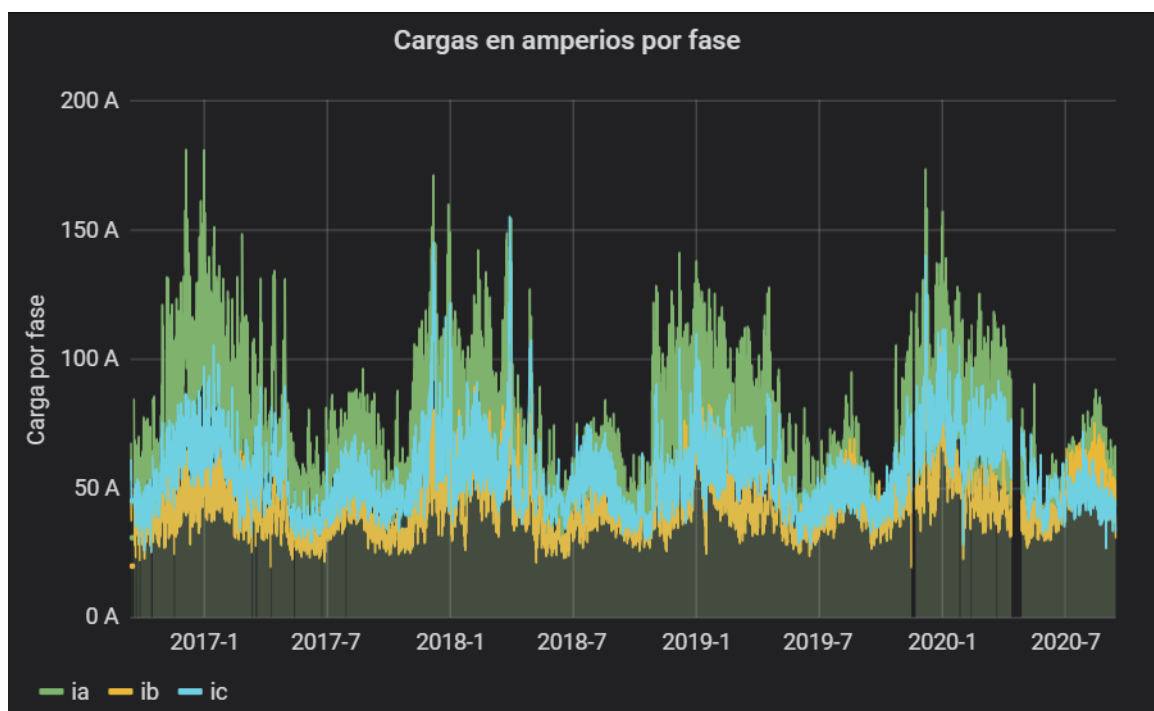


Figura 5.4 – Disposição geográfica dos contadores inteligentes ativo

Tensão por fase

As tensões nas 3 fases expostas na figura 5.5 apresentam algumas oscilações, normais à exploração e funcionamento da rede.

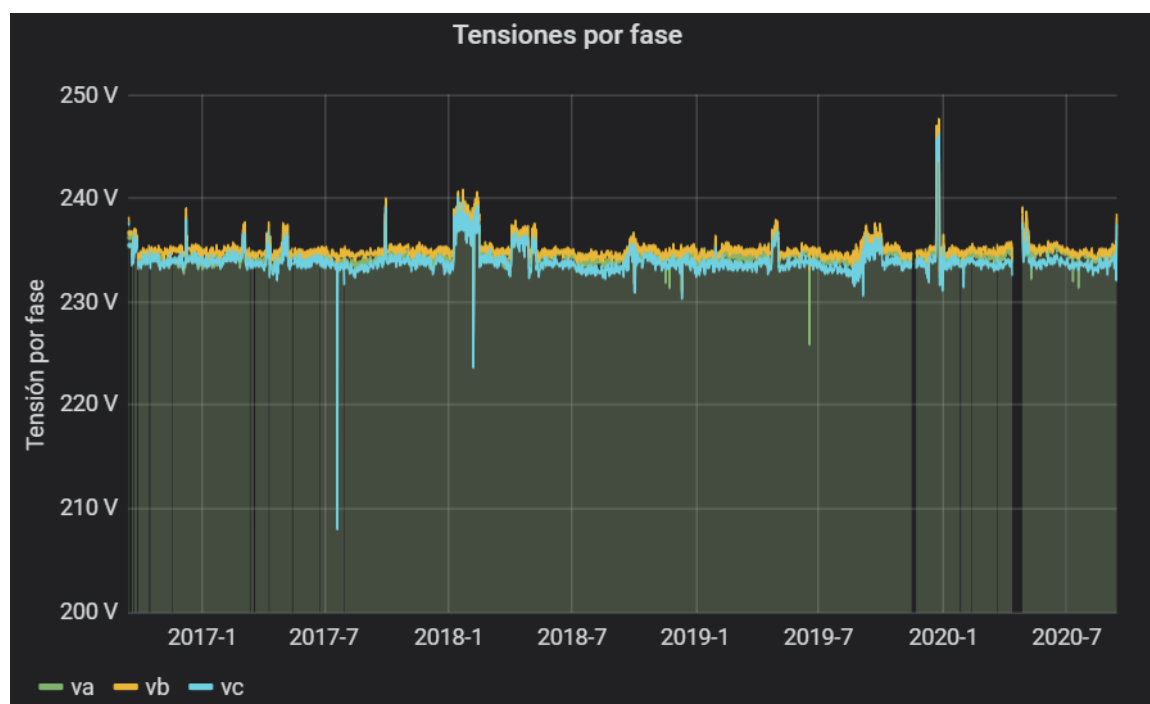


Figura 5.5 – Tensão em V por fase ao longo dos últimos 3 anos.

Temperatura

Na figura 5.6 é apresentada a variação de temperatura do posto de transformação, temperatura que varia consoante os meses do ano, verificando-se picos mais elevados, evidentemente, nos meses de verão.

