Universidade de Aveiro
2015

Departamento de Electrónica, Telecomunicações e Informática

Marco Castro Pereira

Análise de Soluções “Big Data” para Monitorização de Consumos de Água

Universidade de Aveiro
2015
Departamento de Electrónica, Telecomunicações e Informática

Marco Castro Pereira

Análise de Soluções “Big Data” para Monitorização de Consumos de Água

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre no Mestrado em Sistemas de Informação, realizada sob a orientação científica do Doutor Diogo Nuno Pereira Gomes, Professor do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri / the jury

presidente / president	**Prof. Doutor Luís Seabra Lopes** Professor Associado da Universidade de Aveiro
vogais / examiners comittee	**Mestre Ricardo Azevedo Guerra Raposo Pereira** Especialista da Critical Software (Arguente Principal)
	Prof. Doutor Diogo Nuno Pereira Gomes Professor Auxiliar da Universidade de Aveiro (Orientador)

agradecimentos

Aproveito para agradecer, em primeiro lugar ao Professor Diogo Nuno Pereira Gomes, por todo o apoio e ajuda dada ao longo da elaboração desta dissertação.

Agradeço ainda ao Eng. Fernando Santiago pelo apoio e a todos os elementos do grupo ATNoG pela disponibilizaçaão de informação e recursos.

Por fim, agradeço aos meus pais, por tudo.

palavras-chave

Máquina-a-Máquina, Internet das Coisas, Smart Metering, Big Data, Armazenar, Processar, Analisar.

resumo

O elevado crescimento das comunicações sem fios, tem provocado um aumento no número de dispositivos que impulsionaram o desenvolvimento de novas áreas, como o Smart Metering. Estes avanços, por sua vez, proporcionaram um aumento da quantidade de informação que flui entre máquinas, o que implica que sejam necessárias novas medidas para efectuar a gestão de toda esta informação.

O presente trabalho pretende fornecer métodos e técnicas capazes de efectuar o armazenamento e processamento de grandes quantidades de dados. Desta forma, numa primeira instância será fornecida uma análise teórica sobre a área e as principais metodologias para construção de soluções de Big Data. Posteriormente, será apresentada toda a informação dos dados envolvidos, e dos métodos propostos para o desenvolvimento da solução. Por fim serão apresentados os resultados práticos da solução e efectuada uma avaliação da solução implementada.

keywords Machine-to-Machine, Internet of Things, Smart Metering, Big Data, Store, Process, Analyze

abstract High wireless communications growth leads to an increased number of devices that boosted the development of new areas, as the Smart Metering.
In turn, this developements provided an increase data flowing between machines, that requires new actions to manage this data amount.
This present MSc work pretends to provide methods ans techniques able to store and process high data amounts.
Thus, in the first instance it is provided a theorical analysis about the area and the principals methodologies to develop Big Data solutions. Then, it will be presented all information of the involved data and the proposed methods to develop the solution. Lastly it will presented the pratical results and an evaluation of the implemented solution.

Conteúdo

Índice de Figuras

Índice de Tabelas

Acrónimos

Notação	Descrição
M2M	Machine-to-Machine
IoT	Internet of Things
AMR	Automatized Meter Read
AMI	Advanced Metering Infrastructure
PLC	Power Line Communications
GSM	Global System for Mobile
NoSQL	Not only SQL
HDFS	Hadoop Distributed File System
RDBMS	Relational Database Management System
CQL	Cassandra Query Language
JSON	JavaScript Object Notation
XML	Extensible Markup Language
POSIX	Portable Operating System Interface
API	Application Progamming Interface
BJSON	Binary JavaScript Object Notation
YARN	Yet Another Resource Negotiator
CEP	Complex Event Processing
ESP	Event Stream Processing
RFID	Radio-Frequency Identification
RDD	Resilient Distributed Dataset
INE	Instituto Nacional de Estatística
CTT	Correios, Telégrafos e Telefones
AdRA	Águas da Região de Aveiro
SQL	Structured Query Language
AMQP	Advanced Message Queuing Protocol
LSH	Locality-Sensity Hashing

1 Introdução

O elevado crescimento das comunicações, especialmente na área das tecnologias sem fio, tem provocado um aumento exponencial do número de pequenos dispositivos, como smartphones/tablets, computadores e acessórios do dia-a-dia (por exemplo televisores e fechaduras), com capacidade, de não só estarem ligados entre si, numa rede de dispositivos, mas também com capacidade de se ligarem à Internet, numa rede de redes. Este crescimento, em conjunto com a evolução tecnológica de diversas áreas, como a dos sensores, foram os principais agentes no impulso da Internet das Coisas, e por consequente, das comunicações máquina-a-máquina.

O uso de comunicações máquina-a-máquina entre este tipo de dispositivos, significa que estes são capazes de interagir entre si, sem a intervenção humana, o que levou a um aumento da quantidade de informação que flui nestas redes, denominadas de redes do futuro. Por exemplo, no caso dos sensores, a informação é extraída periodicamente a partir do ambiente onde estes se encontram e enviada através da rede para exploração futura. A estes sensores foi atribuído o nome de sensores inteligentes.

Por forma a obter uma melhor compreensão da quantidade de informação que estas redes terão de suportar, em [1] e [2], afirma-se que apesar, de em geral cada dispositivo, não enviar informação de forma corrente para a rede, e a informação enviada ser relativamente pequena, o facto de puderem existir milhares de dispositivos ligados a essas redes, a enviar informação periodicamente, leva a que as redes sejam atravessadas por grandes fluxos de informação. Por exemplo, supondo que cada habitação em Portugal é dotada de um sensor inteligente capaz de medir o consumo de água, e em que são efectuadas medições com um intervalo de tempo médio de cinco minutos, facilmente nos apercebemos da quantidade gigantesca de informação, que flui numa destas redes por dia.

Esta tem vindo a ser cada vez mais uma realidade, uma vez que os sensores inteligentes estão a ser implementados em diversas áreas de negócio, por forma a recolher informação dos seus activos, como é o caso da indústria da água. Os sensores inteligentes na indústria da água permitem medir propriedades, como a qualidade da água, condições dos aquedutos e consumo de água. Por exemplo, sensores inteligentes de uso doméstico, podem ser utilizados para registar o uso de água por hora dos consumidores finais, através dos dados de séries temporais recolhidos automaticamente e continuamente. Os dados dos sensores inteligentes

podem ser utilizados para preparar a emissão das facturas de água, da mesma forma, que actualmente as leituras dos contadores são efectuadas.

No entanto, o foco desta tecnologia, tem-se centrado na possibilidade de melhorar o uso eficiente da água através de uma melhor evidência para a tomada de decisão. A extracção de informação valiosa a partir dos dados fornecidos pelos sensores inteligentes e a partilha desta informação entre empresas e clientes é conhecida como Smart Metering.

Diversos estudos de custo-beneficio detalharam o potencial de negócio associado ao Smart Metering na indústria da água [3] [4].

Os contadores inteligentes permitem aos consumidores de água, obter informação de como e quando a água é utilizada, proporcionando-lhes a possibilidade de poupar água e assim reduzir os custos do seu uso. Permite também fornecer informação para apoio à decisão da facturação de água e investimento futuro dos fornecedores de água.

O Smart Metering possui também benefícios ao nível ambiental, como a conservação da água e redução de emissões de carbono, que contribuem para resiliência do clima. Um benefício operacional adicional significativo do Smart Metering é o baixo custo de manutenção com o pessoal em comparação com os contadores de água tradicionais.

No entanto, embora os benefícios do Smart Metering sejam bem evidentes, técnicas para lidar com os Smart Meterings são ainda uma emergente, nomeadamente no que toca à gestão (armazenamento e processamento) da quantidade de informação que estes envolvem.

Como já referido anteriormente, a quantidade de informação existente nas comunicações entre máquinas, é o principal foco do problema de desenvolvimento de soluções no contexto que envolve o Smart Metering. O desenvolvimento destas soluções “Big Data”, implicam a análise de grandes quantidades de dados, provenientes de diversas fontes, a uma alta velocidade, requerem a utilização de técnicas especificas para lidar com os dados, para que o seu armazenamento e processamento seja efectuado de forma eficiente.

Desta forma, dada a importância da análise “Big Data”, inúmeros estudos e desenvolvimentos têm sido efectuados com o intuito de criar uma solução capaz de lidar com os dados gerados pelos sensores inteligentes. No entanto, embora existam diversos métodos propostos, em áreas como a energia eléctrica ou circulação rodoviária, esta é uma área ainda em crescente exploração e desenvolvimento, onde surgem cada vez mais novas áreas de aplicação, que necessitam de conhecimento e medidas especificas, pelo que ainda não existem soluções usadas globalmente de forma transparente.

1.1 Motivação

O paradigma das comunicações entre máquinas (M2M) é visto como um dos grandes impulsionadores da inovação durante os próximos anos. Através deste paradigma é possível alcançar a visão da Internet das

Coisas (IoT), onde os objectos com que interagimos no quotidiano e o ambiente que nos rodeia se tornam mais inteligentes. O enorme crescimento da integração de sensores digitais, por exemplo, para monitorização de consumo energético, localização remota ou monitorização intensiva de diagnósticos médicos, permite efectuar a recolha de dados do nosso ambiente, das nossas interacções e até mesmo de nós próprios.

Através da integração da informação, em sistemas capazes de lidar com grandes quantidades de dados (BigData), torna-se possível obter um melhor conhecimento do nosso mundo, e consequentemente agir de forma mais informada, na tomada de decisões, melhorando o conforto ou até mesmo a produtividade em diversas áreas.

Devido à importância deste tópico, muitos operadores de telecomunicações estão a desenvolver, ou já possuem soluções de gestão e comunicação de equipamentos M2M. Neste âmbito, a PTIn tem desenvolvido soluções na área do Smart Metering, para recolha de dados de contadores de água inteligentes. Com o surgir destas novas soluções, a PTIn lançou-me o desafio, com o intuito de criar um sistema capaz de efectuar a gestão dos dados recolhidos de contadores de água inteligentes. Deste modo, pretende-se que seja desenvolvido um sistema capaz de efectuar a gestão de informação, bem como a aquisição de conhecimento apartir dos dados, por forma a que seja possível oferecer aos seus clientes informação sobre os seus consumos de água, permitindo assim o auxilio para a tomada de decisões relativas ao mesmo.

O desenvolvimento de componentes de solução “Big Data” para o contexto especifico do Smart Metering, assim como a aplicação de algoritmos de aprendizagem, como por exemplo, análise estatística e previsão, são fundamentais para dar o próximo passo na inovação do paradigma de comunicação entre máquinas, uma vez que estes possibilitam a transformação de dados em informação, e informação em conhecimento, o que em conjunto com soluções de gestão eficiente dos dados, permite o rápido acesso à informação.

Deste modo, torna-se evidente que é necessário efectuar uma investigação na área, com o intuito de definir soluções capazes de lidar com grandes quantidades de dados e que permitam a monitorização de consumos de água.

1.2 Objectivos

O objectivo principal deste projecto é o desenvolvimento de componentes de solução “Big Data” para o contexto especifico de comunicações máquina-a-máquina e Internet das Coisas. Como tal pretende-se que seja implementado um protótipo “End-to-End” centrado na área do Smart Metering, mais especificamente na sub-área Smart Metering Waters.

Deste modo, os objectivos deste projecto passam pela avaliação e implementação de métodos para o armazenamento de grandes quantidades de dados, bem como a implementação de métodos para processamento de informação, capazes de efectuar uma análise de grandes quantidades de informação, através de algoritmos de machine learning que permitam obter conhecimento sobre os dados.

Pretende-se também que sejam desenvolvidos componentes que forneçam serviços para acesso à informação

de alta performance, que sejam ao mesmo tempo fiáveis e flexíveis.

Resumidamente, os principais objectivos deste projecto passam pela:

- implementação de métodos para armazenamento de grandes quantidades de dados
- implementação de métodos de acesso à informação
- implementação de algoritmos capazes de obter conhecimento a partir dos dados
- visualização da informação

1.3 Estrutura do Documento

Este documento encontra-se organizado em 6 capítulos. Tendo em conta, que o primeiro capítulo é a introdução, onde foi feita uma apresentação do tema da dissertação e motivação para a sua realização, assim como a descrição dos seus principais objectivos, os restantes capítulos são:

- Capítulo 2: fornece uma descrição do estado da arte. São apresentados os principais conceitos de Smart Metering, Internet das Coisas, M2M e Big Data Analytics. Além disso, também é feita uma análise ás ferramentas e metodologias para armazenamento e processamento de dados.
- Capítulo 3: fornece uma descrição em relação os dados utilizados, nomeadamente em relação à sua aquisição, e estrutura, bem como as técnicas de exploração e modelação aplicadas sobre os dados.
- Capítulo 4: fornece uma descrição da solução proposta, nomeadamente a descrição detalhada de todo o processo de implementação, a arquitectura da solução, principais métodos implementados e limitações da mesma.
- Capítulo 5: fornece uma descrição dos principais testes efectuados, bem como são apresentados os resultados da implementação da solução.
- Capítulo 6: fornece uma descrição das principais conclusões do trabalho realizado, assim como os possíveis trabalhos futuros a serem explorados.

2 Estado da Arte

2.1 Internet das coisas

Com o surgir de dispositivos de pequeno porte que podem ser conectados à Internet, como sensores e outros dispositivos do quotidiano, como televisores, chaves de portas, ou até mesmo carros, em conjunto com outros dispositivos que têm, ao longo do tempo, feito parte assídua da Internet, como computadores, impressoras, smart-phones, estima-se que o número de máquinas conectadas à Internet irá exponencialmente aumentar nos próximos anos, com a possibilidade de chegar às dezenas de milhares de milhões de máquinas conectadas [5]. Portanto, a divulgação pelo mundo de todos os sistemas e dispositivos mencionados anteriormente, bem como a sua integração na sociedade, deram origem ao conceito de Internet das Coisas.

Embora o conceito não tenha sido denominado até 1999 [6], a Internet das Coisas está em desenvolvimento há décadas. O primeiro aparelho de Internet, por exemplo, era uma máquina de Coca-Cola no Carnegie Melon University no início de 1980 [60]. Os programadores poderiam conectar-se à máquina através da Internet, verificar o estado da máquina e determinar se existia ou não, uma bebida fresca disponível, e nesse caso decidir fazer a viagem até a máquina.

A Internet das Coisas apesar de ser baseada num paradigma, onde a computação aparece em toda a parte e em qualquer lugar, assim como a computação ubíqua, difere na medida em que a IoT foca-se, em como a informação é gerada.

Ao contrário do que tem acontecido até agora nas redes existentes, onde a informação existente é fornecida principalmente por seres humanos, nas chamadas redes do futuro, as comunicações M2M serão usadas em grande escala, o que significa que a informação é gerada e transmitida a partir das máquinas, sem a necessidade da intervenção humana. Esta automatização permitirá não só uma maior fiabilidade da informação, uma vez que esta é obtida directamente a partir de dispositivos que estão a interagir com o mundo real (sensores), mas também permitirá a obtenção de informação em “tempo-real” ou quase, uma vez que esta não tem de ser actualizada por seres humanos que retardam o seu processo.

No entanto, a investigação na área da IoT, ainda se encontra numa fase inicial. Os vários nomes dados a IoT, como Internet of Everything, Internet do Futuro ou Internet das Coisas, mostram que ainda não foi alcançada nenhuma definição “final”. Apesar disso, de acordo com a ITU-T [7], IoT é definida como "A infra-estrutura

global para a informação da sociedade, permitindo serviços avançados através da interligação de coisas (físicas e virtuais), com base na informação inter-operável e tecnologias de comunicação já existente e em evolução".

A Internet das Coisas (IoT) é a interconexão de dispositivos informáticos identificáveis, incorporados exclusivamente dentro da infra-estrutura de Internet existente. Normalmente, quando se fala de IoT, espera-se que esta ofereça uma conectividade avançada de dispositivos, sistemas e serviços, que vai além das comunicações máquina-a-máquina (M2M) e contemple uma variedade de protocolos, domínios e aplicações. A interligação desses dispositivos(incluindo objectos inteligentes), está prevista para inaugurar a automação em quase todos os campos, ao mesmo tempo, permitindo a criação de aplicações avançadas, como as Smart Grids [8].

Além da infinidade de novas áreas de aplicação para expandir a Internet ligada à automação, também se espera que a IoT lide com grandes quantidades de dados provenientes de diversos locais agregados e a uma velocidade muito alta, aumentando assim a necessidade de melhorar a indexação, armazenamento e processo de tais dados.

O conceito de Internet das coisas e os números apresentados no primeiro parágrafo deste sub-capitulo mostram que a Internet de hoje, como sabemos, está em mudança, e no futuro será composta por inúmeras redes, numa quantidade muito superior à actual da Internet, e compostas por um maior número de dispositivos e sistemas de diferentes dimensões e áreas. Todos os dispositivos e sistemas, organizados em redes, permitirão uma maior e melhor interacção com o mundo real, e não apenas por causa da capacidade de detecção e actuação fornecidas pelos sensores, mas também porque as coisas estão espalhadas e incorporadas no nosso quotidiano, o que nos permitirá tirar partido da informação presente nas redes para benefício próprio.

2.1.1 Máquina-a-Máquina

O termo M2M e o termo IoT são vulgarmente utilizados em conjunto para descrever a próxima geração da Internet. No entanto, o facto de que a Internet das coisas é mais associado com M2M, e o termo M2M ser usado de forma ampla e livre para designar vários cenários diferentes, pode causar uma confusão compreensível sobre a relação entre estes dois conceitos.

O conceito de comunicação máquina-a-máquina não é novo, ou seja, a comunicação entre máquinas sem a intervenção humana tem vindo a acontecer ao longo do tempo, sendo um mercado consistente, que permite reduzir custos, tempo e melhorar a eficiência. A comunicação entre máquinas, como routers e servidores, ou sensores e máquinas, são um caso de uso comum, por exemplo para a conservação de energia e monitorização de maquinaria industrial.

No entanto, o despertar por de trás do IoT, e o uso deste termo para descrever as comunicações subjacentes, tem provocado, uma evolução do termo. Assim, apesar de o termo significar literalmente Máquina-a-Máquina, o que significa, no contexto da Internet das coisas, é que as máquinas não só são capazes de conversar entre si, sem intervenção humana, como tem acontecido até agora, mas também são capazes de dar

sentido à informação recebida, a fim de tomar acções automatizadas.

Para resumir, M2M pode ser visto como sendo um subconjunto de IoT, no sentido de que é o M2M que proporciona a conectividade que permite tornar possível as capacidades IoT.

Sem tecnologias que permitam comunicações M2M, ou seja, se as máquinas não fossem capazes de avaliar as informações recebidas, a Internet das coisas não seria possível.

2.2 Smart Metering

Smart metering é sem dúvida uma área que recentemente tem atraído muita atenção. Muitos países na Europa e fora da Europa estão envolvidos em projectos relacionados com o Smart Metering, numa demonstração de larga escala. O conceito de Smart Metering está intrinsecamente ligado ao surgimento de uma nova área, denominada, de Smart Cities.

Uma "Cidade" pode ser definida como "Inteligente", quando os investimentos, quer sejam em capital humano e social ou em infra-estruturas de comunicação, possibilitam o desenvolvimento económico sustentável e uma elevada qualidade de vida, através de uma gestão sensata dos recursos naturais e administração participativa. A seguinte figura ilustra os principais elementos envolvidos no desenvolvimento de um Smart City.

Figura 1: Principais elementos de uma Smart City

Uma Smart City baseia-se na troca de informações entre os seus diferentes subsistemas, através da utilização de tecnologias de informação, capazes de efectuar uma análise em tempo-real, que possibilitam o seu desenvolvimento económico.

Muitas questões surgem quando se aborda o conceito de Smart Metering, nomeadamente se, os smart meters

se concentram apenas para o uso na energia eléctrica? Quais as funções exactas dos smart meters e os seus benefícios? Qual é o estado da tecnologia? Que projectos smart metering estão a ser desenvolvidos neste momento? Se existe algum obstáculo para a sua implementação e qual a natureza desses obstáculos (técnicos,económicos, organizacionais)? Qual o futuro previsto e qual o impacto do smart metering?

Nesta secção irá ser efectuada uma abordagem, de forma resumida destas questões principais, e fornecida uma visão sobre a situação e futuro do smart metering.

Smart Metering geralmente envolve a instalação de um smart meter nas residências dos clientes, leituras regulares, processamento e feedback dos dados de consumo para o cliente.

Os smart meters são medidores de serviços públicos (energia eléctrica, gás, água, temperatura), que podem ser utilizados para estimar a facturação de um determinado serviço e efectuar leituras do medidor, e assim como fornecer aos clientes e distribuidores de energia informações valiosas sobre os serviços.

Um smart meter tem as seguintes capacidades:

- registo em tempo real, ou quase dos consumos efectuados ou da energia gerada localmente
- permite a possibilidade de ler o medidor localmente ou remotamente
- limitação remota do caudal através do medidor(ex: corte do fornecimento de água)
- interligação com redes e dispositivos
- capacidade de ler diversos tipos de informação, como água, gás, entre outros

Inicialmente os smart meters foram considerados apenas para o registo de consumo de energia eléctrica e gás, no entanto, devido ao seu desenvolvimento, o registo do consumo de água é actualmente também uma possibilidade.

Smart metering é muitas vezes referido como a leitura automatizada do medidor (Automatized meter read (AMR)); no caso de tempo real, ou comunicação bidirecional, como infra-estrutura de medição avançada (Advanced Metering Infrastructure (AMI)).

Devido ao seu recente aparecimento, o futuro do smart metering é uma questão muito abordada, sendo posta em causa a sua viabilidade, uma vez que para muitos ainda não é possível afirmar, se esta não passa de uma campanha publicitária que irá terminar após o desenvolvimento de alguns projectos ou se, se irá tornar a tecnologia da próxima década. Os smart meters são sem dúvida um desenvolvimento inovador e indispensável para todos os intervenientes no mercado:

- para empresas de medição, com o objectivo de diminuir os custos de leitura dos contadores
- para operadores de rede, que querem preparar a sua rede para o futuro
- para fornecedores de energia, gás, água que querem introduzir novos serviços para o cliente e reduzir os custos de chamadas centrais
- para os governos, que pretendem alcançar objectivos de poupança e eficiência energética e melhorar os processos de mercado
- para clientes finais, nomeadamente no aumento da consciência energética e diminuição do consumo, e respectivos custos de energia eléctrica, gás e água

A introdução de smart metering é um passo lógico, que tem vindo a ser dado, quer seja na Europa ou fora

dela, uma vez que nos dias que correm, toda a comunicação é digitalizada e padronizada e os custos da "inteligência digital" estão em corrente decréscimo.

2.2.1 Projectos Actuais

Esta secção contém uma visão geral de alguns projectos de smart metering a decorrer na Europa, mais precisamente em Itália, Bulgária, Suécia e Portugal.
Na Itália, a empresa ENEL introduziu o smart metering já desde 2001, no seu projecto "Telegstore" [9]. Antes da desregulamentação do mercado de energia, a ENEL tomou a decisão de investir na introdução de smart meters, sendo a primeira empresa em tudo mundo introduzir os smart meters. As razões principais da ENEL foram as poupanças esperadas, receitas na áreas de compras e logística, operações de campo, serviços ao cliente e protecção de receita. A ENEL optou pela implantação de um smart meter ligado à energia eléctrica, que comunica através de PLC (Power Line Communications) para a sub-estação mais próxima. De seguida, os dados são lidos através de GSM. Até 2014, já foram instalados 32 milhões de smart meters, sendo estes na sua maioria geridos remotamente.

Na Bulgária, a empresa EVN Bulgaria, iniciou em 2007 o desenvolvimento do projecto piloto Smart Metering in Risk Areas. O projecto foca-se sobre as regiões com altas perdas e baixas percentuais de recolha de energia. A actor principal do projecto é a empresa de distribuição de energia "EVN Bulgaria Elektrorazpredelenie EAD", com a cooperação de empresas externas usadas para a reconstrução de rede e instalação de meters. O projecto envolve também a utilização de operadores de serviços de comunicação móvel e até ao momento já foram instalados 67440 meters.

Na Suécia, na área da cidade Sundsvall, a empresa Sundsvall elnät AB, iniciou em 2009 um projecto "Roll Out" denominado por, Smart Grid Sundsvall elnät [10]. O projecto inclui a comunicação com todos os clientes, capacidade de recuperar dados de hora a hora, qualidade de energia, falta de energia e uso de um interruptor interno em todos os meters de energia eléctrica.
Também tem instalado um instrumento de qualidade de energia e um sensor de curto-circuito nas subestações(MV/LV) e pode ser efectuada a leitura de dados no sistema de monitorização e usá-la para a resolução de problemas.

Também em Portugal, em Outubro de 2007, a EDP anunciou o projecto InovGrid, que tinha como principal objectivo atender a grandes mudanças que se perspectivavam no sector eléctrico. O foco do projecto InovGrid centra-se no conceito das redes inteligentes, com impacto directo ao nível da rede de distribuição, na qual se pretende reforçar os activos de gestão remota inteligente, dotando a mesma de capacidade de geração dispersa.
O projecto InovGrid, desenvolvido pela EDP Distribuição, com o apoio de parceiros nacionais de produção industrial, de tecnologia e investigação (EDP Inovação; Lógica; Inesc Porto; Efacec; Janz e Contar), visa

dotar a rede eléctrica de informação e equipamentos capazes de automatizar a gestão das redes, melhorar a qualidade de serviço, diminuir os custos de operação, promover a eficiência energética e a sustentabilidade ambiental e potenciar a penetração das energias renováveis.

Em 2009, no âmbito de projecto InovGrid [11] [12], a EDP deu inicio à instalação do primeiro conjunto de dispositivos G Smart, com a instalação de 390 unidades. A função principal do G Smart é o controlo de postos de transformação MT/BT, bem como a concentração da informação das contagens. Com uma média de 100 contadores por unidade, mas com a capacidade de integrar até 1000 contadores, os controladores G Smart integram cerca de 30 mil contadores no âmbito da primeira fase do projecto InovGrid.

Com a iniciativa do InovGrid, a EDP promoveu o projecto Évora InovCity, a primeira cidade inteligente em Portugal, que abrange 54 mil habitantes. Com mais de 340 Controladores de Postos de Transformação instalados (G Smart) e com mais de 30 mil contadores inteligentes, os moradores de Évora podem beneficiar de um sistema eléctrico integrado e inteligente.

2.3 Sistemas de gestão de dados de larga escala

Nesta secção, irá ser feita a análise de algumas ferramentas, tecnologias de base de dados, com o intuito de analisar as suas características, e assim poder definir quais as possíveis candidatas para resolver a solução relacionada com o armazenamento e processamento de grandes quantidades de dados distribuídos, como o que o contexto dos Smart Meterings implica.

2.3.1 NoSQL

"NoSQL" é o nome dado a bases de dados que compartilham uma característica em comum: são base de dados não relacionais, no entanto, possuem características diferentes e seguem diferentes abordagens para o armazenamento de dados.

Ao contrário das bases de dados relacionais, as bases de dados NoSQL geralmente não têm qualquer estrutura imposta, sendo os candidatos ideais para o armazenamento de dados aleatórios não estruturados.

Normalmente, com uma base de dados relacional, é necessário definir a estrutura dos dados antes, criando o esquema de base de dados (tabelas e colunas), dependendo da estrutura dos dados. Isto dá-lhes muito pouca flexibilidade em relação aos dados que esta pode armazenar. Com uma base de dados NoSQL sem esquema (algumas base de dados NoSQL não são 100% sem esquema), todos os dados podem ser armazenados. Além dos ganhos de flexibilidade, permite também oferecer uma melhor eficiência de armazenamento, uma vez que não existem colunas vazias desperdiçadas.

