Submission date: 20/Jan/2017
Resubmission date: 19/Jun/2017
Camera ready submission: 02/Dec/2017
Section: prototype

1st round notification: 03/Apr/2017
2nd round notification: 05/Nov/2017
Available online: 21/Dec/2017
Publication date: 30/Jun/2018

Investigando a Mobilidade Urbana Através de Dados Abertos Governamentais Enriquecidos com Proveniência

Title: Using Government Open Government Data Enriched With Provenance To Investigate Urban Mobility

Sergio Manuel Serra da Cruz[1,2,3], Jonice de Oliveira Sampaio[3]

[1]Programa de Pós-Graduação em Modelagem Matemática e Computacional – UFRRJ
Seropédica, Rio de Janeiro – Brasil

[2]Programa de Educação Tutorial - PET-SI/UFRRJ
Seropédica, Rio de Janeiro – Brasil

[3]Programa de Pós-Graduação em Informática – UFRJ
Cidade Universitária, Rio de Janeiro – Brasil

serra@pet-si.ufrrj.br, jonice@dcc.ufrj.br

Abstract. *One of the main challenges of smart cities applications in emerging countries is the limited availability of tools that increase the interaction and collaboration between government and civil society. This paper discusses the advantages of enriching open government data about public transport with data provenance. We present a distributed architecture and its prototype named BusInRio. We also, present use cases and experiments based on real users using open data enriched with provenance. Our experiments indicate the degrees of acceptance and correction of the proposal.*

Keywords*. Smart Cities; Open Data; Data Provenance; ETL workflows; NoSQL; Semantic Web,*

Resumo. *Atualmente, os principais desafios para a consolidação das cidades inteligentes em países emergentes são a disponibilidade de ferramentas que aprofundem a colaboração entre o governo e a sociedade civil. Este artigo estuda os desafios da mobilidade urbana em cidades inteligentes e apresenta uma arquitetura distribuída e seu protótipo intitulado BusInRio. Diferentemente dos trabalhos relacionados, a proposta utiliza exclusivamente dados abertos governamentais enriquecidos por proveniência do tipo retrospectiva. Este artigo também avalia quantitativamente a proposta através de experimentos de campo baseados na análise de dados de proveniência oriundos das interações de usuários reais. As primeiras análises e resultados indicam os graus de correção e aceitação da proposta.*

Palavras-Chave. *Cidades Inteligentes; Dados Abertos; Proveniência de Dados; Workflows ETL; NoSQL; Web Semântica.*

Cite as: Cruz, S.M.S & Sampaio, J.O. (2018). Using Government Open Government Data Enriched With Provenance To Investigate Urban Mobility (Investigando a Mobilidade Urbana Através de Dados Abertos Governamentais Enriquecidos com Proveniência). iSys: Revista Brasileira de Sistemas de Informação (Brazilian Journal of Information Systems), 11(2), 91-123.

1. Introdução

No Brasil, o acesso à informação é tido como um direito garantido pela Constituição Federal de 1988. A transparência na gestão pública é essencial para aprimorar a administração e auxiliar o diálogo entre a sociedade e seus governantes. Ao lado das crescentes demandas por transparência somam-se as iniciativas ligadas aos "Dados Abertos Governamentais" formando uma injunção para assegurar a disponibilidade de informações para a efetiva participação do cidadão no controle social das ações do Estado (Eaves, 2009; PBDA, 2017; Zuiderwijk *et al.,* 2014a).

Dados Abertos são dados que podem ser utilizados, modificados e compartilhados por qualquer pessoa e para qualquer finalidade, estando sujeito as exigências que preservem sua proveniência, ou seja, sua publicação deve ser em formato aberto e estar sob uma licença aberta (PBDA, 2017 e ODaF, 2017). Os Dados Abertos Governamentais (DAG) são dados produzidos pelos governos e colocados à disposição da sociedade civil, estes podem ser acessados, reutilizados em novos projetos, publicados em sítios e utilizados em aplicativos sociais.

Atualmente, os DAG são utilizados em diversas iniciativas correlacionadas com o conceito de "Cidade Inteligente" (Giffinger, Haindlmaier, 2010; Dohler *et al.,* 2011, Schaffers *et al.,* 2011 e Caragliu *et al.,* 2011). O conceito ainda não possui uma definição clara, ele é multifacetado e está profundamente relacionado com economia do conhecimento (Hollands, 2008 e Caragliu *et al.,* 2011). Segundo Nan e Pardo (2011), o conceito tem despontado como a combinação de ambientes tecnológicos com as competências de capital social além de fatores político-institucionais. O conceito possui diferentes conotações que podem variar segundo as perspectivas do usuário, da tecnologia, do autor ou do modelo de negócio ao qual se aplica. No entanto, o principal fundamento nele contido é o uso intensivo das Tecnologias da Informação e Comunicação (TIC), visando aperfeiçoar o desempenho dos serviços urbanos suportando o desenvolvimento social e econômico.

Santana *et al.* (2016) estudaram iniciativas europeias, asiáticas e americanas de cidades inteligentes comparando-as com iniciativas brasileiras, os autores constataram que aqui persistem ações isoladas e de amplitude limitada. Os autores também constataram que algumas prefeituras de cidades brasileiras tentam reduzir essas disparidades. Por exemplo, a prefeitura da Cidade Rio de Janeiro estimula a utilização de DAG em aplicativos de cidades inteligentes. Ela promove concursos de criatividade (RioApps[1] e *hackathons*) para estimular o desenvolvimento de novas ferramentas que contribuirão para empoderar o cidadão carioca. Adicionalmente, a prefeitura mantém, através do portal *Data.Rio*[2]*,* repositórios de DAG sobre diversos assuntos tais como: transporte, meteorologia, educação, saúde, impostos, administração pública, entre outros.

Terán *et al.* (2016), Santana *et al.* (2016) e Batista *et al.* (2016) realizaram uma ampla revisão da literatura relatando os principais desafios do desenvolvimento de aplicativos móveis voltados para cidades inteligentes em países emergentes. A partir desses trabalhos, verificamos que existem diversas oportunidades de pesquisas na área

[1] http://portalrioapps.com.br/
[2] http://data.rio/

de qualidade de dados abertos em cidades inteligentes, em especial, há uma carência de estudos que explorem questões relativas a mobilidade urbana utilizando DAG enriquecidos com proveniência. A proveniência fornece suporte para registrar a origem, a qualidade e a autoria dos dados, auxiliando na capacidade de validar resultados de processos comerciais ou experimentos científicos (Freire *et al.*, 2008).

O gerenciamento de dados de proveniência em dados abertos tem sido amplamente discutido na comunidade científica (Simmhan *et al.*, 2005, Freire *et al.*, 2008, Hartig, 2009, Cruz et al., 2009, Gil *et al.* 2010). Todavia, segundo Corsar e Edwards (2012), Emaldi *et al.* (2013) e McArdle e Kitchin (2015) apesar de ser considerada como promissora e de já existirem muitos trabalhos voltados para a publicação e recuperação de dados abertos, ainda há carência de estudos sobre proveniência de dados em aplicações móveis para cidades inteligentes.

Este artigo tem como objetivo contribuir com a oferta de novas abordagens relacionadas aos desafios da mobilidade urbana nas grandes cidades brasileiras. Apresentamos versão ampliada da arquitetura denominada *BusInRio,* inicialmente proposta por Cruz, Andrade e Oliveira (2016). Além disso, apresentamos seu protótipo e sua avaliação experimental. Diferentemente dos trabalhos relacionados, a solução utiliza exclusivamente DAG enriquecidos com proveniência, compatível com a especificação PROV da W3C (Moreau *et al.,* 2013) sobre as viagens dos ônibus na Cidade do Rio de Janeiro. A proposta permite que os usuários utilizem dados curados para escolher a linha de ônibus em função das condições de trânsito no trajeto desejado.

Os DAG são coletados e transformados utilizando-se *workflows* ETL e posteriormente armazenados em bancos de dados não relacionais (neste artigo, eles são denominados como *NoSQL*) interligados que são consumidos por aplicações móveis. Este artigo também apresenta duas avaliações experimentais dos rastros de proveniência. A primeira avaliação utiliza informações fornecidos pelos usuários, a segunda utiliza dados de proveniência retrospectiva coletados pela própria arquitetura.

Este artigo está organizado da seguinte forma: A seção 2 discute os principais referenciais teóricos, a seção 3 apresenta a metodologia adotada. A seção 4 apresenta os principais módulos da arquitetura *BusInRio.* A seção 5 apresenta a prova de conceito e experimentos e avaliações realizadas com os usuários e discute os resultados principais. A seção 6 apresenta os trabalhos relacionados. Por fim, a última seção a apresenta conclusão, limitações e discute os trabalhos futuros.

2. Referenciais Teóricos

Esta seção tem como objetivo introduzir conceitos necessários para um melhor entendimento desse artigo.

2.1 Cidades Inteligentes

Segundo Hollands (2008) e Caragliu *et al.* (2011), o conceito de "Cidade Inteligente" ainda é considerado nebuloso e impreciso. Neste estudo, optou-se pela definição de Caragliu *et al.* (2011) por ser uma das mais citadas na literatura: "Uma cidade inteligente forma-se quando há coalizão de dois fatores chave: (i) investimentos em capital humano, transportes, infraestrutura e TIC para alimentar o crescimento econômico sustentável e a qualidade de vida (ii) gestão responsável dos recursos naturais por meio de uma governança participativa".

Atualmente, os estudos de cidades inteligentes podem ser conduzidos em ambientes reais ou em simuladores de cidades que utilizam dados sintéticos para apoiar experimentos baseados em cenários tais como tráfego e energia (Picone *et al.,* 2012; Darus e Bakar, 2013 e Siafu, 2017). Em geral, os simuladores são úteis para auxiliar os gestores, desenvolvedores e operadores de sistemas de cidades. No entanto, tais *softwares* não são diretamente acessíveis pela população. Além disso, não há relatos na literatura de que utilizem DAG ou beneficiem-se de proveniência de dados.

Segundo Hernandéz-Munõz *et al.* (2011), a oferta de aplicações de cidades inteligentes baseadas em ambientes reais difunde-se rapidamente nos países emergentes pois despertam interesse imediato da população. Tais aplicações são fortemente relacionados com uso das TICs e DAG. Adicionalmente, essa abordagem implica em estabelecer um compromisso entre a inovação e a transparência da gestão pública, oferecendo respostas às necessidades da sociedade civil (Dohler *et al.* 2011, Alawadhi e Scholl, 2013 e Gonçalves, Viterbo e Souza, 2016).

No Brasil já existem iniciativas de cidades inteligentes, por exemplo, Piraí, Búzios (RJ), São Paulo (SP), Porto Alegre (RS), Maceió (AL), entre outras. Caragliu *et al.,* (2011), Terán *et al.* (2016) e Batista *et al.* (2016) investigaram os desafios da mobilidade urbana nas cidades inteligentes dos países emergentes. Os autores avaliaram as iniciativas segundo perspectivas sociais, econômicas e tecnológicas, eles ressaltam que do ponto de vista tecnológico as infraestruturas computacionais mais adequadas para as atuais cidades devem ser baseadas por *middlewares* ou *softwares* multicamadas baseados em serviços de nuvens de computadores com tratamento de grandes volumes de dados apoiados por *workflows*, análise típicas de *big data* e internet das coisas, ontologias, entre outros.

2.2 Dados Abertos Governamentais

Recentemente, diferentes iniciativas internacionais (*e.g. Open Knowledge Foundation* (OKF, 2017)) e nacionais, com destaque para a Lei de Acesso à Informação Pública[3] e a Infraestrutura Nacional de Dados Abertos[4] (INDA) contribuíram para ampliar a difusão do DAG na sociedade brasileira. No Brasil, a utilização dos DAG vem se consolidando partir dos movimentos sociais que demandam transparência pública e ações de inovação aberta. Porém, a atividade de abrir dados sistematicamente ainda é uma realidade distante para uma parcela dos governos. Se considerarmos os governos municipais essa fração é ainda menor. Para grande parte dos órgãos públicos a falta de conhecimento refinado da legislação aliado com a deficiência de pessoal técnico capacitado são os principais motivos que contribuem com essa realidade (Tygel *et al.,* 2015 e Democracia Digital, 2015).

Os DAG devem seguir os seguintes oito princípios básicos (ODaF, 2017), a saber: completude; primários; atualidade; acessibilidade; processáveis por máquinas; acesso não discriminatório; formato não proprietário e ser livre de licença.

Administrações públicas oferecem catálogos de DAG com o intuito de que sua abertura e utilização possam gerar oportunidades econômicas, promover a transparência e melhorar a qualidade de vida dos cidadãos e a oferta de serviços públicos (Davies e

[3] http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm
[4] http://www.governoeletronico.gov.br/acoes-e-projetos/Dados-Abertos/inda-infraestrutura-nacional-de-dados-abertos

Edwards, 2012). Em geral, cada catálogo de DAG possui sua estrutura sintática e semântica própria e podem ser ofertados em vários formatos abertos (XML, JSON, CSV, ODF, entre outros). Portanto, compreender essas estruturas, por si só já é um desafio.