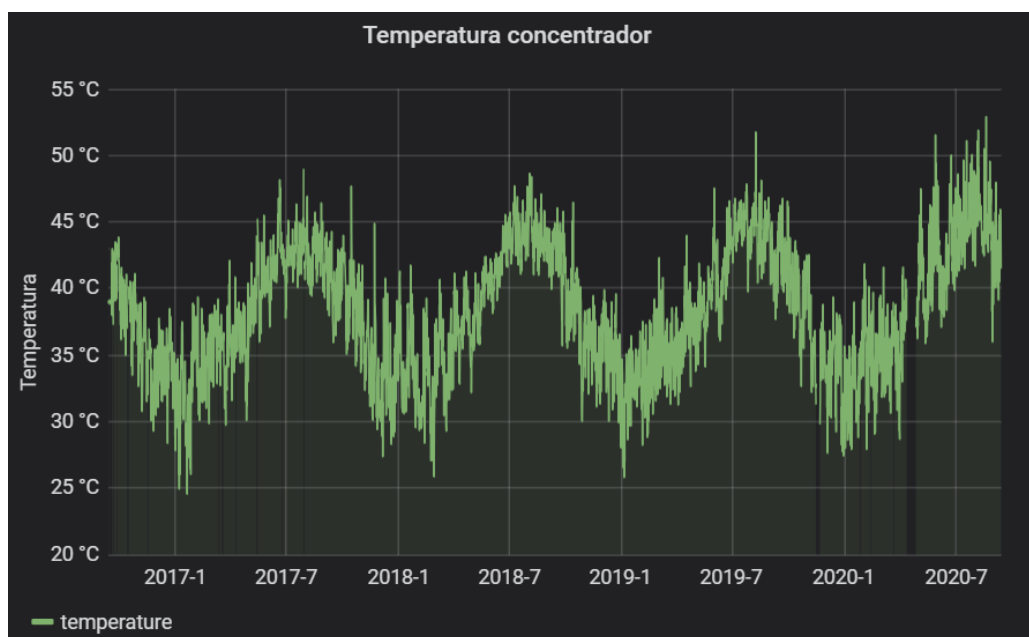


Figura 5.6 – Variações em °C da temperatura do PT ao longo dos últimos 3 anos.

CPU e Memória RAM

A figura 5.7 representa em percentagem (%) a utilização *CPU* e memória *RAM* do posto de transformação, porém, numa lógica tradicional os transformadores não possuem unidades de processamento tão avançadas. No entanto, como estamos a lidar com uma *smart grid*, é neste posto que são agrupadas todas as leituras provenientes dos vários *smart meters* associados para serem posteriormente enviadas e armazenadas daí a necessidade da introdução de *CPU* e memória.

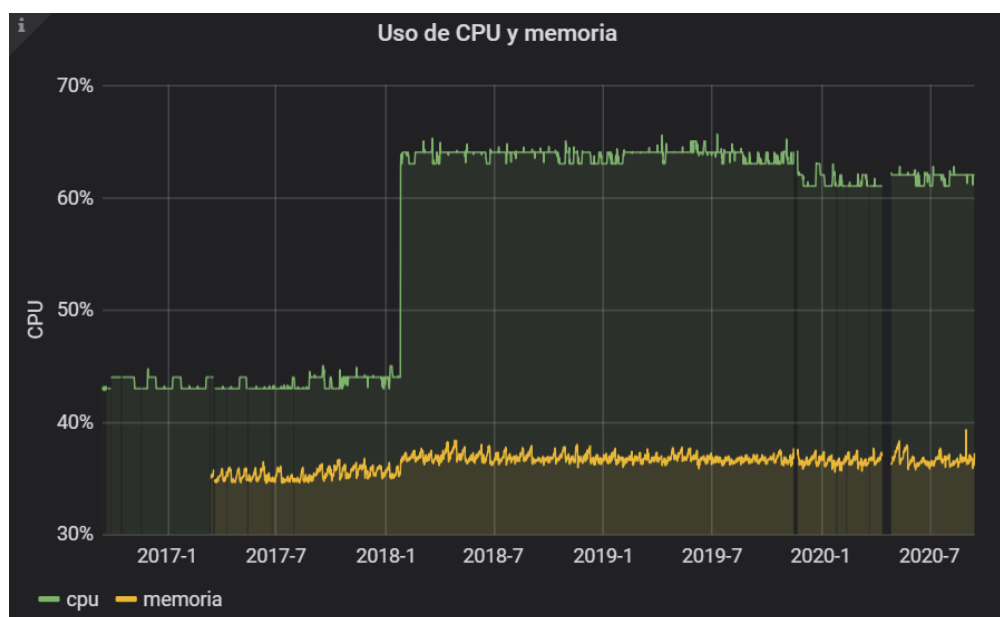


Figura 5.7 – Utilização em % do uso do *CPU* e memória *RAM* ao longo dos últimos 3 anos.

5.1.4 Potência total da rede

5.1.5.1 Durante um dia

A figura 5.8 representa o consumo total de potência ativa da rede global para o dia 25 de agosto, é visível um decréscimo espectável durante as horas de madrugada, aumentado gradualmente a partir das primeiras horas da manhã. O pico atingiu-se por volta das 23h, o que em algumas alturas do ano é pouco comum. A razão do sucedido está fortemente relacionada com o facto de ser um dia de verão no ano da pandemia corona vírus.

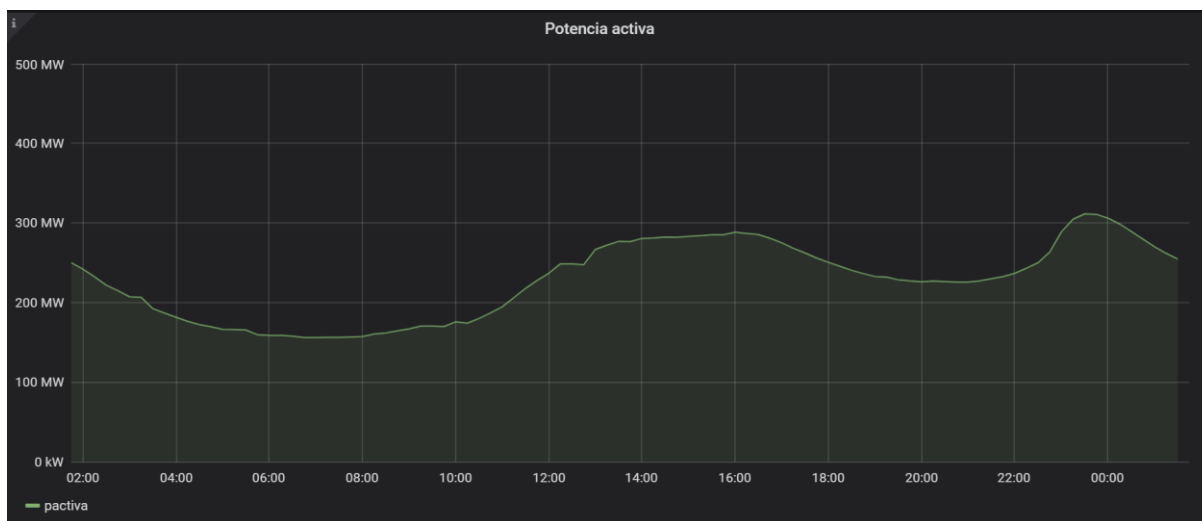


Figura 5.8 – consumo total de potência ativa da rede global no dia 25 de agosto de 2020.

5.1.5.2 Durante uma semana

A figura 5.9 apresenta a curva de potência ativa da rede global entre os dias 6 e 13 de setembro de 2020. Os picos máximos superam os 300MW de potência e registam-se próximos da meia-noite, já os mínimos ascendem os 150MW e ocorrem durante a madrugada.

