XXI ENGEMA
Encontro Internacional sobre Gestão Empresarial e Meio Ambiente

ISSN: 2359-1048
Dezembro 2019

Proposta de estrutura computacional decentralizada, de fácil uso, para aplicação em seleção de dados ambientais dispersos e não estruturados

LEANDRO COLEVATI DOS SANTOS
ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

ELIPHAS WAGNER SIMÕES
ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

SEBASTIAO GOMES DOS SANTOS FILHO

MARIA LÚCIA PEREIRA DASILVA
ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

TÍTULO: Proposta de estrutura computacional decentralizada, de fácil uso, para aplicação em seleção de dados ambientais dispersos e não estruturados

1 INTRODUÇÃO

Uma consequência imediata da evolução dos sistemas eletrônicos, e em especial da microeletrônica, foi a proliferação de instrumentos pequenos, de baixo custo e, mais recentemente, de fácil interligação com ferramentas de controle e análise de dados, além, obviamente, da possibilidade de acessar e inserir dados em estrutura de nuvem (CoT – *Cloud of Things*). Um exemplo emblemático são os sensores, produzidos nas mais diferentes formas e para as mais inesperadas utilizações, como exemplo, apenas no ano de 2019, já foram revisados os usos de sensores em questões tão díspares quanto usinagem, vibrações e o controle das forças aplicadas nesse processo (POSTEL et al, 2019) e a inspeção de infraestrutura férrea (FALAMARZI et al, 2019), além disso, foi também apresentado um livro sobre sensores flexíveis (NAG, MUKHOPADHYAY e KOSEL, 2019) que, já na introdução, alerta para a importância de tais sensores na área de saúde, ambiental e aplicações industriais. O custo destes sensores, devido as mudanças na produção, tais como manufatura aditiva e processos mais limpos, como corte a laser, vem caindo consistentemente. Em última análise, os sensores são suporte nas decisões importantes, como por exemplo, manutenção preventiva.

Por outro lado, a facilidade de aquisição e uso de tais instrumentos traz como consequência direta uma gigantesca massa de dados para analisar, o que gerou, entre outras coisas, várias ferramentas de seleção e análise de dados. Desse modo, termos como *data mining* e *big data* tornaram-se estranhamente familiares nas últimas décadas e a Tabela 1 apresenta o número de aparições de tais termos no Google Acadêmico®. Para comparação, outros modos de selecionar e manipular tais dados são também apresentados.

Outra tendência importante no mundo tecnológico em que vivemos é a necessidade de mobilidade e, portanto, torna-se necessário poder manipular e/ou apresentar tais dados em dispositivos móveis, como os *tablets* e *smartphones*. Assim, quase uma década atrás Preechaburana et al (2012) já preconizava a importância da estratégia de se utilizar os eletrônicos de consumo (*standard consumer electronic devices* - CEDs) como plataforma para obtenção de dados na área ambiental e de saúde, simplesmente pela adição de novas plataformas de sensoriamento a "*scanners*, DVD, RFIDs, câmeras compactas e, principalmente, telefones celulares, que apresentam uma grande gama de funcionalidades, estão em constante evolução e são adaptáveis a medidas na área química". Com a adaptação de um celular os autores foram capazes de determinar a concentração de etanol em água, enzima na urina, etc. Sutherland et al (2019), na revisão das metas globais de conservação avalia que "10 anos antes, o uso de celulares para avaliação do ambiente era apenas uma possibilidade, mas que a situação mudou rapidamente e que atualmente estes são extensivamente utilizados", muito embora ainda haja muito por fazer, como a Tabela 1 indica, apresentando apenas 280 aparições para "*Mobile-Sensing Technology*". Uma das limitações para o uso dessa ferramenta, então, fica evidente quando se relaciona "*Mobile-Sensing Technology*" e "*Computation Offloading*", com apenas quatro resultados, todos posteriores a 2016. Isso é decorrente da limitação do celular em processar dados, o que enseja o uso de computação em nuvem. Por outro lado, torna-se necessária a interação do dispositivo e os dados analisados em outro ambiente, decorre daí a importância do desenvolvimento de soluções de softwares baratas, rápidas e na forma de aplicativos, ou seja, tanto quanto possível de acordo com as práticas previstas sobre interação humano computador.

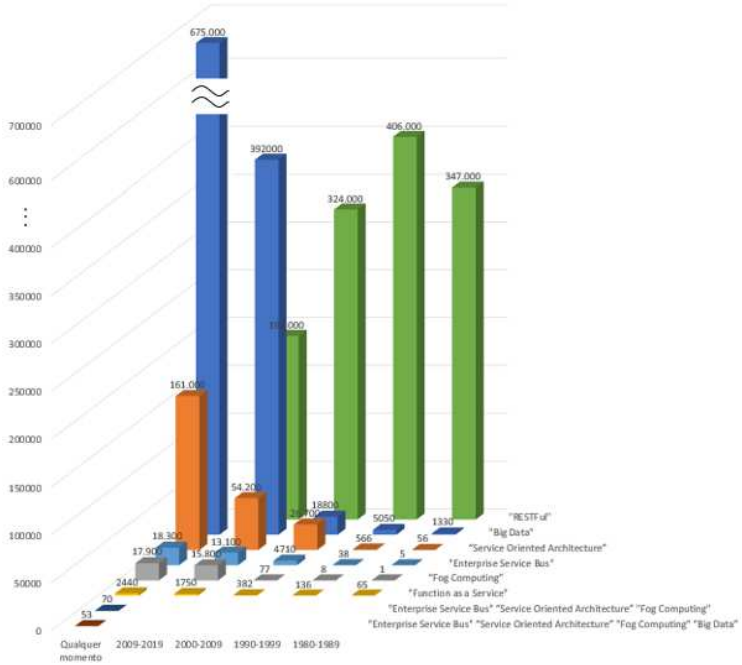
Tabela 1 - Número de aparições, a qualquer momento, no Google Acadêmico®, de termos relacionados à procura e análise de dados