Dentre os países que mantém os portais de DAG, destacam-se: Reino Unido (data.gov.uk[5]), Canadá (open.canada.ca[6]), Estados Unidos (data.gov[7]), entre outros. Segundo o portal estadunidense, existem 52 países com portais de dados abertos e 164 portais de organismos internacionais, como a União Europeia, a Organizações das Nações Unidas, entre outros. Atualmente, o pais mantém diversos portais de DAG nos níveis Federal (*e.g.*data.gov.br), Estadual (*e.g.* governoaberto.sp.gov.br, dados.rs.gov.br, dados.al.gov.br/, entre outros) e Municipal (*e.g. Data.Rio*, dados.recife.pe.gov.br, GeoSampa[8], entre outros). Segundo o *ranking* mantido pela *Open Knowledge Internacional*[9], o Brasil ocupa o 12º lugar dentre os países que mais mantém catálogos de DAG em portais governamentais.

2.3 Portais de DAG

Um portal de DAG pode ser entendido como um serviço público essencial à publicação de catálogos de dados e seus metadados na *Web* (Oliveira e Lóscio, 2014). Em geral, os partais de DAG disponibilizam uma grande quantidade de dados da administração pública como orçamento e despesas, dos serviços oferecidos pela cidade e dados sobre a população da cidade. O ciclo de abertura dos DAG é amplo e precisa ser observado para que as iniciativas de acessibilidade sejam sustentáveis ao longo do tempo. Assim, duas das atividades primordiais nesse processo são: (i) a publicação de catálogos de dados com sua proveniência e (ii) a manutenção constante das interfaces por quem os disponibiliza e para quem os consome.

Atualmente, a ferramenta mais utilizada nos portais de DAG é o CKAN (http://ckan.org/). CKAN é um *software* livre, maduro, concebido pela OKF e mantido por ampla comunidade de usuários; ele permite a exposição de catálogos de dados, bem como funções para publicação, armazenamento e gerenciamento dos conjuntos de dados. No entanto, ainda não oferece suporte para o modelo PROV da W3C (Moreau *et al.,* 2013). A ferramenta conta com uma *API* para acesso automatizado, pré-visualização de dados, gráficos e mapas; somado à busca de catálogos de dados geolocalizados. A ferramenta é utilizada em portais governamentais brasileiros e em portais do Canadá, Reino Unido, Estados Unidos, Alemanha, Austrália, Holanda, Itália, entre outros.

2.4 Proveniência em Dados Abertos

Buneman *et al.* (2001) indica que a proveniência de dados também chamada de linhagem, genealogia ou *pedigree*, consiste de metadados que descrevem as origens de um item de dado ou do processo pelo qual o dado foi produzido. Ou seja, ela representa a ancestralidade de um objeto e pode ser descrita em diferentes termos, dependendo do domínio abordado (Gil *et al.,* 2010).

[5] https://data.gov.uk/
[6] http://open.canada.ca/en
[7] https://www.data.gov/open-gov/
[8] http://geosampa.prefeitura.sp.gov.br/
[9] http://index.okfn.org/place/

Como regra geral, a proveniência fornece a documentação essencial para registrar a origem, a qualidade e a autoria dos dados, auxiliando na capacidade de reproduzir e validar resultados de experimentos científicos ou comerciais (Freire *et al*, 2008). Segundo vários autores, (Simmhan *et al.*, 2005, Freire *et al.*, 2008, Cruz *et al.*, 2009, Oliveira *et. al.*, 2014, Pimentel *et al.,* 2016), a proveniência pode ser classificada de diversos modos. A proveniência prospectiva, que é aquela que registra a especificação de tarefas computacionais, ela corresponde às etapas a serem seguidas para chegar a um resultado. Já a proveniência retrospectiva diz respeito ao registro da execução de processos computacionais, levando em conta as características operacionais do ambiente computacional utilizado para derivar um resultado, a proveniência consiste de um histórico estruturado e detalhado de uma execução de tarefas computacionais. Por limitações de escopo, neste trabalho consideramos apenas este tipo de proveniência.

A proveniência pode ser representada por modelos conceituais e grafos, um dos principais modelos para representar a proveniência na Web é o PROV-DM derivado da especificado PROV da W3C (Moreau *et al.,* 2013) (Figura 1). O modelo expressa a proveniência através de estruturas cujos elementos centrais são:

(i) Entidades (*e.g. Entity*, *Activity* e *Agent*);
(ii) Relacionamentos e propriedades, ou seja, dependências causais entre as entidades (*e.g. wasAttributedTo*, *wasGeneratedBy*, entre outros).

Além disso, o modelo permite representar a proveniência através de estruturas estendidas tais como:

(iii) Relações expandidas alternativas, *bundles* (representação da proveniência da proveniência), propriedades, entre outros.

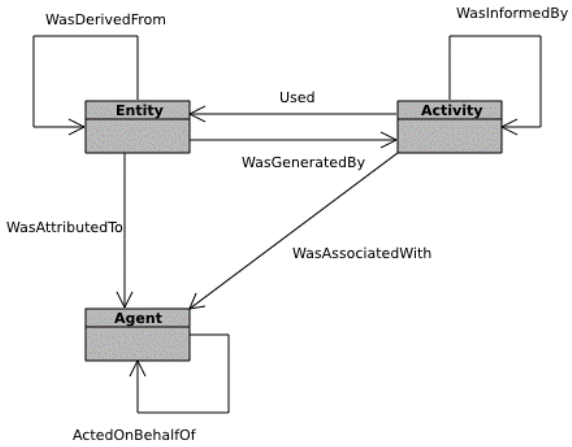

Figura 1. Representação esquemática (em linguagem UML) entre os elementos centrais do modelo PROV-DM (adaptado de Moreau *et al.,* 2013).

O modelo PROV-DM auxilia na identificação da cadeia de processos envolvidos na transformação dos DAG Brutos. As semânticas elementos de proveniência relacionados com o DAG Brutos utilizados neste trabalho são apresentados na Tabela 1.

Além dos modelos conceituais, alguns domínios exigem que a proveniência seja armazenada em vários níveis de detalhes, isto é, granulosidades distintas. O conceito de granulosidade de dados que é uma importante definição oriunda da área de *datawarehouse* também se aplica à área de proveniência (Simmhan *et al.*, 2005).

Tabela 1. Elementos centrais do modelo PROV-DM utilizados pelo *workflow* ETL.

Elemento (tipo)	PROV-DM	Descrição
ENTITY (nó)	Entidade	Entidade que identifica um objeto físico, digital ou um catálogo de dados (DAG)
wasAttributedTo (aresta)	Relacionamento	Relaciona o catálogo de dados ao agente transformador do objeto
wasDerivedFrom (aresta)	Relacionamento	Relaciona os objetos envolvidos em uma derivação
wasGeneratedBy (aresta)	Relacionamento	Relaciona uma atividade a um objeto
Usage (aresta)	Relacionamento	Relaciona um objeto a uma atividade
ACTIVITY (nó)	Entidade	Entidade que identifica o processo de criação de uma entidade
wasAssociatedWith (aresta)	Relacionamento	Relaciona o agente com a execução de uma atividade
wasInformedBy (aresta)	Relacionamento	Relaciona os processos envolvidos na execução de uma atividade
AGENT (nó)	Entidade	Entidade que identifica uma pessoa ou *software* (agente) que está associado ou é responsável por uma entidade
actedOnBehalfOf (aresta)	Relacionamento	Relaciona os agentes por delegação de responsabilidades
Timestamp (propriedade)	Extensão do modelo PROV	Data e hora da transformação do registro em um catálogo em DAG curado
Status (propriedade)	Extensão do modelo PROV	Indicador do estado da transformação do registro
TipoErro *(propriedade)*	Extensão do modelo PROV	Indicador de detecção de erro ao processar um registro. Em caso de erro há a inclusão da descrição do erro detectado.

O grão é o nível de detalhe da informação, ele é definido de acordo com as necessidades estabelecidas na aplicação. Ou seja, é possível que haja uma variação no tamanho do grão do item de dado. Quanto menor for o tamanho do item, menor será sua granulosidade. Por exemplo, informações de *timestamp* são consideradas como grão pequeno, sendo indivisíveis em unidades menores. Logo, quanto maior o detalhamento da proveniência, mais refinadas serão as possibilidades de consultas sobre os DAG. Por outro lado, quanto mais detalhado, mais espaço de armazenamento será necessário para armazenar os dados e sua proveniência. Portanto, definir a granulosidade da proveniência em aplicações voltadas para as cidades inteligentes é uma questão importante.

Nos últimos anos, gerência da proveniência tem sido extrapolada em áreas tais como o monitoramento de aplicações (Salfer *et al.,* 2011), publicação e ligação de dados abertos (Hartig, 2009 e Mendonça *et al.,* 2014, 2016), problemas de *Big Data* em ambientes distribuídos (Imran e Hummel, 2008; Bertino, 2013 e Ferreira *et al.,* 2014), entre outros. No entanto, verifica-se que existem poucas iniciativas que exploram conjuntamente DAG, sua proveniência e usos em aplicações de cidades inteligentes sendo executados em ambientes distribuídos onde a demanda por armazenamento e consultas pode variar dinamicamente ao longo do tempo.

Corsar e Edwards (2012) e Emaldi *et al.* (2013) afirmam que apesar da área de proveniência de dados ser considerada como promissora e de já existirem trabalhos voltados para a publicação e recuperação de dados abertos, ainda há carência de estudos sobre proveniência de dados em aplicações móveis para cidades inteligentes.

Adicionalmente, segundo McArdle e Kitchin (2015) existem poucos estudos evidenciando ou exemplificando aplicações que agreguem proveniência em dados abertos não ligados em aplicações voltadas para cidades inteligentes.

Neste trabalho, vamos ao encontro das observações dos autores supracitados. Para isso, adotamos uma abordagem onde coletamos DAG e fazemos uso da especificação PROV-DM para identificar e anotar os dados de proveniência retrospectiva de baixa granulosidade, mapeando o processo de ETL (Extração, Transformação e Carga) de DAGs, oferecendo evidências da consistência e qualidade dos catálogos de DAG curados utilizados no sistema *BusInRio.*

3. Materiais e Métodos

Esta seção apresenta os catálogos de dados e a metodologia utilizada no desenvolvimento do *BusInRio*.

3.1. Catálogos de dados e suas estruturas

Nesse trabalho utilizamos vários catálogos de dados sobre mobilidade urbana disponibilizados através do portal de dados da Prefeitura da Cidade do Rio de Janeiro. Resumidamente, os volumes de dados produzidos diariamente são medidos em megabytes/dia. Os dados são de natureza textual, semiestruturada e são caracterizados por arquivos com milhões de registros produzidos pelos GPS de aproximadamente 9.000 ônibus e centenas arquivos únicos das 490 linhas de ônibus que atendem ao município.

Os catálogos de dados contendo a localização das paradas (pontos de ônibus) de cada linha de ônibus são disponibilizados pela FETRANPOR[10], através do *Data.Rio*, em arquivos de texto no formato CSV. Os dados possuem a seguinte estrutura: descrição de início e fim daquela linha; agência que disponibiliza os dados; número de sequência em que o ônibus passa em um determinado ponto; coordenadas (latitude e longitude) do ponto de ônibus.

Os catálogos de dados dos GPS instalados nos ônibus são produzidos a cada minuto e disponibilizados em tempo real[11] em formato JSON. Este conjunto de dados cobre parte da frota das concessionárias da Cidade do Rio de Janeiro e algumas linhas intermunicipais. A estrutura geral dos dados dos GPS é a seguinte: data e hora da geração do dado; identificação alfanumérica encontrada na lateral do ônibus; identificador da linha do ônibus; coordenadas (latitude e longitude) do ônibus no trajeto; velocidade do ônibus no momento da coleta e medida em graus em relação ao norte, que representa a direção do veículo.

Os catálogos de dados de GPS das trajetórias dos ônibus não são compostos por dados livres de ruídos, pelo contrário, eles apresentam muitos erros e falhas tais como: ausência de informação da direção, erros de representação das coordenadas, falhas de coleta de dados (apesar da frequência de coleta de dados de GPS deveria ser inferior a um minuto, algumas medidas são superiores a 10~12 minutos), erros no registro das velocidades dos coletivos, entre outros.

[10] http://data.rio/dataset/pontos-de-parada-de-onibus
[11] http://data.rio/dataset/gps-de-onibus

3.2. Metodologia

Esta subseção descreve a metodologia adotada no trabalho. As três fases principais realizadas após a pesquisa bibliográfica dos trabalhos científicos e aplicativos da área se caracterizam por.

Primeira fase - foi necessário buscar trabalhos relacionados e compreender como e quais dados o poder público disponibilizava para a sociedade civil. Houve a necessidade de avaliar os portais e os catálogos de DAG brutos e propor um método sistemático de coleta e tratamento desses dados.

Segunda fase - concebeu-se uma arquitetura aberta capaz de utilizar DAG curados e enriquecidos com proveniência retrospectiva. A arquitetura utiliza *workflows* ETL para detectar *outliers* nos catálogos de dados e efetuar a ordenação, limpeza e enriquecimento de dados. A arquitetura alinha-se com os indicadores de cidades inteligentes enunciados por Nam e Pardo (2011), Terán *et al.* (2016) e Batista *et al.* (2016). Os resultados desta fase são apresentados e discutidos nas seções 4 e 5.

Terceira fase - avaliaram-se os artefatos tecnológicos desenvolvidos. Realizamos estudos de caso com experimentos da arquitetura e além de estudos do tipo *survey* (Wohlin *et al.,* 2012) envolvendo amostragem com dados fornecidos por usuários frequentes do sistema de transporte da cidade. Os dados desses experimentos são coletados por meio da própria arquitetura e por questionários de auto aplicação voltados para os usuários da aplicação. Os resultados e análises dessa fase serão apresentados e discutidos na seção 5.