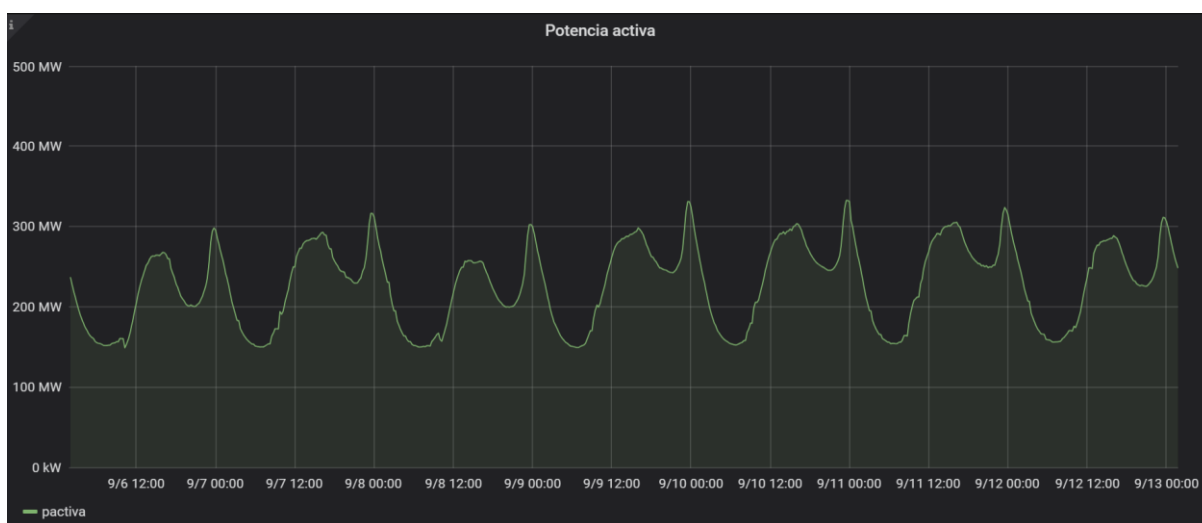


Figura 5.9 – curva de potência ativa da rede global entre os dias 6 e 13 de setembro de 2020

5.1.5.3 Durante um mês

O padrão de consumos replica-se com relativa aproximação ao longo de todo o mês de setembro. São visíveis na figura 5.10, ainda alguns momentos em que os valores de potência ativa são nulos. Esses comportamentos ocorrem, pois, sendo uma rede de testes é necessário por vezes reiniciar processos. Contudo, o armazenamento dos dados apresenta redundância, estando a informação em falta guardada num servidor de *backup*.



Figura 5.10 – curva de potência ativa da rede global entre os dias 13 de agosto e o dia 13 de setembro de 2020

5.1.5.4 Durante o último ano

A figura 5.11 revela o padrão de consumos de toda a rede durante o último ano.

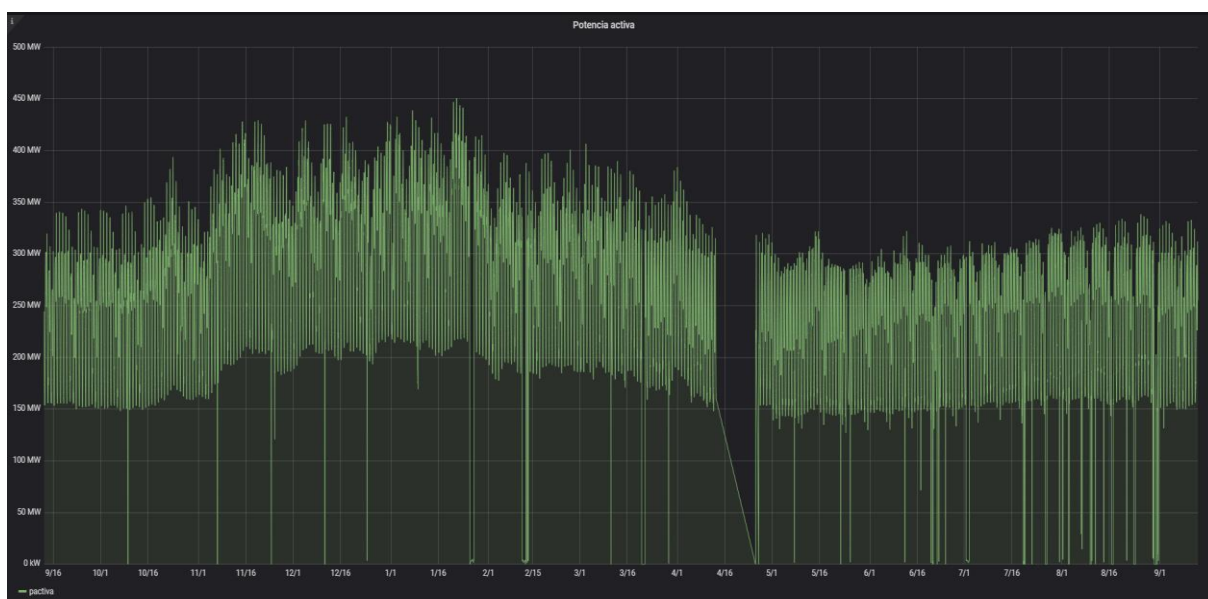


Figura 5.11 – curva de potência ativa da rede global compreendida entre agosto de 2019 e agosto de 2020.

5.1.6 Comparação dos consumos nos últimos dois anos

Foi introduzida uma imagem comparativa do ano 2020 e do ano 2019 com o objetivo claro de se identificarem padrões distintos entre os dois anos, estimulados pela necessidade de adaptação das empresas e das pessoas à situação da pandemia.

Da análise dos dados, não se encontram comportamentos vincados característicos de um ano e de outro. No mês de março e abril o consumo do ano 2020 superou em aproximadamente 8% o ano anterior, não obstante, é difícil estabelecer diferenças concretas uma vez que o consumo doméstico aumenta e o industrial diminui, traduzindo-se esta relação de proporcionalidade inversa na figura 5.12.