Termos	# de aparições
RESTFul	2.240.000
Big Data	675.000
“Service Oriented Architecture”	161.000
"Enterprise Service Bus"	18.300
Fog Computing	17.900
Function as a Service	2440
"Mobile-Sensing Technology"	280
"Enterprise Service Bus" “Service Oriented Architecture” "Fog Computing"	70
"Enterprise Service Bus" “Service Oriented Architecture” "Fog Computing" “Big Data”	53
"Mobile-Sensing Technology" "computation offloading"	4

Fonte: dos Autores (2019)

O desenvolvimento para tais softwares implica no uso de novas tecnologias e a Figura 1 apresenta a maioria destas tecnologias e sua curva de desenvolvimento, estimada através do número de citações no Google Acadêmico®. Na Figura 1 é possível observar que todos os termos apresentam comportamento similar em função do tempo, à exceção de RESTful, uma tecnologia já consolidada ao início do milênio. Para as outras, há grande aumento no interesse apenas na última década, o que é consistente com o também desenvolvimento de aplicativos para celulares. Salienta-se a opção por não apresentar na figura a coluna referente a RESTFul qualquer momento por sua distinção dos outros dados em termos de variação de eixo Y no gráfico.

Figura 1 - Curva de desenvolvimento das tecnologias para produção de softwares, estimada através do número de citações no Google Acadêmico®



Fonte: dos Autores (2019)

Essa pesquisa se justifica não só pela baixa quantidade de sistemas computacionais apresentados como também a dispersão dos meios e modos dos atuais sistemas. O Quadro 1 apresenta, resumidamente alguns desses sistemas e suas limitações.

Quadro 1 - Trabalhos correlatos

Referência	Comparação
Towards a Platform for Prototyping IoT Health Monitoring Services (ZAMFIR, 2016)	Propõe sistema computacional para comunicação e armazenamento de dados de sensores mas não inclui busca de dados já consolidados em bases
A Resource Service Model in the Industrial IoT System Based on Transparent Computing (LI, 2018)	Propõe sistema computacional para análise de sensores que compõe sistema IoT, mas não inclui busca de dados já consolidados em bases
Adaptive mobile Web server framework for Mist computing in the Internet of Things (LIYANAGE, 2018)	Implementação de *Platform as a Service* para dispositivos móveis para melhor desempenho, proposta um pouco distinta da apresentada nesse artigo que foca em arquitetura de *Fog Computing* para ganho de desempenho e utilização de *Function as a Service*
Mobile Cloud Business Process Management System for the Internet of Things: A Survey (CHANG, 2017)	Faz uma análise de sistemas já propostos na literatura, baseado em modelo BPMN, proposta distinta da apresentada nesse artigo que propõe o desenvolvimento de estrutura computacional

Fonte: (ZAMFIR et al., 2016) (LI et al., 2018) (LIYANAGE et al., 2018) (CHANG et al., 2017)

2 PROBLEMA DE PESQUISA E OBJETIVO

A integração de tecnologias, o aumento das capacidades computacionais de processamento e armazenamento tem propiciado uma grande geração de dados, não estruturados e de armazenamento totalmente disperso. Môro (2018) apresenta que, em 2018, 90% dos dados gerados por toda a humanidade foram gerados nos 2 anos anteriores, complementando que a expectativa é a geração de 1,7 MB(Megabytes) por pessoa por segundo até 2020.

Como os dados são gerados por diversos equipamentos e/ou aplicações, eles são agrupados em arquivos dos mais diversos formatos, bancos de dados com diversas configurações e, as vezes disponibilizados em portais de dificil acesso, o que demanda muito tempo e esforço para criar uma base consistente para fazer análises, gerar conhecimento e tomar decisões.

Esse artigo pretende apresentar uma possível maneira para, utilizando tecnologias que extraiam o máximo dos recursos, considerando baixo custo financeiro e computacional, criar um sistema que permita buscar, decodificar e persistir dados afins, que estejam espalhados em em formatos variados, inclusive podendo ser transmitidos por microssensores, como os contidos em *smartphones*. O sistema deve também permitir que esses dados possam ser consultados com baixo consumo de recurso e com baixa latência.

O objetivo é criar uma estrutura computacional em versão *alpha* contendo, para fins de testes, partes críticas que viabilizem a análise da solução proposta. A partir dessa versão, proceder uma prova de conceito, com condições de contorno definidas para validar a solução.

3 FUNDAMENTAÇÃO TEÓRICA

Este item apresenta os principais aspectos teóricos e definições utilizados para o desenvolvimento do sistema aqui proposto, tais como Arquitetura Orientada a Serviços, Computação em Nuvem e suas derivações e Barramento de Serviços Corporativos.

A escolha pela tecnologia de Arquitetura Orientada a Serviços se deu em função da busca pela eficiência e a possibilidade de escalabilidade de funções específicas, pois como explica Villamizar (2015), aplicações monolíticas são dificeis de escalonar, uma vez que apenas algumas de suas partes são mais exigidas que outras e, nessa abordagem, toda a aplicação deverá ser redimensionado, consumindo recursos mesmo quando não são utilizados. Os serviços são criados em tecnologia REST por ser menos tipado e independente de linguagem, a distribuição dos serviços em formato de *FaaS* em uma estrutura de barramentos de serviços corporativos com *Fog Computing*, escolhidos para criar uma independência de grandes servidores.

3.1 Arquitetura Orientada a Serviços (SOA)

A Arquitetura Orientada a Serviços, muitas vezes representada pela sigla internacional SOA (*Service Oriented Architecture*), trata de um modelo cujas unidades lógicas são divididas em pequenas unidades, com funções definidas, independentes, que serão a composição de um sistema maior. Considerando a melhoria do processo de desenvolvimento de software, esse modelo auxilia o reuso de software, uma vez que, ao invés de reescrever o mesmo modelo de negócio diversas vezes, cada unidade (serviço) pode ser, apenas, referenciada quantas vezes forem necessárias (MOHAMMADI e MUKHTAR, 2011).