Para assegurar a significância estatística dos experimentos, selecionamos aleatoriamente uma amostra de 36 usuários dentre uma população de 57 voluntários (em sua maioria universitários ou secundaristas moradores das cidades do Rio de Janeiro ou da Baixada Fluminense) para que utilizassem o *BusInRio* durante um período de 14 dias e respondessem aos questionários. Os respondentes, segundo Ferris *et al.* (2010) devem ter algumas características importantes: ser usuários frequentes de transporte público, possuir *smartphones* e, ter familiaridade com o uso da Internet.

O questionário contém questões fechadas do tipo múltipla escolha, com opções quantitativas e escalonadas representadas em escala de *Likert* em quatro níveis (Vieira, 2009). A mensuração da satisfação avaliada através de um questionário tem como objetivo simplificar o tratamento estatístico das respostas. Os resultados desta fase são apresentados e discutidos na seção 5.

4. Projeto e Desenvolvimento

Esta seção apresenta uma versão ampliada da arquitetura *BusInRio* e seus novos componentes e camadas, que utilizam DAG curados de mobilidade urbana anotados com proveniência retrospectiva.

4.1 Arquitetura *BusInRio*

Segundo Kon e Santana (2016), a maneira mais racional para o desenvolvimento de Cidades Inteligente é a utilização de plataformas de software. O *BusInRio* é uma arquitetura que amplia as pesquisas iniciadas por Andrade e Cruz (2015), Cruz, Andrade e Oliveira (2016), adotando os referenciais teóricos e tecnológicos apresentados por Terán *et al.* (2016) e Batista *et al.* (2016). *BusInRio* foi reprojetada para disseminar

informações públicas em conformidade com a política do governo brasileiro para dados abertos definido pelo INDA. A arquitetura *BusInRio* é do tipo distribuída, modular e escalável e adota o padrão arquitetônico cliente-servidor.

A Figura 2 ilustra sua representação simplificada e os tipos de principais fluxos de dados e metadados. Sua porção servidora executa em ambiente de nuvem de computadores do tipo pública e modelo orientado a serviços SaaS (Chee, Franklin Jr, 2013) e a porção cliente pode ser executada em desktops, *tablets* ou *smartphones* com diversos tipos de sistema operacional.

Os módulos do servidor de aplicação têm como principais funcionalidades conectar-se ao site do *Data.Rio* através do serviço CKAN, executar os *workflows* ETL para coletar, tratar os DAG de transporte público e armazená-los juntamente com sua proveniência em bases *NoSQL* da *Data Fabric* (MongoDB) e disponibilizar uma *API* e um sistema de filas para atender as requisições da aplicação móvel que consulte e exibe os DAG curados e mapas.

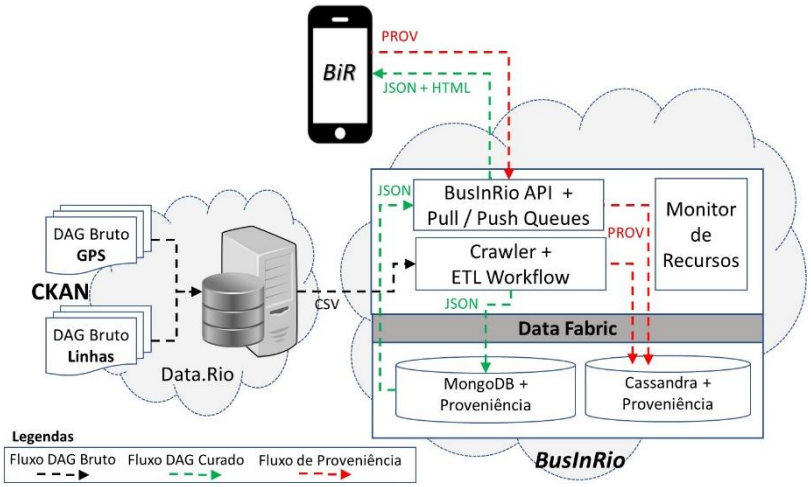

Figura 2. Representação conceitual da arquitetura *BusInRio* e dos fluxos de dados e proveniência *(baseada em Cruz, Andrade e Oliveira, 2016)*.

Os clientes (aplicativo *BiR*) possuem as seguintes funcionalidades: consultar os DAG curados através da *API* e exibir para o usuário lista de linhas de ônibus e exibir as representações gráficas através de mapas que utilizam as *API*s do *Google Maps* que permitem e visualização de dados sobre o ônibus e sobre trânsito do entorno.

BusInRio é uma arquitetura que explora os princípios de distribuição e elasticidade das infraestruturas das nuvens subjacentes, à medida que aumentam as requisições dos clientes ou percentagem de uso de *CPU* das máquinas virtuais e de memória variam para acima (ou para baixo) de nível pré-definidos, novas instâncias são dinamicamente iniciadas (ou desabilitadas) conforme o caso.

4.2 *Workflow* ETL

O gerenciamento de dados é uma etapa fundamental para qualquer estratégia de desenvolvimento de aplicações para cidades inteligentes baseadas em nuvem de computadores.

De acordo com Mendonça *et al.* (2014, 2016), a abordagem ETL pode ser utilizada para orquestrar os processos extração dos dados de múltiplas fontes

heterogêneas, seguidos das atividades de limpeza, consolidação, agregação e integração, antes de transformá-los em DAG curados para serem disponibilizadas aos usuários. Os DAG utilizados pela *BusInRio* possuem tais características que demandam frequentes etapas de detecção de *outliers*, limpeza e tratamento antes de serem disponibilizados para os usuários finais.

Neste artigo, adotamos a abordagem de *workflows* ETL, proposta por Mendonça *et al.* (2014, 2016), para tratar os dados abertos. Resumidamente, *workflows* ETL são programas encadeados que realizam tarefas de *Extração, Transformação, Carga, Enriquecimento e Publicação* de dados abertos. Nossa abordagem é composta por cinco atividades para normalizar os dados e garantir a qualidade e a integridade dos DAG (Figura 3), a saber:

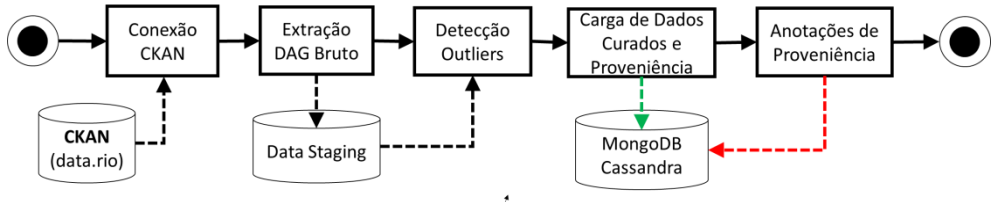

Figura 3. Representação conceitual do *Workflow* ETL utilizado no *BusInRio*.

A. *Conexão com fonte de dados primários*: nesta atividade se estabelece a conexão com o servidor CKAN do *Data.Rio*, ele é o servidor de DAG brutos que serão utilizados pelo *BusInRio*.

B. *Extração de dados abertos brutos*: nesta atividade os dados brutos são coletados do servidor, os dados são temporariamente armazenados em repositório na nuvem (*data staging area*) antes que qualquer tipo de manipulação;

C. *Detecção de outliers*: nesta atividade os dados são corrigidos, etiquetados com proveniência, consolidados e transformados em dados curados de acordo com os requisitos da aplicação. As regras de limpeza de dados são:

(i) detecção de registros duplicados ou incompletos;

(ii) detecção de registros com distâncias superiores à 100 metros a partir da rota esperada;

(iii) detecção de trajetórias com intervalos de GPS inferiores a 1 minuto ou superiores a 10 minutos;

(iv) detecção de registros com *timestamps* superiores a data/hora da extração do CKAN;

(v) detecção de registros de ônibus e linhas de ônibus não registradas,

(vi) detecção de registros de ônibus com velocidades superiores a 80 Km/h, entre outras.

Como atividade de consolidação calculamos a direção de cada ônibus em um dado período de tempo e ao fim ligamos os catálogos de dados das linhas de ônibus e GPS. A tarefa de normalização dos dados assemelha-se à proposta apresentada por Barbosa *et al.* (2014), utilizamos como heurística o cálculo da distância euclidiana entre duas medidas do GPS para calcular a trajetória percorrida pelos ônibus.

D. *Carga de dados curados:* nesta atividade os dados consolidados são (re)agrupados em documentos ou família de colunas e carregados para os repositórios *NoSQL* apropriados.

E. *Anotação de proveniência retrospectiva*: nesta atividade os dados de proveniência do tipo retrospectiva que foram produzidos pelos processos ETL são utilizados par armazenar, anotar e ligar a cada um dos repositórios de dados curados. As anotações de proveniência se baseiam em termos de vocabulários tradicionais da Web de dados (*e.g. PROV, FOAF (Friend Of Friend)*[12]*, DCTerms (Dublin Core)*[13] e BiR*(BusInRio)*[14]). As anotações auxiliam na descrição de dados e grafos oferecendo detalhes do tipo, quando, onde e como catálogo de dados foi coletado e quando foi processado antes da publicação como DAG curado. Esses termos, em última análise, se adequam a especificação PROV da W3C e contribuem para ampliar a garantia da qualidade dos DAG curados. Detalhes do processo de enriquecimento de DAG estão disponíveis nas subseções 4.5 e 5.3.

4.3 *BusInRio Crawler*

O *crawler* é um módulo central da arquitetura, ele executa no servidor e consiste em acessar recursivamente de tempos em tempos os servidores CKAN do *Data.Rio* para recuperar novos DAG brutos ao invocar a execução do *workflow* ETL. O *crawler* não só alimenta as bases de dados *NoSQL* com os dados curados sobre os GPS dos ônibus, como registra quais foram os intervalos de coletas de dados brutos.

4.4 *Data fabric* e bancos *NoSQL*

A *data fabric* é uma camada de armazenamento e gerenciamento de dados, composta por *softwares* e baseada em memória que utiliza recursos de todas máquinas virtuais alocadas (memória, CPU, largura de banda de rede e disco local), ela é usada para gerenciar os DAG, sua proveniência e anotações geradas pela aplicação. Quando necessário, a *data fabric* utiliza técnicas de replicação dinâmica e particionamento de dados para oferecer disponibilidade contínua e escalabilidade para a arquitetura, sem comprometer a integração ou consistência dos DAG.

Os bancos de dados *NoSQL* são considerados como sendo soluções escaláveis e ideais para tratar os problemas de *Big Data*. Tais bancos não são baseados em esquemas fixos e recentemente passaram a ser considerados uma boa alternativa para armazenar a proveniência de *workflows* executados em ambientes de nuvens pois provêm escalabilidade do tipo horizontal e baixa latência (Ferreira *et al.,* 2014 e Guimarães *et al.,* 2015).

Na nossa proposta, a *data fabric* explora as funcionalidades de acesso, segurança, escalabilidade e integração de dados de dois tipos de bancos *NoSQL* para armazenar DAG e sua proveniência (Abramova, Bernardino, 2013). O sistema *NoSQL* utilizado nesta versão da arquitetura é o MongoDB (MongoDB, 2017).

O MongoDB é um sistema não relacional de alta performance de código aberto e sem esquemas fixos que adota o paradigma da orientação a documentos. O MongoDB

[12] http://xmlns.com/foaf/spec/
[13] http://dublincore.org/documents/dcmi-terms/
[14] Por questões de escopo, o vocabulário BiR não será discutido neste artigo.

nos permitiu modelar a proveniência e as informações do posicionamento dos ônibus de modo simplificado, os dados JSON foram aninhados em hierarquias complexas e continuaram a ser indexáveis e facilitando sua recuperação por parte da aplicação cliente. A este serviço foram criados vários bancos, a saber: “transporte”, uma coleção “onibus” capaz de armazenar as coordenadas do GPS de cada ônibus monitorado pelo sistema; uma coleção chamada “transporte capaz de armazenar os percursos e linhas para representar os DAG curados das linhas, percursos e pontos de ônibus. Em "linha" armazena-se os dados sobre a linha do ônibus (identificação e descrição) e no "percurso" os pontos pelo qual o ônibus passa.

4.5 Enriquecimento de DAG com Anotações de Proveniência

A proveniência pode enriquecer dois tipos de dados na arquitetura *BusInRio*: (i) registros e catálogos de DAG transformados pelo *workflow* ETL e (ii) dados de utilização da arquitetura *BusInRio* pelos usuários do aplicativo móvel.

A Figura 4 exemplifica o enriquecimento de registros por meio da detecção de *outliers* ao se aplicarem as regras de limpeza de dados. Na figura 4(A), o destaque na cor vermelha indica novos campos que enriquecem os dados brutos oriundos do CKAN. Esses campos são complementares e auxiliam na elaboração de consultas mais seletivas pelo aplicativo *BusInRio*. Por exemplo, graças aos novos campos é possível excluir ônibus ou linhas que possuem problemas em suas rotas, coordenadas de GPS ou linhas, consequentemente amplia-se a acurácia do aplicativo móvel.

A Figura 4(B) (direta) ilustra um registro (em formato JSON) gerado pelo *workflow* ETL, ele contém apenas dados curados e enriquecidos por proveniência. De acordo com o exemplo, destaques na cor verde, o campo “linha” não possui valor; neste caso a detecção da falha foi identificada e mapeada como um erro.

