DATA LAKES APLICADOS À INTERNET DAS COISAS: UMA REVISÃO SISTEMÁTICA DA LITERATURA

Ysabelle Pinheiro de Sousa (IC) e Orlando Bisacchi Coelho (Orientador)
Apoio:PIBIC Mackenzie

RESUMO

Atualmente, com o desenvolvimento da tecnologia, convivemos com uma gama de objetos que geram dados a todo momento, sejam eles dispositivos eletrônicos que são conectados à Internet ou objetos não eletrônicos que geram dados relevantes, como por exemplo: balança e termômetro não digital. A partir deste cenário, surge o conceito de “Internet das Coisas”, que se refere à possibilidade de captar dados de qualquer tipo de objeto, inteligente ou não, e, a partir desses dados, gerar conhecimento e novas possibilidades de negócios ou valor. Esses dados são obtidos através de sensores e enviados para computadores com a finalidade de serem analisados para tomada de decisão. Devido à essa comunicação entre diversos objetos, gera-se um grande volume e heterogeneidade de informação, o que consequentemente dificulta o processo de captação e análise, e acaba por limitar a eficiência em gerar *insights* para o negócio. Sendo assim, surge o conceito de *Data Lake*, que pode ser definido como uma plataforma única de armazenamento (no formato original), governança e visualização de dados, com o propósito de viabilizar seus recursos a diversos usuários, permitindo um maior acesso a esses dados por diversos segmentos do negócio. Este artigo tem como objetivo desenvolver uma Revisão Sistemática da Literatura, com foco em uma solução adequada de *Data Lake* para IoT, considerando todos os passos de coleta, limpeza, armazenamento, pré-processamento, qualificação, análise e visualização dos dados coletados por sensores, para que esses dados estejam qualificados para serem analisados e utilizados para tomadas de decisão.

Palavras-chave: *Data Lake*. Internet das Coisas. Revisão Sistemática da Literatura.

ABSTRACT

Currently, with the development of technology, we live with a range of objects that generate data at any moment, whether electronic devices that are connected to the Internet or non-electronic objects that generate relevant data, such as: scales and non-digital thermometer. From this scenario, the concept of "Internet of Things" arises, which refers to the possibility of capturing data of any type of object, intelligent or not intelligent, and from this data, generate knowledge and new possibilities of business or value. These data are obtained through sensors and sent to computers for the purpose of being analyzed for decision making. Due to this communication between several objects, a great volume and heterogeneity of information

is generated, which consequently hinders the capture and analysis process, and ends up limiting the efficiency in generating insights for the business. Thus, the concept of Data Lake arises, which can be defined as a unique storage platform (in the original format), governance and visualization of data, with the purpose of enabling its resources to several users, allowing greater access to this data by several segments of the business. This article aims to develop a Systematic Review of Literature, focusing on a suitable Data Lake solution for IoT, considering all steps of collection, cleaning, storage, preprocessing, qualification, analysis and visualization of the data collected by sensors, so that these data are qualified to be analyzed and used for decision making.

Keywords: Data Lake. Internet of Things. Systematic Review of Literature.

1. INTRODUÇÃO

A Internet das Coisas ou, no termo em inglês, *Internet Of Things* (IoT) é um dos assuntos mais discutidos atualmente no mundo da tecnologia. Composta pelos termos "Internet", o qual é um sistema global de redes de computadores interconectados que serve bilhões de pessoas ao redor do mundo e "*Things*" que pode ser considerada como qualquer objeto que pode ser distinguido na vida real, não somente dispositivos eletrônicos que utilizam tecnologias avançadas, mas também "coisas" que não consideramos eletrônicas como comida, roupas, *displays* de propaganda, animais e plantas, por exemplo. Resumidamente, "coisas" são objetos reais no mundo físico e material (MADAKAM et al, 2015). Sendo assim, não há uma única definição disponível para IoT, mas o que há em comum entre as definições de pesquisadores, acadêmicos, pessoas do mundo corporativo e inovadores por exemplo, é a ideia de que a primeira versão da Internet é baseada em dados gerados por pessoas e agora, nessa nova versão, em dados gerados por "coisas" (MADAKAM et al, 2015).

Segundo o *Gartner Hype Cycle* (JAYAVARDHANA et al., 2013), um modo de representar a emergência, adoção, maturidade e impacto das tecnologias, mostra que a tecnologia IoT, levará de 2 a 5 anos para adoção no mercado (contando a partir de 2017), baseado no gráfico de 2017 (Figura 1).