São características dos serviços na arquitetura SOA, o baixo peso computacional de processamento, o fraco acoplamento e a possibilidade de intercomunicação por mensagens. Essas características tornam os serviços flexíveis e escaláveis, ou seja, podem ser modificados os substituídos para se adaptarem a novas necessidades. As apresentações mais comuns da arquitetura são em formato de serviços baseados na *web*, denominados *Web Services*, sendo REST e SOAP as tecnologias mais utilizadas (LI e SVÄRD, 2010).

3.1.1 Protocolo Simples de Acesso a Objetos (SOAP)

Referenciada pela sigla SOAP (*Simple Object Access Protocol*), é uma estrutura composta por 3 elementos, provedor de serviços, consumidor de serviços e o registro de serviços. O provedor de serviços, com a unidade lógica e sua referência em uma rede que permite a entrada de requisições e um eventual retorno de dados. O consumidor de serviços é uma aplicação que executa diversas funções, sendo que uma delas é enviar requisição, com ou sem dados, para um serviço e receber seu retorno. O registro de serviços é um modelo lógico que contém o endereçamento dos serviços disponíveis (MUMBAIKAR et al, 2013).

Utilizando um protocolo como o Protocolo de Transporte de Hipertexto (HTTP), a aplicação consumidora faz uma solicitação ao serviço utilizando um documento XML como mensagem, numa estrutura que ainda contém um cabeçalho, com as informações de envio (MUMBAIKAR et al, 2013). O modelo de mensagens utilizados nessa estrutura é considerado pesado para transporte e, para aplicações com grandes necessidades de trocas de dados ou mudanças de redes, como aplicações mobile, não é considerada uma estrutura adequada, sendo, comumente, substituída pela arquitetura RESTFul (WAGH e THOOL, 2012), ainda assim, Muehlen et al. (2005) explica que SOAP e RESTFul não são arquiteturas opostas, mas com finalidades distintas, pois RESTFul é baseado em um estilo estrutural e SOAP pode ser utilizado como elemento em múltiplas arquiteturas.

3.1.2 Transferência de Estado Representacional (REST)

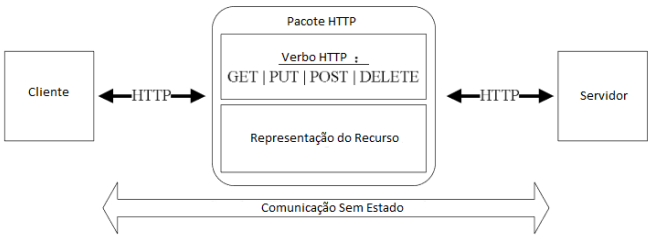
Diferentemente da estrutura do modelo SOAP, o modelo baseado na arquitetura REST é muito menos tipado e considera verbos do protocolo HTTP para fazer a comunicação entre aplicação consumidora e serviços, ao invés de mensagens em documentos XML (MUMBAIKAR et al, 2013).

Mumbaikar et al. (2013) explica que esta estrutura se baseia em verbos e nomes, na mesma linha, Muehlen et al. (2005) demonstra que existem mais objetos, representados por identificadores, denominados URI e poucos métodos, que são os verbos descritos no protocolo HTTP, como GET, POST, PUT e DELETE, dentre outros.

A comunicação entre serviço e consumo, conforme

Figura ***2***, convenciona-se uma operação GET para buscar uma sequência de dados em algum formato, POST transfere algum formato de objeto para um recurso, PUT cria ou atualiza um recurso e DELETE remove um recurso existente (FENG et al, 2009).

Figura 2 - Representação da Arquitetura HTTP



Fonte: Adaptado de (FENG et al, 2009)

3.2 Serviços de Computação

Dentre os modelos de computação, a computação em nuvem é a que tem o maior destaque, trazendo consigo os modelos de serviço, como Software como Serviço, Plataforma como Serviço, dentre outros. A computação em névoa é um modelo semelhante à computação em nuvem, mas em escopo reduzido e voltado para aplicações que necessitem de baixa latência.

3.2.1 Computação em Nuvem (*Cloud Computing*)

Modelo de computação onde se provê ubiquidade, acesso a rede em tempo real a um número, não obrigatoriamente contabilizado de recursos, que podem ser físicos ou lógicos, como serviços, servidores, unidades de armazenamento ou aplicativos. A estrutura de computação em nuvem deve ser estruturada de forma que seus recursos estejam disponibilizados sempre que houver demanda, sem a necessidade de comunicação com qualquer provedor de serviços (MELL, 2011). Propondo uma definição mais heurística, Wang (2010), explica que a computação em nuvem é um "conjunto de serviços em rede, com escalabilidade, garantia de qualidade e serviço, normalmente personalizada, com estrutura computacional de baixo custo e sob demanda, que pode ser acessado de maneira simples e pervasiva".

Mell (2011), apresenta as cinco características essenciais, detalhados no Quadro 2, de um modelo de computação em nuvem, conforme estabelecido pelo Instituto Nacional de

Padronização e Tecnologia (NIST) dos EUA, sendo elas o autosserviço sob demanda, amplo acesso à rede, agrupamento de recursos, rápida elasticidade e serviços mensuráveis.

Quadro 2 - Características essenciais da computação em nuvem

Característica	Detalhamento
Autosserviço sob demanda	Qualquer unidade consumidora poderá solicitar qualquer recurso de hardware ou de software sem a necessidade de interação humana
Amplo acesso à rede	Qualquer recurso deve estar disponível pela rede podendo ser acessado por qualquer porte de estrutura computacional
Agrupamento de recursos	Os recursos disponíveis devem estar agrupados para atender demandas consumidoras distintas, de hardware e software sob demanda. Não há a necessidade de saber, fisicamente, onde o recurso está
Rápida elasticidade	O aumento ou a redução de recursos solicitados pelo consumidor deve ser atendido o mais rápido possível e automaticamente
Serviços mensuráveis	O controle e o monitoramento dos recursos deve ser feito automaticamente pelos serviços de nuvem

Fonte: (MELL, 2011)

Conforme apresentado por Mell (2011), Wang (2008) e Spillner (2017) e dispostos no quadro , são modelos de nuvem *Hardware as a Service (HaaS), Software as a Service (Saas), Platform as a Service (PaaS), Infastructure as a Service (IaaS), Function as a Service (FaaS)*, dentre outros.