A necessidade do desenvolvimento de base de dados NoSQL surgiu principalmente devido a problemas de escalabilidade nas base de dados relacionais, como a falta de suporte para ser distribuída, que alguns argumentam como "a chave para escrever escalabilidade" [16].

Embora nem todas as bases de dados NoSQL sejam distribuídas, todas elas tentam resolver problemas de escalabilidade. A grande vantagem que este tipo de bases de dados têm é então o aumento do desempenho e capacidade de escalabilidade que elas oferecem. Esta vantagem de desempenho é obtido graças a dois

factores importantes, a natureza não-relacional e sem esquema destas base de dados, o que implica que os modelos de dados sejam muito mais simples, aumentando a velocidade de acesso à base de dados, de forma significativa. O outro factor é que a maioria das bases de dados NoSQL tem a capacidade de ser distribuídas [17].

Devido à capacidade de distribuir uma base de dados através de vários nós, surgem vários problemas, tais como manter a consistência dos dados. Uma maneira de garantir a consistência dos dados numa base de dados distribuídos é através de "MultiVersion Consistency Control" [61].

Existem basicamente dois tipos de bases de dados NoSQL, que devem ser considerados para um sistema de armazenamento de grandes quantidades de dados, como o que o Smart Metering implica, e serão descritos em detalhes nas próximas secções:

- Orientado a Colunas(secção 2.3.1.1)
- Orientado a Documentos(secção 2.3.1.2)

2.3.1.1 Orientado a Colunas

Numa visão muito simplificada, o conceito de base de dados Orientado a Colunas, consiste em armazenar informações agrupadas por colunas, em vez de o agrupar por linhas, como nos Sistemas de Gestão de Base de Dados Relacionais tradicionais.

Isto permite que o processamento de dados seja mais rápido entre os itens de dados semelhantes, uma vez que eles se encontram fisicamente próximos em disco. A sua principal vantagem está, assim, no desempenho e na eficiência do disco.

De seguida será descrito, de forma pormenorizada, dois exemplos de base de dados NoSQL Orientado a Colunas, sendo elas:

- Apache Hbase [18]
- Cassandra [21]

2.3.1.1.1 Apache Hbase

Apache HBase é uma base de dados distribuída não-relacional open-source, modelada com base no Google's BigTable [22]. O Hbase, assim como o Google BigTable usa/aproveita o armazenamento de dados distribuído fornecido pelo Google File System(GFS); o Apache HBase fornece capacidades em cima de Hadoop e HDFS.

Em HBase, os dados são armazenados em tabelas, que possuem linhas e colunas. Pode ser visto como uma sobreposição da terminologia de bases de dados relacionais, no entanto esta não é uma analogia útil. Em vez disso, é útil pensar numa tabela HBase como um mapa multi-dimensional.

O modelo de dados HBase é muito similar ao do BigTable. As linhas de dados são armazenadas em tabelas etiquetadas, em que cada linha de dados contém uma chave ordenada e várias colunas. As colunas são identificadas por <familia>:<label>, onde a família e o label pode ser matrizes de bytes arbitrários.

As famílias de colunas são definidas administrativamente aquando da criação das tabelas, enquanto os labels

podem ser adicionados a qualquer momento.

As colunas não têm valores para todos as linhas de dados. Todas as famílias de colunas são armazenadas fisicamente próximas no disco, de modo a que itens da mesma família de colunas, tenham características semelhantes de leitura/escrita e tempos de acesso idênticos [19].

O exemplo seguinte, foi retirado da página wiki da Arquitectura do Hbase [20], e mostra uma tabela que contém duas linhas (*com.cnn.www* e *com.example.www*), três famílias de colunas denominados por *contents*, *anchor* e *people*. Neste exemplo, na primeira linha (*com.cnn.www*), a familia de colunas *anchor* contém duas colunas (*anchor: cssnsi.com*, *anchor: my.look.ca*) e a familia *content* contém uma coluna (*content:html*). Este exemplo contém cinco versões da linha com a chave de linha *com.cnn.www* , e uma versão da linha com a chave de linha *com.example.www*. O label da coluna *content:html* contém todo o HTML de um determinado site. Cada label da família coluna *anchor* contêm o site externo, que liga para o local representado pela linha, junto com o texto que é usado no *anchor* do seu link. A família de colunas *people* representa as pessoas associadas ao site.

Row Key	Time Stamp	ColumnFamily contents	ColumnFamily anchor	ColumnFamily people
"com.cnn.www"	t9		anchor:cnnsi.com = "CNN"	
"com.cnn.www"	t8		anchor:my.look.ca = "CNN.com"	
"com.cnn.www"	t6	contents:html = "<html>..."		
"com.cnn.www"	t5	contents:html = "<html>..."		
"com.cnn.www"	t3	contents:html = "<html>..."		
"com.example.www"	t5	contents:html = "<html>..."		people:author = "John Doe"

Figura 2: Tabela Hbase (Visão conceptual)

O modelo de dados Hbase suporta quatro tipos de operações:

- GET - retorna atributos de uma linha especifica.
- PUT - adiciona novas linhas a uma tabela(chave nova) ou actualiza linhas existentes(chave já existente)
- SCAN - permite iterar sobre múltiplas linhas para atributos específicos
- DELETE - remove uma linha de uma tabela

O Hbase não suporta Joins, pelo menos não da forma como são feitos nas RDBMS. No entanto, isso não significa que não seja possível serem efectuados. As duas principais estratégias são desnormalizar os dados na escrita para o HBase, ou ter tabelas de pesquisa e fazer a junção entre tabelas HBase na própria aplicação ou código MapReduce, sendo estas dependentes da abordagem adquirida.

A utilização ou não do Hbase, assim como grande parte das bases de dados não-relacionais, deve tomar em conta, os três seguintes princípios [20]:

- Quantidade de dados é justificável ou seja grandes quantidades de dados. Se existem centenas de milhões ou biliões de linhas então o Hbase é uma boa solução.
- Sobrevivência sem recursos extra que o RDBMS fornecem(colunas tipificadas, índices secundários, transacções, linguagens de consulta avançadas, etc).

- Garantir que existe hardware suficiente. Até o HDFS não funciona bem com qualquer coisa com menos de 5 DataNodes(bloco de replicação HDFS – padrão=3)

2.3.1.1.2 Apache Cassandra

Apache Cassandra é uma base dados open source NoSQL extremamente escalável, inicialmente desenvolvida pelo Facebook para alimentar o seu motor de busca [22], promovida a projecto Apache Incubator, em Março de 2009 e a projeto Top-Level, em Fevereiro de 2010.

Cassandra é perfeita para gerir grandes quantidades de dados de múltiplos centros de dados, e fornece uma disponibilidade contínua, escalabilidade linear, e simplicidade operacional através de múltiplos servidores sem um único ponto de falha, juntamente com um modelo de dados poderoso projectado para máxima flexibilidade e tempos de resposta rápidos.

Cassandra tem uma arquitectura "masterless", ou seja, cada nó tem um papel igual e não há um nó mestre. Cassandra fornece distribuição automática de dados em todos os nós que participam no "ring" ou cluster de base de dados [23].

No modelo de dados do Cassandra, uma tabela é um mapa multidimensional distribuído, indexado por uma chave. O valor é um objecto que é altamente estruturado, e cada operação sobre uma única chave de linha é atómica, não importa quantas colunas estão a ser lidas ou escritas.

As colunas são agrupadas em conjuntos, designados de famílias de colunas(semelhante a BigTable e Apache HBase). Existem dois tipos de família colunas, famílias de colunas simples e super famílias de colunas. Super família de colunas pode ser visto como uma família coluna dentro de outra família de colunas.

Existem dois particionadores: um que distribui os dados de forma aleatória em todo o cluster, e outro para preservação da ordem, que preserva a ordem dos dados. Além disso, os dados são replicados para vários nós, de forma a ser tolerante a falhas.

O modelo de dados Cassandra suporta três tipos de operações:

- insert(table; key; rowMutation) - adiciona nova linha à tabela
- get(table; key; columnName) - retorna uma linha especifica da tabela
- delete(table; key; columnName) - elimina uma linha especifica da tabela

A arquitectura Cassandra permite que qualquer utilizador autorizado se conecte a qualquer nó, em qualquer centro de dados, e aceda aos dados usando a linguagem Cassandra Query Languague(CQL).
Cassandra Query Languague é a interface primária e padrão para DBMS Cassandra. CQL é semelhante a SQL, no entanto não suporta joins e subqueries, exceto para análise batch através de Hive [24].
A figura seguinte [25] mostra uma única família de colunas, UserURLs, com uma única linha, identificada pela chave "98725". Esta família coluna tem duas super colunas ("http://techcrunch.com/..."and" http: //cnn.com / ... "), e cada super coluna tem exactamente duas colunas.

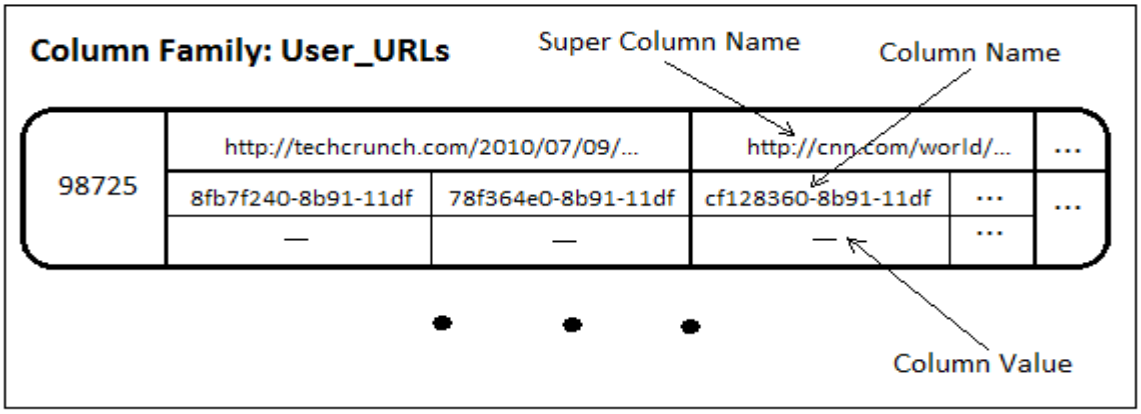

Figura 3: Modelo de dados Twitter: User_URLs

2.3.1.2 Orientado a Documentos

As bases de dados Orientada a Documentos são uma evolução das bases de dados de armazenamento chave / valor; em vez de armazenar qualquer tipo de dados, como a maioria das base de dados chave / valor, a bases de dados Orientada a Documentos armazenam apenas documentos correctamente formatados, oferecendo um conjunto de recursos, em cima de armazenamentos chave / valor.

A maioria das bases de dados oferecem a possibilidade de pesquisa por documentos, de acordo com o seu valor semântico (pesquisa de campos específicos em documentos), ou a possibilidade de filtragem de documentos, também com base no seu valor semântico (por exemplo, listagem apenas dos documentos que começam com tag X e Y).

A maioria das bases de dados Orientadas a Documentos suportam JSON (JavaScript Object Notation) e documentos XML.

Estes sistemas de armazenamento são geralmente livre de esquema, ou sem esquema, ou seja, não impõem qualquer tipo de estrutura nos dados armazenados (desde que todos os documentos sejam válidos).

Embora existam muitas bases de dados Orientadas a Documentos XML (como o Apache Xindice, MonetDB e BaseX), a maioria das implementações recentes suportam apenas documentos JSON. Isto justifica-se pelo facto de que os documentos JSON são menores, e transmitem o mesmo significado semântico que documentos XML, o que se traduz numa eficiência maior em disco.

De seguida serão descritos, dois exemplos de base de dados NoSQL Orientado a Documentos, sendo eles:

- Apache CouchDB
- MongoDB

2.3.1.2.1 Apache CouchDB

CouchDB é um sistema de gestão de base de dados open-source, acessível através de uma API RESTful JSON. O termo “Couch” é o acrónimo de “Cluster Of Unreliable Commodity Hardware”, que reflecte as

principais características do CouchDB, como o facto de ser extremamente escalável, altamente disponível e confiável, mesmo quando executado em hardware tipicamente propenso a falhas. O CouchDB foi criado por Damien Katz, ex-desenvolvedor da Lotus Notes, na altura funcionário da IBM, e inicialmente lançado em 2005. Foi originalmente escrito na linguagem C++, e mais tarde alterado para Erlang OTP (Open Platform Telecom), devido à sua ênfase na tolerância a falhas. Suporta objectos JSON, onde cada documento é identificado por um identificador (Id) ou chave [26].

CouchDB possui uma API JSON HTTP RESTful que permite que aplicações possam ver, inserir, modificar e eliminar documentos. Isso faz com que a linguagem de acesso à base de dados, tenha uma única exigência de ser uma biblioteca HTTP (embora possa ser necessário uma biblioteca JSON para solicitar e manipular documentos).

Na sua documentação, o CouchDB é descrito como sendo "ad-hoc e com um espaço de endereço plano e livre de esquema". Por isso não suporta nenhum esquema, não fazendo qualquer distinção significativa entre os documentos que armazena [27]. O único requisito necessário é que os documentos sejam documentos JSON. Isso significa que não é necessária nenhuma modelação de base de dados prévia ou normalização, reduzindo significativamente a complexidade do modelo de dados, e simplificando "o desenvolvimento de aplicações Orientadas a Documento" .

Os documentos do CouchDB são documentos JSON identificados por um DocumentID único (_id). Além do DocumentID, todos os documentos também têm um RevisionId (_rev), usado para identificar diferentes versões do mesmo documento.

2.3.1.2.2 MongoDB

MongoDB é uma base de dados open-source escalável e de alto desempenho, orientada a documentos, implementada, na linguagem C++, pela 10gen e uma comunidade open-source, focada no desempenho de base de dados não-relacionais[28]. O nome Mongo vem do "huMONGOus".

O MongoDB além de armazenar documentos, também suporta dados binários, como vídeos e imagens. Tem suporte para índices, replicação, Map Reduce e suporte comercial.

MongoDB armazena documentos BSON (Binary-JSON), uma serialização binária JSON de documentos JSON. BSON suporta objectos incorporados, e objectos de referência.

MongoDB suporta actualizações “in-place”, ou seja, apenas os atributos alterados devem ser enviadas para a base de dados, ao contrário do CouchDB, onde as actualizações provocam uma inserção do mesmo documento, com uma revisão diferente.

MongoDB suporta um esquema (MongoDB não é sem esquema), onde cada objecto ou documento de nível superior, é representado por uma "colecção" [29].

Cada documento está associado a um ID, utilizado como chave primária e índice, embora o desenvolvedor possa criar índices em todos os campos que estão disponíveis, o que permite que sejam feitas consultas a esses campos específicos.

MongoDB tem um modelo de consulta rico (em padrões de base de dados Orientados a Documentos), que

permite consultas usando comparadores, operadores lógicos, operadores condicionais e agregação. Isto significa que os desenvolvedores provenientes de bases de dados relacionais descubram que são possíveis muitas consultas SQL e podem ser traduzidas para consultas MongoDB.

2.4 Análise Big Data

Análise é um termo usado para descrever a decomposição de algo complexo em partes menores e mais simples. A *análise de dados* neste contexto, significa o processo de obtenção de conhecimento utilizando decomposição e análise estatística dos dados.

A *ciência da análise de dados* é um termo mais recente para análise de dados, que significa a aplicação de uma quantidade de tecnologias diferentes em análise estatística, tais como, machine learning, data mining ou métodos de visualização para extracção de conhecimento dos dados.

A ciência dos dados é um termo similar usado na análise de dados, mas com um peso mais relevante na área do machine learning e recomendações de acções futuras.

A história da ciência da análise de dados segundo [30], surgiu por volta de 1962, quando John W. Tukey escreveu em “The Future of Data Analysis” a crescente importância da análise de dados. Desde então, o campo da análise de dados tem vindo a ser expandido para diversas áreas, e o interesse pela descoberta de conhecimento dos dados consequentemente tem vindo a crescer.

Nos dias de hoje, a ciência dos dados é considerada uma área madura, no entanto continua a ser uma área com um emergente e amplo campo de investigação, no que diz respeito aos processos de armazenamento e processamento de dados, análise quantitativa e qualitativa, modelação preditiva e tudo o resto que pode ser feito com os dados.

A análise de grandes quantidades de dados é o processo que permite examinar grandes conjuntos de dados contendo uma variedade de tipos de dados, ou seja “Big Data”, para descobrir padrões e correlações desconhecidas, tendências de mercado, preferências dos clientes e outras informações de negócio úteis, consoante o contexto onde se enquadra. Os resultados analíticos proporcionam vantagens a diversas níveis, como por exemplo: levar a um marketing mais eficaz, novas oportunidades de receita, melhor serviço ao cliente, a melhoria da eficiência operacional, vantagens competitivas sobre organizações rivais e outros benefícios de negócios.

Um processo comum de análise de dados pode ser caracterizado pelas seguintes etapas: armazenamento de dados, preparação e pré-procesamento de dados, exploração de dados, análise de dados e visualização dos resultados da análise.

Neste projecto vamos seguir este processo tanto quanto o possível, para a implementação da solução posteriormente apresentada.

2.4.1 Big Data

Big Data segundo [31], pode ser visto, como a quantidade de dados que excede a capacidade de processamento dos sistemas de gestão de base de dados convencionais. Esta propriedade pode advir de diversos factores como, a quantidade de dados ser demasiado grandes, mover-se muito rapidamente ou não seguir a estrutura das arquitecturas das base de dados convencionais.

No contexto das Smart Grids, o alto volume, velocidade e diversidade de dados que fluem entre os seus nós requerem a criação e utilização de técnicas eficientes especificas para lidar com os dados. A extracção de informação valiosa dos dados Big Data gerados por dispositivos inteligentes é um dos aspectos centrais do conceito de Smart Metering.

Deste modo, como podemos quantificar a grandeza de dados Big Data? Geralmente neste contexto, “grande” significa que o volume de dados não pode ser lido, de uma só vez, para a memória principal de um único sistema. Portanto, na maioria dos casos, é necessário recorrer a um sistema distribuído ou um sistema extremamente poderoso capaz de lidar com esta quantidade de dados, através de algoritmos eficientes.

Big Data tornou-se uma realidade em muitas organizações, devido ao crescente aumento de dados gerados essencialmente por máquinas, mas também pela velocidade com que estes são partilhados entre estas. Assim sendo, a combinação entre estes dois factores, volume de dados e velocidade, são nos dias de hoje um grande desafio para a indústria.

O uso de Big Data proporciona uma compreensão mais profunda ao nível do domínio aplicacional, o que implica um aumento da segurança e um melhor planeamento. No entanto, como indicado pela Intel em 2012, em [32], uma mudança tecnológica é necessária a fim de analisar grandes quantidades de dados de forma eficiente.

Muitas vezes, os dados vêm em forma de fluxo ilimitado e o seu processamento não permite qualquer tipo de interrupção, o que implica que sejam utilizados métodos sofisticados de análise, capazes de processar grandes quantidades de dados a uma velocidade constante.

2.4.2 Classificação de Métodos Analíticos

Existem duas abordagens principais relacionadas com o hardware para lidar com dados “Big Data”:

- armazenar os dados numa máquina com uma memória principal de grande capacidade
- utilizar um sistema distribuído

O uso de um sistema distribuído tem inúmeras vantagens quando comparado com um sistema centralizado de uma só máquina, nomeadamente, um sistema distribuído possui uma capacidade de escalabilidade maior, é mais fiável devido à sua capacidade de ser redundante e o seu poder de computação tem a capacidade de aumentar incrementalmente.

Os métodos analíticos de dados podem ser divididos em quatro categorias:

- sistemas analíticos padrão, com volume de dados limitado ao tamanho do disco rígido
- sistemas analíticos em memória
- sistemas analíticos distribuídos
- processamento de eventos complexos e processamento de transmissão de eventos

Estas quatro categorias de métodos analíticos possuem diferentes características em termos de uso para processamento de dados, sendo a principal diferença, que as primeiras três não são capazes de processar informação dinâmica em tempo-real, ao contrário dos sistemas de processamento de eventos que foram projectados para lidar com tarefas em tempo-real.

2.4.3 Análise de Dados Distribuídos

De acordo com [33], um sistema distribuído é uma colecção de computadores autónomos conectados, que são apresentados ao utilizador como um sistema único.

Como referido anteriormente, a análise de dados distribuídos oferece mais vantagens do que a utilização de um sistema convencional de uma única máquina. A simples paralelização de uma única máquina, muitas vezes não é rápida o suficiente para uma aplicação Big Data ser executada num período de tempo consideravelmente razoável.

A análise de dados distribuídos permite optimizações, predições e prescrições quase em tempo-real.

A Google introduziu em [34], um novo paradigma de programação e uma framework de software designada de MapReduce, que permite efectuar computações distribuídas de larga escala num cluster e em [35], um sistema de ficheiros distribuídos subjacente apropriado.

Em 2006, uma implementação construída sobre as ideias apresentadas nestes papers foi transferida para um projecto open-source designado de Apache Hadoop, que atraiu imensos colaboradores em todo o mundo. Nos anos seguintes, até à data actual, o Apache Hadoop, incluindo o HDFS(Hadoop Distributed File System) e o MapReduce tornou-se para muitos utilizadores o sistema escolhido para executar computações paralelas num cluster.

O Apache Hadoop oferece diversas possibilidades de análise de grandes quantidades de dados, no entanto é impossível de alcançar baixa latência e desempenho em tempo-real, devido ao armazenamento dos resultados computacionais no disco (como HDFS). Por isso, muitos projectos adicionais em volta do Hadoop surgem hoje em dia, o que permite o processamento de eventos em tempo-real, utilizando a lógica do MapReduce.

A mudança para o armazenamento de dados em memória, pode reduzir significativamente a latência e oferecer a capacidade de processamento quase em tempo-real.

Nos seguintes sub-capítulos irá ser feita uma descrição dos fundamentos básicos da framework Apache Hadoop, incluindo os seus principais componentes HDFS, MapReduce e YARN.

2.4.3.1 Apache Hadoop

A biblioteca de software Apache Hadoop [12] é uma framework que permite o processamento e armazenamento distribuído de grandes conjuntos de dados através de clusters("grupos/aglomerados") de computadores usando modelos de programação simples. Foi concebido para escalar a partir de um servidor para milhares de máquinas, cada uma delas oferecendo computação e armazenamento local.

Em vez de confiar em hardware para proporcionar alta disponibilidade, a biblioteca foi concebida para detectar e tratar falhas ao nível da camada aplicacional, de modo a fornecer um serviço altamente disponível no topo de um cluster de computadores, em que cada um dos deles, normalmente pode ser propenso a falhas.

O Apache Hadoop base é composto pelos seguintes módulos :

- Hadoop Commom - contém livrarias e utilidades necessárias para outros módulos do Hadoop
- Hadoop Distributed File System (HDFS) - sistema de ficheiros distribuídos que armazena dados nas máquinas, proporcionando uma largura de banda agregada em todo o cluster
- Hadoop YARN - plataforma de gestão de recursos, responsável pela gestão de recursos de computação em clusters e uso deles para programação de aplicações
- Hadoop Map Reduce - modelo de programação para processamento de dados em larga escala

2.4.3.1.1 Hadoop Distributed File System

Hadoop Distributed File System tem muitas semelhanças com os sistemas de ficheiros distribuídos existentes, no entanto existem algumas diferenças significativas. HDFS é altamente tolerante a falhas, e foi concebido para ser construído em hardware de baixo custo. HDFS fornece alta taxa de transferência de acesso a dados de aplicações, e é adequado para aplicações com conjuntos de dados muito grandes.

Uma instância HDFS pode conter centenas ou milhares de servidores, que armazenam parte dos dados do sistema de ficheiros. O facto de possibilitar a existência de uma grande quantidade de componentes, e cada um deles ter uma probabilidade não trivial de falhar, significa que algum componente de HDFS é sempre não-funcional. Portanto a rápida detecção de falhas e recuperação automática a partir das mesmas é o objectivo central do HDFS. De seguida é apresentada a arquitectura base do HDFS:

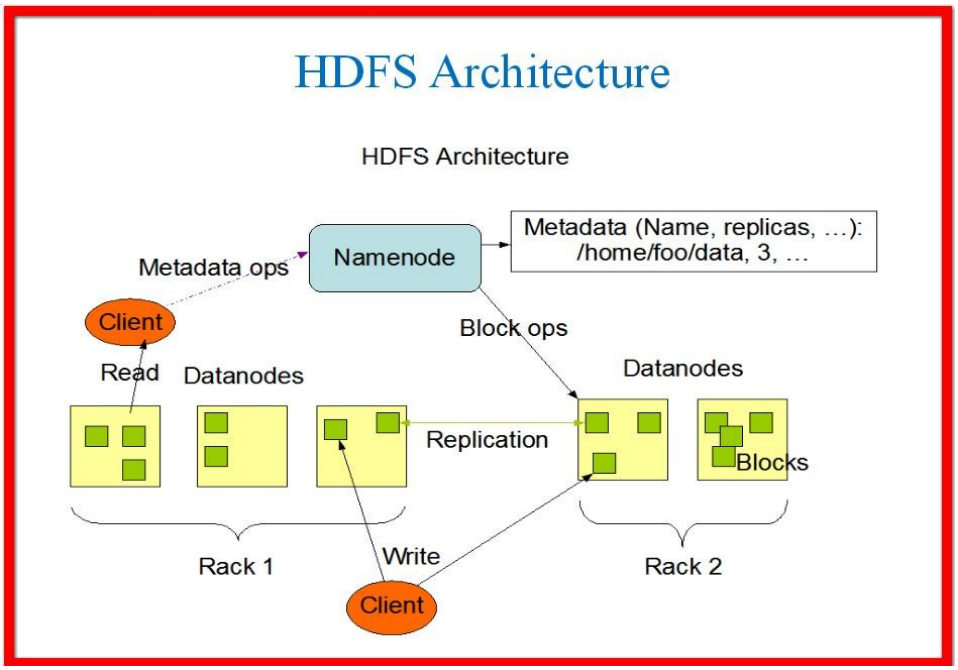

Figura 4: Arquitectura HDFS [13]

2.4.3.1.2 MapReduce

O Map Reduce é uma framework que permite a escrita de programas que processam grandes quantidades de dados não estruturados em paralelo, através de um cluster distribuído de processadores ou computadores .

Os dados de entrada e saída de MapReduce são armazenados no HDFS, e o cluster para MapReduce incorpora um node master e alguns worker nodes.

Basicamente, o paradigma MapReduce pode ser dividido em duas partes:

- Map - função que divide o trabalho por diferentes nós do cluster
- Reduce - função que reúne o trabalho e resolve os resultados num único valor

Durante a etapa Map, o master node subdivide a entrada em tarefas e distribui-as entre os worker nodes. Este passo pode continuar hierarquicamente até que um worker-leaf node processe a tarefa e retorne um resultado. Os resultados são agrupados por chave e recolhidos em paralelo no passo Reduce.

Formalmente, uma função Map recebe como entrada um par chave-valor (*K1*, *V1*) e retorna uma lista de outros pares chave-valor(*K2*,*V2*), que são distribuídos entre os nodes workers.

$$map(K1,V1) \rightarrow list(K2,V2)$$

No próximo passo, os elementos da lista são agrupados pela chave *K2*, preparando a entrada para a função Reduce.