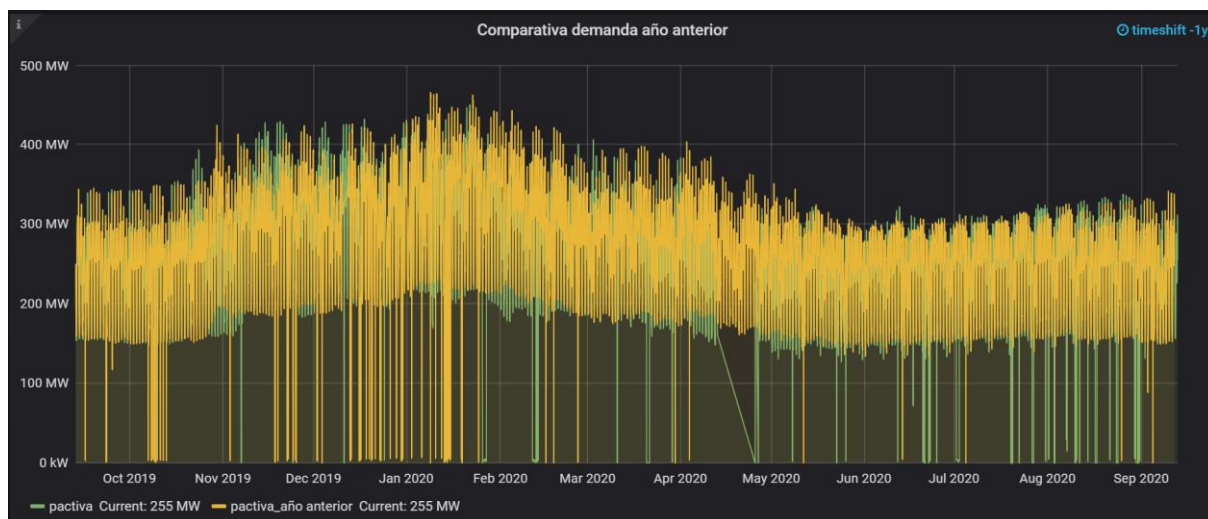


Figura 5.12 – Comparação de consumos entre o ano 2019 e 2020.

5.1.7 Monitorização dos servidores

Na imagem 5.13 são apresentados os *bytes* recebidos por segundo no servidor principal ao longo de 8 dias do mês de setembro de 2020. É visível um pico superior a 2.5 MBs no dia 9, ainda assim como as bases de dados se encontram num processo migratório é expectável que nesta fase do projeto os valores não ascendam taxas demasiado altas.

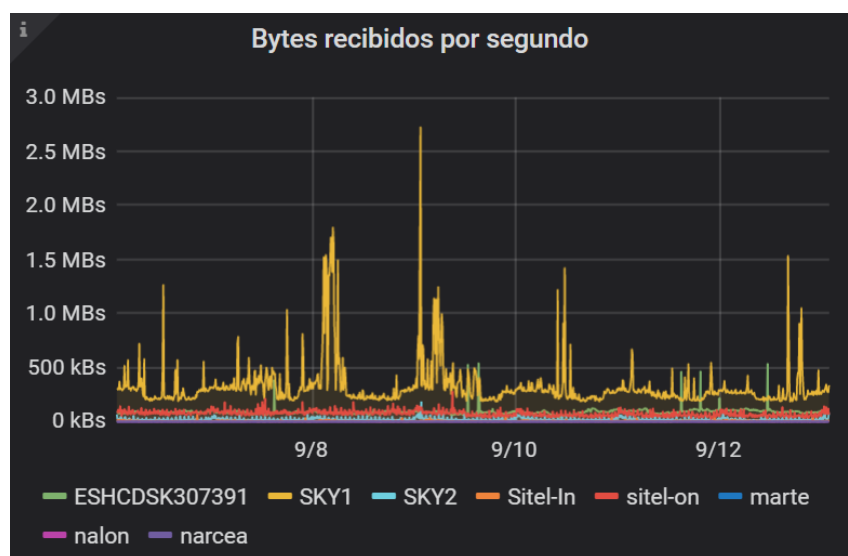


Figura 5.13 – Bytes recebidos por segundo

No seguimento dos *bytes* recebidos, são apresentados na figura 5.14 os *bytes* enviados. Uma vez que o servidor principal, tem a principal função de armazenar dados, o envio não tem grande projeção.

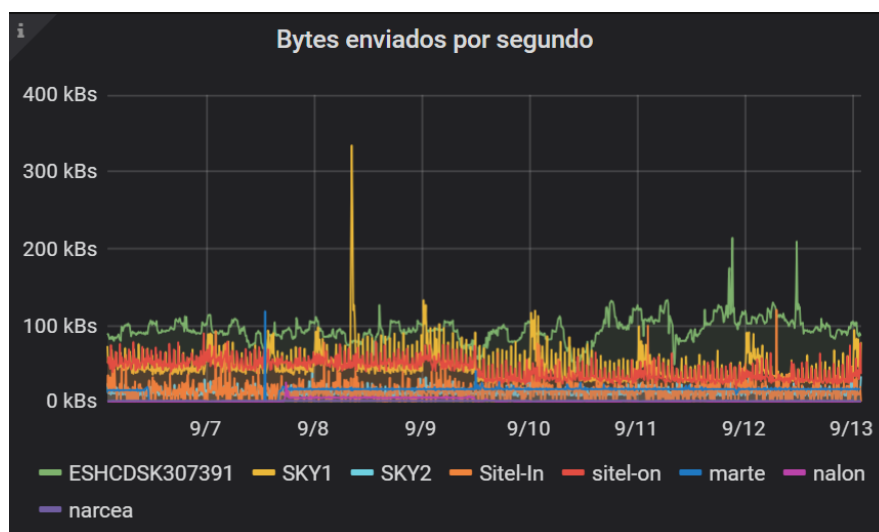


Figura 5.14 – Bytes enviados por segundo

Em questão de utilização, conclui-se segundo a imagem 5.15 que a plataforma apresenta condições para expandir em grande escala, isto porque a taxa de ocupação do *CPU* é inferior a 2.5%.

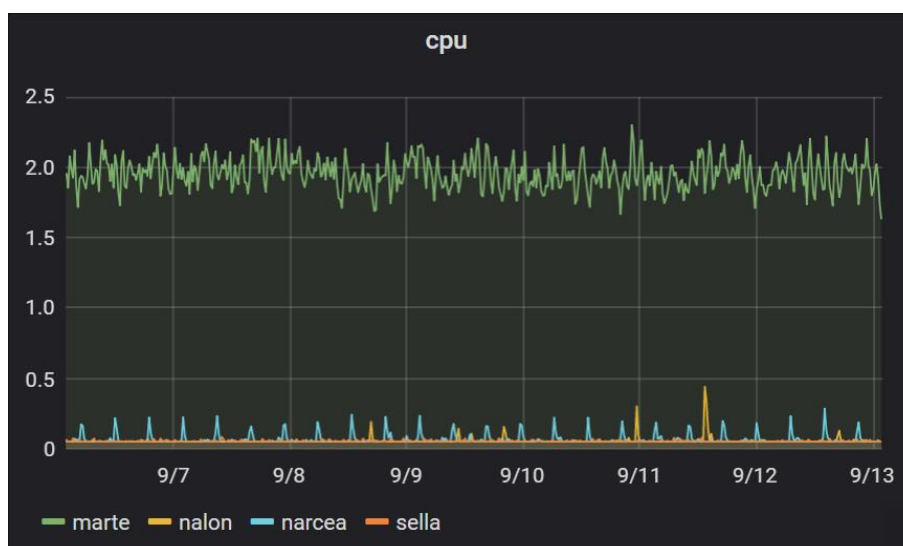


Figura 5.15 – Utilização do CPU

Na figura 5.16 são apresentadas as curvas de memória *RAM*. Adianta-se que a totalidade de memória que o servidor possui é 128GB. Da análise das curvas, conclui-se que a quantidade de *RAM* disponível constituirá um obstáculo à progressão da rede, visto, para as condições atuais, estar em uso, cerca de 40% do espaço total disponível.