Quadro 3 - Modelos de aplicações em nuvem

Modelo	Definição
Hardware as a Service (HaaS)	Utilizando conceitos de Virtualização de Hardware, métricas de uso de recursos de TI, pode-se comprar o uso de recursos de hardware no modelo *pay-as-you-go*, ou seja, paga-se pelo uso da capacidade computacional contratada (WANG, 2008).
Software as a Service (SaaS)	Aplicações são disponibilizadas inteiramente para uso na internet. Paga-se pelo acesso integral ou parcial do software, de alguns softwares dentro de um conjunto ou de um conjunto inteiro de softwares (WANG, 2008).
Platform as a Service (PaaS)	Plataformas voltadas a desenvolvedores, com linguagens de programação, sistemas gerenciadores de bancos de dados, bibliotecas, disponibilizados em um conjunto de hardware pré-definido e contratado, que dispensa a criação de servidores físicos para softwares desenvolvidos para aplicações específicas (MELL, 2011).
Infrastructure as a Service (IaaS)	Conjuntos completos de hardwares, podendo definir, pelo contratante, quantidade de CPU, Memória RAM, unidade de armazenamento e até estruturas de rede, onde o contratante tem, a infraestrutura a sua disposição para configurar conforme sua necessidade (MELL, 2011).
Function as a Service (FaaS)	Estrutura sem estado, ou seja, tende a desaparecer após a transação, é a conversão das pequenas funcionalidades de uma aplicação em funções, disponíveis como serviços, em unidades de nuvem, independente de servidor, para aplicações de baixo tempo de execução (SPILLNER, 2017).

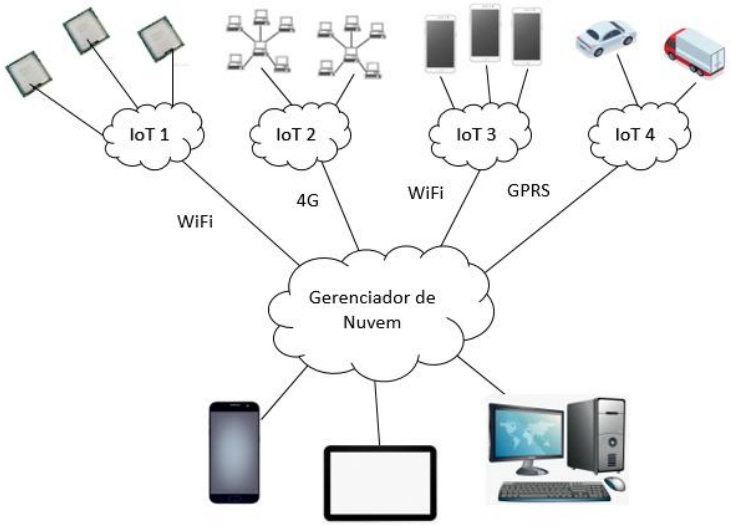
Fonte: (WANG, 2008) (MELL, 2011) (SPILLNER, 2017)

3.2.2 Nuvem das Coisas (*Cloud of Things*)

A internet das coisas (IoT), como explicado por Atzori et al. (2010), é a presença, difundida por todos os meios, de coisas, que são quaisquer elementos (pessoas ou equipamentos), com uma identificação, interconectados a partir de uma rede.

O conceito supracitado traz uma grande versatilidade, no entanto, tem como limitações, a capacidade de processamento e problemas de heterogeneidade que, conforme demonstrado por Son e Lee (2018) podem ser mitigados utilizando o conceito de Nuvem das Coisas, internacionalmente conhecido por *Cloud of Things*, combinando elementos IoT virtualizados em uma estrutura de nuvem, , que provê a capacidade de processamento. A grande capacidade de processamento e armazenamento da estrutura de nuvem, conforme a Figura 3 , combinado com as coisas, podem, ao longo do tempo, solucionar diversos problemas, sendo que, o primeiro deles é conceber uma estrutura cenntralizada de recebimento e distribuição de dados (RUI e DANPENG, 2015).

Figura 3 - Nuvem das Coisas (*Cloud of Things*)



Fonte: Adaptado de (AAZAM, 2014)

3.2.3 Computação em Névoa (*Fog Computing*)

Definição proposta pela CISCO em 2012 que visa subdividir parte da estrutura de nuvem, com parte do processamento ou do armazenamento colocada mais próxima dos nós (coisas, como elucidado no item 3.2.2), que são o topo da estrutura de névoa (MEBREK, 2017). Também denominado, em artigos, como *Edge Computing, Cloudlets, Multi-access Edge Computing ou Mobile Edge Computing*, é considerado como uma movimentação da computação, que se faz em ambiente de nuvem, para a borda da estrutura (GARCIA et al, 2018).

Uma das aplicações da computação em núvem é o *Computation Offloading*, definido por Chamas (2018), como a transferência de processamento de um dispositivo, majoritariamente móvel, para outro ambiente de processamento, por baixa capacidade de

recursos (processamento, energia, armazenamento, etc.), no entanto, Ye et al. (2016) explica que esse processamento gera tráfego, nas redes de computadores, de objetos muito pesados, que acabam aumentado a latência nas operações, sendo, portanto, a computação em névoa, uma solução para operações que precisem de baixa latência pois, por estar próximo dos nós, os dados trafegam por uma distância menor ou podem ser fragmentados.

3.3 Barramento de Serviços Corporativos (ESB)

Combina a Arquitetura Orientada a Serviços com uma abordagem dirigida a eventos visando criar uma possibilidade de comunicação direta, e quando necessário fazer as traduções, entre serviços heterogêneos, uma vez que os serviços incluidos no barramento podem ser acessados por aplicações consumidoras ou por outros serviços (MARÉCHAUX, 2006).

Além da comunicação, por ser implementado, majoritariamente, em *Web Services* (REST ou SOAP), os barramentos devem descrever os destinos de maneira neutra, apenas valendo-se dos identificadores. O gerenciamento dos serviços e a segurança de acesso deverão ser prioridades na implementação dos ESB pois estes se comunicarão também com aplicações críticas.