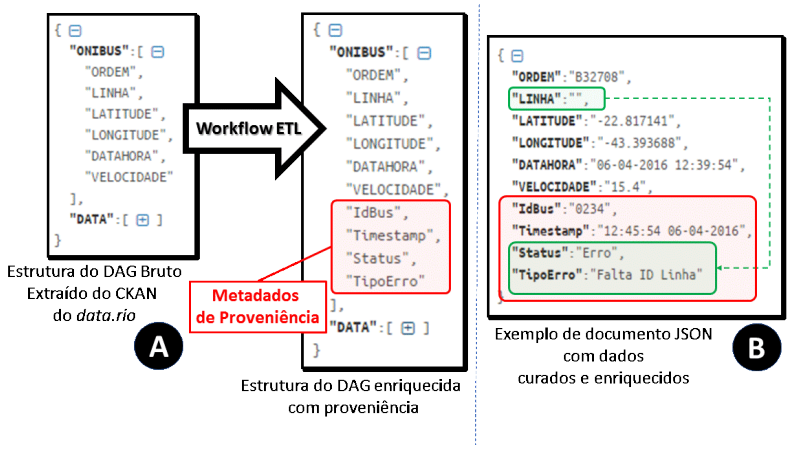

Figura 4. (A) Exemplo de transformação e enriquecimento dos DAG (B) Exemplo de fragmento de documento JSON enriquecido com proveniência.

4.6 API do *BusInRio* e Serviço de Filas

A *API* do *BusInRio* é composta por funções acessíveis somente por programação que exploraram as características de ubiquidade da arquitetura, ela permite que as requisições da aplicação cliente sejam executadas em *smartphones* e convertam os dados para o formato JSON que é facilmente interpretado pelas aplicações móveis e navegadores *web*.

Através da *API BusInRio* e do serviço de mensagens (filas), o aplicativo pode enviar requisições HTML parametrizadas para o servidor de aplicação, por exemplo, número da linha escolhida pelo usuário e data; então o servidor responde com DAG curados sobre os ônibus (em formato JSON) que será interpretado pelo aplicativo e disposto na interface para visualização do usuário.

O serviço de filas de mensagens do *BusInRio* utiliza de dois tipos de filas (*push* e *pull*). Elas operam isoladamente e representam a maneira pela qual as operações (requisições de dados e suas respostas) sejam atendidas. A fila *push* permite acelerar operações. Por exemplo, ela recebe as requisições de dados do aplicativo móvel e dispara uma tarefa de busca de dados (consulta) na *data fabric* ou dispara a execução do *crawler* caso não o dado solicitado pelo aplicativo não esteja disponível. A fila *pull* não dispara tarefas de execução, ela recebe os dados da camada *data fabric* e disponibiliza o DAG curado ou mensagens de erro para a aplicação móvel.

O serviço de filas de mensagens desempenha uma outra tarefa, ele foi configurado para coletar a proveniência retrospectiva geradas pelas operações de requisições ou respostas de DAG. Os dados de proveniência de baixa granulosidade são armazenados no repositório MongoDB. Mais especificamente, armazenam-se dados (em formato JSON) de acordo com as estruturas de requisição de serviço. As Figuras 5 (A e B) ilustra as estruturas de requisições e respostas.

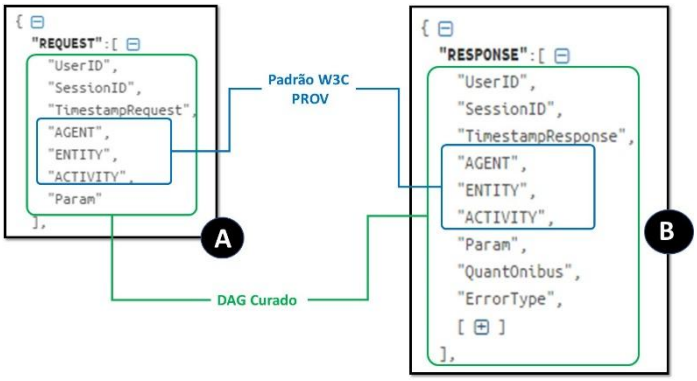

Figura 5. Representação (em formato JSON) das mensagens de *request* e *response*. (A) Estrutura da mensagem de requisição de DAG curado à arquitetura *BusInRio* (fila *push*), (B) Estrutura da mensagem de resposta (fila *pull*) com DAG curados e enriquecidos com proveniência.

As semânticas dos descritores de proveniência utilizados pelas mensagens de requisição de dados (*request)* e resposta (*resposnse)* são compatíveis com os elementos de proveniência apresentados na Tabela 1. Além disso, há o identificador de usuário (*UserID*), sessão (*SessionID*), rótulo de *timestamp* e parâmetro que identifica a linha de ônibus sendo consultada pelo usuário. Ressaltamos que nas consultas realizadas pelo aplicativo móvel utilizamos apenas elementos centrais da PROV-DM (*e.g. Entity, Agent e Activity*). Essa decisão de projeto visou reduzir: (i) a quantidade de dados a ser processada; (ii) o tempo de transferência entre a arquitetura e o aplicativo móvel, e (iii) a sobrecarga de processamento na nuvem.

A proveniência coletada pelo serviço de filas permite avaliar, por exemplo: quais itens de DGA curados foram acessados pelos usuários móveis; quais foram os erros do sistema, quais as estimativas tempos de processamento das operações. Esses dados serão analisados na subseção 5.4.3.

4.7 Monitor de Recursos

O serviço de monitoramento de recursos coleta dados de desempenho no sistema operacional das máquinas virtuais e dos servidores de dados NoSQL. As métricas avaliadas estão limitadas a percentual de uso de CPU, taxa de entrada e saída de dados, taxa de transferência de leitura e gravação de dados nos repositórios NoSQL. O serviço armazena os dados em arquivos de log no MongoDB e quando configurado pode reagir automaticamente a alterações na disponibilidade de recursos, disparando ou encerrando instâncias de máquinas virtuais da arquitetura para garantir seu nível de serviço e desempenho.

5. Provas de Conceito

Nesta seção, serão apresentados detalhes sobre as avaliações dos protótipos da arquitetura e do aplicativo.

5.1 Protótipo da Arquitetura

O protótipo da arquitetura *BusInRio* utilizou o serviço de nuvem pública baseado no modelo SaaS oferecido pelo provedor Koding[15] ele disponibiliza uma *stack* com o ambiente de desenvolvimento sob a forma de instâncias em servidores virtuais do tipo AmazonEC2.

As instâncias utilizadas nesse trabalho foram do tipo *t2.medium* (4 CPU e 6 GB de RAM virtuais), a configuração da *stack* forneceu um ambiente computacional de uso geral e de custo reduzido onde se pode hospedar servidores *web*, ambientes de desenvolvimento, *workflows*, serviços de fila, e os repositórios *NoSQL* de DAG curados, dados de proveniência retrospectiva e dados dos experimentos. Os códigos da aplicação servidora foram desenvolvidos em linguagem PHP/Java, o *workflow* ETL foi desenvolvido em Python, já o cliente é totalmente desenvolvido com tecnologias *Web* responsivas e os repositórios de dados curados e de proveniência são do tipo não relacional.

5.2 Protótipo do Aplicativo

O protótipo do aplicativo *BiR* (acrônimo de *BusInRio*) foi desenvolvido no *framework* Ionic[16]. O *framework* é um *kit* de desenvolvimento que permite o desenvolvimento de aplicativos híbridos capazes de executar em *smartphones* com diversos tipos de sistema operacional e navegadores *web*. O *framework* está baseado no AngularJS e seu núcleo compõe funcionalidades do Apache Cordova. O aplicativo *BiR* foi desenvolvido em HTML5, Javascript e CSS e consome os dados semiestruturados disponibilizados pela *API* do Google Maps.

Segundo Ferris *et al.* (2010), o desenvolvimento de aplicativos voltados para o contexto das cidades inteligentes deve considerar o público alvo. No caso do aplicativo *BusInRio* utilizamos a mesma classificação recomendada pelos autores. Classificamos os usuários potenciais como frequentes e não frequentes. Os usuários frequentes estão familiarizados com a Cidade do Rio de Janeiro e o transporte coletivo. Já os não frequentes não possuem informações sobre a cidades ou rotas, porém necessitam utilizar

[15] https://koding.com/
[16] http://www.ionicframework.com

o modal. Tendo em vista essa classificação, o aplicativo *BiR* teve suas interfaces gráficas simplificadas para atender as duas classes de usuários. O aplicativo permite aos usuários escolher, a partir do *smartphone*, a linha de ônibus que desejar e consultar a localização dos ônibus daquela linha em tempo real.

5.3 Anotações e Visualizações Grafos de Proveniência Retrospectiva

Conforme apresentado anteriormente, os dados de proveniência produzidos pelo *BusInRio* são armazenados na camada *data fabric* e são representados em conformidade com o modelo conceitual PROV-DM proposto por Moreau *et al.* (2013). As anotações de proveniência foram produzidas através da biblioteca PROV 1.5.0[17] e são serializadas no formato PROV-JSON no servidor MongoDB. Os diagramas de proveniência presentes nesta subseção foram desenvolvidos através da ferramenta ProvStore[18] com os dados produzidos pela arquitetura *BusInRio*.

5.3.1 Proveniência gerada pelo *workflow* ETL

A Figura 6 representa um fragmento de dados de proveniência no formato PROV-JSON produzido pela arquitetura *BusInRio* que contém anotações, relacionamentos e dados de proveniência retrospectiva. O fragmento representa um grafo de proveniência composto por nós, arestas e anotações que ilustra as todas transformações aplicadas a um catálogo de DAG (coleção) representada pelo arquivo DAG-2016-06-04-FILE015.CSV que contém os dados bruto extraídos diretamente do portal da Prefeitura.

O grafo é uma rede de relacionamentos que utiliza os elementos centrais do modelo PROV-DM (apresentado na subseção 2.4). Além disso, também utilizamos termos dos vocabulários *PROV, FOAF, DCTerms* e *BiR* para anotar e descrever os grafos de proveniência gerado pela arquitetura. Resumidamente, na Figura 7 verificam-se que os agentes BiR:*AdminBiR* que está associado com a execução de todas as atividades do *workflow* e o agente BiR:*MobileUser* associado com as consultas realizadas através de dispositivos aos DAG curados disponíveis no MongoDB.

As entidades representam as coleções de DAG e as atividades são os processos de transformação de dados, cada atividade possui rótulos de *timestamp* de início e fim de processamento (*e.g.* BiR*:ExtracaoDAGBruto*). Os elementos do grafo conectam-se através de arestas, cujas semânticas estão disponíveis na Tabela1.

No fragmento, verifica-se que o agente BiR*:AdminBiR* está conectado através de três tipos de arestas (relacionamentos) com os demais elementos. O relacionamento *wasAssociatedWith* indica que o agente está associado com as atividades de transformação dos dados brutos em DAG curado. O relacionamento *actedOnBehalfOf* indica que a administrador delega as atividades de consulta para o usuário móvel. O relacionamento *wasAttributedTo* indica que as transformações aplicadas aos objetos, isto é, catálogos de dados são de atribuição do administrador da arquitetura. Neste caso, a semântica dos relacionamentos indica que este agente se conectou ao depósito de DAG brutos do *Data.Rio*, realizou as operações de ETL através das entidades subjacentes com sucesso; caso essas ações não fossem executadas, o grafo não poderia

[17] https://pypi.python.org/pypi/prov
[18] https://provenance.ecs.soton.ac.uk/store/

ser representado e os DAG curados não estariam disponíveis nos repositórios NoSQL para as operações de consulta através dos dispositivos móveis.

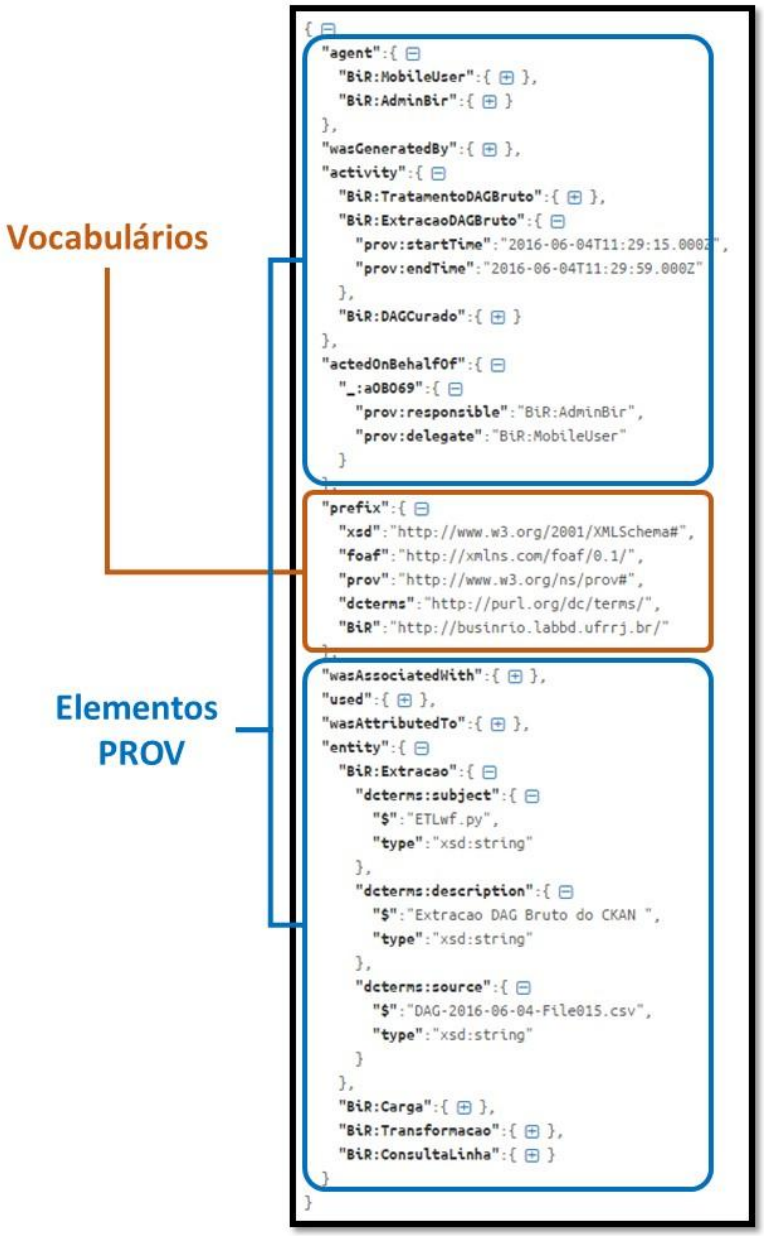

Figura 6. Exemplo fragmento de arquivo proveniência retrospectiva gerada pela arquitetura *BusInRio* em conformidade com o modelo PROV-DM.