Na etapa Reduce, uma função aplicada nas listas agrupadas é distribuída entre os worker nodes e os valores dos resultados da função são retornados como a saída de dados.

$$reduce(K2,list(V2)) \rightarrow list(V3)$$

A figura seguinte demonstra o processamento dos dados efectuado no paradigma MapReduce:

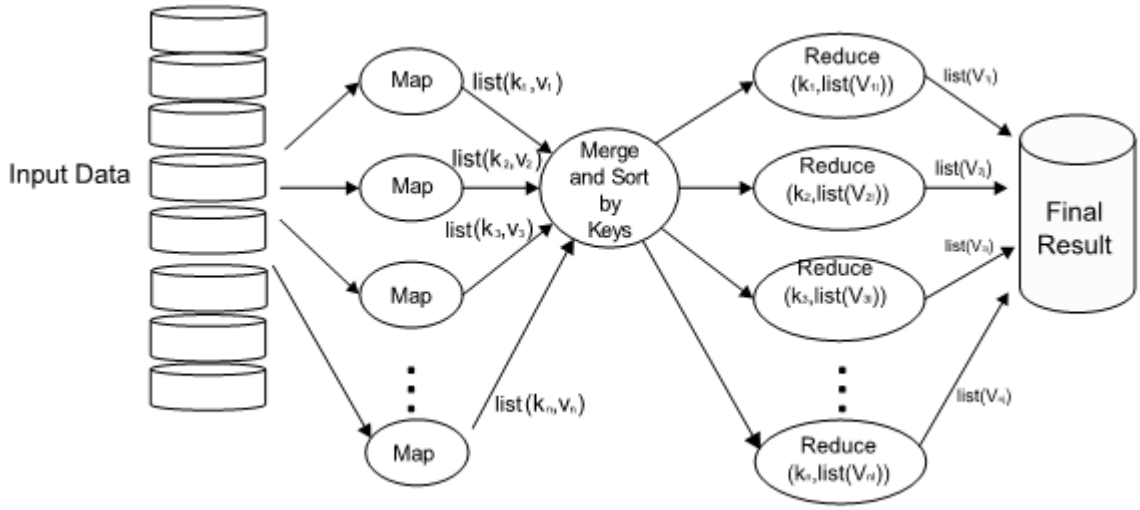

Figura 5: Conceito MapReduce

Ao contrário da programação funcional normal, o MapReduce utiliza funções map e reduce que são aplicadas em pares chave-valor e não em listas simples.

2.4.3.1.3 YARN

Com a introdução do Apache Hadoop YARN(Yet Another Resource Negotiator) [14], a gestão dos recursos Hadoop foi separado das tarefas MapReduce. O MapReduce original era encarregue de efectuar ambos, recursos e tarefas.

A ideia fundamental do YARN é dividir as duas maiores funcionalidades do JobTracker, a gestão de recursos e a monitorização/programação de trabalho, em programas separados. A ideia é ter um ResourceManager (RM) e um ApplicationMaster (AM) por aplicação [14].

O ResourceManager e o “escravo” por nó(NodeManager), formam a framework de computação de dados. O ResourceManager é a autoridade que arbitra os recursos entre todas as aplicações no sistema.

A ApplicationMaster é a biblioteca especifica da framework, encarregue de gerir os recursos do ResourceManager, e trabalhar com o NodeManager para executar e monitorizar as tarefas.

O ResourceManager tem dois componentes principais: Scheduler e o ApplicationManager.

O Scheduler é responsável por alocar recursos para as várias aplicações em execução, e executa qualquer

monitorização do estado de aplicações. O Scheduler não permite reiniciar tarefas que falhem, quer seja por falha da aplicação ou falha no hardware.

O ApplicationManager é responsável pela aceitação de tarefas, por efectuar a negociação do primeiro conteúdo, para execução do ApplicationMaster e por fornecer um serviço para reiniciar o ApplicationMaster, em caso de falha.

O NodeManager é o agente responsável pelos containers, monitoriza a utilização dos seus recursos(CPU, memória, disco, rede) e relata para o ResourceManager/Scheduler.

A ApplicationMaster é responsável por negociar os recursos apropriados do Scheduler, acompanhar o seu estado e monitorizar o seu progresso.

A seguinte figura ilustra a arquitectura YARN:

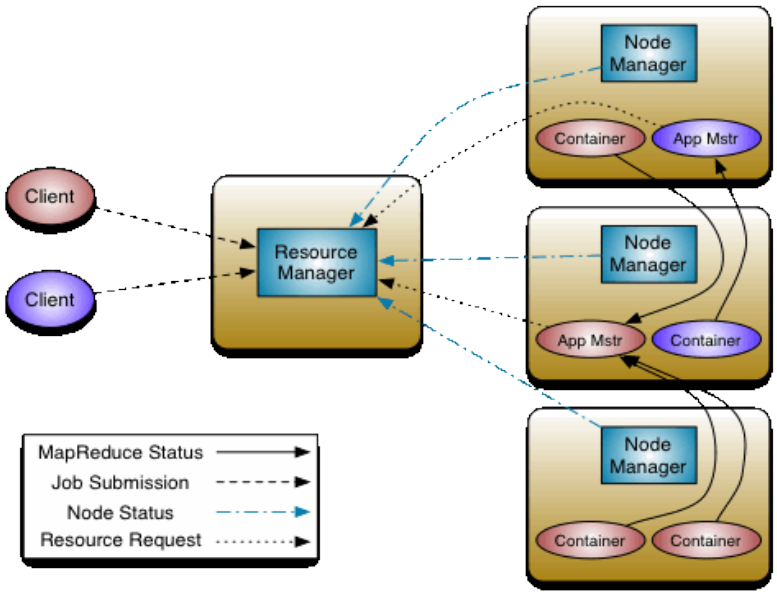

Figura 6: Arquitectura YARN

2.4.4 Processamento de Eventos Complexos (CEP)

O processamento de eventos complexos (CEP) é uma nova técnica de análise Big Data, que envolve a seguinte consideração: os dados são uma série desordenada de eventos provenientes de diversas fontes. Os dados de entrada são monitorizados, utilizando condições declarativas. Além disso, a monitorização e processamento de dados é efectuada com uma latência próxima de zero.

Um evento pode ser entendido como uma acção atómica predefinida de entrada ou uma transacção a partir do ambiente. Um evento complexo é uma composição de vários eventos que pertencem a um objecto semântico.

Eventos diferentes podem ser provenientes de diferentes fontes e o sistema CEP pode montar um evento complexo internamente, modelando as suas partes para formar um objecto.

Um sistema CEP é basicamente utilizado para resolver o problema de velocidade do processamento de grandes quantidades de dados, quando os dados são recebidos como um fluxo predefinidos de eventos. Um sistema CEP utiliza uma abordagem de janela deslizante, que assegura que apenas uma parte dos dados reais passa simultaneamente na memória principal, enquanto que os eventos antigos, já processados, podem ser descartados ou armazenados. Desta forma, não é necessário utilizar todos os dados de uma só vez em memória, podendo assim analisar de forma eficiente só os dados recentes.

Uma framework CEP permite agregar dados, dada uma função, em pequenas porções de tempo. A informação agregada pode ser utilizada desde que esta não seja descartada da memória.
Os sistemas CEP pressupõem a extracção de uma série de eventos a partir de um conjunto de eventos com um determinado padrão, normalmente efectuado através de uma consulta semelhante a uma consulta SQL, portanto os eventos devem seguir uma estrutura comum. O conceito de publish-subscribe, seguido pelo primeiro artigo em sistema CEP [36], inclui duas partes: um conjunto de subscribers que estão interessados num determinado tópico e obtêm notificações quando algo do seu interesse é publicado, e um publisher que transmite indirectamente informações sob a forma de eventos para os subscribers, sem entrar em contacto directo com qualquer um deles [37].
O processamento de eventos complexos é utilizado para detecção de falhas e anomalias, detecção de fraudes, análise de logs em tempo-real, análise de smart meterings, operações de negócios entre outras.
Uma pesquisa sobre frameworks CEP [38], contém o resumo de diferentes implementações e oferece uma visão geral da área em 2010.

2.4.5 Processamento de Fluxo de Eventos (ESP)

Os sistemas de processamento de fluxo de eventos (ESP), são em subconjunto de sistemas de processamento de eventos complexos. Ao contrário de um sistema CEP, um sistema de processamento de fluxo de eventos, baseia-se em dados ordenados, através de um timestamp fornecido, geralmente proveniente uma única fonte de dados.

Um evento num sistema ESP é representado por um tuplo (x,y), em que x é o timestamp e y é um vector que contém os restantes atributos do evento. O fluxo de eventos é um conjunto de eventos, que chegam com uma ordem linear, consoante o seu timestamp.

Assim como os sistemas CEP, também os sistemas de processamento de fluxo de eventos permitem efectuar análises de dados com uma latência baixa. No entanto, como o processamento de fluxo de eventos se baseia num único fluxo estritamente ordenado por tempo, este deve executar algoritmos de forma mais rápida do que um sistema CEP.

Stonebraker et. al. discute em [39], os requisitos de software necessários para um processamento de fluxo em

tempo-real e compara as capacidades de motores de regras, sistemas de gestão de base de dados e sistemas de processamento de fluxo de eventos. Os autores concluem que qualquer sistema tradicional, que não seja um sistema de processamento de fluxo de eventos especial não cumpre os requisitos necessários para o processamento de fluxo em tempo-real bem sucedido. Os requisitos necessários para estes sistemas são: movimento de dados contínuos, manipulação de falhas de transmissão, segurança e disponibilidade dos dados, baixa latência, escalabilidade, geração de resultados de previsão, integração de dados de fluxo e armazenados e consulta de dados através de uma window.

O processamento de fluxo de eventos é utilizado em mudanças de cotas de acções e análise smart metering.

Em [40] , é apresentado um sistema de processamento de fluxo de eventos, capaz de consultar leituras RFID e processá-las em tempo-real.

2.4.6 Ferramenta – Apache Spark

O Apache Spark é ferramenta de processamento “Big Data” open-source construída em torno da velocidade, facilidade de uso e análise sofisticada. Foi originalmente desenvolvida em 2009 em UC Berkeley’s AMPLab e tornada open-source em 2010 como um projecto Apache.

Spark permite escrever rapidamente aplicações em Java, Scala e Python, através das APIs de alto nível que este fornece. Tem integrado um conjunto com mais de 80 operações de alto nível e pode ser utilizado de forma interactiva para pesquisar dados, através da sua linha de comandos.

Spark leva o conceito de MapReduce para um nível seguinte, através de um processamento de dados mais simples e claro. Através dos seus recursos, como armazenamento de dados em memória e processamento quase em tempo-real, o seu desempenho é bastante mais rápido quando comparada com outras tecnologias de processamento de grandes quantidades de dados. O Spark também suporta avaliação “lazy” de pesquisas “Big Data”, que permite a optimização das etapas de fluxo de processamento de dados.

O Spark guarda os resultado intermédios em memória, em vez de os escrever directamente para o disco, o que é bastante útil, quando é necessário trabalhar no mesmo conjunto de dados múltiplas vezes, uma vez que o acesso a memória é mais rápido do que o acesso ao disco. O Spark foi desenvolvido como um mecanismo de execução que funciona tanto em memória como em disco. Os operadores Spark executam operações externas quando os dados não cabem todos em memória, o que permite que este seja usado para processamento de conjuntos de dados maiores do que a memória agregada de um cluster.

Spark tenta armazenar o máximo de dados possível em memória, e em seguida armazena em disco. Uma parte dos dados podem ser armazenada em memória e o restante em disco, pelo que quando se pretende construir uma solução com recurso ao Spark, é necessário primeiro analisar os seus dados e casos de uso, para avaliar os requisitos de memória. Através da capacidade de armazenamento de dados em memória, o Spark apresenta uma grande vantagem ao nível do desempenho.

Para além da API principal do Spark, o Apache Spark suporta um conjunto de bibliotecas de nível superior, que permitem pesquisas SQL, streaming de dados, machine learnig e processamento de dados de grafos.

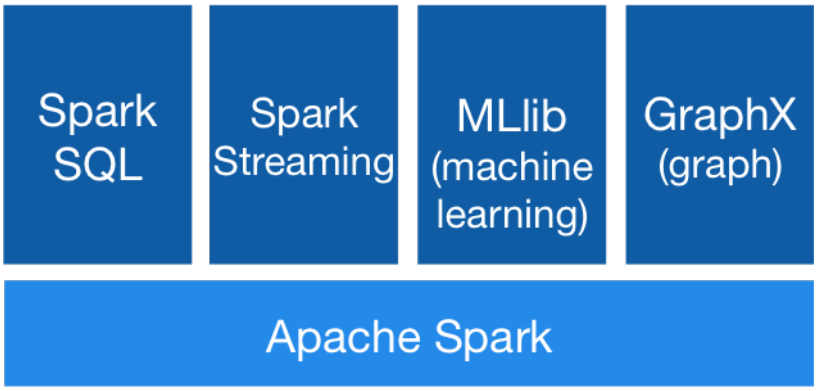

Figura 7: Bibliotecas Apache Spark

Estas bibliotecas tem incluído:

- **Spark Streaming:** Spark Streaming pode ser usado para processar fluxos de dados em tempo-real. Baseia-se no modelo de computação e processamento de pequenos intervalos de tempo("micro batch"). Uitliza o Dstream, que é basicamente uma série de RDDs, para processar os dados em tempo-real.
- **Spark SQL:** Spark SQL fornece a capacidade de expor o conjunto de dados sobre a API JDBC e permitir a execução de pesquisas de dados SQL. Permite a extracção, transformação e leitura(ETL) de dados com diversos formatos e diversas fontes, como JSON, Parquet ou base de dados,
- **Spark Mllib:** Mllib é uma biblioteca de machine learning escalável, composta por algoritmos de aprendizagem comuns e utilitários, incluindo classificação, regressão, clustering, filtragem colaborativa, redução de dimensionalidade, bem como optimizações primitivas.
- **Spark GraphX:** GraphX é nova API do Spark, para grafos e computação paralela de grafos. Num alto nível, o GraphX extende o Spark RDD através da introdução do RDPG(Resilient Distributed Property Graph): um multi-grafo com propriedades anexadas a cada vértice e aresta. Inclui um crescente conjunto de algoritmos de de grafos e construtores para simplificar a análise de grafos.

A arquitectura Spark inclui 3 principais componentes:

- Armazenamento de dados
- API
- Gestão de recursos

Spark utiliza o sistema de ficheiros HDFS para armazenamento de dados, e possibilita a utilização de qualquer outra fonte de dados compatível com o Hadoop, como o Cassandra ou Hbase. A API permite, aos

programadores de aplicações, criar aplicações Spark através de uma interface da API padrão. Spark pode ser implementado como um servidor autónomo ou como uma estrutura de computação distribuída, como o Mesos ou YARN. A imagem seguinte mostra os componentes do modelo de arquitectura Spark:

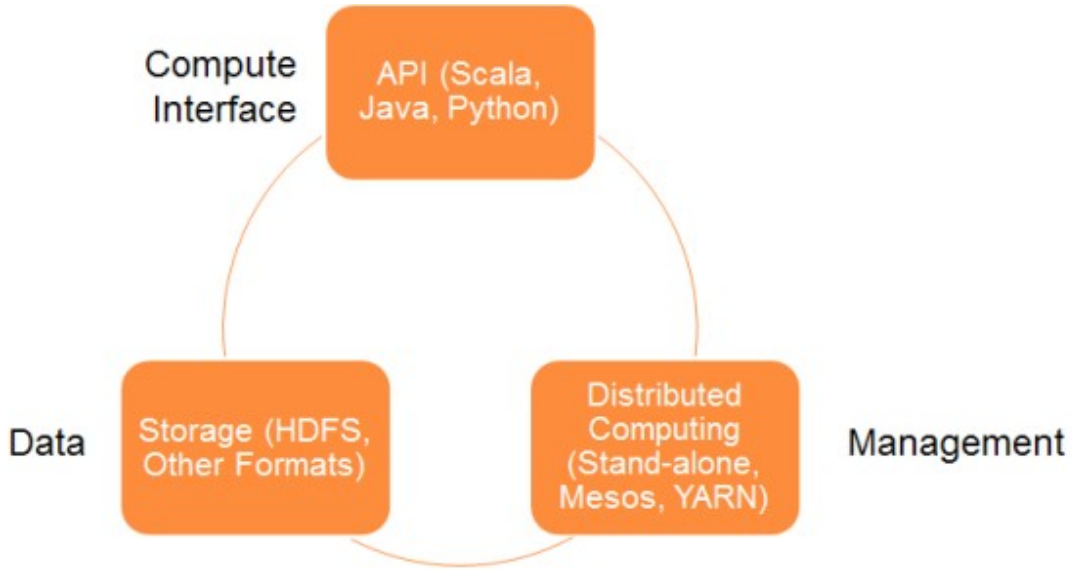

Figura 8: Arquitectura Spark

O Resilient Distributed DataSet ou RDD, com base em [62] é o conceito principal da ferramenta Apache Spark. O RDD pode ser visto como uma tabela de uma base de dados. Tem capacidade para armazenar qualquer tipo de dados e o Spark armazena os dados no RDD em diferentes partições.

Os RDDs auxiliam a reorganização dos cálculos e optimizam o processamento de dados. São tolerantes a falhas, pois estes são capazes de recriar e recalcular o conjunto de dados.

Os RDDs são imutáveis, ou seja, o RDD pode ser modificado através de uma transformação, mas a transformação retorna um novo RDD, enquanto que o RDD original se mantém o mesmo.

Os RDDs suportam dois tipos de operações:

- Transformações
- Acções

As transformações não retornam um único valor, retornam um novo RDD. Nada é avaliado quando é invocada uma função de transformação num objecto RDD, é apenas retornado um novo RDD. Algumas das funções de transformação são, map, filter, flatMap, groupByKey, reduceByKey, aggregateByKey, updateByKey, pipe e coalesce.

As acções avaliam e retornam um novo valor. Quando uma função de acção é invocada num objecto RDD,

todas as pesquisas de processamento de dados são computadas naquele momento e é retornado o valor do resultado. Algumas das funções de acção são, reduce, collect, count, first, take, countByKey e foreach.

2.5 Análise e Previsão do Consumo de Água

A água é indiscutivelmente um recurso natural de importância vital para a sobrevivência de qualquer forma de vida existente no nosso planeta. Para todos os seres vivos a água tem um carácter de necessidade imprescindível, mas existem alguns organismos com uma dependência de água superior quando comparada com a dependência de outros.

A sociedade onde nos enquadra-mos é o exemplo disso, pois esta é altamente dependente do fornecimento e consumo de água, por isso a sua preservação é da maior importância e o seu uso deve ser feito de forma sustentável.

Neste contexto, as smart grids surgem como um possível contribuinte na ajuda da monitorização do uso e consumo de água, pelo que as suas características permitem, que através de diversas técnicas, seja efectuada uma análise do consumo de água, por parte dos seres humanos.

A informação disponibilizada pelos dispositivos de smart metering, permite que seja efectuado um controlo do consumo de água e seja fornecida informação analítica e preditiva sobre o mesmo.

2.5.1 Métricas e Objectivos da Análise e Previsão do Consumo de Água

A análise e previsão de informação do consumo de água tem como principal objectivo a monitorização do consumo de água, por forma a que seja possível ter um maior controlo sobre a utilização da mesma. Desta forma, é possível efectuar um acompanhamento mais preciso, que permite auxiliar aos clientes a tomada de decisões, com a perspectiva de reduzir custos e de efectuar um consumo de água mais sustentável.

Do ponto de vista, dos operadores de negócio, a análise e previsão de informação também possui um grande impacto, pois permite que seja adquirido conhecimento sobre o consumo dos seus clientes, identificação de padrões e tendências de consumo, o que capacita os operadores a serem mais activos e afectivos, por exemplo na identificação de picos de consumo ou detecção de fugas de água.

A análise e previsão do consumo de água pode ser efectuada em diferentes níveis de intervalos de tempo. O intervalo de tempo definido, em geral depende dos dados e do objectivo da análise e previsão.

Por norma, os níveis de intervalo de tempo são classificados em três tipos: análise e previsão de curto prazo, análise e previsão de médio prazo, análise e previsão de longo prazo.

A análise e previsão de curto prazo, significa fornecer informação, quer seja analítica ou preditiva, para os próximos minutos. Esta análise e previsão pode ser utilizada para programação, planeamento e controlo de

sistemas.

A análise e previsão de consumo de médio prazo é utilizada para planeamento e operação de sistemas. A informação pode ser fornecida num intervalo de tempo que pode ir de um dia, até um mês.

Por sua vez, a análise e previsão de longo tempo é utilizada para auxilio de tomada de decisões estratégicas, com um intervalo de tempo de um ano ou mais.

As métricas para medir a qualidade da análise e previsão de consumo de água podem ser divididas em duas categorias principais: métricas para medir a precisão dos valores obtidos e métricas para medir o tempo de processamento (latência). De seguida são apresentadas as principais métricas:

Nome	**Fórmula**
Erro Relativo	$(Ai - Fi)/Fi$
Percentagem Erro Relativo	$((Ai - Fi)/Fi) * 100$
Erro Quadrático Médio	$1/n * \sum_{1}^{n} (Ai - Fi)^2$
Raíz do Erro Quadrático Médio	$\sqrt{1/n * \sum_{1}^{n} (Ai - Fi)^2}$
Ai representa o valor obtido e Fi representa o valor real	

Tabela 1: Métricas para avaliação da análise e previsão do consumo de água

2.5.2 Métodos de Análise e Previsão do Consumo de Água

A análise de dados têm sido, desde há muitos séculos, instrumento essencial à compreensão do mundo que nos rodeia. Com o avanço das técnicas estatísticas de análise de dados, é possível encontrar padrões e tendências em conjuntos de dados provenientes de muitas áreas distintas, como é o caso do consumo de água.

A análise de dados representa por isso uma mais valia na exploração de dados, pois permite que a partir da sua análise se retire conhecimento absoluto sobre padrões, estatísticas e tendências dos dados.

Em aplicações estatísticas, algumas pessoas dividem a análise de dados em três campos:

- estatística descritiva
- análise exploratória de dados
- análise confirmatória de dados

Existe uma variedade de técnicas e métodos que se podem aplicar quando se refere a análise exploratória de dados, nomeadamente pode resultar numa limpeza de dados e agregação de dados, mas também a aplicação de estatísticas descritivas, como médias, medianas, desvio-padrão, podem ser usadas para ajudar a perceber melhor os dados.

Outros modelos simples que se podem aplicar quando se pretende efectuar uma análise dos dados, são as correlações, que permitem identificar a relação entre as variáveis, assim como também os modelos de regressão e o clustering, que permite identificar grupos de dados com características semelhantes.

Ao contrário da análise de dados, a previsão é uma área onde existe um certa relatividade e uma componente de probabilidade, pois nem todas as previsões podem ser confirmadas para serem consideradas verdade.

A previsão baseia-se na análise dos dados para assegurar que os dados base têm um papel importante na qualidade e previsibilidade de um evento ou valor. Se os dados não possuem qualquer tipo de padrão então, a previsão é provável que seja quase aleatória ou semelhante aos valores observados.

Existem dois modelos de previsão, modelos qualitativos e modelos quantitativos. Neste trabalho apenas se vai fazer referência aos modelos quantitativos, uma vez que estes são baseados em dados e não em declarações. A utilização de modelos quantitativos, permite que se obtenha um resultado objectivo, o que é do maior interesse para a previsão de consumo de água.

Um modelo de previsão é uma expressão matemática, que tem um determinado número de variáveis independentes que servem como dados de entrada e uma variável dependente que é a saída do modelo ou previsão.

O modelo de previsão mais simples, no entanto poderoso, é o pressuposto de que o valor no ponto de tempo de previsão irá ser o mesmo que o valor real. Este modelo é designado de modelo de persistência. Outros métodos de previsão básicos podem ser classificados em duas categorias: métodos de análise de séries temporais e causais.

Os métodos causais baseiam-se no pressuposto de que o valor da variável dependente pode ser calculado através de uma função, passando o valor das variáveis independentes com parâmetro de entrada. Por sua vez, os métodos de séries temporais dependem de padrões históricos cíclicos, grande parte das vezes observados em dados do passado.

2.5.2.1 Análise Estatística

A *estatística* é o estudo da recolha, análise, interpretação, apresentação e organização dos dados.

Existem duas metodologias estatísticas principais que são usadas na análise de dados :

- estatística descritiva – resume os dados de conjunto de dados utilizando índices, tais como média e desvio-padrão

- estatística inferencial – tira conclusões a partir de dados que estão sujeitos a uma variação aleatória

Idealmente as estatísticas contêm dados sobre toda a população. As estatísticas descritivas são metodologias preocupadas com dois conjuntos de propriedades de uma distribuição: tendência central, que procura caracterizar o valor central ou típico da distribuição; dispersão, que caracteriza a extensão dos membros da distribuição a partir do seu centro. De uma forma geral, as estatísticas descritivas são usadas para resumir os dados da população. Para dados contínuos as representações numéricas incluem a média e desvio-padrão para descrever os dados, enquanto que para dados categóricos é utilizada a frequência e a percentagem para representar os dados.

Quando o conjunto de dados (população) não é viável, é escolhido uma parte do conjunto de dados para ser estudada, designada de amostra. Uma vez que, a representação da população está determinada, os dados são recolhidos para os membros da amostra num cenário observacional ou experimental. Neste cenário, estatísticas descritivas podem ser usadas para resumir os dados da amostra, no entanto uma vez que a amostra foi sujeito a uma componente de aleatoriedade, as representações numéricas feitas a partir da amostra contém um grau de incerteza. De forma, a obter conclusões significativas sobre os dados é necessário recorrer a estatística inferencial. A estatística inferencial usa padrões nos dados de amostra para efectuar inferências sobre a população representada pela amostra. As inferências sobre estatísticas matemáticas são efectuadas no âmbito da teoria da probabilidade, que trata a análise dos fenómenos aleatórios. As inferências utilizadas podem assumir a forma de: respostas a perguntas sim ou não sobre os dados (testes de hipóteses), estimar características numéricas dos dados (estimativas), descrever associações entre as variáveis existentes nos dados(correlação) e modelar relações entre as variáveis dos dados (regressão). As inferências podem ser utilizadas para extender a previsões ou estimativas de valores não observados, incluir a extrapolação e interpolação de séries temporais ou espaciais dos dados e também incluir data mining sobre os dados.

2.5.2.2 Clustering

De acordo com Vladimir Estivill-Castro a noção de “cluster” não pode ser definida com precisão, daí a razão de existirem diversos algoritmos de clustering [41], no entanto existe um denominador comum: um grupo de objectos de dados. A noção de cluster, encontrada por diferentes algoritmos, varia significativamente consoante as suas propriedades.

Um “clustering” é essencialmente um conjunto de “clusters”, que geralmente contêm todos os objectos de um conjunto de dados. Além disso, pode também especificar a relação dos “clusters” uns com os outros. “Clusterings” podem ser distinguidos aproximadamente como:

- hard clustering: cada objecto pertence a um “cluster” ou não
- soft clustering: cada objecto pertence a um “cluster”, com um determinado grau(por exemplo, uma probabilidade de pertencer a um “cluster”)

O termo análise de “clustering”, usado inicialmente por Tryon em 1939 [42], engloba uma série de

algoritmos e métodos distintos para agrupar objectos com características semelhantes em respectivas categorias. A questão geral colocada por investigadores em diversas áreas quando se fala em análise de "cluster", é como organizar os dados observados em estruturas significativas, ou seja, de forma a que seja possível desenvolver taxonomias.

A análise de "clustering" é um método de análise exploratória de dados, que tem como objectivo classificar diferentes objectos em grupos, de forma que o grau de associação entre dois objectos deve ser máximo se os objectos pertencem ao mesmo grupo e mínimo caso contrário. A análise de "clustering" permite que sejam descobertas estruturas de dados sem que seja necessário fornecer uma explicação ou interpretação da sua existência.