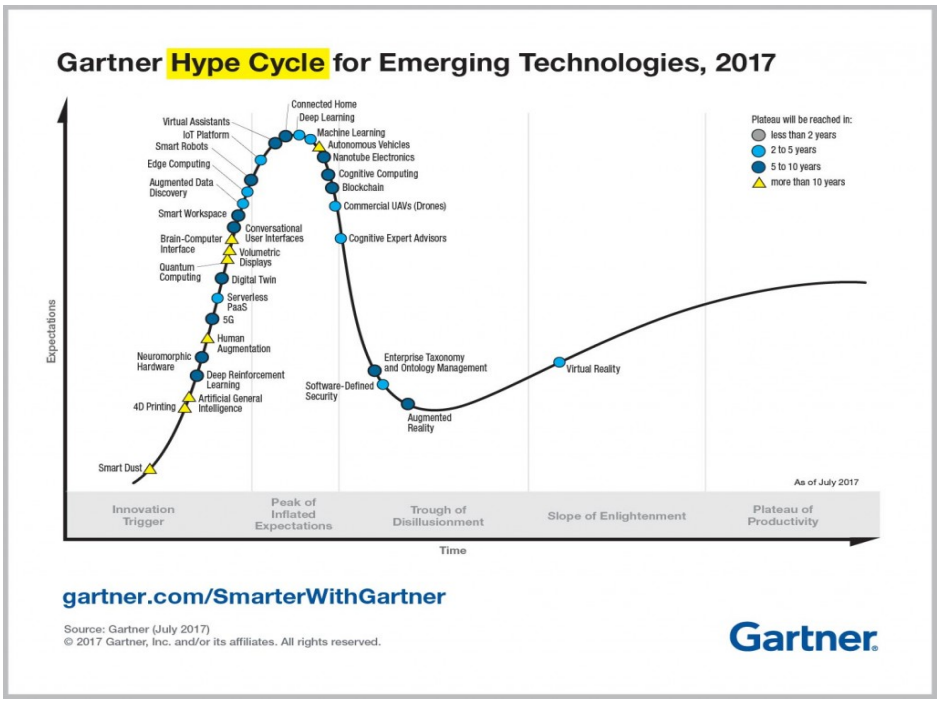

Figura 1 - *Gartner Hyper Cycle of* 2017, disponível em: https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/

Este impacto será positivo pois trará diversos benefícios para o nosso dia a dia, simplificando e trazendo mais conforto em nossa vida, além de possuir diversas aplicações na área de medicina, indústria, transporte, educação, por exemplo.

Em IoT, sensores – dispositivos responsáveis por captar dados analógicos no contexto que estão inseridos – e atuadores – que executam uma ação dentro do contexto inserido – implantados em objetos físicos são conectados através de redes com ou sem fio, gerando grande volume de dados que são enviados para computadores para serem analisados. Madakam (2015) et al., cita alguns pré-requisitos para que a implementação da IoT seja efetiva: demanda dinâmica de recursos, necessidade de dados em tempo real, crescimento da demanda, disponibilidade de aplicações, proteção de dados e privacidade do usuário, consumo de energia eficiente de aplicativos, aplicações executadas próximas aos usuários finais e acesso à um sistema de nuvem aberto e operável. Ele também acrescenta que a implementação do IoT requer três componentes: (i) uma camada de *hardware*, composta por sensores, atuadores e componentes embarcados de comunicação; (ii) uma camada de *middleware* que possui capacidade de armazenamento e ferramentas analíticas que utilizam nuvem e um ambiente propício para lidar com *Big Data*; e (iii) uma camada de apresentação, possuindo ferramentas de fácil entendimento da visualização e interpretação dos dados consumidos e que pode ser customizada para diversas aplicações. Por ser um conceito muito vasto, não há uma arquitetura uniforme e correta para a implementação de IoT, mas todas as implementações bem-sucedidas possuem direta ou indiretamente as três camadas citadas anteriormente.

Devido a essa comunicação e dinamismo entre objetos, gera-se grande quantidade de dados variados, os quais geram os maiores desafios das corporações que são justamente coletar e analisar estes dados. Sendo assim, surge o conceito de *Data Lake*, que pode ser considerado como uma plataforma única que combina armazenamento de dados no seu formato original, governança de dados e *analytics*, desenhada para direcionar estes desafios (MALLORY and WRIGH, 2017). O *Data Lake* é uma plataforma centralizada, segura, com armazenamento baseado em nuvem, que permite ingestão de dados estruturados e não-estruturados assim como a possibilidade de transformar estes dados não tratados de acordo com cada necessidade, já que o *Data Lake* viabiliza o acesso a estes dados e suas ferramentas analíticas a diversos usuários, abrangendo variados segmentos de negócio os quais requerem *insights* específicos.

Com a necessidade de armazenamento destes novos tipos de dados, o *Data Lake* está começando a ganhar mais espaço no mercado e começa a substituir

gradativamente o *Data Warehouse*, no qual não possui a capacidade de armazenar dados não-estruturados, mas somente dados com formato tabelar, deixando de atender as necessidades de negócio do mercado atual. Desse modo, comparando algumas variáveis entre as duas arquiteturas citadas, o *Data Lake* destaca-se em pontos como: custo baixo de armazenamento, maior *performance* no processo de ETL (*Extract, Transform and Load*) e principalmente tipos variados de dados (KHINE P., 2017).

Atualmente, há uma quantidade pequena de trabalhos disponíveis que agregam os dois conceitos, de *Data Lake* e IoT, para orientar o leitor a seguir uma arquitetura adequada de implementação destas duas tecnologias em conjunto. Por isso o projeto aqui proposto, que tem como objetivo apresentar uma Revisão Sistemática da Literatura, com a metodologia discutida em Okoli e Schabram (2010), detalhando todo o processo de pesquisa, a fim de investigar a melhor solução para coleta, limpeza, armazenamento, pré-processamento, qualificação e visualização dos dados coletados por sensores IoT.

2. REFERENCIAL TEÓRICO

Uma Revisão Sistemática da Literatura (RSL) é um estudo que utiliza métodos explícitos e sistemáticos para identificar, selecionar, criticar e avaliar criteriosamente uma área de pesquisa relevante, coletando e analisando dados de um conjunto importante de estudos selecionados na área, podendo ser utilizados métodos estatísticos (meta-análise) no processo (MOHER et. al, 2009). Segundo Fink (2005), uma RSL deve ser sistemática na abordagem metodológica, explícita na explicação dos procedimentos adotados, compreensível no escopo da inclusão de todo o material relevante e reproduzível por outros que seguirem a mesma abordagem na revisão do tópico. O valor da RSL é caracterizado pelo o que foi feito, o que foi encontrado e pela clareza do relatório. Nesta RSL foi utilizada a metodologia de Okoli e Scahabram (2010), o guia mais recente de RSL na área de Sistemas de Informação, no qual consiste, em um método de oito passos: Propósito da Revisão, Protocolo e treinamento, Pesquisa da Literatura, Método Prático, Avaliação de Qualidade, Extração de Dados, Síntese dos Estudos e a Escrita da Revisão. Vale mencionar que, além da metodologia proposta por Okoli e Scahabram, também existe o PRISMA (MOHER et. al, 2009), o qual possui um *checklist* robusto do processo de Revisão da Literatura, o que permite um desenvolvimento qualificado do processo; porém este é voltado para a área da saúde.