4 MÉTODOS

Este trabalho, por sua natureza, trata-se de pesquisa aplicada. Quanto aos objetivos, considerando-se a necessidade da definição das condições de contorno e, preliminarmente, no intuito de verificar a funcionalidade do software, a existência da Prova de conceito, é, também, uma pesquisa exploratória e com abordagem experimental. Como é interesse do trabalho determinar se determinadas premissas são adequadas, mas, igualmente, determinar a performance do software sob desenvolvimento, trata-se de pesquisa combinada – qualitativa e quantitativa.

Quanto à prova de conceito, sua utilização neste trabalho é, como definido por Tonella (2007 apud ALVES, 2011), "a descrição do funcionamento do aplicativo em um determinado exemplo, que pode ser real ou não. O objetivo, portanto, não é realizar um experimento formal, mas sim demonstrar como a utilização do aplicativo contribui para o conhecimento. ". Dentro deste contexto, a massa de dados a ser analisada não precisa necessariamente ter origem ambiental. Assim, optou-se pela análise de informações de bases de dados médicas, por inúmeros fatores, entre eles, o fato da informação estar distribuída, ou seja, não estruturada, corresponder a um massivo volume de dados e, via de regra, depender diretamente ou indiretamente da existência de sensores distribuídos aleatoriamente (HERSH, 2011). Para realizar a prova de conceito, se fez uma extração de dados públicos de algumas fontes médicas, em formatos diversos. Procedeu-se a uma extração do universo de dados, com, aproximadamente 30 milhões de registros e cada registro, na média com 25 dados.

Quanto às condições de contorno, como será evidenciado nos Resultados, decorrem do tipo de dado que se quer analisar, como descrito na prova de conceito.

Quanto aos métodos, utilizaram-se as etapas de desenvolvimento, como proposto na para a Engenharia de Software. Pichi Junior (2011) revisou as principais etapas necessárias ao desenvolvimento de protótipos de produto inovador, dependente da produção de hardware o software. O Quadro 4 resume as principais questões metodológicas, como proposto pelo autor.

Quanto aos requisitos do projeto, a Introdução deste trabalho apresentou as principais informações requeridas. Quanto à Engenharia, a maioria das definições necessárias já foi apresentada na fundamentação teórica e nos Resultados encontram-se as etapas de simulação

inicial e testes. A avaliação final, com possíveis usuários ou especialistas da área, não foi efetuada neste trabalho.

Quadro 4 - Principais questões metodológicas para desenvolvimento de software

Quanto aos requisitos do projeto:	Etapas: **1) Análises em produtos similares encontrados no Mercado;** **2) Análise em relação ao uso;** **3) Análise funcional;** **4) Resultados das análises;** **5) Estruturação e Descrição do conceito**
Quanto à engenharia	Etapas: **1) Engenharia do produto,** com uma linguagem de representação**;** **2) Engenharia de processo – com a simulação** do processo, e sua respectiva otimização**;** **3) Gerenciamento de projeto –** monitoração das etapas
Avaliação de interface – não abordada neste trabalho	

Fonte: (PICHI JUNIOR, 2011)

A escolha do universo de dados para a prova de conceito foi por dados na área médica, pois se trata de uma área que gera uma quantidade muito grande de dados, disponíveis em diversos sistemas, tais como: CNES, SIASUS, SIHSUS, CIHASUS, ANS, ANVISA, DECEX entre mais 50 bases de dados, no entanto, pouco desta informação gera conhecimento para o trabalho cotidiano dos profissionais. Também, ferramentas avançadas de busca permitem que se monte um quadro da medicina baseada na ideia de arquétipos e taxonomia ontológica de todas as enfermidades, procedimentos, rotinas e técnicas disponíveis em qualquer nível de atenção da saúde, mas esta informação fica restrita ao extrato acadêmico.

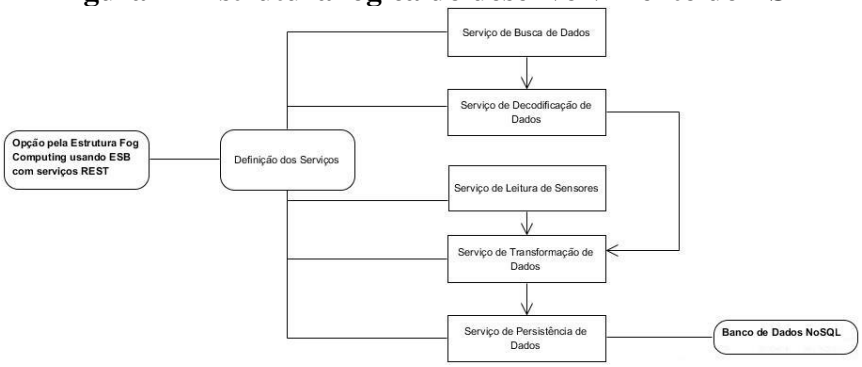
Para realizar a prova de conceito, se fez uma extração de dados públicos de algumas das fontes supracitadas, em formatos diversos. Se optou por não fazer leitura de sensores nesse momento. Essas fontes de dados são uma extração do universo de dados, com, aproximadamente 30 milhões de registros e cada registro, na média com 25 dados.

5 RESULTADOS E DISCUSSÕES

Definida a Arquitetura Orientada a Serviços em um Barramento de Serviços Corporativos, que, por se colocar mais próximo da borda da rede, será o *Fog Computing*, o primeiro passo foi definir a estrutura do sistema e seu desenvolvimento.

Conforme demonstrado na Figura 4, se modelou quais tipos de serviços deveriam ser criados, qual integração entre eles e as possíveis aplicações que consumiriam os serviços (*FaaS*), e, como seriam persistidos os dados.