De seguida é apresentada uma visão geral dos métodos de análise de "clustering" mais importantes:

- **Clustering baseado em conectividade:** também designado por clustering hierárquico, baseia-se na ideia de que os objectos se encontram mais relacionados com os objectos mais próximos do que com os objectos mais distantes. Estes algoritmos ligam os objectos para formar os "clusters" com base na sua distância. Em diferentes distâncias, diferentes clusters são formados e podem ser representados através de um dendograma. Estes algoritmos não fornecem uma única partição do conjunto de dados, mas sim uma hierarquia de clusters que se fundem uns com os outros em determinadas distâncias. As principais escolhas conhecidas para este tipo de algoritmo são: single-linkage clustering(o mínimo da distância de objectos), complete linkage clustering(o máximo da distância de objectos) e UPGMA(Unweighted Pair Group Method with Arithmetic Mean, também conhecido por average linkage clustering).

- **Clustering baseado em centróides:** os clusters são representados por um vector central, que não é necessariamente um elemento do conjunto de dados. Quando o número de clusters é fixo, o algoritmo de clustering k-means, segue a seguinte definição como optimização do problema: encontrar os k centros do cluster e atribuir cada um dos objectos ao centro do cluster mais próximo, de modo a que as distâncias quadradas do cluster sejam mínimas. Um método particularmente conhecido é o algoritmo de Lloyd, principalmente referido como algoritmo K-means. A maioria dos algoritmos de K-means, exigem que o número de clusters k, seja conhecido com antecedência, o que representa uma das maiores desvantagens destes algoritmos. Além disso, os algoritmos de K-means preferem clusters com dimensão semelhante, uma vez o algoritmo irá sempre atribuir um objecto ao centróide mais próximo.

- **Clustering baseado em distribuição:** são os métodos de cluster mais relacionados com estatísticas. Os clusters podem ser facilmente definidos, como objectos provavelmente pertencentes à mesma distribuição. Um dos problemas chave destes métodos é o overfitting, a menos que sejam colocadas restrições na complexidade do método. Um método bastante conhecido é denominado de modelo de misturas Gaussianas, onde o conjunto de dados é normalmente modelado com um número fixo de

distribuições Gaussianas, que são inicializadas aleatoriamente e os seus parâmetros são iterativamente optimizados para se ajustarem melhor ao conjunto de dados. Clustering baseado em distribuição produz métodos complexos, para clusters que podem capturar a correlação e dependência entre atributos, no entanto estes colocam um peso extra o utilizador.

- **Clustering baseado em densidade:** os clusters são definidos como áreas de maior densidade do que o restante conjunto de dados. Os objectos mais dispersos, necessários para separar os clusters, são considerados pontos fronteira. O método mais popular baseado em densidade é denominado por DBSCAN(Density-based spatial clustering of applications with noise). Semelhante ao linkage based clustering, no entanto, ele só se conecta a pontos que satisfazem um critério de densidade, na variante original é definido como um número mínimo de outros objectos dentro desse raio. A desvantagem principal do DBSCAN é que ele espera que exista algum tipo de decréscimo na densidade para detectar as fronteiras entre clusters. Além disso, ele não consegue detectar estruturas de fragmentação intrínsecas, que são predominantes na maioria dos dados. A variação de DBSCAN, denominada por EnDBSCAN, permite detectar eficientemente esses tipos de estrutura.

2.5.2.3 Análise de Séries Temporais

Análise de Séries Tempo é um conjunto de técnicas aplicadas quando os dados possuem uma auto-correlação. Uma série temporal pode ser decomposta em três componentes: tendência, temporada e ciclo.

A tendência corresponde a um decréscimo ou aumento dos dados a longo prazo.

Uma série temporal tem uma componente sazonal, quando o seu comportamento tem um bloco de tempo definido relacionado com mudanças periódicas, por exemplo devido ao dia da semana. Uma mudança periódica maior na série temporal é denominada por ciclo, no entanto os seus períodos não fixos.

Auto-regressão é uma regressão de valores de séries temporais com as suas observações passadas e os seus valores futuros.

As propriedades de uma série temporal não dependem do momento em que a série temporal é observada.

De seguida, é apresentada uma visão geral dos métodos de análise de séries de tempo mais importantes com base em [43] :

- **Métodos Simples :** uma das abordagens mais básicas de séries de tempo é o simple moving averages, quando o valor da previsão é obtido pelo cálculo da média de alguns dos valores passados, atribuindo-lhe um peso igual. Uma abordagem semelhante fornecendo um peso diferente em cada nó, é denominada por weighted moving averages. No entanto, a abordagem de moving averages não permite seguir as tendências, encontrando-se sempre em “atraso” em relação aos valores reais.
- **Métodos de suavização exponencial e sazonal :** Single Exponencial Smoothing pondera automaticamente dados passados com pesos que diminuem exponencialmente ao longo do tempo.

Estes métodos podem modelar dados com uma tendência grande. Double Exponential Smoothing aplica duas vezes o Single Exponencial Smoothing. É útil quando a série de dados históricos não é estacionária. Holt Double Exponential Smoothing permite lidar com séries de tempo que não são estacionárias. Método Holt-Winter's Seasonal fornece três equações para modelação da sazonalidade e tendência com suavização.

2.5.2.4 Modelos de Regressão

Uma técnica importante utilizada para análise e principalmente previsão, que faz uso das relações entre variáveis dependentes e variáveis independentes e mais alguns parâmetros é denominada por análise de regressão.

Existem diferentes modelos de regressão que podem ser classificados consoante o tipo das variáveis dependentes e independentes. Se a variável dependente, depende apenas de uma única variável independente, o modelo é denominado por regressão simples, se a variável dependente é um escalar e a variável independente é um vector, o modelo é denominado por regressão múltipla, se a variável dependente e independente são ambas vectores, o modelo é denominado por regressão multi-variada.

O método dos quadrados mínimos apresentado por Gauss e a teoria de regressão de Pearson e Yule conduziram ao modelo de regressão moderno apresentado por R. A. Fisher em 1922 em [44].

De seguida os dados de entrada e os resultados de saída, são representados por Xt e Yt, onde *t* representa o ponto no tempo. Assumindo que, dado um conjunto de observações do passado uma das seguintes relações entre as variáveis dependentes e independentes pode ser usada, como apresentado em [45]:

- Regressão linear simples - é representado pela seguinte função:

  $Yt = m.Xt + b$, onde m e b são respectivamente o declive e intersecção da linha.

- Função exponencial:

  $Yt = m^{Xt}.b$

- Função potência:

  $Yt = Xt^{m}.b$

- Função logarítmica:

  $Yt = m.log(Xt) + b$

- Função logística:

  $Yt = 1/(c + m^{Xt}.b)$

- Função parábola:

$$Yt = c.Xt^2 + m.Xt + b$$

2.5.2.5 Métodos de Machine Learning

Machine Learning é um ramo da ciência que lida com a programação de sistemas de tal forma, eles aprendem e melhoram automaticamente com base na sua experiência. Neste contexto, aprendizagem significa reconhecer e compreender os dados de entrada e tomar decisões sábias com base nos dados fornecidos.

È muito difícil de fornecer decisões com base em todos os dados de entrada, por isso foram desenvolvidos algoritmos que permitem resolver este problema. Esses algoritmos constroem conhecimento a partir de dados específicos e experiências anteriores com os princípios das estatísticas, teoria da probabilidade, lógica, optimização combinatória, pesquisa, aprendizagem reforçada e teoria de controle.

Os algoritmos desenvolvidos formam a base de diversas aplicações, tais como:

- Processamento de Visão
- Processamento de Linguagem
- Previsão
- Reconhecimento de Padrões
- Data Mining
- Sistemas Inteligentes

Machine Learning é uma vasta área, e existem diversas maneiras de implementar técnicas de machine learning, no entanto as mais utilizadas e que vamos fazer referência são a aprendizagem supervisionada e não-supervisionada.

A aprendizagem supervisionada lida com a aprendizagem da função a partir de dados de treino disponíveis. Um algoritmo de aprendizagem supervisionada, analisa os dados de treino e constrói uma função inferida, que pode ser utilizada para mapear novos exemplos. Exemplos de aprendizagem supervisionada incluem:

- Support Vector Machine, Regressão Linear, Regressão Logística, Redes de Bayes, Análise Discriminante Linear, Árvores de Decisão, K-Neirest Neighbor algorithm e Redes Neurais.

A aprendizagem não-supervisionada faz sentido quando os dados não possuem qualquer tipo de label, ou seja, quando não existe um conjunto de dados predefinidos para treino. Aprendizagem não-supervisionada é uma ferramenta extremamente poderosa para análise de dados e pesquisa de padrões e tendências. È geralmente usada para clustering, de entradas de dados semelhantes em grupos lógicos. Exemplos de aprendizagem não-supervisionada incluem:

- Clustering, algoritmo de maximização da expectativa, método de momentos, Análise componentes principais, matriz de factorização não-negativa e decomposição de valor único.

Aprendizagem bacth ou offline refere-se a aprendizagem de uma função usando uma conjunto de treino apenas uma vez. Após o procedimento de aprendizagem a função(modelo), pode ser aplicada sobre os dados de teste para prever o seu label(classificação) ou valor(regressão).

Aprendizagem incremental ou online, por sua vez não inclui um procedimento de treino no inicio, em vez disso, permite actualizar a função(modelo) após cada novo dado na fase produtiva.

Outro modo, descrito e referenciado em [45] é denominado por aprendizagem bacth incremental, que representa a junção dos modos anteriores. Na aprendizagem bacth incremental, como os dados de treino são "buffered" à medida que chegam e o modelo é relido no modo de aprendizagem bacth, após o buffer de um determinado tamanho predefinido estar cheio. Após isso, o procedimento é reinicializado novamente com um buffer vazio.

De acordo com [45], os resultados de performance alcançados, no modo de aprendizagem batch incremental, são comparáveis com os resultados do modo de aprendizagem incremental pura.

De seguida são apresentados os métodos de machine learning mais importantes em análise e previsão:

- **Aprendizagem de modelos de regressão:** a ideia é aprender um modelo de regressão a partir de um conjunto de valores de treino. O algoritmo de aprendizagem tenta minimizar o erro entre o modelo e todos os valores do conjunto de treino.
- **Sistemas baseados em conhecimento:** foi desenvolvido em 1960, e geralmente utiliza um conjunto de valores referentes ao passado. Esses valores são adaptados, como se nova informação estivesse disponível e são usados na modelação da previsão do valor. Muitas vezes, este tipo de conhecimento é representado sob a forma de um sistema baseado em regras [46].
- **Fuzzy Logic:** os sistemas fuzzy logic podem ser muito robustos quando utilizados para previsão. As regras de um sistema fuzzy logic obtêm uma função inferida a partir de uma entrada fuzzy, que é mapeada para uma determinada saída. A semelhança entre os valores de entrada e os valores históricos é usada para prever o próximo valor. No sistema baseado em regras fuzy apresentado em [47], a obtenção de resultados de precisão são altamente compatíveis.
- **Artificial Neural Networks:** a invenção das artificial neural networks foi baseada na suposição de, como modelar o funcionamento do cérebro humano. A saída de uma ANN é uma combinação linear ou não-linear das suas entradas. Desenvolvido na segunda metade do século XX, as ANN rapidamente se tornaram uma das ferramentas mais populares de previsão e com uma área de investigação ainda em fase de desenvolvimento e optimização. No entanto é uma ferramenta de previsão amplamente utilizada [48] [46].
- **Support Vector Machine:** foram desenvolvidos utilizando os resultados da teoria de aprendizagem estatística e introduzido por Boser, Guyon e Vap-nik em 1992 [49]. Foi inicialmente desenvolvido para classificação e posteriormente adaptado para regressão. Desde a sua introdução, o método

SVM foi continuamente alterado e melhorado tornando-se numa ferramenta de machine learnig importante para aprendizagem de relações não-lineares e previsão.

- **Clustering:** clustering surgiu em 1932 na ramo da antropologia por Driver e Kroeber, e foi introduzido por Zubin em 1938 e Robert Tryon em 1939 [42] no campo da psicologia. Clustering representa a atribuição de um conjunto de observações em subconjuntos(clusters), de tal modo que as observações do mesmo cluster são semelhantes, de alguma forma. È uma técnica comum para análise estatística dos dados.

3 Data

3.1 Fonte de Dados

O conjunto de dados utilizado tem origem num simulador de contadores de água criado no âmbito do projecto, devido à inexistência de uma fonte de dados passível de ser utilizada. O simulador de contadores de água pretende simular a leitura dos registos de água ocorridos em contadores reais, de forma a que seja possível a utilização de um conjunto de dados com valores o mais próximo da realidade. A criação do simulador será descrita no próximo sub-capitulo.

Desta forma, o conjunto de dados tomado em consideração contempla cerca de 50 habitações do distrito de Aveiro e o período de recolha de dados utilizado foi de seis meses, de Novembro de 2014 a Junho de 2015.

Por forma a enriquecer os dados das leituras dos registos de água, e assim obter mais informação sobre o meio envolvente onde estas ocorrem, foi utilizado um ficheiro auxiliar que contém informação geográfica e demográfica de cada uma das habitações do conjunto de dados tomado em consideração.

Devido à natureza dos dados, as medições utilizadas possuem algumas limitações, principalmente no que se refere à inexistência de anomalias verificadas no consumo de água, inexistência de diferentes tipos de consumo consoante o período temporal, por exemplo diferentes tipos de consumo consoante a estação do ano e inexistência de períodos contínuos de consumo.

3.1.1 Simulador de Contadores de Água

Devido à inexistência de um conjunto de dados passível de ser utilizado, para responder ás necessidades do projecto, foi necessário proceder à criação de um simulador de dados que seja capaz de simular as leituras de consumo água, efectuadas pelos dispositivos de smart metering implementados na vida real.

Para isso foi necessário primeiramente efectuar uma análise das leituras produzidas por um contador de água, de forma a perceber qual a informação presente na leitura de água e qual a frequência com que as leituras são recolhidas. A partir desta análise, efectuada a um conjunto de dados exemplo fornecido pela PT Inovação, podemos concluir que as leituras de registos de consumo de água são recolhidas periodicamente num

intervalo de tempo que varia entre os 4 e os 6 minutos, e possuem informação relativamente ao identificador do contador(meter_id); ao momento de recolha da medida(meter_moment), que possui informação sobre a data e hora da recolha; ao tipo de medida utilizada(meter_quantity); ao valor do leitor registado no momento de leitura(meter_reading); ao factor de multiplicação usado para calcular o valor real do contador(meter_factor); ao valor real do contador de água(meter_value), calculado através da fórmula *meter_reading * power(10, meter_factor)* e à unidade de medida utilizada(meter_units). Na seguinte tabela pudemos observar a informação que é recolhida em cada momento de leitura:

Contador	meter_id
	meter_moment
	meter_quantity
	meter_reading
	meter_value
	meter_units

Tabela 2: Dados de leitura de consumo de água de um contador

Após efectuar a análise da estrutura e frequência da recolha das leituras de consumo de água, foi necessário proceder à identificação do cenário onde ocorre o processo de recolha de informação. Desta forma, foi definido qual a área de abordagem tomada em consideração para a simulação de recolha de leituras de consumo de água e identificado padrões de consumo de água existentes, de maneira a que o simulador criado, seja capaz de representar as leituras o mais próximo da realidade possível.

A área de abordagem definida, onde o simulador se encontra, foi o concelho de Aveiro, do distrito de Aveiro. Assim sendo foi necessário proceder à recolha de informação relativa à zona de actuação onde se encontra o simulador, nomeadamente no que diz respeito ao número de edifícios existentes na área, número de alojamentos familiares, número de famílias e população total.

Segundo o Instituto Nacional de Estatística (INE) [50], que tem como objectivo produzir e divulgar informação estatística oficial de qualidade, promovendo a coordenação, desenvolvimento e divulgação da actividade estatística nacional, a informação relativa aos Censos 2011 [51], permitiu que fosse retirada a seguinte informação relativamente ao concelho de Aveiro:

- número de edifícios - 22817
- número de alojamentos familiares – 40570
- número de famílias - 31142
- população – 78450

Para o desenvolvimento do simulador em questão, apenas foram tomados em consideração edifícios de alojamentos familiares com uma família, sendo excluídos todos os edifícios com mais que um alojamento familiar, edifícios industriais e públicos, como por exemplo centros de saúde, bibliotecas, câmaras

municipais, entre outros.

Desta forma, a informação recolhida é referente a 30240 habitações, que corresponde a 30240 contadores, uma vez que só foram tomados em consideração o tipo de edifícios referidos anteriormente.

Uma vez que a informação disponibilizada pelos Censos, não permite obter informação geográfica de cada uma das habitações, foi necessário recorrer à informação disponibilizada pelos CTT [52] e pela API Geocoding [53] da Google, por forma a obter informação geográfica relativa a cada contador.

Os CTT, disponibilizam três ficheiros, com informação relativamente aos códigos administrativos de todos distritos do país e respectiva descrição; códigos administrativos de todos os concelhos de cada distrito e respectiva descrição; códigos postais e respectiva informação relativamente a localidades, arruamentos e troços de rua e clientes CTT com código postal próprio(informação reduzida, uma vez que na maioria dos dados referente a este campo de encontra vazia). A API Geocoding disponibilizada pela Google, permite que seja efectuada a conversão de um endereço em coordenadas geográficas, que podem ser utilizadas para representar a localização de um contador num mapa.

Para a identificação de padrões de consumo existentes, foi necessário recolher informação relativa ao consumo médio por habitação, consumo médio mensal por pessoa, consumo médio dia por pessoa, assim como informação sobre as principais acções do dia-a-dia em que é consumida a água e qual o período de tempo do dia, em que essas acções ocorrem.

Segundo as fontes [54] [55], pudemos concluir a seguinte informação sobre o consumo de água em Portugal e mais concretamente, no distrito de Aveiro:

- consumo médio mensal por habitação – 15 m^3
- consumo médio mensal por pessoa – varia entre 5.2 e os 5.4 m^3
- consumo médio dia por pessoa – varia entre os 172 e os 180 litros

Relativamente, a informações sobre as principais acções do dia-a-dia praticadas pelas pessoas no consumo de água, a seguinte imagem fornecida pelas Águas da Região de Aveiro (AdRA) [54], fornece informação sobre as principais acções praticadas e respectiva percentagem que cada uma delas possui, no consumo total de água de uma habitação.

Figura 9: Acções do dia-a-dia no consumo de água

Assim sendo, pudemos concluir que existem 7 actividades chave que contribuem para o consumo de água no dia-a-dia, verificado em cada habitação, sendo elas: banhos e duches, WC, electrodomésticos, higiene pessoal, lavagem de automóveis e rega do jardim, preparação alimentar e consumo pessoal de água.

Uma vez reunida a informação necessária para o desenvolvimento do simulador de contadores de água, foi inicialmente criado um ficheiro com informação geográfica e demográfica, para cada uma das habitações presentes na área de abordagem definida. Para tal, e com o auxilio da informação presente nos ficheiros dos CTT e API Geocondig da Google, foi criado um ficheiro com os seguintes campos de informação, presentes na tabela seguinte:

Nome do campo	Descrição	Tipo
cod_distrito	código do distrito a que pertence a habitação	Numérico
cod_concelho	código do concelho a que pertence a habitação	Numérico
cod_localidade	código da localidade a que pertence a habitação	Numérico
nome_localidade	nome da localidade a que pertence a habitação	Texto
cod_postal	código postal a que pertence a habitação	Numérico
cod_postal_ext	extensão do código postal a que pertence a habitação	Numérico
nome_rua	nome da rua a que pertence a habitação	Texto
num_porta	número da porta da habitação	Numérico
num_agregado	número de pessoas presentes na habitação(agregado)	Numérico
lat	valor geográfico sobre a latitude	Numérico
long	valor geográfico sobre a longitude	Numérico

Tabela 3: Informação geográfica de cada habitação

A informação presente na tabela anterior, foi obtida com auxilio ás fontes enunciadas anteriormente, à excepção do ***num_agregado***, uma vez que essa informação não se encontra disponível para visualização. Para tal, e com base na informação sobre o número de famílias e população, retirada dos Censos 2011 [51], procedeu-se à criação de um número aleatório que varia entre 1 e 4, por forma a representar informação sobre um possível valor de agregado familiar de cada habitação.

De seguida foram definidos os períodos de tempo durante o dia, em que ocorrem as acções do dia-a-dia de consumo de água definidas anteriormente, e as respectivas frequências de ocorrência(número de vezes que uma determinada acção ocorre). Como tal, foram definidos 5 períodos de tempo onde ocorrem as seguintes acções:

- Período 1(7horas - 9horas) : duche e banhos; WC; higiene pessoal
- Período 2(12horas - 14horas) : preparação alimentar; WC; consumo pessoal
- Período 3(17horas - 19horas) : lavagem de automóveis e rega de jardim
- Período 4(19horas – 21 horas) : preparação alimentar; consumo pessoal

- Período 5(21horas – 23horas): electrodomésticos; higiene pessoal; WC

Foi definido um período de avaliação de 6 meses, onde se pretende que sejam simulados as leituras efectuadas pelos sensores inteligentes, em cada uma das habitações presentes no ficheiro gerado, com informação sobre cada uma das habitações.

Uma vez reunidos todos os elementos necessários para o desenvolvimento do simulador de contadores de água, foi criado um programa na linguagem Java que permite efectuar a simulação das leituras de água efectuadas pelos sensores de smart metering. A simulação das leituras é efectuada da seguinte forma:

- A partir da data de inicio do período de avaliação, para cada uma das habitações presentes no ficheiro gerado com informação sobre a habitação, são gerados ficheiros, com um intervalo de tempo aleatório que varia entre os 4 e os 6 minutos;
- Em cada um dos períodos de tempo onde ocorrem as acções definidos, é gerado um momento aleatório para cada uma das acções, que varia entre o momento inicial e final do período definido;
- Para cada uma das acções do dia-a-dia existentes em cada período, o valor do consumo de água é incrementado (consoante a percentagem de consumo que a acção tem no consumo total do dia-a-dia e consoante o agregado familiar da habitação), no momento aleatório gerado;
- Estas acções são efectuadas iterativamente para cada uma das habitações, até ao final do período de avaliação definido.
- Os ficheiros gerados contém a informação apresentada na tabela 2 para cada momento de leitura ocorrido

A implementação do simulador de dados pretende ao máximo representar a simulação das leituras efectuadas pelos dispositivos de medição inteligente, no entanto a simulação de eventos reais representa uma tarefa complexa e difícil de alcançar, pelo que, o simulador desenvolvido contém algumas limitações, quando comparado com aquilo que ocorre na realidade. De seguida são apresentadas as principais limitações do simulador, uma vez que estas se reflectem no desenvolvimento da solução e resultados obtidos. Uma das principais limitações do simulador é o facto dos dados relativos ás leituras efectuadas pelos dispositivos não ocorrerem em paralelo, ou seja, para o período de avaliação definido, os dados são gerados individualmente, para cada um dos contadores. Na realidade, os dados das leituras ocorrem em paralelo para todo o conjunto de dispositivos que estão a efectuar as medições. Outra das limitações do simulador, é o facto de uma acção ocorrer num determinado momento, em vez de se encontrar distribuída por um período de tempo, que é o que acontece na realidade. Por exemplo, a acção de tomar banho em média corresponde a um período de tempo de 10 a 15 minutos, pelo que segundo o que acontece na realidade deveria existir dois momentos de leitura pelo menos, em que o valor do consumo de água é incrementado, no entanto no simulador apresentado o consumo é incrementado num único momento de leitura. O facto de existir só um tipo de habitação implica também que não haja uma grande variação do consumo de água, e uma vez que o único factor que influencia

esse consumo é o agregado familiar, quando na realidade o consumo de água é influenciado por diversos factores, como a estação do ano, o tipo de habitação, a localização da habitação, entre outros. Esta abordagem seguida, implica também que não haja anomalias nos dados gerados, como por exemplo, fugas de água, ou consumo excessivos de água.

3.2 Estrutura dos Dados

Os dados utilizados no desenvolvimento do projecto, são representados sob a forma de uma String, que é composta por um objecto JSON. Cada leitura efectuada possui um conjunto de atributos fixos, que serão descritos na tabela seguinte:

Nome	Descrição	Tipo
meter_id	Identificador do contador	Texto
meter_moment	Momento de recolha da medida(data e hora)	Data
meter_quantity	Tipo de medida utilizada	Texto
meter_reading	Valor do leitor registado no momento de leitura	Numérico
meter_factor	Factor de multiplicação usado para calcular o valor real do contador	Numérico
meter_value	Valor real do contador	Numérico
meter_units	Unidade de medida utilizada	Texto

Tabela 4: Dados de uma leitura do contador de água

Os dados representam diversas séries temporais uni-variadas, sendo o valor da leitura a variável dependente e todas as restantes variáveis, as variáveis independentes. O intervalo de amostragem, é em média cinco minutos, para um período de amostragem de seis meses. Por cada intervalo de amostragem, os valores da leitura são armazenados num ficheiro distinto. De seguida, é apresentada informação sobre um sub-conjunto de ficheiros com o registo de leituras de 3 dispositivos:

“{“meter_id” : “381021431”, “meter_moment”: “2015/04/22 10:51:00”, “meter_quantity”: “Volume”, “meter_reading” : 1380, “meter_factor”: -3, “meter_units”: “m³” }”

“{“meter_id” : “381021432”, “meter_moment”: “2015/04/22 10:52:00”, “meter_quantity”: “Volume”, “meter_reading” : 830, “meter_factor”: -3, “meter_units”: “m³” }”

“{“meter_id” : “381021435”, “meter_moment”: “2015/04/22 10:55:00”, “meter_quantity”: “Volume”, “meter_reading” : 1110, “meter_factor”: -3, “meter_units”: “m³” }”

“{“meter_id” : “381021432”, “meter_moment”: “2015/04/22 10:56:00”, “meter_quantity”: “Volume”, “meter_reading” : 830, “meter_factor”: -3, “meter_units”: “m³” }”

“{“meter_id” : “381021431”, “meter_moment”: “2015/04/22 10:57:00”, “meter_quantity”: “Volume”, “meter_reading” : 1380, “meter_factor”: -3, “meter_units”: “m³” }”

“{“meter_id” : “381021435”, “meter_moment”: “2015/04/22 11:00:00”, “meter_quantity”: “Volume”, “meter_reading” : 1180, “meter_factor”: -3, “meter_units”: “m³” }”

3.3 Exploração dos Dados

Por forma a obter uma visão geral dos dados, foi considerada a estrutura dos ficheiros de dados anteriormente apresentada e efectuada uma visualização inicial dos registos de leitura do primeiro dia de medições, os primeiros 10945 ficheiros de dados. O intervalo de tempo entre dois registos de leitura para o mesmo dispositivo/medidor é aproximadamente em média 5 minutos. Por forma a avaliar o comportamento dos dados, importa explorar a frequência com que os dados são recebidos, assim como a distribuição do consumo ao longo do tempo.