Deste modo, o objetivo desta RSL é investigar a existência de modelos adequados de arquitetura para armazenamento de dados de sensores juntamente com as melhores práticas de visualização de dados, com a finalidade de gerar melhores *insights* e monitoramento de dados colhidos em tempo real. Dado isso, as perguntas da pesquisa que a revisão pretende responder são:

P1) Atualmente, existem modelos de arquitetura *Data Lake* para armazenamento de dados de sensores?

P2) Há um processo adequado para garantir a qualidade dos dados captados por esses sensores?

A metodologia que foi realizada para responder as perguntas citadas acima está detalhada no tópico seguinte.

3. METODOLOGIA

Como somente uma única pessoa realizou a revisão, não houve protocolos desenvolvidos. Com o objetivo de selecionar estudos científicos claros em relação à finalidade da pesquisa, as bases de dados resultantes da pesquisa do Portal do CAPES foram: *SpringerLink*, MDPI, *SAGE Journals*, Elsevier e *ResearchGate*. Somente artigos em inglês e revisados por pares foram considerados, em um período de publicação abrangendo os últimos 5 anos. As palavras-chave que foram utilizadas na pesquisa foram ("*Data Lake*" AND "IoT"), nas quais foram consideradas em toda a extensão do texto, inclusive no título e *abstract*. A pesquisa foi realizada em janeiro de 2018.

O resultado da busca relacionado à existência de modelos de arquitetura *Data Lake* resultou em 190 artigos utilizando os primeiros critérios de busca mencionados anteriormente. Após a aplicação de mais um filtro demandando que os artigos contivessem pelo menos um dos seguintes termos - *Cloud Computing, Big Data, Sensors, Internet of Things ou Computer Science* - foram obtidos 86 artigos não distintos, eliminando, deste modo, artigos não pertinentes ao assunto da pesquisa. Após este filtro, a busca (OKOLI and SCHABRAM, 2010) continuou com base na leitura do título, *abstract* e palavras-chave de cada artigo e quando necessário, uma leitura rápida sobre o texto, para uma abstração maior de cada artigo resultante nesta etapa da pesquisa. Considerando os artigos que possuem as palavras-chave definidas no título ou no *abstract*, chegou-se a 33 artigos. Após, foram desconsiderados os artigos de consulta paga, no qual excluiu uma grande quantidade de artigos que seriam relevantes para a pesquisa, restando 18 artigos.

Após a concretização dos artigos obtidos, 9 artigos foram excluídos devido à inclusão de um critério de qualidade, como por exemplo trabalhos que possuem um caso de uso testado ou uma discussão teórica bem desenvolvida. Este foi o último critério de exclusão considerado no processo. No final, restaram 9 artigos, os quais foram lidos na íntegra para extrair as informações necessárias para responder as três questões listadas anteriormente. Os resultados do processo são apresentados na sessão seguinte.

4. RESULTADO E DISCUSSÃO

Com a palavra-chave: ("*Data Lake*" e "IoT"), junto com os critérios de qualidade e exclusão usados, a pesquisa resultou em 9 artigos, publicados entre os anos de 2013 a 2018. O número de artigos publicados em cada ano está exibido na Figura 2.

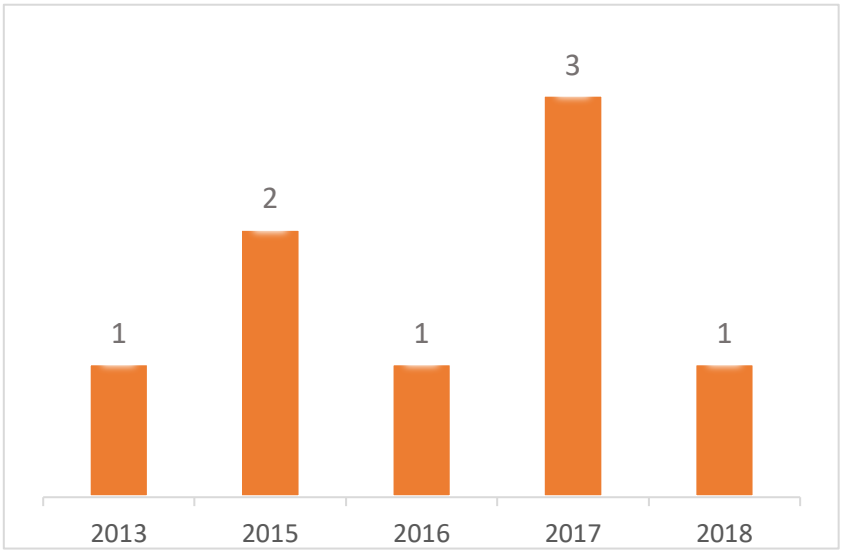

Figura 2 - Quantidade de artigos por ano de publicação

Para responder à primeira pergunta proposta, sobre as arquiteturas para *Data Lake* existentes voltadas para o armazenamento de dados de sensores, é necessário relembrar que, como já dito, a proposta do *Data Lake* é armazenar os dados do modo que foram produzidos, sem pré-processamento, até se ter conhecimento de como serão analisados.

Os sistemas tradicionais de gerenciamento de dados atendem ao armazenamento, recuperação, e atualização de itens, registros e arquivos de dados elementares. No contexto do IoT, os sistemas de coleta de dados podem fazer uma série de processamentos básicos, com resumo de dados ou outras tarefas *on-line* que visam a transmissão mais eficiente dos dados de sensores até o ambiente onde reside o *Data Lake*. Por outro lado, o *Data Lake* deve fornecer instalações de armazenamento, *log* e auditoria para análises *off-line* mais complexas (ABU-ELKHEIR,

2013). Isso expande o conceito de gerenciamento de dados do armazenamento *off-line*, possibilitando que filtragem e sumarização ocorram tanto no processo *on-line* quanto no *off-line*.