Figura 4 - Estrutura lógica do desenvolvimento do ESB



Fonte: dos Autores (2019)

Os serviços escolhidos foram os de Busca de dados, Decodificação de Dados, Transformação de Dados, Recebimento de dados de sensores e Persistência de Dados. O serviço de busca de dados se faz obrigatório, uma vez que o propósito inicial é filtrar dados dispersos de um mesmo universo. Serviços de decodificação e transformação de dados foram definidos pois os dados se disponibilizam em formatos distintos e, para melhor desempenho computacional, uma estrutura de banco de dados único foi definido, portanto, serviço de persistência de dados, que recebem dados transformados e os persistem também são necessários. Serviços de leitura de sensores se tornam interessantes para o objetivo inicial, pois, uma vez que o universo foi definido, massas de dados podem ser trazidos diretamente para a base de dados do sistema.

A escolha pela persistência de dados em NoSQL MongoDB se deu por base no estudo desenvolvido por Oliveira et al. (2018) que demonstra que os recursos de armazenamento são otimizados, fazendo muita diferença em massas muito grandes e a consulta tem melhor desempenho que em SQL. A dificuldade com a redução do desempenho na inserção de dados é contornada mantendo um serviço que fica lendo as fontes de dados e inserindo sem a necessidade de interação humana.

Conforme a estrutura computacional apresentada na Figura 4, se criou um barramento de serviços para o desenvolvimento de uma prova de conceitos. A Figura 4 demonstra a construção dos serviços baseados em seus parâmetros de entrada e saída. Por se tratar de serviços baseados no conceito de *FaaS*, a definição dos parâmetros é primordial para criar sua funcionalidade específica.

O serviço de busca de dados recebe um endereçamento lógico que contém arquivos, não tendo relevância a sua quantidade, identifica quais arquivos tem dados relevantes e retorna uma estrutura de dados no formato de lista com endereçamento absoluto de cada arquivo e seu formato.

O serviço de decodificação de dados recebe a lista gerada pelo serviço de busca, faz a leitura de cada arquivo, transforma os dados em formato de texto sem formatação e retorna essa estrutura de texto.

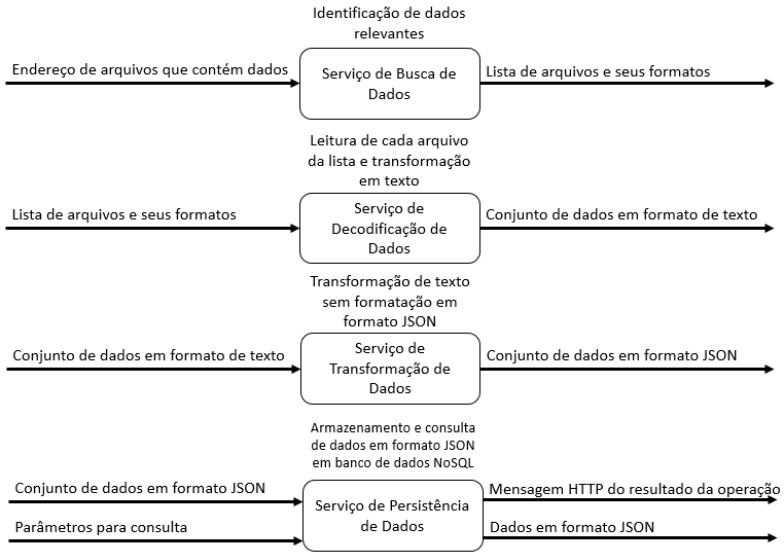
O serviço de transformação de dados recebe uma massa de texto não formatado e tenta convertê-lo em formato JSON, que é o formato esperado pelo banco de dados NoSQL e retorna um conjunto de documentos JSON.

O serviço de persistência de dados trabalha em duas possibilidades, podendo receber um ou diversos documentos em formato JSON e persistir os dados no banco de dados NoSQL, retornando apenas a mensagem com código HTTP que determina se a operação foi feita com sucesso ou não, ou receber parâmetros de consulta para retornar conjuntos de dados para análise ou formação de arquétipos.

Para os testes do sistema, foi feita uma extração de arquivos públicos, com dados da área de saúde, de um período de tempo de 12 anos inteiros e parte de 2019. Os arquivos estavam em diversos formatos e estruturados cada um de acordo com o período de tempo que foi consolidado.

A arquitetura do sistema se apresentou semelhante ao exposto por Stallings (2010) quando define o ciclo de uma instrução, ou seja, partes do ciclo são bem definidas e a entrada de uma função é a saída da outra. Ainda nesse assunto, existe o conceito de *pipeline*, que descreve que cada parte do processamento do ciclo de uma instrução pode ser realizado por um circuito específico e sobrepor seu funcionamento, melhorando o seu desempenho. Assim sendo, os testes se deram sobrepondo as funcionalidades dos serviços, ou seja, o serviço de transformação não espera o serviço de busca finalizar todas as buscas possíveis, pois este já retorna alguns endereços com arquivos para serem transformados e assim por diante.

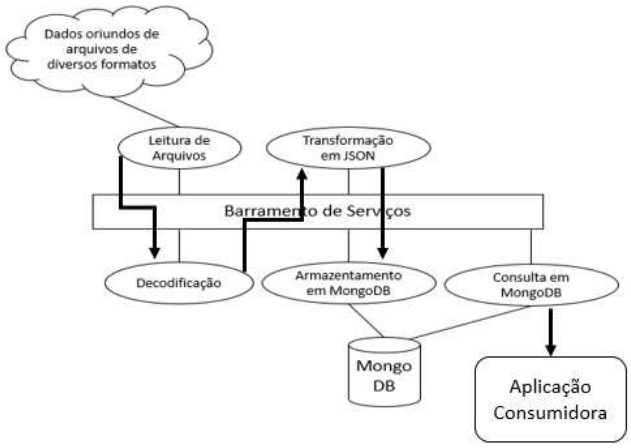
Figura 5 - Estrutura de desenvolvimento dos serviços por seus parâmetros de entrada e saída



Fonte: dos Autores (2019)

A prova de conceito, conforme demonstrado na Figura 6, foi feita utilizando os dados dos arquivos associados aos serviços de extração, decodificação e transformação dos dados oriundos de arquivos, persistência dos dados e consulta. Nesta versão, a leitura de sensores não foi integrada ao barramento.