Como pudemos observar a partir do sub-conjunto de dados apresentado no sub-capitulo anterior, os dados chegam de forma irregular não existindo um período de tempo fixo entre cada leitura, pelo que, em cada momento de leitura existe um número diferente de leituras recebidas. A figura seguinte permite verificar este facto:

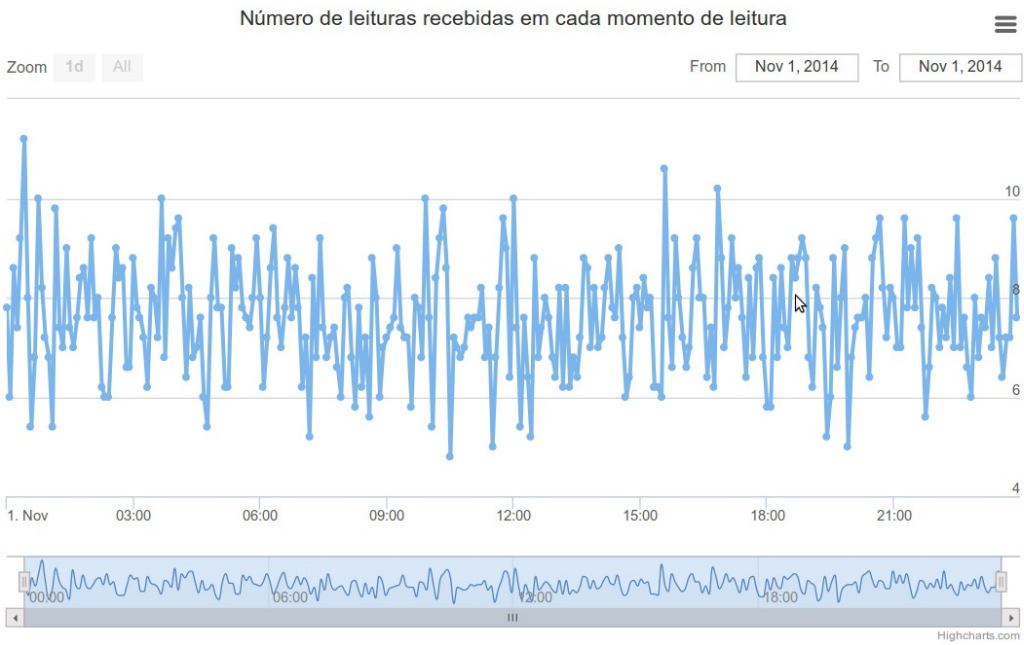

Figura 10: Número de leituras por momento de leitura

De seguida foi efectuada uma visualização da distribuição do valor de consumo de cada leitura ao longo do tempo. Uma vez que o valor dos identificadores dos dispositivos apresentam uma variação bastante grande

entre eles, a figura apresentada apenas contempla uma parte dos dados explorados por forma a obter uma compreensão mais clara.

Figura 11: Distribuição do consumo de água por contador

Como se pode observar a partir da figura, a distribuição do consumo de água não é regular para todos os contadores. Para o conjunto de dados explorado, cada dispositivo tem em média cerca de 290 registos por dia, pelo que pudemos concluir que existem muitos registos de leitura que possuem o valor de consumo de água igual, como já seria de esperar, uma vez que o consumo de água não se altera constantemente ao longo do dia. A existência de valores nulos deve-se ao facto de serem os primeiros registos de leitura efectuados e as medições terem sido iniciadas no período nocturno, onde o consumo de água é reduzido.

3.4 Modelação dos Dados

A estrutura de dados apresentada anteriormente no sub-capitulo 3.2, fornece uma excelente recomendação para o modelo de dados de um evento de leitura. No entanto existem alguns atributos desnecessários para a sua representação, nomeadamente, o meter_quantity, meter_factor, meter_value e meter_units, pelo que não foram tomados em consideração. O modelo de dados utilizado para representação de um registo de leitura foi o seguinte:

MeterReading
- meter_id: String
-meter_moment_year: Integer
- meter_moment_month: Integer
- meter_moment_day: Integer
- meter_moment_hour: Integer
- meter_moment_minutes: Integer
- meter_reading: Integer

Tabela 5: Modelo de dados de um registo de leitura

No que diz respeito à informação auxiliar, utilizada para enriquecimento dos dados de cada habitação, presente no ficheiro utilizado para criação do simulador de dados, esta não se baseia no identificador do medidor pelo que foi necessário proceder à criação de um identificador, para modelar os dados com base no identificador do medidor. Desta forma, o identificador do medidor é uma string composta pelos atributos num_post, num_post_ext e o num_porta presentes no ficheiro com informação de cada uma das habitações.

O modelo de dados utilizado para representação da informação auxiliar de cada medidor foi o seguinte:

MeterInformation
- meter_id: String
- cod_distrito: Integer
- cod_concelho: Integer
- cod_localidade: Integer
- nome_localidade: String
- nome_rua: String
- num_agregado: Integer
- lat: Double
- long: Double

Tabela 6: Modelo de dados da informação auxiliar de cada medidor

Para além da modelação dos dados anteriormente apresentadas aquando da aquisição dos dados, estes também tiveram que ser sujeitos a uma modelação na fase posterior a serem processados, por forma a que sejam armazenados segundo um modelo de dados que permita responder aos requisitos da solução. A modelação dos dados seguida, tem como base o modelo de dados do sistema de armazenamento utilizado, Apache Cassandra, anteriormente apresentado no sub-capitulo 2.3.1.1.2. Desta forma de seguida são apresentadas e descritas as diversas modelações de dados tomadas em consideração para o armazenamento e acesso dos dados.

A primeira modelação de dados apresentada é referente ao armazenamento dos dados inicialmente pré-processados para cálculo do valor agregado de consumo por mês, dia e hora. De seguida é apresentado o

modelo de dados para cada um dos valores agregados de consumo:

meter_id	***year, month1***	***year, month2***	***year,month3***	***year,month...***
	consume_month1	consume_month2	consume_month3	consume_month...

Tabela 7: Modelo de dados - Consumo agregado por mês

meter_id	***year, month, day1***	***year, month,day2***	***year,month,day3***	***year,month,day....***
	consume_day1	consume_day2	consume_day3	consume_day...

Tabela 8: Modelo de dados - Consumo agregado por dia

meter_id	***year, month, day, hour1***	***year, month, day, hour2***	***year, month, day, hour3***	***year,month,day, hour...***
	consume_hour1	consume_hour2	consume_hour3	consume_hour...

Tabela 9: Modelo de dados - Consumo agregado por hora

Os modelos de dados apresentados, segundo a representação do Cassandra, têm como super-chave o valor do identificador do dispositivo, a super-coluna possui o momento temporal do consumo agregado e o valor-coluna possui o valor de consumo agregado.

A utilização deste modelo de dados, tem como principal objectivo a optimização da performance no acesso à informação, uma vez que no que toca a escritas o Cassandra é extremamente eficiente. Visto que a informação se encontra armazenada sequencialmente quer ao nível da super-chave, quer ao nível da super-coluna, o acesso à informação é efectuado de forma bastante rápida, uma vez que o acesso efectuado também segue uma lógica sequencial. Por outro lado, o facto de se ter definido o identificador do dispositivo como super-chave também garante a optimização da performance no acesso à informação, uma vez que a maioria dos acessos efectuados são referentes a um identificador do dispositivo e não a vários identificadores. No caso, da maioria dos acessos à informação ser referente a vários identificadores de contadores, teria sido tomado em consideração, a super-chave como o momento de temporal do consumo agregado e a super-coluna como o identificador do dispositivo.

A segunda modelação de dados apresentada é referente ao armazenamento dos dados resultantes do métodos de análise estatística. De seguida é apresentado modelo de dados utilizado:

meter_id	average_consume, variance_consume

Tabela 10: Modelo de dados - Consumo médio mensal e variância por contador

O modelos de dados apresentado, segundo a representação do Cassandra, têm como super-chave o valor do identificador do dispositivo e o valor-coluna possui o valor de consumo médio e variância mensal.

A terceira modelação de dados apresentada é referente ao armazenamento dos dados resultantes do método de clustering, Kmeans. De seguida é apresentado o modelo de dados utilizado:

meter_id	***cluster***
	average_consume

Tabela 11: Modelo de dados - método de clustering

O modelo de dados apresentado, segundo a representação do Cassandra, têm como super-chave o valor do identificador do dispositivo, super-coluna o identificador do cluster a que foi atribuído e o valor-coluna o valor de consumo médio mensal.

A quarta modelação de dados apresentada é referente ao armazenamento dos dados resultantes dos métodos de previsão implementados. De seguida é apresentado o modelo de dados utilizado:

meter_id	***month1***	***month2***	***month...***
	predict_consume	predict_consume	predict_consume...

Tabela 12: Modelo de dados - métodos de previsão

O modelo de dados apresentado, segundo a representação do Cassandra, têm como super-chave o valor do identificador do dispositivo, super-coluna o mês para o qual foi previsto o consumo e o valor-coluna o valor de consumo mensal previsto.

4 Implementação

4.1 Infra-Estrutura

Para o desenvolvimento do projecto em questão foi utilizado um sistema operativo Linux, com uma memória RAM de 2 GigaBytes, 1CPU e 200GigaBytes de memória em disco. O core do protótipo foi escrito na linguagem Java, e a interface para visualização de resultados em HTML, que pode ser facilmente acedida através do browser.

Para tarefas que requerem um consumo intensivo de recursos, como é o caso, é necessário um poder computacional suficiente para que o seu desempenho seja efectuado com sucesso. Como referido anteriormente, tanto o Spark, como o Cassandra possuem como principal característica a alta escalabilidade, pelo que o poder computacional da infra-estrutura neste contexto é um aspecto a ter em conta.

O protótipo elaborado, foi desenvolvido com uma estrutura modular, pelo que os seus módulos podem ser facilmente convertidos num executável, ficheiro JAR, para posteriormente serem executados.

4.2 Método

Para alcançar os objectivos propostos no primeiro capítulo, foram necessárias diversas abordagens e iterações, por forma a alcançar uma solução que responda a esses mesmos objectivos. Desta forma, nesta secção irá ser efectuada uma descrição de todos os passos tomados em consideração, sendo os principais passos enunciados de seguida:

1. conhecer o funcionamento de análise de dados distribuída
2. processamento de dados em batch processing
3. armazenamento de dados
4. processamento de dados em streaming
5. implementação de algoritmos de machine learning
6. visualização de informação

Inicialmente numa primeira abordagem, e com o intuito de adquirir alguma informação sobre a construção e funcionamento de uma solução de Big Data, foi necessário adquirir competências no que diz respeito ao processamento de dados de forma distribuída. Dessa forma, na primeira abordagem tomada em conta foram construídos alguns programas para pré-processamento dos dados, com recurso ao modelo de programação Map Reduce disponibilizado pela framework Apache Hadoop [12].

A framework Apache Hadoop disponibiliza um sistema de ficheiros distribuídos(HDFS), como já referido anteriormente, que permite que sejam armazenadas grandes quantidades de dados. Assim sendo inicialmente, os ficheiros com os registos de leituras de cada dispositivo, gerados pelo simulador de dados foram armazenados num directório do HDFS para posteriormente serem processados. Os dados aqui utilizados, uma vez que se trata de uma fase experimental e de integração com a análise de processamento distribuído, são referentes a 10 dispositivos e possuem apenas um período de avaliação de uma semana.

Após o armazenamento dos ficheiros no HDFS, foi desenvolvido uma aplicação segundo o paradigma MapReduce para calcular os valores de consumo agregados por hora. Uma vez que a aplicação segue o modelo de programação MapReduce, esta encontra-se dividida em duas funções de processamento principais, denominadas de Map e Reduce.

A função Map, tem como parâmetro de entrada o directório do HDFS onde se encontram armazenados os ficheiros, com os registos de leitura. Na função Map é efectuada a leitura de cada um dos ficheiros e retirada informação sobre o identificador do dispositivo, momento em que a leitura ocorreu e valor de leitura ocorrido. A função Map tem como parâmetro de saída um par chave-valor, cuja chave corresponde ao identificador do dispositivo e o valor corresponde a um tuple composto pelo momento de leitura e valor de leitura, que por sua vez vai ser utilizado como parâmetro de entrada na função Reduce.

Na função Reduce, para cada chave do parâmetro de entrada, é calculado o valor máximo e o mínimo de consumo ocorrido em cada hora e efectuada a subtracção entre o máximo e o mínimo por forma a obter o valor do consumo agregado por hora. A função Reduce tem como parâmetro de saída um par chave-valor, cuja chave é o identificador do dispositivo e o valor corresponde a um tuple composto pelo valor da hora (data e hora) e respectivo consumo agregado dessa hora. A informação devolvida pela função de Reduce é armazenada num ficheiro de texto criado no sistema de ficheiros do Hadoop.

Após o desenvolvimento e exploração da ferramenta, conclui-se que esta apresenta algumas debilidades para a implementação da solução pretendida, nomeadamente, não é muito eficiente para casos de uso que requerem cálculos iterativos e algoritmos. Os dados de saída entre cada passo têm de ser armazenados no sistema de ficheiros distribuído, antes de se iniciar o próximo passo, o que implica que o sistema se torne mais lento devido à replicação e armazenamento em disco. Para além disso, o processo de configuração necessário para executar um programa é bastante complexo, e a execução de tarefas mais complexas implica o encadeamento de uma série de trabalhos MapReduce, que têm de ser executados sequencialmente, ou seja, não é possível iniciar um nova tarefa sem que a anterior tenha terminado.

Dado que a utilização da ferramenta Apache Hadoop não permite a realização optimizada dos objectivos que se pretendem alcançar, optou-se por explorar outra ferramenta utilizada no processamento de grandes quantidades de dados, denominada por Apache Spark. O Apache Spark é uma ferramenta poderosa para processamento de grandes quantidades de dados, e tem como principais referências nas suas características o facto de executar aplicações, 100 vezes mais rápido em memória e 10 vezes mais rápido em disco do que o MapReduce do Hadoop. Pelo que a sua escolha foi tomada em consideração para a construção da solução em causa.

Nesta segunda iteração procedeu-se ao desenvolvimento de um programa para pré-processamento dos dados com recurso à ferramenta Apache Spark. A primeira abordagem tomada em consideração para o processamento de dados segue o modo de processamento “batch processing”. O “batch processing” permite de forma eficiente processar grandes quantidades de dados, onde os dados são processados em grupo, para um determinado período de tempo. Os dados são recolhidos, processados e de seguida são produzidos resultados em “batch”. “Batch processing” requer a utilização de programas separados para entrada, processamento e saída de dados.

Nesta iteração, os dados são lidos a partir do directório do HDFS, que contém os ficheiros com os registos das leituras de cada dispositivo. O Apache Spark disponibiliza, uma biblioteca denominada de Spark SQL que permite efectuar a leitura de conjuntos de dados, provenientes de diversas fontes.

Uma vez que os dados presentes em cada ficheiro seguem a estrutura de um objecto Json, o Spark SQL permite inferir automaticamente o esquema do conjunto de dados e carregá-lo como um JavaSchemaRDD. Esta conversão é feita através do método jsonFile() presente no JavaSQLContext.

Após esta operação, o esquema inferido foi registado como uma tabela temporária e a partir deste momento podem ser executadas operações SQL, directamente à tabela temporária denominada de “meterinformation”. A tabela “meterinformation” possui a seguinte modelação de dados:

MeterInformation
- meter_id: String
-meter_moment_year: Integer
-meter_moment_month: Integer
-meter_moment_day: Integer
-meter_moment_hour: Integer
-meter_moment_minutes: Integer
-meter_reading: Integer

Tabela 13: Modelo de dados da tabela temporário MeterInformation

Visto que os dados são representados ao longo do tempo, esta é uma boa representação para efectuar posteriormente uma análise dos dados com base em séries de tempo. No entanto o intervalo de tempo entre cada leitura é muito pequeno, aproximadamente 5 minutos, por isso é necessário proceder à criação de

granularidades de tempo do perfil de consumo maiores, para que posteriormente se proceda à análise de informação. Como tal, de seguida vão ser calculadas três granularidades de tempo, nomeadamente, hora, dia e mês.

Por forma a calcular o valor do consumo agregado por hora, dia e mês para cada um dos contadores, foram executadas as seguintes operações de pesquisa à tabela:

- Cálculo do valor do consumo agregado por hora

  Pesquisa: *"SELECT meter_id, year, month, day, hour, sub(max(consume),min(consume)) FROM meterinformation GROUP BY id, year, month, day, hour;"*

- Cálculo do valor do consumo agregado por dia

  Pesquisa: "*SELECT meter_id, year, month, day, sub(max(consume),min(consume)) FROM meterinformation GROUP BY id, year, month, day;*"

- Cálculo do valor do consumo agregado por mês

  Pesquisa: "*SELECT meter_id, year, month, sub(max(consume),min(consume)) FROM meterinformation GROUP BY id, year, month;*"

Uma vez que os valores dos consumo agregado já foram calculados, importa agora pensar como armazenar os dados por forma a que este armazenamento seja efectuado de forma eficiente, tanto ao nível das escritas, como ao nível das leituras, para que mais tarde sejam utilizados na construção dos métodos de análise e previsão e visualização de informação.

Para efectuar o armazenamento dos dados, a ferramenta utilizada foi o Apache Cassandra. O Apache Cassandra devido ás suas características permite que sejam armazenados e geridos grandes volumes de dados sem um único ponto de falha, é altamente escalável, permite que sejam criadas e eliminadas tabelas rapidamente e de forma eficiente, e é altamente indicado para escritas devido às suas características.

No contexto do método que se pretende implementar o Apache Cassandra é uma óptima escolha para o armazenamento dos dados, uma vez que os dados que se pretendem armazenar têm como base séries temporais. Os dados de séries temporais encontram-se organizados de forma sequencial, e uma vez que o armazenamento de dados efectuado pelo Cassandra também é sequencial, esta é uma boa solução para o problema. Visto que os dados são escritos sequencialmente na base de dados, estes posteriormente também são lidos de forma sequencial, o que representa uma vantagem para o sistema a implementar. Para além das vantagens de utilização do Cassandra como sistema de armazenamento já enunciadas , o Cassandra apresenta outras características que foram tomadas em conta na escolha da ferramenta, como: permite que sejam efectuadas um número de operações de escrita/leitura bastante superiores, quando comparada com outros sistemas de gestão e armazenamento, como o Apache Hbase e o MongoDB; ajuda a garantir que a latência é mitigada em grande escala, o que normalmente não se verifica noutros sistemas, pois a latência tende a aumentar com o aumento do fluxo de carga de trabalho [56].

O armazenamento dos dados foi efectuado com recurso ao Spark Cassandra Connector, que permite que seja feita diversas operações, como: leitura de dados do Cassandra para o Spark, escrita de dados do Spark para o Cassandra, conversão de tipos implícitos e mapeamento de objectos e exposição de tabelas Cassandra como Spark RDDs + Spark Dstreams. De seguida é apresentada uma figura que ilustra como os dados são escritos do Spark para o Cassandra:

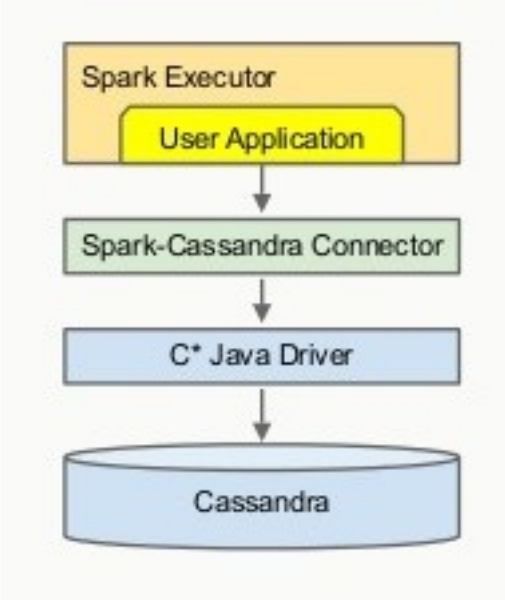

Figura 12: Spark cassandra connector

O modelo de dados utilizado para o armazenamento de dados dos valores de consumo agregado calculados, foi o modelo de dados anteriormente apresentado no sub-capítulo 3.4.

Nesta segunda iteração, os valores de tempo de execução ocorridos são bastante melhores do que os verificados na primeira iteração, como já era de prever devido ás características da ferramenta utilizada nesta iteração. No entanto existem algumas limitações, na utilização da abordagem “batch processing” tomada em consideração nomeadamente, no que diz respeito à impossibilidade de se efectuar uma análise em tempo-real e por consequente a impossibilidade de tomar medidas no imediato.

No contexto em que se enquadra o projecto, um dos objectivos principais é o acesso a informação de tal forma, que seja possível ter uma visão necessária do que está acontecer, para que possam ser tomadas decisões com prudência no momento certo. Pelo que a abordagem “batch processing” tomada, neste contexto não permite que essas decisões sejam tomadas no imediato, uma vez que os dados são processados em intervalos de tempo bastante grandes. Exemplo disso, é o facto de para o cálculo dos valores de consumo agregado por hora, os dados recebidos na última hora serem processados apenas quando estiverem reunidos todos os registos relativos a essa hora, pelo que a informação processada da última hora, só pode ser tomada em conta para efectuar decisões, no final de cada hora. Este exemplo repercute-se para o caso do cálculo dos

valores de consumo agregado por dia e mês.

Outra das limitações desta abordagem, reflecte-se no facto de o volume de dados para processamento ser demasiado extenso em cada intervalo de tempo, uma vez que vão ser processados os dados de todos os contadores em cada intervalo de tempo(hora,dia,mês), o que implica um aumento da latência nesta operação. Para além, das desvantagens já referidas, o facto de existir uma elevada redundância dos dados, uma vez que a informação se encontra replicada, foi decidido tomar em consideração uma nova abordagem que permiti-se resolver os problemas da abordagem anterior.

A nova abordagem tomada em consideração para processamento de dados segue o modo de processamento "streaming processing". O "streaming processing" envolve uma entrada, processamento e saída de dados contínua, e os dados devem ser processados num período de tempo pequeno ou quase em tempo-real.

Para proceder ao desenvolvimento desta abordagem foi necessário inicialmente efectuar umas pequenas alterações ao nível do simulador de dados, para que este possibilitasse que os registos das leituras dos contadores fossem enviados continuamente sob a forma de mensagens ou eventos, para posteriormente serem processados com recurso à ferramenta Spark Streaming.

Desta forma, foi necessário recorrer à utilização da ferramenta RabbitMQ para que as leituras geradas pelo simulador fossem enviadas como mensagens, a simular as leituras efectuadas pelos dispositivos de smart metering na vida real.

O RabbitMQ é um software para troca de mensagens ("message broker"), que usa o protocolo AMQP (Advanced Message Queuing Protocol) [57]. O servidor RabbitMQ é escrito em Erlang e foi construído sobre a plataforma OTP(Open Telecom Plataform) para clustering e failover. O RabbitMQ oferece um sistema de mensagens confiável, altamente disponível, escalável e com uma taxa de transferência e latência previsível e consistente. Resumidamente o RabbitMQ, é composto por três agentes principais, sendo eles: o "producer", que envia as mensagens; a "queue", que armazena as mensagens; o "consumer", que recebe as mensagens. A imagem seguinte, ilustra o diagrama de fluxo básico do RabbitMQ:

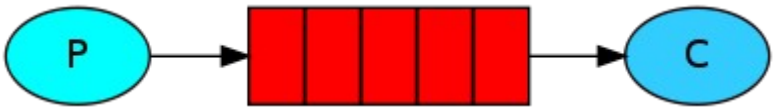

Figura 13: Diagrama RabbitMQ

Deste modo, foi necessário proceder à construção de dois programas, um para enviar as mensagens e outro para receber as mensagens, denominados respectivamente de NewTask e Worker. Existem inúmeras abordagens, que podem ser seguidas quando se pretende desenvolver um processo de troca de mensagens, nomeadamente, a simples implementação de um processo para enviar e receber mensagens; a utilização de filas de trabalho, para distribuir tarefas por diversos trabalhadores; a utilização de um processo de "publish/subscribe", que permite enviar diversas mensagens de uma só vez; a utilização de um processo de

“routing”, para receber mensagens de forma selectiva; utilização de um processo baseado em tópicos, que permite receber mensagens com base num padrão.

Para o desenvolvimento do processo de troca de mensagens deste projecto, a abordagem seguida foi a utilização de filas de trabalho, visto que as mensagens que se pretendem enviar seguem todas o mesmo padrão e o principal objectivo é o envio de um fluxo de mensagens elevado, em que se pretende evitar a perda de dados em caso de falha do sistema.

A ideia principal da utilização de filas de trabalho é para evitar a perda de informação e garantir que uma tarefa é sempre executada. A mensagem é encapsulada e enviada para a fila de trabalho. O processo de trabalho em execução irá receber as mensagens e executá-las.

No programa para envio de mensagens, NewTask, a mensagem é enviada como uma String, contendo a informação da leitura do contador. Uma vez que os registos das leituras não são efectuadas com um período de tempo constante e por isso o simulador não envia as mensagens consoante o intervalo de tempo ocorrido entre cada leitura, foi necessário definir um intervalo de tempo para as leituras serem enviadas. O intervalo de tempo definido foi de dois segundos.

Uma das vantagens de utilização de uma fila de trabalho é a capacidade de executar trabalho em paralelo, pois permite que sejam adicionados mais trabalhadores e assim escalar facilmente, o que é bastante útil pois assim pode-se dividir as mensagens por diversos trabalhadores, quando a carga de mensagens é muito elevada. No entanto, ainda é necessário garantir que as mensagens são todas processadas, e que no caso de falha do trabalhador ou servidor do RabbitMQ estas não são perdidas.

Para o caso de um trabalhador falhar, o RabbitMQ suporta um processo de confirmação de mensagens. Quando uma mensagem é recebida e processada o trabalhador envia uma confirmação para o RabbitMQ a comunicar que a mensagem foi processada e o RabbitMQ pode excluí-la. Assim se um trabalhador falhar sem enviar a confirmação, o RabbitMQ entende que a mensagem não foi processada e vai enviá-la novamente para outro trabalhador, garantindo que nenhum mensagem é perdida.

Para o caso do servidor RabbitMQ falhar, é necessário garantir que o servidor não esquece a fila e as mensagens. Assim sendo foi necessário garantir que as mensagens não são perdidas e para isso, a fila e as mensagens foram definidas como duráveis. Para além disso, também foi necessário definir as mensagens como persistentes.

Desta forma garante-se que se existir uma falha tanto ao nível do servidor como dos trabalhadores as mensagens não são perdidas e são processadas na sua totalidade.

Após o processo de troca de mensagens estar em funcionamento, ou seja os registos de leitura dos contadores serem enviados e recebidos sem que haja nenhuma perda de informação, de seguida foi efectuado o processamento do fluxo de mensagens recebido. Para isso foi utilizado o Spark Streaming.

O Spark Streaming é uma extensão do Spark que permite o processamento de fluxo dados com alto

rendimento, escalável e tolerante a falhas. Os dados podem ser consumidos a partir de diversas fontes, como o RabbitMQ, Kafka, Flume, Twiter, entre outros e processados através de algoritmos complexos, expressos com funções de alto nível como map, reduce, state, join e window. Os dados processados podem ser armazenados em ficheiros, base de dados ou apresentados em sistemas de visualização de informação de tempo-real.

Internamente, o Spark Streaming funciona da seguinte forma: o Spark Streaming recebe o fluxo de dados como entrada, divide os dados em pequenas partes e são processados pelo motor Spark para gerar o fluxo final de resultados em pequenas partes. A imagem seguinte representa o seu funcionamento:

Figura 14: Fluxo de dados spark streaming

O Spark Streaming fornece uma abstracção de alto nível denominado de fluxo de dados discretos (DStream), que representa um fluxo de dados contínuo. Os Dstreams podem ser criados a partir fluxo de dados de diversas fontes e internamente são vistos como uma sequência de RDDs.

Para o desenvolvimento desta abordagem, inicialmente foi criado um novo programa, que é inicializado através de criação de um objecto StreamingContext, que é o ponto de entrada de todas as funcionalidades de Spark Streaming. De seguida procedeu-se à definição do intervalo de tempo em que os dados são processados. O intervalo de tempo definido foi de um segundo. Visto que as mensagens são enviadas com um intervalo de tempo de dois segundos, isto permite que os dados sejam processados tão rápido quanto eles são recebidos.

Após definir o intervalo de tempo com que os dados são processados, procedeu-se à implementação de um receptor de dados personalizado para receber os dados do RabbitMQ. Para isso, apenas foi necessário que a classe Worker desenvolvida anteriormente para receber os dados enviados do RabbitMQ extenda da classe abstracta Receiver, fornecida pelo Spark Streaming, e implemente os seus métodos abstractos onStart() e onStop().