Em todos os artigos analisados, direta ou indiretamente é utilizado um modelo de arquitetura de três camadas denominada Lambda (WARREN e MARZ, 2015), que é descrito de forma bem clara por Visakh (2015) e será utilizado como a principal fonte de detalhamento dessa arquitetura que será aprofundada mais adiante.

Criada por Nathan Marz (NAIR, 2015), a arquitetura Lambda é um modelo de arquitetura que é capaz de lidar com uma grande escala de dados (NAIR, 2015). Muitas das soluções desenvolvidas para *Big Data* são uma evolução das soluções para BI (*Business Intelligence*), pois o modelo de arquitetura de BI, que trata de modelos relacionais, não pode ser reutilizado na arquitetura de *Big Data*, a qual trata de diversas quantidades de dados heterogêneos (NADAL, 2017). Ao longo dos anos, o fenômeno *Big Data* foi se caracterizando por seus cinco V's: Volume, Velocidade, Variedade, Variabilidade e Veracidade.

Reforçando a ideia de que uma única ferramenta ou aplicação não é suficiente para resolver todos os problemas de processamento de *Big Data*, a arquitetura Lambda foi construída em camadas; cada ferramenta constrói uma camada e cada camada realiza funcionalidades e construções específicas sobre a anterior.

Um sistema construído no modelo de arquitetura Lambda é caracterizado por um sistema robusto e tolerante a falhas. Deste modo, a arquitetura segue três princípios:

- *Fault-Tolerance*: Tolerância a falhas

- *Data Immutability*: O sistema armazenará os dados, mas nunca poderá atualizá-los. Considerando o modelo de persistência CRUD (NAIR, 2015), podemos criar, ler e deletar, mas não atualizar.

- *Recomputation*: É possível reprocessar os resultados a qualquer momento.

A Arquitetura Lambda possui três camadas, como mostrado na Figura 3.

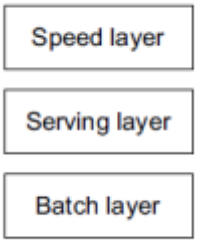

Figura 3 - Camadas da Arquitetura Lambda

A *Batch Layer* (Camada em Lotes) é considerada o coração da arquitetura Lambda. Caracterizada como de alta latência; esta é responsável por armazenar a cópia *master* de todos os dados, na qual é imutável, e processar funções em cima deste conjunto de dados, o que constitui o *Batch Processing* (NAIR, 2015). O *Batch Processing* é responsável por preparar, transformar e executar algoritmos iterativos sobre os dados armazenados no *Data Lake* para moldá-los de acordo com cada necessidade analítica do caso de uso selecionado (NADAL, 2017).
A *Serving Layer* (Camada de Serviço) é responsável pelo rápido acesso a leitura dos resultados computados na *Batch Layer.*
Já a *Speed Layer* (Camada de Velocidade) é caracterizada por baixa latência, sendo responsável por lidar somente com novos dados em tempo real que serão processados e disponibilizados na *Serving Layer*, onde poderão ser mesclados com os dados do *Batch Layer* para serem consumidos por ferramentas de visualização de dados.
Um esboço da arquitetura Lambda pode ser visto na Figura 4.

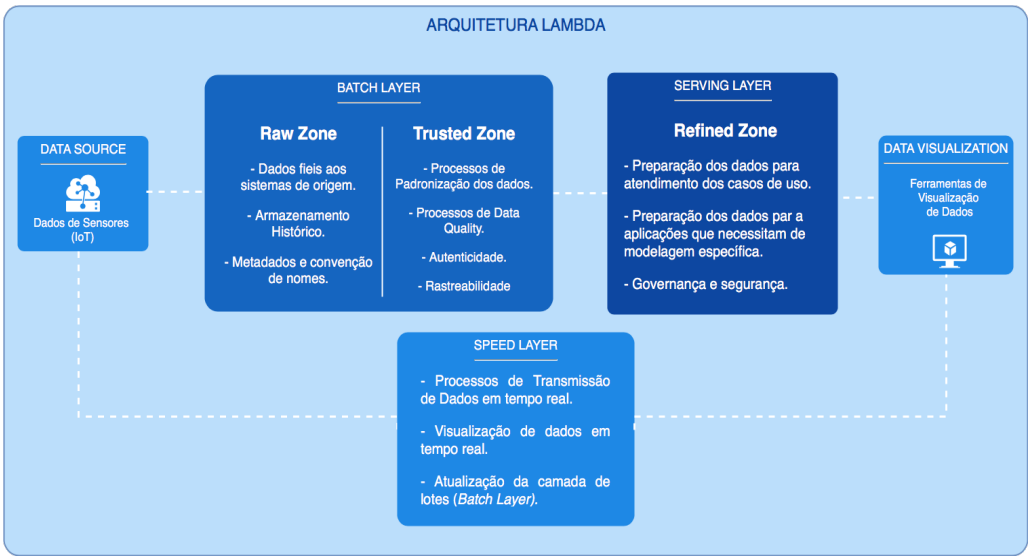

Figura 4 - Arquitetura Lambda Proposta

Dentro da *Batch Layer* há duas zonas: *Raw Zone* (Zona Crua) e *Trusted Zone*. (Zona Confiável). Na *Raw Zone* são armazenados os dados imutáveis; na *Trusted Zone* acontece o tratamento e padronização dos dados. Já na *Serving Layer* existe a *Refined Zone* (Zona Refinada) na qual acontece a preparação e filtro dos dados para atender os *user cases* ou aplicações que necessitam de uma modelagem específica, além da governança e segurança dos dados.

Na Figura 5 e no texto que lhe segue apresentamos uma implementação de referência do modelo Lambda.