Uma outra forma de verificar a proveniência produzida pela *BusInRio* é através dos diagramas de Sankey. A Figura 7 apresenta um diagrama gerado a partir dos dados de proveniência produzidos pelo *workflow* ETL ao se processar o arquivo DAG-06-04-2016.CSV. Como pode ser visto, ela demonstra o envolvimento de todas as atividades do *workflow* desde a extração dado bruto até a consulta dos dados curados pelo usuário do aplicativo móvel.

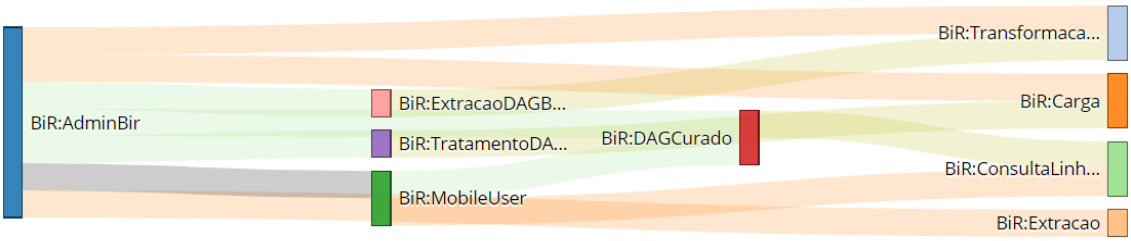

Figura 7. Diagrama de Sankey ilustra a proveniência retrospectiva relacionada ao agente BiR:*AdminBiR*.

O diagrama da Figura 7, cuja leitura faz-se da direita para a esquerda, exibe os fluxos da proveniência ligados a produção de DAG curados correlacionados ao ator BiR:*AdminBiR*. Os fluxos coloridos descrevem os tipos de relacionamentos existentes entre as entidades que são identificadas por barras horizontais. As cores dos fluxos descrevem os tipos de relacionamentos. Na Figura 7, temos que os fluxos na cor azul clara indicam associações entre os agentes e as atividades. O fluxo na cor lilás claro indica delegação entre os agentes. Por fim, os fluxos na cor rosa claro indicam as atribuições do agente.

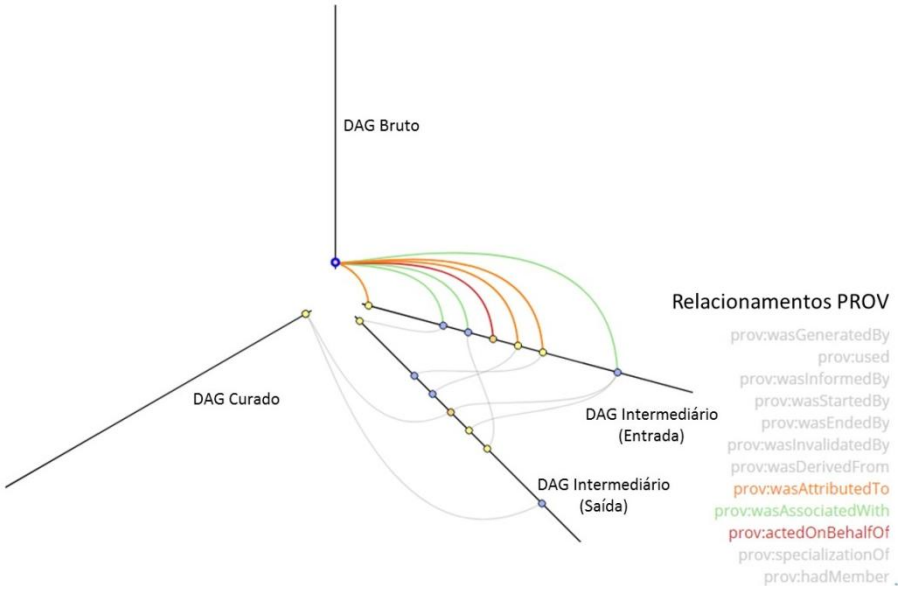

Figura 8. Diagrama do tipo Hive ilustrando a proveniência retrospectiva dos dados de entrada, intermediários e saída relacionadas ao agente BiR:*AdminBiR*.

A Figura 8 indica um diagrama de Hive, ele é utilizado para ilustrar os relacionamentos entre as atividades, entidades e agentes do modelo PROV de um arquivo de proveniência retrospectiva. Os eixos indicam a proveniência dos dados de entrada, intermediários e de saída processados do *workflow* ETL. Os relacionamentos do modelo PROV estão agrupados ao longo dos eixos. Na Figura 8, os arcos indicam os relacionamentos existentes entre agentes, entidades e atividades do modelo PROV. Por exemplo, os arcos coloridos indicam os sete tipos de arestas existentes partindo do agente que inicia o processamento dos DAG bruto (*e.g.* BiR:*AdminBiR*) para as entidades e atividades relacionadas com as transformações de dados. As arestas na cor verde claro indicam os três relacionamentos do tipo prov:*wasAssociatedWith* entre o agente BiR:*AdminBiR* e dados intermediários produzidos pela arquitetura *BusInRio*. As três arestas na cor laranja indicam o relacionamento do tipo prov:*wasAttributedTo* entre

o mesmo agente e dados intermediários produzidos pelo *workflow* ETL. A aresta na cor vermelha indica o relacionamento por delegação entre os agentes BiR:*AdminBiR* e BiR:*MobileUser*.

5.3.2 Proveniência retrospectiva gerada pela aplicação móvel

A Figura 9 representa um fragmento de dados de proveniência no formato PROV-JSON produzido pela aplicação móvel *BiR,* ela contém anotações, relacionamentos e dados de proveniência retrospectiva. O fragmento representa um grafo de proveniência composto por nós, arestas e anotações que ilustra três consultas aos dados curados realizadas por um usuário móvel.

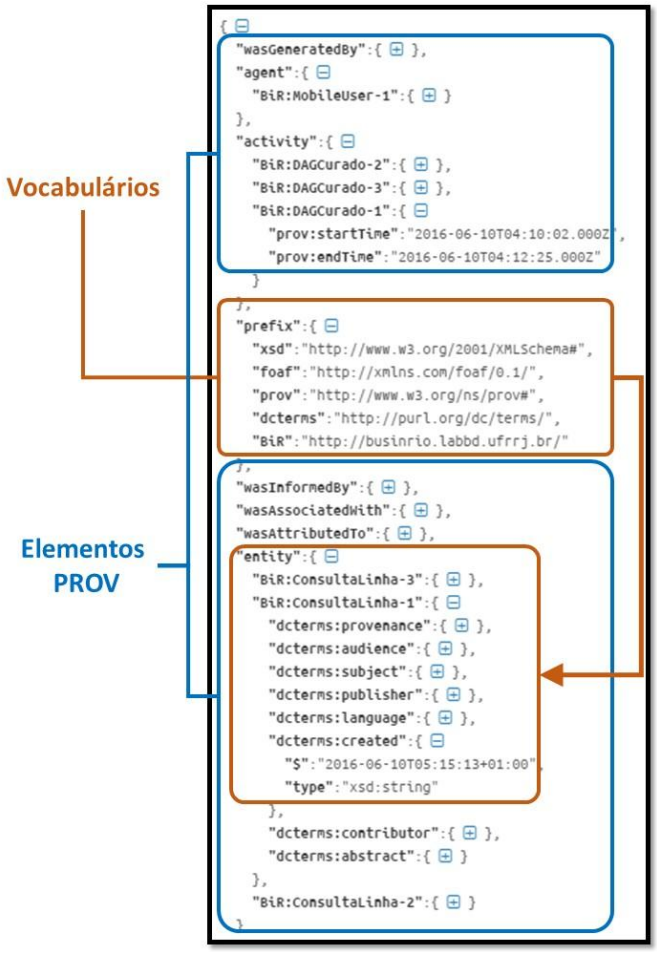

Figura 9. Exemplo fragmento de arquivo proveniência retrospectiva gerado pelas filas *Push* e *Pull* da arquitetura *BusInRio* em conformidade com o modelo PROV-DM.

A Figura 10 apresenta um diagrama de Sankey gerado a partir dos dados de proveniência retrospectiva capturados pelas filas *Push* e *Pull* da arquitetura *BusInRio* descritas na subseção 4.6. Como pode ser visto, o diagrama demonstra três interações do agente BiR:*MobileUser-1*, a cada interação o usuário do aplicativa móvel consulta (*e.g.* BiR:*ConsultaLinha*) um catálogo de dados curados bruto. Destacamos que as cores dos fluxos descrevem os tipos de relacionamentos. Por exemplo, na Figura 10, temos que os fluxos na cor rosa clara indicam associações entre os agentes e as atividades de consulta de dados curados. Destacamos que apesar de existirem três interações entre o agente e as atividades, apena duas foram executadas com sucesso.

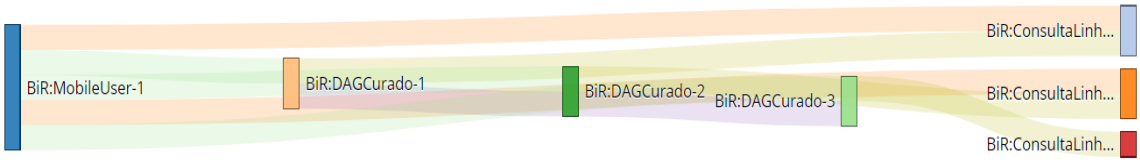

Figura 10. Diagrama de Sankey descrevendo a proveniência relacionada ao agente BiR:*MobileUser-1*.

A Figura 11 indica um diagrama do tipo Hive que complementa a Figura 10. Por exemplo, os arcos coloridos indicam os cinco tipos de arestas existentes partindo do agente BiR:*MobileUser-1* que consulta os dados currados através de três interações com o sistema.

As arestas na cor verde claro indicam os três relacionamentos do tipo prov:*wasAssociatedWith* entre o agente BiR:*MobileUser-1* e dados curados (*e.g.* BiR:DAGCurado-1) gerados arquitetura *BusInRio* utilizados a cada consulta disparada pelo usuário através do ap0licativo móvel. As arestas *links* na cor laranja indicam o relacionamento do tipo prov:*wasAttributedTo* entre o mesmo agente e os arquivos dados curados produzidos pelo *workflow* ETL. Ressaltamos que apesar de existirem três consultas se verifica a inexistência de uma terceira aresta na cor laranja. Isso indica que houve um erro de processamento capturado pelo sistema de fila da arquitetura e registrado no sistema através da ausência da criação do relacionamento de proveniência.

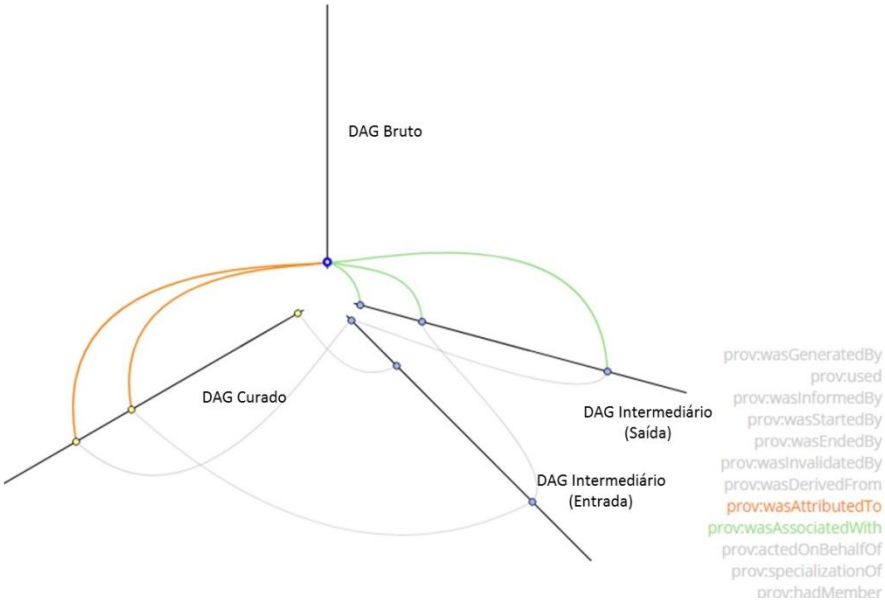

Figura 11. Diagrama do tipo Hive ilustrando a proveniência retrospectiva das consultas aos dados currados ao agente BiR:*MobileUser*.