Figura 5.16 – Utilização da memória RAM

5.2 Plataforma *Web de monitorização*

Após toda a infraestrutura de armazenamento e visualização em bruto estarem corretamente implementadas, estão reunidas as condições para se prosseguir com a visualização *web* da rede. Neste seguimento, começou a ser desenvolvida uma ferramenta que permite agregar dados provenientes das vias supramencionadas.

Este subcapítulo explica as estratégias adequadas para implementar um meio de visualização de dados *GIS* via *web*, interligados com curvas de carga, intervenções de mão de obra e supervisão de barramentos fora de serviço.

5.2.1 *Tigergraph*

Foi gerada uma quantidade de dados geográficos com o intuito de se recriar digitalmente um modelo de distribuição, cuja necessidade e decisão de forma de armazenamento, constituíram uma decisão importante no rumo do projeto. Inicialmente, os dados foram armazenados numa base de dados tradicional sob a forma de um vetor com 7 posições, onde consta a informação do transformador inicial, a morada física do mesmo, secção do cabo, o tamanho do segmento, posição do vértice, coordenada X e coordenada Y.

Com base nestes dados foi projetada uma solução assente em *Tigergraph*. O *Tigergraph* é uma ferramenta de análise gráfica projetada para trabalhar com dados interligados e, desta forma, conectar as várias coordenadas, segmentos e vértices da rede, criando o efeito visual apresentado na figura 5.17.



Figura 5.17 – Cidade exemplo reconstruída digitalmente a partir de dados GIS

Dada a imensidão da informação recolhida, a dimensão da rede e a necessidade de se criar uma plataforma que permitisse expandir funcionalidades ao longo do tempo, fez sentido recorrer-se ao *Tigergraph* uma vez que a arquitetura do mesmo é totalmente direcionada para a conexão de vários vértices. Perfazendo, conseqüentemente, uma teia global, com suporte direto para IoT, *BigData*, *Machine Learning* e *Parallel Processing*.

É importante entender que a criação de uma rede com milhões de pontos tornar-se-á numa outra base de dados, neste caso uma base de dados gráfica em que, todos os valores pertencentes a cada nó têm de ser armazenados. O *Tigergraph* está projetado para isso, porém, é importante que a BD seja nativa.

Destacam-se cinco pontos-chave: armazenamento, motores de armazenamento, velocidade de execução, linguagem de consulta gráfica *GSQL*, e modelos de cálculo distribuído. Iniciando pelo armazenamento, obtêm-se taxas de compressão na ordem de duas a dez vezes, mantendo uma taxa de descompressão igualmente eficiente. Os motores de armazenamento e escrita são escritos em C++, facilitando a integração no projeto. A velocidade de execução é garantida por algoritmos de *Parallel computing* permitindo lidar com gráficos de 100 milhões a mais de 100 bilhões de vértices. A linguagem de consulta gráfica *GSQL*, tem similaridade às demais linguagens abordadas ao longo deste documento, desta forma, compreendidos os conceitos *SELECT*, *FROM* e *WHERE* rapidamente é possível avançar para objetivos mais complexos. Conta ainda com modelos de cálculo distribuído que suportam os cálculos mais exigentes no menor intervalo de tempo possível.

5.2.2 Apache Kafka

De forma a interligar as curvas de carga e restantes informações da BD em *InfluxDB* foi instalado suplementarmente uma outra ferramenta denominada *Apache Kafka*.

Foi necessário incrementar-se a totalidade dos *softwares* aplicados, por duas razões. Em primeiro instante, por não existir nenhuma *API* nativa à data deste relatório em *InfluxDB* que permitisse conectar as duas tecnologias e; em segundo instante, as especificações dos servidores são diferentes, sendo inferiores no servidor *Tigergraph*. Deste modo, o *Kafka* irá funcionar como um *broker*, isto é, será um intermediário no sistema, cuja função passa por fazer o envio de informação de um servidor para o outro. Apresenta a vantagem de ser projetado totalmente para esta função permitindo assim transmitir grandes volumes de dados entre diferentes sistemas, gerindo se necessário a cadência do fluxo, garantindo a fiabilidade e integridade dos valores desde o ponto de partida até ao ponto de chegada.

5.3 Conclusão

Neste capítulo analisou-se a importância da visualização e tratamento dos dados. Foram apresentadas duas soluções de *software* para a visualização: o *Grafana* e o *Tigergraph*. A primeira, direcionada para uma visualização mais prática e relativamente rápida dos dados, enquanto, a segunda, carece de maior planeamento de dados a visualizar e quando visualizar.

Foram apresentados *screen shots* da rede em contexto real, com postos de transformação ativos, de onde se consegue visualizar com relativo grau de clareza a dimensão global da rede.

Foi ainda escolhido um Posto de Transformação da rede de testes, de forma a se conseguir exemplificar o tipo de dados recolhidos e como os mesmos são visualizados. Deste PT foram extraídos dados de potência em diferentes períodos.

A potência total da rede foi também alvo de análise, variou-se o intervalo temporal a fim de se obterem melhores resultados e, fizeram-se análises comparativas para os mesmos períodos em anos diferentes. Concluiu-se que, existe um modelo de consumo bem pautado ao longo do dia, com variações de acordo com as diferentes estações do ano.

Os servidores são permanentemente monitorizados e os *screen shots* foram expostos e analisados.

Na fase final deste capítulo foi apresentada uma solução de interligação de dois servidores, recorrendo ao *Apache Kafka*, manifestando-se a mesma uma boa solução para o problema de dois servidores com diferentes especificações de *hardware*.

6 Conclusões e Propostas de Desenvolvimentos Futuros

Este capítulo está dividido em três subcapítulos. Inicialmente, serão apresentados em suma todos os elementos conclusivos dos vários capítulos reforçando desse modo os temas tratados. Seguindo-se de algumas propostas sugestivas de abordagens futuras ao projeto, visando continuar o progresso desenvolvido até à data e, por último, apresentadas as vantagens aplicacionais do projeto em domínio empresarial.

6.1 Conclusões

É perceptível ao longo do documento o desafio constante da gestão da quantidade de informação gerada, enviada, processada e armazenada. Nesse seguimento, todo o trabalho empírico desenvolvido no projeto foi direcionado para a pesquisa e implementação de soluções cada vez mais avançadas, que encaram com eficiência as exigências das redes do futuro. Grande parte da investigação feita atualmente, constituirá alguns dos principais alicerces de toda uma infraestrutura que está a ser criada e continuará, pelas próximas dezenas de anos, melhorando e aumentando cada vez mais a dependência em energias renováveis, como apresentado nos capítulos iniciais.