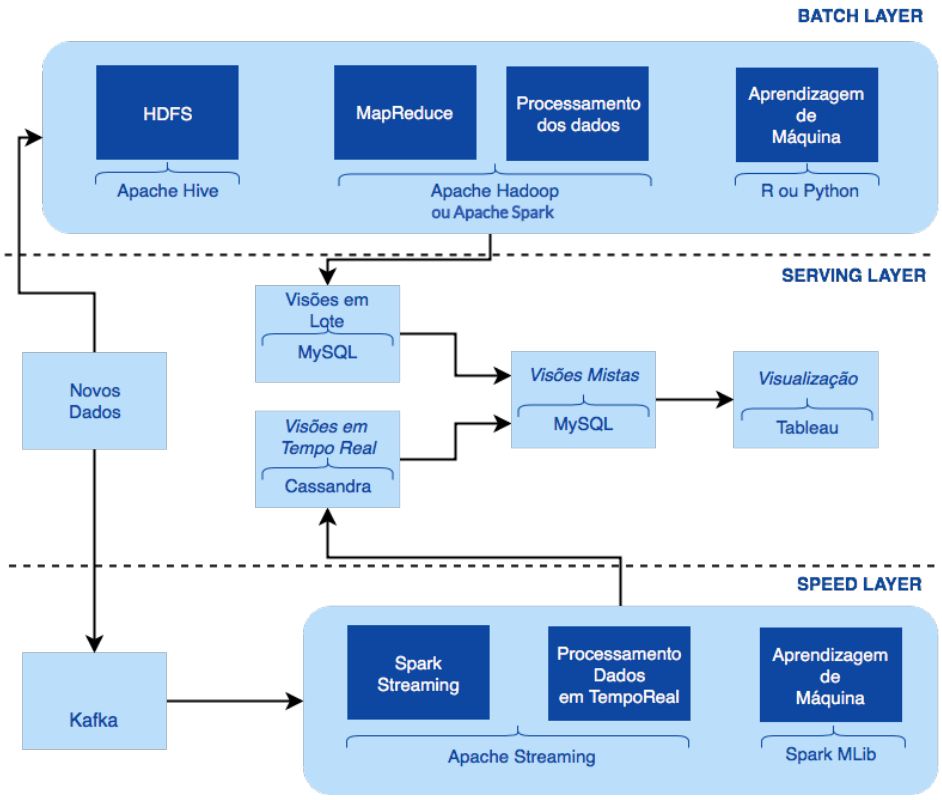

Figura 5 - Ferramentas sugeridas para cada camada
da Arquitetura Lambda (NAIR, 2015).

O Batch Layer é implementado usando as seguintes tecnologias (NAIR, 2015). *Apache Hadoop* (2018) é uma estrutura desenvolvida para uma computação confiável, escalável e distribuída, que permite o processamento de grandes conjuntos de dados em *clusters* de computadores através de modelos de programação simples. Seus principais componentes são: *Hadoop Distributed File System* (HDFS)(2018), que é o componente responsável pelo armazenamento distribuído, e *Hadoop MapReduce* (2018), responsável pelo processamento paralelo dos conjuntos de dados.

Focado em tarefas de consultas interativas, *machine learning* e processamento em tempo real, o *Apache Spark* (2018) que também é um sistema de processamento distribuído, torna-se uma alternativa ao Hadoop. Ele utiliza memória cache, resultando

em uma performance melhor, além de possuir alto rendimento e tolerância a falhas; e pode ser programado em Scala, Java ou Python. Porém, o Spark não possui um sistema próprio de armazenamento (como o Hadoop), mas permite integração com qualquer sistema de armazenamento, como o HDFS por exemplo. Muitas empresas, acabam utilizando o Spark junto com o Hadoop, pois este pode compartilhar do mesmo componente de gerenciamento de recursos computacionais que o Hadoop possui, o YARN (2018), garantindo mais um nível de consistência na aplicação.

Uma alternativa para o Apache Spark é a combinação Apache Storm (2018) e Apache Trident (2018). Apache Storm, é um sistema distribuído de computação em tempo real, escalável e tolerante a falhas, no qual pode ser implementado *real-time analytics* e aprendizagem de máquina *online*. O Trident oferece um alto nível de abstração no Storm com consultas distribuídas de baixa latência e a garantia de pelo menos um único processamento.
Apache Hive (2018) é o componente responsável pelo acesso aos dados armazenados. Nele é possível consultar, estruturar, resumir e processar os dados, tanto estruturados quanto não estruturados, através de uma linguagem própria chamada Hive QL, que foi construída com base na linguagem SQL.

Para o *Serving Layer*, para os dados que vão ser acessados em lote, pode ser utilizado o banco de dados relacional MySQL (2018), que pode ser conectado, sem dificuldade, com qualquer ferramenta de visualização comercial ou *open-source*. Já o banco de dados não relacional Cassandra (2018), que oferece alta escalabilidade, disponibilidade e tolerância a falhas, é a solução mais adequada para dados em tempo real. Sua linguagem é a CQL, que possui uma sintaxe parecida com SQL. Temos, como alternativa para o Cassandra, tanto o Apache HBase (2018) que é um modelo de banco de dados distribuído NoSQL incluso no ecossistema Hadoop, como o Riak (2018) que é um banco não relacional construído no formato “chave-valor”.
Já o *Speed Layer* pode ser implementado utilizando as tecnologias Apache Spark e Apache Kafka (2018). O Apache Kafka é um sistema de código aberto para gerenciamento de fluxo de mensagens em tempo real (*message broker application*) que permite a criação de aplicativos em tempo real. Para garantir tolerância a falhas, ao invés de as mensagens serem deletadas após consumidas elas são armazenadas por um período configurável. Caso o processo de *streaming* perca algum dado, ele pode ser recuperado. Após a captação destes dados pelo Kafka, o Apache Spark, já mencionado e explicado anteriormente, processa, filtra, agrega e transforma essas mensagens, possibilitando seu consumo pelas ferramentas de visualização de dados.