Depois da aplicação já ter capacidade de receber os dados dos registos das leituras dos contadores, iniciou-se o processamento dos dados para efectuar o cálculo dos valores de consumo agregado por hora, dia e mês. Desta forma, primeiramente os dados recebidos em formato String, contendo um objecto Json com os registos das leituras, foram analisados e retirados os dados necessários para o cálculo do agregado, ou seja, o identificador do dispositivo, momento em que a leitura ocorreu e valor de leitura. Esta operação foi

efectuada com recurso à função de alto nível mapToPair, que tem como parâmetro de entrada as String recebidas pelo receptor e como parâmetro de saída um JavaPairDStream composto por uma String(identificador do dispositivo), e por um objecto do tipo MeterInfoAux(momento em que a leitura, valor de consumo da leitura) criado para o efeito.

De seguida, foi utilizada a função updateStateByKey fornecida pelo Spark Streaming. Esta operação permite manter o estado arbitrário enquanto novas informações são actualizadas continuamente. Desta forma, é possível manter o estado anterior de cada uma das chaves(identificador do dispositivo) e ao mesmo tempo receber nova informação. O diagrama seguinte demonstra o funcionamento da função updateStateByKey:

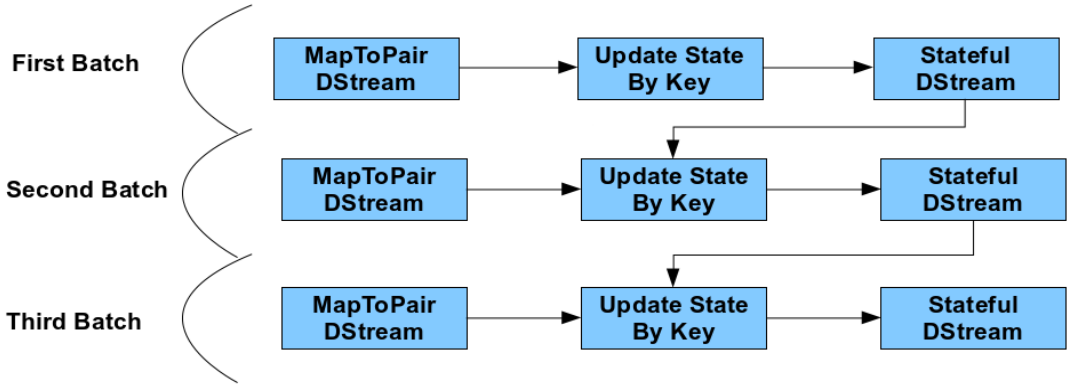

Figura 15: Funcionamento da função updateStateByKey

A função updateStateByKey tem como parâmetro de entrada uma lista de objectos do tipo MeterInfoAux, que contêm a informação das novas leituras ocorridas no período de tempo definido e um objecto também do tipo MeterInfoAux, que contém os valores do estado anterior. Assim nesta função é efectuada a subtracção entre o valor do consumo da nova leitura recebida e o valor do consumo da leitura anterior. Para efectuar o cálculo dos valores de consumo agregado por hora, dia e mês é mantida a informação sobre o valor acumulado da subtracção para cada uma das agregações(hora, dia e mês). No caso da hora, dia ou mês do novo momento de leitura ocorrido ser diferente da hora, dia ou mês do momento de leitura anterior o valor acumulado da subtracção é colocado a zero.

Para evitar que a informação calculada esteja sempre a ser escrita para a base de dados, uma vez que grande parte das vezes o valor acumulado da subtracção não se altera, foi criado um atributo booleano para identificar se o valor acumulado da subtracção actual é diferente do valor acumulado da subtracção anterior. Desta forma, garante-se que posteriormente a informação calculada só é escrita na base de dados se o valor acumulado da subtracção for diferente do anterior e evita-se que a informação esteja a ser sempre escrita por cada nova entrada na função updateStateByKey.

A função updateStateByKey tem como parâmetro de saída um objecto do tipo MeterInfoAux (momento de leitura, valor de consumo de leitura, valor acumulado da subtracção para a hora, valor acumulado da

subtracção para o dia, valor acumulado da subtracção para a mês, identificador se o valor acumulado da subtracção é diferente do anterior).

A utilização da função updateStateByKey implica a configuração de um directório de ponto de verificação (checkpoint). A configuração de um directório checkpoint é necessária uma vez que a informação do estado anterior está a ser utilizada para calcular a informação do novo estado, o que implica que o tamanho da cadeia de dependência a manter aumente com o tempo. Para evitar esse aumento sem limites no tempo de recuperação, os RDDs das transformações de estado são armazenados periodicamente, para cortar as cadeias de dependência. Desta forma, foi necessário um definir um directório no HDFS (uma vez que este é confiável e tolerante a falhas) para que as informações de verificação sejam armazenadas.

Para além disso, a criação do directório checkpoint permite que a aplicação seja capaz de recuperar de falhas, uma vez que para isso bastou efectuar algumas alterações na aplicação, para que o seu comportamento siga as seguintes cláusulas:

- Se a aplicação é iniciada pela primeira vez, é criado um novo StreamingContext
- Se o programa é reiniciado após falha, então este irá recriar o StreamingContext a partir dos dados presentes no directório de checkpoint

Esta solução é uma mais valia visto que a aplicação construída tem um período de operação de 24 horas sob 7 dias, e portanto deve ser tolerante a falhas não relacionadas com a lógica da aplicação.

Uma vez que a informação já está calculada, de seguida é armazenada para o Cassandra segundo o modelo de dados apresentado no sub-capitulo 3.4. Como tal, para cada resultado retornado pela função updateStateByKey é verificado se o valor do identificador, que indica se o valor acumulado da subtracção é diferente do anterior, é verdadeiro, e nesse caso a informação do identificador do dispositivo, data do valor de consumo agregado e valor de consumo agregado, contida no resultado é escrita para o Cassandra.

Esta abordagem permite que o acesso à informação seja quase em tempo-real, uma vez que esta também é processada quase em tempo-real. Desta forma, é possível obter uma visão dos valores de consumo ocorridos no imediato e assim proceder à tomada de decisões. No contexto do projecto este ponto não se encontra abrangido, mas por exemplo a utilização de processamento em tempo-real seria útil para a detecção de fugas de água, permitindo assim tomar uma decisão de imediato, no caso da sua ocorrência. Através da abordagem “batch”, esta detecção também poderia ser efectuada, no entanto a tomada de decisão não seria no imediato.

Ou seja, para proceder à definição do modo como os dados devem ser processados, primeiramente deve ser efectuada uma análise das necessidades que a solução pretende colmatar, e de seguida tomada a decisão sobre o modo de processamento adequado para implementação da solução especifica, uma vez que ambos as abordagens possuem vantagens e desvantagens.

Nos dias de hoje, são utilizadas abordagens que recorrem à utilização dos dois modos de processamento para a implementação de uma solução. Este facto deve-se essencialmente à necessidade de existência de soluções

capazes de fornecer informação em tempo-real, no entanto esta abordagem também é mais susceptível à perda de informação, uma vez que a informação é processada sob a forma de fluxo de dados, existe uma maior probabilidade de existir falhas no sistema, por mais que este seja tolerante a falhas. Para além disso a informação processada sob a forma de fluxo de dados, também é mais receptível à existência de erros nos valores de cálculo. Pelo que a utilização em paralelo do modo de processamento batch, permite em grande parte colmatar estas lacunas, e assim obter uma informação de melhor qualidade e ao mesmo tempo fornecer informação quase em tempo-real.

Neste trabalho, a solução apresentada pretende de certo modo seguir o raciocínio anteriormente descrito.

Nos sub-capítulos seguintes são descritos em detalhe a arquitectura do sistema proposto e os passos 5 e 6 do sub-capitulo 4.1.

4.2.1 Arquitectura

De seguida será apresentada uma visão geral da arquitectura e fluxo de dados do método implementado. A arquitectura é representada através das diversas camadas lógicas definidas para o desenvolvimento da solução, por forma a ilustrar os vários componentes utilizados, que pretendem responder aos requisitos funcionais e não-funcionais, até agora enunciados, da solução proposta. As camadas lógicas permitem simplesmente organizar os componentes da solução, que efectuam as funções especificas da solução. A solução apresentada contempla as seguintes camadas:

- Fonte de dados
- Camada de tratamento e armazenamento de dados
- Camada de análise
- Camada de consumo

A imagem seguinte representa o diagrama da arquitectura do solução proposta:

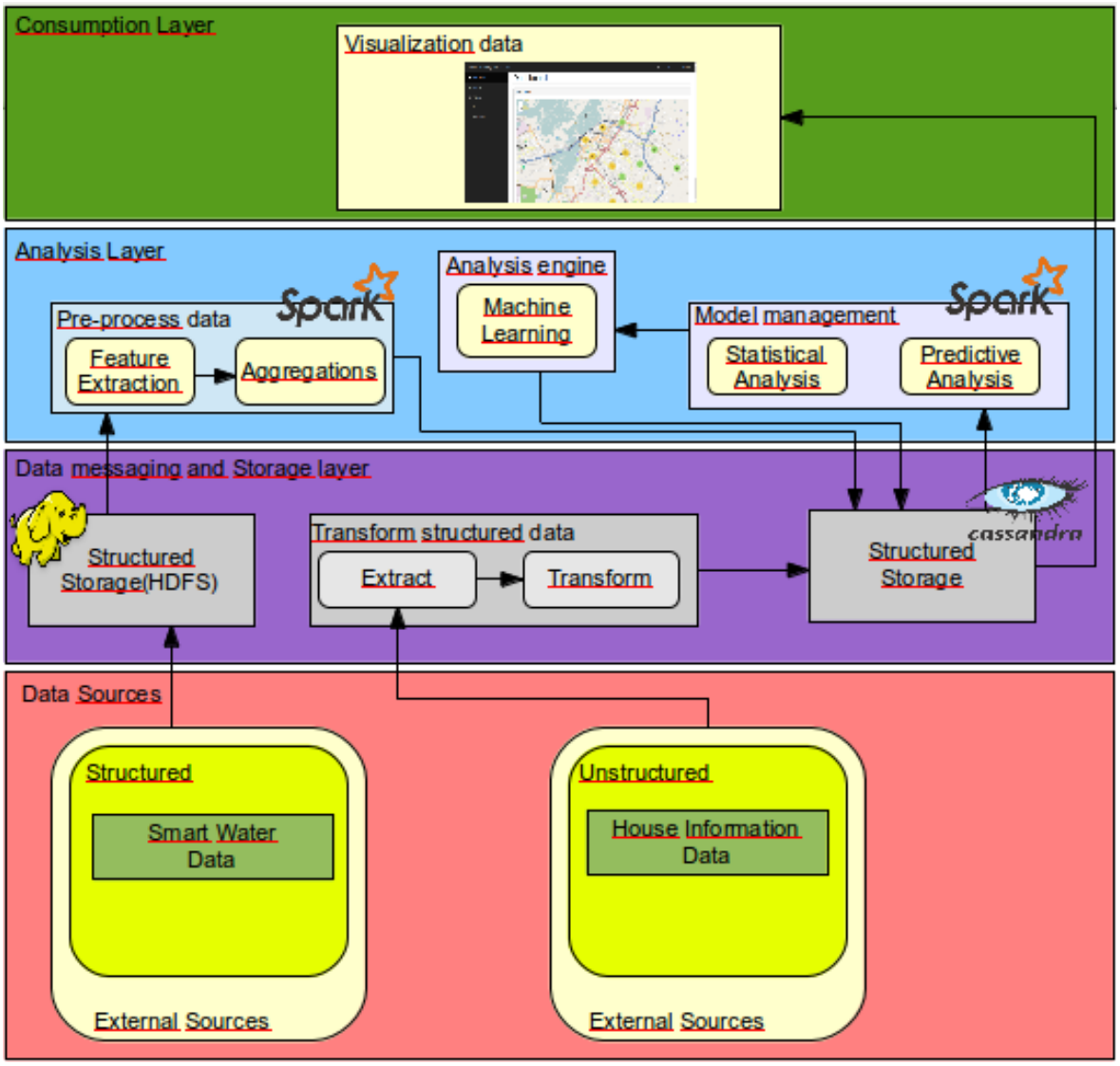

Figura 16: Diagrama da arquitectura da solução

Como representado no diagrama os dados são adquiridos a partir de fontes externas, sendo estes divididos em duas categorias, dados estruturados e dados não-estruturados. Os dados estruturados são provenientes de dispositivos inteligentes, que fornecem dados sobre as leituras de contadores de água, efectuadas pelos mesmos. Os dados não-estruturados são provenientes de fontes externas, e tem como principal objectivo o enriquecimento dos dados estruturados. Os dados não-estruturados contêm informação geográfica e demográfica das habitações incluídas na área de abordagem.

Nos dados estruturados são aplicadas duas abordagens distintas, sendo elas o processamento batch e o processamento streaming. Na abordagem batch os dados são adquiridos e armazenados no sistema de armazenamento estruturado Hadoop Distributed File System (HDFS), e posteriormente são pré-processados periodicamente. O pré-processamento efectuado calcula o valor de consumo agregado por hora, dia e mês,

por forma a obter diferentes granularidades de tempo do perfil de consumo, que posteriormente serão utilizadas na camada de análise. Após o pré-processamento estar concluído, os dados são armazenados no sistema de armazenamento de dados estruturado Apache Cassandra. Na abordagem streaming os dados adquiridos, são directamente enviados para a camada de análise, através do RabbitMQ, por forma a que seja efectuado o pré-processamento dos dados. O pré-processamento efectuado é o mesmo que na abordagem batch, mas com uma lógica diferente, uma vez que neste caso os dados de entrada são recebidos sob a forma de fluxo de dados. De seguida, os dados resultantes do pré-processamento são armazenados no sistema de armazenamento dados estruturado Apache Cassandra.

Os dados não-estruturados adquiridos são extraídos e transformados em dados estruturados, por forma a que estes sigam o modelo de dados desejado e de seguida armazenados no sistema de armazenamento de dados estruturados.

Os dados armazenados são posteriormente utilizados na camada de análise, para que sejam aplicados mecanismos de aprendizagem, e efectuados diversos tipos de análise e previsão do consumo de água. Depois de efectuada a análise e previsão dos dados, os resultados da aplicação dos métodos utilizados para o efeito, são armazenados no sistema de armazenamento de dados estruturados.

Por fim na camada de consumo, os dados armazenados no sistema de armazenamento são apresentados para visualização através de mapas, gráficos e tabelas numa página Web.

4.2.2 Métodos de Análise e Previsão

A criação de métodos de análise e previsão de informação é um campo importante no consumo de água. Através da implementação destes métodos torna-se possível adquirir informação sobre os dados do consumo de água.

Para a implementação dos métodos de análise e previsão, foi utilizada a biblioteca, Spark MLlib, disponibilizada pelo Spark. O Spark MLlib é uma biblioteca de machine learning escalável, composta por algoritmos de aprendizagem comuns e utilitários, incluindo classificação, regressão, clustering, filtragem colaborativa, redução de dimensionalidade, bem como optimizações primitivas.

De seguida são apresentados e descritos de forma detalhada os métodos implementados.

4.2.2.1 Estatística

O primeiro método a ser implementado pretende efectuar o cálculo de algumas estatísticas básicas, que servirão de apoio para implementação dos métodos seguintes e para visualização de informação estatística.

Assim sendo o primeiro passo a ser efectuado, foi o cálculo da média e variância do consumo mensal de cada um dos contadores. Para isso, foi utilizada a função colStats() disponível através da classe Statistics do Spark MLlib. A função colStats() retorna uma instância de resumo estatístico multi-variado, que contém o máximo,

mínimo, média, variância, número de elementos diferentes de nulo e contagem total.

Inicialmente foi efectuada uma pesquisa à base de dados, para recolha dos dados de consumo mensal de cada um dos contadores, através do módulo Spark SQL. De seguida os dados foram passados como parâmetro de entrada da função colStats() e armazenada a informação do consumo médio e variância de cada um dos contadores, numa tabela da base de dados.

O segundo método implementado foi o método de correlações. O cálculo de correlações entre séries de dados permite identificar o valor de correlação entre as séries de dados. Desta forma é possível identificar quais as variáveis que têm maior influência no consumo de água. Para isso, foi utilizado a função corr() disponível através da classe Statistics do Spark MLlib.

Primeiro foi efectuada uma pesquisa à base de dados, para recolha dos dados de consumo mensal de cada um dos agregado familiar existentes. De seguida foi calculada a média de consumo mensal por agregado familiar, através do método anteriormente criado. Os dados do consumo médio mensal por agregado familiar e os dados do agregado familiar são passados como parâmetro de entrada da função corr(), assim como o coeficiente de correlação que se pretende utilizar. Neste caso foi utilizado o coeficiente de correlação de Spearman [63].

De seguida foi efectuado o mesmo processo, mas para cálculo do valor de correlação entre o consumo médio mensal por localidade e a localidade, por forma a aferir se a localização da habitação onde está integrado o dispositivo tem influência no consumo mensal de água.

4.2.2.2 Kmeans

O próximo método implementado, denominado de Kmeans, pertence à classe de algoritmos de clustering. O clustering é um problema de aprendizagem não supervisionada, onde se pretende agrupar o conjunto de dados, em subconjuntos que possuem alguma noção de similaridade entre os seus elementos.

O Kmeans é um algoritmo de clustering baseado em centróides, pelo que como referido já anteriormente no sub-capítulo 2.5.3.2, uma das suas maiores desvantagens é o conhecimento a priori do número de clusters k a ser utilizado, para que os seus resultados sejam o mais optimizados possível. Uma das soluções possíveis para resolver este problema seria utilização da tentativa erro, até encontrar uma solução óptima. Esta solução pode se tornar bastante demorada e não garante que se encontre a solução óptima.

A solução proposta para resolver este problema, é a utilização do método de correlação para identificar qual a variável com maior influência no consumo de água mensal. Após identificar a variável com maior influência, foi calculado o número de elementos distintos dessa variável e esse valor é utilizado para definir o número de cluster a utilizar no algoritmo Kmeans.

O método implementado foi construído com recurso à biblioteca Spark MLlib, e para isso inicialmente foi necessário definir os seguintes parâmetros de entrada:

- k, número de clusters
- maxIterations, número máximo de iterações a ser executado
- initializationMode, especifica se a inicialização é aleatória ou através do kmeans

Depois de definidos os parâmetros de entrada, foi efectuada uma pesquisa à base de dados para obter os dados sobre os consumos mensais de cada um dos contadores, e de seguida calculada o consumo médio mensal por contador através do método estatístico implementado anteriormente. Foram utilizados os consumos médios mensais dos contadores, em vez dos valores de consumo de um determinado mês, por forma a obter melhores resultados, uma vez que os valores médios de consumo têm um peso mais significativo para a representação do consumo de um contador, do que a utilização do valor de consumo de um único mês.

De seguida os dados da pesquisa e os parâmetros anteriormente definidos, foram passados à função de treino do modelo Kmeans, para que sejam calculados os valores dos centróides. Depois de calculados os centróides, cada uma das observações do consumo médio é atribuída a um cluster. Esta atribuição é feita através da distância quadrada euclidiana mínima verificada entre os valores de consumo médio de cada contador e o valor dos centróides.

Através desta solução, cada um dos contadores fica associado a um dos clusters existentes, possibilitando a identificação de contadores com consumos semelhantes e identificação de diferentes padrões de consumo. Esta informação(identicador do dispositivo, consumo médio, valor do centróide, cluster a que pertence) é armazenada numa tabela Cassandra para posteriormente ser utilizada para consulta.

4.2.2.3 Regressão Linear

Para a implementação do método de regressão linear, foi utilizada a biblioteca Spark MLlib. O Spark MLlib fornece uma implementação de regressão linear com gradiente estocástico descendente. O gradiente estocástico descendente é um método de optimização para problemas de optimização sem restrições.

A utilização de um modelo de regressão com gradiente estocástico descente, tem algumas implicações, no que diz respeito aos valores dos dados utilizados. O gradiente descente é incapaz de prever bons resultados quando são utilizadas matrizes esparsas, que contêm valores com um grau de oscilação muito elevado. Deste modo, a regressão linear com gradiente descendente foi tomada em conta para implementação deste método, uma vez que os dados não apresentam grandes oscilações.

A um nível teórico, gradiente descendente é um algoritmo que minimiza funções. Dada uma função definida por um conjunto de parâmetros, a descida de gradiente começa com um conjunto inicial de valores de parâmetros e de forma iterativa move-se em direção a um conjunto de valores de parâmetros que minimizem a função. Esta minimização iterativa é conseguida usando cálculos, tomando medidas no sentido negativo da função gradiente [58].

A implementação seguida faz uso da função de regressão linear simples apresentada de seguida:

$$Y = mx + b$$

A regressão linear simples utiliza a fórmula dos quadrados mínimos linear, em que a melhor aproximação/previsão é definida como aquela que minimiza soma da diferença dos quadrados entre os valores dos dados e os valores do modelo de treino.

Para a implementação deste método, foi inicialmente efectuada uma pesquisa à base de dados para recolha dos consumos mensais de cada um dos contadores. De seguida os dados dos consumos mensais e respectivos meses são normalizados e passados como parâmetros de entrada da função de treino do modelo de regressão linear. Para a realização do treino de dados foram utilizadas 30 iterações.

Após a etapa de treino dos dados, é aplicado o modelo de previsão para todos os meses presentes nos dados utilizados para treino e para o mês seguinte ao último dos dados de treino. A informação dos valores previstos de cada mês e respectivo identificador do dispositivo é armazenada na base de dados, segundo o modelo de dados apresentado no sub-capítulo 3.4.

4.2.2.4 Movimento Geométrico Browniano

Por fim, o último método implementado tem como objectivo a previsão do consumo de água do próximo mês tomando como base o consumo dos meses anteriores. A implementação deste método, foi efectuada com recurso ao Spark. Neste método não foi utilizada, a biblioteca Spark MLlib, uma vez que este método não faz parte de nenhum algoritmo de machine learning, mas é bastante utilizado no campo da previsão, como por exemplo no campo da matemática financeira, para modelar os preços das acções segundo o modelo de Black-Scholes.

O método que se pretende implementar é denominado, de movimento geométrico browniano. O movimento browniano é o processo estocástico mais simples para dados de tempo contínuo. A representação matemática do movimento browniano é dada pelo processo de Wiener e apresentada de seguida:

$dSt = \mu . St.dt + \sigma . St.dWt$, onde Wt é o processo de Wiener, μ a percentagem de derivação e σ a percentagem de volatilidade.

Segundo este modelo, o valor da variável é alterado numa determinada quantia por unidade de tempo, que é normalmente distribuída pela média e desvio-padrão. Para prever o valor do próximo consumo mensal para um determinado contador, segundo o movimento browniano, foi utilizada a seguinte equação:

$$Si + 1 = Si.\mu . dt + Si.\sigma . \varepsilon . dt$$

O primeiro passo a ser efectuado para calcular o valor de consumo do próximo mês, segundo o movimento browniano, foi a pesquisa dos consumos mensais para cada um dos contadores. De seguida foi calculada a média e desvio-padrão entre cada intervalo de tempo dos dados da pesquisa, através do método apresentado

no sub-capitulo 4.2.2.1 e calculado o valor do consumo previsto do próximo mês, segundo a equação anteriormente apresentada. O valor de previsão é calculado diversas vezes, tantas quanto o número de elementos da pesquisa, e no final é calculada a média de todos os valores previstos. A informação do valor de consumo previsto para o próximo mês de cada contador é armazenada na base de dados.

4.2.3 Interface Web

Por forma a visualizar a informação dos resultados obtidos com a implementação da solução, foi criada uma página Web. A página Web foi desenvolvida na linguagem HTML e JavaScript e com auxilio da ferramentas HighCharts, HighStocks [64] e OpenStreetMaps [65]. O HighCharts é uma biblioteca de gráficos, escrita em JavaScript, que oferece uma forma simplificada de adicionar gráficos interactivos numa página Web. O HighCharts suporta uma grande diversidade de tipos de gráficos, como gráficos linha, coluna, área, bolha, pizza, medidas de ângulo entre outros. O HighStocks é uma extensão da biblioteca HighCharts que permite criar gráficos ou cronogramas, que incluem opções sofisticadas de navegação, como navegador de series pequenas, intervalo de datas predefinidas, pesquisa por data, scrolling e deslocamento.

O OpenStreetMaps é um projecto desenvolvido por uma comunidade voluntária de mapeadores que contribuem e mantêm actualizados os seus dados. O OpenStreetMaps permite a criação de mapas do mundo editáveis , que contêm informação sobre estradas, trilhos, cafés, estações ferroviárias e outras informações.

Desta forma, foi desenvolvida uma plataforma Web simples, que permite rapidamente aferir e visualizar alguns resultados da análise e previsão efectuada.

A informação disponível na plataforma Web é carregada a partir de um serviço externo criado, que permite que a informação seja lida do sistema de armazenamento Apache Cassandra e carregada para a página Web. O serviço criado tem implementados diversos métodos que permitem carregar informação especifica para uma determinada operação, como pesquisar informação de consumo por hora, dia ou mês de um determinado contador, pesquisar informação de todos os contadores, pesquisar informação sobre todos os contadores com um consumo similar, pesquisar informação sobre o consumo médio por mês de todos os contadores, consumo médio por agregado e localidade e pesquisar informação sobre o consumo previsto do próximo para um determinado contador. A invocação dos métodos do serviço é feita através de jQuery Ajax, com pedidos HTTP GET. A informação devolvida pelos pedidos é enviada sob o formato JSON, e posteriormente lida para os respectivos componentes da página Web.

A plataforma Web desenvolvida é um protótipo, com o intuito de fornecer uma visualização dos resultados da solução implementada, de forma mais apelativa e dinâmica, pelo que não foram tomados em consideração requisitos de autenticação, administração, design e segurança. O protótipo apresentado é uma possível solução de visualização de informação no contexto dos Smart Meterings e análise BigData, que pretende oferecer aos utilizadores, sejam consumidores finais ou distribuidores de água, uma plataforma simples, intuitiva e de fácil interacção, para que estes tenham o acesso a informação diversificada do consumo de água

e dos respectivos dispositivos de medição.

Os resultados obtidos, através da plataforma desenvolvida, serão posteriormente apresentados e descritos no próximo capítulo.

4.2.4 Detalhes e Limitações da Implementação

A solução apresentada encontra-se implementada sobre a disposição de diversos módulos capazes de efectuar as operações descritas nos métodos implementados, pelo que para proceder a sua execução é necessário executar o ficheiro Jar do respectivo módulo, uma vez que esta não se encontra integrada numa plataforma única e autónoma.

Toda a solução funciona com base nos modelos de dados definidos, uma vez que esta se encontra implementada para a estrutura de dados seguida e apresentada anteriormente no sub-capítulo 3.2. Pelo que a alteração da estrutura de dados utilizada implica ligeiras alterações à forma como a solução foi implementada, para que esta se encontre em funcionamento.

Os métodos de armazenamento de informação desenvolvidos implicam a construção prévia das respectivas tabelas do sistema de armazenamento utilizado, uma vez que a gestão do sistema de armazenamento ao nível da criação e remoção de tabelas não é assegurada pela solução apresentada.

Relativamente aos métodos de pré-processamento, o método de pré-processamento de informação em streaming, apresenta algumas limitações, no que diz respeito ao seu funcionamento quando comparado com o que acontece na realidade, uma vez que a forma com que os dados são recebidos na solução implementada é diferente da forma com que estes são recebidos numa solução real. Esta limitação advém do facto, já anteriormente referido nas limitações do simulador de dados, uma vez que as medições de diversos dispositivos não ocorrem de forma paralela, mas sim ocorrem em separado para cada um dos dispositivos.