O projeto realizado resultou da contribuição de um conjunto de vários elementos, dos quais se destacou um modelo que interliga a base de dados temporal *InfluxDB*, nutrida por milhares de *Smart Meters*, com a visualização em primeira instância garantida pela plataforma *Grafana*, culminando com uma ferramenta direcionada para a visualização *web* baseada em *TigerGraph*.

Numa sequência de eventos, foi necessário criar primeiramente dispositivos capazes de executar leituras precisas e bidirecionais dos consumos. Foram apresentados alguns equipamentos com estas funcionalidades, que associados a uma boa rede de transporte da informação permitiram assegurar uma ótima rede de distribuição energética, adaptando assim dinamicamente a produção ao consumo.

De modo a dar suporte aos sensores, foi igualmente crucial desenvolver métodos de armazenamento que permitissem instantaneamente processar e devolver a informação armazenada, de modo a tomar decisões assertivas nos demais setores da rede. Em virtude dos fatos mencionados, foi apresentada uma lógica assente em bases de dados temporais, cujos resultados experimentais apresentaram tempos de execução para recursos de *hardware* extremamente satisfatórios. Como foi possível depreender, em alguns casos, as comparações temporais de execução de pedidos idênticos de informação, apresentaram vantagens temporais onze vezes superiores à lógica tradicional. Por conseguinte, a interligação de milhões de dispositivos poderá avançar por um caminho sólido com pilares bem fundamentados.

Em aspetos de visualização da informação, foram também apresentadas soluções igualmente robustas e eficazes, permitindo uma vista detalhada dos dispositivos e das redes, com ferramentas comparativas e aritméticas, que permitem estimar e antecipar acontecimentos futuros bem como identificar falhas, intervenções de mão de obra e barramentos desligados.

Em suma, este estudo pretendeu dar um contributo no domínio das cidades inteligentes, apresentando estratégias para o avanço, e implementação das mesmas. Destarte, dada a

importância do tema, ainda são imensas as possibilidades a serem estudadas e implementadas, tornando a área fértil e apelativa para outras equipas de investigação.

6.2 Trabalho Futuro

Como descrito ao longo deste relatório, foram cumpridas as metas estabelecidas. Atualmente, o projeto continua em várias frentes ativas. Por um lado, estão a ser criadas e adicionadas funcionalidades de visualização gráfica com resistência das linhas, funções de seleção de transformadores e respetivos contadores por zona, adicionadas as curvas de cargas para o transformador selecionado e adicionados alertas. Por outro lado, está ainda a ser desenvolvida uma ferramenta para cálculo de trânsitos de potência em tempo real.

As smart cities serão um conceito espontâneo e natural num futuro próximo, pretendendo-se criar uma ferramenta o mais completa possível que contemple as necessidades de supervisão e gestão das mesmas.

Propõe-se que investigações futuras passem pelo contínuo aprofundamento do tema, visando a criação de infraestruturas capazes de gerir e dar respostas em tempo real à exigência dos utilizadores, garantindo um serviço de qualidade, com alicerces que permitam uma contínua expansão das tecnologias, nomeadamente na área da mobilidade e telecomunicações, como a progressiva aceitação dos veículos elétricos e surgimento do 5G.

6.3 Contributos Empresariais

Atualmente, os operadores do sistema de distribuição não dispõem da infraestrutura e do grau de digitalização necessários para implementar de uma forma drástica as mudanças exigidas pelo patamar atual da tecnologia, nem existem modelos de gestão claros que permitam aos diferentes intervenientes do sistema elétrico beneficiar da sua implementação. É necessário gerir esta mudança de forma gradual e que, não implique o desencadeamento avultado do investimento necessário na infraestrutura de distribuição, o que acabará por conduzir a um aumento do preço final dos serviços, manifestando-se diretamente nas faturas suportadas pelos consumidores finais.

Assim, o projeto ambiciona propor soluções que se ajustem à infraestrutura existente, solucionando problemas como o elevado fluxo de informação proveniente de milhares de fontes independentes. Estas considerações estão expostas nos capítulos 4 e 5, e podem servir como recomendações a seguir, mostrando o que se aferiu como sendo mais eficiente e mais eficaz comparativamente a técnicas utilizadas no presente

6.4 Papel da inteligência Artificial

As redes neuronais que servem de substrato aos diversos modelos de IA, têm uma dependência vital de dados para o correto funcionamento e refinamento de resultados obtidos. Isto é, para que uma rede neuronal consiga obter um nível confiável de aprendizagem, a quantidade e qualidade dos dados à qual o modelo é submetido, constitui um ponto extremamente forte para o sucesso das metodologias implementadas.

Com base neste documento, existe uma certa clareza e paralelismo entre os SM e os modelos de IA, isto porque, os SM são as peças responsáveis por nutrir as redes de IA. Como foi possível aferir, as métricas adquiridas e enviadas pelos SM apresentam um grau elevado de confiabilidade, fazendo pleno sentido serem utilizadas para o treino de redes neuronais.

A IA começa a nascer no seio das redes elétricas e, o seu papel será fundamental para estabilizar uma rede que cada vez mais se encontra descentralizada. Atualmente, como constatado no capítulo 2, as fontes renováveis assumem cada vez mais um papel determinante na produção de energia, cujo impacto no diagrama de cargas se encontra velozmente mais vincado. A IA, introduz a capacidade de prever a produção energética expectável destes sistemas renováveis, sobretudo ao nível dos grandes parques eólicos e solares e, com base nessas previsões tomar decisões antecipadas visando uma diminuição do desperdício energético.

Por outro lado, numa rede sem recursos de IA, apenas são passíveis de identificar os consumos totais ao longo do dia, semana e mês. As redes neuronais, com base em algoritmos específicos, permitem a identificação de assinaturas energéticas geradas por dispositivos domésticos, nomeadamente sistemas de climatização ou outra categoria de eletrodomésticos. É importante realizar-se este exercício, pois, caso o consumo específico aumente num determinado período, com uma determinada potência, poderá indiciar o arranque de um certo dispositivo em concreto. É com base nesta associação de resultados que, com recurso à IA, será possível executarem-se técnicas de desagregação de energia e identificar padrões gerados por equipamentos residenciais, culminando no traçar de perfis de consumo individuais que permitam uma maior eficiência energética e maior aproveitamento dos recursos.

Para os DSOs, estes algoritmos podem também conferir uma vantagem na deteção de anomalias e perdas, quer do ponto de vista de operação da rede, quer de perdas relacionadas com fraudes e furtos.