Para a camada de visualização de dados, há diversas ferramentas que poderiam ser utilizadas. Uma das que mais se destacam é o Tableau (2018), que é uma plataforma profissional de visualização de dados que oferece *drivers* nativos de conexão para as principais plataformas de *Big Data* (Spark SQL, Hadoop, Hive, MySQL, etc) e uma das mais utilizadas no mundo corporativo atualmente.

Um dos modelos que diretamente abordou a arquitetura Lambda foi criado por Sergi Nadal (2017), nomeado de Bolster, que é um refinamento da arquitetura Lambda e nele é acrescentado mais uma camada ao modelo original, chamada de *Semantic Layer* (Camada Semântica). Ela contém o Sistema de Gerenciamento de Meta-dados que é responsável por fornecer a informação necessária para descrever e modelar os dados históricos.

Segundo Sergi Nadal (2017), a arquitetura Lambda não atende os dois último V's da arquitetura de *Big Data* (Variabilidade e Veracidade), daí surge sua proposta de arquitetura denominada Bolster, que inclui uma nova camada que resolveria esta divergência e abraçaria todos os V's abordados pela arquitetura. Esta camada é denominada *Semantic Layer* e contém o Sistema de Gerenciamento de Metadados, o qual é responsável por fornecer aos outros componentes as informações necessárias para descrever e modelar os dados históricos. Porém esta camada é utilizada para modelar os dados com uma visão mais "amigável" e voltada para negócio, destinada a usuários que não possuem o conhecimento técnico necessário, no caso a linguagem SQL, para modelar os dados diretamente da fonte.
Caso se deseje implementar a *Semantic Layer*, Sergi Nadal (2017) sugere utilizar as ferramentas Apache Stanbol (2018) ou Apache Atlas (2018), que possuem o mesmo propósito de trazer a tecnologia da semântica para os sistemas de conteúdo, porém destaca que o Apache Atlas satisfaz os principais requisitos mais naturalmente do que o Apache Stanbol, o que provavelmente o torna a melhor escolha.

Dentro dos outros artigos que respondem a primeira pergunta, todos abordam um tipo de arquitetura que são similares ao modelo Lambda. Por exemplo, o autor Abu-Elkheir (2013) foca em uma solução para gerenciamento de dados de IoT que tem como base as soluções comuns de gerenciamento de dados IoT baseadas em WSN - um subconjunto de IoT que tem como foco coleta de dados em tempo real para tomada de decisão rápida - mas que possui um aprimoramento sobre como lidar com as três principais funções de manipulação de dados: coleta, armazenamento e processamento, adicionando novas camadas que seriam necessárias para um gerenciamento a longo prazo, já que ele destaca que os modelos atuais possuem capacidade de armazenamento limitada para uso a longo prazo. Sendo assim, o seu

modelo proposto seria capaz de explorar volumes de dados heterogêneos, sendo possível encontrar padrões globais e oportunidades estratégicas interessantes.

Já o autor Cun Ji e seus colegas (2016), propõe um modelo de dispositivo de ingestão de dados de diferentes tipos para uma plataforma industrial de *Big Data*. Ele ressalta que um dos desafios que a Internet das Coisas está enfrentando atualmente é como ingerir estes dados heterogêneos em grande escala que provém de diferentes dispositivos, assim como suprir a necessidade de possuir um ecossistema *Big Data* para consumi-los. Sendo assim, sua solução é semelhante à estrutura Lambda, inclusive utilizando as mesmas tecnologias. Além disso, ele propõe algumas estratégias de processamento de dados para sincronização de dados, fatiamento de dados, divisão de dados e indexação de dados, respectivamente; nos quais ajudam no processo de ingestão de dados heterogêneos e a unificação dos mesmos em um formato único.

Por último, o Meilan Jiang e colegas (2015), propõem um sistema de monitoramento de meio ambiente em tempo real (RTEM - *Real Time Environmental Monitoring*), destacando que os sistemas atualmente existentes não possuem um modelo de sistema bem definido, tornando-se como desafio integrações com outros sistemas, suporte a diferentes tipos de sensores e suporte a um monitoramento de larga escala, por exemplo. Dessa forma, ele também propõe uma solução que se assemelha ao Lambda, já que possui camadas dedicadas ao recebimento dos dados de sensores, outra de armazenamento, outra de atualização e administração dos dados *real time*.

Portanto, após a análise dos artigos e soluções propostas, conclui-se que não há um modelo de arquitetura que seja correto, mas sim que isso depende do problema a ser resolvido. Todos se assemelham à arquitetura Lambda, e a partir dela, há alguns aprimoramentos que se ajustam ao negócio em questão. Sendo assim, tomar como base a arquitetura Lambda, talvez seja a melhor solução inicial para o problema proposto.

A segunda pergunta contempla como adquirimos valor dos dados obtidos, após armazená-los. Este conceito chama-se *Data Quality*, que fornece as condições iniciais para as etapas necessárias de preparação do uso dos dados. Segundo Agung Wahyudil e seus colegas (2018), um dos desafios atualmente das empresas é controlar estes dados, já que estes provêm de diversas fontes e quando não são tratados de maneira adequada, torna a tarefa um pouco mais árdua. Wahyudil propõe um modelo padrão de processo que uma organização pode utilizar para lidar com os

dados no quesito de qualidade, permitindo identificar, avaliar a qualidade, organizar e combinar dados; com o objetivo final de retirar valor dos dados obtidos, realçando que este deve ser independente da tecnologia que está sendo utilizada. Ele propõe um modelo que é exibido na Figura 6.