5 Avaliação e Resultados

5.1 Cenário de Teste

A avaliação de uma solução representa uma parte importante do seu desenvolvimento, uma vez que é nesta fase que o protótipo desenvolvido é testado, por forma a avaliar a sua integração num ambiente mais próximo da realidade, mas também para avaliar a qualidade dos resultados obtidos com o mesmo.

Desta forma, existem inúmeros aspectos que podem ser verificados para avaliar a qualidade de uma solução. Neste projecto vão ser tomados em consideração os seguintes aspectos de avaliação:

- avaliação da performance do protótipo em diferentes cenários
- avaliação da veracidade dos resultados obtidos

Para a construção de testes capazes de avaliar o comportamento da solução num ambiente mais realista, é necessário primeiramente definir diferentes cenários de teste onde a solução irá ser avaliada. No contexto deste projecto, diferentes cenários implicam diferentes quantidades de informação, que por sua vez implicam diferentes performances no seu armazenamento e processamento. Desta forma, vão ser utilizados quatro cenários de teste distintos, para avaliar a performance da solução apresentada. Os cenários de teste definidos são apresentados de seguida:

- **Cenário 1:** utilização de dados de leituras, referentes a 25 habitações
- **Cenário 2:** utilização de dados de leituras, referentes a 50 habitações
- **Cenário 3:** utilização de dados de leituras, referentes a 150 habitações
- **Cenário 4:** utilização de dados de leituras, referentes a 1 habitação, para um período de avaliação de 6 meses

Para efectuar o desenvolvimento de testes capazes de avaliar a veracidade dos resultados obtidos, é necessário obter informação sobre estes valores no ambiente real, por forma a confrontar os resultados obtidos com a solução implementada e os valores reais verificados. Numa situação de desenvolvimento, onde

os dados utilizados são provenientes de fontes reais, esta abordagem torna-se mais difícil de avaliar, uma vez que não existe um modelo definido nos dados utilizados. Desta forma, seria necessário recorrer à pesquisa de informação auxiliar para obter informação sobre o modelo seguido pelos dados utilizados.

Uma vez que os dados utilizados no desenvolvimento deste projecto, foram dados provenientes de uma fonte desenvolvida no decorrer do projecto, para a simulação de dados, estes seguem um modelo, que vai ser utilizado para confrontar os resultados obtidos e assim verificar a sua veracidade através do cálculo de diferentes métricas.

5.1.1 Testes

Por forma a avaliar a qualidade da solução desenvolvida foram implementados diferentes testes, consoante os objectivos que se pretendem avaliar. Existem inúmeros testes que podem ser efectuados quando se pretende avaliar uma solução.

No contexto da solução apresentada, a principal medida que se pretende avaliar na performance da solução é a sua latência. A avaliação da latência em diferentes cenários, permite verificar a capacidade da solução se adaptar a uma variedade de ambientes, bem como descobrir possíveis casos onde a solução apresente um comportamento menos positivo.

Desta forma, primeiro foram implementados testes para verificar a latência da solução ao nível do pré-processamento e armazenamento dos dados. Como tal, os métodos de pré-processamento e armazenamento de informação implementados, vão ser testados nos três primeiros cenários de teste definidos no sub-capitulo anterior e avaliado o valor de latência nos respectivos casos. Uma vez, que foram implementados três métodos para processamento e armazenamento de informação, que representam o cálculo do valor agregado do consumo de água com diferentes granularidades (hora,dia,mês), os dados utilizados em cada teste, possuem um período de avaliação igual ao método que se pretende avaliar.

Por forma, a obter um resultado mais preciso foram efectuadas 10 iterações para cada um dos casos e calculado o valor médio das iterações realizadas.

De seguida, foram efectuados testes para avaliar a performance dos métodos de previsão implementados. Como tal, os diferentes métodos de previsão de consumo de água, foram testados no cenário de teste número 4. Por forma a avaliar o comportamento de execução em paralelo dos métodos de previsão, estes vão ser testados com um número de threads diferentes em cada teste, por forma a avaliar a sua latência nos diversos casos.

Posteriormente foram avaliados os resultados obtidos com a solução proposta. Como tal, foram efectuados testes aos métodos de análise estatística, método de clustering e métodos de previsão. Nos métodos de análise estatística, vão ser analisados os valores dos resultados da média e variância. Para avaliar os resultados dos métodos de análise estatística vai ser calculado o erro relativo entre o valor obtido e o valor esperado.

No método de clustering, Kmeans, vão ser testados diferentes valores nos parâmetros de entrada do modelo e calculado o erro da soma dos quadrados do conjunto.

Nos métodos de previsão vão ser analisados os valores de previsão obtidos com os métodos implementados e comparados com os valores esperados, através do cálculo do erro quadrático médio e erro relativo. Por forma, a avaliar os métodos de previsão vão ser utilizados os dados do consumo agregado por mês, calculado através do método de pré-processamento de informação, para o conjunto de dados do cenário 4. Desta forma, é possível obter informação sobre o consumo mensal de um determinado contador, para um período de 6 meses.

Os dados utilizados para testar os métodos de previsão implementados, são os dados dos primeiros 5 meses de consumo, ou seja, deste modo pretende-se calcular o valor do consumo mensal do sexto mês de um determinado contador.

5.2 Resultados

5.2.1 Avaliação da performance

O primeiro teste a ser efectuado, foi a avaliação da performance dos métodos de pré-processamento e armazenamento dos dados. Como tal, pretende-se avaliar o valor da latência destes métodos, em diferentes cenários por forma a avaliar o seu comportamento de execução quando sujeitos a uma quantidade de informação distinta.

Uma vez que os métodos implementados vão ser executados de forma separada e a execução destes implica que seja efectuada a inicialização da ferramenta utilizada, Apache Spark, o valor da latência tende sempre a ser um pouco mais elevado devido ao tempo de inicialização da ferramenta.

De seguida são apresentados os resultados do tempo de execução obtidos nas diversas iterações, bem como o tempo de execução médio das diversas iterações, do teste de performance do método de pré-processamento e armazenamento do cálculo do valor de consumo agregado por hora, efectuado no cenário de teste 1 e 2 anteriormente definidos:

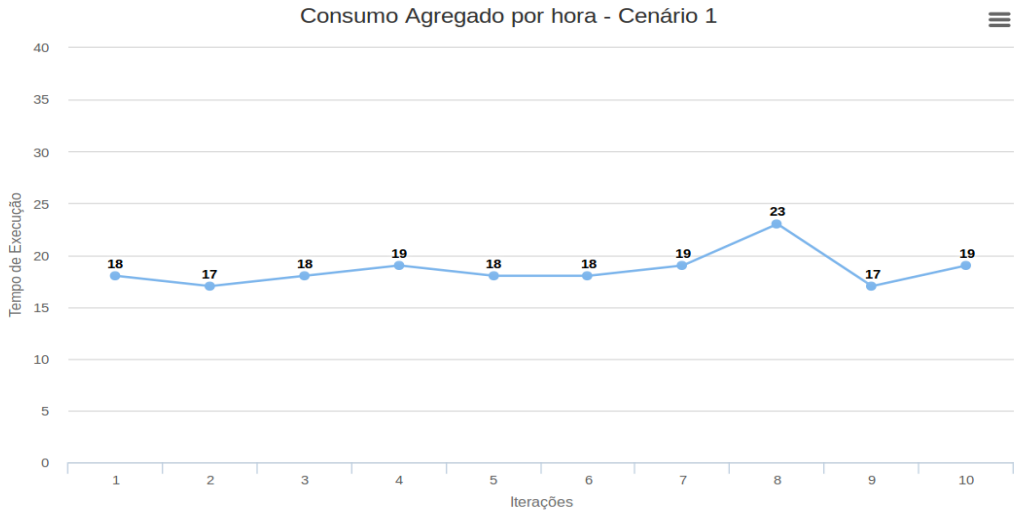

Figura 17: Resultados teste de performance: Consumo agregado por hora - Cenário 1

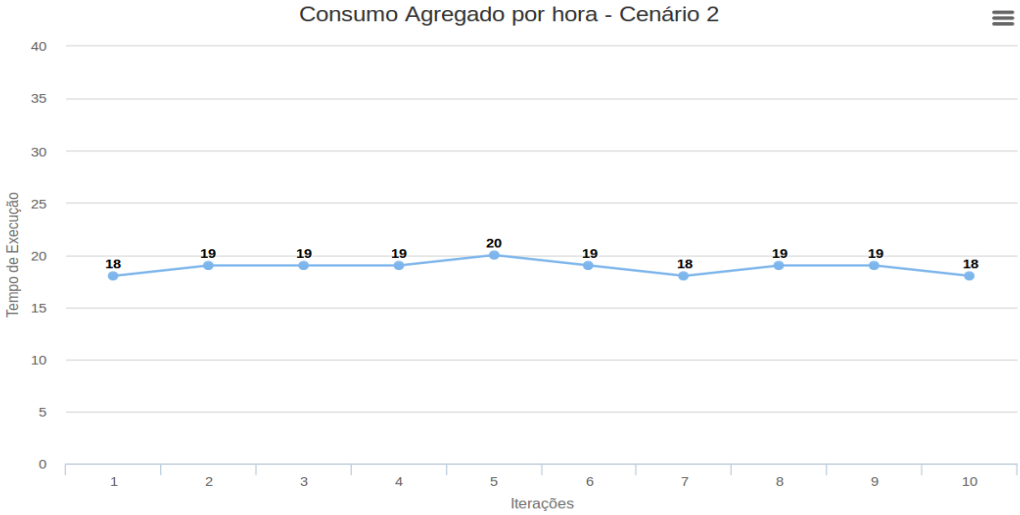

Figura 18: Resultados teste de performance: Consumo agregado por hora - Cenário 2

	Cenário de teste 1	**Cenário de teste 2**
Tempo de execução	18.6s	18.8s

Tabela 14: Resultado de performance do método de armazenamento e pré-processamento – consumo agregado por hora

De seguida são apresentados os resultados do tempo de execução obtidos nas diversas iterações, bem como o tempo de execução médio das diversas iterações, do teste de performance do método de armazenamento e pré-processamento do cálculo do valor de consumo agregado por dia, nos cenários de teste 1 e 2:

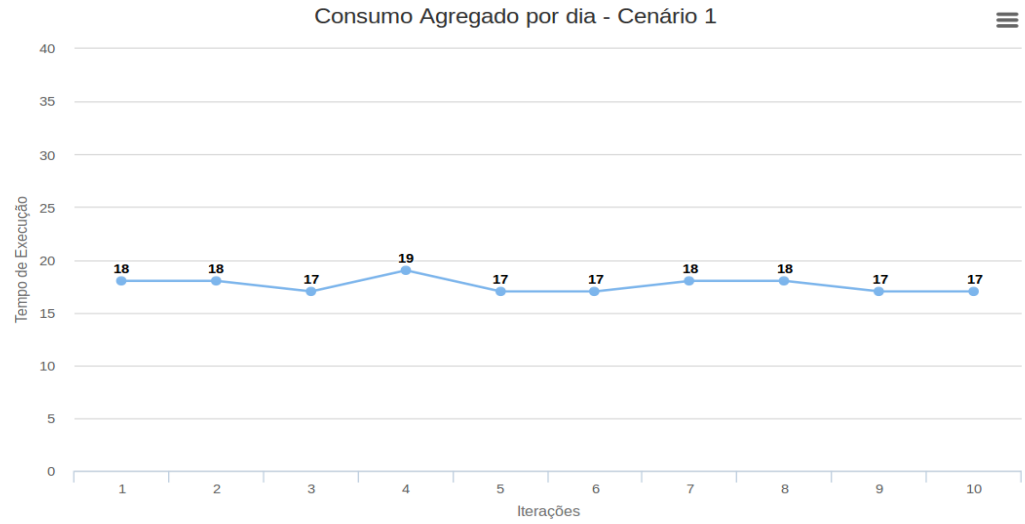

Figura 19: Resultados teste de performance: Consumo agregado por dia - Cenário 1

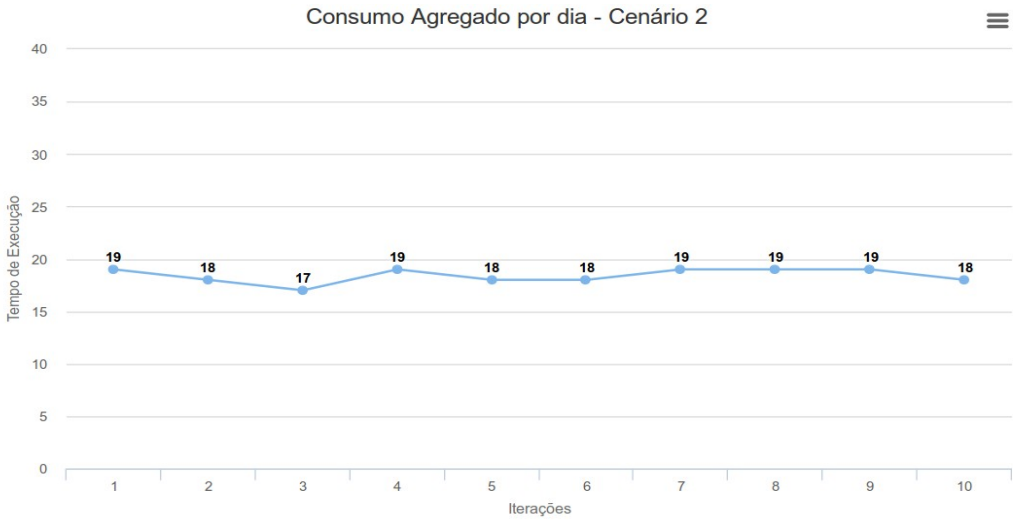

Figura 20: Resultados teste de performance: Consumo agregado por dia - Cenário 2

	Cenário de teste 1	**Cenário de teste 2**
Tempo de execução	17.6s	18.4s

Tabela 15: Resultado de performance do método de armazenamento e pré-processamento - consumo agregado por dia

Por fim, são apresentados os resultados do tempo de execução obtidos nas diversas iterações, bem como o tempo de execução médio das diversas iterações, do teste de performance do método de armazenamento e pré-processamento do cálculo do valor de consumo agregado por mês, nos cenários de teste 1, 2 e 3:

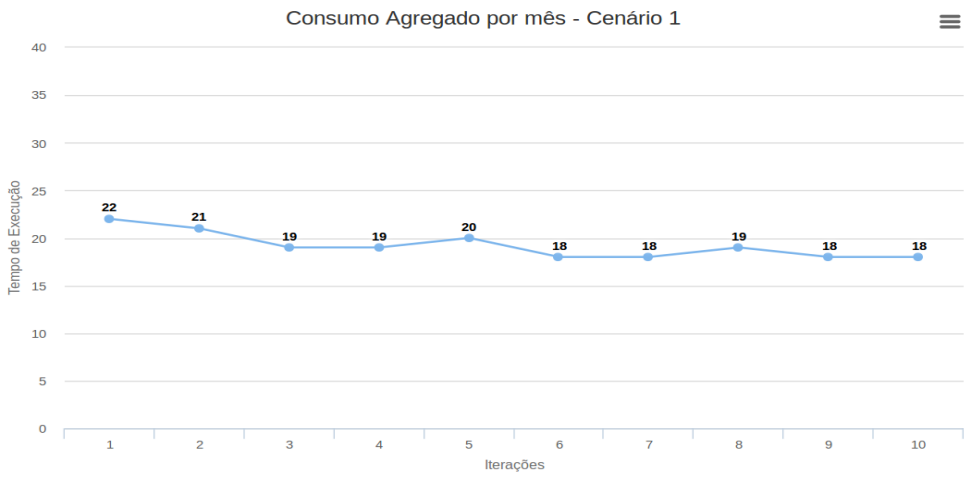

Figura 21: Resultados teste de performance: Consumo agregado por mês - Cenário 1

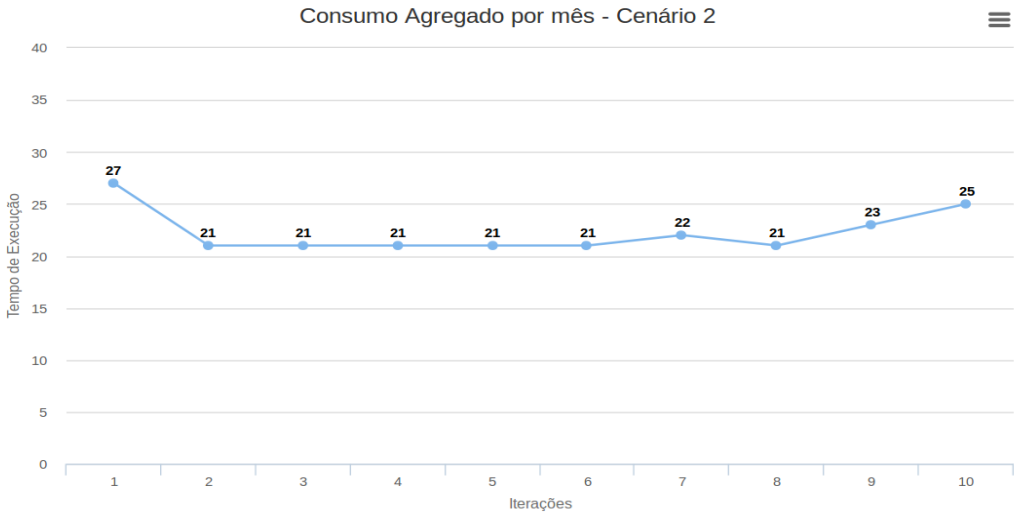

Figura 22: Resultados teste de performance: Consumo agregado por mês - Cenário 2

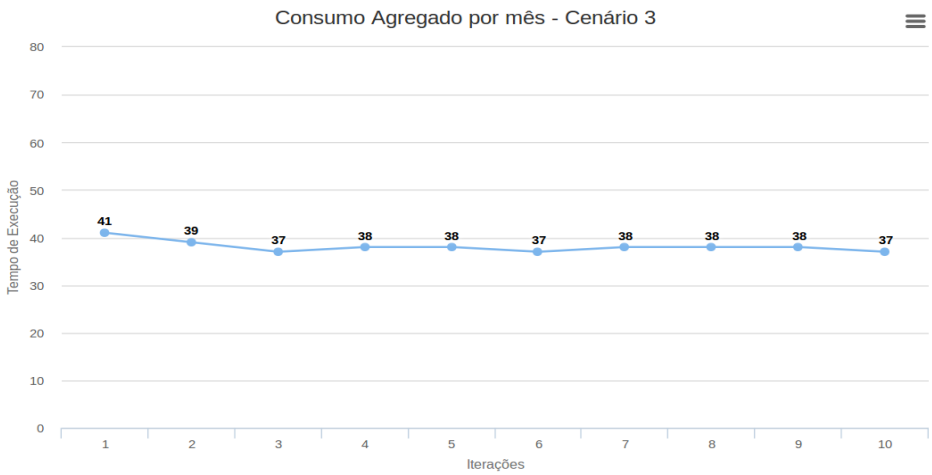

Figura 23: Resultados teste de performance: Consumo agregado por mês - Cenário 3

	Cenário de teste 1	**Cenário de teste 2**	**Cenário de teste 3**
Tempo de execução	19.2s	22.3s	38.1s

Tabela 16: Resultado de performance do método de armazenamento e pré-processamento - consumo agregado por mês

O resultado do tempo de execução dos métodos testados nos diferentes cenários é apresentado em segundos. Como se pode verificar através dos resultados apresentados, o tempo de execução dos diferentes testes é relativamente baixo, o que significa que estes apresentam uma baixa latência e consequentemente fornecem uma solução rápida e eficiente para pré-processamento e armazenamento de informação.

Os resultados dos testes efectuados, quando comparados entre si nos diferentes cenários, apresentam valores significativamente bons, uma vez que a quantidade de dados do cenário 3 é o triplo da quantidade de dados do cenário 2 e o sêxtuplo da quantidade de dados do cenário 1 e o tempo de execução entre os três cenários não apresenta um crescimento linear consoante o aumento da quantidade de dados, ou seja, o aumento da quantidade de dados não é proporcional ao aumento do tempo de execução. Desta forma, é possível aumentar a quantidade de dados a ser processada e armazenada, que o tempo de execução tem um aumento inferior ao factor de aumento da quantidade de dados.

Quando comparados os resultados entre os diferentes métodos, estes também apresentam bons resultados, uma vez que a quantidade de dados no teste de performance do método de pré-processamento e armazenamento do consumo agregado por mês é cerca de 30 vezes maior que a quantidade de dados no teste de performance do método de pré-processamento e armazenamento do consumo agregado por dia, e cerca 720 vezes maior que a quantidade de dados no teste de performance do método de pré-processamento e armazenamento do consumo agregado por hora, pelo que os tempos de execução entre os diferentes métodos apresentam diferenças pouco significativas do tempo de execução, ou seja, estes têm capacidade para processar e armazenar quantidades de dados superiores sem que o tempo de execução esteja comprometido.

O segundo teste a ser efectuado, foi a avaliação da performance dos métodos de previsão do consumo de água do mês seguinte. Como tal, pretende-se avaliar também o valor da latência dos métodos de previsão implementados. Visto que, os métodos de previsão tem como principal objectivo prever o consumo mensal do próximo mês de um determinado contador, o cenário de teste utilizado é diferente dos cenários anteriores. Neste caso foi utilizado o cenário de teste 4, em que é calculado o valor de latência de cada um dos métodos, para um número de threads Spark variável. De seguida é apresentada a tabela com os resultados obtidos:

Método/Número de threads	**2**	**4**	**6**	**8**
Regressão Linear	28s	30s	30s	28s
Movimento Geométrico Browniano	22s	23s	23s	21s

Tabela 17: Resultado de performance dos métodos de previsão

Os resultados obtidos com os testes efectuados, permitem concluir que o método de previsão Movimento Geométrico Browniano apresenta uma latência mais baixa quando comparado com o valor de latência do método de previsão Regressão Linear. O facto do método de previsão Movimento Geométrico Browniano apresentar um valor de latência mais baixo, pode estar intrinsecamente ligado ao facto de este não necessitar da fase de treino do modelo.

Ao contrário do que seria esperado, os valores dos resultados obtidos quando executados com um número de threads diferentes, não apresentam uma variação significativa, pelo que se torna difícil tirar conclusões sobre os mesmos. O facto dos resultados não apresentarem grandes variações pode estar ligado ao facto da quantidade de informação ser reduzida, ou ao facto dos poucos recursos da infra-estrutura onde foram efectuados os testes.

5.2.2 Avaliação de veracidade dos resultados

Por forma a avaliar a veracidade dos resultados obtidos com a solução proposta, foram efectuados testes aos métodos de análise e previsão de consumo de água implementados. Como tal, foi utilizado o o erro relativo e o erro quadrático médio, para calcular o erro associado entre a comparação dos resultados obtidos com a solução e os resultados do modelo seguido para criação dos dados.

O primeiro teste efectuado pretende avaliar o resultado do consumo médio mensal por pessoa e a respectiva variância, uma vez que o modelo seguido pelos dados apresenta informação relativa ao consumo médio por pessoa. Os valores apresentados encontram-se representados sob a medida de litros. De seguida é apresentada

a tabela com informação sobre o consumo médio mensal por pessoa e variância do modelo seguido:

Consumo médio mensal	Variância do consumo mensal
176	8

Tabela 18: Consumo médio mensal por pessoa e variância - Modelo seguido pelos dados

De seguida é apresentada a tabela com informação sobre o consumo médio mensal e variância obtido com a solução:

Consumo médio mensal	Variância do consumo mensal
198	10

Tabela 19: Consumo médio mensal por pessoa e variância - Solução proposta

Por forma a avaliar os resultados obtidos, foi utilizada a métrica erro relativo para calcular o erro entre o consumo médio mensal por pessoa e variância, do modelo e o consumo médio mensal por pessoa e variância obtido com a solução. De seguida é apresentado o resultado do erro relativo do consumo médio mensal por pessoa:

- **Erro relativo:** 0.125
- **Erro relativo percentagem:** 12.5%

O resultado do erro relativo da variância média mensal por pessoa foi o seguinte:

- **Erro relativo:** 0.25
- **Erro relativo percentagem:** 25%

Os resultados obtidos tanto no consumo médio, como na variância apresentam um erro relativo baixo, no entanto seria de esperar que os valores do erro fossem mais baixos, devido à simplicidade do método testado. O facto de os erros serem mais elevados do que o esperado, pode ser justificado pelo facto de os valores de consumo de água gerados pelo simulador serem um pouco mais elevados do que os valores reais do consumo de água. Pelo que consequentemente, os valores da média e variância também tendem a ser mais elevados.

O segundo teste efectuado pretende avaliar o resultado do consumo mensal previsto do próximo mês com os métodos de previsão implementados na solução. Para tal, vai ser calculado o erro quadrático médio e o erro relativo.

De seguida é apresentada a tabela com os respectivos consumos mensais de um determinado contador, para o período de avaliação de 6 meses:

Mês 1	Mês 2	Mês 3	Mês 4	Mês 5	Mês 6
5877	6549	6380	5889	6320	5921

Tabela 20: Consumo mensal de um determinado contador

A tabela seguinte apresenta o valor do consumo mensal previsto para o sexto mês, através dos dois métodos de previsão implementados:

Método/Consumo previsto	**Consumo mensal previsto**
Regressão Linear	6144,3
Movimento Geométrico Browniano	6613,9

Tabela 21: Consumo mensal previsto

O resultado do erro quadrático médio e erro relativo do método de Regressão Linear foi o seguinte:

- **Erro quadrático médio:** 49862.89
- **Erro relativo:** 0.0377
- **Percentagem:** 3.8%

O resultado do erro quadrático médio e erro relativo do método de Movimento Geométrico Browniano foi o seguinte:

- **Erro quadrático médio:** 480110.41
- **Erro relativo:** 0.117
- **Percentagem:** 11.7%

Através dos resultados obtidos pode-se concluir que o método de Regressão Linear é mais preciso na previsão do consumo de água do próximo mês do que o método do Movimento Geométrico Browniano. Apesar de apresentar uma latência mais elevada, como já verificado no sub-capitulo anterior, este apresenta resultados mais assertivos. O facto de o Movimento Geométrico Browniano apresentar um valor de previsão menos preciso, deve-se ao facto de o seu modelo se basear no consumo de mês anterior para efectuar a previsão do consumo do mês seguinte.

Por fim, foi testado o método de clustering implementado, Kmeans. O algoritmo de Kmeans, como já referido anteriormente implica a definição prévia do número de cluster a ser utilizado pelo modelo. Pelo que de seguida serão demonstrados os resultados obtidos do algoritmo, com diferentes números de cluster definidos.

No primeiro teste efectuado, o número de clusters definido foi 4. O valor definido do número de cluster, foi calculado com base no método de correlação como já referido anteriormente na descrição da implementação do método de clustering. De seguida é apresentado o valor do erro da soma dos quadrados do conjunto e a imagem dos respectivos clusters.