5.4 Experimentos Sobre a Satisfação dos Usuários

A experimentação de *softwares* oferece um modo sistemático, computável e controlado para avaliar a atividade humana. A avaliação experimental das funcionalidades da *BusInRio* junto aos seus usuários foi dividida em duas partes. A primeira consistiu de uma avaliação do tipo *survey* (Travassos *et al.,* 2002 e Wohlin *et al.,* 2012), onde os usuários expressavam diretamente suas opiniões sobre o uso do aplicativo. A segunda

avaliação foi de natureza quantitativa, onde analisaram-se os dados de proveniência gerados pelos usuários durante as sessões de utilização do aplicativo.

A seleção dos usuários baseou-se nas recomendações de Ferris *et al.* (2010) para o desenvolvimento de aplicações de mobilidade urbana voltadas para cidades inteligentes. Foram selecionados aleatoriamente usuários, todos eram estudantes universitários ou secundaristas. Para evitar vieses analíticos, todos os selecionados eram passageiros frequentes de ônibus e possuíam experiência no uso de rede Internet em dispositivos móveis. O período de testes do sistema foi definido como sendo 14 dias, onde os usuários poderiam responder ao *survey* e avaliar a ferramenta.

Justifica-se a adoção do *survey* nas primeiras avaliações experimentais pelos seguintes fatos: (i) permite investigação em retrospectiva e de caráter exploratório sendo capaz de coletar informações qualitativas e quantitativas fornecidas diretamente pelos usuários; (ii) utiliza instrumentos simples do tipo questionários de autoaplicação dirigidos a grupos focais.

Paralelamente, realizamos um segundo estudo experimental com os mesmos usuários para estimar o tempo médio de utilização da ferramenta e o tempo médio que os usuários consumiam para executar operações básicas do aplicativo *BiR* em seus *smartphones*. Este tipo de estudo utiliza dados de proveniência coletados pela ferramenta.

Os dados de proveniência foram coletados de modo transparente, diretamente no sistema de mensagens da arquitetura (filas *pull* e *push*) e sem conhecimento prévio dos usuários participantes do estudo. Os *timestamps* são registrados pela porção servidora da aplicação durante as interações dos usuários no período de testes e registrados no servidor MongoDB, os dados possuem a estrutura discutida no item 4.6.

5.4.1 Caracterização dos Participantes dos Experimentos

Primeiramente, para prover um mínimo de significância estatística dos experimentos, selecionamos aleatoriamente 36 pessoas dentre uma população de 57 voluntários. Os participantes dos experimentos são usuários frequentes do sistema de transporte de ônibus públicos. Eles forneceram seus dados demográficos livremente antes da primeira utilização do *BusInRio,* todos são moradores das cidades do Rio de Janeiro**,** Seropédica**,** Itaguaí ou de Nova Iguaçu.

Tabela 2. Distribuição de gêneros e faixas etárias dos participantes dos experimentos.

Faixa Etária	Masculino	Feminino	Sub Totais
17-22 anos	11	8	**19 (52,78%)**
23-26 anos	10	7	**17 (47,22%)**
Totais	**21**	**15**	**36 (100,00%)**

O conjunto de participantes é composto por 52,78% de pessoas do gênero masculino e 47,22 % do gênero feminino, sendo distribuídos de acordo com faixas etárias apresentadas na Tabela2.

Os participantes utilizaram diariamente *smartphones* com configurações distintas de *hardware*, porém todos com o sistema Android na versão 4.0 ou superior. Também se verificou que todos os participantes já possuíam conhecimento sobre utilização da Internet. Com relação a frequência de uso das linhas de ônibus verificamos

que aproximadamente 72% dos participantes fazem duas viagens de ônibus por dia e aproximadamente 28% fazem até quatro viagens entre suas residências e sua instituição de ensino.

5.4.2 Avaliações de Satisfação

As avaliações de utilização do aplicativo *BusInRio* foram realizadas através de questionário de autoaplicação. Os participantes eram convidados a preencher o questionário de satisfação após as sessões de utilização do aplicativo (pós-avaliação). O questionário foi concebido para ser curto, sendo composto de apenas quatro perguntas diretas. As questões e a escala de *Lickert* utilizadas nos questionários estão representadas na Quadro 1.

As perguntas do questionário de avaliação são de natureza quantitativa e do tipo múltipla escolha, exceto a última questão onde os respondentes expressavam-se livremente com vistas a trazer novas contribuições, opiniões ou informações adicionais para a pesquisa. As questões foram elaboradas para se obter um retorno sobre a satisfação de uso após cada utilização do aplicativo.

Como as quatro questões iniciais do questionário eram fechadas e baseadas em uma escala de *Likert,* foi possível analisar estatisticamente as amostras fornecidas pelos respondentes (Spiegel, 2009). Os respondentes relataram através dos questionários que 38,61% usaram o *BiR* várias vezes ao dia, enquanto que uma 30,91% informaram usá-lo uma vez e 23,33% ocasionalmente e 7,25% nunca utilizaram o aplicativo durante o período de testes.

Com relação a atribuição de uma nota do aplicativo 28,32% o consideraram "muito satisfatório", 41,24% o consideraram "satisfatório" e 20,42% consideram "indiferente" e 10,13% "não satisfeito". Segundo os resultados obtidos, verificou-se que mais de 2/3 das avaliações sobre a utilização do aplicativo foram consideradas positivas.

Com relação a precisão, 27,31% revelaram que a aplicação é muito precisa com relação a localização e deslocamento dos ônibus, 44,63% precisa, 18,13% como neutro e 10,23% como não precisa. Verificou-se que a maioria, aproximadamente 5/4 dos respondentes, consideraram como boa a precisão do aplicativo. Por fim, 60,1% recomendariam o aplicativo para outras pessoas, 27,7% talvez recomende; já 11,1% não recomendariam e apenas 1,2% como nunca recomendariam. Verifica-se que aproximadamente pouco menos de 1/4 dos usuários não recomendariam o sistema.

Quadro 1. Questões e escala de *Lickert* utilizados no experimento de satisfação.

Questão	Escala de Lickert			
Q1- Quantas vezes você utilizou o *BusInRio hoje*?	várias vezes ao dia	duas vezes ao dia	uma vez ao dia	Nunca
Q2- Qual a nota você daria para essa sessão do *BusInRio*?	muito satisfeito	Satisfeito	pouco satisfeito	nada satisfeito
Q3- Qual a precisão do *BusInRio*?	muito preciso	Preciso	pouco preciso	muito impreciso
Q4- Você recomendaria o *BusInRio* para algum amigo(a)?	sim	Talvez	não	Jamais

A Figura 12 apresenta um gráfico do tipo *tree map* que ilustra as distribuições das respostas dos questionários de autoaplicação. A escala de *Linkert* foi convertida em escala de cores que varia do verde ao vermelho, sendo que as respostas consideradas positivas são representadas em tons de verde e as negativas em tons de vermelho.

Figura 12. Representação gráfica do tipo *tree map* com as distribuições percentuais das respostas dos questionários de autoaplicação.

5.4.3 Avaliações de Tempos De Repostas Baseados Em Dados de Proveniência

Esta subseção apresenta as avaliações de tempos médios de utilização e resposta da arquitetura baseados em dados de proveniência coletados pelo sistema de filas.

As aferições preliminares de tempos médios não têm como objetivo fazer o *benchmark* completo do sistema executando em ambiente de nuvem com os testes de carga, *stress* dos servidores, latência da rede, escalabilidade e tolerância a falhas. Pelo contrário, por contarmos com poucas instâncias de baixo custo além de *smartphones* com configurações de *hardware* distintos não seria viável realizar um experimento controlado dessa natureza. Se buscou utilizar os dados de proveniência retrospectiva para compreender o comportamento dos usuários frente as requisições, erros e visualizações dos mapas, estabelecendo uma linha de base mínima da capacidade do sistema.

Os dados de proveniência retrospectiva utilizados neste experimento foram coletados diretamente pelo sistema de mensagens da arquitetura (filas *pull* e *push*) sendo armazenados como documentos JSON no MongoDB com as estruturas apresentadas na subseção 4.6.

Durante o período de avaliações foram registradas 1.115 sessões de usuários do *BusInRio*, sendo que 1.048 sessões (93,99%) foram finalizadas com sucesso e apenas 67 sessões (6,01%) foram incorretas. A Tabela 3 apresenta o quantitativo e os tipos de erros ocorridos distribuídos durante o período de avaliação. Vale ressaltar que se verificou uma pequena assimetria na distribuição das sessões ao longo do tempo. Aproximadamente 53,43% do total de sessões ocorreram na primeira semana de avaliação e 46,57% demais na segunda semana do experimento.

A Figura 13 ilustra a natureza dos erros no período avaliado. Ressalta-se que a maior parte dos erros apresentados pela plataforma são oriundos de problemas de conexão do dispositivo móvel com a rede de telefonia ou erros de sincronismo de relógio entre o *smartphone* e o horário dos servidores na nuvem, este tipo de erro impede a recuperação correta dos ônibus que atuam em uma linha. A taxa de 4% de erros relacionados a "dado indisponível" é um indício interessante, indica que a arquitetura é capaz de atender a maioria das requisições dos usuários participantes.

Tabela 3. Distribuição dos erros ao longo das duas semanas de avaliação.

Tipo de Erro	Semana 1	Semana 2	Subtotais
Falha conexão	28	23	**50**
Processamento	2	2	**4**
Relógio atrasado	6	5	**11**
Dado indisponível	1	1	**2**
Total	**37**	**30**	**67**

Ao aprofundarmos as análises dos dados de proveniência, verificou-se que 82,42 % de todas as seções, ou seja, a maioria absoluta das sessões, ocorreram durante dias úteis e apenas 17,58 % ocorreram em finais de semana. Ao comparar esses dois conjuntos de valores estima-se que a maioria das sessões ocorreram no trajeto entre a residência dos usuários e a Universidade.

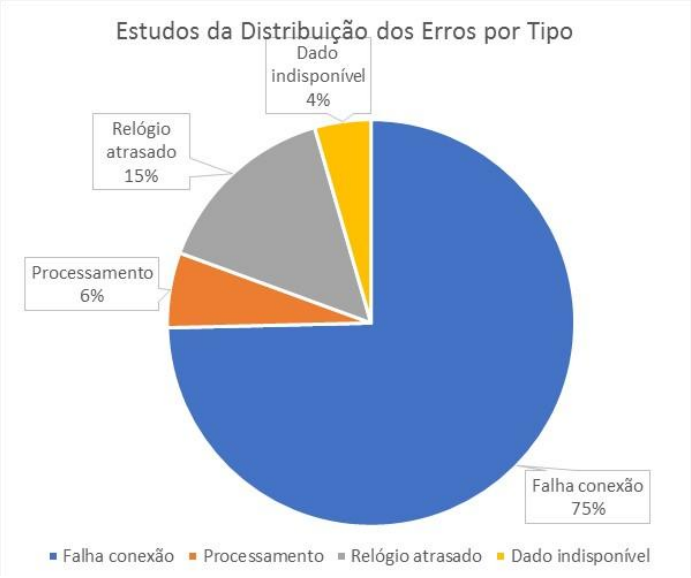

Figura 13. Gráfico do tipo pizza com as distribuições percentuais dos erros ocorridos nas sessões do *BusInRio* durante o período avaliado.

A Tabela 4 apresenta a distribuição das sessões e erros ocorridos durante o período de avaliação. A Figura 14 ilustra a mesma distribuição em termos percentuais.

Verificou-se que o tempo médio de espera da abertura de conexão do aplicativo com a nuvem e recuperação das linhas de ônibus foi de apenas 2,1 segundos. Estes tempos médios pouco variam durante as duas semanas, indicando que o tempo de conexão independe da carga de trabalho da arquitetura.

Tabela 4. Distribuição dos erros ao longo da avaliação.

	Sessões em Dias Úteis	Sessões em Fins de Semana	SubTotais
Sessões sem erros - Semana 1	462	120	582
Sessões com erros - Semana 1	30	7	37
Sessões sem erros - Semana 2	401	65	466
Sessões com erros - Semana 2	26	4	30
Distribuição das Sessões	**919**	**196**	**1115**
% de Sessões por semana	**82,42%**	**17,58%**	**100%**

Verificou-se que o tempo médio de utilização do aplicativo pelos usuários se reduziu ao longo do experimento. Os cálculos dos tempos médios baseiam-se em *timestamps* dos dados de proveniência das sessões. Por exemplo, o cálculo do tempo de utilização (T_U) se apresenta na expressão (1). Esta expressão representa o intervalo de tempo transcorrido entre a última consulta (requisição do formulário) e a primeira interação do usuário do sistema (primeira consulta à uma linha de ônibus). O cálculo do tempo médio de utilização (T_{MU}) na expressão (2).

Expressão 1

T_U = *timestamp* da requisição do formulário − *timestamp* primeira consulta ao aplicativo

Expressão 2

$T_{MU} = \Sigma\, T_U$ / total de sessões sem erro na semana

Ao longo primeira semana, o T_{MU} do aplicativo foi de um minuto e quarenta e cinco segundos já o valor do T_{MU} na segunda semana foi de apenas um minuto e doze segundos. Credita-se essa redução do tempo médio ao autoaprendizado dos usuários, simplicidade da interface do aplicativo e pela também redução da ordem de 78 % das seções durante a segunda semana de avaliações.