Não esquecer que as SC são movidas por energia, mas integram um elevado nível de dispositivos dos tipos mais variados. No topo da cadeia alimentar energética, encontram-se os veículos elétricos, uma vez que os mesmos representam um impacto forte quando iniciam o seu ciclo de carregamento. É necessário ter em mente que a introdução de um número elevado de veículos na rede, sem qualquer tipo de controlo, poderá representar um risco para a rede de distribuição. Uma vez mais, o papel da IA é determinante, sendo que, com base no perfil dos utilizadores, poderão ser desenvolvidos mecanismos controlados, que permitam o carregamento consoante o perfil e necessidade individual.

É claro que a IA apresenta um leque avultado de soluções e benefícios, todavia, operar com grandes volumes de informação acarreta sempre uma elevada capacidade computacional que dê resposta a esta necessidade, manifestando-se um impacto direto nos orçamentos propostos de implementação deste tipo de modelos. Além disso, a informação é, na grande maioria das vezes, de carácter pessoal, sendo a segurança o cume supremo da IA aplicada a este setor.

Referências

- Alaton, C., & Tounquet, F. (2020, March). *Benchmarking smart metering deployment in the EU-28*. European Commission. doi: 10.2833/492070
- Burunkaya, M., & Pars, T. (2017). *A smart meter design and implementation using ZigBee based Wireless Sensor Network in Smart Grid*. 2017 4th International Conference on Electrical and Electronic Engineering (ICEEE), 158–162. doi: 10.1109/iceee2.2017.7935812
- Casazza, J. (2010). *Understanding Electric Power Systems: An Overview of the Technology, the Marketplace, and Government Regulations* (2nd Ed.). Publisher: Wiley-IEEE Press.
- Child, M., Kemfert, C., Bogdanov, D., & Breyer, C. (2019). *Flexible electricity generation, grid exchange and storage for the transition to a 100% renewable energy system in Europe*. *Renewable Energy*, 139, 80–101. doi: 10.1016/j.renene.2019.02.077
- DB-Engines. (2020). *DB-Engines Ranking*. Retrieved from: <https://www.db-engines.com/en/ranking>
- European Commission. (Consulted in 2020). *2030 climate & energy framework*. Retrieved from: https://www.ec.europa.eu/clima/policies/strategies/2030_en
- European Commission. (Consulted in 2020). *Smart Metering deployment in the European Union*. Retrieved from: <https://www.ses.jrc.ec.europa.eu/smart-metering-deployment-european-union>.
- Gersdorf, T., Hertzke, P., Schaufuss, P. & Schenk, S. (2020). *McKinsey Electric Vehicle Index: Europe cushions a global plunge in EV sales*. Retrieved from: <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/mckinsey-electric-vehicle-index-europe-cushions-a-global-plunge-in-ev-sales>
- Grafana Requirements. (Consulted in 2020) Retrieved from: <https://www.grafana.com/docs/grafana/latest/installation/requirements>
- Harrison, C. et al. (2010). *Foundations for Smarter Cities*. *IBM Journal of Research and Development*, 54(4), pp. 1-16. doi: 10.1147/JRD.2010.2048257
- Henze, V. (2019, December 3). *Battery Pack Prices Fall As Market Ramps Up With Market Average At \$156/kWh In 2019*. Retrieved from: <https://www.about.bnef.com/blog/battery-pack-prices-fall-as-market-ramps-up-with-market-average-at-156-kwh-in-2019/>
- Hlaing, W., Thepphaeng, S., Nontaboot, V., Tangsunantham, N., Sangsuwan, T., & Pira, C. (2017). *Implementation of WiFi-based single phase smart meter for Internet of Things (IoT)*. 2017 International Electrical Engineering Congress (IEECON), 1–4. doi: 10.1109/ieecon.2017.8075793
- InfluxDB 1.8 documentation. (Consulted in 2020). Retrieved from: <https://docs.influxdata.com/influxdb/v1.8/>
- ITU. (2016). *Internet of things and smart cities and communities – Evaluation and assessment*. doi: 11.1002/1000/12627

- Kassakian, J. G., Schmalensee, R., *et al.* (2011). *The Future of the Electric Grid: An interdisciplinary MIT study*. Retrieved from: <https://www.energy.mit.edu/wp-content/uploads/2011/12/MITEI-The-Future-of-the-Electric-Grid.pdf>
- Koirala, A., Suárez-Ramón, L., Mohamed, B. & Arboleya, P. (2020). Non-synthetic European low voltage test system. *International Journal of Electrical Power & Energy Systems*, 118. doi: 10.1016/j.ijepes.2019.105712.
- Marques, G. S., Dias, M. A. P. & Vianna, J. N. S. (2020). Innovation in the electricity sector in the age of Disruptive Technologies and renewable Energy Sources: A Bibliometric study from 1991 to 2019. *International Journal of Advanced Engineering Research and Science (IJAERS)*. 7(2), 261-272. doi: 10.22161/ijaers.72.35
- Majumder, A., & Caffery, J. (2004). Power line communications: an overview. *IEEE Potentials*, 23(4), 4–13. doi: 10.1109/mp.2004.1343222
- Naqvi, S. N. Z. & Yfantidou, S. (2017). *Time Series Databases and InfluxDB*. University “Université Libre de Bruxelles”.
- Nasar, M. and Kausar, M. A. (2019). Suitability Of Influxdb Database For IoT Applications. *International Journal of Innovative Technology and Exploring Engineering*. 10(8). doi: 10.35940/ijitee.J9225.0881019
- Nithin.S, Radhika, N., & Vanitha, V. (2012). SMART GRID TEST BED BASED ON GSM. *Procedia Engineering*, 30, 258–265. doi: 10.1016/j.proeng.2012.01.859
- Popescu, G. H., Mieila, M., Nica, E. & Andrei, J. V. (2018). The emergence of the effects and determinants of the energy paradigm changes on European Union economy. *Renewable and Sustainable Energy Reviews*, 81, 768–774. doi: 10.1016/j.rser.2017.08.055
- Ritchie, H. (2017). Renewable Energy. Retrieved from: <https://ourworldindata.org/renewable-energy>
- Smil, V. (2017). *Energy and Civilization: A History*. Publisher: The MIT Press.
- Styczynski, Z. A. & Buchholz, M. B. (2014). *Fundamentals and Technologies in Electricity Networks*. Publisher: Springer Vieweg
- Weranga, K., Kumarawadu, S., & Chandima, D. P. (2013). Evolution of Electricity Meters. *Smart Metering Design and Applications*, 17–38. doi: 10.1007/978-981-4451-82-6_2
- Zhao, Y., Liang, Z., Ma, X., & Li, Y. (2017). The near-field intelligent meter reading system based on Bluetooth — Broadband carrier. 2017 Chinese Automation Congress (CAC), 4519–4523. doi: 10.1109/cac.2017.8243576