- **Erro da soma dos quadrados:** 301675.74

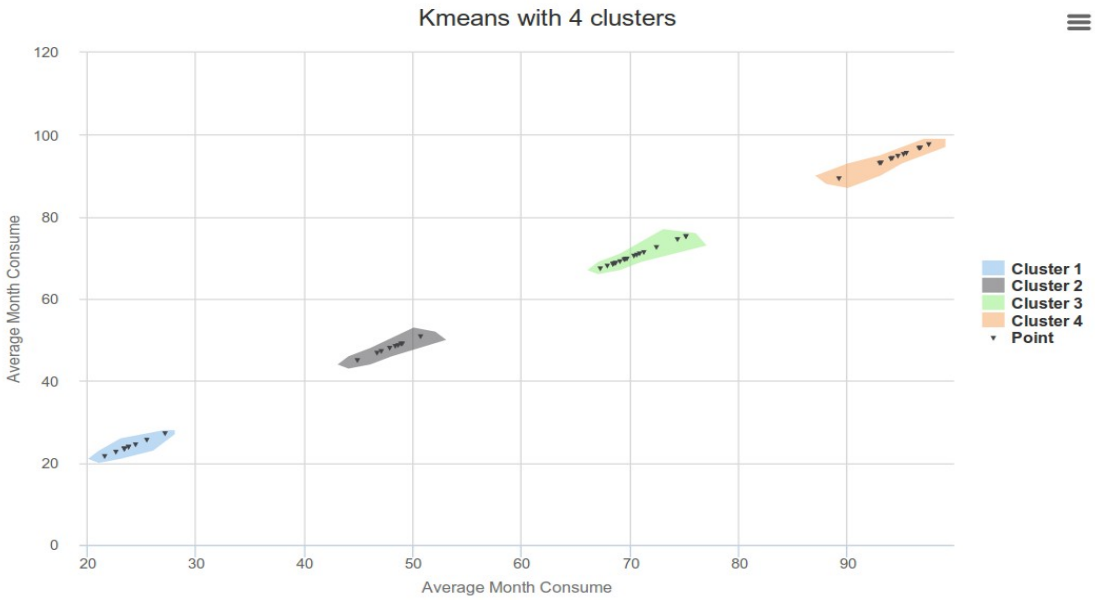

Figura 24: Resultado do algoritmo Kmeans com 4 clusters

Por forma, a obter uma visualização mais clara da informação, os valores do consumo mensal médio por contador foram divididos por 100, para que os pontos sejam mais perceptíveis.

No segundo teste efectuado ao método de clustering, o número de clusters definido foi 3. O valor definido tem como objectivo variar o número de clusters, por forma a avaliar o resultado da métrica utilizada na avaliação do método. De seguida é apresentado o valor do erro da soma dos quadrados do conjunto:

- **Erro da soma dos quadrados:** 41664320.06

Como se pode observar, através dos resultados do valor do erro da soma dos quadrados, o teste efectuado com um número de cluster igual a 3 apresenta um valor mais elevado do que o teste efectuado com um número de cluster igual a 4. Ou seja, uma vez que o método implementado pretende minimizar o valor do erro da soma dos quadrados, pode-se concluir que o método fornece melhores resultados quando tem como parâmetro de entrada um número de clusters igual a 4.

Também através da visualização da representação dos consumos médios mensais de cada contador é facilmente identificável o número de clusters óptimo a ser utilizado, por forma a que o algoritmo Kmeans forneça a solução mais optimizada. Através da visualização dos pontos é possível inferir que o número de clusters definido com base no método de correlação é o mais indicado. Desta forma, pode-se concluir que o método de clustering implementado apresenta valores de qualidade no agrupamento dos consumos médios mensais, quando executado com um número de clusters igual a 4.

5.2.3 Visualização dos resultados

De modo a representar alguma informação dos resultados obtidos com os métodos implementados e descritos no capítulo anterior, foi desenvolvida uma plataforma Web simples, que permite rapidamente aferir e visualizar alguns resultados da análise e previsão efectuada.

A interface da aplicação é composta por diversas páginas, sendo a página principal de entrada a representação do mapa mundo, centrado na região de Aveiro. No mapa encontram-se referenciados, através de um marcador com as coordenadas geográficas, todos os contadores tomados em consideração na implementação da solução proposta. Desta forma, é possível rapidamente e facilmente visualizar a localização de todos os contadores de água analisados e interagir directamente com cada um deles. A imagem seguinte representa a página principal de entrada da aplicação Web desenvolvida:

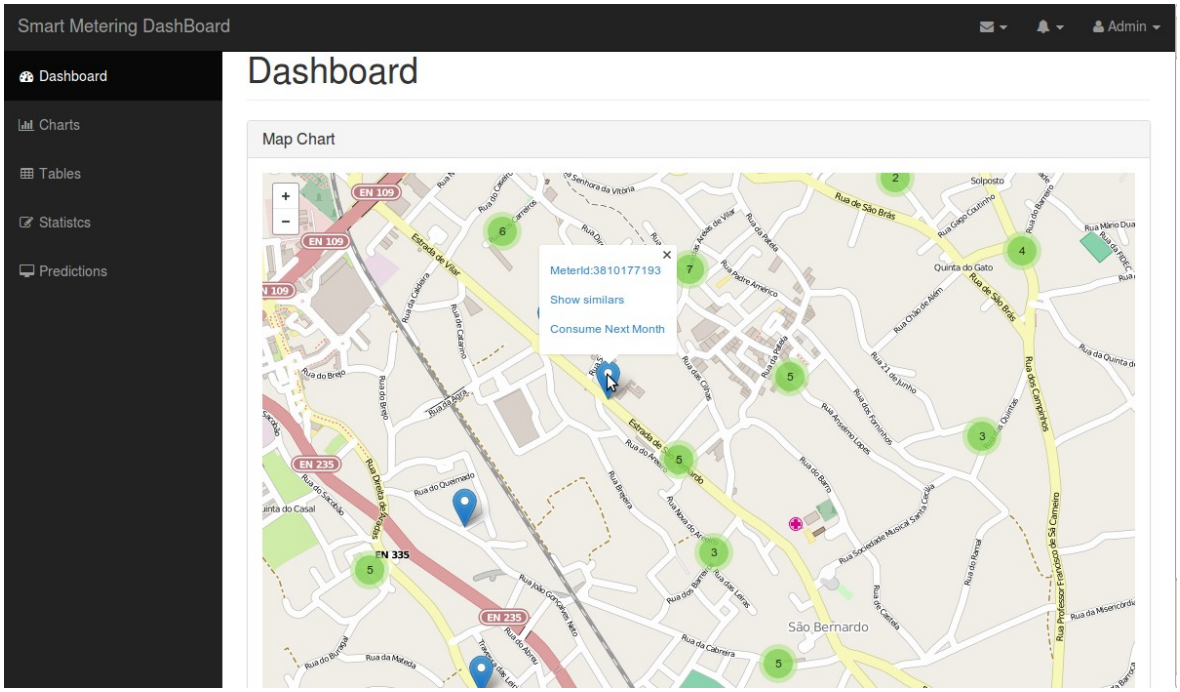

Figura 25: Página principal da aplicação Web

Através da interacção com o mapa e respectivos elementos que o constituem é possível interagir com cada um dos contadores por forma a obter informação sobre cada um deles e entre eles. Para isso é necessário apenas clicar sobre um dos marcadores presentes no mapa e é apresentada uma janela popup com informação sobre as operações disponíveis a efectuar. As operações disponíveis são a visualização de informação estatística de consumo de um contador, visualização de informação estatística de consumo por tempo e localidade, visualização de contadores com um consumo semelhante ao contador seleccionado, e visualização de informação sobre a previsão de consumo mensal de um contador.

A visualização de informação estatística de consumo de um contador, permite aos utilizadores visualizar a

informação de consumo por hora e dia do contador. A informação de consumo por hora é apresentada através de um gráfico linha, e a informação de consumo por dia apresentada através de um gráfico de barras. Ambos os gráficos permitem ao utilizador interagir facilmente com o gráfico, através de operações de navegação de tempo, que permitem visualizar a informação em diferentes períodos de tempo. A visualização de informação estatística de consumo por tempo e localidade, permite aos utilizadores visualizar a informação do consumo médio mensal de todos os contadores, assim como o consumo médio mensal por localidade. De seguida são apresentadas duas imagens das página de visualização de informação estatística de consumo:

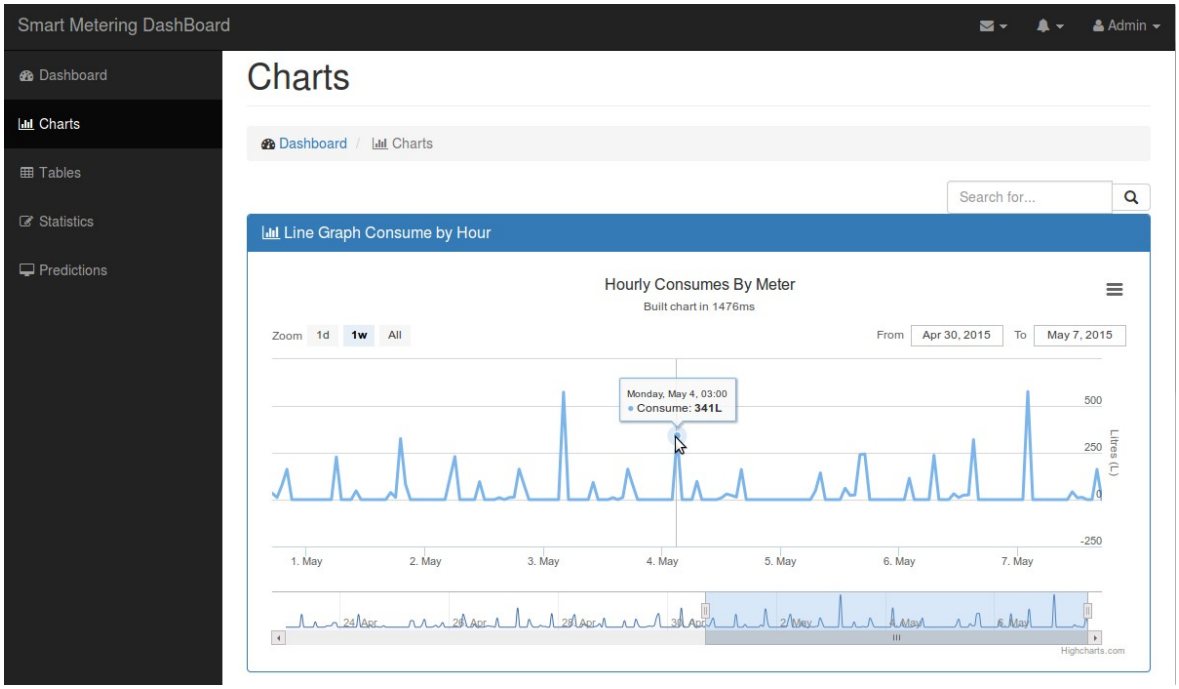

Figura 26: Consumo agregado por hora de um determinado contador

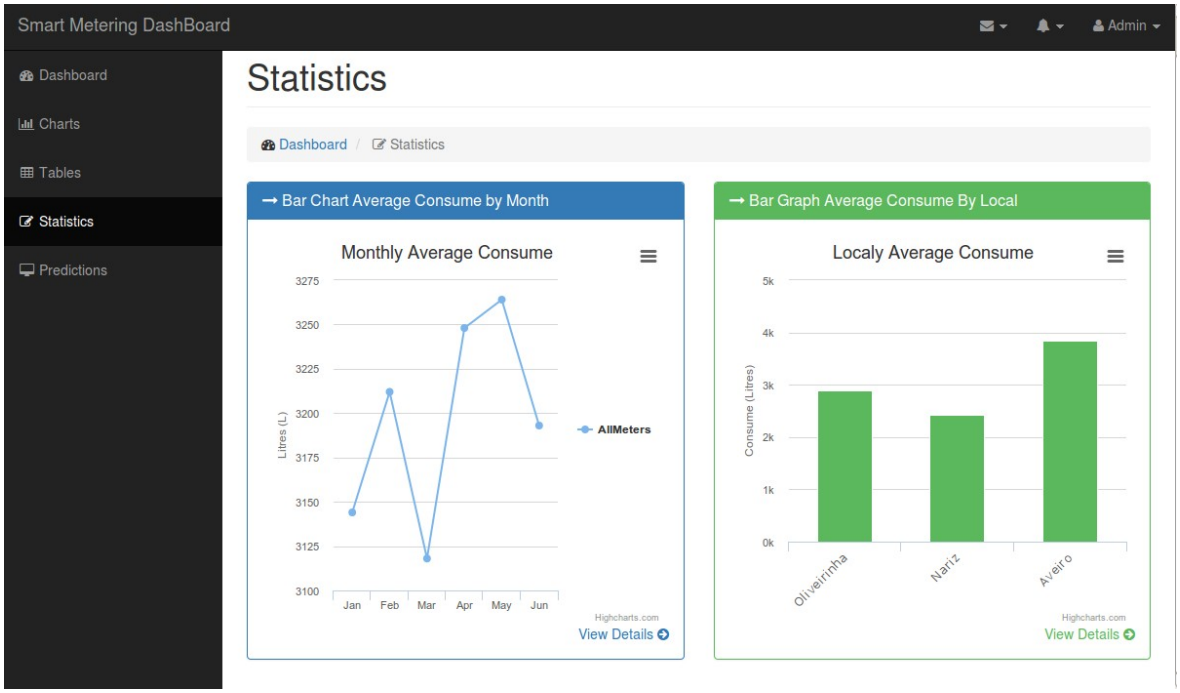

Figura 27: Consumo médio por mês e localidade

A operação para visualização de contadores com um consumo semelhante, permite ao utilizador visualizar no mapa todos os contadores com um consumo semelhante ao contador seleccionado. Desta forma, o utilizador tem um percepção da localização de contadores com consumos semelhantes.

A visualização de informação sobre a previsão de consumo do contador, permite aos utilizadores do sistema visualizar os valores previstos de consumo do próximo mês através dos diferentes métodos de previsão implementados. Por forma, a obter uma representação do método de regressão linear mais elucidativa é apresentado ao utilizador, um gráfico auxiliar que contém informação sobre as observações passadas, ou seja, valores de consumo dos meses anteriores e a recta de regressão traçada para a previsão efectuada. A imagem seguinte representa a página de visualização de informação de previsão de consumo:

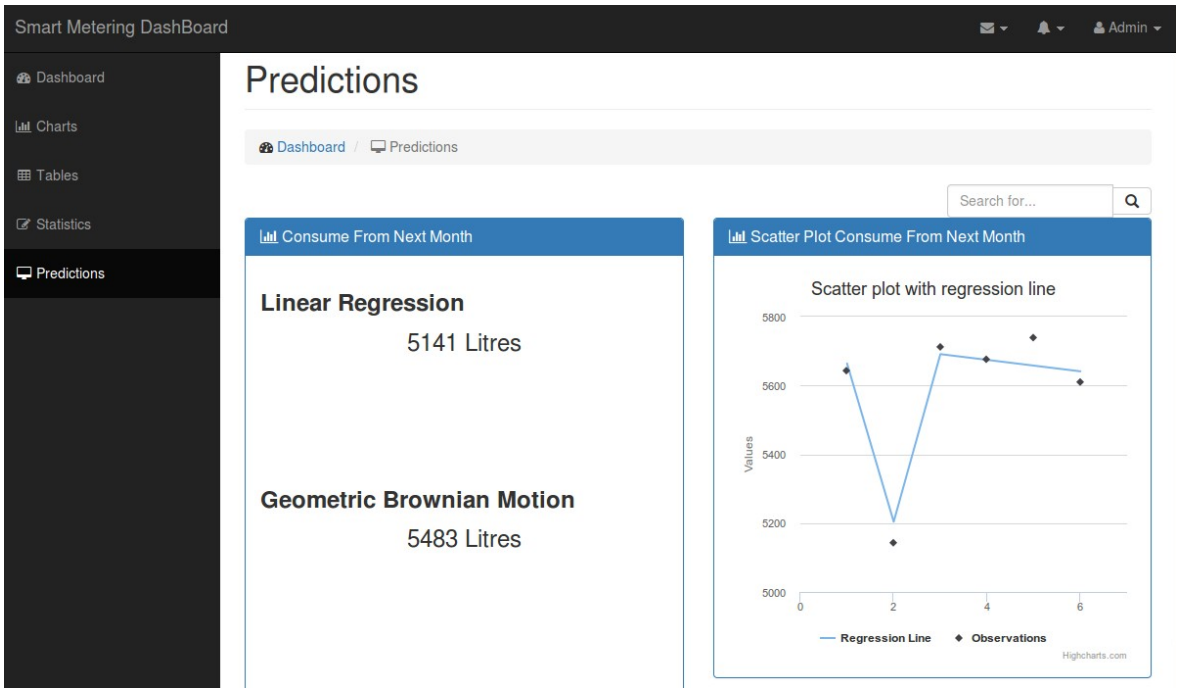

Figura 28: Consumo previsto para o próximo mês

Para além das operações já enunciadas, foram ainda implementadas mais algumas funcionalidades que permitem aos utilizadores visualizar informação estatística sobre o consumo médio por agregado familiar, bem como visualizar a informação dos registos de leitura ocorridos.

6 Conclusões

6.1 Conclusões

Com os crescentes avanços tecnológicos, verificados nos últimos anos, muitos operadores de telecomunicações têm desenvolvido soluções que permitem alcançar uma visão, onde no nosso mundo real, os objectos com que interagimos e o ambiente que nos rodeia se tornam mais inteligentes. Este paradigma, denominado de Internet das Coisas, tem vindo a revolucionar a sociedade onde nos enquadramos, com implementações em diversas áreas que permitem que seja possível adquirir conhecimento sobre o que nos rodeia. Estes avanços tecnológicos dão-nos a ideia, que no futuro tudo pode ser monitorizado, e que existirão dispositivos na maioria dos objectos que nos rodeia, o que implica que sejam necessários novas formas de gestão da informação, uma vez que a quantidade de informação presente nestes sistemas é gigantesca.

Deste ponto visto, muitos investigadores têm explorado novas soluções que permitam efectuar a gestão de toda esta informação, sem que ainda se tenha chegado a um consenso sobre uma abordagem padrão.

O presente trabalho aqui descrito fornece uma visão sobre uma possível abordagem que pode ser tomada em consideração para efectuar o armazenamento e processamento de grandes quantidades de informação. Para isso foram avaliados diversos métodos de armazenamento e processamento de informação, que são descritos ao longo deste documento. Para além das soluções exploradas para gestão da informação, foram também exploradas técnicas de aprendizagem, que permitem adquirir conhecimento sobre toda a informação tratada. A utilização destas técnicas e métodos de gestão de informação aqui apresentadas pretendem que seja possível, lidar com a informação e fornecer uma visão do que está acontecer à nossa volta, podendo assim serem tomadas decisões com o objectivo de melhorar o nosso quotidiano.

Deste ponto de visto, os principais objectivos apresentados no inicio deste documento, foram amplamente explorados, possibilitando assim a obtenção de resultados positivos, para a resolução dos problemas desta área, sendo que novos trabalhos futuros deveram continuar a ser efectuadas com o objectivo de explorar mais aprofundadamente estes problemas, uma vez que esta é uma área ainda em crescente expansão e com bastante trabalho de investigação a ser desenvolvido.

6.2 Trabalho Futuro

Devido à enorme abrangência da área da Internet das Coisas e do Smart Metering, muitos trabalhos de futuro pode ser desenvolvidos com o intuito de melhorar a solução aqui apresentada. Deste modo, de seguida vão ser enunciadas algumas medidas que pode ser tomadas em consideração para melhorar esta solução. Do ponto de vista, de armazenamento e processamento de informação no futuro a solução podia ser extendida a um ambiente de computação distribuída, uma vez que no presente trabalho esta vertente não foi explorada, e seria importante verificar o seu comportamento num ambiente distribuído, por forma a avaliar a sua performance num caso de solução real.

Ao nível da aquisição de conhecimento a partir da informação presente, diversas trabalhos podiam ser desenvolvidos, por forma a aumentar o grau de conhecimento, como por exemplo, desenvolvimento de métodos capazes de fornecer informação sobre fugas de água e informação sobre consumos excessivos. Assim como também puderiam ser implementadas novas técnicas de aprendizagem, com o intuito de reduzir a quantidade de recursos necessários para lidar com a informação, com por exemplo implementação modelos de Hashing. Um exemplo prático seria a substituição do algoritmo de clustering implementado por a implementação do algoritmo LSH.

Do ponto de vista, da visualização de informação poderiam ser implementadas novas funcionalidades, que permitam que a plataforma ofereça aos utilizadores mais informação. Caso disso, seria a implementação de um sistema de notificações para determinados eventos específicos, como existência de fugas, consumos elevados e emissão de facturas.

Referências

[1] R H Glitho. Applications architectures for machine to machine communications: research agenda vs. State-of-the-art. Broadband and Biomedical Communications, 2011

[2] D G Velev. Internet of things - analysis and challenges. Economic alternatives, 2011

[3] Marchment Hill Consulting Pty Ltd. Smart Water Metering Cost Benefit Study, 2010

[4] Water Corporation. Kalgoorlie Smart Metering Trial Frequently Asked Questions, 2010

[5] M Hatton. The global m2m market in 2013. White Paper, Machina Research, 2013

[6] K Ashton. That “Internet of Things'”Thing, http://www.rfidjournal.com/articles/view?4986, 2009

[7] “Next generation networks – frameworks and functional architecture models: overview of the internet of things”. International Telecommunication Union. Recommendation ITU-T Y.2060, 2012

[8] O Monnier. A smarter grid with the Internet of Things, http://e2e.ti.com/blogs/b/smartgrid/archive/2014/05/08/a-smarter-grid-with-the-internet-of-things.aspx, 2014

[9] ”Evaluating The Leading-Edge Italian Telegestore Project”, apresentado por Fabio Borghese. ENEL, Infrastructure and Networks Division

[10] EDP Distribuição. InovGrid, http://www.edpdistribuicao.pt/pt/rede/InovGrid/Pages/InovGrid.aspx

[11] EDP Distribuição. EDP lança conceito InovCity em Évora, http://www.edp.pt/pt/media/noticias/2010/Pages/EDPlancaconceitoinovCityemEvora.aspx

[12] The Apache Software Foundation. What is Apache Hadoop?, http://hadoop.apache.org/

[13] The Apache Software Foundation. HDFS Architecture, http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

[14] The Apache Software Foundation. Apache Hadoop NextGen MapRduce(YARN), http://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html

[15] Jeffrey Dean and Sanjay Ghemawat. Simplified Data Processing on Large Clusters. Nos Proceedings of the 6th Symposium on Operating System Design and Implementation, 2004

[16] Gaurav Vaish. Getting Sarted with NoSql, 2010

[17] Neal Leavitt. Will NoSQL Databases Live Up to Their Promise? IEEE Computer Science, 2010

[18] The Apache Software Foundation. Welcome to Apache Hbase, http://hbase.apache.org/

[19] The Apache Software Foundation. Hbase Wiki: Hbase Architecture, http://hbase.apache.org/book.html#architecture

[20] The Apache Software Foundation. Hbase Wiki: Hbase Data Model, http://hbase.apache.org/book.html#datamodel

[21] The Apache Software Foundation. Welcome to Cassandra, http://cassandra.apache.org/

[22] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C Hsieh, Deborah A Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes e Robert E Gruber. BigTable: A Distributed Storage System for Structured Data, 2006

[23] DataStax: About Apache Cassandra, http://www.datastax.com/documentation/getting_started/doc/getting_started/gettingStartedCassandraIntro.html

[24] DataStax : CQL, http://www.datastax.com/documentation/cassandra/2.0/cassandra/cql.html

[25] Maxim Grinev. A Quick Introduction to the Cassandra Data Model, http://maxgrinev.com/2010/07/09/a-quick-introduction-to-the-cassandra-data-model/

[26] Joe Lennon. IBM developerWorks: Exploring CouchDB, http://www.ibm.com/developerworks/open-source/library/os-couchdb/index.html

[27] The Apache Software Foundation. CouchDB Documentation: Introduction, http://cwiki.apache.org/confluence/display/COUCHDB/Introduction

[28] MongoDB. The MongoDB 3.0 Manual, http://docs.mongodb.org/manual/

[29] MongoDB. Data Model Introduction, http://docs.mongodb.org/manual/core/data-modeling-introduction/

[30] Forbes, Gil Press. A Very Short Story of Data Science, http://www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history-of-data-science/

[31] Radar, Edd Dumbill. What is big data?, http://radar.oreilly.com/2012/01/what-is-big-data.html

[32] Intel IT Center. Distibuted Data Mining and Big Data

[33] James W. Taylor, Lilian M. de Menezes, Patrick E. McSharry. A comparison on univariate methods for forescating electricity demand up to a day ahead. International Journal od Forecasting 22, 2006

[34] Jeffrey Dean, Sanjay Ghemawat. MapReduce: simplified data processing on large cluster

[35] Sanjay Ghemawat, Howard Gobioff, Shun-Tank Leung. The google file system, em ACM SIGOPS Operating System Review, 2003

[36] Ronald G. Ross. The lantency of decisions. New Ideas on the Roi of Business Rules

[37] Gianpaolo Cugola, Alessandro Margara. Processing flows of information: From data stream to complex event processing. ACM Computing Surveys, 2012

[38] Lajos Jeno Fulop, Gabriella Tóth, Róbert Rácz, János Pánczél, Tamás Gergely, Arpád Beszédes, Lóránt Farkas. Survey on complex event processing and predictive analytics. Nokia Siemens Networks, 2010

[39] Andrew S Tanenbaum, Maarten van Steen. Distributed systems, volume 2. Prentice Hall Upper Saddle River, 2002

[40] Xinghuo Yu, C. Cecati, T. Dillon, M.G. Simoes. The new frontier of smart grids. Industrial Electronics Magazine, IEEE

[41] Estivill-Castro, Vladimir. Why so many clusters algorithms. ACM SIGKDD Explorations Newsletter, 2002

[42] Tryon, Robert C. Cluster Analysis: Correlation Profile and Orthometric(factor) Analysis for the Isolation of Unities in Mind and Personality

[43] EzForecaster. Forecast Methods used in ezForecaster, http://www.ezforecaster.com/fcmethod.htm

[44] John Aldrich. Fisher and Regression. Statistical science, 2005

[45] Dr. rer. nat. Florian Leitenstorfer, Vorlesung multivariate verfahren, kapitel “multivariate regression”

[46] Eugene A. Feinberg, Dora Genethliou. Load Forecasting

[47] Sami Karjalainen. Consumer preferences for feedback on household electricity consumption. Energy and Buildings, 2011

[48] Liang Zhao, Liyuan He, Wong Harry, Xing Jin. Intelligent agricultural forescating system based on wireless sensor. JNW, 2013

[49] Bernhard E. Boser, Isabelle M. Guyon, Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory, 1992

[50] Instituto Nacional de Estatística. Dados Estatísticos, https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_main

[51] Censos 2011. Instituto Nacional de Estatística, https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_base_dados

[52] Correios, Telégrafos e Telefones(CTT). Códigos Postais http://www.ctt.pt/feapl_2/app/open/postalCodeSearch/postalCodeSearch.jspx

[53] Google Maps Geoconding API. Google Developers, https://developers.google.com/maps/documentation/geocoding/

[54] Águas da Região de Aveiro. Água no dia-a-dia, http://www.adra.pt/

[55] Entidade Reguladora dos Serviços de Águas e Resíduos, http://www.ersar.pt/website/

[56] Planet Cassandra. What is Apache Cassandra?, http://planetcassandra.org/what-is-apache-cassandra/

[57] Pivotal Software. What can RabbitMQ do for you?, https://www.rabbitmq.com/features.html

[58] Matt Nedrich. An Introduction to Gradient Descendent and Linear Regression. Atomic Object, 2014

[59] Avinash Lakshman, Prashant Malik. Cassandra - A Decentralized Structured Storage System, 2010

[60] Jen Sweeney. Future Look: The Internet of Things, Vitalyst, 2013

[61] Kai Orend. Analysis and Classification of NoSQL Databases and Evaluation of their Ability to Replace an Object-relational Persistence, PhD thesis, Thecnische Univesitat Munchen, 2010

[62] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, Ion Stoica. Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing, 2012

[63] Jan Hauke, Tomasz Kossowski. Comparison of values of Pearson's and Spearman's correlation

coefficient on the same sets of data, 2011

[64] Joe Kuan. Learning HighCharts, 2012

[65] OpenStreetMaps Foundation. OpenStreetMaps, https://www.openstreetmap.org/