Figura 6 - Estrutura da prova de conceito



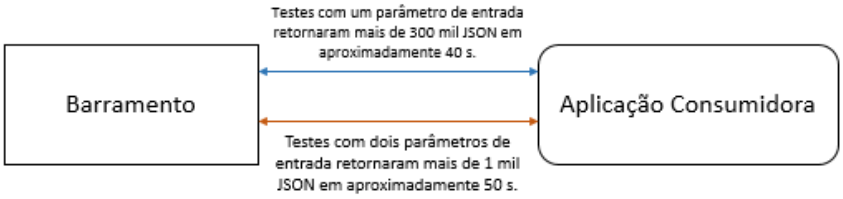
Fonte: dos Autores (2019)

Com todos os serviços rodando, o universo gerou uma base de dados superior a 30 milhões de documentos JSON com uma média de 25 atributos cada. Se fez uma aplicação para obter dados visando consultar arquivos e possibilitar a criação arquétipos, que, primeiramente, ressalte-se a praticidade de obter agrupamento de dados com 1 ou 2 linhas de código, a contrário do que se tentava fazer, com análise de arquivos, programação de planilhas.

Foi construída uma aplicação consumidora que se comunica com o barramento de serviços enviando valores como parâmetros de consulta e esperam conjuntos de JSON como

retorno, com os dados de interesse para as análises. A Figura 7 demonstra a comunicação e resultados de testes feitos com a aplicação consumidora, considerando que os outros serviços continuam em funcionamento.

Figura 7 - Comunicação de aplicação consumidora ao barramento



Fonte: dos Autores (2019)

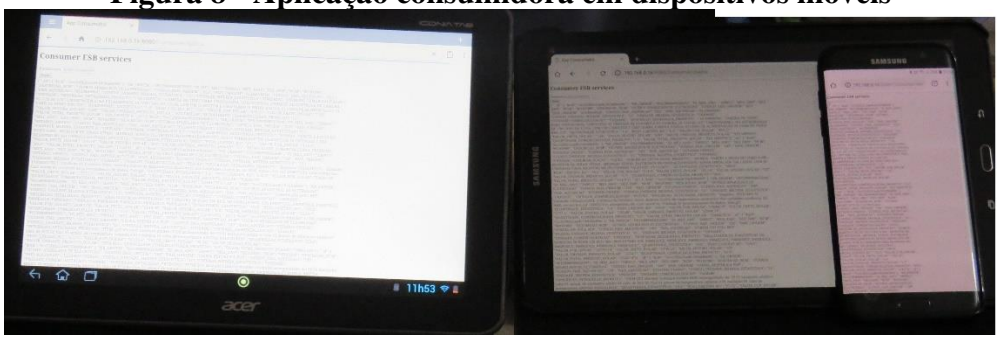
Nos testes realizados, a aplicação consumidora enviou um ou dois parâmetros de entrada de consulta e recebeu um conjunto de documentos JSON com os dados filtrados pelo serviço de persistência de dados. Os dados já são enviados com uma formatação que facilita a interpretação humana, em arquivos, o que interfere no tempo de resposta quando comparado com o envio bruto dos dados.

Com um parâmetro de entrada, o retorno trouxe, aproximadamente, 300 mil documentos JSON, sempre com mais de 20 dados distintos por JSON. Esses testes trouxeram os dados formatados em aproximadamente 40 segundos.

A mesma execução foi feita passando dois parâmetros para o serviço de persistência e o retorno apresentou, aproximadamente 1 mil documentos JSON com as mesmas características da consulta com um parâmetro em tempo de execução de aproximadamente 50 segundos.

A Figura 8 apresenta testes feitos com aplicação consumidora em 3 dispositivos móveis, sendo dois *tablets* diferentes (um da marca Acer e outro da marca Samsung) e um smartphone Samsung Galaxy® S7, consultando os serviços de persistência de dados e os resultados. Cabe salientar que os valores supracitados se apresentam em qualquer execução.

Figura 8 - Aplicação consumidora em dispositivos móveis



Fonte: dos Autores (2019)

5 CONCLUSÃO

Como Problema de Pesquisa esse trabalho abordou a possibilidade de definir uma estrutura computacional que permita capturar dados decentralizados e desestruturados, dispostos em rede ou disponibilizáveis por sensores, fazer serviços para a análise, tratamento e transformação, centralizar em um banco de dados. A resposta encontrada utiliza Arquitetura Orientada a Serviços associada ao princípio de tecnologia de computação em nuvem, porém,

aproximando-se do *Fog Computing*, que se encontra na fronteira da tecnologia. Esta ferramenta tem a vantagem de se encontrar fisicamente mais próxima dos nós do que da nuvem, o que viabiliza a execução da tarefa em tempo muito menor, portanto, este trabalho mostrou ter um caráter inovador.

Para o desenvolvimento, como banco de dados foi utilizado o NoSQL, para possibilitar que aplicações consumidoras façam buscas e filtragens nesses dados com a finalidade de obter dados brutos ou criar arquétipos para a geração de conhecimento. Por concepção, todos os serviços rodam simultaneamente e, por mais que a tecnologia de nuvem escolhida permita que troquem dados entre si, a interoperabilidade não interrompe a funcionalidade de nenhum serviço.

Como prova de conceito do produto desenvolvido utilizou-se uma massa de dados de origem médica. Considerando-se que essa prova foi realizada com universo de registros da ordem de dezenas de milhões com diversos dados por registros, obtidos anualmente e dispostos em arquivos de diversos formatos, a busca e filtragem destes dados implicaria em trabalho analógico, o que pode resultar em diversos dias de espera para gerar informação consistente, especialmente pela necessidade de contínua rechecagem. Assim, este trabalho apresenta como os serviços podem ficar em tempo integral, capturando tais dados e disponibilizando-os para consultas. Por sua vez, tais consultas apresentam dados formatados e em estrutura que pode ser lida por qualquer computador, *smartphone*, etc., em menos de 1 minuto, o que é consistente com a expectativa de baixa latência.

Por ser tratar de leitura de dados brutos, considera-se que este sistema computacional está apto para apresentar resultados semelhantes se o universo de dados estiver na área de manufatura ou ambiental.