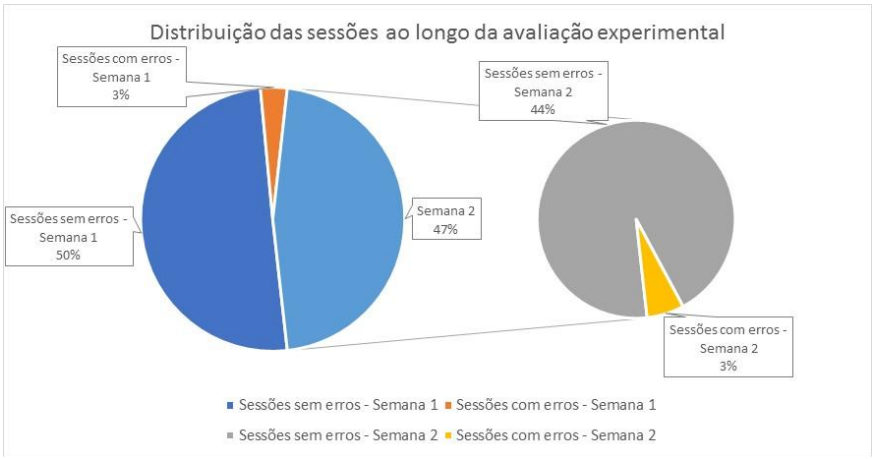

Figura 14. Gráfico do tipo pizza com as distribuições percentuais das sessões durante o período do experimento.

O tempo médio de navegação (T_N) também utiliza dados de proveniência, ele avalia quanto tempo os usuários aguardavam por uma resposta de uma requisição (consulta de ônibus) no aplicativo. O cálculo do tempo de navegação também é direto,

ele se baseia na expressão (3) e o cálculo do tempo médio de navegação (T_{MN}) na expressão (4).

Expressão 3

T_N = *timestamp* da última resposta de uma requisição de dado – *timestamp* da primeira requisição de dado

Expressão 4

$T_{MN} = \Sigma T_N$ / total de sessões sem erro na semana

Avaliamos os intervalos de tempo entre a escolha uma linha de ônibus na interface, submetê-la ao sistema, processá-la na nuvem, receber as respostas e a representação dos resultados em um mapa no *smartphone.* O tempo médio transcorrido em todo esse processo foi de 8,6 segundos na primeira semana e reduzindo para 6,4 segundos na segunda semana. Acredita-se que essa redução deve-se não só ao ganho cognitivo dos usuários em relação a interface, como também a redução da quantidade de seções da primeira para a segunda semana.

6.Trabalhos Relacionados

Em diversas cidades do mundo podem ser encontrados aplicativos sobre mobilidade urbana. Inúmeras cidades publicam uma variedade de dados abertos através das suas autoridades de trânsito para que desenvolvedores, pesquisadores, planejadores urbanos, jornalistas de dados e *startups* os utilizem.

Atualmente existem diversos aplicativos comerciais de classe mundial que auxiliam o cidadão a planejar seus deslocamentos pelas cidades. Por exemplo, o aplicativo *Citymapper*[19] está integrado ao sistema de metrô, trem e ônibus de Londres. Em Nova Iorque e em outras cidades americanas existe o aplicativo *OneBusAway*[20]que apresenta informações de transporte público integrando dados abertos sobre os diversos tipos de modais de transporte.

Moovit[21], Chariot[22] e o Trafi[23] são aplicativos comerciais, disponíveis em várias cidades do mundo, que oferecem aos seus usuários serviços de navegação e planejamento de viagens de ônibus, metrô e trem. Por serem produtos protegidos, nossas investigações não foram capazes de reunir informações suficientes sobre suas arquiteturas ou algoritmos utilizados. No entanto, constatou-se através da literatura (Thorpea, Namdeo, 2016 e Heiskala *et al.,* 2016) que tais aplicativos diferenciam-se da nossa proposta, pois são baseados em serviços são do tipo *crowd sensing*, onde não só há utilização dos dados oriundos das autoridades de trânsito, como também ocorrem bonificações pela utilização dos serviços em troca das informações fornecidas pelos usuários.

Algumas cidades no Brasil começam a disponibilizar aplicativos comerciais que utilizam dados abertos governamentais (*RioBus, Buus*, Cadê o Onibus?)[24]. O aplicativo *RioBus* é um sistema colaborativo de monitoramento de ônibus em tempo real que

[19] https://citymapper.com/
[20] http://onebusaway.org/
[21] http:// https://www.moovitapp.com/
[22] https://www.chariot.com/
[23] http://trafi.com
[24] Desenvolvedores: Tormenta Labs; Buus Serviços de Tecnologia; Luis Picanço (respectivamente).

provê algumas funcionalidades semelhante as primeiras versões do *Buus*. Os aplicativos utilizam as *APIs* de DAG de mobilidade urbana disponibilizada pela Prefeitura da Cidade do Rio de Janeiro. Ressaltamos que nenhum dos aplicativos supracitados provêm maiores informações técnicas, publicações ou maiores detalhes sobre a arquitetura da plataforma, algoritmos utilizados, estrutura de dados nem mesmo sobre o funcionamento pois tratam-se de programas protegidos por interesses comerciais.

O sistema *UbiBusRoute* (Tito *et al.,* 2012) tem como foco a Cidade do Recife (PE), está baseado em um *middleware* que faz uso de informações contextuais de *tweets* e de computação ubíqua para se adaptar às situações do trânsito e recomendar rotas alternativas aos usuários de ônibus. O sistema Busão (Leite *et al.,* 2013) desenvolvido com base nos dados da cidade de Campina Grande (PB) oferece os itinerários dos ônibus e as melhores rotas dos ônibus aos usuários. O sistema utiliza técnicas de geoprocessamento baseada na localização do usuário para indicar as melhores rotas dos ônibus. Ambos os sistemas utilizam a capacidade de processamento dos *smartphones* dos usuários, não utilizam DAG, mas dados dos GPS coletados e compartilhados pelos passageiros.

Barbosa *et al.* (2014), desenvolveram o Vistradas, uma proposta mais elaborada que os trabalhos anteriores. O sistema utiliza técnicas de visualização de dados abertos sobre as viagens de ônibus no município do Rio de Janeiro, os autores aplicam diferentes técnicas de análise de dados para tratar os DAG e oferecem visualização das trajetórias de ônibus do município. O trabalho tem como foco auxiliar o gestor municipal a avaliar a uniformidade e a qualidade dos serviços dos ônibus do município em função dos grandes eventos sediados na cidade.

Bessa *et al.* (2015), também exploram a visualização de dados abertos sobre os transportes através da ferramenta de visualização *RioBusData*. O trabalho tem como principal característica propor métodos de detecção de *outliers* em catálogos de dados de transporte público através de uso de redes neurais do tipo *Convolutional Neural Networks* (CNN).

Diferentemente do *BusInRio*, as duas soluções anteriores são exclusivamente voltadas para auxiliar os gestores urbanos na visualização as rotas do ônibus. Os trabalhos não discutem os aspectos arquiteturais ou funcionais das soluções apresentadas do ponto de vista do usuário do sistema e não relatam possíveis formas de integração com aplicativos móveis. Por fim, diferentemente da nossa proposta, nenhum dos trabalhos relacionados não consideram o importante papel do enriquecimento dos DAG com proveniência.

7. Conclusões, Limitações e Trabalhos Futuros

Cidades inteligentes são esforços resultantes de conjuntos de ações que buscam melhorar da qualidade de vida da população com melhores serviços e auxiliar os gestores através da oferta de ferramentas que ofereçam novos usos aos dados públicos.

Diversas soluções baseadas em DAG têm sido propostas para melhorar o problema da mobilidade urbana em várias partes do globo, no entanto ainda não há uma solução ou arquitetura definitiva que atenda a todas as cidades ou mesmo estudo seminais sobre o tema.

Neste artigo apresentamos um novo protótipo da arquitetura *BusInRio* que difere dos trabalhos relacionados, ela está totalmente baseada em *workflows* ETL que processam dados (DAG brutos de mobilidade urbana) da Cidade do Rio de Janeiro enriquecendo-os com proveniência retrospectiva representada segundo a especificação PROV da W3C.

As regras de limpeza de dados foram implementadas sob a forma de *workflows* ETL. Elas mostraram-se adequadas, justifica-se a afirmativa devido aos fatos de que foi possível elaborar diversos tipos de grafos de proveniência apresentando exemplos de casos reais de processamento e ainda que dentre as sessões de usuários, poucas foram errôneas e causadas por ausência de dados.

Com relação aos repositórios *NoSQL*, verificou-se que o MongoDB foi estável no que diz respeito a velocidade de resposta das consultas do aplicativo. Como os processos de captura, etiquetagem dos DAG curados e a gerência dos dados de proveniência retrospectiva podem ser considerados como um problema de *Big Data* consideramos que o uso do MongoDB se mostrou adequado, pois é otimizado para múltiplas escritas e acessos concorrentes as bases de dados e de proveniência. Destacamos que para avaliar o desempenho dos módulos da arquitetura serão necessários testes suplementares com mais usuários simultâneos.

Destacamos que desenvolver e avaliar aplicativos móveis para cidades inteligentes ainda é um desafio em aberto, pois não há referenciais teóricos ou testes consolidados na literatura para aferir seu grau de aceitação por parte dos usuários. Muitas das ferramentas são baseadas em ferramenta de *analytics* oferecido pelas lojas de apliacativos (*e.g. Google* e *Apple store*). Adicionalmente, ressaltamos que existem poucos trabalhos que explorar os possíveis usos de dados de proveniência em Cidades Inteligentes.

Este artigo apresentou uma ampliação da arquitetura e o protótipo do aplicativo *BusInRio* capaz de oferecer DAG curados e enriquecidos com proveniência para seus usuários. Além disso, apresentamos dois pequenos conjuntos de experimentos que avaliaram sua utilização. Os experimentos utilizaram tanto os dados fornecidos por usuários quando dados de proveniência retrospectiva coletados pela própria arquitetura. A proposta, diferentemente dos trabalhos relacionados é capaz de registrar de modo transparente tanto as atividades de produção de DAG curado quando sua posterior utilização.

Os resultados preliminares obtidos através do *survey* são significativos e indicam que houve boa aceitação da proposta por parte dos usuários. No entanto, apesar de positivos, não podem ser generalizados para qualquer classe de usuários ou para qualquer tipo de *smartphone* ou mesmo para qualquer cidade brasileira. Faz-se necessário aprofundar os testes e ampliar o número de usuários do sistema para avaliar as questões de desempenho da arquitetura em situações de centenas de consultas por segundo ou mesmo adotar outras metodologias de avaliação, tais como o modelo TAM (*Technology Acceptance Model*).

Este trabalho não se encerra em si mesmo, ainda restam várias perspectivas a serem avaliadas. Como trabalhos futuros, sugere-se o desenvolvimento de um vocabulário para aplicações voltadas para cidades inteligentes, ele poderá apoiar o desenvolvimento de aplicações semânticas que podem auxiliar tanto o cidadão quanto ampliar a transparência da gestão pública. Além disso, novos experimentos são

sugeridos, por exemplo, aprofundar as análises quantitativas das regras ETL sobre o tratamento de DAG. Por fim, se pretende ampliar a arquitetura para que esta seja capaz de integrar dados de outros meios de transporte público (BRT, metrô e trens). Também se vislumbra: (i) avaliar novos sistemas *NoSQL* tais como o MonetDB[25] ou o AsterixDB[26]; (ii) desenvolver novos algoritmos que permitam que os usuários obtenham uma estimativa sobre os melhores percursos e rotas comparando com outros modais de transportes públicos na Cidade do Rio de Janeiro (iii) avaliar a arquitetura com um número maior de usuários para verificar tanto a utilidade quanto a facilidade percebida de uso; (iv) aprofundar os estudos sobre o vocabulário BiR e padrões utilizados em aplicações de cidades inteligentes.

Agradecimentos

Os autores agradecem ao programa ao Programa de Educação Tutorial (PET), PET-SI do MEC/SeSu pelo apoio ao FNDE, CNPq e FAPERJ pelos financiamentos parciais e bolsas concedidas durante a realização deste trabalho. Além disso, os agradecem à Red CYTED (smartlogistics@ib) pelos financiamentos externos. Agradecemos aos avaliadores do trabalho pelas valiosas sugestões de aprimoramentos do texto. Por fim, agradecemos a Luan Andrade por participar do desenvolvimento da primeira versão do *BusInRio*.

Referências Bibliográficas

Abramova, V., Bernardino, J. (2013). "*NoSQL* Databases: MongoDB vs Cassandra". In: Proceedings of the International C* Conference on Computer Science and Software Engineering, pp. 14–22. [GS Search]

Alawadhi, S.; Scholl, H. J. (2013). "Aspirations and Realizations: The Smart City of Seattle". In: Proceedings of the 46th Hawaii International Conference on System Sciences, Wailea, pp. 1695-1703. [GS Search]

Andrade L. S.; Cruz, S. M. S. (2015) "*BusInRio*: Explorando Dados Abertos de Transporte Público do Município do Rio de Janeiro: In: II Escola regional de Sistemas de Informação do Rio de Janeiro, pp. 53-60. [GS Search]

Batista D. M. B.; Goldman, A.; Hirata Jr., R; Kon, F.; Costa, F. M.; Endler M. (2016). "InterSCity: Addressing Future Internet Research Challenges for Smart Cities". In: 7th IEEE International Conference on Network of the Future. 6 pp. [GS Search]

Barbosa. L.; Kormaksson, M.; Vieira, M. R.; Tavares, R. L.; Zadrozny. B. (2014). "Vistradas: Visual Analytics for Urban Trajectory Data". In: 15th Brazilian Symposium on Geoinformatics. [GS Search]

Bertino, E., (2013). "Challenges and Opportunities with Big Data". In: IEEE 37th Annual Computer Software and Applications Conference pp. 479-480. doi: 10.14778/2367502.2367572.