Anexo A Código em Python

```
1. # -*- coding: utf-8 -*-
2. """
3.
4. @author: alcar
5. """
6.
7. # Importação de bibliotecas
8. import pandas as pd #pandas para csv
9. from sys import getsizeof # getsizeof para tamanho em memoria RAM
10. from datetime import datetime # trabalhar com datas e tempo
11. from tqdm import tqdm # para percent bar
12.
13.
14. from sys import getsizeof
15. import random
16. import math
17. from influxdb import InfluxDBClient
18. from tqdm import tqdm
19. import mysql.connector
20. from timeit import default_timer as timer
21.
22.
23.
24. def influxdb_connection():
25.
26.     host = '192.168.1.91'
27.     port = 8086
28.
29.     try:
30.         client = InfluxDBClient(host, port)
31.         return client
32.     except:
33.         print("Erro na tentativa de conexão ao host:",host)
34.         return 0
35.
36. def influxdb_sendData():
37.
38.     client = influxdb_connection()
39.     data =
40.     pd.read_csv("Curvas_Pola_Siero_201901.csv",delimiter=";")
41.
42.     # Ver as primeiras n linhas da variável
43.     #data.head(2)
44.
45.     # tamanho da variável
46.     #size = getsizeof(data)
47.     #print(size)
48.     i = 5000000
49.
50.     for i in tqdm(range(i, 5100000)):
51.
52.         #Obter grandezas para guardar em BD
53.         REFERENCE = str(data.loc[i, '#REFERENCIA'])
54.         STATION = str(data.loc[i, 'ESTACION'])
55.         P_IN = float(data.loc[i, 'ACTIVA_E'].replace(',','.'))
56.         P_OUT = float(data.loc[i, 'ACTIVA_S'].replace(',','.'))
57.         Q_1 = float(data.loc[i, 'REACTIVA1'].replace(',','.'))
58.         Q_2 = float(data.loc[i, 'REACTIVA2'].replace(',','.'))
59.         Q_3 = float(data.loc[i, 'REACTIVA3'].replace(',','.'))
```

```

59.     Q_4 = float(data.loc[i, 'REACTIVA4'].replace(',','.'))
60.     STAGE = data.loc[i, 'ESTADO']
61.
62.     #Obter o tempo:
63.     date = data.loc[i, 'FECHA']
64.     #converter a data do ficheiro csv para ms:
65.     dt_obj = (datetime.strptime(date, '%d/%m/%Y %H:%M:%S'))
66.     value_ns = int(dt_obj.timestamp() * 1000000000)
67.
68.     time = (value_ns + i)
69.
70.     json_body = [
71.         {
72.             "measurement": "values",
73.             "time": time,
74.             "fields": {
75.                 "REFERENCE": REFERENCE,
76.                 "STATION": STATION,
77.                 "P_IN": P_IN,
78.                 "P_OUT": P_OUT,
79.                 "Q_1": Q_1,
80.                 "Q_2": Q_2,
81.                 "Q_3": Q_3,
82.                 "Q_4": Q_4,
83.                 "STAGE": STAGE,
84.             }
85.         }
86.     ]
87.     client.write_points(json_body, database='meters',
88. time_precision='n')
89.     i += 1
90.
91. def mysql_connection():
92.
93.     try:
94.         mydb = mysql.connector.connect(
95.             host="192.168.1.92",
96.             user="sql",
97.             password="123",
98.             database="values_m"
99.         )
100.        print(mydb)
101.        return mydb
102.    except:
103.        print("Erro na tentativa de conexão com o Mysql")
104.        return 0
105.
106. def mysql_sendData():
107.
108.     data =
109.         pd.read_csv("Curvas_Pola_Siero_201901.csv", delimiter=";")
110.
111.     # Ver as primeiras n linhas da variável
112.     #data.head(2)
113.
114.     # tamanho da variável
115.     #size = getsizeof(data)
116.     #print(size)
117.     i = 0
118.     mydb = mysql_connection()
119.     mycursor = mydb.cursor()
120.

```

```

121.     for i in tqdm(range(i, 1000000)):
122.
123.         #Obter grandezas para guardar em BD
124.         REFERENCE = str(data.loc[i, '#REFERENCIA'])
125.         STATION = str(data.loc[i, 'ESTACION'])
126.         P_IN = float(data.loc[i, 'ACTIVA_E'].replace(',','.'))
127.         P_OUT = float(data.loc[i, 'ACTIVA_S'].replace(',','.'))
128.         Q_1 = float(data.loc[i, 'REACTIVA1'].replace(',','.'))
129.         Q_2 = float(data.loc[i, 'REACTIVA2'].replace(',','.'))
130.         Q_3 = float(data.loc[i, 'REACTIVA3'].replace(',','.'))
131.         Q_4 = float(data.loc[i, 'REACTIVA4'].replace(',','.'))
132.         STAGE = data.loc[i, 'ESTADO']
133.
134.         #Obter o tempo:
135.         date = data.loc[i, 'FECHA']
136.         #converter a data do ficheiro csv para ms:
137.         dt_obj = (datetime.strptime(date, '%d/%m/%Y %H:%M:%S'))
138.         value_ns = int(dt_obj.timestamp() * 1000000000)
139.
140.         time = value_ns
141.
142.         sql = "INSERT INTO meters (DATE_TIME, REFERENCE_M, STATION,
143.         P_IN, P_OUT, Q_1, Q_2, Q_3, Q_4, STAGE) VALUES (%s, %s, %s, %s, %s,
144.         %s, %s, %s, %s, %s)"
145.         val =
146.         (time,REFERENCE,STATION,P_IN,P_OUT,Q_1,Q_2,Q_3,Q_4,STAGE)
147.         mycursor.execute(sql, val)
148.
149.         mydb.commit()
150.         i += 1
151.
152. class Test():
153.
154.     def __init__(self):
155.         self.client = influxdb_connection()
156.         self.mydb = mysql_connection()
157.         self.mycursor = self.mydb.cursor()
158.
159.         #----- InfluxDB-----
160.         def influx(self):
161.             """
162.             função para extrair n valores limitados por LIMIT
163.             """
164.             query = 'SELECT mean("P_IN") FROM(SELECT ("P_IN") AS "P_IN"
165.             FROM "meters"."autogen"."values" GROUP BY time(1h) FILL(null)'
166.             SELECT = self.client.query(query)
167.             return SELECT
168.
169.         #-----MySQL-----
170.         def mysql(self):
171.             self.mycursor.execute("SELECT AVG(`P_IN`), Date(DATE_TIME)
172.             FROM `measurements` WHERE `DATE_TIME` BETWEEN '2019-01-01 00:00:00'
173.             AND '2019-01-30 00:00:00' GROUP BY Hour(DATE_TIME)")
174.             myresult = self.mycursor.fetchall()
175.             return myresult
176.
177.         if __name__ == "__main__":
178.             start = timer()
179.             test = Test()

```

Anexo A

```
178. #-----
179. result = test.influx()
180. #-----
181. end = timer()
182. print()
183. print("Tempo decorrido [s]= ", end - start)
184.
185.
186.
187.
```