6 BIBLIOGRAFIA

AAZAM, M. et al. Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved. **Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST)**, Islamabad, p. 414-419, Janeiro 2014.

ALVES, L. P. S. **Um aplicativo baseado em inteligência coletiva para compartilhamento de rotas em redes sociais**. [S.l.]: Universidade Tecnológica Federal do Paraná, 2011.

ATZORI, L. et al. The internet of things: A survey. **Computer networks**, v. 54, n. 15, p. 2787-2805, 2010.

CHAMAS, C. L. **Consumo de energia em dispositivos móveis Android:** análise das estratégias de comunicação utilizadas em Computation Offloading. São Paulo: USP, 2018.

CHANG, C. et al. Mobile Cloud Business Process Management System for the Internet of Things: A Survey. **ACM Computing Surveys (CSUR)**, v. 49, n. 4, 2017.

FALAMARZI, A. et al. A Review on Existing Sensors and Devices for Inspecting Railway Infrastructure. **Jurnal Kejuruteraan**, v. 31, n. 1, p. 1-10, 2019.

FENG, X. et al. An alternative to RPC for Web services architecture. **2009 First International Conference on Future Information Networks**, p. 7-10, IEEE, 2009.

GARCIA, J. et al. Do We Really Need Cloud? Estimating the Fog Computing Capacities in the City of Barcelona. **IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)**, p. 290-295, 2018.

HERSH, W. et al. Health-care hit or miss. **Nature**, v. 470, n. 7334, p. 327-329, 2011.

LI, W. et al. A Resource Service Model in the Industrial IoT System Based on Transparent Computing. **Sensors**, 2018.

LI, W.; SVÄRD, P. REST-based SOA application in the cloud: a text correction service case study. **2010 6th World Congress on Services**, p. 84-90, IEEE, 2010.

LIYANAGE, M. et al. Adaptive mobile Web server framework for Mist computing in the Internet of Things. **International Journal of Pervasive Computing and Communications**, v. 14, n. 3/4, p. 247-267, 2018.

MARÉCHAUX, J.-L. Combining service-oriented architecture and event-driven architecture using an enterprise service bus. **IBM developer works**, p. 1269-1275, 2006.

MEBREK, A. et al. Efficient green solution for a balanced energy consumption and delay in the IoT-Fog-Cloud computing. **IEEE 16th International Symposium on Network Computing and Applications (NCA)**, p. 1-4, 2017.

MELL, P. et al. **The NIST definition of cloud computing**. [S.l.]: [s.n.], 2011.

MOHAMMADI, M.; MUKHTAR, M. SOA-based business process for Supply Chain Management. **2011 Malaysian Conference in Software Engineering**, p. 213-216, IEEE, 2011.

MÔRO, D. K. **Reconhecimento de Entidades Nomeadas em Documentos de Língua Portuguesa**. Araranguá: Universidade Federal de Santa Catarina, 2018.

MUEHLEN, M. Z. et al. Developing web services choreography standards—the case of REST vs. SOAP. **Decision Support Systems**, v. 40, n. 1, p. 9-29, 2005.

MUMBAIKAR, S. et al. Web services based on soap and rest principles. **International Journal of Scientific and Research Publications**, v. 3, n. 5, p. 1-4, 2013. ISSN 2250-3153.

NAG, A.; MUKHOPADHYAY, S. C.; KOSEL, J. **Printed Flexible Sensors:** Fabrication, Characterization and Implementation. [S.l.]: Springer, 2019.

OLIVEIRA, M. D. S. et al. Banco de Dados No-SQL X Banco De Dados SQL. **South American Development Society Journal**, São Paulo, v. 4, n. 11, p. 298-320, 2018.

PICHI JUNIOR, W. **Construção de Protótipo para Ensino na Área Tecnológica:** Cromatografia como Estudo de Caso. [S.l.]: Centro Paula Souza, 2011. 156 p.

POSTEL, M. et al. Monitoring of vibrations and cutting forces with spindle mounted vibration sensors. **CIRP Annals**, 2019.

PREECHABURANA, P. et al. Surface Plasmon Resonance Chemical Sensing on Cell Phones. **Angew. Chem. Int. Ed**, p. 11585-11588, 2012.

RUI, J.; DANPENG, S. Architecture design of the Internet of Things based on cloud computing. **Seventh International Conference on Measuring Technology and Mechatronics Automation**, p. 206-209, 2015.

SON, Y.-H.; LEE, K.-C. Cloud of things based on linked data. **2018 International Conference on Information Networking (ICOIN)**, p. 447-449, 2018.

SPILLNER, J. Snafu: Function-as-a-service (faas) runtime design and implementation. **arXiv preprint arXiv:1703.07562**, 2017.

STALLINGS, W. **Arquitetura e Organização de Computadores**. 8ª. ed. [S.l.]: Pearson, 2010.

SUTHERLAND, W. J. Ten Years On: A Review of the First Global Conservation Horizon Scan. **Trends in Ecology & Evolution**, v. 34, p. 139-153, 2019.

TONELLA, P. et al. Empirical studies in reverse engineering: state of the art and future trends. **Empirical Software Engineering**, Hingham, p. 551-571, 2007.

VILLAMIZAR, M. et al. Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud. **10th Computing Colombian Conference (10CCC)**, p. 583-590, 2015.

WAGH, K.; THOOL, R. A comparative study of soap vs rest web services provisioning techniques for mobile host. **Journal of Information Engineering and Applications**, v. 2, n. 5, p. 12-16, 2012. ISSN 2225-0506.

WANG, L. et al. Scientific cloud computing: Early definition and experience. **10th ieee international conference on high performance computing and communications**, p. 825-830, 2008.

WANG, L. et al. Cloud Computing: a Perspective Study. **New Generation Computing**, p. 137-146, 2010.

YE, D. et al. Scalable fog computing with service offloading in bus networks. **IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)**, p. 247-251, 2016.

ZAMFIR, M. et al. Towards a Platform for Prototyping IoT Health Monitoring Services. **Borangiu T., Dragoicea M., Nóvoa H. (eds) Exploring Services Science**, v. 247, 2016.