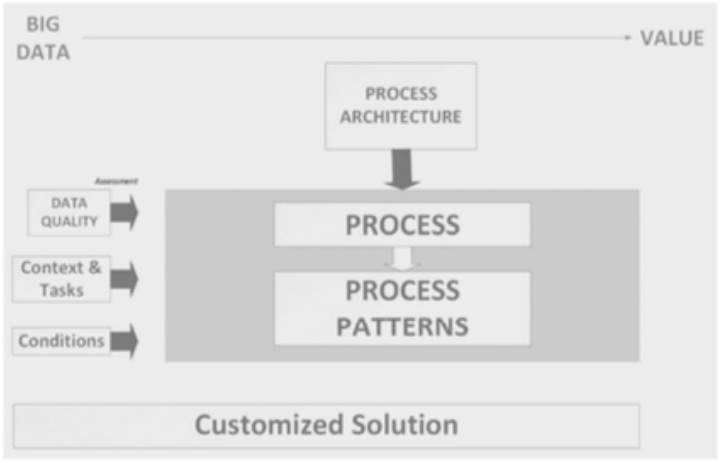

Figura 6 - Modelo proposto para aquisição de qualidade dos dados Big Data (AGUNG et al., 2018)

O estágio de processamento dos dados se inicia considerando os contextos e tarefas que representam o objetivo da reutilização dos dados, ou seja, para qual propósito os mesmos serão analisados. Por exemplo, um sistema de detecção de fraudes e um sistema de gerenciamento do ciclo de vida de um cliente de uma empresa: os dados que serão analisados são os mesmos, mas os procedimentos analíticos não.

Após o estudo de seu caso de uso, Agung (2018) et al. afirmam que há diversos processos para cada tipo de conjunto de dados que dependem de sua qualidade. Para atingir o processo “ideal”, foi necessário seguir alguns passos de acordo com sua classificação, que é exibida na Figura 7.

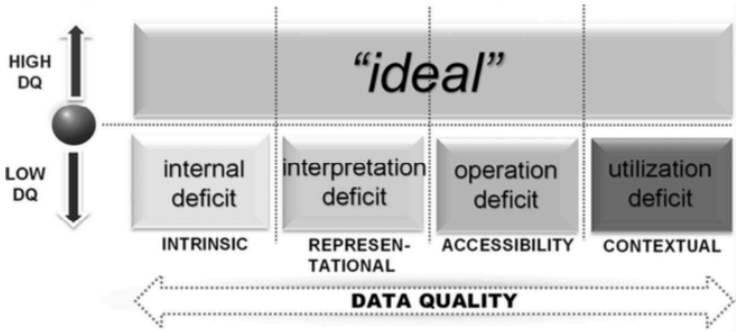

Figura 7 - Classificação de padrões (WAHYUDIL et al., 2018)

Segundo o autor, as organizações frequentemente lidam com conjunto de dados de baixa qualidade, como por exemplo, dados captados por sensores IoT, precisam passar por um processo de transformação para que se tornem adequados para serem utilizados. E, conforme mostra a Figura 7, existem 4 tipos de baixa qualidade. Seguem abaixo os tipos e suas estratégias de solução.

• *Intrinsic*: A baixa qualidade intrínseca gera um déficit interno, ou seja, quando os usuários internos percebem as propriedades internas dos dados de baixa qualidade, como sendo tendenciosos, imprecisos, não confiáveis, etc. Para resolver este problema, as organizações podem melhorar a autenticidade dos conjuntos de dados, avaliando a precisão e representatividade dos mesmos, classificando a credibilidade dos provedores de dados e os pré-processando, por exemplo.

• *Representational*: Se o conjunto de dados possuem uma baixa qualidade de representação, como por exemplo o formato inconsistente ou falta de metadados, as organizações enfrentam o déficit de interpretação, ou seja, a divergência entre o entendimento real dos dados e a interpretação correta. A interpretação precisa é essencial para compreensão do valor dos dados para então serem explorados e analisados. As estratégias que podem ser utilizadas para diminuir o déficit de interpretação são: 1) Padronização (ex: forma de observação e metadados) e 2) Fornecimento de Repositório para toda a organização (ex: terminologia, semântica, biblioteca de sentimento, etc).

• *Acessibility*: As organizações que possuem dificuldade para operar conjunto de dados com baixa qualidade de acessibilidade, como déficit de operação ou acesso limitado, podem impedi-los de criar *insights* que possam ser descobertos somente com dados de alta granularidade. As estratégias para resolver este problema implicam que os provedores/proprietários dos dados executem algumas tarefas como privacidade por *design* (proteção dos dados através de tecnologia), contrato entre o provedor dos dados e usuários e conformidade com a regulamentação.

• *Contextual*: O déficit de utilização ocorre quando existe uma lacuna entre o contexto esperado e o resultado real da utilização dos dados, como por exemplo, quando os dados não são oportunos para o problema em questão ou quando os dados são muito complexos para serem analisados e gerarem *insights*. A estratégia para este problema é aprimorar a capacidade analítica da organização, adotando tecnologias de *big data*, por exemplo.

Dessa forma, essas etapas orientam as organizações para definir um processo adequado de extração de valor de seus dados a partir da variável em que ela se encontra, dentro dos 4 níveis de *Low Data Quality*. Outro autor que aborda o conceito de *Data Quality* é o Sergi Nadal (2017), criador da arquitetura Bolster. Por outro lado, ele aborda um lado mais técnico de como implementar esta camada de semântica dentro da arquitetura, já mencionado e explicado anteriormente.

5. CONSIDERAÇÕES FINAIS

Após o processo de pesquisa, estudo dos artigos abordados e elaboração da Revisão Sistemática da Literatura, conclui-se que o objetivo proposto foi atingido com sucesso, o que permitiu a investigação de soluções existentes em *Data Lake* para armazenamento de dados de sensores IoT, chegando a conclusão de que as soluções encontradas abordam, direta ou indiretamente, a arquitetura denominada Lambda. Esta possui as camadas fundamentais para todo o processo de captação, limpeza, armazenamento, pré-processamento e visualização dos dados coletados por sensores IoT. As demais soluções, por exemplo, possuem camadas com propósitos semelhantes além de complementar a arquitetura Lambda, como é o caso da solução denominada Bolster, que acrescenta uma camada semântica, que tem como objetivo qualificar os dados que serão analisados.