Bessa, A.; Silva F. M.; Nogueira R. F.; Bertini. E, Freire, J. (2015). "RioBusData: Visual Data Analysis of Outlier Buses in Rio de Janeiro". In: Symposium on

[25] https://www.monetdb.org/
[26] https://asterixdb.apache.org/

Visualization in Data Science. https://arxiv.org/pdf/1601.06128.pdf. Acesso em janeiro de 2017.

Buneman, P., Khanna, S. e Chiew, W. (2001). "Why and Where: a Characterization of Data Provenance. In: ICDT'01, 8th International Conference on Database Theory, LNCS, v.1973, pp. 316–330. [GS Search]

Caragliu, A; Del Bo, C.; Nijkamp P. (2011). "Smart Cities in Europe". *Journal of Urban Technology*. vol 18 (2), pp. 65-82. https://doi.org/10.1080/10630732.2011.601117

Chee, B. J. S; Franklin Jr., C. (2013). "Computação em Nuvem – Tecnologias e estratégias" M Books: Brasil.

Corsar D.; Edwards, P. (2012). "Enhancing Open Data with Provenance". In: Digital Futures. Aberdeen, UK. 3 pp. [GS Search]

Cruz, S.M.S; Campos, S.M.S., Mattoso, M. (2009). "Towards a Taxonomy of Provenance in Scientific *Workflow* Management Systems", In: Congress on Services, IEEE, pp. 259–266. [GS Search]

Cruz, S. M. S.; Andrade L. S.; Oliveira, J. (2016) "Explorando Dados Abertos Governamentais sobre a mobilidade Urbana na Cidade do Rio de Janeiro: In: 43 Seminário Integrado de Software e Hardware, SEMISH 2016. [GS Search]

Dados Abertos Governamentais (2011) "Manual dos dados abertos: governo - traduzido e adaptado de opendatamanual.org" http://www.w3c.br/pub/Materiais/PublicacoesW3C/Manual_Dados_Abertos_WEB.pdf. Acesso em janeiro de 2017.

Darus, M. Y.; Bakar, K. A. (2013). "Congestion control algorithm in vanets". *World Applied Sciences Journal*, 21(7), pp. 1057–1061. doi: 10.5829/idosi.wasj.2013.21.7.242

Davies, T.; Edwards, D. (2012). "Emerging Implications of Open and Linked Data for Knowledge Sharing in Development". IDS Bulletin, v. 43, pp. 117–127. [GS Search]

Democracia Digital (2015). "Projeto Democracia Digital vol. 3: Dados Abertos nos municípios, estados e governo federal brasileiro". http://bibliotecadigital.fgv.br/dspace/bitstream/handle/10438/16373/Dados_Abertos_nos_Munic%C3%ADpios_Estados_e_Governo_Federal_Brasileiros_Volume_3.pdf?sequence=1&isAllowed=y.

Dohler, M.; Vilajosana, I.; Vilajosana, X.; Llosa, J. (2011). "Smart Cities: An action plan," In: Proceedings of Barcelona Smart Cities Congress, Barcelona, Spain, pp. 1–6. http://www.futureenterprise.eu/sites/default/files/SmartCityPaper.pdf

Eaves, D. (2009). "The Three Laws of Open Government Data". https://eaves.ca/2009/09/30/three-law-of-open-government-data/.

Emaldi M.; Penã, O.; Lázaro J.; López-de-Ipinã, D.; Vanhecke S.; Mannens E. (2013). "To trust, or not to trust: Highlighting the need for data provenance in mobile apps for smart cities". In: Proceedings of the IMMoA, pp. 68-71. [GS Search]

Ferreira, G. R.; Filipe Jr. C.; Oliveira D (2014). “Uso de SGBDs *NoSQL* na Gerência da Proveniência Distribuída em *Workflows* Científicos”. In 29 Simpósio Brasileiro de Banco de Dados, pp. 187-196. [GS Search]

Ferris, B.; Watkins, K.; Borning, A. (2010). “Location-Aware Tools for Improving Public Transit Usability” Pervasive Computing v.9 (1), pp. 13-19. [GS Search]

Freire, J.; Koop, D.; Santos, E.; Silva, C. T. (2008). “Provenance for Computational Tasks: A Survey”. *Journal Computing in Science and Engineering*. V.10. N. 3, pp. 11-21. doi:10.1109/MCSE.2008.79

Giffinger, R.; Haindlmaier, G. (2010). “Smart Cities Ranking: An Effective Instrument For The Positioning Of Cities?” ACE: Architecture, City and Environment. Ano IV, 12 Febrero, pp. 7-25. [GS Search]

Gil, Y. *et al.* (2010). "Provenance XG Final Report", W3C, www.w3.org/2005/Incubator/prov/XGR-prov.

Gonçalves, R. R.; Viterbo, J.; Sousa, P. C (2016) “Um estudo preliminar sobre a o uso de dados abertos na implementação de serviços para cidades inteligentes”. In: II Workshop de Pesquisa e Desenvolvimento em Inteligência Artificial, Inteligência Coletiva e Ciência de Dados - 2016 - Niterói, RJ, pp. 122 – 131. [GS Search]

Guimarães V.; Hondo F.; Almeida R.; Vera H.; Holanda, M.; Walter M. E.; Lifschitz S. (2015). “A study of genomic data provenance in NoSQL document-oriented database systems”. In: IEEE International Conference on Bioinformatics and Biomedicine, pp. 1525-1531. [GS Search]

Heiskala, M.; Jokinen J.-P.; Tinnilä, M. (2016) “Crowdsensing-based transportation services—An analysis from business model and sustainability viewpoints”, *Research in Transportation Business and Management*. V. 18, pp. 38-48. [GS Search]

Hartig, O. (2009). “Provenance Information in the Web of Data”. http://ceur-ws.org/Vol-538/ldow2009_paper18.pdf.

Hernández-Muñoz, J. M.; Vercher, J. B.; Muñoz, L.; Galache, J. A.; Presser, M.; Hernández Gómez, L. A.; Pettersson J. (2011). “Smart Cities at the forefront of the future Internet” The Future Internet, LNCS., v. 6656, pp. 447–462. https://link.springer.com/chapter/10.1007/978-3-642-20898-0_32

Hollands, R.G. (2008). “Will the real smart city please stand up? Intelligent, progressive or entrepreneurial?” *City*, 12 (3), pp. 303-320. [GS Search]

Imran, M.; Hummel K. A (2008) “On using provenance data to increase the reliability of ubiquitous computing environments” In: Proceedings of the 10th International Conference on Information Integration and Web-based Applications, pp. 547-550. [GS Search]

Kon, F. Santana, E. F. Z. (2016) “Cidades Inteligentes: Conceitos, plataformas e desafios”. In: JAI-CSBC, 2016, Porto Alegre. [GS Search]

Leite, D. F. B; Rocha, J. H.; Baptista, C. S. (2013). “Busão: um Sistema de Informações Móvel para Auxílio à Mobilidade Urbana Através do Uso de Transporte Coletivo”. In: SBSI, 2013, João Pessoa. [GS Search]

McArdle, G.; Kitchin, R. (2015) “Improving the Veracity of Open and Real-Time Urban Data”. The Programmable City Working Paper 13. http://dx.doi.org/10.2139/ssrn.2643430.

Miranda, C. M. C. (2011). “A Disseminação de Dados Governamentais como Serviço Público” – Os Dados Abertos Governamentais e a Experiência Brasileira Dados Abertos para a Democracia na Era Digital. Fundação Alexandre Gusmão, 2011. [GS Search]

Mendonça, R. R.; Cruz, S. M. S, Campos, M. L. M (2014). “Gerência de Proveniência Multigranular em Linked Data com a Abordagem ETL4LinkedProv”. In: SBBD, 2014 Curitiba. [GS Search]

Mendonça, R. R.; Cruz, S. M. S, Campos, M. L. M (2016). ETL4LinkedProv: Managing Multigranular Linked Data Provenance. *Journal Of Information and Data Management*. vol. 7(2): pp. 70-85. [GS Search]

MongoDB (2017). https://docs.mongodb.com/manual/.

Moreau, L. *et al.* (2013) “PROV-DM: The PROV Data Model”. W3C Recommendation REC-prov-dm-20130430.

Nam, T.; Pardo, T. (2011). “Conceptualizing Smart City with Dimensions of Technology, People, and Institutions”. In: Proceedings of the 12th Annual International Conference on Digital Government Research, College Park, Maryland, pp. 282 – 291. doi: 10.1145/2037556.2037602

Ojo, A.; Curry, E.; Zeleti, F. (2015). “A Tale of Open Data Innovations in Five Smart Cities”. In: Proceedings of 48th Hawaii International Conference on System Sciences, Kauai, HI, USA: IEEE Computer Society. doi:10.1109/HICSS.2015.280

Oliveira, L.E.R.A.; Lóscio, B. F. (2014). “Uma Abordagem Para Captura De Informações Sobre Aplicações Que Fazem Uso De Dados Abertos” Revista Brasileira de Administração Científica V5 (2), pp. 127-140. doi:10.6008/SPC2179-684X.2014.002.0010

Oliveira, W.; Oliveira D.; Braganholo, V. (2014) “Experiencing PROV-Wf for Provenance Interoperability in SWfMSs” In: Proceedings of the 6th International Provenance and Annotation Workshop on Provenance and Annotation of Data and Processes, pp. 294-296. [GS Search]

ODaF – Open Data Foundation (2016). http://www.opendatafoundation.org/. Acesso em janeiro de 2017.

OKF - Open Knowledge Foundation (2017). https://okfn.org/. Acesso em janeiro de 2017.

Picone, M.; Amoretti, M.; Zanichelli, F. (2012). “Simulating smart cities with DEUS”. In: Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques, pp. 172–177. [GS Search]

Pimentel, J. F.; Freire, J. Braganholo, V.; Murta, L. (2016) “Tracking and Analyzing the Evolution of Provenance from Scripts”. In: Proceedings of the 6th International Provenance and Annotation Workshop on Provenance and Annotation of Data and Processes – v. 9672, pp. 16-28. doi:10.1007/978-3-319-40593-3_2

PBDA - Portal Brasileiro de Dados Abertos. (2017). “O que são dados abertos?", https: http://dados.gov.br/paginas/dados-abertos. Acesso em: janeiro de 2017.

Salfer, M.; Wohgemuth, S.; Schrittwieser S.; Bauer, B. (2011). “Data Provenance with Watermarks for Usage Control Monitors at Disaster Recovery”. In: International Conference on and 4th International Conference on Cyber, Physical and Social Computing. doi:10.1109/iThings/CPSCom.2011.129

Santana, E. F. Z.; Chaves, A. P.; Gerosa, M. A.; Kon, F.; Milojicic D. S. (2016). “Software Platforms for Smart Cities: Concepts, Requirements, Challenges, and a Unified Reference Architecture”. http://arxiv.org/abs/1609.08089.

Schaffers, H. *et al.* (2011). “Smart cities and the future internet: Towards cooperation frameworks for open innovation”. The Future Internet, LNCS., v. 6656, pp. 431–446. https://link.springer.com/chapter/10.1007/978-3-642-20898-0_31

Simmhan, Y. L., Plale, B., Gannon, D., (2005). “A survey of data provenance in e-science”. SIGMOD Record, v. 34, n. 3, pp. 31-36. doi: 10.1145/1084805.1084812

Spiegel M. R. (2009). “Estatística” - Coleção Schaum, Bookman: Brasil.

Tito, A. O., *et al.* (2012). “UbiBus: Um Sistema de Informações Inteligentes para Transporte Público”. In: Workshop Tecnologias da Informação e Comunicação nos Grandes Eventos Esportivos (WTICEE), Aracaju-SE.

Thorpea, N.; Namdeob, A. (2016). “Innovations in Technologies for Sustainable Transport”. Research in Transportation Business and Management. V. 18, pp. 1-3. [GS Search]

Travassos, G. H.; Gurov, D.; Amaral. E. A. G. (2002). “Introdução à Engenharia de Software Experimental”, Relatório Técnico – RT-ES-590/02. COPPE/UFRJ. [GS Search]

Tygel, A. F.; Gonçalvez, L.; Santos. M., Marques., G.; Campos M. L. M. (2015). “Informação para Ação: Desenvolvimento de um Portal de Dados Abertos Sobre Agrotóxicos”. Revista Tecnologia e Sociedade, Curitiba, v.11(22) pp.99-119. [GS Search]

Vieira, S. (2009). “Como elaborar questionários”. Atlas: São Paulo.

Wang, G., Tang, J. (2012). "The *NoSQL* Principles and Basic Application of Cassandra Model". In: International Conference on Computer Science Service System pp. 1332–1335. [GS Search]

Zuiderwijk, A.; Janssen, M.; Davis, C. (2014a). “Innovation with open data: Essential elements of open data ecosystems”. Information Polity, 19(2), pp. 17-33. [GS Search]

Zuiderwijk, A.; Janssen, M. (2014b). “Data policies, their implementation and impact: A framework for comparison” In: Government Information Quarterly, v. 31, pp. 17–29. [GS Search]