Outra investigação que a RSL permitiu foi a existência de um processo adequado para garantir a qualidade dos dados captados pelos sensores IoT. Dos artigos selecionados, o único processo que aborda teoricamente as etapas para qualificação dos dados é o criado por Agung Wahyudil (2018) et al., que sugere a partir de uma classificação que a corporação se encontra em questão de deficiência dos dados, diversas tarefas que a mesma pode seguir para solucionar ou amenizar os desafios relacionados a *Data Quality.* Além desta solução teórica, o autor Sergi Nadal (2017), sugere como implementar esta camada através de sua arquitetura Bolster, já mencionada anteriormente.

Sendo assim, a pesquisa aqui proposta teve como objetivo apresentar uma Revisão Sistemática da Literatura, afim de orientar o leitor a elaborar uma arquitetura adequada de Data Lake para armazenamento de dados de sensores IoT, juntamente com o processo de qualificação destes dados para análise, já que atualmente um dos maiores desafios das corporações é como armazenar estes dados e prepará-los adequadamente para tomadas de decisão de seus negócios internos.

6. REFERÊNCIAS

ABU-ELKHEIR, M.; HAYAJNEH, M.; ABU ALI, N. **Data Management for the Internet of Things: Design Primitives and Solution**. Faculty of Computer and Information Sciences, Mansoura University, 2013.

APACHE ATLAS: ferramenta de gerenciamento e governança de metadados. Disponível em <http://atlas.incubator.apache.org>. Acesso em: 17 de jun. 2018.

APACHE HADOOP: *framework* para processamento de grande volume de dados. Disponível em <http://hadoop.apache.org>. Acesso em: 16 de jun. 2018.

APACHE HBASE: banco de dados. Disponível em <http://hbase.apache.org>. Acesso em: 17 de jun. 2018.

APACHE HDFS: armazenamento distribuído utilizado para aplicações Hadoop. Disponível em <http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html - Purpose>. Acesso em: 16 de jun. 2018.

APACHE HIVE: *software* de *Data Warehouse*. Disponível em <https://hive.apache.org/>. Acesso em: 17 de jun. 2018.

APACHE KAFKA: sistema com envio distribuído de mensagens para criação de aplicativos em tempo real. Disponível em <https://aws.amazon.com/pt/kafka/>. Acesso em: 17 de jun. 2018.

APACHE MAP REDUCE: sistema para processamento paralelo de grande volume de dados. Disponível em <http://hadoop.apache.org>. Acesso em: 16 de jun. 2018.

APACHE SPARK: sistema de processamento distribuído utilizado por fluxos de Big Data. Disponível em <https://aws.amazon.com/pt/big-data/what-is-spark/>. Acesso em: 16 de jun. 2018.

APACHE STANBOL: ferramenta de gerenciamento de conteúdo semântico. Disponível em <https://stanbol.apache.org>. Acesso em: 17 de jun. 2018

APACHE STORM: sistema distribuído para computação em tempo real. Disponível em <http://storm.apache.org/index.html>. Acesso em: 17 de jun. 2018.

APACHE YARN: *framework* para agendamento de tarefas e gerenciamento de recursos. Disponível em <http://hadoop.apache.org>. Acesso em: 16 de jun. 2018.

CASSANDRA: banco de dados. Disponível em <http://cassandra.apache.org>. Acesso em: 17 de jun. 2018.

CUN, J. et al. **Device Data Ingestion for Industrial Big Data Platforms with a Case Study**. School of Computer Science & Technology, Shandong University, Jinan 250101, China, 2016.

FINK, A. **Conducting Research Literature Reviews: From the Internet to Paper**. 2 ed. Thousand Oaks: Sage Publications, 2005.

JIANG, M. et al. **A Data Stream-Based, Integrative Approach to Reliable and Easily Manageable Real Time Environmental Monitoring**. Department of Advanced Technology Fusion, Konkuk University, 120 Neungdong-ro, Gwangjin-gu, Seoul 05029, Republic of Korea, 2015.

KHINE, P e WANG, Z. **Data Lake: A New Ideology in Big Data Era.** Conference Paper, 2017.

MALLORY, J e WRIGHT, R. **Building Big Data Storage Solutions (Data Lakes) for Maximum Flexibility.** Amazon Web Services, 2017.

MCKENDRICK, J. **The Data Scene in 2017: More Cloud, Greater Governance, Higher Performance**, 2017.

MOHER, D. et al. The PRISMA Group. **Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement.** PLoS Med, 2009.

MYSQL: banco de dados. Disponível em <www.mysql.com>. Acesso em: 17 de jun. 2018.

NADAL, S. et al. **A software reference architecture for semantic-aware Big Data systems**. Universitat Politècnica de Catalunya, BarcelonaTech, Spain, 2017.

OKOLI, C. e SCHABRAM, K. **A Guide to Conducting a Systematic Literature**

Review of Information Systems Research. Sprouts: Working Papers on Information Systems, 2010.

RIAK: banco de dados. Disponível em <http://basho.com/products/>. Acesso em: 17 de jun. 2018.

TABLEAU: ferramenta para análise e visualização de dados. Disponível em <www.tableau.com>. Acesso em: 17 de jun. 2018.

TRIDENT: abstração de alto nível para computação em tempo real. Disponível em <http://storm.apache.org/releases/1.0.6/Trident-tutorial.html>. Acesso em: 17 de jun. 2018.

VISAKH, N. **Aligning Machine Learning for the Lambda Architecture (AMLLA)**. Aalto University, School of Science, 2015.

WAHYUDI, A.; JANSSEN, M. **A Process Pattern Model for Tackling and Improving Big Data Quality,** 2018.

Contatos: ysabellesousa@gmail.com e orlandoc@mackenzie.br